ibi™ WebFOCUS®

Language リファレンス

バージョン 9.3.0 | 2024 年 4 月

Copyright © 2021-2025. Cloud Software Group, Inc. All Rights Reserved.

目次

目次 **2**

レポート作成の概要 **72**
レポート作成の要件 72
レポートタイプ 73
リクエストの作成 75
リクエストの開始 77
リクエストの開始 77
リクエストの発行時のヘルプの表示 77
リクエストの終了 78
レポート作成の例 78
　単純なレポートの作成 78
レポートのカスタマイズ 80
レポート出力先の選択 81

レポートデータの表示 **83**
リクエストでの表示コマンドの使用 83
リクエストでの表示コマンドの使用 83
個々の値の表示 84
　特定フィールド値の表示 85
　レコードの一覧表示 86
全フィールドの表示 87
　全フィールドの表示 87
全セグメントフィールドの表示 90
全セグメントフィールドの表示 90
　全セグメントフィールドの表示 90
マルチパスデータソース構造とデータ取得順序の表示 91
　マルチパスデータソース構造の表示 92

マルチパスデータソース取得順序の表示 93
マルチパスデータソースのフィールドの表示 94
値の合計 96
値の合計 96
数値以外の合計 97
値の集計 97
値の集計 98
ソート句による値の集計 98
データインスタンス数の集計 99
セグメントインスタンス数の集計 100
マルチパスデータソースのセグメント数の集計 100
COUNT および LIST コマンドの表示桁数拡張 101
COUNT コマンドと LIST コマンド用の表示桁数の設定 101
COUNT コマンドと LIST コマンド用の表示桁数の設定 102
リクエストで使用可能な表示フィールド最大数 103
演算接頭語による表示フィールドの操作 103
演算接頭語の基本 105
演算接頭語の使用 105
演算接頭語の使用上の注意 105
演算接頭語で実行可能な機能 106
フィールド平均値の計算 109
フィールド平均値の計算 109
平方和フィールドの平均計算 110
平方和フィールドの平均計算 110
フィールド最大値と最小値の計算 110
フィールド最大値と最小値の計算 111
フィールドの中央値および最頻値の計算 111
中央値および最頻値の計算 111
行と列のパーセント計算 112
列のパーセント計算 112
行のパーセント計算 113
個数のパーセント計算 114

個数のパーセント計算 114
ユニーク値の合計とリスト表示 115
除外演算子の使用 115
除外演算子の使用 116
複数表示コマンドでのフィールド値の計算 117
除外演算子の制限事項 117
先頭レコードと最終レコードの取得 118
先頭レコードの取得 119
セグメントタイプとレコードの取得 119
値の合計と個数の集計 121
CNT による値の集計 121
SUM 演算接頭語による値の集計 122
TOT による値の合計 122
RNK を使用したソートフィールド値の順位付け 123
RNK. を使用した順位付け演算接頭語 123
ソートグループ内の順位付け 124
WHERE TOTAL での RNK. の使用 125
COMPUTE での RNK. の使用 125
集計行での集約計算 126
集計値の集約 127
平均の集約 128
最上位ソート区切りへの集約の継承 130
ROLL 使用上の注意 132
母集団または標本の標準偏差の計算 132
母集団の標準偏差の計算 133
レポートレベルの演算接頭語の使用 133
レポートレベルの平均、最大値、最小値、件数の計算 134
見出しでの演算接頭語の使用 134
レポートレベル演算接頭語の使用上の注意 136
フィールドタイトルのポップアップ説明表示 137
POPUPDESC コマンドの使用 137
POPUPDESC コマンドの使用 138

ポップアップフィールド説明付きレポートの ReportCaster による配信 138

表形式レポートのソート 140
表形式レポートソートの概要 140
データのソートと表示 141
行のソート 141
行のソート 141
行をソートする際の注意 142
BY 句による行のソート 143
複数 BY フィールドの使用 143
複数 BY フィールドによるソート 144
ソート句により除外された行の表示 144
ソート句により除外された行の表示 144
PLUS OTHERS の使用上の注意 146
ソート句により除外されたソートフィールド値行の表示 146
ソートフィールドグループに含まれないデータ行の表示 147
フィールドのソート 148
フィールドのソート 148
フィールドをソートする際の注意 149
ACROSS 句によるフィールドのソート 150
単一フィールドの ACROSS タイトルの表示制御 150
単一フィールドの ACROSS タイトルの表示制御 151
ACRSVRBTITL による ACROSS タイトルの非表示 151
レポート出力の ACROSS タイトル配置 154
ACROSS フィールド名の配置制御 155
SET ACROSSTITLE 使用上の注意 155
ACROSS タイトルの ACROSS 値との同列配置 155
ACROSS タイトルの間隔 157
ACROSSTITLE=SIDE 使用時の ACROSS 値の背景色指定 159
複数 ACROSS フィールドの使用 160
複数 ACROSS ソート句によるソート 160
ACROSS 使用時の PRINT の圧縮 161
レポート行の圧縮 161

SET ACROSSPRT 使用上の注意 162
SET ACROSSPRT によるレポート出力の圧縮 162
ACROSS グループのブランク列の非表示 164
ACROSS ブランク列の非表示 164
ACROSS グループ内のブランク列を非表示にする際の注意 165
ACROSS ブランク列の非表示がサポートされない機能 165
BY 改ページ内の ACROSS グループおよび列の非表示 166
ACROSS のブランクグループの非表示 166
ACROSS グループ内の列の非表示 168
複数の ACROSS フィールドでのブランク列の非表示 171
集計行の生成および ACROSS ブランク列の非表示 174
列合計の生成および ACROSS ブランク列の非表示 174
列スタイルの使用および ACROSS ブランク列の非表示 177
列スタイルの使用および ACROSS ブランク列の非表示 177
FML リクエストでの ACROSS ブランク列の非表示 182
FML リクエストでの ACROSS ブランク列の非表示 182
ソートフィールド値の表示制御 186
ソートフィールド値の表示制御 187
レポート出力でのソートフィールド値の表示制御 188
ソートフィールドのフォーマット再設定 190
ソートフィールドのフォーマット再設定 190
ソートフィールドのフォーマット再設定時の注意 191
ソートフィールドのフォーマット再設定 191
ソートグループでの表示フィールド値の操作 193
WITHIN 句による表示フィールドの操作 193
ソートグループでの値の集計 194
マトリックスレポートの作成 195
単純なマトリックスの作成 195
複数のソートフィールドによるマトリックスの作成 195
照合順序の制御 196
バイナリまたは大文字と小文字を区別しない照合順序の設定 196
SET COLLATION 使用上の注意 197

ソート順の指定 198
ソート順の指定 198
昇順によるソート 198
降順によるソート 199
ソート順のユーザ定義 200
ソート順のユーザ定義 201
ソート順を定義する際の注意 202
行のソート順の定義 202
フィールドソート順の定義 203
ACROSS COLUMNS 句 203
フィールドソート順の定義 204
ACROSS 値の列タイトルの選択および適用 204
ACROSS 値への列タイトルの割り当て 205
ACROSS 値に列タイトルを割り当てる際の注意 205
ACROSS 値の列タイトルの選択および適用 206
ソートフィールド値の順位付け 206
ソートフィールド値の順位付け 207
ソートフィールド値の順位付け 207
ソートフィールド値の順位付けと表示データの限定 208
DENSE および SPARSE の順位付け 209
順位付けの最適化 209
順位付け方法の制御 209
SET RANK 使用時の注意 210
FOCUS データソースの値の順位付け 211
ソートフィールド値の個数の制限 212
範囲指定による数値データのグループ化 213
均等範囲によるグループの定義 214
均等範囲によるグループの定義 215
均等範囲の定義 215
均等範囲の定義 216
データ値のカスタムグループの定義 217
データ値のカスタムグループの定義 218

数値データのグループ配分表示 218
数値データのグループ配分表示 220
データの 5 グループ配分表示 221
先頭 3 グループの配分表示 221
上位 3 グループの配分表示 222
データをグループに配分して表示する際の注意 223
ソートフィールド値の順位による限定 224
ソートフィールド値の順位による限定 224
ソートフィールド値のグループの限定 225
レポートフィールドのソートおよび集計 225
レポートフィールドのソートおよび集計 226
レポートフィールドのソートおよび集計 226
レポートフィールドのソート、集計、順位付け 227
COMPUTE によるレポートフィールドのソートと集計 228
ACROSS 句を含むリクエストでの一時項目 (COMPUTE) に対する BY TOTAL の使用 228
BY TOTAL 使用上の注意 229
レポートに含めるフィールド数の制限 229
ソート値の非表示 230
ソート値の非表示 230
ソート値の非表示 230
ソート処理パフォーマンスの考慮事項 231
複数表示コマンドによるソート 233
複数の表示フィールドとソートフィールドの使用 233
複数表示コマンドを含むレポートのフォーマット制御 234
複数表示コマンドを含むレポートのフォーマット制御 235
複数表示コマンドを使用したレポートの表示 235
SET DUPLICATECOL=ON でのレポートスタイル設定 237
特定表示コマンドに関連する行のスタイル設定 237
複数動詞リクエストでの AS 名の照合に基づく重複フィールドの積み上げ表示 239
複数動詞リクエストでの AS 名の照合に基づく重複フィールドの積み上げ表示 239
外部ソートによる効率化 241

外部ソート使用上の注意 241
使用するソート方法を決定するには 242
外部ソートの制御 242
ソートタイプのクエリ 242
ソートに使用する入力レコードまたはレポートサイズの推定値の取得 243
ソートに使用する入力レコードまたはレポートサイズの推定値を取得するには 243
ソート作業ファイルとリターンコード 244
UNIX、Windows でのソート用作業ファイル 244
ibi™ WebFOCUS® 外部ソートのリターンコード 245
階層レポート - BY HIERARCHY 246
マスターファイルでの階層の指定 247
ディメンション階層が定義されたサンプルマスターファイル 250
階層レポートの作成 253
ディメンション階層レポートの作成 255
WHEN 句による階層メンバーの選択 258
選択された階層メンバーの WHERE 句による選別 259

レコードの選択 262
レコード選択の概要 262
フィルタ条件の選択 263
特定値によるレコード選択 263
WHERE 句によるレコード選択 263
WHERE 句の使用上の注意 264
単純な WHERE テストの使用 265
複数の WHERE 句の使用 265
分割 FOCUS データソースのレコードの選択 266
マルチパスデータソースのレコード選択制御 266
マルチパスデータソースでのレコード選択の制御 267
MULTIPATH = COMPOUND の要件と使用上の注意 268
複数パスからのデータ取得 268
MULTIPATH と SET ALL の組み合わせ 270
セグメントの必須とオプションを決定する際の規則 272

集計値による選択 ... 273
WHERE TOTAL によるレコードの選択 ... 273
WHERE TOTAL 使用上の注意 ... 274
WHERE TOTAL によるレコードの選択 ... 274
WHERE TOTAL と WHERE の組み合わせによるレコードの選択 ... 275
COMPUTE 処理前の内部マトリックスへの選択条件の適用 ... 276
WHERE_GROUPED 選択条件の適用 ... 276
WHERE_GROUPED テストの使用 ... 276
WHERE_GROUPED 使用上の注意 ... 280
複合式によるレコード選択 ... 280
AND と OR の組み合わせによるレコード選択 ... 281
演算子によるレコード選択 ... 281
WHERE と IF で使用可能な演算子 ... 282
演算子によるフィールドと値の比較 ... 285
WHERE 句での IF-THEN-ELSE ロジックの使用 ... 286
変数によるレコード選択 ... 288
レコード選択のタイプ ... 289
FROM ... TO による範囲 ... 289
範囲の指定 (FROM ... TO) ... 290
FROM ... TO による範囲 ... 290
NOT-FROM ... TO による範囲 ... 290
FROM ... TO によるソートフィールドの範囲 ... 291
GE ... LE または GT ... LT による範囲 ... 291
範囲の指定 (GE ... LE) ... 292
範囲内の値の選択 ... 292
範囲外の値の選択 ... 293
ミッシングデータのテスト ... 293
ミッシングデータのテスト ... 294
ミッシングでないデータのテスト ... 294
CONTAINS と OMITS による文字列の選別 ... 294
CONTAINS と OMITS によるレコードの選択 ... 295
マスクフィールドによる選別 ... 296

マスクによるフィールドの選別 (LIKE と NOT LIKE の使用) 296
IF 句の LIKE および UNLIKE による選別 297
マスクによるフィールドの選別 (IS と IS-NOT の使用) 298
マスクに使用する文字の制限 298
先頭文字による選別 299
フィールド任意の位置での文字選別 299
先頭文字と特定の文字数による選別 300
先頭文字と任意の文字数による選別 301
ワイルドカード文字の無効化 301
SET EQTEST によるレコード選択 301
LIKE でのエスケープ文字の使用 302
WHERE 句でのエスケープ文字の使用 303
IF 句マスクへのエスケープ文字の指定 303
エスケープ文字使用上の注意 304
WHERE 句内のエスケープ文字の使用 304
IF 句内のエスケープ文字の使用 305
INCLUDES と EXCLUDES による親セグメントの選択 306
INCLUDES および EXCLUDES の使用上の注意 307
INCLUDES および EXCLUDES によるレコードの選択 307
グループキー値による選択 308
グループキーによるレコードの選択 308
読み込みレコード数の制限 308
読み込みレコード数の制限 309
読み込みレコード数の制限 310
IF 句によるレコードの選択 310
IF 句によるレコードの選択 310
複数 IF 句の使用 311
ファイルからの選択値読み込み 312
ファイルからの選択値の読み込み - WHERE フィールド IN ファイル 312
ファイルからの選択値の読み込み - WHERE フィールド演算子 (ファイル) 313
ファイルからの選択値の読み込み - IF 313
ファイルから選択値を読み込む際の注意 314

ファイルからの選択値の読み込み (WHERE フィールド IN ファイル) 315
ファイルからの選択値の読み込み - WHERE フィールド演算子 (ファイル) 315
ファイルからの選択値の読み込み (IF) 316
ファイルへの選別条件の割り当て 317
フィルタの宣言 318
フィルタに一時項目 (DEFINE) を使用する際の注意 319
フィルタの宣言 319
フィルタの有効化と無効化 320
フィルタの有効化と無効化 321
フィルタステータスのクエリ 322
フィルタのクエリ 322
JOIN 間のフィルタ保持 324
KEEPFILTERS によるフィルタ定義の保持 325
KEEPFILTERS によるフィルタの保持 325

一時項目の作成 328
一時項目の概要 328
一時項目のタイプ 328
一時項目の評価 329
一時項目 (DEFINE) と一時項目 (COMPUTE) の区別 329
一時項目の選択 330
一時項目 (DEFINE) の定義 331
一時項目 (DEFINE) 作成時の注意 331
一時項目 (DEFINE) の作成 332
一時項目 (DEFINE) の定義 335
一時項目 (DEFINE) の複数定義 336
一時項目 (DEFINE) の追加 336
一時項目 (DEFINE) の追加 336
一時項目 (DEFINE) の表示 337
一時項目 (DEFINE) を表示するには 337
一時項目 (DEFINE) を表示するには 337
一時項目 (DEFINE) のクリア 337
一時項目 (DEFINE) のクリア 338

一時項目 (DEFINE) セグメント位置の設定 339
セグメント位置の設定 340
マルチパスデータソースでの一時項目 (DEFINE) の定義 340
一時項目 (DEFINE) 計算速度の向上 341
DEFINE FILE SAVE と RETURN による一時項目 (DEFINE) の保持 341
一時項目 (DEFINE) の上書き防止 342
レポートフィールドへの動的フォーマットの適用 342
フォーマットフィールドの定義および適用 343
フィールドを基準とするフォーマット再設定を行う際の注意 345
動的にフォーマットが設定されたフィールドの作成 346
SQL.Function 構文による関数呼び出しのリレーショナルエンジンへの直接転送 347
直接呼び出しの SQL 関数を使用する際の注意 347
リクエストでの SQL CONCAT 関数の呼び出し 348
一時項目 (COMPUTE) の作成 349
一時項目 (COMPUTE) 使用上の注意 349
一時項目 (COMPUTE) の作成 350
計算を伴わない一時項目 (COMPUTE) の作成 351
一時項目 (COMPUTE) の作成 352
一時項目 (COMPUTE) の位置参照 353
位置指定によるフィールド参照の使用 353
一時項目 (COMPUTE) での ACROSS 句の使用 354
表示コマンドの一部としての COMPUTE の使用 354
一時項目 (COMPUTE) での ACROSS 句の使用 354
一時項目のソート 355
一時項目 (COMPUTE) の選別 355
フィールド参照番号の割り当て 355
作成する列番号の制御 356
リクエストでの列表記の使用 356
FML 以外のリクエストで CNOTATION=ALL を使用した列表記 357
FML 以外のリクエストで CNOTATION=PRINTONLY を使用した列表記 358
FML リクエストで CNOTATION=PRINTONLY を使用した列番号 358

FML リクエストで CNOTATION=PRINTONLY を使用した連続列の RECAP 計算 360
FML リクエストで CNOTATION=PRINTONLY を使用した相対列アドレス指定 361
FML リクエストによる CNOTATION=PRINTONLY を使用した列表記 362
NOPRINT、フィールドのフォーマット再設定、COMPUTE を使用した列表記 363
FML リクエスト以外の NOPRINT への列表記の使用 364
FML リクエストでのセル番号の使用 366
列番号使用上の注意 368
COMPUTE コマンドでの FORECAST の使用 368
FORECAST による傾向値と予測値の計算 369
FORECAST の処理 369
FORECAST の使用上の注意 370
FORECAST の制限事項 371
データ行と予測行の区別 371
データ行と予測行の区別 371
FORECAST による傾向値と予測値の計算 372
FORECAST の処理 373
傾向値と予測値の計算 374
FORECAST の使用上の注意 376
FORECAST の制限事項 377
単純移動平均の使用 378
新しい単純移動平均フィールドの計算 378
単純移動平均フィールドとしての既存フィールドの使用 380
単純指数平滑法の使用 381
単純指数平滑フィールドの計算 382
二重指数平滑法の使用 384
二重指数平滑フィールドの計算 385
三重指数平滑法の使用 386
三重指数平滑フィールドの計算 387
線形回帰式の使用 388
新しい線形回帰フィールドの計算 389
FORECAST によるレポート作成 391
複数 FORECAST フィールドの生成 391

FORECAST フィールドの移動 392
データ行と予測行の区別 393
重回帰 REGRESS による傾向値と予測値の計算 394
線形重回帰フィールドの作成 395
REGRESS 使用上の注意 395
重回帰フィールドの作成 396
DEFINE および COMPUTE でのテキストフィールドの使用 398
文字式結果のテキストフィールドへの割り当て 399
マスターファイルから独立した一時項目の作成 399
関数の定義 400
DEFINE 関数の制限事項 401
関数の定義 402
DEFINE 関数を表示するには 404
定義済み関数の表示 404
DEFINE 関数のクリア 405

合計と中間合計の追加 406
行合計と列合計の計算 406
行合計と列合計の計算 407
行合計と列合計の計算 408
ON TABLE による列合計の指定 408
マトリックスレポートでの行合計と列合計の使用 409
行合計と列合計への一時項目 (COMPUTE) の追加 410
ROW-TOTAL の ACROSS および複数表示コマンドとの併用 410
ACROSS フィールド値に対する行合計の生成 413
ACROSS フィールド値に対する行合計の生成 413
ACROSS フィールド値に対する行合計の生成 414
ACROSS-TOTAL 使用上の注意 415
セクション合計と総合計の追加 415
セクション合計と総合計の使用 416
中間合計の追加 417
中間合計の生成 417
集計行の配置制御 419

データ上部への中間合計配置 419
中間合計使用上の注意 421
中間合計の生成 422
SUB-TOTAL と SUBTOTAL の比較 423
中間合計行値の再計算 424
一時項目 (COMPUTE) の中間合計 424
SUMMARIZE の使用 426
RECOMPUTE の使用 427
文字フィールドの集計 428
集計行でのすべてのフィールドの表示 429
集計行での文字フィールドの表示 429
文字フィールド集計の使用上の注意 430
演算接頭語による集計値の操作 430
集計値への演算接頭語の使用 432
集計演算接頭語使用上の注意 433
SUBTOTAL と演算接頭語の使用 434
ソート区切りレベルと総合計レベルでの SUBTOTAL の使用 435
複数の演算接頭語が指定されたフィールドの区別 437
集計行での文字フィールドの表示 438
集計行でのすべてのフィールドの表示 439
集計行処理の制御 440
集計行処理の制御 441
SET SUMMARYLINES 使用上の注意 442
SET SUMMARYLINES と SUBTOTAL の併用 443
COLUMN-TOTAL と SET SUMMARYLINES=EXPLICIT の併用 444
一時項目 (COMPUTE) での演算接頭語の使用 446
RECOMPUTE での演算接頭語の使用 446
ソート区切りレベルと総合計レベルでの RECOMPUTE の使用 448
複数 SUB-TOTAL または SUMMARIZE コマンドと演算接頭語の併用 449
複数 SUB-TOTAL コマンドと演算接頭語の併用 449
集計コマンドの組み合わせ 451
リクエストでの SUBTOTAL と RECOMPUTE の使用 452

SUB-TOTAL と複数の集計コマンドとの併用 453
複数の集計コマンドと演算接頭語の使用 454
フィールドリストを含む集計コマンドの継承 455
集計コマンドの組み合わせ時の注意 456
ACROSS フィールド集計列の生成 457
ACROSS フィールドの集計演算の実行 457
ACROSS フィールド集計の使用上の注意 458
ACROSS での集計コマンドの使用 460
ACROSS グループ内の単一フィールドの中間合計 461
ACROSS グループでの一時項目 (COMPUTE) 値の集計 462
集計コマンドの演算接頭語と ACROSS の使用 463
ACROSS 集計コマンドの組み合わせの使用 464
ソートフィールド区切りでの計算実行 466
計算での中間合計の使用 466
RECAP および COMPUTE の使用上の注意 467
RECAP の使用 467
複数 RECAP コマンドの使用 468
総合計の非表示 470
総合計の非表示 470
総合計の非表示 470
集計行とテキストの条件付き表示 471
集計行とテキストの条件付き表示 471

式の使用 473
コマンドおよび句での式の使用 473
式のタイプ 474
式とフィールドフォーマット 476
フィールドフォーマットの長さの確保 476
数値式の作成 476
指数表記での数値の表示 478
指数表記での数値の評価 479
算術演算子 479
評価の順序 480

評価順序の変更 480
演算式での IF-THEN-ELSE ロジックの使用 481
ネイティブモード計算による数値式評価 482
ネイティブモード計算による同一被演算子フォーマットの使用 483
同一の被演算子フォーマットの使用 (ネイティブモード計算) 483
ネイティブモード計算による異なる被演算子フォーマットの使用 484
日付式の作成 485
日付値のフォーマット 486
日付フォーマットの基準日 486
格納用と表示用の日付フォーマット 487
日付の計算 488
日付の計算 488
DEFINE および COMPUTE による 4 桁の西暦の日付 489
フィールドフォーマットの選択 489
返されるフィールドフォーマットの選択 489
日付定数の使用 490
日付定数によるフィールドの初期化 490
日付要素の抽出 490
日付から月数要素の抽出 490
フォーマットが異なるフィールドの組み合わせ 491
YYMD と MDY フィールドフォーマットの組み合わせ 491
並び順が異なる要素の結果フィールドへの割り当て 492
日付時間式の作成 492
日付および日付時間のフォーマット自動変換 492
日付値および日付時間値の割り当て 493
日付値および日付時間値の比較 495
日付時間フィールドの日付構成要素の順序を指定 496
日付時間値の指定 496
日付時間リテラルの割り当て 499
日付時間フィールドの日付コンポーネントの順序指定 499
トランザクションファイルからの日付時間値の読み取り 500
COMPUTE コマンドでの日付時間値の使用 501

WHERE 条件での日付時間値の使用 501
IF 条件での日付時間値の使用 502
世界標準時による日付入力値の指定 502
日付時間値の操作 503
日付時間ミッシング値のテスト 503
日付時間フィールドへの異なるフォーマットの割り当て 504
文字式の作成 505
引用符区切りリテラル列での引用符の埋め込み 506
引用符区切りリテラル列での O'BRIEN データ値の指定 506
文字列の連結 507
文字列の連結 507
文字式での IF-THEN-ELSE ロジックの使用 508
可変長文字式の作成 509
AnV フィールドでの連結の使用 509
AnV フィールドでの EDIT 関数の使用 510
nV フィールドでの CONTAINS および OMITS の使用 510
AnV フィールドでの LIKE の使用 511
AnV フィールドでの EQ、NE、LT、GT、LE、GE 演算子の使用 512
AnV フィールドでの DECODE 関数の使用 513
AnV フィールドでの代入演算子の使用 514
論理式の作成 515
論理演算子 515
関係式の記述 516
ブール式の記述 517
条件式の作成 517
条件式の記述 518
条件式による値の指定 518
TRUE または FALSE 条件の定義 519

レポート出力の保存と再利用 521
レポート出力の保存 521
レポート出力ファイル名の指定および保存 522
HOLD ファイルの作成 523

HOLD ファイルを作成するには 523
ホットスクリーンからの HOLD ファイルの作成 526
HOLD での DATASET の使用 527
デフォルト HOLD フォーマットの設定 528
HOLD ファイルへのデータの抽出 528
HOLD マスターファイルのクエリ 529
HOLD マスターファイルに対するレポートの実行 529
ibi FOCUS フォーマットによるレポート出力の保存 532
ibi FOCUS フォーマットによる HOLD ファイルの作成 532
ibi FOCUS ファイル構造の制御 533
ibi FOCUS フォーマット HOLD ファイルの作成 534
PRINT を使用した FOCLIST フィールドによる FOCUS データソースの作成 535
2 セグメント FOCUS データソースの作成 535
3 セグメント FOCUS データソースの作成 536
HOLD マスターファイルの属性制御 537
HOLD マスターファイルでのフィールド名の制御 538
HOLD マスターファイルでのフィールド名の制御 538
HOLD ファイルのフィールド名制御使用上の注意 539
HOLD マスターファイルでのフィールド名の制御 540
SET ASNAMES による一意のフィールド名の作成 541
SET ASNAMES の ACROSS 句への使用 541
SET ASNAMES=FLIP による HOLD ファイルの生成 543
HOLD マスターファイルのフィールド制御 545
HOLD ファイルのフィールド制御 545
HOLDLIST=ALL の使用 546
HOLDLIST= PRINTONLY の使用 547
異なる HOLDLIST 設定で作成したマスターファイルの比較 549
HOLD マスターファイルでの属性の制御 551
TITLE および ACCEPT 属性の制御 551
HOLD マスターファイルでの TITLE および ACCEPT 属性の制御 552
HOLD ファイルからのキー検索 553
HOLD ファイルのキー検索制御 553

HOLD ファイルからのキー検索のマスターファイル 554
抽出ファイルからのキー検索の選択条件 554
HOLD ファイルの保存および検索 555
HOLD マスターファイルの格納場所の指定 (Windows および UNIX) 555
HOLD データファイルおよびマスターファイルの格納場所の指定 556
データベース一時テーブルの HOLD ファイルとしての使用 557
コマンドによるネイティブ一時テーブルとしてのレポート出力の保存 558
一時テーブル の SAME_DB PERSISTENCE プロパティ 560
HOLD ファイル内のフィールド名 562
HOLD ファイル内の主キーとインデックス 562
SAVE および SAVB ファイルの作成 563
SAVE ファイルの作成 563
SAVE ファイルの作成 564
SAVB ファイルの作成 565
SAVB ファイルの作成 566
PCHOLD ファイルの作成 567
PCHOLD ファイルの作成 567
出力ファイルフォーマットの選択 569
FORMAT AHTML 570
FORMAT AHTMLTAB 570
FORMAT ALPHA 571
FORMAT APDF 571
FORMAT BINARY 572
FORMAT COMMA 572
FORMAT COM 573
FORMAT COMT 574
FORMAT DATREC 575
FORMAT DB2 575
FORMAT DBASE 575
FORMAT DFIX 576
FORMAT DHTML 576
FORMAT DIF 577

FORMAT DOC 577
FORMAT XLSX 578
FORMAT EXL2K 578
FORMAT EXL2K FORMULA 579
FORMAT EXL2K PIVOT 579
FORMAT EXL97 579
FORMAT FLEX 580
FORMAT FOCUS 580
FORMAT GIF 580
FORMAT HTML 581
FORMAT HTMTABLE 581
FORMAT INGRES 582
FORMAT INTERNAL 582
FORMAT JPEG 583
FORMAT JSCHART 583
FORMAT JSON 583
FORMAT LOTUS 584
FORMAT PDF 584
FORMAT PDF OPEN/CLOSE 586
FORMAT PPT 586
FORMAT PPTX 586
FORMAT REDBRICK 586
FORMAT SQL_SCRIPT 587
FORMAT SQLDBC 587
FORMAT SQLINF 588
FORMAT SQLMAC 588
FORMAT SQLMSS 588
FORMAT SQLODBC 589
FORMAT SQLORA 589
FORMAT SQLPSTGR 589
FORMAT SQLSYB 590
FORMAT SYLK 590

FORMAT TAB 590
FORMAT TABT 591
FORMAT WK1 592
FORMAT WP 592
FORMAT XFOCUS 593
FORMAT XML 594
ON TABLE MERGE による既存データソースへのデータの結合 594
ON TABLE MERGE 処理 595
ON TABLE MERGE による既存データソースへのデータの結合 597
ON TABLE MERGE によるレコードの更新および挿入 599
ON TABLE MERGE によるレコードの追加 601
ON TABLE MERGE によるレコードの削除 602
ON TABLE MERGE のベストプラクティス 603
ON TABLE MERGE でのエイリアス名の使用 607
出力ファイルでのテキストフィールドの使用 608
出力ファイルでのテキストフィールドの規則 608
HOLD ファイルでのテキストフィールド規則の適用 609
区切り付きシーケンシャルファイルの作成 611
区切り付きシーケンシャルファイルの作成 612
HOLD FORMAT DFIX 使用上の注意 614
パイプ区切りファイルの作成 614
タブ区切りファイルの作成 616
区切り付きファイルの作成とブランクの保持 617
レコード区切り文字の指定 619
HOLD ファイルのミッシングデータ 621
INTERNAL フォーマットでのレポート出力の保存 621
HOLD ファイルでのフィールドパディングの抑制 622
HOLD ファイルでフィールドパディングを抑制する際の注意 623
HOLD FORMAT INTERNAL を使用しない HOLD ファイルの作成 624
HOLD FORMAT INTERNAL を使用した HOLD ファイルの作成 624
HOLD FORMAT SQL_SCRIPT によるサブクエリまたはシーケンシャルファイルの作成 625

HOLD FORMAT SQL_SCRIPT による SQL スクリプトファイルまたはシーケンシャルファイルの作成 626
HOLD FORMAT SQL_SCRIPT による SQL スクリプトファイルの作成 626
HOLD FORMAT SQL_SCRIPT によるシーケンシャルファイルの作成 627
構造化 HOLD ファイルの作成 628
リクエストの構造化 HOLD ファイルの有効化 629
構造化 HOLD ファイルの作成 630
構造化 HOLD ファイル生成オプションの設定 630
ALPHA フォーマット構造化 HOLD ファイルの作成 631
ibi FOCUS フォーマット構造化 HOLD ファイルの作成 632
構造化 HOLD ファイルの再構築 633
構造化 HOLD ファイルのフィールドの除外 634
構造化 HOLD ファイルに格納される要素 635
構造化 HOLD ファイルに格納されない要素 636
構造上および動作上の注意 636

表示フォーマットの選択 639
レポート表示フォーマット 639
PCHOLD による表示フォーマットの選択 640
SET ONLINE-FMT による表示フォーマットの選択 641
WebFOCUS レポートの MIME タイプの指定 642
レポート出力での先頭および内部ブランクの保持 643
HTML および EXL2K レポートの先頭および内部のブランクの保持 643
HTML および EXL2K レポート出力の先頭および内部ブランクの保持 644
Web 表示フォーマットの使用 - HTML 645
HTML レポート表示のカスタマイズ 645
WebFOCUS デフォルト JavaScript 関数の無効化 647
HTML レポート出力の使用上の注意 648
絶対配置による HTML レポートの作成 648
Web アーカイブレポート出力の作成 649
DHTML レポートの作成 649
DHTML フォーマットの使用上の注意 650
DHTML レポートの作成 650

印刷表示フォーマットの使用 - PDF 651
PDF 表示フォーマットの使用 651
PDF 出力ファイルの圧縮 652
PDF 出力での透かしの表示 652
背景色を使用した PDF レポートでのイメージ挿入 653
PDF レポートへの透明イメージの挿入 654
サポートされる機能 656
制限事項 657
使用上の注意 657
ページ幅に合わせた PDF レポート出力の拡大縮小 657
ページ幅に合わせた PDF レポート出力の拡大縮小 658
PAGE-SCALE 使用上の注意 658
ページ幅に合わせた PDF レポート出力の拡大縮小 659
PDF レポートのページ位置揃え 663
PDF レポートのページ左揃え 664
PDF レポートのページ中央揃え 665
PDF レポートのページ右揃え 666
WebFOCUS PDF レポートのアクセシビリティサポート 667
アクセシビリティ対応 PDF コードの制御 668
PDF コードアクセシビリティの制御 668
アクセシビリティ対応 PDF コードの制御 668
ページ見出しの各要素と列データとの位置揃え 672
ページ見出しの各要素と列データとの位置揃え 673
ページ見出しの各要素と列データとの位置揃え 673
しおりの追加 676
複合レイアウトへのしおりの追加 677
イメージへの説明テキストの追加 681
イメージに説明テキストを追加するには 682
イメージへの説明テキストの追加 682
ドリルダウン情報の説明 683
ドリルダウン情報の説明の追加 684
アクセシビリティの制限事項 686

ワードプロセッサ表示フォーマットの使用 - DOC、WP 686
Excel XLSX フォーマットでのレポート出力の保存 687
EXL07/XLSX フォーマットの概要 688
.xlsx ブックファイルの作成 689
EXCELSERVERURL による .xlsx ファイルの圧縮方法の選択 690
.xlsx ファイル圧縮での C ベース ZIP ローカルモードの指定 691
Excel XLSX ブックの生成 692
XLSX レポート出力を開く 692
Excel 2000/2003 で XLSX レポート出力を開く 692
Excel ブックのブラウザ表示と Excel アプリケーション表示 694
XLSX レポート出力のセル値のフォーマット設定 694
XLSX レポート出力でのフォーマット設定済み数値の表示 695
XLSX レポート出力への数値フォーマットの転送 696
レポート見出しおよび脚注での数値フォーマットの使用 698
見出しおよび脚注での数値フォーマット区切り文字の使用 698
脚注を含む XLSX レポートのソートに関する使用上の注意 699
XLSX レポート出力への日付の転送 701
WebFOCUS 日付の Excel XLSX 日付への変換 702
日構成要素を含まない日付の転送 703
日構成要素を含まない WebFOCUS 日付の XLSX レポート出力への転送 703
Excel 数式で使用する日構成要素の転送 704
XLSX レポート出力への数値日付構成要素の転送 704
四半期フォーマットの転送 705
四半期構成要素を含む日付の XLSX レポート出力への転送 705
変換済みテキストとして定義された日付構成要素の転送 706
変換済みテキストとして定義された日付構成要素の XLSX レポート出力への転送 706
XLSX レポート出力での日付値の使用上の注意 707
XLSX への日付時間値の転送 707
XLSX への日付時間値の転送 708
XLSX レポート出力での Excel 数式の生成 709
数式および値の理解 709

FORMAT XLSX FORMULA の変換サポート 710
FORMAT XLSX FORMULA としてのレポート保存 711
列合計の Excel 数式の生成 712
行合計の Excel 数式の生成 713
一時項目 (COMPUTE) の Excel 数式の生成 714
WebFOCUS 関数の Excel 数式の生成 715
再計算値を含む数式の生成 716
再計算値を含む数式の生成 716
XLSX FORMULA と演算接頭語の使用 718
FORMAT XLSX FORMULA での集計演算接頭語の使用 719
FORMAT XLSX FORMULA での表示コマンドと演算接頭語の使用 720
数式での NODATA の使用 721
XLSX 数式の使用上の注意 722
XLSX レポート出力での列幅および折り返しの制御 722
XLSX レポート出力での列幅の設定 723
XLSX レポート出力でのデータの折り返し 724
XLSX レポート出力での列幅および折り返しの制御 725
XLSX レポートの列の固定 726
XLSX レポートでの列の固定 727
列の固定使用上の注意 727
XLSX レポートの列の固定 727
WebFOCUS の改ページと Excel の改ページとの同期 729
XLSX レポート出力フォーマットでの WebFOCUS 改ページと Excel 改ページの同期 729
レポート出力での先頭および内部ブランクの保持 735
XLSX レポート出力の先頭および内部ブランクの保持 736
XLSX レポート出力の先頭および内部ブランクの保持 737
XLSX レポート出力でのドリルダウンのサポート 739
リダイレクトと Excel ドリルダウンレポート 740
Excel ページ設定 741
Excel ページ設定の定義 742
レポートへのイメージの追加 743

Excel XLSX レポートへのイメージの挿入 743
XLSX レポートの WebFOCUS レポート要素へのイメージの追加 743
XLSX レポートの見出しまたは脚注へのイメージの配置 745
単一テーブルリクエストへの GIF イメージの追加 748
複合リクエストへの GIF イメージの追加 750
BYTOC 複合リクエストへの GIF イメージの追加 754
XLSX ブックの見出しおよび脚注へのテキストおよびイメージの挿入 755
XLSX ブックの見出しおよび脚注へのテキストおよびイメージの挿入 756
Excel の見出しおよび脚注へのイメージ挿入とページ設定の定義 757
XLSX ワークシートの見出しおよび脚注にテキストおよびイメージを挿入する際の注意 759
XLSX レポート出力での透かしの表示 759
XLSX 見出しへの透かしの配置 760
テーブル範囲構文を使用した Excel テーブルの作成 761
テーブル範囲構文を使用た Excel テーブルの作成 762
テーブル範囲構文を使用した Excel テーブルの作成 762
テーブルレベル構文およびテーブル範囲構文を使用した Excel テーブルの作成 764
IN-XSLXTABLE 属性使用上の注意 766
Excel の名前付け規則に関する注意事項 767
テンプレートによる Excel XLSX ワークシートの作成 768
サポート対象のテンプレートファイルタイプを使用した XLSX レポートの作成 769
XLSX TEMPLATE 使用上の注意 770
Excel 目次レポートの作成 770
Excel 目次機能の使用 771
単純な BYTOC レポートの作成 771
ワークシートの名前指定 772
大文字と小文字が区別されたデータ使用時の XLSX ワークシート名 772
オーバーフローワークシート機能による Excel 2007/2010 での行数制限への対処 773
オーバーフローワークシート機能の有効化 774
XLSX オーバーフローワークシート使用上の注意 775
オーバーフローワークシートの作成 776

WebFOCUS 改ページを含むオーバーフローワークシートの作成 778
XLSX による Excel 複合レポート 780
XLSX による Excel 複合レポートの使用上の注意 781
コンポーネントを含む複合 Excel レポート 781
目次を含む Excel 複合レポート (BYTOC) 786
Excel 複合ドキュメントへのコンポーネントの配置 791
レガシー OPEN、CLOSE、NOBREAK キーワードおよび SET COMPOUND の使用に関するガイドライン 792
XLSX による Excel 複合レポートの作成に関するガイドライン 793
XLSX による単純な複合レポートの作成 794
NOBREAK キーワードによる複合レポートの作成 795
複合レポートでの XLSX FORMULA の使用 797
XLSX での WebFOCUS ピボットのサポート 799
WebFOCUS レポートからピボットテーブルおよびピボットグラフへのデータフィード 799
FORMAT XLSX の制限事項 803
PowerPoint PPT 表示フォーマットの使用 803
PowerPoint PPT テンプレートの使用 804
PowerPoint PPT レポート出力の作成 804
PowerPoint PPT テンプレートの使用 805
PPTX フォーマットでのレポート出力の保存 806
プレゼンテーションファイル (.pptx) の作成 807
.pptx ファイルの圧縮方法の選択 807
PPTX プレゼンテーションの生成 809
PPTX レポート出力を開く 809
Microsoft PowerPoint 2000/2003 で PPTX レポート出力を開く 810
ブラウザおよび PowerPoint アプリケーションでの PowerPoint プレゼンテーションの表示 811
PowerPoint でのテーブルとコンポーネントのグループ化 812
PowerPoint でのテーブルとコンポーネントのグループ化 812
標準レポートでのテーブルとコンポーネントのグループ表示 813
複合レポートでのテーブルとコンポーネントのグループ表示 814
日付およびページ/スライド番号 817

テキストオブジェクトのテキストフォーマットマークアップタグ 817
マークアップタグによる複合レイアウトテキストオブジェクトのフォーマット設定 822
ページマスターへのテキストおよび線オブジェクトの配置 824
PPTX レポート出力でのテキストの縦方向配置マークアップ 826
黒丸、白丸、黒い四角を使用した箇条書きリスト (番号なし) の表示 827
黒丸、白丸、黒い四角を使用した箇条書きリスト (番号なし) の表示 827
PowerPoint PPTX レポート要素へのイメージの挿入 830
WebFOCUS PPTX レポートへのイメージの挿入 831
レポート見出しおよび脚注へのイメージの挿入 831
レポートのデータセルへのイメージの挿入 835
PNG イメージフォーマットでの PPTX グラフの表示 837
PNG イメージフォーマットでの PPTX グラフの表示 838
透明度を使用した PNG グラフの表示 838
透明度を使用した PNG イメージの指定テンプレートでの表示 841
Microsoft PowerPoint からのドリルダウン 843
外部 URL へのドリルダウン 844
テンプレートを使用した PowerPoint PPTX プレゼンテーション 846
標準 PowerPoint テンプレート (POTX) の使用 847
複数リクエストによる指定スライドのテンプレートへの挿入 849
PowerPoint PPTX 複合ドキュメント構文 854
複合ドキュメントの作成 854
統合された複合レイアウトレポート 858
統合された複合レイアウトレポートの生成 859
複合レポートのテンプレート 863
複合レポートリクエストへのイメージの追加 864
複合レポートリクエストへのイメージの追加 864
テンプレートマスターおよびスライドレイアウト 868
PowerPoint でのスライドマスター属性の指定 868
PowerPoint でのスライドマスター属性の指定 871
複合レポートによる複数マスターへのアクセス 872
WebFOCUS コンテンツと PowerPoint テンプレートコンテンツとの結合 879

WebFOCUS コンテンツと PowerPoint テンプレートコンテンツとの結合 ... 879
WebFOCUS コンテンツと PowerPoint テンプレートコンテンツとの結合 ... 879
ターゲットスライドコンテンツの上書き ... 881
ReportCaster による配信およびバースト ... 882
PPTX の制限事項 ... 882
関連情報 ... 883

レポートの別リソースへのリンク ... 884
スタイルシートを使用したリンク ... 884
スタイルシートを使用したリンクを作成するには ... 885
別レポートへのリンク ... 885
レポートおよびプロシジャへのリンク ... 885
PDF フォーマットでのドリルダウンレポート使用上の注意 ... 887
WebFOCUS ファイル名を決定するには ... 887
脚注からレポートへのリンク ... 888
URL へのリンク ... 891
URL へのリンク ... 891
URL へのリンク ... 893
WebFOCUS Server ドリルダウンプロシジャを実行する URL へのリンク ... 894
URL へのリンク使用上の注意 ... 897
ハイパーリンクの色の定義 ... 897
ハイパーリンクの色の定義 ... 897
ハイパーリンクの色の定義 ... 898
HYPERLINK-COLOR 使用上の注意 ... 899
JavaScript 関数へのリンク ... 899
JavaScript 関数へのリンク ... 900
JavaScript 関数へのリンク ... 901
カスケードメニューおよびユーザ定義のスタイルを使用したマルチドリルダウン機能 ... 903
アクセシビリティサポート ... 903
複数ドリルダウンリンクの作成 ... 903
グローバルメニューのスタイル設定 ... 904
メニュー項目のスタイル設定 ... 906

ドリルダウンアクションのオプション 908
ドリルダウンリンクの概要 909
ドリルダウンメニューのスタイルシート構文の例 909
マルチドリルダウンメニュー使用上の注意 913
条件付きスタイル設定の適用 913
　条件付きスタイル設定の複数ドリルダウンレポートへの適用 914
パラメータの作成 915
パラメータの作成 916
　定数値指定によるパラメータの作成 917
　フィールドの指定によるパラメータの作成 920
　変数の指定によるパラメータの作成 922
　ドリルダウンリクエストでの DRILLMETHOD の使用 924
　複数パラメータの使用 926
条件付きリンク 928
条件付きリンク 928
　条件付きリンク 930
グラフィックイメージからのリンク 931
グラフィックイメージからのリンクの指定 932
　イメージからのリンクの指定 934
ベース URL の指定 937
ベース URL の指定 937
　ベース URL の指定 938
ターゲットフレームの指定 938
ターゲットフレームの指定 939
デフォルトターゲットフレームの指定 940
　ターゲットフレームの指定 941
複合レポートの作成 941
ドキュメント構文による複合レイアウトレポートの作成 942
SECTION 宣言および構文 944
PAGELAYOUT 宣言 945
ページマスター 946
COMPONENT 宣言 947

ドキュメント構文による複合レイアウトレポートの作成 950
ドキュメント構文による統合されたグラフの作成 952
ドキュメント構文による複数ページレイアウトの作成 955
ドキュメント構文によるページオーバーフローの作成 957
ドキュメント構文によるオブジェクトの描画 961
ドキュメント構文によるオブジェクトの描画 966
ドキュメント構文によるグリッドの表示 967
テキストオブジェクトのテキストフォーマットマークアップタグ 967
マークアップタグによる複合レイアウトテキストオブジェクトのフォーマット設定 972
ページマスターへのテキストおよび線オブジェクトの配置 974
PDF レポート出力でのテキストの縦方向配置マークアップ 977
複合レイアウトレポートとミッシングデータ 978
設定 - ミッシングデータを含む複合レイアウトレポートの作成 979
手順 1 - データファイルの作成 979
手順 2 - コンポーネントレポートの作成 982
手順 3 - 複合レイアウトレポートの作成 985
PPTX、PDF 複合レイアウトレポートの BY フィールドによる目次作成 989
目次の機能 990
PPTX 複合レポートの目次生成 991
PDF 複合レポートの目次生成 993
目次使用上の注意 995
しおりページおよび目次ページの作成 996
複合 PDF レポートの作成 998
複合レポートの表示 999
複合 PDF レポートの作成 1000
複合レポートへのグラフの埋め込み 1002
複合レポート内のレポートフォーマットおよびグラフの組み合わせ 1003
EXL2K による Excel 複合レポートの作成 1006
OPEN、CLOSE、NOBREAK キーワードおよび SET COMPOUND の使用に関するガイドライン 1007
EXL2K による Excel 複合レポートの作成に関するガイドライン 1008
EXL2K による単純な複合レポートの作成 1009

PivotTable および数式による複合レポートの作成 1012
NOBREAK キーワードによる複合レポートの作成 1016
ドリルスルーリンクを使用した PDF 複合レポートの作成 1019
ドリルスルーとドリルダウンの比較 1019
他の機能との併用 1020
PDF 複合レポートでドリルスルーを作成するには 1021
PDF 複合レポートでドリルスルーを作成するには 1021
ドリルスルーハイパーリンクの指定 1021
ドリルスルーパイパーリンクの指定 1023
複合レイアウトレポートへのハイパーリンク関係設定 1023
DRILLMAP 属性を使用したサンプルコンポーネント宣言 1024
ドリルスルー使用上の注意 1024
サンプルのドリルスルー PDF 複合レポート 1025
概要レポートの作成 (手順 1) 1025
詳細レポートの作成 (手順 2) 1026
ハイパーリンクによるレポートの関連付け (手順 3) 1027
複合レイアウトレポートの作成 (手順 4) 1029
概要レポートと詳細レポートを結合した PDF 複合レポートの作成 (手順 4) 1031
ドリルスルーレポートの実行 (手順 5) 1032
FIRST を使用したリンクに関するガイドライン 1033
ドリルスルーハイパーリンクの規則 1034

HTML レポートのナビゲーション 1036
目次からソートグループへのナビゲーション 1036
HTML 目次で HTMLARCHIVE を使用する際の注意 1038
レポートへの HTML 目次ツリーコントロールの追加 1038
SET コマンドによるレポートへの TOC ツリーコントロールの追加 1038
PCHOLD コマンドによるレポートへの TOC ツリーコントロールの追加 1039
スタイルシート宣言によるレポートへの TOC ツリーコントロールの追加 1040
レポートのオブジェクトとして HTML TOC を追加 (レポートオプション) 1041
表示用ソートフィールドのグループ化 1044
改ページによるレポートセクションのカスタマイズ 1044
複数階層 TOC でのナビゲーション動作 1045

複数階層 TOC からのソートデータのナビゲーション 1046
見出しへの TOC ドロップダウンリストコントロールの追加 1049
ページ見出しへの HTML TOC ドロップダウンリストの追加 1051
ページ見出しでの複数階層 HTML TOC のナビゲーション 1053
レポート間のナビゲーション 1056
HTML 目次の制限事項 1056
アコーディオンレポート機能によるソートデータの表示制御 1057
アコーディオンレポートの要件 1058
アコーディオンレポートでの HTMLARCHIVE 使用上の注意 1058
ReportCaster によるアコーディオンレポートの配信 1059
同一列内アコーディオンレポートの作成 1060
同一列内で展開するアコーディオンレポートの作成 1061
SUM を使用した同一列内アコーディオンレポートの作成 1062
PRINT を使用した同一列内アコーディオンレポートの作成 1065
強化されたインターフェースによるアコーディオンレポートの作成 1068
強化されたインターフェースによるアコーディオンレポートの作成 1068
EXPANDBYROW および EXPANDBYROWTREE 使用上の注意 1073
同一列内アコーディオンレポートのツールヒント 1074
同一列内アコーディオンレポートの作成 - ポップアップフィールド説明をオフ 1075
同一列内アコーディオンレポートの作成 - ポップアップフィールド説明をオン 1077
同一列内アコーディオンレポートでの NOPRINT の使用 1078
同一列内アコーディオンレポートの作成 - 明示的な NOPRINT の使用 1079
BY フィールドのフォーマット再設定と再定義の相違点 1081
同一列内アコーディオンレポートの作成 - BY フィールドの動的なフォーマット再設定 1081
同一列内アコーディオンレポートの作成 - BY フィールドの再定義 1083
SET EXPANDBYROW および EXPANDBYROWTREE 使用上の注意 1084
列単位のアコーディオンレポートの作成 1085
列単位のアコーディオンレポートのサポート 1085
列単位のアコーディオンレポートの作成 1085
列単位のアコーディオンレポートの作成 1086

Web ビューアによる複数ページレポートのナビゲーション 1087
Web ビューアでの HTMLARCHIVE 使用上の注意 1088
Web ビューアでナビゲートするには 1089
Web ビューア検索オプションの使用 1090
レポートページの接続 1090
レポートページのリンク 1090
ページ番号および見出し要素によるページの接続 1091

複数 HTML ファイルへのレポート分割 1094
バーストレポートの概要 1094
複数 HTML ファイルへのレポートのバースト 1094
インデックスページおよびバーストレポートでの見出しおよび脚注の規則 1095
レポートのバースト 1096

ミッシングフィールド値を含むレコード処理 1100
関連値を持たないレポートデータ 1100
関連値を持たないレポートデータ 1100
ミッシングフィールド値 1101
ミッシング値の個数計算 1102
ミッシング値を使用した平均計算 1102
マスターファイルの MISSING 属性 1103
MISSING 属性によるミッシング値の処理 1103
DEFINE または COMPUTE コマンドの MISSING 属性 1104
DEFINE または COMPUTE コマンドによるミッシング値の指定 1105
MISSING OFF による一時項目 (DEFINE) のミッシング値の処理 1106
SOME および ALL による一時項目 (DEFINE) のミッシング値の処理 1107
DEFINE および COMPUTE の MISSING ON 動作の設定 1108
一時項目 (DEFINE) および一時項目 (COMPUTE) での MISSING ON のデフォルト値の設定 1109
IF-THEN-ELSE 式でのミッシング値のテスト 1110
IF-THEN-ELSE 式での SET MISSINGTEST の使用 1112
IF-THEN-ELSE 式での SET MISSINGTEST の使用 1113
ミッシングフィールド値を含むセグメントのテスト 1114

ミッシングフィールド値を含むセグメントのテスト 1115
ミッシングフィールド値のテスト 1115
既存のフィールド値のテスト 1116
ブランクまたは 0 (ゼロ) のテスト 1116
ミッシング値のテストからの除外 1117
ミッシングデータ値の出力ファイルへの保存 1117
抽出ファイルでのミッシング値の区別 1118
MISSING 値の抽出ファイルへの統合 1118
ミッシング値の HOLD ファイルでの保存 1119
HOLD ファイルでのミッシング値使用上の注意 1119
HOLDMISS によるミッシング値の保持 1120
ミッシング値をフォーマット再設定フィールドへ継承するリクエスト 1120
フォーマット再設定フィールドのミッシング値制御 1121
フォーマット再設定フィールドのミッシング値制御 1121
SET COMPMISS の使用上の注意 1123
ミッシングセグメントインスタンスの処理 1123
子インスタンスを持たないセグメントに対するレポート 1124
子インスタンスを持つセグメントに対するレポート 1125
ALL. 接頭語によるミッシングインスタンスのレポートへの追加 1127
ALL. 接頭語によるミッシングセグメントインスタンスの追加 1127
SET ALL パラメータによるミッシングインスタンスのレポートへの追加 1128
子が欠落している親インスタンスの追加 1129
SET ALL によるミッシングセグメントインスタンスの追加 1129
LEFT OUTER JOIN でのショートパス処理の制御 1130
SET SHORTPATH = SQL 使用上の注意 1131
OUTER JOIN 処理の制御 1131
一致クロスリファレンスレコードが存在しないホストレコードの検索 1134
FOCUS データソースでのミッシングインスタンスのテスト 1134
FOCUS データソースのミッシングインスタンスのテスト 1135
NODATA 文字列の設定 1136
NODATA 列の設定 1136
文字を非表示にするための NODATA の設定 1137

NODATA 文字列の設定 1137

データソースの結合 (JOIN) 1138

JOIN のタイプ 1138

JOIN データ構造 1139

結合可能および不可能なデータソース 1140

結合データ構造での DBA セキュリティの注意 1140

ユニークおよび非ユニーク JOIN 構造 1141

ユニーク EQUIJOIN 構造 1141

非ユニーク EQUIJOIN 構造 1142

ユニーク JOIN でのラギング値を修正するには 1143

プロシジャによるユニークセグメントと複数の子セグメントのラギング値の修正 1144

再帰的 JOIN 構造 1145

再帰的 JOIN 構造の理解 1145

再帰的 JOIN 構造使用上の注意 1147

再帰的 JOIN 構造の使用 1147

JOIN コマンドの動作原理 1150

JOIN データソースの検索の高速化 1151

EQUIJOIN の作成 1151

実フィールドの結合 1151

単純ユニーク JOIN 構造の作成 1154

INNER JOIN の作成 1154

LEFT OUTER JOIN の作成 1157

RIGHT OUTER JOIN の作成 1158

マルチパス構造の 2 つの INNER JOIN の作成 1160

EQUIJOIN のクロスリファレンスフィールドの要件 1162

グループフィールドの制限事項 1163

INNER JOIN および OUTER JOIN コマンド構文使用上の注意 1164

EQUIJOIN による一時項目 (DEFINE) から実フィールドへの結合 1164

一時項目 (DEFINE) から実フィールドへの結合 1165

JOIN データソースによる一時項目 (DEFINE) 使用上の注意 1167

JOIN 構造の仮想ホストフィールドの作成 1168

EQUIJOIN の JOIN モード 1169
レコードベースデータソースの JOIN モードの制御 1170
共有フィールドのデータフォーマット 1170
異なる数値データタイプを含むフィールドの結合 1171
データタイプ変換付き JOIN の有効化 1172
  データタイプ変換を含む JOIN の発行 1172
条件付き JOIN の使用 1173
条件付き JOIN の作成 1173
  条件付き JOIN の使用 1176
FULL OUTER JOIN 1177
FULL OUTER JOIN の指定 1178
  Microsoft SQL Server テーブルの FULL OUTER JOIN の最適化 1183
マルチファクトクラスタシノニムに対するレポートの作成 1185
  マルチファクトクラスタシノニムに対するレポートの作成 1186
  マルチファクトシノニムへの新しいファクトの追加 - JOIN AS_ROOT 1187
  新しい親セグメントの追加 1188
  クラスタシノニムコンテキストでの OUTER JOIN の作成 1191
  クラスタシノニムコンテキストでの JOIN 処理の制御 1192
  マルチファクトシノニムからの JOIN の作成 1195
クラスタシノニム間の JOIN のナビゲーション 1198
クラスタシノニム JOIN 使用上の注意 1199
  クラスタシノニム間の JOIN のナビゲーション 1200
データベース間の JOIN の最適化 1203
ファントラップ時のスタースキーマコンテキスト分析の呼び出し 1209
JOIN 条件への DBA 制限の追加 - SET DBAJOIN 1210
JOIN 条件への DBA 制限の追加 1211
  リレーショナルテーブルでの DBAJOIN 設定の使用 1211
JOIN 解析中の一時項目 (DEFINE) の保持 1214
KEEPDEFINES による一時項目 (DEFINE) の保持 1214
KEEPDEFINES の使用 1215
KEEPDEFINES の使用上の注意 1215
  KEEPDEFINES による JOIN 解析中の一時項目 (DEFINE) の保持 1216

DEFINE FILE SAVE と RETURN による一時項目 (DEFINE) の保持 1218
DEFINE FILE SAVE および RETURN による一時項目 (DEFINE) の保持 1219
条件付き JOIN 式によるセグメントの選別 1221
JOIN の WHERE 条件の解析 1221
JOIN 構造の表示 1222
JOIN 構造の表示 1222
JOIN 構造の表示 1222
JOIN 構造の一覧表示 1223
JOIN 構造のクリア 1224
JOIN のクリア 1224
条件付き JOIN のクリア 1225
JOIN のクリア 1225

データソースの結合 (MATCH) 1227
データの結合 1227
データソースの結合 1228
MATCH 処理のタイプ 1229
MATCH 処理の制御 1230
MATCH リクエスト使用上の注意 1230
データソースの結合 (MATCH) 1231
グループ化と非グループ化処理の比較 1232
結合句 1235
結合句の指定 1236
共通上位ソートフィールドによる MATCH プロセス 1238
共通上位ソートフィールドによる MATCH プロセス 1239
共通上位ソートフィールドによる結合 1242
共通上位ソートフィールドのない結合 1242
MATCH プロセスの細部調整 1245
MATCH プロセスでの表示コマンドの使用 1245
ユニバーサル連結 1248
データソースの連結 1249
データソースの連結 1250
フィールド名およびフォーマットの一致 1251

フィールド名およびフォーマットの一致 1252
連結したデータソースの統合 1254
連結したデータソースの結合 1255
MATCH リクエストでのソートフィールドの使用 1256
共通上位ソートフィールドによる連結したデータソースの結合 1257
共通ソートフィールドのない連結データソースの結合 1259
直積 1260
直積の有効化と無効化 1260
直積使用上の注意 1261
直積によるレポートの実行 1261

レポートフォーマットの概要 1264
利用可能なフォーマット 1264
レポートフォーマットの利点 1265
レポートフォーマットの指定 1266
注文の収入レポートへのフォーマットの指定 1267
スタイルシートタイプの選択 1270
標準およびレガシーフォーマット 1271
すばやく簡単なフォーマット設定のためのテクニック 1271
レポートからその他のリソースへのナビゲーション 1272

WebFOCUS スタイルシートの作成と管理 1274
WebFOCUS スタイルシートの作成 1274
リクエストでの WebFOCUS スタイルシートの作成 1275
リクエストでの WebFOCUS スタイルシートの作成 1275
リクエストでの WebFOCUS スタイルシートの作成 1276
スタイルシートファイルへの別スタイルシートの追加 1277
スタイルシートファイルへの別スタイルシートの追加 1278
WebFOCUS スタイルシートファイルの作成と適用 1278
WebFOCUS スタイルシートファイルの名前を付けて保存 1279
WebFOCUS スタイルシートファイルのレポートへの適用 1279
一般的な WebFOCUS スタイルシート構文 1280
WebFOCUS スタイルシート宣言の指定 1280

サンプルの WebFOCUS スタイルシート 1281
WebFOCUS スタイルシートの読みやすさの向上 1281
WebFOCUS スタイルシートへのコメントの追加 1282
マクロを使用した WebFOCUS スタイルシート宣言の再利用 1283
WebFOCUS スタイルシートマクロの定義 1283
WebFOCUS スタイルシートマクロの定義 1283
WebFOCUS スタイルシートマクロの適用 1284
WebFOCUS スタイルシートマクロの適用 1284
WebFOCUS スタイルシートマクロの定義、適用、上書き 1285
WebFOCUS スタイルシート属性の継承 1286
WebFOCUS スタイルシート継承の階層 1287
継承した WebFOCUS スタイルシート属性の補足 1288
継承した WebFOCUS スタイルシート属性の上書き 1289
JAWarm スタイルシートを使用したレポートの作成 1291
レポートのスタイル設定 1291
データ、レポート、タイトルのスタイル設定 1291
見出しおよび脚注スタイル 1292
中間見出しおよび中間脚注のスタイル設定 1293
ACROSS のスタイル設定 1295
中間合計および列合計のスタイル設定 1295
Analytic Document 1296
WebFOCUS Analytic Document のページ、メニュー、Hover テキストへのスタイル設定 1296
JAWarm.sty 使用上の注意 1297

レポートフォーマット設定の管理 1298
HTML レポート内部カスケードスタイルシートの生成 1298
内部カスケードスタイルシートの生成 1299
内部カスケードスタイルシートの要件 1300
測定単位の選択 1300
測定単位の設定 1300
スタイルシートでの条件付きフォーマット設定、表示、リンク 1301

連続条件付きフォーマット設定の適用 1302
スタイルシートでの条件付きのフォーマット設定、表示、リンク 1302
連続条件付きのフォーマット設定の使用 1304
基本的な条件付きフォーマット設定の適用 1306
フィールドの条件付きフォーマット設定 1307
ACROSS 値に基づく条件付きスタイル設定 1308
ピアグラフの条件付きフォーマット設定 1309
非表示 (NOPRINT) のフィールド値を基準にした条件付きフォーマット設定の適用 1310
ソートグループへの条件付きフォーマット設定の適用 1312
予測値への条件付きフォーマット設定の適用 1314
集計行、下線、ブランク行、改ページの追加 1316
集計行、下線、ブランク行、改ページの表示 1316
ON 句処理の制御 1317
存在しないフィールドの ON 句の無視 1318
集計行、下線、ブランク行、改ページの条件付き追加 1319
集計行、下線、ブランク行、改ページの条件付き表示 1319
ソートオプションでの WHEN 条件の使用 1321
複数ソートオプションでの WHEN 条件の使用 1322
空レポートの表示制御 1323
空レポートの表示制御 1324
空レポートの表示制御 1325
スタイルシートデフォルト値のみによるレポートフォーマット設定 1326
スタイルシートデフォルト値のみによるレポートフォーマット設定 1326

WebFOCUS スタイルシートを使用したレポートコンポーネントの選択 1328

レポート全体、列全体、行全体の選択 1328
レポート全体の選択 1329
レポート全体の選択 1330
列全体の選択 1331
列全体の選択 1332
ACROSS フィールド全体の選択 1333

FML 行全体の選択 1334
FML 行全体の選択 1335
合計または中間合計行全体の選択 1335
合計行全体の選択 1336
行合計の選択 1337
SUBTOTAL および GRANDTOTAL 行のタグの選択 1338
中間合計タグおよび総合計タグのスタイル設定 1339
SUBTOTAL および GRANDTOTAL タグのスタイル設定 1340
データの選択 1341
全データの選択 1342
レポート全データの選択 1342
データフィールドの選択 1343
データフィールドの選択 1344
ACROSS フィールドデータ行の選択 1345
ACROSS フィールドデータ行の選択 1345
ACROSS フィールドデータ合計行 (ACROSS-TOTAL) の選択 1346
合計と中間合計の選択 1347
総合計、中間合計、中間合計計算の指定 1348
総合計の選択 1349
中間合計の選択 1350
中間合計の計算 (RECAP/COMPUTE) の選択 1352
見出し、脚注、タイトル、FML テキストの選択 1353
フィールドタイトルと行タイトルの選択 1354
フィールドタイトルの選択 1354
ACROSS フィールドのタイトルおよび値の選択 1355
フィールドタイトルおよび ACROSS フィールド値の選択 1356
FML レポートのテキストの選択 1357
FML レポートのテキストの選択 1357
見出しまたは脚注の選択 1358
見出しおよび脚注の選択 1360
ページ見出しとレポート脚注の選択 1361
見出しと脚注の特定行の選択 1362

見出しの特定行の選択 1362
見出しまたは脚注テキストの選択 1363
スポットマーカーによる見出しテキストの選択 1364
見出しと脚注の埋め込みフィールドの選択 1365
見出し埋め込みフィールドの選択 1366
WHEN 句に基づく特定の見出しまたは脚注の選択 1367
WHEN による複数見出しのスタイル設定 1367
ページ番号、下線、ブランク行の選択 1369
ページ番号、下線、ブランク行の選択 1370
下線とページ番号の選択 1370
ブランク行の選択 1371

外部カスケードスタイルシートの使用 1373
カスケードスタイルシートの概要 1373
カスケードスタイルシートのルールおよびクラス 1374
外部カスケードスタイルシートを使用する理由 1375
外部カスケードスタイルシートによるレポートフォーマット設定 1376
外部カスケードスタイルシートを使用してレポートにフォーマットを設定するには 1377
フィールドタイトルクラスの CSS ルール 1378
TD 要素の CSS ルール 1379
レポート内の ACROSS 値への CSS クラスの適用 1379
レポートスタイルの外部カスケードスタイルシートへのリンク 1379
外部カスケードスタイルシート (CSS) によるレポートフォーマット 1380
外部カスケードスタイルシートの使用 1383
外部カスケードスタイルシートの選択 1383
外部カスケードスタイルシートの保存先 1384
複数外部カスケードスタイルシートの使用 1384
外部カスケードスタイルシートの編集 1384
カスケードスタイルシートルールの選択 1385
カスケードスタイルシートクラス名の指定 1386
外部カスケードスタイルシートフォーマットの適用 1386
CLASS 属性を使用した CSS フォーマットの適用 1387

外部カスケードスタイルシート (CSS) とその他のフォーマット設定方法の組み合わせ 1389
外部カスケードスタイルシートと WebFOCUS スタイルシートの組み合わせ 1390
外部カスケードスタイルシートへのリンク 1391
CSSURL 属性およびパラメータの使用 1391
CSSURL 属性を使用した外部カスケードスタイルシート (CSS) へのリンク 1392
CSSURL 属性を使用した外部カスケードスタイルシートへのリンク 1393
CSSURL パラメータを指定した外部カスケードスタイルシート (CSS) へのリンク 1394
継承および外部カスケードスタイルシート 1395
TD 要素からフォーマットを継承するレポートフィールド 1396
HTML 以外のレポートでの外部カスケードスタイルシートの使用 1398
複数の出力タイプでの外部カスケードスタイルシート (CSS) の使用 1399
PDF および HTML 出力との外部カスケードスタイルシート (CSS) の使用 1401
外部カスケードスタイルシートの要件 1403
外部カスケードスタイルシート (SET HTMLCSS ON) 使用上の注意 1404
外部カスケードスタイルシートの使用に関する FAQ 1405
外部カスケードスタイルシートのトラブルシューティング 1407

レポートページのレイアウト 1411
ページサイズ、用紙の方向、色の選択 1411
ページサイズ、用紙の方向、色の属性 1411
ページサイズの設定 1412
用紙方向の設定 1417
用紙方向の設定 1417
ページ色の設定 1418
ページ色の設定 1418
ページマージンの設定 1419
ページマージンの属性 1419
測定単位の設定 1420
マージンサイズの設定 1421
左マージンの設定 1422

レポートコンポーネントの配置 1423
配置属性 1423
列開始位置の指定 1423
列絶対開始位置の指定 1424
列相対開始位置の指定 1425
レポートコンポーネント周囲へのブランクの追加 1426
データ値の上へのブランクの追加 1427
列左側へのブランクの追加 1428
列のページ内整列 1429
列整列機能 1430
列幅の決定 1430
列幅の決定 (HTML) 1431
デフォルト列幅の使用 (HTML) 1432
フィールドフォーマットに基づく列幅の使用 (HTML) 1433
列幅の決定 (PDF) 1433
列幅の決定 (PDF) 1434
列間隔の制御 1435
列間隔の制御 1435
ACROSS フィールド列間隔の制御 1436
列順序の変更 1437
列順序の変更 1437
列順序の変更 1437
列の積み重ね 1438
FOLD-LINE による列の積み重ね 1439
FOLD-LINE 使用による列の積み重ね 1440
OVER 使用による列の積み重ね 1440
OVER 使用による列の積み重ね 1441
PDF レポートでの OVER による列の位置揃え 1442
レポートの列間隔の制御 1443
外側と内側の間隔の比較 1444
レポートでの GAPINTERNAL の使用 1445
列の配置 1449

列の配置 1449
列の配置 1450
ACROSS フィールドの配置 1451
OVER フィールドの配置 1452
フィールドの非表示 1453
フィールド非表示コマンド 1453
フィールドの非表示 1453
ソートフィールドの非表示 1454
中間合計によるソートフィールドの非表示 1456
アルファベット順ソート 1457
改ページの挿入 1458
改ページコマンド 1459
複数テーブル HTML レポートの使用 1459
改ページの挿入 1460
改ページの挿入 1460
複数テーブル HTML レポートの表示 1462
不要な分割の防止 1464
不要な分割の防止 1465
不要な分割の防止 1465
ページ番号の挿入 1468
ページ番号コマンド 1468
現在のページ番号の挿入 1469
ソート項目を指定している脚注への現在のページ番号の挿入 1469
総ページ数の挿入 1470
総ページ数の挿入 1470
TABLASTPAGE 使用上の注意 1470
現在のページ番号および総ページ数の挿入 1471
ソートグループ内での総ページ数の表示 1472
各ソートグループ内の総ページ数の表示 1472
ソートグループ内のページ設定 1473
開始ページ番号の指定 1475
先頭ページへのページ番号の割り当て 1475

先頭ページへのページ番号の割り当て 1476
連続レポート内のページ番号の制御 1476
ページ番号表示の制御 1478
ページ番号の表示制御 1478
デフォルトページ番号の非表示 1479
AHTML リクエストでのレポート各ページのデータ行数の設定 1480
AHTML リクエストでの 1 ページあたりのデータ行数の設定 1480
罫線および境界線の追加 1483
罫線表示属性 1484
HTML レポート内の罫線表示の制御 1485
境界線の追加およびフォーマット 1486
境界線の挿入およびフォーマットの設定 1488
HTML レポート内のデフォルト罫線の表示 1489
罫線の HTML レポートのセル全体への適用 1490
HTML レポートからの罫線の削除 1491
Excel レポート出力への境界線の追加 1492
XLSX フォーマットレポート出力への境界線の追加 1492
見出しまたは脚注への内側と外側の境界線の挿入 1495
PDF レポートでの見出しと脚注に表示される要素の境界線制御 1498
中間見出しと中間脚注マージンのレポートとの位置揃え 1500
中間見出しと中間脚注マージンをレポートと位置揃えする際の注意 1501
単一パネル PDF レポートの中間見出しと中間脚注マージンの位置揃え 1505
複数パネルレポートの中間見出しと中間脚注マージンの位置揃え 1509
罫線の追加および調整 (PDF) 1512
レポートデータへの罫線の適用 (PDF) 1513
PPTX および PDF フォーマットでのボックス境界線の定義 1514
データ行、見出し行、脚注行での上付き文字の表示 1516
レポートのデータ行、見出し行、脚注行での上付き文字の表示 1516
PDF 出力のデータ行および脚注行での上付き文字の表示 1517
XLSX 出力の見出しおよび脚注行での上付き文字の表示 1519
下線およびブランク行の追加 1521
セクション分割機能 1521

ブランク行の追加 1522
ソートグループ間へのブランク行の追加 1523
ブランク行のフォーマット 1524
ブランク行への色の追加 1525
ソートグループの下線 1526
ソートグループの下線 1526
下線のフォーマット 1527
ソートグループの下線フォーマット 1528
レポートコンポーネント下線の追加または削除 1529
フィールドタイトルからの下線の削除 1530
フィールド下線の追加およびフィールドタイトル下線の削除 1530
レポート列幅全体に下線を拡張 1531
EXTUNDERLINE 属性使用上の注意 1532
下線をレポート列幅全体に拡張 1532
FML レポート内の下線の太さの変更 1535
FML レポート (HTML) デフォルト下線の変更 1535
FML レポート (PDF) デフォルト下線の変更 1536
レポートからのブランク行の削除 1537
レポート出力の自動ブランク行の制御 1537
SET DROPBLNKLINE=HEADING 使用時の注意 1538
DROPBLNKLINE パラメータ設定の比較 1539
レポートへのイメージの追加 1543
HTML ドキュメントのイメージをサポートするブラウザおよびデバイス 1544
イメージ属性 1546
HTML レポートへのイメージの追加 1547
URL の指定 1548
HTML レポート見出しへの GIF イメージの追加 1549
埋め込み JPEG イメージ付きレポート見出しの作成 1550
HTML レポートでのデータソースフィールドファイル名の使用 1552
ALT 属性を使用したイメージの説明 1553
背景イメージの追加 1555
背景イメージの追加 1556

内部カスケードスタイルシートを使用した PDF および HTML レポートへのイメージの追加 ... 1557
GIF イメージの内部カスケードスタイルシート付き HTML レポートへの追加 ... 1559
PDF レポートへの GIF イメージの追加 ... 1560
PDF レポートへの PNG イメージの追加 ... 1561
BLOB フィールドに格納されたイメージの PDF、DHTML、HTML レポートへの追加 ... 1562
BLOB フィールドのイメージのレポートフィールドへの挿入 ... 1566
BLOB フィールドのイメージの中間見出しへの挿入 ... 1572
BLOB フィールドのイメージのサイズ変更 ... 1575
概要レポートでの BLOB フィールドのイメージの挿入 ... 1577
BLOB フィールドのイメージに関するファイルサイズと圧縮の考慮事項 ... 1579
レポートデータへのピアグラフの関連付け ... 1580
ピアグラフのフォーマットオプション ... 1581
ピアグラフフォーマット属性の組み合わせ ... 1582
レポートフィールドへのピアグラフの関連付け ... 1585
レポートでのピアグラフの作成 ... 1585
ACROSS フィールドでのピアグラフスケール制御 ... 1586
ピアグラフの方向の設定 ... 1587
ピアグラフの方向の設定 ... 1588
ピアグラフへのスケールの適用 ... 1589
ピアグラフへのスケールの適用 ... 1589
GRAPHSCALE による縦のピアグラフの個別表示 ... 1590
宛名ラベルと複数枠ページの使用 ... 1591
宛名ラベルおよび複数枠印刷用属性 ... 1592
レポートの宛名ラベル印刷を設定するには ... 1592
宛名ラベルまたは複数枠レポートの印刷 ... 1593
宛名ラベルの印刷 ... 1594
複数枠レポートの印刷 ... 1595

見出し、脚注、タイトル、ラベルの使用 ... 1597
見出しと脚注の作成 ... 1597
見出しと脚注の制限事項 ... 1598

見出し脚注リクエストコードの複数行での指定 1599
リクエストの見出し、脚注コードを複数行で指定するには 1599
見出し脚注リクエストコードの複数行での指定 1600
カスタムレポートタイトルまたはワークシートタイトルの作成 1601
カスタムレポートタイトルの作成 1602
HTML カスタムレポートタイトルの作成 1602
EXL2K カスタムレポートタイトルの作成 1603
レポート見出しとレポート脚注の作成 1604
レポート見出しの作成 1604
単数行レポート見出しの作成 1606
レポート脚注の作成 1608
単数行レポート脚注の作成 1610
ページ見出しまたはページ脚注の作成 1611
ページ見出しの作成 1612
単一行のページ見出しの作成 1614
ページ脚注の作成 1615
複数行のページ脚注の作成 1617
WHEN 句による見出脚注の指定 1618
複数見出し使用上の注意 1619
HTML、AHTML、XLSX での 見出し、脚注、列タイトルの固定 1619
HFREEZE で HTMLARCHIVE を使用する際の注意 1620
HTML レポートでのスクロール可能領域の作成 1620
AHTML レポートでのスクロール可能領域の作成 1621
XLSX レポートでのスクロール可能領域の作成 1622
HFREEZE でのブランクフィールドタイトルのサポート 1622
HTML レポート出力の固定領域を使用する際の注意 1623
AHTML レポート出力の固定領域を使用する際の注意 1623
XLSX レポート出力の固定領域を使用する際の注意 1624
ソート項目を指定している見出しおよび脚注の作成 1624
中間見出しおよび中間脚注の位置揃え 1625
ソート項目を指定している見出しの作成 1625
製品説明変更時のソート項目を指定している見出しの作成 1628

条件付きソート項目を指定している見出しの作成 1629
ソート項目を指定している脚注の作成 1630
製品説明変更時のソート項目を指定している脚注の作成 1633
複数ソートオプションを持つ条件付きソート項目を指定している脚注の作成 1634
ソート項目を指定している脚注の非表示 1635
ソート項目を指定している脚注によるデフォルト RECAP ラベルの置換 1636
ソート項目を指定している脚注内の表示コマンドの省略 1637
中間脚注の使用上の注意 1638
見出しまたは脚注内への要素の追加 1638
見出しまたは脚注へのフィールド値の追加 1639
見出しまたは脚注へのフィールド値の追加 1639
ページ見出しおよび脚注内への部門名の追加 1640
見出しでのキャレット記号 (<) の表示 1642
文字フィールド内の末尾のブランクの保持 1642
ページ見出し内の演算接頭語 TOT の使用 1643
ページ見出し内の複数演算接頭語の使用 1644
ソート項目を指定している脚注内の複数演算接頭語の使用 1645
見出しまたは脚注へのテキストフィールドの追加 1646
見出しまたは脚注内のテキストフィールドの制限事項 1646
見出しまたは脚注へのテキストフィールドの追加 1646
ソート項目を指定している脚注へのテキストフィールドの追加 1646
見出しまたは脚注へのページ番号の追加 1647
見出しまたは脚注へのダイアログマネージャ変数の追加 1647
見出しまたは脚注内へのダイアログマネージャ変数の追加 1648
レポート見出しへの現在日付の追加 1649
見出しまたは脚注内へのイメージの追加 1650
見出しおよび脚注オブジェクトでの構文コンポーネントの表示 1651
レポート構文コンポーネントの表示 1651
PDF レポートパネル単位の見出しと脚注の繰り返し 1653
パネルでの見出し要素の繰り返し 1653
レポートパネルでの見出しと脚注すべての繰り返し表示 1655
PDF レポートパネルでの中間脚注の繰り返し表示 1659

パネルページでのスタイル設定された見出しと脚注の繰り返し表示 1663
フィールドタイトルのカスタマイズ 1667
デフォルトフィールドタイトルの使用 1667
フィールドタイトルの制限事項 1669
リクエストでのフィールドタイトルのカスタマイズ 1669
リクエストでのフィールドタイトルのカスタマイズ 1670
フィールドタイトルの非表示 1671
一時項目 (COMPUTE) のフィールドタイトルのカスタマイズ 1672
マスターファイルでのフィールドタイトルのカスタマイズ 1673
重複フィールド名の区別 1673
重複フィールド名の区別 1673
SET コマンドによるフィールドタイトルの下線の制御 1674
SET コマンドによるフィールドタイトルの下線の制御 1674
SET コマンドによるフィールドタイトルの下線の制御 1675
スタイルシート属性によるフィールドタイトルの下線の制御 1676
スタイルシート属性によるフィールドタイトルの下線の制御 1677
スタイルシート属性によるフィールドタイトルの下線の制御 1677
データ識別ラベルの作成 1679
行または列合計ラベルの作成 1679
行または列合計のラベルの作成 1680
行および列合計ラベルの作成 1681
ACROSS による行合計ラベルの作成 1681
中間合計ラベルおよび総合計ラベルの作成 1682
中間合計ラベルまたは総合計ラベルの作成 1682
中間合計ラベルおよび総合計ラベルの作成 1683
特定フィールド中間合計のラベルの作成 1684
一時項目 (COMPUTE) 中間合計のラベルの作成 1685
一時項目 (COMPUTE) 中間合計のラベルの作成 1686
財務レポートの行ラベル作成 1687
見出し、脚注、タイトル、ラベルのフォーマット設定 1687
見出し、脚注、タイトル、ラベルのフォント属性適用 1689
レポート見出しおよびフィールドタイトルへのフォント特性の適用 1689

内部カスケードスタイルシートを使用したレポート見出しフォントサイズの設定 1690
レポート見出しシステム変数へのフォントスタイルの適用 1691
境界線および罫線の追加 1692
PDF レポート見出し周囲への罫線の追加 1693
PDF レポートフィールドタイトルの横線強調 1693
レポート見出し周囲境界線のフォーマット設定 1694
上下境界線によるレポート見出しフォーマット設定 1695
見出し、脚注、タイトル、ラベルの位置揃え 1696
見出しおよび脚注の位置揃え 1696
スタイルシートでの見出しまたは脚注の位置揃え 1697
レポートの見出しの位置揃え 1698
複数行レポート見出しの個別位置揃え 1699
複数行レポート見出し全行の中央揃え 1700
位置揃えの範囲および動作 1701
レガシーフォーマットを使用したページ見出しまたは脚注の中央揃え 1702
ページ見出しの中央揃え 1703
フィールドタイトルの位置揃え 1704
スタイルシートを使用したフィールドタイトルの位置揃え 1705
表示フィールドおよび BY フィールドのタイトル位置揃え 1706
スタイルシートを使用した ACROSS および ROW-TOTAL フィールドタイトルの位置揃え 1707
一時項目 (COMPUTE) フィールドタイトルの位置揃え 1708
レガシーフォーマットを使用した表示フィールドまたは BY フィールドタイトルの位置揃え 1709
レガシーフォーマットを使用した表示フィールドおよび BY フィールドタイトルの位置揃え 1710
行合計ラベルまたは列合計ラベルの位置揃え 1711
レガシーフォーマットを使用した行合計ラベルまたは列合計ラベルの位置揃え 1711
行合計ラベルの作成 1712
中間合計ラベルまたは総合計ラベルの位置揃え 1713
中間合計ラベルまたは総合計ラベルの位置揃え 1713

見出しおよび脚注の位置揃え 1714
HTML、XLSX、EXL2K、PDF、PPTX、DHTML レポートの見出しまたは脚注の位置揃え 1717
HTML または PDF レポートの見出しおよび脚注の位置揃え 1718
PDF レポートでの中間脚注項目に表示されるレポート列との位置揃え 1719
PDF レポートでの HEADALIGN=BODY と OVER の使用 1720
HEADALIGN オプションによる出力結果の比較 1722
ソート項目が指定された脚注要素とデータ列の位置揃え 1726
別の HTML テーブルを使用したページ見出し要素の位置揃え 1727
ソート項目を指定している脚注内のフィールドを使用したテキストフィールドの位置揃え 1729
ソート項目を指定している脚注内のテキストフィールド位置揃えおよびスタイル 1730
HTML または PDF レポートの見出しおよび脚注要素の複数列範囲での位置揃え 1731
HTML または PDF レポートの見出しおよび脚注要素の複数列範囲での位置揃え 1732
3 列にわたるページ見出しの中央揃え 1733
複数フィールドにわたるフィールド値の位置揃え 1734
PDF レポートでのフィールド値の複数列範囲の位置揃え 1735
複数行見出しまたは脚注内容の位置揃え 1737
列内の見出しテキストおよびデータの位置揃え 1737
複数行の見出しまたは脚注内の行と項目のフォーマット 1740
複数行の見出しまたは脚注内のデータおよびテキストの位置揃え 1740
複数行見出し、脚注の小数点位置揃え 1742
列幅を決定するには 1743
小数点の位置を決定するには 1744
見出しおよび脚注の行列フォーマット組み合わせ 1744
列と行のフォーマットの組み合わせを使用した項目の位置揃え 1745
見出し、脚注、その内部の項目の配置 1750
見出しまたは脚注開始位置の設定 1750
PDF レポート見出し開始位置の設定 1751
HTML レポート見出し開始位置の設定 1751

見出しまたは脚注要素開始位置の設定 1752
見出し項目の絶対開始位置設定 1754
見出し項目の相対開始位置設定 1755
フィールド見出し項目の位置揃え 1756
PRINTPLUS の使用 1757
PRINTPLUS の使用 1758
SUBFOOT および FOOTING への PRINTPLUS の使用 1758
スポットマーカーを使用した配置調整 1760
HTML レポートフィールド直後への文字配置 1760
PDF レポートフィールド直後への文字配置 1762
フォント属性位置指定のカスタマイズ 1763
見出しまたは脚注の縦方向配置制御 1764
見出しまたは脚注へのブランク行の追加 1765
レポート見出し上部または下部へのブランク行の追加 1765
見出しまたは脚注内の縦間隔の制御 1766
PDF レポートで罫線行と見出しテキストを分離するブランク 1767
ソート項目を指定している脚注下部縦間隔の調整 1768
ページ下部へのページ脚注の配置 1769
ページ末尾へのページ脚注の配置 1771
レポート見出しまたは脚注の単独ページ配置 1772
レポート見出しまたは脚注の単独ページ配置 1772
別ページへのレポート見出しの配置 1774
レポート脚注の別ページへの配置 1775

レポートデータのフォーマット 1777
レポートのフォントフォーマット指定 1777
レポートのフォントサイズ指定 1777
レポートのフォントサイズ指定 1778
フォントサイズ変更時の注意 1778
レポートの太字、斜体フォント指定 1779
レポートのフォントスタイルの指定 1780
レポートへの継承フォントスタイルの追加および削除 1780
レポートのフォント色指定 1781

レポートの色値 1782
レポートのフォント指定 1785
レポートフォントの指定 1785
HTML レポートでの複数フォントの指定 1786
HTML レポートのデフォルトブラウザフォント指定 1787
デフォルトブラウザフォントの指定 1787
レポート背景色の指定 1788
レポート背景色の指定 1788
レポート背景色およびフォント色の指定 1789
レポートデータ行の代替背景色の指定 1790
レポートデータ行の代替背景色の指定 1791
折り返し行への交互に異なる背景色の適用 1792
折り返し行への交互に異なる背景色の適用 1792
折り返し行への交互に異なる背景色の適用 1793
レポートのデータフォーマットの指定 1795
レポートフィールド値のフォーマット変更 1796
フィールド値のフォーマット変更 1796
フィールド値のフォーマット変更 1796
フィールドの複数フォーマット指定 1797
フィールドフォーマット変更時の注意 1798
フォーマットが再設定されたフィールドのミッシング値の制御 1798
フォーマット再設定フィールドのミッシング値制御 1799
フォーマット再設定フィールドのミッシング値制御 1799
SET COMPMISS の使用上の注意 1801
カンマ表記とコンチネンタル 10 進表記の相違 1801
ミッシング値を表す文字の設定 1801
ミッシング値を表す文字の設定 1801
SET コマンドによる NODATA 文字の設定 1802
リクエストでの NODATA 文字の設定 1802
レポートデータの配置 1804
レポートデータの折り返し制御 1804
レポートデータの折り返し制御 1805

Web ブラウザデータ折り返し機能の使用 1806
レポートデータ折り返しの禁止 1807
OVER による列の積み重ね 1808
ACROSS 値の折り返し 1809
PDF レポートでの ACROSS 値の折り返し 1810
ブランクフィールドタイトルの OVER 1812
ブランク AS 名の OVER と WRAP 1812
ブランク以外のフィールドタイトルの OVER と WRAP 1814
フィールドタイトルの OVER と WRAP 1814
折り返し行の行間の制御 1815
折り返し行の行間指定 1816
WRAPGAP 使用上の注意 1818
レポートフィールドの位置揃え 1818
レポートフィールドの位置揃え 1819
レポートフィールドデータの位置揃え 1819
フィールド基準のフォーマット再設定 1820
フォーマットフィールドの定義および適用 1821
通貨の値の異なる小数位置の表示 1823
複数行 An および AnV フィールド表示 1823
複数行に改行を含む An および AnV フィールドの表示 1823

グラフの作成 1825
内容分析 - グラフ化目的の決定 1825
GRAPH コマンド 1826
GRAPH と TABLE の類似点 1826
GRAPH と TABLE の相違点 1827
TABLE リクエストの GRAPH リクエストへの変換 1828
HTML5 グラフの作成 1829
HTML5 グラフ出力の作成 1829
HTML5 縦棒グラフの作成 1830
コンテナに格納する HTML5 グラフ出力のサイズ変更 1830
グラフタイプの選択 1831
グラフタイプ 1831

スケールの選択 1832
スケールの選択 1833
表示コマンドおよびソート句によるグラフスタイルの決定 1834
折れ線グラフの作成 1835
折れ線グラフの作成 1836
横棒グラフの作成 1837
縦棒グラフの作成 1837
横棒グラフの作成 1838
縦棒グラフの作成 1838
円グラフの作成 1839
円グラフの作成 1840
散布図の作成 1841
散布図の作成 1842
LOOKGRAPH によるグラフスタイルの決定 1842
LOOKGRAPH によるグラフスタイルの指定 1843
折れ線グラフのスタイルオプション 1843
棒グラフのスタイルオプション 1845
円グラフのスタイルオプション 1847
散布図のスタイルオプション 1848
3D グラフのスタイルオプション 1848
面グラフのスタイルオプション 1850
株価グラフのスタイルオプション 1851
極グラフのスタイルオプション 1853
レーダーグラフのスタイルオプション 1853
バブルグラフのスタイルオプション 1854
ブロックグラフのスタイルオプション 1855
その他のグラフタイプ 1855
HTML5 固有のグラフタイプのオプション 1856
X 軸値および Y 軸値の選択 1857
X 軸値および Y 軸値の選択 1858
Y 軸 フィールドの非表示 1858
線形回帰による X 軸値および Y 軸値の補間 1859

複数グラフの作成 1859
複数グラフの統合 1860
グラフの統合 1860
GRMERGE ON による複数グラフの統合 1861
GRMERGE ADVANCED による複数グラフの統合 1862
複数 OLAP グラフの統合 1863
複数 OLAP グラフの統合 1863
OLAP 対応グラフの統合 1864
複数グラフの列表示 1865
複数グラフの列表示 1865
複数グラフの列表示 1866
グラフの日付プロット 1866
グラフでの日付フィールドの表示 1867
X 軸および Y 軸の基本日付サポート 1868
日付サポートの制限事項 1868
Y 軸値の日付フォーマット指定 1869
グラフデータセットの調整 1870
GRAPH リクエストでの WHERE 条件の指定 1870
ミッシングデータ値のグラフ表示 1871
グラフへのミッシング値の表示 1872
グラフでミッシング値を 0 (ゼロ) として表示 1872
ミッシング値をギャップとして表示 1873
0 (ゼロ) への点線としてのミッシング値の表示 1874
補間点線としてのミッシング値の表示 1875
グラフへの条件付きスタイルの適用 1876
条件付きスタイルのグラフへの適用 1876
グラフへの条件付きスタイルの適用 1877
グラフの別リソースへのリンク 1878
グラフの別リクエストへのリンク 1879
追加レポートまたはグラフへのリンク 1881
URL へのリンク 1881
JavaScript 関数へのリンク 1883

複数ドリルダウンリンクの作成 1885
構文ガイドライン 1885
ドリルダウンリンクの概要 1886
パラメータの作成 1886
グラフへのラベルの追加 1887
グラフへの見出しおよび脚注の追加 1887
グラフへの縦 (Y 軸) および横 (X 軸) ラベルの追加 1888
グラフへのカスタムスタイルの適用 1888
グラフの高さと幅の設定 1889
グラフの幅設定 1889
グラフの高さ設定 1889
SET パラメータによるグラフのカスタマイズ 1890
GRAPH リクエストでの SET パラメータの使用 1890
GRAPH リクエストでの SET パラメータの使用 1891
グラフの SET パラメータ値および関数 1891
X 軸固定スケールの設定 1895
X 軸固定スケールの設定 1896
Y 軸固定スケールの設定 1896
Y 軸固定スケールの設定 1897
グラフ API および HTML5 JSON プロパティによるグラフのカスタマイズ 1897
グラフ API によるグラフのカスタマイズ 1898
グラフ API によるグラフのカスタマイズ 1899
イメージファイルとしてのグラフの保存 1900
イメージファイルとしてのグラフの保存 (GRAPHSERVURL 使用) 1900
イメージファイルとしてのグラフの保存 (GRAPHSERVURL 使用) 1901
イメージファイルとしてのグラフの保存 1901
GIF イメージの PDF レポートへの挿入 1902
グラフ保存上の注意 1905

財務レポートの作成 1907
FML レポート 1907
FML リクエストのサンプル 1908

データからの行作成 1910
データソースからの FOR フィールド値の抽出 1911
データソース値からの行の作成 1912
複数レコードからの行作成 1913
OR 句による行の合計値 1914
行内の値の合計 1915
TO 句による値範囲の指定 1915
値の範囲の特定 1916
マスク文字によるタグ値の抽出 1916
マスク文字によるタググループの検索 1917
複数行での同一 FOR フィールド値の使用 1917
複数行での同一値の参照 1918
外部ファイルからのタグの使用 1919
BY 句を使用した FML リクエスト 1919
BY 句と FOR 句を併用した FML リクエスト 1920
リクエストの直接データ入力 1921
リクエストの直接データ入力 1921
リクエストの直接データ入力 1922
行間の計算実行 1922
行間の計算の定義 1923
RECAP の使用上の注意 1924
計算の行参照 1924
明示的行ラベル指定 1925
RECAP 式の暗黙的行ラベル参照 1925
計算の列参照 1927
RECAP 式への列宣言の適用 1928
列番号参照を使用した計算 1928
RECAP 式の列番号参照 1929
連続列参照を使用した計算 1930
連続列の RECAP 計算 1930
列アドレス参照を使用した計算 1930
RECAP 式の列アドレス使用 1931

RECAP 式への列アドレスの適用 1931
相対列アドレス参照を使用した計算 1932
RECAP 式への相対列アドレス適用 1932
列参照番号生成の制御 1933
作成する列番号の制御 1933
列値参照を使用した計算 1934
RECAP 式の値による列参照 1934
行列参照を使用した計算 1935
マトリックス内の複数 RECAP ステートメントへのスタイル設定 1935
セル参照を使用した計算 1936
RECAP 式のセル表記による行列参照 1936
RECAP 式のセル表記による 列参照 1937
関数を使用した RECAP 計算 1938
RECAP コマンドでの関数呼び出し 1938
RECAP コマンドでの関数の呼び出し 1939
テキスト行の挿入 1940
テキストの挿入 1940
テキスト行へのデータ変数の挿入 1941
テキスト行へのデータ変数の挿入 1941
FML レポートへのフィールドの追加 1942
FML レポートへのフィールドの追加 1942
再帰的モデルの作成 1944
再帰的モデルの作成 1945
動的階層レポート 1945
FML 階層の要件 1946
マスターファイルでの階層の定義 1947
FML 階層の表示 1948
FML 階層の表示 1948
FML 階層の表示 1950
FML 階層の統合 1952
FML 階層の概要行の作成 1952
FML 階層の概要行の表示 1954

FML 階層データの任意のレベルおよび層への統合 1955
FML 階層データの統合 1957
階層の手動ロード 1959
別のマスターファイルからの階層のロード 1960
FML 階層の使用上の注意 1961
行タイトルのカスタマイズ 1962
FML の行タイトルのカスタマイズ 1962
タグ行タイトルの変更 1963
FML レポートのフォーマット 1964
列への下線文字の追加 1965
列への下線の追加 1965
FML レポートでの改ページの指定 1966
FML レポートでの改ページの指定 1966
FML レポートの行、列、セルへのフォーマット設定 1967
FML レポートでの行のフォーマット 1968
行とセルへの境界線追加およびフォーマット設定 1974
統一境界線による行の強調 1976
異なる上下左右の境界線による行の強調 1977
セル周囲への統一境界線の追加 1978
FML ラベル、タグ、キャプションへのインデント指定 1979
FML 階層のタグ行のインデント 1980
FML 階層行タイトルのインデント 1982
HTML レポート FML 階層キャプションのインデント 1983
FML 階層キャプションでのデフォルトインデントの使用 1983
FML 階層キャプションインデント値の指定 1984
行の非表示 1985
TAG 行の非表示 1986
データのない行の非表示 1986
データのない行の非表示 1987
中間レポート結果の保存と取得 1987
データの保存 1988
データのファイルへの追加 1988

行の作業ファイルへの追加 1988
作業ファイルからのデータの抽出 1989
作業ファイルからのデータの抽出 1990
FML レポートからの HOLD ファイルの作成 1991
FML レポートからの HOLD ファイルの作成 1991

自由形式レポートの作成 1993
自由形式レポートの作成 1993
自由形式レポートの作成 1994
「EMPLOYEE EDUCATION HOURS REPORT」のリクエスト 1996
自由形式レポートの設計 1998
自由形式レポートへのテキスト挿入 1998
自由形式レポートへのデータフィールド挿入 1999
自由形式レポートへの図形文字挿入 2000
自由形式レポートのレイアウト指定 2000
自由形式レポートでのレコードのソートおよび選択 2001

SQL によるレポートの作成 2002
SQL ステートメントのサポート 2002
サポートされる SQL ステートメント 2002
サポートされない SQL ステートメント 2004
SQL トランスレータの予約語 2004
SQL トランスレータコマンドの使用 2005
SQL トランスレータコマンドの使用 2005
SQL トランスレータコマンドの使用 2006
SQL リクエスト内の TABLE フォーマット句 2007
SQL SELECT ステートメント 2007
FROM 句を伴わない SQL SELECT ステートメントの使用 2008
SQL JOIN 2009
INNER JOIN の作成 2009
OUTER JOIN の作成 2010
SQL トランスレータからの JOIN 名の割り当て 2011
SQL JOIN でのフィールドの修飾名の使用 2012

SQL の再帰的 JOIN の使用 2012
SQL FULL OUTER JOIN の使用 2012
SQL JOIN の考慮事項 2013
直積スタイルのアンサーセット 2013
コンチネンタル式 10 進表記 (CDN) 2014
桁の分割に CDN を使用 2014
SQL リクエストでのフィールド名の指定 2014
区切り文字を含む識別子を持つフィールド名の指定 2015
区切り文字を含むフィールド名の修飾 2015
SQL UNION、INTERSECT、EXCEPT 演算子 2015
数値定数、リテラル、式、関数 2016
日付、時間、タイムスタンプフィールドの SQL トランスレータサポート 2016
日付、時間、タイムスタンプフィールドの SQL トランスレータサポート 2016
SQL トランスレータによる日付時間構成要素の抽出 2018
SQL トランスレータで使用可能な日付、時間、タイムスタンプ関数 2018
SQL トランスレータの日付、時間、およびタイムスタンプ関数の使用 2019
SELECT ステートメント内の SQL トランスレータの日付、時間、およびタイムスタンプ関数の使用 2020
日付時間構成要素を抽出する SQL トランスレータの EXTRACT 関数の使用 2021
EXTRACT 関数の使用 2022
インデックスが最適化された抽出 2022
最適化 JOIN 2022
TABLEF の最適化 2023
SQLTOPTTF によるパフォーマンスの向上 2023

パフォーマンスの改善 2025
データ構造回転による検索速度の改善 2025
代替ビューの要求 2026
データ再構成時の注意 2026
データの再構成 2027
ibi FOCUS データソースの検索速度最適化 2028
自動インデックス検索 2028
インデックス検索の使用 2029

インデックス検索使用上の注意 2029
インデックス検索の使用 2030
TABLEF によるデータ検索 2031
高速テーブル検索による表示 2032

マスターファイルと構造図 2033
EMPLOYEE データソース 2033
EMPLOYEE マスターファイル 2034
EMPLOYEE 構造図 2035
JOBFILE データソース 2036
JOBFILE マスターファイル 2037
JOBFILE 構造図 2037
EDUCFILE データソース 2038
EDUCFILE マスターファイル 2038
EDUCFILE 構造図 2038
SALES データソース 2039
SALES マスターファイル 2039
SALES 構造図 2040
CAR データソース 2041
CAR マスターファイル 2042
CAR 構造図 2043
LEDGER データソース 2043
LEDGER マスターファイル 2044
LEDGER 構造図 2044
FINANCE データソース 2044
FINANCE マスターファイル 2045
FINANCE 構造図 2045
REGION データソース 2045
REGION マスターファイル 2046
REGION 構造図 2046
EMPDATA データソース 2046
EMPDATA マスターファイル 2047

EMPDATA 構造図 2047
TRAINING データソース 2047
TRAINING マスターファイル 2048
TRAINING 構造図 2048
COURSE データソース 2048
COURSE マスターファイル 2049
COURSE 構造図 2049
JOBHIST データソース 2049
JOBHIST マスターファイル 2050
JOBHIST 構造図 2050
JOBLIST データソース 2050
JOBLIST マスターファイル 2050
JOBLIST 構造図 2051
LOCATOR データソース 2051
LOCATOR マスターファイル 2051
LOCATOR 構造図 2052
PERSINFO データソース 2052
PERSINFO マスターファイル 2052
PERSINFO 構造図 2053
SALHIST データソース 2053
SALHIST マスターファイル 2053
SALHIST 構造図 2054
VIDEOTRK、MOVIES、ITEMS データソース 2054
VIDEOTRK マスターファイル 2054
VIDEOTRK 構造図 2055
MOVIES マスターファイル 2056
MOVIES 構造図 2056
ITEMS マスターファイル 2057
ITEMS 構造図 2057
Gotham Grinds データソース 2058
GGDEMOG マスターファイル 2058
GGDEMOG 構造図 2059

GGORDER マスターファイル 2059
GGORDER 構造図 2060
GGPRODS マスターファイル 2061
GGPRODS 構造図 2061
GGSALES マスターファイル 2062
GGSALES 構造図 2062
GGSTORES マスターファイル 2063
GGSTORES 構造図 2063
Century Corp データソース 2064
CENTCOMP マスターファイル 2065
CENTCOMP 構造図 2065
CENTFIN マスターファイル 2066
CENTFIN 構造図 2066
CENTHR マスターファイル 2067
CENTHR 構造図 2069
CENTINV マスターファイル 2070
CENTINV 構造図 2070
CENTORD マスターファイル 2071
CENTORD 構造図 2072
CENTQA マスターファイル 2073
CENTQA 構造図 2074
CENTGL マスターファイル 2075
CENTGL 構造図 2075
CENTSYSF マスターファイル 2076
CENTSYSF 構造図 2076
CENTSTMT マスターファイル 2076
CENTSTMT 構造図 2077

エラーメッセージ 2079
メッセージの表示 2079

テーブル構文の概要と制限事項 2080
TABLE 構文の概要 2080

階層レポート構文の概要 2082
TABLEF 構文の概要 2082
MATCH 構文の概要 2084
FOR 構文の概要 2085
TABLE の制限事項 2086

リクエストのフィールド参照 2088
個別フィールドの参照 2088
　個別フィールドの参照 2089
修飾フィールド名によるフィールドの参照 2089
長い修飾フィールド名の有効化 2090
　修飾フィールド名によるフィールド参照 2090
長い修飾フィールド名使用上の注意 2090
セグメント全フィールドの参照 2091
　セグメント全フィールドの参照 2091
フィールド名のリスト表示 2091
フィールド名、エイリアス、フォーマット情報リスト 2092

Legal and Third-Party Notices 2093

レポート作成の概要

ibi™ WebFOCUS® は、データの取得と分析を行う機能を備えた情報管理システムです。このシステムを使用してレポートをすばやく、簡単に作成することができます。WebFOCUS は、複雑で高度なレポートを作成する機能を備えたシステムです。その特長はシンプルなクエリを使用することにあります。シンプルなクエリから始め、新しい機能を徐々に習得し、最終的に複雑なレポートを作成できるようになります。

WebFOCUS は、データ処理に関して学習したことのないエンドユーザおよびアプリケーションの開発に強力なツールを必要としている開発者のニーズに応えるものです。HTML や WebFOCUS レポート言語のコマンドと構文の知識がなくても、さまざまなツールを利用してレポートとグラフを作成することができます。

レポート作成の要件

レポートの作成に必要な要素は次の 2 つです。

- **データ**　レポートの作成に使用するデータが必要です。セキュリティシステムでデータが保護されている場合は、そのデータソースからレポートを作成するための許可が必要な場合もあります。また、サーバがデータソースの場所を特定する必要があります。データソースについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

  レポートの作成には、次のタイプのデータソース (異なるオペレーティングシステム環境が対象) を使用することができます。

  - リレーショナル型 (DB2、Teradata、Oracle、Sybase など)
  - 階層型 (FOCUS など)
  - インデックス使用
  - ネットワーク型 (CA-IDMS など)
  - シーケンシャルデータソース (固定フォーマットおよび区切りフォーマット)
  - マルチディメンション型 (SAP BW や Essbase など)
  - XML 型

- **データ記述**　レポートの作成に使用するデータソースを記述するマスターファイルが必要です。マスターファイルには、データソース内の一連のセグメントと、各セグメント内のフィールドが定義されています。データソースの中には、マスターファイルのほかにアクセスファイルを使用するものもあります。

  マスターファイルの情報を見ると、データソースに存在するフィールド名やフォーマットが分かります。また、データソース内のセグメント間の関係を明確にすることができます。この情報を使用しなくても、非常に単純なレポートを作成することはできますが、データソースの構造を理解しておくことで、創造的で洗練されたレポートを作成できるようになります。

  マスターファイルの情報は、データソースの構造を図で表現することにより、より明確にすることができます。これにより、データソースのセグメント間の関係が分かります。この場合、次のコマンドを使用します。

  ```
  CHECK FILE filename PICTURE RETRIEVE
  ```

  表示されるマスターファイルの構造図において、各セグメントは取得される順に表示されます。各セグメントには、それぞれ 4 つのフィールドが表示されます。詳細は、レポートデータの表示を参照してください。

レポートタイプ

WebFOCUS では、グラフィカルツールを使用して、次の基本タイプのレポートを作成できます。

- **表形式レポート**　情報を行と列で表示します。レポートの基礎概念を取り入れた基本的なレポートタイプです。他のレポートフォーマットの大部分は、この概念に基づいています。これらのレポートは、HTML、Excel、PDF などのフォーマットで表示することができます。
- **財務レポート**　財務関係データの作成、計算、公開といった作業のために特別に設計されたレポートです。たとえば、バランスシート、連結決算、予算などのレポートがあります。これらのレポートは、レポートキャンバスを使用して作成できます。マトリックスレポートキャンバスを使用することで、レポートのコンテンツを行単位で定義することができます。この定義方法には、さまざまな利点があり、次の処理を実行することができます。
  - レポートの行ごとのタイトルを識別し、表示する。
  - 行単位で計算を実行し、その計算結果をレポートの任意の位置に挿入する。
  - 1 つのレコードを複数のカテゴリで使用する。

- セル単位で多種多様な書式設定を行う。
- 個別の行と行タイトルを抽出ファイルに保存する。

- **自由形式レポート**1 つのレコードについての詳細情報を、レターやフォームでよく使用される形式で表示します。レポートの 1 ページごとに 1 つのレコードの詳細情報を表示する場合、自由形式レポートを使用して次のことを行えます。
  - 見出し、脚注、テキスト、フィールドをページ上に正確に配置する。
  - 表示変数としてフィールドを追加することにより、見出しと脚注をカスタマイズする。
  - 演算接頭語を見出しと脚注に組み込んで、1 つのフィールドの集計値を計算する。
  - BY ソートを使用して、1 つまたは複数のレポートレコードを各ページに配置する。

  自由形式レポートについての詳細は、自由形式レポートの作成を参照してください。
- **グラフ**　表形式レポートと同じ種類の情報を表しますが、2 次元 (2D) および 3 次元 (3D) のさまざまなグラフ形式で表示することができます。

  詳細は、グラフの作成を参照してください。
- **SQL リクエスト**　SQL レポート言語を使用してデータを取得し、WebFOCUS フォーマットコマンドを直接組み込むことができます。詳細は、SQL によるレポートの作成を参照してください。
- **ドリルスルーレポート**　概要レポートと詳細レポートの両方を含む PDF ファイルを作成することができます。詳細レポートには、概要レポートで指定されたフィールドに対する詳細データがすべて含まれています。ドリルスルーのハイパーリンクをクリックすると、PDF ファイル内部のリンク先に移動します。他のレポートは実行されません。ドリルスルーレポートは静的なレポートです。PDF ファイルはディスクに保存したり、ReportCaster で配信したりすることができます。ドリルスルー機能は、Adobe Reader を使用します。ドリルスルー機能についての詳細は、ドリルスルーリンクを使用した PDF 複合レポートの作成を参照してください。複合レポートについての詳細は、複合レポートの作成を参照してください。
- **Excel 複合レポート**　EXL2K 出力フォーマットを使用して、複数ワークシートレポートを作成することができます。デフォルト設定では、複合レポートの各コンポーネントレポートは、新しい Excel ワークシートとして作成されます。NOBREAK キーワードを使用すると、次のレポートは同一ワークシート上の現在のレポートの下に作成されます。

  詳細は、EXL2K による Excel 複合レポートの作成を参照してください。

- **Excel 目次レポート**　複数ワークシートレポートを作成することができます。WebFOCUS レポートの最初の BY フィールドの値が変わるたびに、それぞれ個別のワークシートが生成されます。

  詳細は、表示フォーマットの選択を参照してください。

リクエストの作成

レポートの作成で要求されることは、データソースを識別することのみです。リクエストの構造は柔軟性に富んでいるため、後は必要なレポート要素を追加するだけです。たとえば、レポートのソートを行う場合はソート順の指定、データのサブセットでレポートを作成する場合は選択基準の追加を行います。

リクエストは、TABLE FILE コマンドで開始し、END コマンドで終了します。レポートの内容とフォーマットは、リクエストの開始から終了までコマンドと句により定義されます。これらのコマンドと句はオプションとして使用し、必要とするレポート機能を生成するものだけを追加します。

リクエストの作成に使用するオプションとして、よく使用するものは次のとおりです。

- **フィールド (列) の指定**　レポートの各列は、それぞれのフィールドを表します。ここでは、表示するフィールド、レポートのソートに使用するフィールド、レコードの選択に使用するフィールド、一時項目の作成に使用するデータソースフィールドを指定することができます。レポートに含めるフィールドを指定することは、基本的にレポートでそのフィールドをどのように使用するかということと関係しています。
- **データの表示**　レポートにデータを表示する方法として、フィールドの全レコードを表示する詳細表示と、フィールドのレコードの合計を表示する概要表示があります。また、フィールドの値で計算を行ったり、他の演算を実行したりすることもできます。たとえば、フィールドの最大値やフィールドのすべての値から平方和の平均を計算し、レポートにその演算結果を表示します。
- **レポートフィールドのソート**　レポートをソートして、フィールド情報を整理することができます。WebFOCUS は、ソート順を制御するソートフィールドを表示します。このフィールドは、縦方向にソートする場合はレポートの左側に、横方向にソートする場合はレポートの上部に表示されます。ソートフィールドは、値が変わったところで表示されます。ソートフィールドを非表示にすることもできます。

  データは、縦 (列方向) にソートすることも、横 (行方向) にソートすることもできます。さらに、縦のソートと横のソートを組み合わせて、単純なマトリックスを作成することもできます。
- **レコードの選択**　レポートを作成する場合、そのレポートにすべてのレコードを含めた

くない場合があります。選択条件に基づいてレコードを選択し、データソースのサブセットを定義すると、そのサブセットだけのレポートを作成することができます。必要に応じて、単純な選択条件を設定したり、複雑な選択条件を設定したりすることが可能です。

- **中間合計と合計の表示**　レポートには、列と行の総合計、中間合計を表示することができます。
- **表示のカスタマイズ**　目的にかなったレポートを作成するには、どのデータを表示するか、そのデータをどのように表示するかという 2 点が重要になります。データフィールドを単純に表示するよりも、関連性のあるデータグループを識別し、重要な事実だけを強調するレポートの方が効果的です。たとえば、次のことが可能です。
  - フィールドタイトルに意味のある名前を付ける。
  - どのフィールドをレポートに表示させるかを制御する。
  - レポートの各レベル (例、ソートグループ、ページ、レポート全体) にそれぞれ異なる見出しと脚注を入れたり、設定した条件に基づいて表示する見出しと脚注を動的に制御したりする。
  - フォント、色、罫線、イメージを追加する。
  - 関連するデータグループを強調し、他のグループと区別するためにブランク行、下線、ページ区切りを挿入する。
- **一時項目の作成**　レポートを作成する場合、そのレポートに含めるフィールドは、データソース内に実在するフィールドに限られるわけではありません。一時項目を作成し、実在するデータソースのフィールドから値を取得して、それをレポートに含めることができます。

  詳細は、一時項目の作成を参照してください。
- **データソースの結合**　複数のデータソースを結合して大規模なデータ構造を構築し、1 つのリクエストでそのデータ構造からレポートを作成することができます。

  詳細は、データソースの結合 (JOIN)を参照してください。
- **結果の保存と再利用**　レポートのデータを 1 つのデータソースとして保存しておき、それに対して新しいクエリを実行することができます。この方法は、データソースのサブセットを作成したり、2 段階でレポートを作成したりする際に役立ちます。また、この新しいデータソースのフォーマットを変更して、スプレッドシートやワードプロセッサなどの他のデータ処理ツールで使用することができます。

  詳細は、レポート出力の保存と再利用を参照してください。

リクエストを 1 回だけのクエリとして実行したり、それを 1 つのプロシジャとして保存したりすることができます。リクエストをプロシジャとして保存しておくと、そのリクエストを必要な時に実行または編集することができます。

リクエストの開始

リクエストは、データソースを指定することから開始します。続いて、リクエストの詳細を定義します。 データソースは、次の方法で指定することができます。

- TABLE FILE filename コマンドを使用して、1 つのリクエストに対してデータソースを設定する。
- FILE SET パラメータを使用して、1 つのプロシジャ内のすべてのリクエストに対してデータソースを設定する。

  FILE SET パラメータについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

リクエストの開始

リクエストを開始するには、次のコマンドを使用します。

```
TABLE FILE filename
```

説明

filename

レポートに使用するデータソースです。

リクエストの発行時のヘルプの表示

プロシジャではなく、コマンドプロンプトからリクエストを対話的に発行した場合は、END コマンドの発行後に、オンラインエラーを修正するためのヘルプテキストが提供されます。たとえば、コマンドプロンプトに次のように入力すると、

```
TABLE FI EMPLOYEE
PRINT LAST_NAME FIRST_NAME
BY DEPARTMENT
END
```

次のエラーメッセージが表示されます。

```
(FOC001) THE NAME OF THE FILE OR THE WORD 'FILE' IS MISSING
```

ただし、エラーメッセージから提供された情報が十分ではない場合は、次のコマンドを発行してエラーの詳細な説明を確認します。

```
? errornumber
```

ユーザが入力したすべてのコマンドはスキャンされ、END コマンドが入力されると即時にレポートが生成されます。

MESSAGE パラメータの値が ON (デフォルト値) になっている場合は、レポートの先頭に、データソースから取得されたレコードの数およびレポートに表示される行の数が表示されます。

リクエストの終了

リクエストを終了するには、END コマンドまたは RUN コマンドを使用します。これらのコマンドは、それぞれ 1 行で記述する必要があります。リクエストを実行しないまま終了するには、QUIT コマンドを使用します。

1 つのセッション中に同じデータソースに対して連続的にリクエストを発行する場合は、RUN コマンドを使用することもできます。RUN コマンドを使用すると、TABLE セッション中は TABLE 機能とデータソースが継続して有効になります。レポートを表示した後で、別のレポートを作成するために TABLE コマンドを再発行する必要はありません。TABLE セッションを終了するには、最後のリクエストの後で END コマンドを発行します。

レポート作成の例

ここで紹介する例は、WebFOCUS の基本機能を説明するための単純なリクエストです。なお、これ以外にも多くの機能がありますが、それらの機能については、このマニュアルの別のトピックで説明しています。

単純なレポートの作成

次の例は、WebFOCUS の基本機能を示しています。この例には注釈が付いています。このリクエストのコードに付けられた番号と、出力結果の後にある解説の番号がそれぞれ対応して

います。このリクエストは、テキストエディタに次のコマンドを入力して生成することができます。

```
1.JOIN PIN IN EMPDATA TO ALL PIN IN TRAINING AS J1
2.DEFINE FILE EMPDATA
    YEAR/YY=COURSESTART;
3.END

4.TABLE FILE EMPDATA
5.HEADING CENTER
    "Education Cost vs. Salary"
6.SUM EXPENSES AS 'Education,Cost' SALARY AS 'Current,Salary'
7.AND COMPUTE PERCENT/D8.2=EXPENSES/SALARY * 100; AS 'Percent'
8.BY DIV
    BY DEPT
9.WHERE YEAR EQ 1991
10.ON TABLE SUMMARIZE
11.ON TABLE SET STYLE *
    TYPE=HEADING, STYLE=BOLD, COLOR=BLUE,$
    TYPE=REPORT, FONT=TIMES, SIZE=8,$
    TYPE=REPORT, GRID=OFF,$
    ENDSTYLE
12.END
```

出力結果は次のとおりです。

Education Cost vs. Salary

DIV	DEPT	Education Cost	Current Salary	Percent
CE	ADMIN SERVICES	1,250.00	$25,400.00	4.92
	PROGRAMMING & DVLPMT	1,580.00	$40,900.00	3.86
CORP	ACCOUNTING	2,050.00	$62,500.00	3.28
NE	CUSTOMER SUPPORT	1,800.00	$19,300.00	9.33
	MARKETING	3,100.00	$32,300.00	9.60
	SALES	1,800.00	$43,600.00	4.13
SE	CONSULTING	3,350.00	$35,900.00	9.33
WE	MARKETING	9,850.00	$102,200.00	9.64
	PROGRAMMING & DVLPMT	5,310.00	$42,900.00	12.38
	SALES	4,500.00	$100,500.00	4.48
TOTAL		34,590.00	$505,500.00	6.84

このリクエストは、次のように処理されます。

1. JOIN コマンドを使用して、EMPDATA データソースと TRAINING データソースを結合します。これで、あたかも 1 つのデータソースのように、リクエストは両データソースにアクセスできるようになります。

2. DEFINE コマンドを使用して、時分秒フォーマットの項目 (COURSESTART) から 4 桁の年を抽出する一時項目を作成します。
3. END コマンドを使用して、DEFINE コマンドを終了します。
4. TABLE コマンドを使用して、リクエストを開始します。
5. HEADING コマンドを使用して、レポート見出し (Education Cost vs. Salary) を追加します。
6. SUM コマンドを使用して、EXPENSES フィールドと SALARY フィールドの値を合計します。AS 句を使用して、フィールドタイトルの名前を変更します。
7. COMPUTE コマンドを使用して、すでに SUM コマンドで集計し、BY コマンドでソートされた値から一時項目 (COMPUTE) を求めます。
8. BY コマンドを使用して、レポートのデータを DIV フィールドでソート、さらに DEPT フィールドでソートします。
9. WHERE 句で一時項目の YEAR に対して、抽出条件を設定し、1991 年のデータのみを抽出します。
10. ON TABLE SUMMARIZE コマンドを使用して、EXPENSES と SALARY の両フィールドのすべての値を合計し、Percent フィールドの値を再計算します。
11. スタイル情報に基づいて、レポート見出しおよび内容にフォーマットが設定されます。
12. END コマンドを使用して、リクエストを終了します。

レポートのカスタマイズ

目的にかなったレポートを作成するには、どのデータを表示するか、そのデータをどのように表示するかという 2 点が重要になります。データフィールドを単純に表示するよりも、関連性のあるデータグループを識別し、重要な事実だけを強調するレポートの方が効果的です。

レポートに含めるデータとその表示形式を選択後、そのまま続けてカスタムフォーマットでレポートを作成することができます。効果的なレポートを作成するために、さまざまなものをリクエストに追加することができます。次の処理を実行することができます。

- タイトル、見出し、脚注を追加する。タイトル、見出し、脚注の追加、AS 句によるフィールドタイトルの変更、ソートグループ、ページ、レポート全体などのレベル単位で見出しと脚注を作成する。

- フィールドフォーマットの変更およびフィールドタイトルの位置調整を行う。詳細は、レポートデータのフォーマットを参照してください。
- レポートの列幅を指定する。詳細は、レポートデータのフォーマットを参照してください。
- 設定した条件に基づいて、中間合計、見出し、脚注の表示を動的に制御する。詳細は、レポートフォーマット設定の管理を参照してください。
- 関連するデータグループを強調し、他のグループと区別するためにブランク行、下線、ページ区切りを挿入する。
- 設定した条件に基づいて、レポートの特定の値だけに色を使用し、そのデータを強調する。詳細は、レポートデータのフォーマットを参照してください。
- 外部カスケードスタイルシートを使用して、レポートにフォーマットを設定する。詳細は、外部カスケードスタイルシートの使用を参照してください。
- レポートにドリルダウン機能を追加する。この機能により、現在のレポートに別のレポートまたは URL をリンクさせて、さらに詳細な情報を提供できるため、レポートに付加価値が付きます。詳細は、レポートの別リソースへのリンクを参照してください。

レポート出力先の選択

レポートを作成した後、そのレポートを次の出力先に送信することができます。

- **画面**　レポートを実行する場合、そのレポートのデフォルトの出力先は画面です。WebFOCUS で実行したレポートは通常、Web ブラウザまたは Web ブラウザ内で開く Adobe Reader や Microsoft Excel などのヘルパーアプリケーションに表示されます。

  レポートは、ブラウザ以外に、Adobe Reader や Microsoft Excel などのスタンドアロンヘルパーアプリケーションで表示することもできます。Windows エクスプローラの [フォルダオプション] の [ファイルタイプ] タブで、[詳細設定] を選択し、.pdf、.xls などのファイルタイプで [同じウィンドウで開く] オプションのチェックをオフにします。[同じウィンドウで開く] オプションが選択されていない場合は、レポート出力はヘルパーアプリケーションウィンドウに表示されるため、WebFOCUS によって作成されたブラウザウィンドウは、ブランクになります。

  WebFOCUS 管理コンソールの [出力先変更設定] 下で [保存レポート] のチェックをオンにすると、出力ファイルを保存するか開くかを選択するウィンドウが表示されます。プロシジャに、出力ファイルに AS 名を指定する PCHOLD コマンドが含まれている場合、ファイルを保存する際に、その名前が保持されます。AS 名が指定されていな

い場合、無作為なファイル名が生成されます。

出力が GRAPH リクエストの結果として作成された場合、JPEG、GIF、SVG などの一時イメージファイルとして保存された実際のグラフ出力へのリンクを含む HTM ファイルが作成されます。このイメージファイルは、最終的にはサーバから削除されます。グラフ出力の保存についての詳細は、グラフの作成を参照してください。

- **ファイル**　HOLD、SAVE、SAVB コマンドのいずれかを使用して、レポートの出力結果を保存しておき、後から再利用することができます。詳細は、レポート出力の保存と再利用を参照してください。
- **プリンタ**　PDF や HTML などのフォーマットでレポートを印刷する場合は、最初にそのレポートを希望する所定のフォーマットで表示しておき、Adobe Reader や Web ブラウザなどの表示用アプリケーションから印刷を実行します。

  WebFOCUS では、レポートをプリンタへ直接出力することはできません。

レポートデータの表示

レポートの作成で最も基本的なプロセスは、データソースからフィールドの値を取得し、その値を表示することです。この方法には次の 3 つがあります。

- 各フィールドの値をリスト表示する (PRINT コマンドと LIST コマンド)。
- すべての値を合計し、その合計値を表示する (SUM コマンド)。
- すべての値の個数を集計し、その数量を表示する (COUNT コマンド)。

リクエストでの表示コマンドの使用

表示コマンドには、PRINT、LIST、SUM、COUNT の 4 つがあります。これらのコマンドは「動詞」とも呼ばれます。柔軟性のあるこれらのコマンドを使用して、たとえば 1 つのコマンドで複数のフィールドからレポートを作成したり、1 つのリクエストに複数の表示コマンドを含めたりすることができます。

リクエストでの表示コマンドの使用

```
display [THE] [SEG.]fieldname1 [AND] [THE] fieldname2 ...
```

または

```
display *
```

説明

display

PRINT、LIST、SUM、COUNT のいずれかのコマンドです。WRITE と ADD コマンドは SUM コマンドと同義なため、SUM の代わりに使用することができます。

SEG.

セグメント内のすべてのフィールドを表示します。セグメントは、マスターファイルで関連付けられたフィールドのグループです。ここでは、セグメント内の任意のフィールド名を指定することができます。

fieldname

レポートに表示するフィールド名です。

レポートに含めることのできる表示フィールドの最大数は、複数の要因の組み合わせにより決定されます。詳細は、リクエストで使用可能な表示フィールド最大数を参照してください。

各フィールドは、リクエストで指定した順序でレポートに表示されます。たとえば、fieldname1 のレポートフィールドを先に表示し、fieldname2 のレポートフィールドをその次に表示するよう指定します。

レポートに表示されるフィールドは、「表示フィールド」とも呼ばれます。

AND

オプションとして、読みやすさを向上するために使用します。任意の 2 つのフィールド名の間に使用します。レポートに影響を与えることはありません。

THE

オプションとして、読みやすさを向上するために使用します。任意のフィールド名の前に置いて使用します。レポートに影響を与えることはありません。

*****

データソースの左パスにあるすべてのフィールドに表示コマンドを適用します。

注意：SEG. と * の両オプションでは、一時項目 (DEFINE) は表示されません。一時項目 (DEFINE) を表示するには、PRINT ステートメントで明示的にそのフィールドを参照します (PRINT * 一時項目 (DEFINE) 名)。これは、一時項目 (DEFINE) が実フィールドの再定義として作成された場合にも適用されます。

個々の値の表示

表示コマンドの LIST コマンドおよび PRINT コマンドを使用して、リクエストで指定したフィールドの値を個別に表示します。LIST コマンドは、レポートの各項目に番号を付けます。PRINT コマンドは、各項目に番号は付けません。

特定のフィールド名を入力する代わりにワイルドカードとしてアスタリスク (*) を使用すると、データソースのすべてのフィールド名を簡単に表示することができます。詳細は、全フィールドの表示を参照してください。

PRINT コマンドおよび LIST コマンドが指定されたすべてのリクエストでは、取得するレコード数と表示する行数は等しくなります。また、レポート行に順序はありません。PRINT コマンドおよび LIST コマンドを使用した場合、データソースで選択したフィールドのすべての値は、その値がアクセスされた順に表示されます。データの表示順序は、AUTOPATH の設定により影響を受ける場合もあります。詳細は、ibi FOCUS データソースの検索速度最適化および『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

通常、PRINT コマンドまたは LIST コマンドを使用する場合、レポートに表示される値の順序は、対象のフィールドがキーフィールドであるかどうかで異なります。

別の方法として、BY (縦方向) または ACROSS (横方向) のいずれかのソート句を使用して値をソートすることもできます。ソート句を含むリクエストで LIST コマンドを使用すると、最も外側のソートフィールドの値が変更されるたびにリストのカウンタがリセットされて 1 に戻ります。ソートについての詳細は、表形式レポートのソートを参照してください。

PRINT * または PRINT SEG.* を使用すると、マスターファイルの実フィールドのみが表示されます。一時項目 (DEFINE) を表示するには、PRINT ステートメントで明示的にそのフィールドを参照します (PRINT * 一時項目 (DEFINE) 名)。これは、一時項目 (DEFINE) が実フィールドの再定義として作成された場合にも適用されます。

PRINT コマンドおよび LIST コマンドの構文については、リクエストでの表示コマンドの使用を参照してください。

特定フィールド値の表示

特定フィールドの値を表示するには、PRINT コマンドを使用します。次のリクエストは、全従業員に対する 2 つのフィールド (LAST_NAME と FIRST_NAME) の値を表示します。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME AND FIRST_NAME
END
```

レポートの出力結果は次のとおりです。

LAST_NAME	FIRST_NAME
STEVENS	ALFRED
SMITH	MARY
JONES	DIANE
SMITH	RICHARD
BANNING	JOHN
IRVING	JOAN
ROMANS	ANTHONY
MCCOY	JOHN
BLACKWOOD	ROSEMARIE
MCKNIGHT	ROGER
GREENSPAN	MARY
CROSS	BARBARA

レコードの一覧表示

レポートの各レコードに番号を付けるには、LIST コマンドを使用します。

```
TABLE FILE EMPLOYEE
LIST LAST_NAME AND FIRST_NAME
END
```

レポートの出力結果は次のとおりです。

LIST	LAST_NAME	FIRST_NAME
1	STEVENS	ALFRED
2	SMITH	MARY
3	JONES	DIANE
4	SMITH	RICHARD
5	BANNING	JOHN
6	IRVING	JOAN

7	ROMANS	ANTHONY
8	MCCOY	JOHN
9	BLACKWOOD	ROSEMARIE
10	MCKNIGHT	ROGER
11	GREENSPAN	MARY
12	CROSS	BARBARA

全フィールドの表示

特定のフィールド名を入力する代わりにワイルドカードとしてアスタリスク (*) を使用すると、データソースの左パスにあるすべてのフィールドを簡単に表示することができます。

全フィールドの表示

次のリクエストで作成されるレポートには、EDUCFILE データソースに存在するすべてのフィールドが表示されます。

```
TABLE FILE EDUCFILE
LIST *
END
```

レポートの出力結果は次のとおりです。

LIST	COURSE_CODE	COURSE_NAME	DATE_ATTEND	EMP_ID
1	101	FILE DESCRPT & MAINT	2010/1/1	212289111

2	101	FILE DESCRPT & MAINT	2010/1/1	117593129
3	101	FILE DESCRPT & MAINT	2010/1/1	071382660
4	101	FILE DESCRPT & MAINT	2010/1/1	451123478
5	101	FILE DESCRPT & MAINT	2010/1/1	112847612
6	102	BASIC REPORT PREP NON-PROG	2010/1/1	326179357
7	103	BASIC REPORT PREP NON-PROG	2010/1/1	212289111
8	103	BASIC REPORT PREP NON-PROG	2010/1/1	117593129
9	103	BASIC REPORT PREP NON-PROG	2010/1/1	112847612
10	104	FILE DESC & MAINT NON-PROG	2010/1/1	326179357

11	106	TIMESHARING WORKSHOP	2010/1/1	326179357
12	202	WHAT'S NEW IN FOCUS	2010/1/1	326179357
13	301	DECISION SUPPORT WORKSHOP	2010/1/1	326179357
14	107	BASIC REPORT PREP DP MGRS	2010/1/1	818692173
15	302	HOST LANGUAGE INTERFACE	2010/1/1	818692173
16	108	BASIC RPT NON-DP MGRS	2010/1/1	315548712
17	108	BASIC RPT NON-DP MGRS	2010/1/1	119265415
18	201	ADVANCED TECHNIQUES	2010/1/1	117593129
19	203	FOCUS INTERNALS	2010/1/1	117593129

全セグメントフィールドの表示

セグメントのフィールドすべてを表示するには、そのセグメントの任意のフィールドに接頭語の SEG. を追加します。

全セグメントフィールドの表示

```
seg.anyfield
```

説明

anyfield

セグメントの任意のフィールドです。

全セグメントフィールドの表示

次のリクエストで作成されるレポートには、QTY_IN_STOCK フィールドを含むセグメントのフィールドがすべて表示されます。

```
TABLE FILE CENTINV
PRINT SEG.QTY_IN_STOCK
BY PRODNAME NOPRINT
END
```

レポートの出力結果は次のとおりです。

Product Number:	Product Name:	Quantity In Stock:	Price:	Our Cost:
1024	110 VHS-C Camcorder 20 X	4000	349.00	249.00
1022	120 VHS-C Camcorder 40 X	2300	399.00	259.00
1020	150 8MM Camcorder 20 X	5961	319.00	240.00
1004	2 Hd VCR LCD Menu	43068	179.00	129.00
1018	250 8MM Camcorder 40 X	60073	399.00	320.00
1016	330DX Digital Camera 1024K P	12707	279.00	199.00
1014	340SX Digital Camera 65K P	990	249.00	199.00
1012	650DL Digital Camcorder 150 X	2972	899.00	710.00
1010	750SL Digital Camcorder 300 X	10758	999.00	750.00
1028	AR2 35MM Camera 8 X	11499	109.00	79.00
1026	AR3 35MM Camera 10 X	12444	129.00	95.00
1006	Combo Player - 4 Hd VCR + DVD	13527	399.00	289.00
1008	DVD Upgrade Unit for Cent. VCR	199	199.00	139.00
1030	QX Portable CD Player	22000	169.00	99.00
1032	R5 Micro Digital Tape Recorder	1990	89.00	69.00
1036	ZC Digital PDA - Standard	33000	299.00	249.00
1034	ZT Digital PDA - Commercial	21000	499.00	349.00

マルチパスデータソース構造とデータ取得順序の表示

表示コマンドを使用する場合、データソースの構造とセグメント間の関係を理解しておくことが重要です。それは、これらの要素が出力結果に影響を与えるためです。CHECK コマンドの PICTURE オプションを使用すると、マスターファイルで定義したデータソースの構造図を表示することができます。

また、CHECK コマンドの PICTURE RETRIEVE オプションを使用して、データソースのデータ取得順序を表示することもできます。この場合、データの取得が最小参照サブツリーにより制御されることに注意してください。

マルチパスデータソース構造の表示

次のコマンドを発行して、CENTINV データソースと CENTCOMP データソースに結合された CENTORD データソースの構造図を表示します。

```
CHECK FILE CENTORD PICTURE
```

構造図の出力結果は、次のとおりです。

```
 NUMBER OF ERRORS=     0
 NUMBER OF SEGMENTS=   4  ( REAL=    2  VIRTUAL=   2 )
 NUMBER OF FIELDS=    23  INDEXES=   4  FILES=     3
 NUMBER OF DEFINES=    8
 TOTAL LENGTH OF ALL FIELDS=  139

SECTION 01
              STRUCTURE OF FOCUS    FILE CENTORD  ON 07/18/03 AT 11.06.34

          OINFO
  01      S1
 **************
 *ORDER_NUM   **I
 *STORE_CODE  **I
 *PLANT       **I
 *ORDER_DATE  **
 *            **
 ***************
  **************
        I
        +-----------------+
        I                 I
        I STOSEG          I PINFO
  02    I KU        03    I S1
 ..............    **************
 :STORE_CODE  :K   *PROD_NUM    **I
 :STORENAME   :    *QUANTITY    **
 :STATE       :    *LINEPRICE   **
 :            :    *            **
 :            :    *            **
 :............:    ***************
  JOINED  CENTCOMPFO**************
                         I
                         I
```

```
                    I
                    I INVSEG
                 04 I KU
                ............
                :PROD_NUM    :K
                :PRODNAME    :
                :QTY_IN_STOCK:
                :PRICE       :
                :            :
                :............:
                 JOINED  CENTINV FOCUS   A1
```

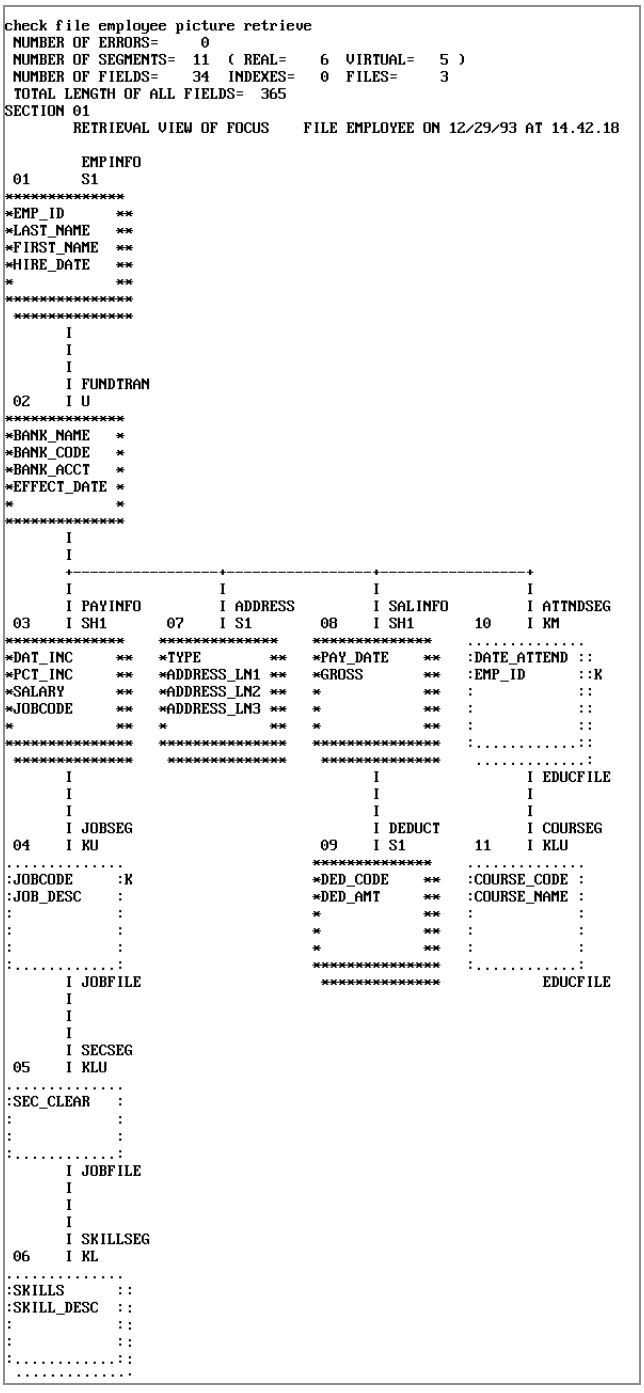
マルチパスデータソース取得順序の表示

次のコマンドを発行して、JOBFILE データソースと EDUCFILE データソースに結合された EMPLOYEE データソースのデータ取得順序を表示します。

```
CHECK FILE EMPLOYEE PICTURE RETRIEVE
```

コマンドの出力結果は次のとおりです。ここでは、セグメントのデータ取得順序を示す番号が各セグメントの左上に追加されています。FUNDTRAN などのユニークセグメントは、データ取得時には、親セグメントに論理的に追加されたものとして扱われます。FUNDTRAN セグメントと SECSEG セグメントはユニークセグメントであるため、親セグメントの一部として扱われます。

次の出力結果には、データ取得順序が表示されています。



マルチパスデータソースのフィールドの表示

次のリクエストで作成されるレポートには、EMPLOYEE データソースの左パスにあるすべてのフィールドが表示されます。

```
TABLE FILE EMPLOYEE
PRINT *
END
```

次のリストは、上記のリクエストで作成された出力フィールドです。レポートのサイズが大きいため、ここにはすべてのインスタンスが表示対象となるフィールド名のみを列記しています。実際のレポートでは、EMP_ID フィールドを先頭に、各フィールドが左から右へと表示されます。

```
EMP_ID
LAST_NAME
FIRST_NAME
HIRE_DATE
DEPARTMENT
CURR_SAL
CURR_JOBCODE
ED_HRS
BANK_NAME
BANK_CODE
BANK_ACCT
EFFECT_DATE
DAT_INC
PCT_INC
SALARY
JOBCODE
JOBCODE
JOB_DESC
SEC_CLEAR
SKILLS
SKILL_DESC
```

このリストの各フィールドは、EMPLOYEE データソースの左パスのセグメントに表示されます。EMPLOYEE データソースのデータ取得順序の構造については、マルチパスデータソース取得順序の表示を参照してください。

ヒント：境によって異なりますが、マルチパスデータソースで PRINT * コマンドを使用すると、次のように PRINT * コマンドでは左パスのみが表示されることを知らせる警告メッセージが表示される場合があります。

```
(FOC757) 警告。マルチパスファイルに PRINT/COUNT * を使用しています
```

値の合計

数値フィールドの値を合計するには、SUM、WRITE、ADD のいずれかのコマンドを使用します。これら 3 つのコマンドは同義であるため、相互に代用することができます。このマニュアルの SUM コマンドに関する説明は、WRITE コマンドと ADD コマンドにも適用されます。

SUM コマンドを使用する場合、データソースから複数のレコードが読み取られますが、出力されるのは合計の行のみです。SUM コマンドを使用する場合、たとえば文字フィールド、テキストフィールド、日付フィールドなどの数値フィールド以外を指定すると、値の合計は実行されず、データソースから最後に取得された値が代わりに表示されます。これは、SUMPREFIX パラメータを使用して、最初の値、最小値、最大値に変更することができます。

SUM コマンド、WRITE コマンド、ADD コマンドの構文についての詳細は、リクエストでの表示コマンドの使用を参照してください。

値の合計

次のリクエストは、CURR_SAL フィールド値をすべて合計します。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL
END
```

リクエストの出力結果は次のとおりです。

CURR_SAL	

$222,284.00	

数値以外の合計

次のリクエストでは、数値以外のフィールドの値を合計してみます。数値以外のデータを合計するリクエストでは、データソースから最後に取得されたレコードが返されます。

```
TABLE FILE EMPLOYEE
SUM LAST_NAME AND FIRST_NAME
END
```

リクエストの出力結果は次のとおりです。

LAST_NAME	FIRST_NAME
---------	----------
CROSS	BARBARA

なお、日付フォーマットのフィールドを合計するリクエストでも、データソースから最後に取得したレコードが返されます。

ヒント：SUMPREFIX パラメータを FST、MIN、MAX、LST に設定して、ソート順を制御することができます。詳細は、表形式レポートのソートを参照してください。

値の集計

指定したフィールドに存在するインスタンスの個数を集計するには、COUNT コマンドを使用します。COUNT コマンドは、BY 句と組み合わせて使用するとさらに効果的です。詳細は、表形式レポートのソートを参照してください。

COUNT コマンドは、レポートのデータ値ではなく、インスタンスの個数を集計します。

COUNT コマンドの構文についての詳細は、リクエストでの表示コマンドの使用を参照してください。

デフォルト設定では、COUNT フィールドのフォーマットは 5 桁の整数に設定されています。COMPUTE コマンドを使用すると、このフォーマットを再設定することができます。フィールドの長さは、SET COUNTWIDTH パラメータで変更します。COMPUTE コマンドについての詳細は、一時項目の作成を参照してください。SET COUNTWIDTH についての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

リクエストに COUNT コマンドを使用した場合、フィールドタイトルを AS 句で変更しない限り、フィールドタイトルにはデフォルトの「COUNT」という語句が追加されます。

値の集計

EMPLOYEE データソースに入力されている従業員数を割り出すには、従業員の ID 番号である EMP_ID のインスタンスの個数を集計します。

```
TABLE FILE EMPLOYEE
COUNT EMP_ID
END
```

リクエストの出力結果は次のとおりです。

```
EMP_ID

COUNT

------

    12
```

ソート句による値の集計

次のリクエストは、EMP_ID のインスタンスの個数を部署別に集計します。

```
TABLE FILE EMPLOYEE
COUNT EMP_ID
BY DEPARTMENT
END
```

リクエストの出力結果を見ると、データソースには EMP_ID の値が全部で 12 個あり、そのうち 6 個は MIS 部から、残りの 6 個は PRODUCTION 部から取得されていることが分かります。

```
             EMP_ID
DEPARTMENT   COUNT
----------   ------
MIS               6
PRODUCTION        6
```

データインスタンス数の集計

次の例では、EMPLOYEE データソースの LAST_NAME フィールド、DEPARTMENT フィールド、JOBCODE フィールドのそれぞれのデータインスタンス数を集計します。

```
TABLE FILE EMPLOYEE
COUNT LAST_NAME AND DEPARTMENT AND JOBCODE
END
```

リクエストの出力結果は次のとおりです。

```
LAST_NAME  DEPARTMENT  JOBCODE
COUNT      COUNT       COUNT
---------  ----------  -------
       12          12       19
```

EMPLOYEE データソースには 12 名分の従業員データがあり、1 名の従業員の LAST_NAME (姓) に対して 1 つのインスタンスが存在します。DEPARTMENT の値は 2 つだけですが、DEPARTMENT フィールドには 12 個のインスタンスが存在します。これは、各従業員が 2 つの部署のうちのいずれか 1 つに所属しているためです。同様に、JOBCODE フィールドには 19 個のインスタンスが存在します。これは、各従業員が雇用期間中に複数の JOBCODE を持つことができるためです。

セグメントインスタンス数の集計

特定のフィールド名を指定する代わりにワイルドカードとしてアスタリスク (*) を使用すると、データソースの左パスにある最下位セグメントのインスタンスの個数を集計することができます。単一セグメントのデータソースでこの方法を使用すると、データソースのすべてのインスタンス数をすばやく集計することができます。

COUNT * コマンドは、セグメントの最初のフィールドのインスタンス数を集計します。最初のフィールドにミッシング値が存在するインスタンスは、個数の集計から除外されます (SET MISSING=ON の場合)。

COUNT * コマンドでは、SET コマンドの ALL パラメータの値に関係なく、ショートパスのセグメントインスタンスは個数の集計から除外されます。

ミッシング値、ショートパス、SET ALL パラメータについての詳細は、ミッシングフィールド値を含むレコード処理を参照してください。

マルチパスデータソースのセグメント数の集計

次のリクエストは、EMPLOYEE データソースの SKILLSEG セグメントに存在するインスタンス数を集計します。

```
TABLE FILE EMPLOYEE
COUNT *
END
```

リクエストの出力結果は次のとおりです。

```
COUNT *

COUNT

-------

      19
```

COUNT * コマンドを使用して、SKILLSEG セグメントのインスタンス数を集計します。このセグメントは、EMPLOYEE データソース構造の左パスにある最下位セグメントです。なお、EMPLOYEE データソースは、JOBFILE データソースと EDUCFILE データソースに結合されています。パス構造の図は、マルチパスデータソース構造とデータ取得順序の表示を参照してください。

ヒント：環境により異なりますが、たとえば上記の EMPLOYEE データソースなど、マルチパスデータソースで COUNT * コマンドを使用すると、次の警告が表示される場合があります。

```
(FOC757) 警告。マルチパスファイルに PRINT/COUNT * を使用しています
```

COUNT および LIST コマンドの表示桁数拡張

デフォルト設定では、COUNT コマンドおよび LIST コマンドで取得した個数の表示桁数は、5 桁です。この桁数は、9 桁に増やすことができます。

たとえば、取得したフィールドのレコード数が 99,999 (5 バイト) を超える場合、オーバーフロー状態を示すアスタリスク (*) がレポートに表示されます。この場合は、SET COUNTWIDTH コマンドを使用して、表示桁数を 999,999,999 (9 バイト) に増やすことができます。

注意：オーバーフロー文字は、SET OVERFLOWCHAR コマンドを発行して変更することができます。

COUNT コマンドと LIST コマンド用の表示桁数の設定

```
SET COUNTWIDTH = {OFF|ON}
```

説明

OFF

COUNT コマンドおよび LIST コマンドで取得した個数を 5 桁 (バイト) で表示します。取得したフィールドのレコード数が 5 桁を超える場合は、アスタリスク (*) が表示されます。デフォルト値は OFF です。

ON

COUNT コマンドと LIST コマンドで取得した個数を 9 桁 (バイト) で表示します。取得したフィールドのレコード数が 9 桁を超える場合は、アスタリスク (*) が表示されます。

COUNT コマンドと LIST コマンド用の表示桁数の設定

これは、SET COUNTWIDTH = OFF の設定で COUNT コマンドを使用した場合の例です。

```
TABLE FILE filename
COUNT Fldxx
BY Fldyy
END
```

	Fldxx
Fldyy	COUNT
value	*****

これは、SET COUNTWIDTH = ON の設定で COUNT コマンドを使用した場合の例です。

```
TABLE FILE filename
COUNT Fldxx
BY Fldyy
END
```

	Fldxx
Fldyy	COUNT
value	999999999

注意：COUNTWIDTH を ON に設定した場合、この機能によりレポートの幅が影響を受けます。レポートの幅を調整するため、COUNT コマンドまたは LIST コマンドのフィールドに、それぞれ 4 桁分を追加する必要があります。

リクエストで使用可能な表示フィールド最大数

TABLE または MATCH リクエストでは、動詞オブジェクトの数に制限はありません。

ただし、次の条件でエラーが発生します。

- レポート出力フォーマットで、使用可能なフィールドの最大数が設定されている。たとえば、Excel、FOCUS、XFOCUS フォーマットでは、フィールド最大数が設定されています。
- オペレーティングシステムで、最大レコード長が設定され、すべてのフィールドが表示できない。
- 出力の保存に必要なメモリ量が不足している。

データ領域内の表示フィールドの長さの合計が最大容量を超えると、エラーメッセージが表示されます。この問題を解決するには、リクエストのフィールドの数または長さを調整します。

演算接頭語による表示フィールドの操作

演算接頭語を使用すると、フィールドの値に対して計算を直接実行することができます。

注意：フィールドを変更したり、AS 句を使用して ACROSS タイトルを変更したりしない限り、演算接頭語が自動的にタイトルに追加されます。AS 句を使用しない場合、フィールドタイトルは、演算接頭語と、マスターファイル内のフィールド名または TITLE 属性 (存在する場合) を使用して作成されます。

- TITLE 属性が存在しない場合、フィールド名が使用されます。
- マスターファイルに TITLE 属性が存在する場合、次の TITLES パラメータの値に基づいてフィールド名または TITLE 属性のいずれかが選択されます。
  - SET TITLES = ON の場合、TITLE 属性が使用されます。
  - SET TITLES = OFF または NOPREFIX の場合、フィールド名が使用されます。

SET PRFTITLE コマンドを使用して、演算接頭語付きフィールドの説明的かつ翻訳可能な列タイトルを作成することができます。たとえば、次のリクエストは、PRFTITLE を LONG に設定します。

```
SET PRFTITLE = LONG
TABLE FILE WF_RETAIL_LITE
SUM COGS_US CNT.COGS_US AVE.COGS_US CNT.DST.COGS_US MIN.COGS_US MAX.COGS_US
MDN.COGS_US
BY PRODUCT_CATEGORY
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。演算接頭語の名前が、説明テキストに変換されています。

Product Category	Cost of Goods	Count Cost of Goods	Average Cost of Goods	Count of Distinct Cost of Goods	Minimum Cost of Goods	Maximum Cost of Goods	Median Cost of Goods
Accessories	$342,877.00	1484	$231.05	50	$16.00	$2,850.00	$174.00
Camcorder	$453,205.00	1417	$319.83	46	$60.00	$8,610.00	$180.00
Computers	$109,281.00	600	$182.13	8	$81.00	$668.00	$167.00
Media Player	$779,593.00	2249	$346.64	77	$36.00	$1,640.00	$310.00
Stereo Systems	$857,042.00	3338	$256.75	100	$48.00	$2,700.00	$166.00
Televisions	$227,820.00	267	$853.26	22	$275.00	$9,750.00	$550.00
Video Production	$180,540.00	645	$279.91	21	$78.00	$1,880.00	$190.00

PRFTITLE がデフォルト設定の SHORT に設定された場合は、説明テキストの代わりに演算接頭語の名前が使用されます。

演算接頭語およびその機能のリストは、演算接頭語による表示フィールドの操作を参照してください。

演算接頭語の基本

ここでは、演算接頭語の基本的な構文および注意事項について説明します。

演算接頭語の使用

1 つの演算接頭語は 1 つのフィールドに適用し、そのフィールドにのみ影響を与えます。

```
{SUM|COUNT} prefix.fieldname AS 'coltitle'
```

```
{PRINT|COMPUTE} RNK.byfield
```

説明

prefix

任意の演算接頭語です。

fieldname

レポートに表示するフィールド名です。

'coltitle'

レポートフィールドのフィールドタイトルです。一重引用符 (') で囲みます。

byfield

レポートで順位付けに使用する縦方向ソートフィールド名です。

演算接頭語の使用上の注意

- PRINT コマンドと LIST コマンドは、集計値ではなくフィールドの値を個別に表示するため、TOT、DST 以外の他の演算接頭語とは併用されません。
- 接頭語コマンドの結果に基づいてソートを行う場合、BY TOTAL 句を使用して数値フィールドの集計とソートを同時に実行します。詳細は、表形式レポートのソートを

参照してください。

- 演算接頭語を使用する場合、WITHIN 句を使用すると便利です。だたし、WITHIN 句は、SUM などの集計表示コマンドで、演算接頭語 MDN.、MDE.、DST.、CNT.DST.、AVE.DST.、SUM.DST. とともに使用することはできません。WITHIN 句は、PRINT 表示コマンドで、DST. 演算接頭語とともに使用することができます。
- COMPUTE コマンドで、演算接頭語の結果を使用することができます。
- PCT.、RPCT.、CNT.、PCT.CNT. を除いては、結果として得られる値には、演算接頭語を実行したフィールドと同一のフォーマットが適用されます。
- パーセントベースの演算接頭語 PCT. および RPCT. については、PCTFORMAT パラメータを OLD に設定した場合、結果として得られる値に、演算接頭語を実行したフィールドと同一のフォーマットが適用されます。デフォルト設定では、PCTFORMAT は PERCENT に設定されており、演算接頭語付きフィールドはパーセント記号付きで表示され、通貨記号などその他のオプションは削除されます。ただし、このフォーマットは再設定することができます。PCT.CNT. フィールドは、フォーマットが再設定されない限り、小数点以下 2 桁、パーセント記号付きで表示されます。PCTFORMAT = PERCENT が設定されている場合、出力フィールドのフォーマットは元のフィールドのフォーマットに依存します。各フォーマットのフィールド表示は次のとおりです。
  - 精度ベースフォーマット (F、D、M、X) の場合、フィールドの長さは 7 バイト、小数点以下 2 桁で表示されます。
  - パック 10 進数フォーマットの場合、フィールドは元の小数点以下の桁数で表示されます。
  - 整数フォーマットの場合、フィールドに小数点以下の桁は表示されません。
- テキストフィールドには、FST.、LST.、CNT. 演算接頭語のみを使用することができます。
- PCT.、TOT.、PCT.CNT.、RNK.、RPCT. は TABLEF ではサポートされず、TABLEF で使用することはできません。
- ibi™ WebFOCUS® Reporting Server または HOLD ファイルからマスターファイルを作成する場合、演算接頭語をマスターファイル名として使用しないことをお勧めします。

演算接頭語で実行可能な機能

下表は、演算接頭語とその機能の一覧です。

接頭語	関数
ASQ.	統計分析で使用される標準偏差の平方和の平均を計算します。
AVE.	フィールド値の平均を計算します。
CNT.	フィールド値の件数を集計します。計算結果のデータタイプは常に整数です。
AVE.DST.	フィールド内の固有値の平均を計算します。
CNT.DST.	フィールド内の固有値の個数を計算します。
SUM.DST.	フィールド内の固有値の合計を計算します。
CT.	指定したフィールドの累計を計算します。この演算子は、中間脚注で使用する場合にのみ適用されます。詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。
DST.	データソース単一パスでの個別値の総数を決定します。
FST.	フィールドの最初の物理インスタンスを出力します。数値フィールドまたはテキストフィールドで使用することができます。
LST.	フィールドの最終の物理インスタンスを出力します。数値フィールドまたはテキストフィールドで使用することができます。
MAX.	フィールドの最大値を出力します。

接頭語	関数
MDE.	フィールド値の最頻値を計算します。
MDN.	フィールド値の中央値を計算します。
MIN.	フィールドの最小値を出力します。
PCT.	フィールドの合計値に基づいて、そのフィールドのパーセントを計算します。PCT 演算子は、集計フィールドまたは詳細フィールドのいずれかに使用することができます。
PCT.CNT.	存在するインスタンス数に基づいて、そのフィールドのパーセントを計算します。結果のフォーマットは常に F6.2 で、フォーマットを再設定することはできません。
RNK.	リクエスト内の BY フィールドのインスタンスを順序付けます。PRINT コマンド、COMPUTE コマンド、および IF または WHERE TOTAL テストで使用することができます。
ROLL.	下位の集計行で合計された値を使用して、上位の集計行の値を再計算します。
RPCT.	フィールドの行全体の合計値に基づいて、そのフィールドのパーセントを計算します。
ST.	指定したフィールドの中間合計の値を、レポートのソート区切りの位置に出力します。この演算子は、中間脚注で使用する場合にのみ適用されます。詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。
STDP.	母集団の標準偏差を計算します。

接頭語	関数
`STDS.`	標本の標準偏差を計算します。
`SUM.`	フィールド値を合計します。
`TOT.`	見出し (脚注、中間見出し、中間脚注を含む) に使用するフィールド値を合計します。

フィールド平均値の計算

AVE. 演算接頭語は、指定したフィールドの平均値を計算します。この計算は、表示コマンドの最下位ソートレベルで実行されます。平均値は、ソートグループのフィールド値の合計をレコード数で除算して計算されます。リクエストにソート句が含まれていない場合、AVE. はレポート全体の平均を計算します。

フィールド平均値の計算

次のリクエストは、EMP_ID フィールドの値の相対的な割合を部署別に計算します。

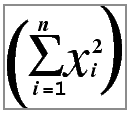
```
TABLE FILE EMPLOYEE
SUM AVE.ED_HRS BY DEPARTMENT
END
```

リクエストの出力結果は次のとおりです。

```
            AVE
DEPARTMENT  ED_HRS
----------  ------
MIS          38.50
PRODUCTION   20.00
```

平方和フィールドの平均計算

ASQ. 演算接頭語は、統計分析において標準偏差の平方和の平均を計算します。次の公式を使用します。



注意：フィールドフォーマットが整数で、大規模な数値セットを取得する場合、ASQ. による結果が、I4 フィールド (2,147,483,647) の上限を超える可能性があります。これより大きい数値の表示では、負の値または正しくない正の値が生成されます。このため、上記の結果が発生する可能性がある場合は、整数フィールドを使用しないことをお勧めします。

平方和フィールドの平均計算

次のリクエストは、DELIVER_AMT フィールドの合計および平方和の平均を計算します。

```
TABLE FILE SALES
SUM DELIVER_AMT AND ASQ.DELIVER_AMT
BY CITY
END
```

リクエストの出力結果は次のとおりです。

```
                              ASQ
CITY             DELIVER_AMT  DELIVER_AMT
----             -----------  -----------
NEW YORK                 300          980
NEWARK                    60          900
STAMFORD                 430         3637
UNIONDALE                 80         1600
```

フィールド最大値と最小値の計算

MAX. および MIN. 演算接頭語は、ソートグループ内の最大値と最小値をそれぞれ出力します。リクエストにソート句が含まれていない場合、MAX. および MIN. は、レポート全体の最大値と最小値を出力します。

フィールド最大値と最小値の計算

次のリクエストは、SALARY の最大値および最小値を計算します。

```
TABLE FILE EMPLOYEE
SUM MAX.SALARY AND MIN.SALARY
END
```

リクエストの出力結果は次のとおりです。

```
    MAX              MIN
    SALARY           SALARY
    ------           ------
$29,700.00        $8,650.00
```

フィールドの中央値および最頻値の計算

MDN. (中央値) および MDE. (最頻値) が使用できます。数値フィールドまたは日付フィールドに集計表示コマンド (SUM、WRITE) を使用する際に、MDN. (中央値) および MDE. (最頻値) 演算接頭語を併用することで、フィールド値の統計として中央値および最頻値を計算することができます。

これらの計算は、DEFINE コマンド、WHERE 式、IF 式、WITHIN 句、集計コマンドではサポートされません。これらの計算を複数動詞リクエストで使用する場合は、最下位の集計で使用する必要があります。

中央値は、一連の値の中で中央に位置する値 (50 パーセンタイル値) です。値の個数が偶数の場合、中央の 2 つの値の平均が中央値になります。最頻値は、一連の値の中で発生頻度の最も高い値です。発生頻度の最も高い値が存在しない場合、MDE. は最小値を返します。

中央値および最頻値の計算

次のリクエストは、EMPLOYEE データソースを使用して、2 つの部門の従業員の現在給与額を表示し、部門別の平均値、中央値、最頻値を計算します。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL AS 'INDIVIDUAL,SALARIES'
AVE.CURR_SAL AS 'DEPARTMENT,AVERAGE'
```

```
MDN.CURR_SAL AS 'DEPARTMENT,MEDIAN'
MDE.CURR_SAL AS 'DEPARTMENT,MODE'
BY DEPARTMENT
ON TABLE SET PAGE NOPAGE
END
```

両方の部門で、従業員数は偶数です。MIS 部門では、中央に位置する 2 つの値が同一であるため、その値 ($18,480.00) が中央値および最頻値になります。PRODUCTION 部門では、中央に位置する 2 つの値 ($16,100.00 と $21,120.00) の平均が中央値になります。また、重複した値が存在しないため、最小値 ($9,500.00) が最頻値になります。

DEPARTMENT	INDIVIDUAL SALARIES	DEPARTMENT AVERAGE	DEPARTMENT MEDIAN	DEPARTMENT MODE
MIS	$108,002.00	$18,000.33	$18,480.00	$18,480.00
PRODUCTION	$114,282.00	$19,047.00	$18,610.00	$9,500.00

行と列のパーセント計算

PCT. 演算接頭語は、列の個々の値がその列の合計値の何パーセントを占めているかを計算します。BY 句を使用して列をソートすることにより、縦方向の値の並び順を制御することができます。

値のパーセントを行方向で計算することもできます。ACROSS 句を使用して行方向の値をソート後、RPCT. 演算接頭語でその行の合計値に対する個々の値のパーセントを計算します。

列のパーセント計算

次のリクエストを発行して、各従業員が教育に費やした時間の割合を計算します。

```
TABLE FILE EMPLOYEE
SUM ED_HRS PCT.ED_HRS BY LAST_NAME
ON TABLE COLUMN-TOTAL
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

下図は、出力結果を示しています。

LAST_NAME	ED_HRS	PCT ED_HRS
BANNING	.00	.00%
BLACKWOOD	75.00	21.37%
CROSS	45.00	12.82%
GREENSPAN	25.00	7.12%
IRVING	30.00	8.55%
JONES	50.00	14.25%
MCCOY	.00	.00%
MCKNIGHT	50.00	14.25%
ROMANS	5.00	1.42%
SMITH	46.00	13.11%
STEVENS	25.00	7.12%
TOTAL	351.00	100.00%

PCFORMAT を OLD に設定した場合、PCT. と RPCT. には、計算対象のフィールドと同一のフォーマットが適用されるため、浮動小数点数演算の性質上、列の合計値が 100 にならない場合があります。

行のパーセント計算

次のリクエストは、都市別に各製品の販売数 (UNIT_SOLD 列) を合計するとともに、各製品の総販売数に対する割合 (RPCT.UNIT_SOLD 列) も併せて計算します。

```
TABLE FILE SALES
SUM UNIT_SOLD RPCT.UNIT_SOLD ROW-TOTAL
BY PROD_CODE
ACROSS CITY    WHERE
CITY EQ 'NEW YORK' OR 'STAMFORD'
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。

	CITY NEW YORK		STAMFORD		TOTAL	
PROD_CODE	UNIT_SOLD	RPCT UNIT_SOLD	UNIT_SOLD	RPCT UNIT_SOLD	UNIT_SOLD	RPCT UNIT_SOLD
B10	30	33%	60	66%	90	99%
B12	.	.	40	100%	40	100%
B17	20	40%	29	59%	49	99%
B20	15	100%	.	.	15	100%
C13	.	.	25	100%	25	100%
C17	12	100%	.	.	12	100%
C7	.	.	45	100%	45	100%
D12	20	42%	27	57%	47	99%
E1	30	100%	.	.	30	100%
E2	.	.	80	100%	80	100%
E3	35	33%	70	66%	105	99%

ここでは、UNIT_SOLD は整数フォーマットのため、RPCT. で作成されたフィールドも整数 (I) フォーマットになります。そのため、行によっては個々のパーセント値が切り捨てられて、その合計が 100 パーセントより小さい値になるものもあります。正確な合計を求めるには、小数点の位置を指定するフォーマット (D、F) でフィールドを再定義する必要があります。

個数のパーセント計算

ファイル内のインスタンスの個数を計算する場合、一般にレポート作成で要求されることは、インスタンスの総数に対する各行の個数の相対的な割合を計算することです。この計算を行うには、次の構文を使用します。これは、フィールドに対してのみ使用できます。

```
PCT.CNT.fieldname
```

フォーマットは、7 桁の小数値で、小数点以下 2 桁およびパーセント記号を含みます。

個数のパーセント計算

次のリクエストは、EMP_ID フィールドの値の相対的な割合を部署別に計算します。

```
TABLE FILE EMPLOYEE
SUM PCT.CNT.EMP_ID
BY DEPARTMENT
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
GRID=OFF,$
```

```
ENDSTYLE
END
```

下図は、出力結果を示しています。

DEPARTMENT	PCT.CNT EMP_ID
MIS	50.00%
PRODUCTION	50.00%

ユニーク値の合計とリスト表示

除外演算子の接頭語 (DST.) を使用して、データソースフィールドのユニーク値の合計とリスト表示を行うことができます。この演算接頭語は、SQL COUNT、SUM、AVG (DISTINCT col) のフィールド機能と同様に、単一パスのデータソースにある種類の個数を集計します。

DST. 演算子は、SUM、PRINT、COUNT のコマンドや集計演算接頭語 SUM.、CNT.、AVE. と併用することができます。複数の DST. 演算子は、TABLE および TABLEF リクエストでサポートされます。これらは、BY、ACROSS、FOR 句を使用するリクエストでサポートされます。

PRINT コマンドおよび複数の DST 演算子を使用するリクエストでは、SET PRINTDST=NEW コマンドを発行する必要があります。詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

除外演算子の使用

```
command DST.fieldname
```

または

```
SUM [operator].DST.fieldname
```

説明

command

SUM、PRINT、COUNT のいずれかのコマンドです。

DST.

除外演算子を表します。

fieldname

表示フィールドオブジェクトまたはフィールド名を表します。

operator

SUM.、CNT.、AVE. のいずれかを表します。

除外演算子の使用

次のいずれかの方法を使用して、ED_HRS のユニーク値の個数を集計します。

```
TABLE FILE EMPLOYEE
SUM CNT.DST.ED_HRS
END
```

または

```
TABLE FILE EMPLOYEE
COUNT DST.ED_HRS
END
```

出力結果は次のとおりです。

```
COUNT
DISTINCT
ED_HRS
--------
       9
```

この結果には、姓が SMITH という 2 名の従業員名の重複値は個数に含まれていますが、教育に費やした時間値 (ED_HRS) の 50.00、25.00、0.0 に重複値があり、それぞれの値に対する 2 番目のレコードが除外されたため、ED_HRS のユニーク値の個数は全部で 9 件になっています。

複数表示コマンドでのフィールド値の計算

次のリクエストは、GGSALES データソースに対して実行され、地域別のレコード総数を計算するとともに、地域別および州別のレコード件数、製品カテゴリの種類数、製品の種類数を計算します。DST 演算子および CNT.DST 演算子は、最後の表示コマンドでのみ使用可能です。

```
TABLE FILE GGSALES
COUNT CATEGORY AS 'TOTAL,COUNT'
  BY REGION
SUM CNT.CATEGORY AS 'STATE,COUNT'
    CNT.DST.CATEGORY    CNT.DST.PRODUCT
  BY REGION
  BY ST
END
```

出力結果は次のとおりです。

```
                                COUNT     COUNT
             TOTAL         STATE  DISTINCT  DISTINCT
Region       COUNT  State  COUNT  CATEGORY  PRODUCT
------       -----  -----  -----  --------  --------
Midwest       1085  IL       362         3         9
                    MO       361         3         9
                    TX       362         3         9
Northeast     1084  CT       361         3        10
                    MA       360         3        10
                    NY       363         3        10
Southeast     1082  FL       361         3        10
                    GA       361         3        10
                    TN       360         3        10
West          1080  CA       721         3        10
                    WA       359         3        10
```

除外演算子の制限事項

- COUNT DST. または CNT.DST 演算子を使用して作成されたフィールドのフォーマットを変更する場合は、そのフィールドのフォーマットを整数 (I) データタイプに変更する必要があります。別のデータタイプを指定した場合、次のエラーが発生します。

```
COUNT, CNT. のフォーマット指定に誤りがあります。
```

- CNT.、SUM.、AVE. の演算接頭語を他の表示コマンドと併用すると、次のエラーが発生します。

```
(FOC1853) CNT/SUM/AVT.DST は集計動詞にのみ使用可能です。
```

- DST. 演算子を MATCH コマンド内で使用すると、次のエラーが発生します。

```
00100(FOC1854) DST 演算子は、TABLE でのみサポートされています。
```

- BY フィールドのフォーマットを変更すると、次のエラーが発生します (PRINT コマンドと併用すると、DST. fieldname は BY フィールドになります)。

```
(FOC1862) フォーマット変換した DST.フィールドは PRINT に使えません。
```

- DST. 演算子を NOSPLIT とともに使用すると、次のエラーが発生します。

```
(FOC1864) DST 演算子は、ACROSS/FOR/NOSPLIT と共に使用できません。
```

- 複数動詞リクエストの SUM DST. fieldname BY field PRINT fld BY fld を使用すると、次のエラーが発生します。SUM コマンドと併用する動詞オブジェクト演算子は、集計の最下位に配置する必要があります。

```
(FOC1867) DST 演算子は、最下位の集計動詞に指定してください。
```

- DST. 演算子は、HEADING または FOOTING の一部として使用することはできません。
- DST.、AVE.DST.、CNT.DST.、SUM.DST. 演算子は、集計表示コマンド (例、SUM) を使用するリクエストで WITHIN とともに使用することはできません。DST. 演算子は、PRINT 表示コマンドを使用するリクエストで WITHIN とともに使用することはできません。

先頭レコードと最終レコードの取得

FST. 演算接頭語は、選択したフィールドから最初に取得したレコードを表示します。LST. 演算接頭語は、選択したフィールドから最後に取得したレコードを表示します。

FST. および LST. 演算接頭語を使用する場合は、対象となるデータソースの構造を理解しておくことが重要です。

- セグメント内のレコードが昇順に整列されている場合 (セグメントタイプ S1)、FST. 演算接頭語で取得される先頭の論理レコードは、すべての値の中で最下位の値になります。逆に、LST. 演算接頭語では、すべての値の中で最上位の値が取得されます。
- セグメント内のレコードが降順に整列されている場合 (セグメントタイプ SH1)、FST. 演算接頭語で取得される先頭の論理レコードは、すべての値の中で最上位の値になります。逆に、LST. 演算接頭語では、すべての値の中で最下位の値が取得されます。

データソースのデータを再整列する方法や、レポート実行時にデータソースを再構築する方法についての詳細は、パフォーマンスの改善を参照してください。

先頭レコードの取得

次のリクエストは、EMP_ID フィールドの先頭の論理レコードを取得します。

```
TABLE FILE EMPLOYEE
SUM FST.EMP_ID
END
```

出力結果は次のとおりです。

```
FST
EMP_ID
------
071382660
```

セグメントタイプとレコードの取得

EMPLOYEE データソースには、DED_CODE フィールドと DED_AMT フィールドのそれぞれの値を昇順に整列する DEDUCT セグメント (セグメントタイプ S1) が保存されています。DED_CODE フィールドは、給与の控除タイプを表しています。そのタイプには、CITY、STATE、FED、FICA などがあります。次のリクエストは、各従業員に対する DED_CODE の先頭の論理レコードを取得します。

```
TABLE FILE EMPLOYEE
SUM FST.DED_CODE
```

```
BY EMP_ID
END
```

出力結果は次のとおりです。

```
           FST
EMP_ID     DED_CODE
------     --------
071382660  CITY
112847612  CITY
117593129  CITY
119265415  CITY
119329144  CITY
123764317  CITY
126724188  CITY
219984371  CITY
326179357  CITY
451123478  CITY
543729165  CITY
818692173  CITY
```

ここで SUM LST.DED_CODE コマンドを使用すると、各従業員に対する DED_CODE の最終の論理レコードが取得できます。

セグメント内のレコードが降順に整列されている場合 (セグメントタイプ SH1)、FST. 演算接頭語で取得される先頭の論理レコードは、すべての値の中で最上位の値になります。逆に、LST. 演算接頭語では、すべての値の中で最下位の値が取得されます。

たとえば、EMPLOYEE データソースには、JOBCODE、SALARY、PCT_INC、DAT_INC フィールドを降順に整列した PAYINFO セグメント (セグメントタイプ SH1) が保存されています。次のリクエストは、各従業員に対する SALARY の先頭の論理レコードを取得します。

```
TABLEF FILE EMPLOYEE
SUM FST.SALARY
BY EMP_ID
END
```

出力結果は次のとおりです。

```
                    FST
EMP_ID              SALARY
------              ------
071382660       $11,000.00
112847612       $13,200.00
```

```
117593129       $18,480.00
119265415        $9,500.00
119329144       $29,700.00
123764317       $26,862.00
126724188       $21,120.00
219984371       $18,480.00
326179357       $21,780.00
451123478       $16,100.00
543729165        $9,000.00
818692173       $27,062.00
```

ここで SUM LST.SALARY コマンドを使用すると、各従業員に対する SALARY の最終の論理レコードが取得できます。

値の合計と個数の集計

CNT.、SUM.、TOT. の演算接頭語を使用すると、1 つの表示コマンドで値の合計とその個数の集計を同時に実行することができます。CNT. 演算接頭語は、COUNT コマンドと同様に、この接頭語が前に付いたフィールドの個数を集計します。SUM. 演算接頭語は、SUM コマンドと同様、この接頭語が前に付いたフィールドの値を合計します。TOT. は見出し (脚注、中間見出し、中間脚注を含む) に使用すると、前置するフィールドの合計を計算することができます。

CNT による値の集計

次のリクエストでは、PRODUCT_ID の個数を集計し、UNIT_PRICE の値を合計します。

```
TABLE FILE GGPRODS
SUM CNT.PRODUCT_ID AND UNIT_PRICE
END
```

出力結果は次のとおりです。

```
Product
Code         Unit
COUNT        Price
-------      -----
     10     660.00
```

SUM 演算接頭語による値の集計

次のリクエストでは、PRODUCT_ID の個数を集計し、UNIT_PRICE の値を合計します。

```
TABLE FILE GGPRODS
COUNT PRODUCT_ID AND SUM.UNIT_PRICE
END
```

出力結果は次のとおりです。

```
Product
Code        Unit
COUNT       Price
-------     -----
     10    660.00
```

TOT による値の合計

次のリクエストは、TOT 演算接頭語を使用して、全従業員の現在給与を合計します。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
BY DEPARTMENT
ON TABLE SUBFOOT
"Total salaries equal: <TOT.CURR_SAL"
END
```

出力結果は次のとおりです。

```
DEPARTMENT  LAST_NAME
----------  ---------
MIS         SMITH
            JONES
            MCCOY
            BLACKWOOD
            GREENSPAN
            CROSS
PRODUCTION  STEVENS
            SMITH
            BANNING
            IRVING
```

```
             ROMANS
             MCKNIGHT
Total salaries equal:     $222,284.00
```

RNK を使用したソートフィールド値の順位付け

RANKED BY *fieldname* を TABLE リクエストのソート句で使用すると、指定したフィールドによるソートだけではなく、そのインスタンスへの RANK 値の割り当ても行われます。RNK. 演算接頭語は、ページの任意の位置への RANK 値の表示を可能にし、順位も計算します。この演算子は、RNK.*fieldname* と指定します。*fieldname* はリクエストの BY フィールドです。

順位付け処理は、レコードの選択およびソートの後に実行されます。したがって、RNK. 演算接頭語を WHERE または IF 選択テスト、および一時項目 (DEFINE) で使用することはできません。ただし、RNK.*fieldname* は、WHERE TOTAL、IF TOTAL テスト、または一時項目 (COMPUTE) では使用することができます。AS 句を使用し、RANK フィールド用のデフォルトフィールドタイトルを変更することができます。

RNK. 演算接頭語は複数のソートフィールドに適用することができます。その場合、各 BY フィールドの順位は上位の BY フィールド内で計算されます。

RNK. を使用した順位付け演算接頭語

PRINT コマンド、COMPUTE 式、または IF/WHERE TOTAL 式で、次のように指定します。

```
RNK.field  ...
```

説明

field

リクエストの BY フィールドです。

ソートグループ内の順位付け

次のリクエストは、部門 (DEPARTMENT) 内の勤続年数 (YRS_SERVICE) の順位を計算し、さらに勤続年数 (YRS_SERVICE) と部門 (DEPARTMENT) 内の給料額 (SALARY) の順位を計算します。勤続年数は、TODAY の値に依存します。この例の出力結果は、2006 年 9 月に実行した場合に有効になります。

```
DEFINE FILE EMPDATA
  TODAY/YYMD = &YYMD;
  YRS_SERVICE/I9 = DATEDIF(HIREDATE,TODAY,'Y');
END
TABLE FILE EMPDATA
PRINT SALARY
  RNK.YRS_SERVICE AS 'RANKING,BY,SERVICE'
  RNK.SALARY AS 'SALARY,RANK'
     BY DEPT
     BY HIGHEST YRS_SERVICE
     BY HIGHEST SALARY NOPRINT
WHERE DEPT EQ 'MARKETING' OR 'SALES'
ON TABLE SET PAGE NOPAGE
END
```

出力結果は次のとおりです。

```
                                                RANKING
                                                BY       SALARY
DEPT                 YRS_SERVICE           SALARY  SERVICE  RANK
----                 -----------           ------  -------  ------
MARKETING                     17       $55,500.00        1       1
                                       $55,500.00        1       1
                              16       $62,500.00        2       1
                                       $62,500.00        2       1
                                       $62,500.00        2       1
                                       $58,800.00        2       2
                                       $52,000.00        2       3
                                       $35,200.00        2       4
                                       $32,300.00        2       5
                              15       $50,500.00        3       1
                                       $43,400.00        3       2
SALES                         17      $115,000.00        1       1
                                       $54,100.00        1       2
                              16       $70,000.00        2       1
                                       $43,000.00        2       2
```

```
                              15       $43,600.00        3       1
                                       $39,000.00        3       2
                              15       $30,500.00        3       3
```

WHERE TOTAL での RNK. の使用

次のリクエストは、勤続年数 (YRS_SERVICE) カテゴリ内の上位 2 番目までの給与 (SALARY) の行のみを表示します。勤続年数は、TODAY の値に依存します。この例の出力結果は、2006 年 9 月に実行した場合に有効になります。

```
DEFINE FILE EMPDATA
  TODAY/YYMD = &YYMD;
  YRS_SERVICE/I9 = DATEDIF(HIREDATE,TODAY,'Y');
END
TABLE FILE EMPDATA
PRINT LASTNAME FIRSTNAME RNK.SALARY
BY HIGHEST YRS_SERVICE BY HIGHEST SALARY
WHERE TOTAL RNK.SALARY LE 2
END
```

出力結果は次のとおりです。

```
                                                              RANK
YRS_SERVICE           SALARY  LASTNAME         FIRSTNAME     SALARY
-----------           ------  --------         ---------     ------
         17      $115,000.00  LASTRA           KAREN              1
                  $80,500.00  NOZAWA           JIM                2
         16       $83,000.00  SANCHEZ          EVELYN             1
                  $70,000.00  CASSANOVA        LOIS               2
         15       $62,500.00  HIRSCHMAN        ROSE               1
                              WANG             JOHN               1
                  $50,500.00  LEWIS            CASSANDRA          2
```

COMPUTE での RNK. の使用

次のリクエストは、部門内の給料順位が 5 番目までのレコード、および勤続年数と勤続年数順位が 6 番目までのレコードのフラグを「Y」に設定します。それ以外のレコードのフラグの値は、「N」に設定されます。勤続年数は、TODAY の値に依存します。この例の出力結果は、2006 年 9 月に実行した場合に有効になります。

```
DEFINE FILE EMPDATA
  TODAY/YYMD = &YYMD;
  YRS_SERVICE/I9 = DATEDIF(HIREDATE,TODAY,'Y');
END
TABLE FILE EMPDATA
PRINT RNK.SALARY RNK.YRS_SERVICE
COMPUTE FLAG/A1 = IF RNK.SALARY LE 5  AND RNK.YRS_SERVICE LE 6
    THEN 'Y' ELSE 'N';
BY DEPT BY SALARY BY YRS_SERVICE
WHERE DEPT EQ 'MARKETING' OR 'SALES'
ON TABLE SET PAGE NOPAGE
END
```

出力結果は次のとおりです。

```
                                          RANK   RANK
DEPT                        SALARY YRS_SERVICE SALARY YRS_SERVICE FLAG
----                        ------ ----------- ------ ----------- ----
MARKETING                $32,300.00          16      1           1 Y
                         $35,200.00          16      2           1 Y
                         $43,400.00          15      3           1 Y
                         $50,500.00          15      4           1 Y
                         $52,000.00          16      5           1 Y
                         $55,500.00          17      6           1 N
                                                     6           1 N
                         $58,800.00          16      7           1 N
                         $62,500.00          16      8           1 N
                                                     8           1 N
                                                     8           1 N
SALES                    $30,500.00          15      1           1 Y
                         $39,000.00          15      2           1 Y
                         $43,000.00          16      3           1 Y
                         $43,600.00          15      4           1 Y
                         $54,100.00          17      5           1 Y
                         $70,000.00          16      6           1 N
                        $115,000.00          17      7           1 N
```

集計行での集約計算

SUMMARIZE および RECOMPUTE を使用してソートフィールド区切りごとに値を再計算することはできますが、この方法では集計行の値の計算に詳細データが使用されます。

ROLL. 演算子とともに別の演算接頭語を集計行で使用すると、ソート区切りごとの値の再計算には、下位ソート区切りで生成された集計行の値が使用されます。

サポートされる演算子の組み合わせは次のとおりです。

- ROLL.SUM. - ROLL. と同義です。SUM. を使用する場合は、文字フィールドがサポートされます。その場合、SUMPREFIX パラメータに応じて、最初の値、最小値、最大値、最後の値のいずれかが返されます。
- ROLL.AVE.
- ROLL.MAX. - 文字フィールドと数値フィールドの両方でサポートされます。
- ROLL.MIN. - 文字フィールドと数値フィールドの両方でサポートされます。
- ROLL.FST. - 文字フィールドと数値フィールドの両方でサポートされます。
- ROLL.LST. - 文字フィールドと数値フィールドの両方でサポートされます。
- ROLL.CNT.
- ROLL.ASQ.

ROLL.**prefix** を集計行に使用すると、この接頭語の演算は、下位ソートフィールドの集計コマンドから得られた集計値に対して実行されます。

ROLL. 演算子に別の演算接頭語を使用しない場合は、SUM. として処理されます。そのため、最下位ソートフィールドの集計コマンドで AVE. を指定し、その上位ソートフィールドで ROLL. を指定した場合、結果は平均値の集計になります。平均値の平均を計算するには、上位ソートフィールドで ROLL.AVE を使用します。

注意：SUMMARIZE と SUB-TOTAL を使用した場合、同一の計算が上位のすべてのソート区切りに継承されます。

集計値の集約

```
BY field {SUMMARIZE|SUBTOTAL|SUB-TOTAL|RECOMPUTE} [ROLL.][prefix1.]
[field1field2 ...|*] [ROLL.][prefix2.] [fieldn ...]
```

または

```
BY field
```

```
ON field {SUMMARIZE|SUBTOTAL|SUB-TOTAL|RECOMPUTE} ROLL.[prefix.]
[field1field2 ...|*]
```

説明

ROLL.

このソートフィールドの集計値を、下位ソートフィールドの集計コマンドから得られた集計値を使用して計算します。

field

リクエストで指定するソートフィールドです。

prefix1, prefix2

集計値の計算に使用する演算接頭語です。演算接頭語のいずれかです。演算接頭語には、SUM. (何も指定しない場合のデフォルト演算子)、AVE.、MAX.、MIN.、FST.、LST.、CNT.、ASQ. があります。

field1field2fieldn

集計するフィールドです。

*****

このアスタリスク (*) は、すべてのフィールド、つまり数値フィールドおよび文字フィールドを集計行に含める必要があることを示しています。アスタリスク (*) を使用してすべてのフィールドを表示することも、表示する特定のフィールドを参照することもできます。

平均の集約

次のリクエストでは GGSALES データソースが使用され、REGION および ST の 2 つのソートフィールドが指定されています。REGION の集計コマンドにより、AVE. 演算子が各州の UNITS 値の合計に適用されます。

```
TABLE FILE GGSALES
   SUM UNITS AS 'Inventory '
     BY REGION
   BY ST
   ON REGION SUBTOTAL      AVE.AS 'Average'
   WHERE DATE GE 19971001
   WHERE REGION EQ 'West' OR 'Northeast'
   ON TABLE SET PAGE NOPAGE
   END
```

出力結果では、各州の UNITS 値から各地域の平均値が計算され、地域ごとの中間合計として表示されています。また、総合計行の平均値の計算にも、各州の UNITS 値が使用されています。

```
Region       State  Inventory
------       -----  ----------
Northeast    CT          37234
             MA          35720
             NY          36248

Average Northeast
                         36400

West         CA          75553
             WA          40969

Average West
                         58261


TOTAL                    45144
```

次のリクエストでは、総合計行の集計コマンドに ROLL. 演算子が指定されています。

```
TABLE FILE GGSALES
   SUM UNITS AS 'Inventory '
     BY REGION
   BY ST
   ON REGION SUBTOTAL  AVE.AS 'Average'
   WHERE DATE GE 19971001
   WHERE REGION EQ 'West' OR 'Northeast'
   ON TABLE SUBTOTAL ROLL.AVE.AS ROLL.AVE
   ON TABLE SET PAGE NOPAGE
   END
```

出力結果では、各州の UNITS 値から各地域の平均値が計算され、地域ごとの中間合計として表示されていますが、総合計行の平均値の計算には、地域ごとの中間合計値が使用されています。

```
Region       State  Inventory
  ------       -----  ----------
  Northeast    CT          37234
               MA          35720
               NY          36248
```

```
  Average Northeast
                           36400

  West          CA         75553
                WA         40969

  Average West
                           58261


  ROLL.AVE                 47330
```

最上位ソート区切りへの集約の継承

次のリクエストでは GGSALES データソースが使用され、3 つのソートフィールドが指定されています。PRODUCT ソートフィールドの SUBTOTAL コマンドで AVE. が指定され、上位の REGION ソートフィールドの SUMMARAIZE コマンドで ROLL.AVE. が指定されています。

```
TABLE FILE GGSALES
SUM UNITS
BY REGION
BY PRODUCT
BY HIGHEST DATE
WHERE DATE GE 19971001
  WHERE REGION EQ 'Midwest' OR 'Northeast'
  WHERE PRODUCT LIKE 'C%'
  ON PRODUCT SUBTOTAL AVE.
  ON REGION SUMMARIZE ROLL.AVE.AS ROLL.AVE
ON TABLE SET PAGE NOPAGE
END
```

出力結果では、各製品の平均値の計算には、各日付の詳細行が使用されています。REGION レベルで ROLL.AVE. が指定されているため、各地域の平均値の計算には、各製品の平均値が使用されています。また、総合計行の平均値の計算は、各地域の平均値が使用されています。

```
Region       Product                 Date  Unit Sales
  ------       -------                   ----  ----------
  Midwest      Coffee Grinder    1997/12/01        4648
                                 1997/11/01        3144
                                 1997/10/01        1597
```

```
*TOTAL PRODUCT Coffee Grinder                    3129

             Coffee Pot        1997/12/01        1769
                               1997/11/01        1462
                               1997/10/01        2346

*TOTAL PRODUCT Coffee Pot                        1859

             Croissant         1997/12/01        7436
                               1997/11/01        5528
                               1997/10/01        6060

*TOTAL PRODUCT Croissant                         6341
ROLL.AVE Midwest                                 3776

Northeast    Capuccino         1997/12/01        1188
                               1997/11/01        2282
                               1997/10/01        3675

*TOTAL PRODUCT Capuccino                         2381

             Coffee Grinder    1997/12/01        1536
                               1997/11/01        1399
                               1997/10/01        1315

*TOTAL PRODUCT Coffee Grinder                    1416

             Coffee Pot        1997/12/01        1442
                               1997/11/01        2129
                               1997/10/01        2082

*TOTAL PRODUCT Coffee Pot                        1884

             Croissant         1997/12/01        4291
                               1997/11/01        6978
                               1997/10/01        4741

*TOTAL PRODUCT Croissant                         5336
ROLL.AVE Northeast                               2754


TOTAL                                            3265
```

ROLL 使用上の注意

- ROLL.**prefix** を集計行に使用すると、この接頭語の演算は、下位ソートフィールドの集計コマンドから得られた集計値に対して実行されます。
- ROLL. が指定されたソートフィールドの下位で集計コマンドが発行されておらず、ROLL. とともに別の演算接頭語が使用されていない場合は、SUM. が計算されます。下位で集計コマンドが発行されておらず、ROLL. とともに別の演算接頭語が使用されている場合は (例、ROLL.AVE.)、指定された演算接頭語が使用されます。たとえば、ROLL.AVE. は、AVE. になります。
- CNT.**prefix** を使用すると、表示されているデータ行の行数が計算されます。この計算は、MULTILINES の影響を受けません。
- ROLL.CNT.**prefix** を使用すると、表示されている集計行の行数が計算されます。この計算は、MULTILINES の影響を受けません。

母集団または標本の標準偏差の計算

標準偏差の演算接頭語は、データの散らばりの度合いを表す数値を返します。データセットは、母集団全体として指定することも標本として指定することもできます。標準偏差は、分散の平方根です。分散は、観測値の予測値 (平均値) からの乖離を表します。母集団を指定した場合、標準偏差の計算の除数 (自由度とも呼ばれる) は、データポイントの総数 N になります。標本を指定した場合、除数は N-1 になります。

xi が観測値、N が観測値の数、μ がすべての観測値の平均である場合、母集団の標準偏差を計算する式は次のとおりです。

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_1-\mu)^2}$$

標本の標準偏差を計算する場合、平均は標本観測値を使用して計算し、除数は N ではなく N-1 となります。

母集団の標準偏差を計算する場合、次の構文を使用します。

```
STDP.field
```

標本の標準偏差を計算する場合、次の構文を使用します。

```
STDS.field
```

説明

field

数値

標準偏差の計算に使用する一連の観測値です。

母集団の標準偏差の計算

次のリクエストは、DOLLARS フィールドの母集団の標準偏差を計算し、倍精度小数点数に変換します。

```
DEFINE FILE ibisamp/ggsales
DOLLARS/D12.2 = DOLLARS;
END
TABLE FILE ibisamp/ggsales
SUM DOLLARS STDP.DOLLARS
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。

DOLLARS	STDP DOLLARS
46,156,290.00	6,156.997845651

レポートレベルの演算接頭語の使用

レポートレベルの演算接頭語は、見出し、脚注、中間見出し、中間脚注、動詞オブジェクト、一時項目 (COMPUTE) に使用でき、レポート全体の平均、最大値、最小値、件数を計算します。見出しに含める合計値を計算する TOT. 演算子をベースにした演算接頭語です。

これらの演算子は、WHERE および WHERE TOTAL テストで参照することはできません。ただし、これらの演算子を COMPUTE コマンドで使用し、WHERE TOTAL テストで使用可能な計算値を生成することは可能です。

レポートレベルの平均、最大値、最小値、件数の計算

```
operator.field
```

説明

operator

次の演算接頭語のいずれかを指定できます。

- **TOTAVE** テーブル全体でのフィールドの平均値を計算します。
- **TOTMAX** テーブル全体でのフィールドの最大値を計算します。
- **TOTMIN** テーブル全体でのフィールドの最小値を計算します。
- **TOTCNT** テーブル全体でのフィールドのインスタンス数を計算します。

field

リクエスト内の動詞オブジェクトまたは一時項目 (COMPUTE) です。

見出しでの演算接頭語の使用

次のリクエストでは、見出しに演算接頭語を使用します。

```
TABLE WF_RETAIL_LITE
HEADING
"Heading Calculations:"
"Total:      <TOT.COGS_US"
"Count:                  <TOTCNT.COGS_US"
"Average:           <TOTAVE.COGS_US"
"Minimum:            <TOTMIN.COGS_US"
"Maximum:      <TOTMAX.COGS_US"
```

```
SUM COGS_US CNT.COGS_US AS Count AVE.COGS_US AS Average
MIN.COGS_US AS Minimum MAX.COGS_US AS Maximum
BY BUSINESS_REGION AS Region
BY PRODUCT_CATEGORY AS Category
WHERE BUSINESS_REGION NE 'Oceania'
ON TABLE SUBTOTAL COGS_US CNT.COGS_US  AS Total
ON TABLE SET PAGE NOPAGE
ON TABLE SET SHOWBLANKS ON
ON TABLE SET STYLE *
type=report,grid=off, size=11,$
ENDSTYLE
END
```

出力結果は次のとおりです。

Heading Calculations:
Total: $2,940,745.00
Count: 9961
Average: $295.23
Minimum: $16.00
Maximum: $9,750.00

Region	Category	Cost of Goods	Count	Average	Minimum	Maximum
EMEA	Accessories	$143,987.00	630	$228.55	$16.00	$2,850.00
	Camcorder	$187,000.00	611	$306.06	$60.00	$8,610.00
	Computers	$47,616.00	266	$179.01	$81.00	$668.00
	Media Player	$328,401.00	956	$343.52	$36.00	$1,640.00
	Stereo Systems	$361,756.00	1412	$256.20	$48.00	$2,295.00
	Televisions	$100,443.00	120	$837.02	$275.00	$6,500.00
	Video Production	$78,722.00	276	$285.22	$78.00	$1,880.00
North America	Accessories	$171,306.00	743	$230.56	$16.00	$2,040.00
	Camcorder	$242,967.00	692	$351.11	$60.00	$6,000.00
	Computers	$53,329.00	284	$187.78	$81.00	$668.00
	Media Player	$386,681.00	1118	$345.87	$36.00	$1,600.00
	Stereo Systems	$412,036.00	1636	$251.86	$48.00	$2,700.00
	Televisions	$101,212.00	121	$836.46	$275.00	$6,500.00
	Video Production	$89,489.00	325	$275.35	$78.00	$1,410.00
South America	Accessories	$25,723.00	103	$249.74	$16.00	$1,470.00
	Camcorder	$22,686.00	110	$206.24	$60.00	$1,000.00
	Computers	$7,845.00	47	$166.91	$81.00	$501.00
	Media Player	$60,183.00	161	$373.81	$36.00	$1,480.00
	Stereo Systems	$81,823.00	284	$288.11	$48.00	$2,700.00
	Televisions	$25,800.00	25	$1,032.00	$275.00	$9,750.00
	Video Production	$11,740.00	41	$286.34	$78.00	$1,410.00
Total		$2,940,745.00	9961			

レポートレベル演算接頭語の使用上の注意

- リクエストの表示コマンドでフィールドが参照されていない場合でも、これらの演算接頭語を見出しまたは脚注内のフィールドに対して使用することができます。
- これらの演算接頭語は、見出しまたは脚注内の実フィールドまたは一時項目 (DEFINE) に対して機能します。また、表示コマンドフィールドリスト内で、実フィールド、一時項目 (DEFINE)、およびリクエストで使用される前に計算される一時項目 (COMPUTE) に対しても機能します。

- これらの演算接頭語を中間見出しおよび中間脚注内で使用して、レポート全体の合計値を参照することもできます。

フィールドタイトルのポップアップ説明表示

HTML レポートのフィールドタイトルにマウスポインタを置いたときに、フィールド説明をポップアップ表示させることができます。フィールド説明のテキストはフィールドタイトル付近にポップアップ表示され、このテキストにはレポートのデフォルトフォントが使用されます。レポートのフィールドタイトルとして表示されるポップアップテキストには、ACROSS 句で作成したタイトルや、OVER 句で作成したスタックフィールドなどがあります。

フィールドタイトルで表示するポップアップテキストを定義するには、そのフィールドに対してマスターファイルで DESCRIPTION 属性を指定します。フィールドタイトルに対応するマスターファイルの説明がブランクの場合、フィールドタイトルにマウスポインタを置いても、ポップアップは表示されません。

POPUPDESC コマンドの使用

```
SET POPUPDESC = {ON|OFF}
```

説明

ON

マウスポインタをフィールドタイトル上に置いたときに表示するポップアップのフィールド説明を有効にします。

OFF

マウスポインタをフィールドタイトル上に置いたときに表示するポップアップのフィールド説明を無効にします。デフォルト値は OFF です。

POPUPDESC コマンドの使用

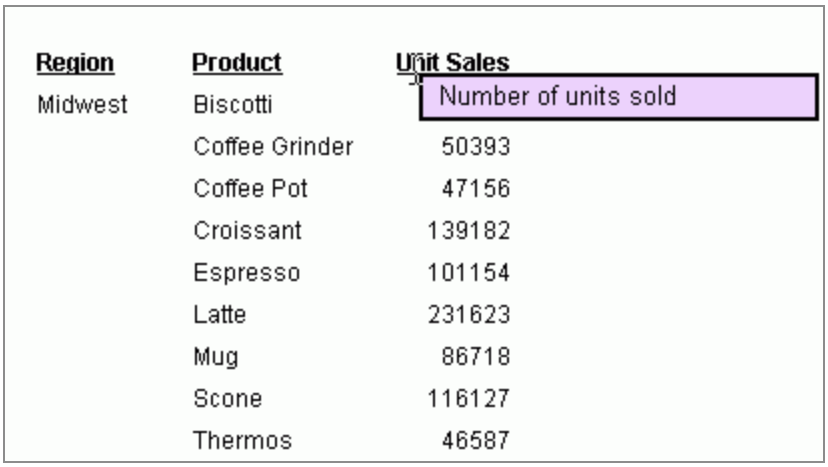
レポートで参照しているマスターファイルには次の記述が含まれます。

```
FIELD=UNITS, ALIAS=E10, FORMAT=I08, TITLE='Unit Sales',
 DESC='Number of units sold',$
```

レポートの作成に使用するコードは次のとおりです。

```
TABLE FILE GGSALES
SUM UNITS
BY REGION
BY PRODUCT
WHERE REGION EQ 'Midwest'
ON TABLE SET POPUPDESC ON
END
```

下図は、ポップアップフィールド説明テキストが表示されたレポート出力を示しています。フィールド説明は、マウスポインタをフィールドタイトル上に置いたときに表示されます。



ポップアップフィールド説明付きレポートの ReportCaster による配信

ReportCaster を使用してポップアップフィールド説明付きの HTML レポートを配信するには、WebFOCUS Client 内にある JavaScript コンポーネントを使用する必要があります。

ReportCaster が配信するレポートからこれらのコンポーネントにアクセスするには、スケジュール済みレポートに SET FOCHTMLURL を追加し、設定をデフォルトの /ibi_html から絶対 URL に変更する必要があります。以下はその例です。

```
SET FOCHTMLURL = http://hostname[:port]/ibi_apps/ibi_html
```

説明

hostname[:port]

WebFOCUS Web アプリケーションの展開先のホスト名とオプションのポート番号です。ポート番号はデフォルト値以外を使用している場合に入力します。

ibi_apps/ibi_html

ibi_apps は、サイトごとに異なる WEBFOCSxx/ibi_apps ディレクトリの Web サーバエイリアスです (ibi_apps はデフォルト値)。ibi_html は、アクセスする必要のある JavaScript ファイルへのパス上のディレクトリです。

レポートを ReportCaster で使用するためのコーディングについての詳細は、『ibi™ WebFOCUS® ReportCaster 利用ガイド』の「ReportCaster レポートコーディング上のヒントとテクニック」を参照してください。

表形式レポートのソート

レポートのソート機能を使用すると、データを縦または横、つまり行または列でソートして分類、整列したり、レポートのデータ項目の順序を任意に指定することができます。

ソートの基準となるソートフィールドには、データソース内の任意のフィールドを指定することができます。必要に応じて、複数のソートフィールドを選択し、一方のフィールドを他方にネストさせることも可能です。ソートフィールドは、そのフィールドの値が変更された場合にのみ表示されます。

表形式レポートソートの概要

レポートをソートするには、BY 句および ACROSS 句を使用します。

- BY 句を使用すると、行が作成されてソートフィールドの値が縦に表示されます。縦のソートフィールドは、レポートの最も左の列に表示されます。
- ACROSS 句を使用すると、列が追加されてソートフィールドの値が横に表示されます。横のソートフィールドは、レポートの最上部に表示されます。
- 1 つのレポートに BY 句と ACROSS 句の両方を使用すると、行と列が作成されて格子状のマトリックスが形成されます。

リクエストには、BY 句と ACROSS 句を任意に組み合わせて、最大で 128 個のソート句を含めることができます。

その他のソートオプションには次のものがあります。

- 値の小さい方から大きい方へ、または値の大きい方から小さい方へとソートしたり、ユーザ独自のソート順を定義したりすることができます。
- 大文字と小文字を区別する場合と区別しない場合のいずれかの照合順序に基づいてソートすることができます。
- ソートフィールドの値をレポートに表示させないようにすることができます。
- 数値データを 100 分位数や 10 分位数などでグループ化することができます。
- 数値フィールドの集計とソートを同時に実行することができます。

- 数値データを特定の範囲でグループ化することができます。
- データに順位を付け、その順位に基づいてデータを選択することができます。

データのソートと表示

使用する表示コマンドのタイプにより、データを 2 通りの方法でソートすることができます。

- PRINT または LIST コマンドを使用して、フィールドの個々の値をソートして表示することができます。
- 情報をグループ化して集計することができます。たとえば、COUNT コマンドを使用してソート値ごとのフィールド値の個数を計算したり、SUM コマンドを使用してフィールド値の集計を計算したりできます。

PRINT および LIST 表示コマンドを使用する場合、ソート値 1 つにつき複数の行がレポートに出力されることがあります。厳密には、表示フィールド 1 つにつき 1 行が作成されます。SUM および COUNT コマンドを使用する場合は、ソート値のユニークセット 1 つにつき 1 行がレポートに出力されます。詳細は、複数表示コマンドによるソートを参照してください。

表示コマンドについての詳細は、レポートデータの表示を参照してください。

行のソート

レポートのデータを縦にソートするには、BY 句を使用します。これにより、レポートに行が作成されます。1 つのリクエストに最大で 128 個のソート句 (BY 句と ACROSS 句の合計) を含めることができます (PRINT または LIST 表示コマンドを使用する場合は最大で 127 個)。

ソートフィールドは、そのフィールドの値が変更されたときに表示されます。ただし、BYDISPLAY パラメータを使用すると、すべてのソート値を表示することができます。詳細は、ソートフィールド値の表示制御を参照してください。

行のソート

```
BY {HIGHEST|LOWEST} [n] sortfield [AS 'text']
```

説明

HIGHEST

降順でソートします。

LOWEST

昇順でソートします。デフォルト値は LOWEST です。

n

レポート含めるフィールド値の個数を指定します。

sortfield

ソートフィールドの名前です。

text

レポート出力のソートフィールド列に使用する列見出しです。

行をソートする際の注意

- LIST 表示コマンドに BY 句を使用する場合、主ソート値が変更されるたびに LIST カウンタがリセットされて 1 に戻ります。
- デフォルトのソートは昇順ですが、オペレーティングシステムの種類によりその順序の定義が多少異なります。UNIX および Windows では、文字フィールドは 0 から 9、A から Z、a から z の順に、数値フィールドは 0 から 9 の順になります。ソート順は他の方法でも指定することができます。詳細は、ソート順の指定を参照してください。
- テキストフィールドは、ソートフィールドとして使用することはできません。テキストフィールドは、マスターファイルで FORMAT の値に TX が定義されたフィールドです。
- DEFINE コマンドまたはマスターファイルの DEFINE 属性で作成された一時項目は、ソートフィールドとして使用することができます。COMPUTE コマンドで作成された一時項目をソートフィールドとして使用するには、BY 句の代わりに BY TOTAL 句を使用する必要があります。
- マルチパスデータソースから複数のソートフィールドを指定してレポートを作成する場合は、すべてのソートフィールドが同一パスに存在していなければなりません。
- ソート句にフィールドのフォーマット情報を含めることはできません。

- ソートフィールドの値は、レポートに 1 回だけ表示されます。たとえば、MIS 部の従業員数が 6 名の場合は、リクエストで次のように宣言します。

  ```
  PRINT LAST_NAME BY DEPARTMENT
  ```

  このリクエストでは、「MIS」という部署名が 1 回だけ表示され、その後に 6 名の従業員の名前が表示されます。縦のソートフィールドの値が同一の場合であってもすべての値を繰り返し表示させるには、SET BYDISPLAY パラメータを使用します。詳細は、ソートフィールド値の表示制御を参照してください。

BY 句による行のソート

次の例は、全従業員の ID を部署別にソートして表示する方法を示しています。

```
TABLE FILE EMPLOYEE
PRINT EMP_ID
BY DEPARTMENT
END
```

この出力結果では、各部署別の EMP_ID の値が一覧表示されています。

DEPARTMENT	EMP_ID
MIS	112847612
	117593129
	219984371
	326179357
	543729165
	818692173
PRODUCTION	071382660
	119265415
	119329144
	123764317
	126724188
	451123478

複数 BY フィールドの使用

複数のソートフィールドを使用して、レポートのデータを整列することができます。複数のソートフィールドを指定する場合、BY 句を定義した順序によりソート順が決定されます。最

初の BY 句により先頭のソート区切りが設定され、2 つ目の BY 句によりその次のソート区切りが設定されます (以降同様)。後に続くソートは、直前のソートにネストされます。

複数 BY フィールドによるソート

次のリクエストは、複数の BY フィールドを使用します。

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL
BY DEPARTMENT BY LAST_NAME
WHERE CURR_SAL GT 21500
END
```

出力結果は次のとおりです。

```
DEPARTMENT  LAST_NAME              CURR_SAL
----------  ---------              --------
MIS         BLACKWOOD            $21,780.00
            CROSS                $27,062.00
PRODUCTION  BANNING              $29,700.00
            IRVING               $26,862.00
```

ソート句により除外された行の表示

ソート句では、表示するソート値の数を制限することができます。PLUS OTHERS 句により、すべての値を別のグループとして集計し、このグループを追加のレポート行として表示することができます。

ソート句により除外された行の表示

```
[RANKED] BY {HIGHEST|LOWEST|TOP|BOTTOM}  n srtfield [AS 'text']
            [PLUS OTHERS AS 'othertext']
            [IN-GROUPS-OF m1 [TOP n2]]
            [IN-RANGES-OF m3 [TOP n4]
```

説明

LOWEST

値の小さい方から大きい方へと昇順にソートします (文字フィールドは a から z、A から Z、0 から 9 の順、数値フィールドは 0 から 9 の順)。BOTTOM は LOWEST の同義語です。

HIGHEST

値の大きい方から小さい方へと降順にソートします。TOP は HIGHEST の同義語です。

n

レポート含めるフィールド値の個数を指定します。

srtfield

ソートフィールドの名前です。

text

ソートフィールド値のフィールドタイトルとして使用するテキストです。

othertext

「Others」グループの行タイトルとして使用するテキストです。この AS 句は、PLUS OTHERS 句の直後に記述する必要があります。

m1

ソートフィールドグループ間の増分値です。

n2

オプションとして指定する値で、最上位グループを示すラベルを定義し、レポートに表示します。

m3

0 (ゼロ) より大きい整数で、ソートフィールド値をグループ化する範囲を表します。

n4

オプションとして指定する値で、最上位の範囲を示すラベルを定義し、レポートに表示します。この値より上位にあるすべてのデータ値が対象範囲になります。

PLUS OTHERS の使用上の注意

- 文字のグループキーはサポートされません。
- 1 つのレポートでは 1 つの PLUS OTHERS 句のみがサポートされます。
- 複数の表示コマンドを含むリクエストでは、PLUS OTHERS 句を持つ BY フィールドは、リクエストの最下位 BY フィールドとなります。そうでない場合は、メッセージが表示され、リクエストは処理されません。
- PLUS OTHERS は、BY ROWS OVER、TILES、ACROSS、BY TOTAL 句とともに使用することはできません。
- PLUS OTHERS は MATCH FILE リクエストではサポートされません。ただし、TABLE リクエストの MORE はサポートされます。
- HOLD は、PDF、HTML、DOC、WP フォーマットでサポートされます。

ソート句により除外されたソートフィールド値行の表示

次のリクエストは、上位 2 つの ED_HRS 値を表示し、これらから除外された値を集計して「Others」というラベルの行に表示します。

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL LAST_NAME
  BY HIGHEST 2 ED_HRS
  PLUS OTHERS AS 'Others'
END
```

出力結果は次のとおりです。

```
ED_HRS         CURR_SAL  LAST_NAME
------         --------  ---------
 75.00       $21,780.00  BLACKWOOD
 50.00       $18,480.00  JONES
             $16,100.00  MCKNIGHT
Others      $165,924.00
```

ソートフィールドグループに含まれないデータ行の表示

次のリクエストは、最上位 2 つの ED_HRS によりソートし、ソートフィールド値を ED_HRS の 25 単位でグループ化します。最下位グループよりも下位の値は、Others カテゴリに追加されます。最上位グループラベルより上位の値は、すべて最上位グループに追加されます。

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL LAST_NAME
  BY HIGHEST 2 ED_HRS
  PLUS OTHERS AS 'Others'
IN-GROUPS-OF 25 TOP 50
END
```

出力結果は次のとおりです。

```
ED_HRS         CURR_SAL  LAST_NAME
------         --------  ---------
 50.00       $18,480.00  JONES
             $21,780.00  BLACKWOOD
             $16,100.00  MCKNIGHT
 25.00       $11,000.00  STEVENS
             $13,200.00  SMITH
             $26,862.00  IRVING
              $9,000.00  GREENSPAN
             $27,062.00  CROSS
Others       $78,800.00
```

BY HIGHEST 句が BY LOWEST に変更された場合、最上位グループ (ED_HRS 50 以上) より上位の値は、すべて Others カテゴリに追加されます。

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL LAST_NAME
  BY LOWEST 2 ED_HRS
  PLUS OTHERS AS 'Others'
IN-GROUPS-OF 25 TOP 50
END
```

出力結果は次のとおりです。

```
ED_HRS          CURR_SAL   LAST_NAME
------          --------   ---------
   .00         $9,500.00   SMITH
              $29,700.00   BANNING
              $21,120.00   ROMANS
              $18,480.00   MCCOY
 25.00        $11,000.00   STEVENS
              $13,200.00   SMITH
              $26,862.00   IRVING
               $9,000.00   GREENSPAN
              $27,062.00   CROSS
Others        $56,360.00
```

フィールドのソート

レポートのデータを横にソートするには、ACROSS 句を使用します。これにより、レポートにフィールドが作成されます。ACROSS フィールドの総数は、ACROSS フィールド値の総数と表示フィールドの総数の積と等しくなります。

リクエストには、BY 句と ACROSS 句を任意に組み合わせて、最大で 128 個のソート句を含めることができます。

レポートに含めることのできる表示フィールドの最大数は、複数の要因の組み合わせにより決定されます。一般的に、ACROSS フィールドにデータ値が多数存在すると、レポートに表示できる許容幅を超えたり、読みにくいレポートが作成されたりすることになります。詳細は、レポートデータの表示を参照してください。

ACROSS-TOTAL、SUBTOTAL、SUB-TOTAL、RECOMPUTE、SUMMARIZE を使用し、ACROSS フィールド値の列合計および集計を生成します。詳細は、合計と中間合計の追加を参照してください。

フィールドのソート

```
ACROSS sortfield
```

説明

sortfield

ソートフィールドの名前です。

フィールドをソートする際の注意

- テキストフィールドは、ソートフィールドとして使用することはできません。テキストフィールドは、マスターファイルで FORMAT の値に TX が定義されたフィールドです。
- DEFINE コマンドまたはマスターファイルの DEFINE 属性で作成された一時項目は、ソートフィールドとして使用することができます。ただし、COMPUTE コマンドで作成された一時項目は、ソートフィールドとして使用することはできません。この場合、BY TOTAL 句を使用するか、最初にこのフィールドを含む HOLD ファイルを作成しておき、その HOLD ファイルからレポートを作成することで、間接的に実行できます。HOLD ファイルについての詳細は、レポート出力の保存と再利用を参照してください。
- ACROSS 句を使用する場合は、SET SPACES パラメータを使用して ACROSS 項目の間隔を調節します。詳細は、レポートページのレイアウトを参照してください。
- マルチパスデータソースから複数のソートフィールドを指定してレポートを作成する場合は、すべてのソートフィールドが同一パスに存在する必要があります。
- スタイル出力フォーマットでは (PDF、HTML、DHTML、PPT、PPTX、XLSX)、データ列の上側に表示される ACROSS タイトルおよび ACROSS 値の幅は、ACROSS グループ内のすべてのデータ列および関連する列タイトルの中で、最大幅のもので決定されます。ACROSS グループのサイズを変更するには、各グループ内のデータ列に SQUEEZE、WRAP、WIDTH 定義のいずれかを適用します。
- ソートフィールドの値は、SET BYDISPLAY コマンドを使用してデフォルト設定を変更しない限り、レポートに 1 回だけ表示されます。たとえば、MIS 部の従業員数が 6 名の場合は、リクエストで次のように宣言します。

  ```
  PRINT LAST_NAME ACROSS DEPARTMENT
  ```

  このリクエストでは、「MIS」という部署名が 1 回だけ表示され、その後に 6 名の従業員の名前が表示されます。

ACROSS 句によるフィールドのソート

次の例は、総給与支出を部署別にソートして表示する方法を示しています。このリクエストは、ACROSS 句を使用してデータを横にソートします。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL ACROSS DEPARTMENT
END
```

出力結果は次のとおりです。

```
DEPARTMENT
MIS                     PRODUCTION
--------------------------------
$108,002.00           $114,282.00
```

横方向のソートにより、1 つのソートフィールド (部署名) に対して 1 つのフィールドが表示されていることが分かります。

単一フィールドの ACROSS タイトルの表示制御

SET ACRSVRBTITL コマンドを使用して、ACROSS グループの ACROSS フィールドタイトルの表示を制御することができます。タイトルの動作は、ACROSS グループ内の動詞フィールドの個数により決定されます。フィールドの個数は、レポートに内部マトリックスフィールドを追加する次の機能の影響を受けます。

- 見出しまたは脚注内のフィールド
- NOPRINT 句で非表示にされたフィールド
- フォーマットが再設定されたフィールド (通常、2 個のフィールドとしてカウントされる)
- 複数のフィールドを参照する COMPUTE コマンド

単一フィールドの ACROSS タイトルの表示制御

```
SET ACRSVRBTITL = {HIDEONE|ON|OFF}
ON TABLE SET ACRSVRBTITL {HIDEONE|ON|OFF}
```

説明

HIDEONE

表示フィールドが 1 つの場合にタイトルを非表示にします。また、表示フィールドが 1 つで、リクエストに内部マトリックスフィールドをレポートに追加する機能が 1 つ以上含まれている場合にもタイトルを非表示にします。これがデフォルト値です。

ON

表示フィールドが 1 つの場合でもタイトルを常に表示します。

OFF

表示フィールドが 1 つの場合にタイトルを非表示にします。表示フィールドが 1 つで、リクエストに内部マトリックスフィールドをレポートに追加する機能が 1 つ以上含まれている場合はタイトルを表示します。これは従来の動作です。

ACRSVRBTITL による ACROSS タイトルの非表示

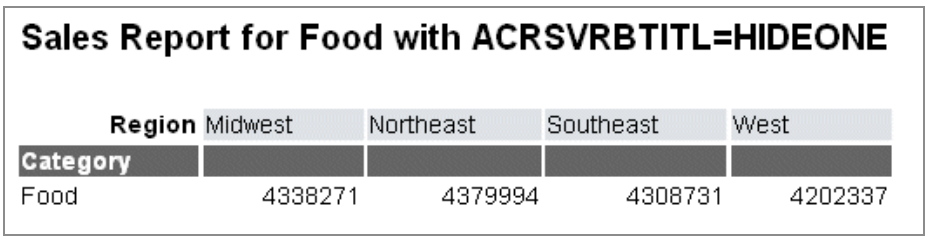
次のリクエストでは、GGSALES データソースを使用し、見出し内に表示フィールドが使用されています。

```
SET ACRSVRBTITL=HIDEONE
TABLE FILE GGSALES
HEADING
"Sales Report for <CATEGORY with ACRSVRBTITL=HIDEONE"
" "
SUM DOLLARS AS Sales
BY CATEGORY
ACROSS REGION
WHERE CATEGORY EQ 'Food'
```

```
ON TABLE SET PAGE NOPAGE
ON TABLE SET ACROSSTITLE SIDE
ON TABLE SET ACROSSLINE SKIP

ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9, SQUEEZE=ON,$
TYPE=TITLE,JUSTIFY=LEFT,BACKCOLOR=RGB(102 102 102),COLOR=RGB(255 255
255),STYLE=BOLD,$
TYPE=HEADING, SIZE=11, STYLE=BOLD,JUSTIFY=CENTER, $
TYPE=ACROSSTITLE,STYLE=BOLD,$
TYPE=ACROSSVALUE,BACKCOLOR=RGB(218 225 232),$
END
```

ACRSVRBTITL および HIDEONE のデフォルト値を使用すると、レポートフィールドを内部マトリックスに追加するフィールド値が見出しに表示される場合も、AROSS タイトルの「Sales」は非表示になります。

下図は、レポート出力を示しています。



Sales Report for Food with ACRSVRBTITL=HIDEONE

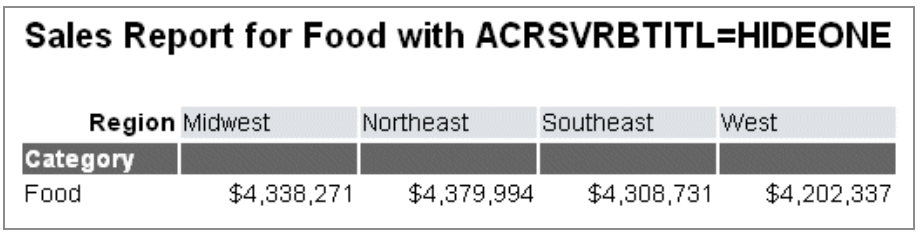
Region	Midwest	Northeast	Southeast	West
Category				
Food	4338271	4379994	4308731	4202337

SUM コマンドを次のように変更します。

```
SUM DOLLARS/D12CM
```

この場合、見出しのフィールドおよびフォーマットが再設定されたDollar Sales 値により、レポートフィールドが内部マトリックスに追加されますが、ACROSS タイトルの「Sales」は非表示のままです。

下図は、レポート出力を示しています。

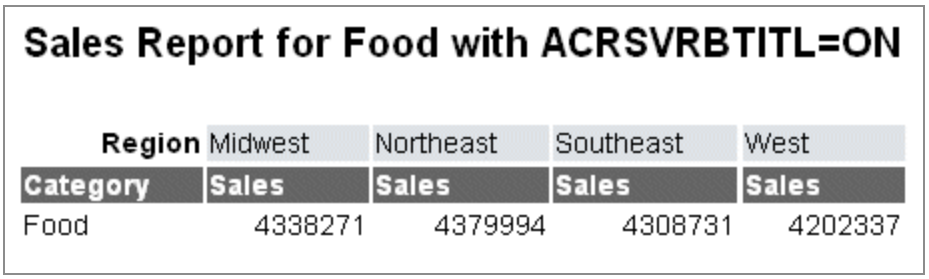


Sales Report for Food with ACRSVRBTITL=HIDEONE

Region	Midwest	Northeast	Southeast	West
Category				
Food	$4,338,271	$4,379,994	$4,308,731	$4,202,337

ACRSVRBTITL の値を ON に設定し、Dollar Sales フィールドのフォーマットを再設定しない場合は、レポートフィールドを内部マトリックスに追加するフィールド値が見出しに表示されるため、ACROSS タイトルの「Sales」は非表示になりません。

下図は、レポート出力を示しています。



Sales Report for Food with ACRSVRBTITL=ON

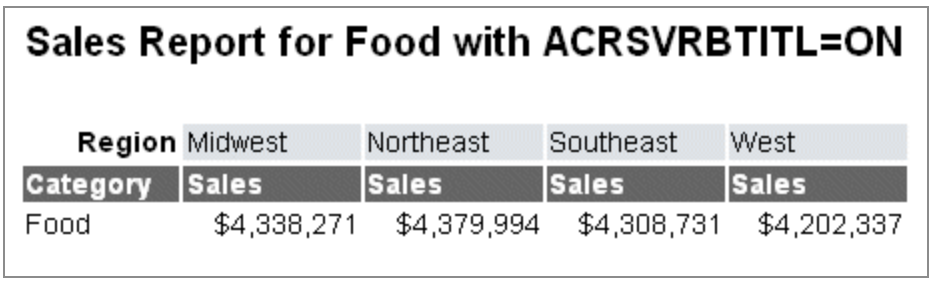
Region	Midwest	Northeast	Southeast	West
Category	Sales	Sales	Sales	Sales
Food	4338271	4379994	4308731	4202337

SUM コマンドを次のように変更します。

```
SUM DOLLARS/D12CMC
```

この場合、見出しのフィールドおよびフォーマットが再設定された Dollar Sales 値により、レポートフィールドが内部マトリックスに追加されるため、ACROSS タイトルの「Sales」は非表示になりません。

下図は、レポート出力を示しています。

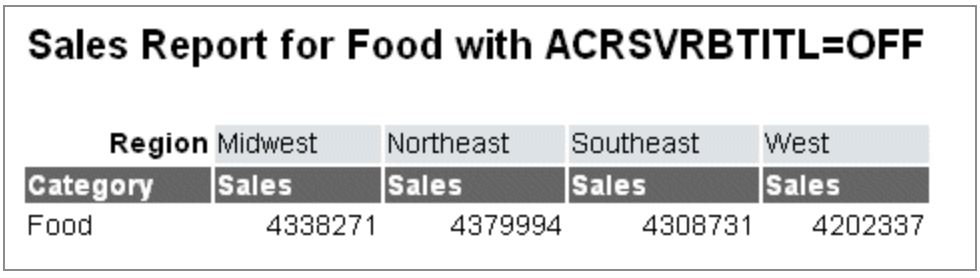


Sales Report for Food with ACRSVRBTITL=ON

Region	Midwest	Northeast	Southeast	West
Category	Sales	Sales	Sales	Sales
Food	$4,338,271	$4,379,994	$4,308,731	$4,202,337

ACRSVRBTITL=OFF に設定されている場合、見出しのフィールドにより、レポートフィールドが内部マトリックスに追加され、ACROSS タイトルの「Sales」は非表示になりません。

下図は、レポート出力を示しています。



Sales Report for Food with ACRSVRBTITL=OFF

Region	Midwest	Northeast	Southeast	West
Category	Sales	Sales	Sales	Sales
Food	4338271	4379994	4308731	4202337

SUM コマンドを次のように変更します。

```
SUM DOLLARS/D12CM
```

この場合、見出しのフィールドおよびフォーマットが再設定された Dollar Sales 値により、レポートフィールドが内部マトリックスに追加されるため、ACROSS タイトルの「Sales」は非表示になりません。

下図は、レポート出力を示しています。

Sales Report for Food with ACRSVRBTITL=OFF

Region	Midwest	Northeast	Southeast	West
Category	**Sales**	**Sales**	**Sales**	**Sales**
Food	$4,338,271	$4,379,994	$4,308,731	$4,202,337

レポート出力の ACROSS タイトル配置

ACROSS ソート句を使用してページの横方向にソートするレポートの場合、デフォルト設定では、レポート出力の ACROSS 列として、2 行が生成されます。最初の行にはソートフィールド名 (ACROSS タイトル) が表示され、2 行目にはそのソートフィールドの値 (ACROSS 値) が表示されます。ACROSS フィールド名は、最初の ACROSS 値の上部に左揃えで表示されます。

PDF、HTML、EXL2K、XLSX レポート出力で ACROSS タイトルと ACROSS 値の両方を 1 行に表示する場合は、SET ACROSSTITLE=SIDE コマンドを発行します。このコマンドは、ACROSS 値の左側に ACROSS タイトルを配置します。デフォルト設定では、タイトルは BY フィールドタイトルの上部に右揃えで表示されます。ACROSSTITLE コンポーネントのスタイルシートに JUSTIFY 属性を追加して、ACROSS タイトルの位置揃えを変更することができます。BY フィールドが存在しない場合は、デフォルト設定で作成される ACROSS タイトルを表示する見出し行は生成されません。

この機能は、ACROSS フィールドと BY フィールドの両方を含むリクエストで使用することができます。リクエストに ACROSS フィールドが存在し、BY フィールドが存在しない場合、この SET コマンドは無視され、ACROSS タイトルは移動されません。

特定の出力フォーマットでは、SET TITLELINE コマンドを使用して、フィールドタイトルに下線を追加するかどうかを制御することができます。SET ACROSSLINE は SET TITLELINE の同義語です。詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。

ACROSS フィールド名の配置制御

```
SET ACROSSTITLE = {ABOVE|SIDE}
```

説明

ABOVE

ACROSS タイトルを ACROSS 値の上部に表示します。デフォルト値は ABOVE です。

SIDE

ACROSS タイトルを ACROSS 値の左、BY 列の上部に表示します。

SET ACROSSTITLE 使用上の注意

- ACROSS 値が折り返される場合、ACROSS タイトルは、折り返された ACROSS 値の先頭行に揃えられます。
- ACROSS タイトルは、BY 列の幅で表示されます。ACROSS タイトルの値が現在のページの BY 列の幅よりも大きい場合、その値は切り捨てられます。SET BYPANEL が BY フィールド数の合計よりも小さい値に設定されている場合、最初のパネルには、それに続くパネルよりも多くの BY フィールドが表示されます。
- この設定により、レポート内にタイトルを配置する新しい列が作成されます。
  - リクエストに BY フィールドが含まれていない場合、ACROSS タイトルは移動されません。
  - BYPANEL=OFF の場合、ACROSS タイトルは、それに続くパネルには表示されません。
- SET ACROSSTITLE=SIDE を設定した ACROSSTITLE では、WRAP はサポートされません。

ACROSS タイトルの ACROSS 値との同列配置

次の例は、GGSALES データソースを使用して作成されたリクエストで、CATEGORY および PRODUCT という 2 つのソートフィールドが含まれています。SET ACROSSTITLE=SIDE によ

り、ACROSS タイトルが ACROSS 値の左側に移動されます。BYPANEL=ON の場合、ACROSS タイトルは、それに続く各パネルで同一の位置に表示されます。

```
SET ACROSSTITLE=SIDE
SET BYPANEL=ON
TABLE FILE GGSALES
SUM
     DOLLARS/I8M AS ''
BY REGION
BY ST
BY CITY
ACROSS CATEGORY
ACROSS PRODUCT
WHERE PRODUCT NE 'Capuccino';
ON TABLE SET PAGE-NUM ON
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
     UNITS=IN,
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,
$
TYPE=REPORT,
     FONT='ARIAL',
     SIZE=10,
     BORDER=LIGHT,
$
TYPE=ACROSSVALUE,
     WRAP=ON,
$
ENDSTYLE
END
```

ACROSS タイトルの「Category」 は、ACROSS 値 「Coffee」、「Food」、「Gifts」の左側に表示されます。ACROSS タイトルの「Product」は、ACROSS 値「Espresso」、「Latte」、「Biscotti」の左側に表示されます。ACROSS タイトルは、フィールド名「Region」、「State」、「City」の上部に、右揃えで表示されます。ACROSS 値「Croissant」は 2 行に折り返され、ACROSS タイトルは先頭行に揃えられます。下図は、1 つ目のパネルを示しています。

PAGE 1.1

		Category	Coffee		Food			Gifts
		Product	Espresso	Latte	Biscotti	Croissant	Scone	Coffee Grinder
Region	State	City						
Midwest	IL	Chicago	$420,439	$978,340	$378,412	$549,366	$595,069	$233,292
	MO	St. Louis	$419,143	$966,981	$368,077	$613,871	$481,953	$181,570
	TX	Houston	$455,365	$938,245	$345,238	$587,887	$418,398	$204,292
Northeast	CT	New Haven	$279,373	$926,052	$589,355	$551,489	$283,874	$169,908
	MA	Boston	$248,356	$917,737	$570,391	$497,234	$332,486	$177,940
	NY	New York	$322,378	$928,026	$642,259	$622,095	$290,811	$161,352
Southeast	FL	Orlando	$256,539	$889,887	$511,597	$602,076	$311,836	$217,204
	GA	Atlanta	$317,389	$907,365	$555,231	$661,806	$273,420	$217,254
	TN	Memphis	$279,644	$820,584	$438,889	$638,477	$315,399	$171,319
West	CA	Los Angeles	$267,809	$809,647	$266,030	$800,084	$315,584	$214,557
		San Francisco	$338,270	$935,862	$269,518	$824,457	$292,839	$187,123
	WA	Seattle	$301,538	$924,896	$328,320	$801,060	$304,445	$201,756

下図は、2 つ目のパネルを示しています。

PAGE 1.2

		Category	Gifts		
		Product	Coffee Pot	Mug	Thermos
Region	State	City			
Midwest	IL	Chicago	$204,828	$376,754	$187,901
	MO	St. Louis	$190,153	$343,852	$195,686
	TX	Houston	$204,897	$366,337	$194,319
Northeast	CT	New Haven	$208,209	$392,967	$221,827
	MA	Boston	$184,119	$401,944	$203,435
	NY	New York	$198,452	$349,300	$178,836
Southeast	FL	Orlando	$212,057	$409,466	$195,526
	GA	Atlanta	$232,552	$355,447	$227,482
	TN	Memphis	$200,694	$337,790	$209,449
West	CA	Los Angeles	$202,285	$381,926	$207,613
		San Francisco	$197,845	$379,399	$165,115
	WA	Seattle	$213,494	$427,339	$198,640

ACROSS タイトルの間隔

次の例は、GGSALES データソースを使用して作成されたリクエストで、2 つの BY フィールドと 2 つの ACROSS フィールドが含まれています。この例では罫線は設定されておらず、ACROSS 値の折り返しは無効になっています。SET ACROSSTITLE=SIDE により、ACROSS タイトルが ACROSS 値の左側に移動されます。SET BYPANEL=1 コマンドは、2 つ目のパネルで最初の BY フィールドのみを繰り返します。ここでは、ACROSS タイトルが BY フィール

ド上部の長さに合わせて切り捨てられることを防止するため、最初の BY フィールドは、デフォルト名よりも長い AS 名を指定してています。

```
SET ACROSSTITLE=SIDE
SET BYPANEL=1
TABLE FILE GGSALES
SUM
     DOLLARS/I8M AS ''
BY ST AS 'State Code'
BY CITY
ACROSS CATEGORY AS 'Categories'
ACROSS PRODUCT AS 'Products'
WHERE PRODUCT NE 'Capuccino';
ON TABLE SET PAGE-NUM ON
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
     UNITS=IN,
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,
$
TYPE=REPORT,
     FONT='ARIAL',
     SIZE=10,
   $
ENDSTYLE
END
```

最初のパネルは次のとおりです。

PAGE 1.1

State Code	Categories / Products / City	Coffee / Espresso	Latte	Food / Biscotti	Croissant	Scone
CA	Los Angeles	$267,809	$809,647	$266,030	$800,084	$315,584
	San Francisco	$338,270	$935,862	$269,518	$824,457	$292,839
CT	New Haven	$279,373	$926,052	$589,355	$551,489	$283,874
FL	Orlando	$256,539	$889,887	$511,597	$602,076	$311,836
GA	Atlanta	$317,389	$907,365	$555,231	$661,806	$273,420
IL	Chicago	$420,439	$978,340	$378,412	$549,366	$595,069
MA	Boston	$248,356	$917,737	$570,391	$497,234	$332,486
MO	St. Louis	$419,143	$966,981	$368,077	$613,871	$481,953
NY	New York	$322,378	$928,026	$642,259	$622,095	$290,811
TN	Memphis	$279,644	$820,584	$438,889	$638,477	$315,399
TX	Houston	$455,365	$938,245	$345,238	$587,887	$418,398
WA	Seattle	$301,538	$924,896	$328,320	$801,060	$304,445

SET BYPANEL=1 コマンドにより、2 つ目のパネルの BY フィールド上部の長さは、最初のパネル上部の長さよりも短くなります。AS 名「State Code」により、ACROSS タイトルの領域が追加されるため、2 つ目のパネルでは、タイトルは切り捨てられません。

PAGE 1.2

Categories Products State Code	Gifts Coffee Grinder	Coffee Pot	Mug	Thermos
CA	$214,557	$202,285	$381,926	$207,613
	$187,123	$197,845	$379,399	$165,115
CT	$169,908	$208,209	$392,967	$221,827
FL	$217,204	$212,057	$409,466	$195,526
GA	$217,254	$232,552	$355,447	$227,482
IL	$233,292	$204,828	$376,754	$187,901
MA	$177,940	$184,119	$401,944	$203,435
MO	$181,570	$190,153	$343,852	$195,686
NY	$161,352	$198,452	$349,300	$178,836
TN	$171,319	$200,694	$337,790	$209,449
TX	$204,292	$204,897	$366,337	$194,319
WA	$201,756	$213,494	$427,339	$198,640

ACROSSTITLE=SIDE 使用時の ACROSS 値の背景色指定

次のリクエストは、GGSALES データソースに対して実行され、ACROSS タイトルを ACROSS 値の横に配置し、ACROSS タイトル、ACROSS 値、列タイトルのフォント色と背景色を統一するため、これらの項目に白のテキストと灰色の背景色を設定します。

```
SET ACROSSTITLE=SIDE
TABLE FILE GGSALES
SUM DOLLARS/I8M AS ''
BY REGION
BY ST
BY CITY
ACROSS CATEGORY
ACROSS PRODUCT
WHERE CATEGORY EQ 'Coffee' OR 'Food';
ON TABLE SET PAGE-NUM NOPAGE
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
```

```
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
SQUEEZE=ON,UNITS=IN,ORIENTATION=PORTRAIT,$
TYPE=REPORT,FONT='ARIAL',SIZE=10,BORDER=LIGHT,$
TYPE=ACROSSTITLE,COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=ACROSSVALUE,COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=TITLE,COLOR=WHITE, BACKCOLOR=GREY,$
ENDSTYLE
END
```

出力結果では、ACROSS タイトル、ACROSS 値、列タイトルが白のテキストと灰色の背景色で表示されています。

		Category	Coffee			Food		
		Product	Capuccino	Espresso	Latte	Biscotti	Croissant	Scone
Region	State	City						
Midwest	IL	Chicago	.	$420,439	$978,340	$378,412	$549,366	$595,069
	MO	St. Louis	.	$419,143	$966,981	$368,077	$613,871	$481,953
	TX	Houston	.	$455,365	$938,245	$345,238	$587,887	$418,398
Northeast	CT	New Haven	$158,995	$279,373	$926,052	$589,355	$551,489	$283,874
	MA	Boston	$174,344	$248,356	$917,737	$570,391	$497,234	$332,486
	NY	New York	$208,756	$322,378	$928,026	$642,259	$622,095	$290,811
Southeast	FL	Orlando	$317,027	$256,539	$889,887	$511,597	$602,076	$311,836
	GA	Atlanta	$352,161	$317,389	$907,365	$555,231	$661,806	$273,420
	TN	Memphis	$274,812	$279,644	$820,584	$438,889	$638,477	$315,399
West	CA	Los Angeles	$306,468	$267,809	$809,647	$266,030	$800,084	$315,584
		San Francisco	$279,830	$338,270	$935,862	$269,518	$824,457	$292,839
	WA	Seattle	$309,197	$301,538	$924,896	$328,320	$801,060	$304,445

複数 ACROSS フィールドの使用

複数のソートフィールドを使用して、レポートをソートすることができます。複数のソートフィールドを使用する場合、ACROSS 句を定義した順序によりソート順が決定されます。最初の ACROSS 句により先頭のソート区切りが設定され、2 つ目の ACROSS 句によりその次のソート区切りが設定されるという具合です。後に続くソートは、直前のソートにネストされます。

複数 ACROSS ソート句によるソート

次のリクエストは、現在の給与合計を最初は部署別に、次にジョブコード別にソートします。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL
ACROSS DEPARTMENT ACROSS CURR_JOBCODE
WHERE CURR_SAL GT 21500
END
```

出力結果は次のとおりです。

```
DEPARTMENT
                  MIS                                PRODUCTION

CURR_JOBCODE
                  A17              B04               A15               A17
-------------------------------------------------------------------------------
           $27,062.00       $21,780.00       $26,862.00       $29,700.00
```

ACROSS 使用時の PRINT の圧縮

PRINT コマンドは、IF 条件または WHERE 条件に一致しないレコードを除外した後、データソースから取得された各レコードを 1 行に表示するレポートを生成します。PRINT コマンドを ACROSS 句とともに使用すると、生成された列の多くはブランクになります。それらの列には、ミッシングデータ記号が表示されます。

このように、レポートに多くのミッシングデータが表示されることを防止するには、SET ACROSSPRT コマンドを使用して、レポートの行を圧縮します。下位行のミッシングではない値を上位行のものと入れ替え、さらにすべての列がミッシング値である行を削除することで、各ソートグループ内の行数が減少します。

データが別のレポート行に移動される可能性があるため、ROW-TOTAL や ACROSS-TOTAL など、圧縮されたレポートの行を基準とする計算は、圧縮されていないレポートとは異なります。レポートの行を圧縮しても、行の計算に影響を与えることはありません。

レポート行の圧縮

```
SET ACROSSPRT = {NORMAL|COMPRESS}
```

```
ON TABLE SET ACROSSPRT{NORMAL|COMPRESS}
```

説明

NORMAL

レポート行を圧縮しません。デフォルト値は NORMAL です。

COMPRESS

データ値を同一ソートグループ内の上位に移動してミッシング値を置き換え、レポート行を圧縮します。

SET ACROSSPRT 使用上の注意

- 圧縮は、ACROSS ... COLUMNS を含めた ACROSS フィールドにのみ適用されます。BY フィールドには影響しません。
- データ値で圧縮の対象になるのは、実際のミッシング値のみです。格納されたデータが 0 (ゼロ)、ブランクのいずれかで、メタデータで MISSING が ON に指定されている場合、その値は圧縮の対象外になります。

SET ACROSSPRT によるレポート出力の圧縮

次のリクエストは、GGSALES データソースを使用して、UNITS を PRODUCT で縦方向に、REGION で横方向にソートして表示します。

```
TABLE FILE GGSALES
PRINT UNITS/I5
BY PRODUCT
ACROSS REGION
WHERE DATE FROM '19971201' TO '19971231';
WHERE PRODUCT EQ 'Capuccino' OR 'Espresso';
ON TABLE SET ACROSSPRT NORMAL
ON TABLE SET PAGE NOPAGE
END
```

レポートの各行は、1 つの REGION の 1 つの UNITS を表すことから、ACROSSPRT を NORMAL にした場合、各行で 1 つの列に、ミッシングではない値が表示されます。

```
                  Region
                  Midwest     Northeast   Southeast   West
Product           Unit Sales  Unit Sales  Unit Sales  Unit Sales
----------------------------------------------------------------
Capuccino                .         936           .           .
                         .         116           .           .
                         .         136           .           .
                         .           .        1616           .
                         .           .        1118           .
                         .           .         774           .
                         .           .           .        1696
                         .           .           .        1519
                         .           .           .         836
Espresso              1333           .           .           .
                       280           .           .           .
                       139           .           .           .
                         .        1363           .           .
                         .         634           .           .
                         .         406           .           .
                         .           .        1028           .
                         .           .        1014           .
                         .           .         885           .
                         .           .           .        1782
                         .           .           .        1399
                         .           .           .         551
```

ACROSSPRT を COMPRESS に設定すると、ミッシングではない値を同一 BY グループ内で上位に移動して、ミッシング値で構成されるすべての行を削除します。

```
TABLE FILE GGSALES
PRINT UNITS/I5
BY PRODUCT
ACROSS REGION
WHERE DATE FROM '19971201' TO '19971231';
WHERE PRODUCT EQ 'Capuccino' OR 'Espresso';
ON TABLE SET ACROSSPRT COMPRESS
ON TABLE SET PAGE NOPAGE
END
```

出力結果は次のとおりです。

```
                  Region
                  Midwest     Northeast   Southeast   West
Product           Unit Sales  Unit Sales  Unit Sales  Unit Sales
----------------------------------------------------------------
```

```
Capuccino                .         936        1616        1696
                         .         116        1118        1519
                         .         136         774         836
Espresso              1333        1363        1028        1782
                       280         634        1014        1399
                       139         406         885         551
```

ACROSS グループのブランク列の非表示

リクエストで ACROSS ソート句を使用すると、ACROSS フィールドのそれぞれの値の下に一連の列が生成されます (リクエストの表示フィールドごとに 1 列)。多くの場合、これらの列の中には、ミッシング値 (NULL 値) のみが存在するブランク列があります。スタイルが設定された出力フォーマットでは、HIDENULLACRS パラメータを使用して、ミッシング値のみが存在する ACROSS グループを非表示にすることができます。BY フィールドを使用して PAGE-BREAK オプションを指定した場合は、その PAGE-BREAK オプションで生成される出力結果のページごとに列が非表示になります。リクエストに BY フィールドの PAGE-BREAK オプションが含まれていない場合は、ミッシング値のみが存在する ACROSS グループの列はレポート全体で非表示になります。

ACROSS のブランク列の非表示は、EXL2K PIVOT および EXL2K FORMULA オプションを除いて、スタイル設定されたすべての出力フォーマットでサポートされます。

ACROSS ブランク列の非表示

```
SET HIDENULLACRS = {ON|OFF}
```

```
ON TABLE SET HIDENULLACRS {ON|OFF}
```

説明

ON

BY で生成された改ページごとに、ACROSS グループ内のミッシング値の列を非表示にします。

OFF

列を非表示にしません。デフォルト値は OFF です。

ACROSS グループ内のブランク列を非表示にする際の注意

- ACROSS を使用したレポートでは、見出し内の項目をデータ列に合わせて配置する機能 (HEADALIGN) はサポートされません。
- ACROSS のブランク列の非表示は、スポットマーカーまたは明示的な配置で見出し要素に配置された項目には影響しません。つまり、ACROSS グループの列を非表示にした後、ACROSS の列に合わせて項目が期待どおりに配置されない場合があります。

ACROSS ブランク列の非表示がサポートされない機能

- EXL2K FORMULA
- EXL2K PIVOT
- OVER.
- HIDENULLACRS は、ON byfieldname PAGE-BREAK 句または BY fieldname PAGE-BREAK 句で指定された改ページとともに使用する場合にのみサポートされます。次の場合はサポートされません。
    - BY **field** ROWS **value** OVER.
    - FML FOR フィールド (FOR field value OVER PAGE-BREAK)

BY 改ページ内の ACROSS グループおよび列の非表示

ブランク列の非表示は、BY ソートフィールドに PAGE-BREAK オプション (BY 句で直接指定、または ON 句で指定) を指定した場合に最も効果的に機能します。BY フィールドの改ページは、その BY フィールドの値が変化する場所で生成されます。BY フィールドの値が複数ページにまたがる場合でも、ACROSS 列が非表示になる境界は、BY フィールドの値が変化する場所で定義されます。

この設定で特定の BY フィールドを指定する方法はありません。そのため、リクエストに複数の BY フィールドがあり、改ページが指定されている場合、この設定はすべての BY フィールドに適用されます。BY フィールドに改ページが指定されていない場合、ACROSS 列を非表示にするには、レポート全体で ACORSS 列の値がミッシング値でなければなりません。

ACROSS グループ全体が非表示になるのは、ACROSS 値が存在しない場合、または特定の BY フィールドの値で、ACROSS 値の下のすべての表示フィールドのすべての行にミッシング値が存在する場合です。

改ページを使用して BY フィールドの値ごとに生成される一連のページが非表示になるのは、その BY フィールド値ですべての ACROSS グループが非表示になる場合です。

ページまたはパネルから列が除外されると、その空き領域に既存の列が再配置されます。

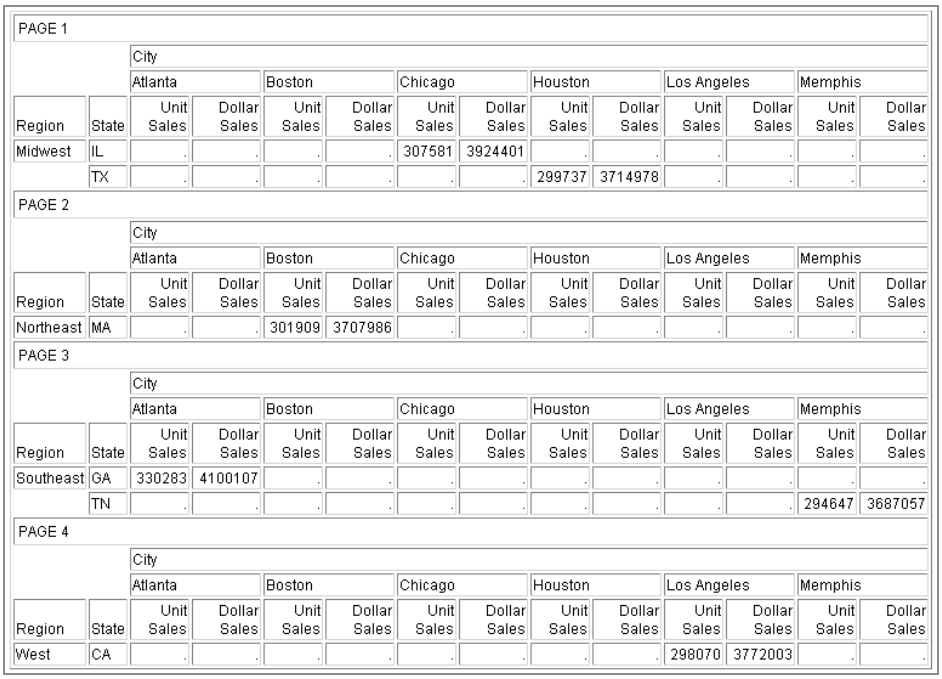
ACROSS のブランクグループの非表示

次のリクエストは、GGSALES データソースに対して実行され、REGION という BY フィールドに改ページを指定し、CITY フィールドに ACROSS 句を指定しています。各 ACROSS グループの表示フィールドは、UNITS および DOLLARS です。

```
SET HIDENULLACRS=OFF
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY REGION PAGE-BREAK
BY ST
ACROSS CITY
WHERE CITY LE 'Memphis'
ON TABLE SET HTMLCSS ON
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET STYLE *
TYPE=REPORT, FONT=ARIAL, SIZE=9,$
```

```
ENDSTYLE
END
```

SET HIDENULLACRS=OFF を使用すると、すべてのフィールドが表示されます。



PAGE 1

		City											
		Atlanta		Boston		Chicago		Houston		Los Angeles		Memphis	
Region	State	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales
Midwest	IL	.	.	.	.	307581	3924401	.	.	.	.	.	.
	TX	.	.	.	.	.	.	299737	3714978	.	.	.	.

PAGE 2

		City											
		Atlanta		Boston		Chicago		Houston		Los Angeles		Memphis	
Region	State	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales
Northeast	MA	.	.	301909	3707986	.	.	.	.	.	.	.	.

PAGE 3

		City											
		Atlanta		Boston		Chicago		Houston		Los Angeles		Memphis	
Region	State	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales
Southeast	GA	330283	4100107	.	.	.	.	.	.	.	.	.	.
	TN	.	.	.	.	.	.	.	.	.	.	294647	3687057

PAGE 4

		City											
		Atlanta		Boston		Chicago		Houston		Los Angeles		Memphis	
Region	State	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales
West	CA	.	.	.	.	.	.	.	.	298070	3772003	.	.

このリクエストを SET HIDENULLACRS=ON で実行すると、地域ごとにミッシング値が存在する都市の ACROSS グループが除外されます。たとえば、Midwest 地域には、Atlanta および Boston 列は表示されていません。

PAGE 1

		City			
		Chicago		Houston	
Region	State	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales
Midwest	IL	307581	3924401	.	.
	TX	.	.	299737	3714978

PAGE 2

		City	
		Boston	
Region	State	Unit Sales	Dollar Sales
Northeast	MA	301909	3707986

PAGE 3

		City			
		Atlanta		Memphis	
Region	State	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales
Southeast	GA	330283	4100107	.	.
	TN	.	.	294647	3687057

PAGE 4

		City	
		Los Angeles	
Region	State	Unit Sales	Dollar Sales
West	CA	298070	3772003

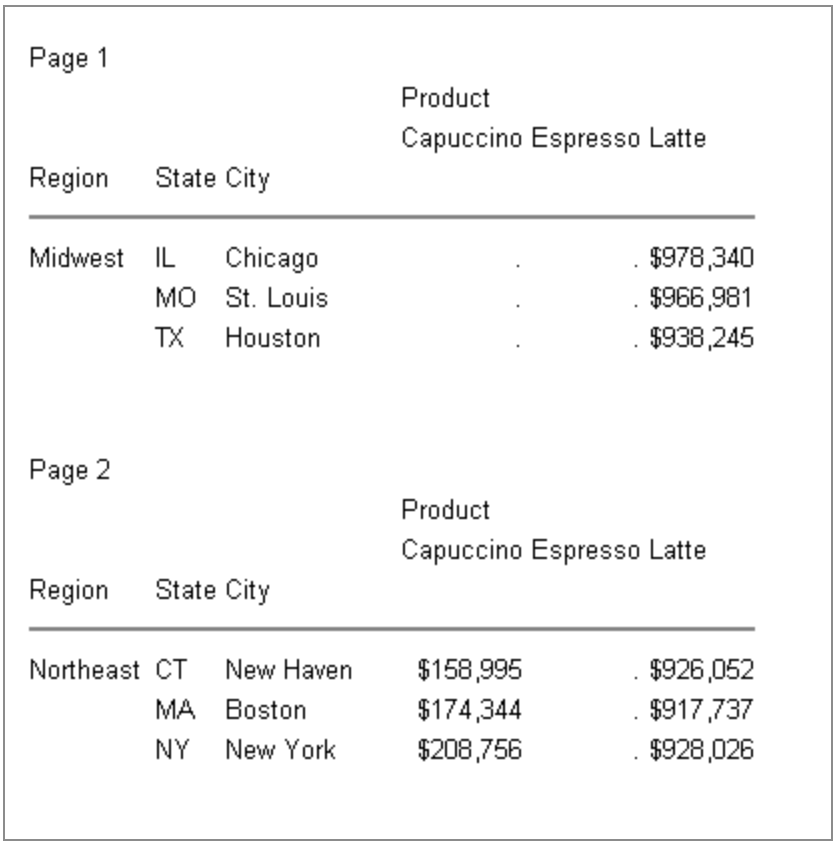
ACROSS グループ内の列の非表示

次のリクエストは、GGSALES データソースに対して実行され、REGION という BY フィールドに PAGE-BREAK オプションを指定し、ACROSS フィールドとして PRODUCT を指定しています。DEFINE コマンドで作成される SHOWDOLLARS という一時項目 (DEFINE) には、ACROSS グループである Coffee の Espresso 列にミッシング値が存在します。

```
SET HIDENULLACRS=OFF
SET BYPANEL=2
DEFINE FILE GGSALES
SHOWDOLLARS/I8M MISSING ON = IF (PRODUCT EQ 'Espresso') THEN MISSING ELSE
DOLLARS;
END
TABLE FILE GGSALES
HEADING
```

```
"Page <TABPAGENO "
SUM SHOWDOLLARS AS ''
BY REGION
BY ST
BY CITY
ACROSS PRODUCT
WHERE REGION EQ 'Midwest' OR 'Northeast'
WHERE CATEGORY EQ 'Coffee';
ON REGION PAGE-BREAK
ON TABLE SET PAGE-NUM ON
ON TABLE NOTOTAL
ON TABLE SET HTMLCSS ON
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET STYLE *
     UNITS=IN,
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,
$
TYPE=REPORT,
     GRID=OFF,
     FONT='ARIAL',
     SIZE=9,
$
ENDSTYLE
END
```

このリクエストを SET HIDENULLACRS=OFF で実行すると、Coffee グループ内でミッシング値が存在する Espresso 列およびその他のすべての列が表示されます。



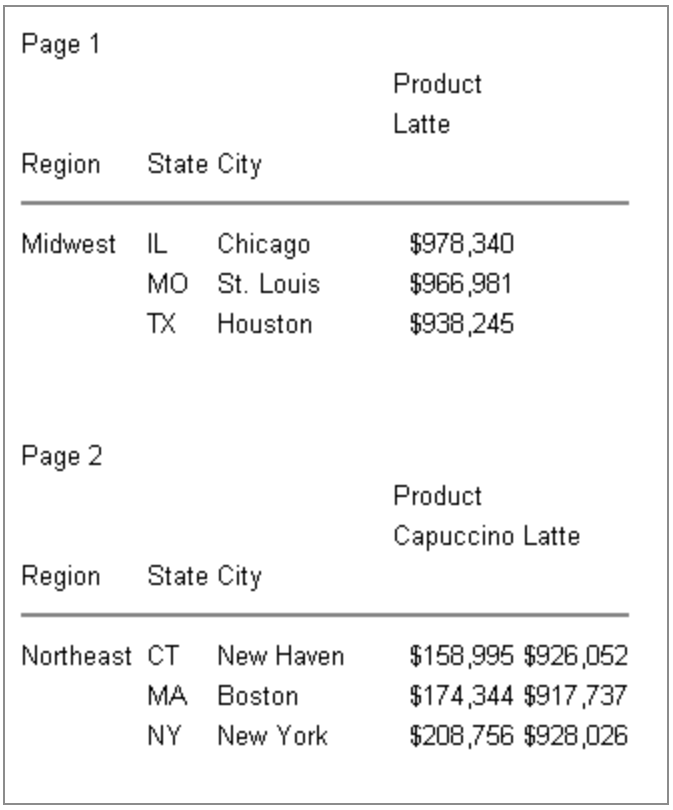
Page 1

Region	State	City	Product Capuccino	Espresso	Latte
Midwest	IL	Chicago	.	.	$978,340
	MO	St. Louis	.	.	$966,981
	TX	Houston	.	.	$938,245

Page 2

Region	State	City	Product Capuccino	Espresso	Latte
Northeast	CT	New Haven	$158,995	.	$926,052
	MA	Boston	$174,344	.	$917,737
	NY	New York	$208,756	.	$928,026

このリクエストを SET HIDENULLACRS=ON で実行すると、地域ごとにミッシング値が存在する列が非表示になります。1 ページ目 (Midwest) では、Capuccino および Espresso 列が非表示になっていますが、2 ページ目 (Northeast) では、Espresso 列のみが非表示になっています。



Page 1

Region	State	City	Product Latte
Midwest	IL	Chicago	$978,340
	MO	St. Louis	$966,981
	TX	Houston	$938,245

Page 2

Region	State	City	Product Capuccino	Latte
Northeast	CT	New Haven	$158,995	$926,052
	MA	Boston	$174,344	$917,737
	NY	New York	$208,756	$928,026

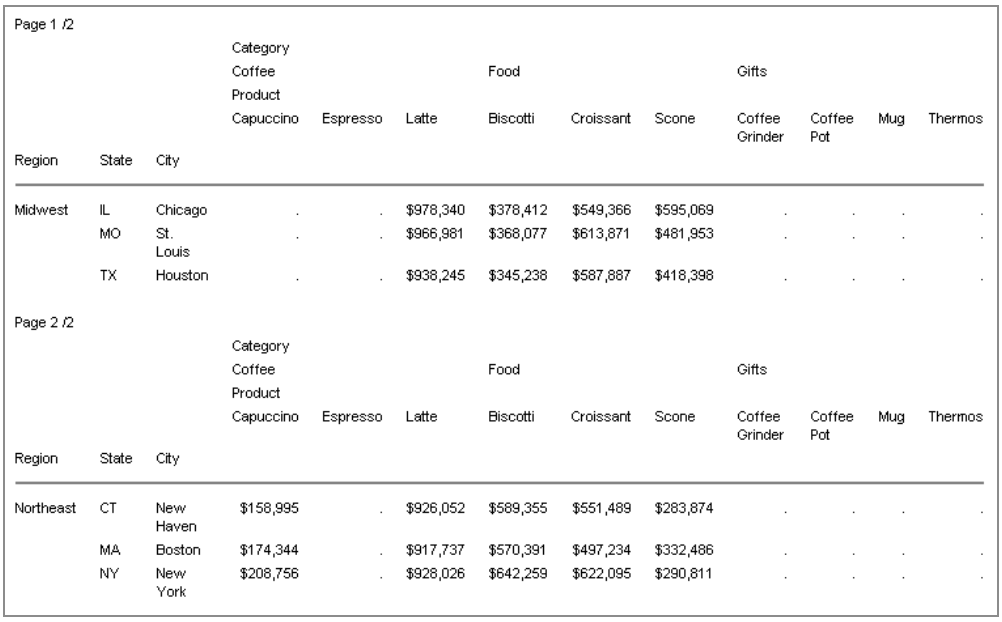
複数の ACROSS フィールドでのブランク列の非表示

次のリクエストは、GGSALES データソースに対して実行され、ACROSS フィールドとして CATEGORY および PRODUCT を指定しています。PAGE-BREAK オプションを使用した BY フィールドとして REGION を指定しています。DEFINE コマンドで作成される SHOWDOLLARS という一時項目 (DEFINE) には、ACROSS グループである Coffee の Espresso 列、および ACROSS グループである Gifts のすべての列にミッシング値が存在します。

```
SET HIDENULLACRS=OFF
DEFINE FILE GGSALES
SHOWDOLLARS/I8M MISSING ON = IF (PRODUCT EQ 'Espresso' OR
   CATEGORY EQ 'Gifts') THEN MISSING ELSE DOLLARS;
END
TABLE FILE GGSALES
SUM SHOWDOLLARS AS ''
BY REGION
```

```
BY ST
BY CITY
ACROSS CATEGORY
ACROSS PRODUCT
WHERE REGION EQ 'Midwest' OR 'Northeast'
ON REGION PAGE-BREAK
HEADING
"Page <TABPAGENO /<TABLASTPAGE "
ON TABLE SET PAGE-NUM OFF
ON TABLE SET BYPANEL ON
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
     UNITS=IN,
     PAGESIZE='Letter',
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,
$
TYPE=REPORT,
     HEADPANEL=ON,
     GRID=OFF,
     FONT='ARIAL',
     SIZE=8,
$
ENDSTYLE
END
```

このリクエストを SET HIDENULLACRS=OFF で実行すると、すべての列が表示されます。



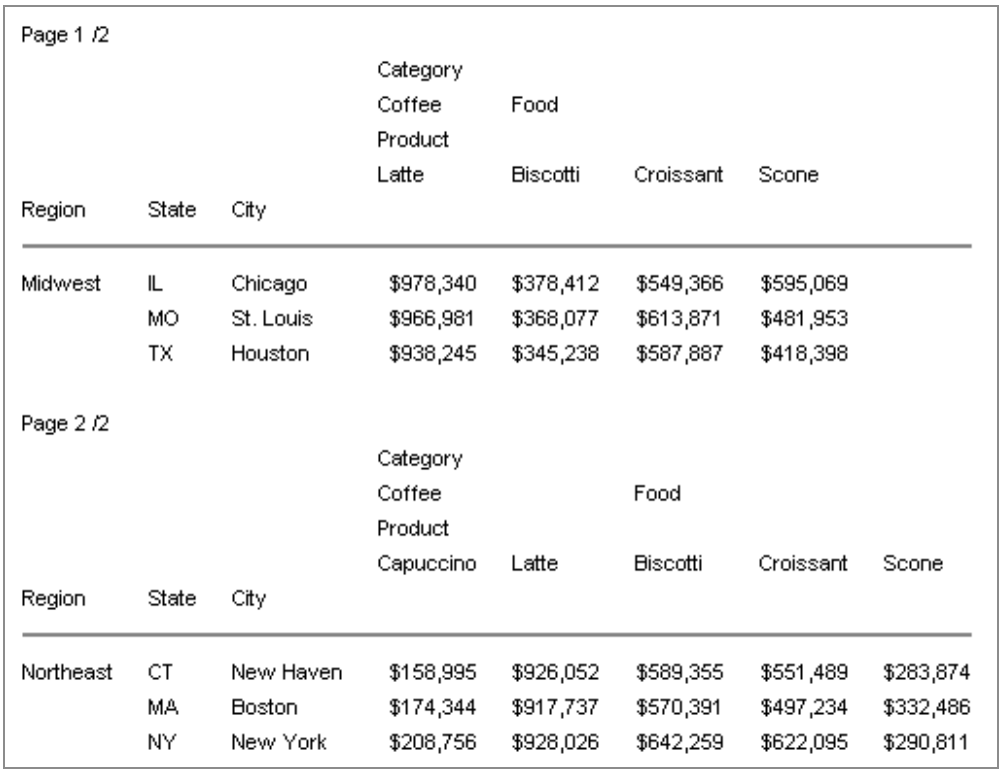
Page 1 /2

Region	State	City	Category Coffee Product Capuccino	Espresso	Latte	Food Biscotti	Croissant	Scone	Gifts Coffee Grinder	Coffee Pot	Mug	Thermos
Midwest	IL	Chicago	.	.	$978,340	$378,412	$549,366	$595,069	.	.	.	.
	MO	St. Louis	.	.	$966,981	$368,077	$613,871	$481,953	.	.	.	.
	TX	Houston	.	.	$938,245	$345,238	$587,887	$418,398	.	.	.	.

Page 2 /2

Region	State	City	Category Coffee Product Capuccino	Espresso	Latte	Food Biscotti	Croissant	Scone	Gifts Coffee Grinder	Coffee Pot	Mug	Thermos
Northeast	CT	New Haven	$158,995	.	$926,052	$589,355	$551,489	$283,874	.	.	.	.
	MA	Boston	$174,344	.	$917,737	$570,391	$497,234	$332,486	.	.	.	.
	NY	New York	$208,756	.	$928,026	$642,259	$622,095	$290,811	.	.	.	.

このリクエストを SET HIDENULLACRS=ON で実行すると、地域ごとに Espresso 列および Gifts のすべての列が非表示になります。1 ページ目 (Midwest) では、Gift のすべての列と Capuccino および Espresso 列が非表示になっていますが、2 ページ目 (Northeast) では、Gift のすべての列と Espresso 列が非表示になっています。



Page 1 /2

Region	State	City	Category Coffee Product Latte	Food Biscotti	Croissant	Scone
Midwest	IL	Chicago	$978,340	$378,412	$549,366	$595,069
	MO	St. Louis	$966,981	$368,077	$613,871	$481,953
	TX	Houston	$938,245	$345,238	$587,887	$418,398

Page 2 /2

Region	State	City	Category Coffee Product Capuccino	Latte	Food Biscotti	Croissant	Scone
Northeast	CT	New Haven	$158,995	$926,052	$589,355	$551,489	$283,874
	MA	Boston	$174,344	$917,737	$570,391	$497,234	$332,486
	NY	New York	$208,756	$928,026	$642,259	$622,095	$290,811

集計行の生成および ACROSS ブランク列の非表示

ACROSS グループ全体が非表示になると、それに関連する BY フィールド値で生成される合計も非表示になります。ACROSS 値のいずれかの列にミッシング値以外の値が含まれる場合、その ACROSS グループにはそのミッシング値以外の値を含む列が表示されます。

集計要素は、ACROSS グループの列に結合された状態で保持されます。ACROSS グループが非表示になると、それに関連する集計値も非表示になり、その後に続く値はそれぞれの ACROSS 列に合わせて再配置されます。

BY フィールドの改ページの位置で生成された集計行は、その BY フィールド値の最終ページの最後部に表示されます。BY フィールド値全体を通してミッシング値以外の値が含まれているすべての ACROSS グループは、BY フィールド値の一部のページでこれらのグループが非表示である場合でも、集計行に表示されます。これにより、関連する集計値を表示することができます。

ACROSS 列が BY フィールド値の一部のページで非表示になっている場合であっても、その ACROSS フィールドの列を総合計に含めることができます。これにより、集計行は常に別ページに表示され、ACROSS グループおよび列がレポート内のどのページで非表示になっているかに関係なく、レポート内のいずれかのページに表示されている限り、すべての ACROSS グループおよび列の集計が表示されます。

BY フィールドの最下位の改ページの外側で BY フィールドの集計行を定義した場合でも、内側の BY フィールドに対して非表示にした ACROSS 列を集計行に含めることができます。このため、これらの集計行には常に ACROSS のすべての列が表示され、集計行は別ページに表示されます。

列合計のすべてがミッシング値の場合は、列ごとに計算されたすべての合計 (ACROSSTOTAL、ROWTOTAL) が非表示になります。

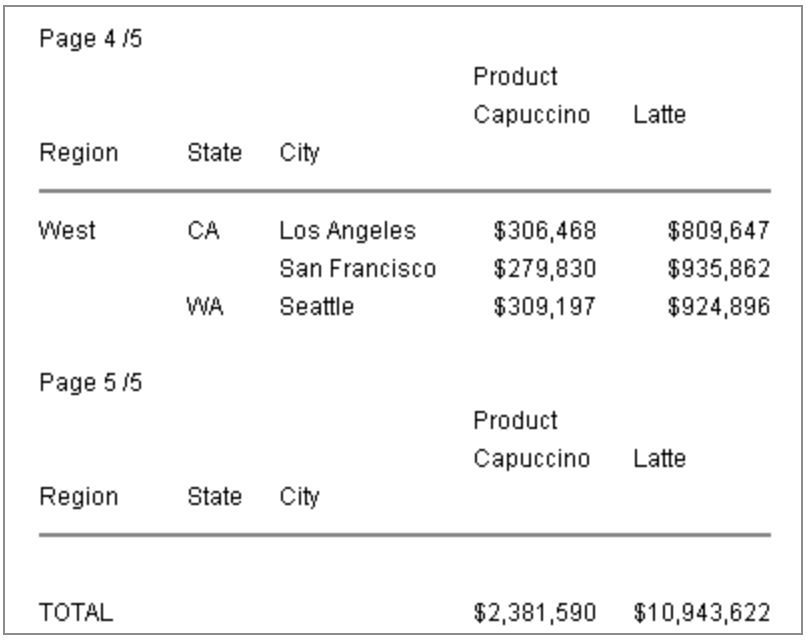
列合計の生成および ACROSS ブランク列の非表示

次のリクエストは、GGSALES データソースに対して実行され、REGION という BY フィールドに PAGE-BREAK オプションを指定し、ACROSS フィールドとして PRODUCT を指定しています。DEFINE コマンドで作成される SHOWDOLLARS という一時項目 (DEFINE) には、ACROSS グループである Coffee の Espresso 列にミッシング値が存在します。列合計は、レポートの最後に生成されます。

```
SET HIDENULLACRS=ON
DEFINE FILE GGSALES
SHOWDOLLARS/I8M MISSING ON = IF (PRODUCT EQ 'Espresso') THEN MISSING ELSE
DOLLARS;
END
TABLE FILE GGSALES
SUM SHOWDOLLARS AS ''
BY REGION
BY ST
BY CITY
ACROSS PRODUCT
ON REGION PAGE-BREAK
HEADING
"Page <TABPAGENO /<TABLASTPAGE "
WHERE CATEGORY EQ 'Coffee';
ON TABLE SET PAGE-NUM OFF
ON TABLE SET BYPANEL ON
ON TABLE COLUMN-TOTAL AS 'TOTAL'
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
     UNITS=IN,
     PAGESIZE='Letter',
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,
$
TYPE=REPORT,
     HEADPANEL=ON,
     GRID=OFF,
     FONT='ARIAL',
     SIZE=9,
$
ENDSTYLE
END
```

このリクエストを実行すると、REGION の改ページごとにブランク列が非表示になり、列合計が別のページに生成されます。

下図は、1 ページ目から 3 ページ目を示しています。1 ページ目では、Espresso および Capuccino 列が非表示になっています。2 ページ目および 3 ページ目では、Espresso 列が非表示になっています。



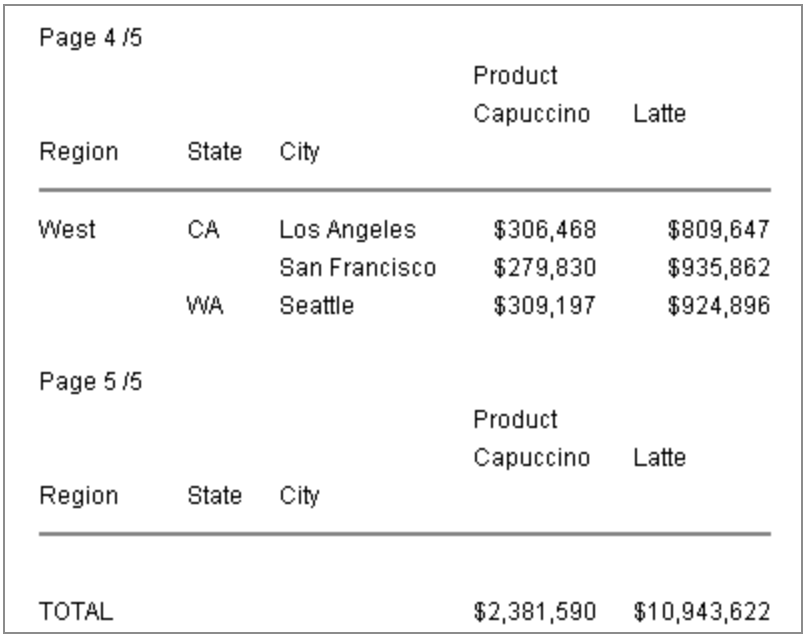
Page 4 /5

Region	State	City	Product Capuccino	Latte
West	CA	Los Angeles	$306,468	$809,647
		San Francisco	$279,830	$935,862
	WA	Seattle	$309,197	$924,896

Page 5 /5

Region	State	City	Product Capuccino	Latte
TOTAL			$2,381,590	$10,943,622

下図は、4 ページ目および 5 ページ目を示しています。4 ページ目では、Espresso 列が非表示になっています。5 ページ目は、合計のページです。このページでは、Espresso 列が非表示になっています。これは、すべての詳細ページで Espresso 列が非表示になっているためです。ただし、Capuccino 列は、表示されている詳細ページがあるため、このページでは非表示になっていません。



Page 4 /5

Region	State	City	Product Capuccino	Latte
West	CA	Los Angeles	$306,468	$809,647
		San Francisco	$279,830	$935,862
	WA	Seattle	$309,197	$924,896

Page 5 /5

Region	State	City	Product Capuccino	Latte
TOTAL			$2,381,590	$10,943,622

列スタイルの使用および ACROSS ブランク列の非表示

ブランク列を非表示にした結果、レポート出力で列の位置が変更された場合でも、列のスタイル設定は元の列に保持されます。具体的には次のようになります。

- 列を非表示にした後の境界および背景色は、結果のパネルまたはページレイアウトに適合するように再調整されます。
- 特定の列に指定したスタイル設定は、その列に保持され、非表示にした列の影響は受けません。たとえば、ACROSS フィールドの 3 列目に条件付きスタイルが定義されている場合に 2 列目を非表示にすると、3 列目は出力結果で 2 列目になりますが、当初 3 列目に設定されていたスタイルがこの列に保持されます。

列のスタイル設定についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

列スタイルの使用および ACROSS ブランク列の非表示

次のリクエストは、GGSALES データソースに対して実行され、REGION という BY フィールドに PAGE-BREAK オプションを指定し、ACROSS フィールドとして PRODUCT を指定しています。DEFINE コマンドで一時項目 (DEFINE) の SHOWDOLLARS フィールドが作成され、このフィールドの Midwest 地域の Capuccino 列、Northeast 地域の Thermos 列、Southeast 地域の Scone 列、West 地域のすべての列にミッシング値が存在します。地域ごとに、列合計、行合計、中間合計が生成されます。

一部の列には、背景色が割り当てられています。

- C5 フィールドには、BACKCOLOR=WHEAT が割り当てられています。C5 は、表示フィールドを左から右へ数えた場合の 5 列目に相当しますが、BY フィールドおよび ROW-TOTAL フィールドはカウントに入れません。C5 フィールドは、Coffee グループの Croissant 列に対応します。
- P5 フィールドには、BACKCOLOR=THISTLE が割り当てられています。P5 は、表示フィールド、BY フィールド、ROW-TOTAL フィールドを左から右へ数えた場合の 5 列目に相当しますが、NOPRINT フィールドはカウントに入れません。P5 フィールドは、Coffee グループの Espresso 列に対応します。
- N7 フィールドには、BACKCOLOR=MEDIUM GOLDENROD が割り当てられています。

N7 は、表示フィールド、BY フィールド、ROW-TOTAL フィールド、NOPRINT フィールドを左から右へ数えた場合の 7 列目に相当します。N7 フィールドは、Food グループの Biscotti 列に対応します。

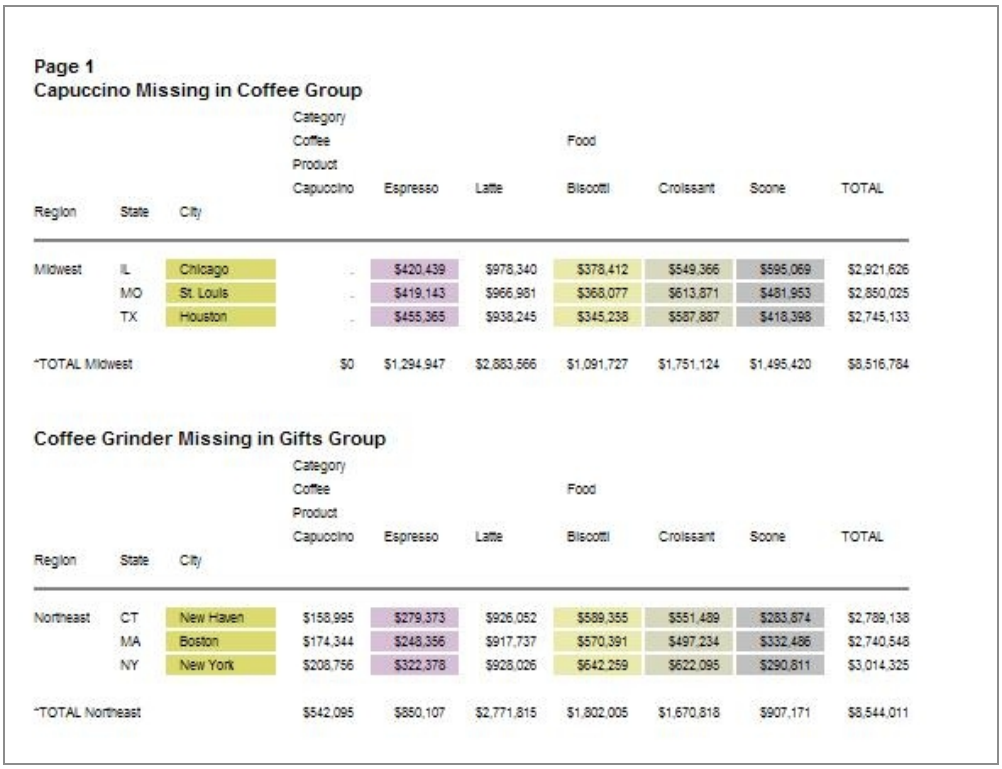
- B3 フィールドには、BACKCOLOR=GOLDENROD が割り当てられています。B3 は、表示しない BY フィールドを含めて、すべての BY フィールドを左から右へ数えた場合の 3 番目の BY フィールドです。B3 フィールドは、CITY ソートフィールドに対応します。
- SHOWDOLLARS(6) フィールドには、BACKCOLOR=SILVER が割り当てられています。SHOWDOLLARS(6) は、SHOWDOLLARS フィールドの 6 番目を表し、Food グループの Scone 列に対応します。

リクエストは次のようになります。

```
SET HIDENULLACRS=OFF
DEFINE FILE GGSALES
SHOWDOLLARS/I8M MISSING ON =
IF ((PRODUCT EQ 'Capuccino' AND REGION EQ 'Midwest') OR
(PRODUCT EQ 'Coffee Grinder' AND REGION EQ 'Northeast') OR
(PRODUCT EQ 'Scone' AND REGION EQ 'Southeast') OR
(REGION EQ 'West')) THEN MISSING ELSE DOLLARS;
END
TABLE FILE GGSALES
SUM SHOWDOLLARS AS ''
BY REGION
BY ST
BY CITY
ACROSS CATEGORY
ACROSS PRODUCT
ON REGION SUBTOTAL AS '*TOTAL'
ON REGION PAGE-BREAK
HEADING
" Page <TABPAGENO "HEADING
" Capuccino Missing in Coffee Group "
WHEN REGION EQ 'Midwest';
HEADING
" Coffee Grinder Missing in Gifts Group "
WHEN REGION EQ 'Northeast';
HEADING
" Scone Missing in Food Group "
WHEN REGION EQ 'Southeast';
WHERE CATEGORY EQ 'Coffee' OR 'Food'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET BYPANEL ON
ON TABLE ROW-TOTAL AS 'TOTAL'
ON TABLE COLUMN-TOTAL AS 'TOTAL'
```

```
ON TABLE SET HTMLCSS ON
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET STYLE *
UNITS=IN,PAGESIZE='Letter',SQUEEZE=ON,ORIENTATION=PORTRAIT,$
TYPE=REPORT,HEADPANEL=ON,GRID=OFF,FONT='ARIAL',SIZE=6,$
TYPE=HEADING, style=bold, size=8,$
TYPE=DATA, COLUMN = C5, BACKCOLOR=WHEAT,$
TYPE=DATA, COLUMN = P5, BACKCOLOR=THISTLE,$
TYPE=DATA, COLUMN = N7, BACKCOLOR=MEDIUM GOLDENROD,$
TYPE=DATA, COLUMN = B3, BACKCOLOR=GOLDENROD,$
TYPE=DATA, COLUMN = SHOWDOLLARS(6), BACKCOLOR=silver,$
ENDSTYLE
END
```

このレポートを SET HIDENULLACRS=OFF で実行すると、すべてのフィールドが表示されます。すべての値がミッシング値であるにも関わらず、West 地域のページが生成され、中間合計が計算されます。



Page 1
Capuccino Missing in Coffee Group

			Category	Coffee		Food			
			Product						
Region	State	City	Capuccino	Espresso	Latte	Biscotti	Croissant	Scone	TOTAL
Midwest	IL	Chicago	.	$420,439	$978,340	$378,412	$549,366	$595,069	$2,921,626
	MO	St. Louis	.	$419,143	$966,981	$368,077	$613,871	$481,953	$2,850,025
	TX	Houston	.	$455,365	$938,245	$345,238	$587,887	$418,398	$2,745,133
*TOTAL Midwest			$0	$1,294,947	$2,883,566	$1,091,727	$1,751,124	$1,495,420	$8,516,784

Coffee Grinder Missing in Gifts Group

			Category	Coffee		Food			
			Product						
Region	State	City	Capuccino	Espresso	Latte	Biscotti	Croissant	Scone	TOTAL
Northeast	CT	New Haven	$158,995	$279,373	$926,052	$589,355	$551,489	$283,874	$2,789,138
	MA	Boston	$174,344	$248,356	$917,737	$570,391	$497,234	$332,486	$2,740,548
	NY	New York	$208,756	$322,378	$928,026	$642,259	$622,095	$290,811	$3,014,325
*TOTAL Northeast			$542,095	$850,107	$2,771,815	$1,802,005	$1,670,818	$907,171	$8,544,011

Scone Missing in Food Group

Region	State	City	Category: Coffee / Product: Capuccino	Espresso	Latte	Food: Biscotti	Croissant	Scone	TOTAL
Southeast	FL	Orlando	$317,027	$256,539	$889,887	$511,597	$602,076	.	$2,577,126
	GA	Atlanta	$352,161	$317,389	$907,365	$555,231	$661,806	.	$2,793,952
	TN	Memphis	$274,812	$279,644	$820,584	$438,889	$638,477	.	$2,452,406
*TOTAL Southeast			$944,000	$853,572	$2,617,836	$1,505,717	$1,902,359	$0	$7,823,484

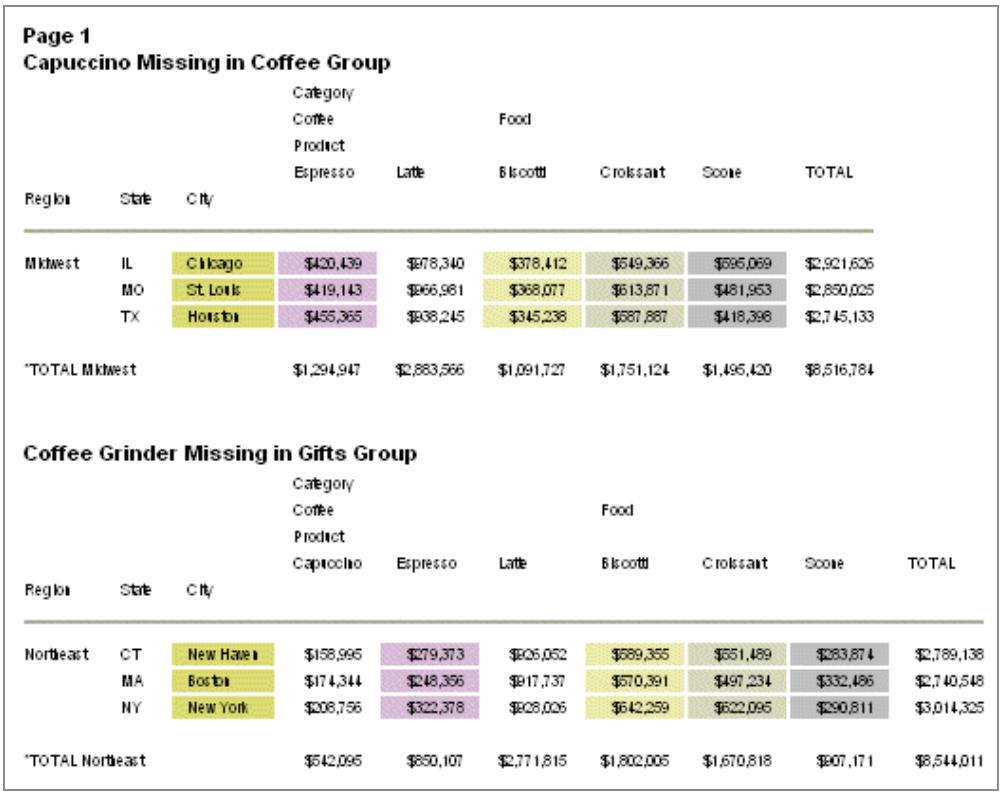
Region	State	City	Category: Coffee / Product: Capuccino	Espresso	Latte	Food: Biscotti	Croissant	Scone	TOTAL
West	CA	Los Angeles	.	.	.	.	.	.	$0
		San Francisco	.	.	.	.	.	.	$0
	WA	Seattle	.	.	.	.	.	.	$0
*TOTAL West			$0	$0	$0	$0	$0	$0	$0

Region	State	City	Category: Coffee / Product: Capuccino	Espresso	Latte	Food: Biscotti	Croissant	Scone	TOTAL
TOTAL			$1,486,095	$2,998,626	$8,273,217	$4,399,449	$5,324,301	$2,402,591	$24,884,279

このレポートを SET HIDENULLACRS=ON で実行すると、次の列が表示されます。

- 1 ページ目では、Capuccino 列が非表示になっているため、レポート上では Espresso フィールドは P5 ではなくなりましたが、この列にはまだ BACKCOLOR=THISTLE が割り当てられています。同様に、Biscotti 列には MEDIUM GOLDENROD、Croissant 列には WHEAT、Scone 列には SILVER が割り当てられています。
- 各地域の中間合計は、それぞれの地域で表示されている列のみを対象として計算されています。
- West 地域では、すべての値がミッシング値であるため、この地域のページは生成されていません。
- 総合計のページには、すべての列が表示されています。

出力結果は次のとおりです。



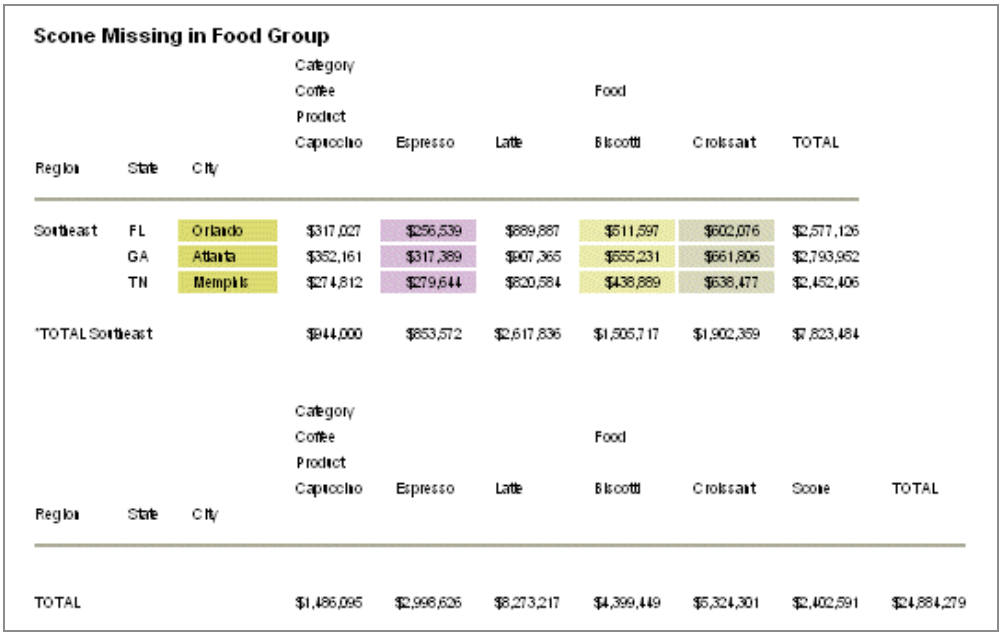
Page 1

Capuccino Missing in Coffee Group

			Category					
			Coffee		Food			
			Product					
Region	State	City	Espresso	Latte	Biscotti	Croissant	Scone	TOTAL
Midwest	IL	Chicago	$420,439	$978,340	$378,412	$549,366	$595,069	$2,921,626
	MO	St. Louis	$419,143	$966,961	$368,077	$613,871	$481,953	$2,850,025
	TX	Houston	$455,365	$938,245	$345,238	$587,887	$416,398	$2,745,133
*TOTAL Midwest			$1,294,947	$2,883,566	$1,091,727	$1,751,124	$1,495,420	$8,516,784

Coffee Grinder Missing in Gifts Group

			Category						
			Coffee			Food			
			Product						
Region	State	City	Capuccino	Espresso	Latte	Biscotti	Croissant	Scone	TOTAL
Northeast	CT	New Haven	$158,995	$279,373	$926,052	$589,355	$551,489	$283,874	$2,789,138
	MA	Boston	$174,344	$248,356	$917,737	$570,391	$497,234	$332,486	$2,740,548
	NY	New York	$208,756	$322,378	$928,026	$642,259	$622,095	$290,811	$3,014,325
*TOTAL Northeast			$542,095	$850,107	$2,771,815	$1,802,005	$1,670,818	$907,171	$8,544,011



Scone Missing in Food Group

			Category					
			Coffee			Food		
			Product					
Region	State	City	Capuccino	Espresso	Latte	Biscotti	Croissant	TOTAL
Southeast	FL	Orlando	$317,027	$256,539	$889,887	$511,597	$602,076	$2,577,126
	GA	Atlanta	$352,161	$317,389	$907,365	$555,231	$661,806	$2,793,952
	TN	Memphis	$274,812	$279,644	$820,584	$438,889	$638,477	$2,452,406
*TOTAL Southeast			$944,000	$853,572	$2,617,836	$1,505,717	$1,902,359	$7,823,484

			Category						
			Coffee			Food			
			Product						
Region	State	City	Capuccino	Espresso	Latte	Biscotti	Croissant	Scone	TOTAL
TOTAL			$1,486,095	$2,998,626	$8,273,217	$4,399,449	$5,324,301	$2,402,591	$24,884,279

FML リクエストでの ACROSS ブランク列の非表示

FML リクエストには、常に特定の行の順序を定義する FOR フィールドが含まれています。ACROSS のブランク列の非表示を開始するために FOR フィールドを使用することはできません。ただし、リクエストには BY フィールドを使用して PAGE-BREAK オプションを指定することができるため、その方法で ACROSS のブランク列を非表示にすることができます。

FML リクエストでの ACROSS ブランク列の非表示

次の FML リクエストは、GGSALES データソースに対して実行され、REGION という BY フィールドに改ページを指定し、QTR という ACROSS フィールドを指定しています。FOR フィールドは、PRODUCT です。DEFINE コマンドで一時項目 (DEFINE) の QTR フィールドが作成され、このフィールドの Midwest 地域の第 4 四半期 (Q4)、Northeast 地域の第 2 四半期 (Q2)、Southeast 地域のすべての四半期にミッシング値が存在します。

```
SET HIDENULLACRS=ON
DEFINE FILE GGSALES
QTR/Q=DATE;
SHOWDOLLARS/D12CM MISSING ON =
          IF REGION EQ 'Midwest' AND QTR EQ 'Q4' THEN MISSING
     ELSE IF REGION EQ 'Northeast' AND QTR EQ 'Q2' THEN MISSING
     ELSE IF REGION EQ 'Southeast' THEN MISSING
     ELSE DOLLARS;
END
TABLE FILE GGSALES
SUM SHOWDOLLARS
BY REGION
ACROSS QTR
FOR PRODUCT
'Biscotti' AS 'Biscotti' LABEL R1 OVER
'Capuccino' AS 'Capuccino' LABEL R2 OVER
'Latte' AS 'Latte' LABEL R3 OVER
'Mug' AS 'Mug' LABEL R4 OVER
'Coffee Pot' AS 'Coffee Pot' LABEL R5 OVER
RECAP R6/D12.2=R1+R2+R3+R4+R5;
 AS ''
ON REGION PAGE-BREAK
```

```
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
```

```
ON TABLE SET STYLE *
     UNITS=IN,
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,$
TYPE=REPORT,
     GRID=OFF,
     FONT='ARIAL',
     SIZE=9,$
TYPE=TITLE,
     STYLE=BOLD,$
TYPE=ACROSSTITLE,
     STYLE=BOLD,$
ENDSTYLE
END
```

このリクエストを SET HIDENULLACRS=OFF で実行すると、すべての列が生成されるとともに、すべての値がミッシング値である Southeast 地域を含めたすべての地域のページが生成されます。

Region		QTR Q1	Q2	Q3	Q4
Midwest	Biscotti	$205,440	$252,072	$275,976	.
	Capuccino	.	.	.	.
	Latte	$630,883	$704,089	$776,434	.
	Mug	$261,825	$264,003	$261,527	.
	Coffee Pot	$128,026	$169,533	$164,565	.
		1,226,174.00	1,389,697.00	1,478,502.00	.

Region		QTR Q1	Q2	Q3	Q4
Northeast	Biscotti	$393,362	.	$512,375	$453,794
	Capuccino	$62,070	.	$175,864	$185,712
	Latte	$791,332	.	$652,872	$628,026
	Mug	$265,133	.	$307,917	$308,783
	Coffee Pot	$130,517	.	$145,670	$163,594
		1,642,414.00	.	1,794,698.00	1,739,909.00

Region		QTR Q1	Q2	Q3	Q4
Southeast	Biscotti	.	.	.	.
	Capuccino	.	.	.	.
	Latte	.	.	.	.
	Mug	.	.	.	.
	Coffee Pot	.	.	.	.
		.	.	.	.

Region		QTR Q1	Q2	Q3	Q4
West	Biscotti	$229,781	$224,070	$230,649	$179,368
	Capuccino	$205,057	$194,149	$237,778	$258,511
	Latte	$727,734	$664,618	$619,832	$658,221
	Mug	$296,839	$281,809	$313,692	$296,324
	Coffee Pot	$132,354	$159,799	$153,072	$168,399
		1,591,765.00	1,524,445.00	1,555,023.00	1,560,823.00

このリクエストを SET HIDENULLACRS=ON で実行すると、Midwest 地域の第 4 四半期 (Q4)、Northeast 地域の第 2 四半期 (Q2)、Southeast 地域のページ全体が非表示になります。

		QTR		
		Q1	Q2	Q3
Region				
Midwest	Biscotti	$205,440	$252,072	$275,976
	Capuccino	.	.	.
	Latte	$630,883	$704,089	$776,434
	Mug	$261,825	$264,003	$261,527
	Coffee Pot	$128,026	$169,533	$164,565
		1,226,174.00	1,389,697.00	1,478,502.00

		QTR		
		Q1	Q3	Q4
Region				
Northeast	Biscotti	$393,362	$512,375	$453,794
	Capuccino	$62,070	$175,864	$185,712
	Latte	$791,332	$652,872	$628,026
	Mug	$265,133	$307,917	$308,783
	Coffee Pot	$130,517	$145,670	$163,594
		1,642,414.00	1,794,698.00	1,739,909.00

		QTR			
		Q1	Q2	Q3	Q4
Region					
West	Biscotti	$229,781	$224,070	$230,649	$179,368
	Capuccino	$205,057	$194,149	$237,778	$258,511
	Latte	$727,734	$664,618	$619,832	$658,221
	Mug	$296,839	$281,809	$313,692	$296,324
	Coffee Pot	$132,354	$159,799	$153,072	$168,399
		1,591,765.00	1,524,445.00	1,555,023.00	1,560,823.00

ソートフィールド値の表示制御

デフォルト設定では、ソートフィールド値が表示される行または列は、ソートフィールド値ごとに生成される一連の詳細フィールド値の先頭行または先頭列のみです。BYDISPLAY パラメータを使用して、この動作を制御することができます。BYDISPLAY はすべての出力フォーマットに適用され、BY ソート値だけでなく、ACROSS ソート値の表示も制御することができます。

この機能を使用すると、ソートフィールドを表示フィールド用とソート用として 2 回指定する必要がなくなります (NOPRINT オプションと併用)。以下はその例です。

```
PRINT FIRST_NAME LAST_NAME
BY FIRST_NAME NOPRINT
```

ソートフィールド値の表示制御

```
SET BYDISPLAY = {OFF|ON|BY|ACROSS|ALL}
```

```
ON TABLE SET BYDISPLAY {OFF|ON|BY|ACROSS|ALL}
```

説明

OFF

ソートフィールドの値を、ソートグループのレポート出力の 1 行目または 1 列目、および各ページの 1 行目または 1 列目のみに表示します。デフォルト値は OFF です。

ON または BY

生成したレポート出力で、関連する BY フィールドの値をすべての行に表示します。BY は ON の同義語です。

ACROSS

生成したレポート出力で、関連する ACROSS フィールドの値をすべての列に表示します。

ALL

関連する BY フィールドの値をレポート出力のすべての行に表示し、関連する ACROSS フィールドの値をレポート出力のすべての列に表示します。

レポート出力でのソートフィールド値の表示制御

次のリクエストは、レポートを生成し、変更された場合のみソートフィールド値を表示します (BYDISPLAY OFF)。

```
-SET &BYDISP = OFF;
SET BYDISPLAY = &BYDISP
TABLE FILE WF_RETAIL_LITE
HEADING CENTER
" BYDISPLAY = &BYDISP"
" "
SUM QUANTITY_SOLD DAYSDELAYED
BY PRODUCT_CATEGORY
BY PRODUCT_SUBCATEG
ACROSS BUSINESS_REGION
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。

BYDISPLAY = OFF

		Customer,Business,Region							
		EMEA		North America		Oceania		South America	
Product Category	Product Subcategory	Quantity Sold	Days Delayed	Quantity Sold	Days Delayed	Quantity Sold	Days Delayed	Quantity Sold	Days Delayed
Accessories	Charger	187	55	220	84	.	.	26	25
	Headphones	403	137	470	180	2	0	56	18
	Universal Remote Controls	287	102	325	132	7	7	51	23
Camcorder	Handheld	466	200	527	209	4	1	94	20
	Professional	23	4	35	17	.	.	.	.
	Standard	344	157	404	130	1	0	58	37
Computers	Smartphone	384	162	411	151	5	2	65	27
Media Player	Blu Ray	1,206	495	1,395	518	15	11	203	78
	DVD Players	32	14	48	17	.	.	13	0
	Streaming	88	30	124	53	3	0	13	2
Stereo Systems	Home Theater Systems	741	239	847	349	4	5	138	59
	Receivers	266	90	282	109	.	.	68	21
	Speaker Kits	451	172	476	199	.	.	95	33
	iPod Docking Station	529	186	673	272	7	0	102	62
Televisions	Flat Panel TV	159	67	166	62	1	0	36	15
Video Production	Video Editing	381	139	445	150	3	1	58	27

BYDISPLAY を ON または BY に変更すると、下図のように、すべての行に BY フィールドの値が表示されます。

BYDISPLAY = BY

		Customer,Business,Region							
		EMEA		North America		Oceania		South America	
Product Category	Product Subcategory	Quantity Sold	Days Delayed	Quantity Sold	Days Delayed	Quantity Sold	Days Delayed	Quantity Sold	Days Delayed
Accessories	Charger	187	55	220	84	.	.	26	25
Accessories	Headphones	403	137	470	180	2	0	56	18
Accessories	Universal Remote Controls	287	102	325	132	7	7	51	23
Camcorder	Handheld	466	200	527	209	4	1	94	20
Camcorder	Professional	23	4	35	17	.	.	.	.
Camcorder	Standard	344	157	404	130	1	0	58	37
Computers	Smartphone	384	162	411	151	5	2	65	27
Media Player	Blu Ray	1,206	495	1,395	518	15	11	203	78
Media Player	DVD Players	32	14	48	17	.	.	13	0
Media Player	Streaming	88	30	124	53	3	0	13	2
Stereo Systems	Home Theater Systems	741	239	847	349	4	5	138	59
Stereo Systems	Receivers	266	90	282	109	.	.	68	21
Stereo Systems	Speaker Kits	451	172	476	199	.	.	95	33
Stereo Systems	iPod Docking Station	529	186	673	272	7	0	102	62
Televisions	Flat Panel TV	159	67	166	62	1	0	36	15
Video Production	Video Editing	381	139	445	150	3	1	58	27

BYDISPLAY を ACROSS に変更すると、下図のように、すべての列に ACROSS フィールドの値が表示されます。

BYDISPLAY = ACROSS

		Customer,Business,Region							
		EMEA	EMEA	North America	North America	Oceania	Oceania	South America	South America
Product Category	Product Subcategory	Quantity Sold	Days Delayed	Quantity Sold	Days Delayed	Quantity Sold	Days Delayed	Quantity Sold	Days Delayed
Accessories	Charger	187	55	220	84	.	.	26	25
	Headphones	403	137	470	180	2	0	56	18
	Universal Remote Controls	287	102	325	132	7	7	51	23
Camcorder	Handheld	466	200	527	209	4	1	94	20
	Professional	23	4	35	17	.	.	.	.
	Standard	344	157	404	130	1	0	58	37
Computers	Smartphone	384	162	411	151	5	2	65	27
Media Player	Blu Ray	1,206	495	1,395	518	15	11	203	78
	DVD Players	32	14	48	17	.	.	13	0
	Streaming	88	30	124	53	3	0	13	2
Stereo Systems	Home Theater Systems	741	239	847	349	4	5	138	59
	Receivers	266	90	282	109	.	.	68	21
	Speaker Kits	451	172	476	199	.	.	95	33
	iPod Docking Station	529	186	673	272	7	0	102	62
Televisions	Flat Panel TV	159	67	166	62	1	0	36	15
Video Production	Video Editing	381	139	445	150	3	1	58	27

BYDISPLAY を ALL に変更すると、下図のように、すべての行に BY フィールドの値が表示され、すべての列に ACROSS フィールドの値が表示されます。

BYDISPLAY = ALL

		Customer,Business,Region							
Product Category	Product Subcategory	EMEA Quantity Sold	EMEA Days Delayed	North America Quantity Sold	North America Days Delayed	Oceania Quantity Sold	Oceania Days Delayed	South America Quantity Sold	South America Days Delayed
Accessories	Charger	187	55	220	84	.	.	26	25
Accessories	Headphones	403	137	470	180	2	0	56	18
Accessories	Universal Remote Controls	287	102	325	132	7	7	51	23
Camcorder	Handheld	466	200	527	209	4	1	94	20
Camcorder	Professional	23	4	35	17	.	.	.	.
Camcorder	Standard	344	157	404	130	1	0	58	37
Computers	Smartphone	384	162	411	151	5	2	65	27
Media Player	Blu Ray	1,206	495	1,395	518	15	11	203	78
Media Player	DVD Players	32	14	48	17	.	.	13	0
Media Player	Streaming	88	30	124	53	3	0	13	2
Stereo Systems	Home Theater Systems	741	239	847	349	4	5	138	59
Stereo Systems	Receivers	266	90	282	109	.	.	68	21
Stereo Systems	Speaker Kits	451	172	476	199	.	.	95	33
Stereo Systems	iPod Docking Station	529	186	673	272	7	0	102	62
Televisions	Flat Panel TV	159	67	166	62	1	0	36	15
Video Production	Video Editing	381	139	445	150	3	1	58	27

ソートフィールドのフォーマット再設定

レポート出力に BY ソート列または ACORSS ソート行を表示する場合、ソート句に新しいフォーマットを指定することで、ソートフィールド値のフォーマットを再設定することができます。フォーマットの再設定は、ソート列またはソート行に表示されるソートフィールド値のみに反映されます。つまり、ソートフィールドとして使用するフィールドが見出し、中間見出し、脚注、集計行で参照されている場合、そのフィールドは元のフォーマットで表示されます。

ソートフィールドのフォーマット再設定

```
{BY [TOTAL]|ACROSS [TOTAL]} sortfield/fmt ...
```

説明

sortfield

ソートフィールドです。

fmt

新しい表示フォーマットです。

ソートフィールドのフォーマット再設定時の注意

- フォーマットの再設定は、ON 句ではサポートされません。
- フォーマットの再設定は、ソート句で生成される行または列にのみ適用され、ソートフィールドを参照する中間見出し、中間脚注、集計行には適用されません。
- ソートフィールドのフィールドベースのフォーマット再設定 (フォーマットがフィールドに格納されている) はサポートされません。フィールドベースのフォーマット再設定についての詳細は、フィールド基準のフォーマット再設定 を参照してください。
- BY フィールドのフォーマット再設定は、HOLD ファイルに継承されます。ACROSS フィールドのフォーマット再設定は、HOLD ファイルに継承されません。
- 次の機能は、警告を発行し、フォーマットの再設定を無視した上で、リクエストの実行を継続します。
    - IN-GROUPS/RANGES-OF
    - ROWS
    - COLUMNS
    - FOR フィールド/フォーマット再設定
- すべての動作は元のフィールドに基づいて決定されるため、フォーマットが再設定されたソートフィールドに指定するオプションは、元のソートフィールドの区切り位置に配置する必要があります。その結果、ソートフィールド値が表示される位置に (例、SUBTOTALS)、元のフィールド値が常に表示されます。

ソートフィールドのフォーマット再設定

次のリクエストは、GGSALES データソースに対して実行され、以下のソートフィールドのフォーマットを再設定します。

- BY CATEGORY - CATEGORY のフォーマットを A3 に再設定します。
- BY PRODUCT - PRODUCT のフォーマットを A4 に再設定します。
- ACROSS REGION - REGION のフォーマットを A6 に再設定します。

```
TABLE FILE GGSALES
SUM UNIT
BY CATEGORY/A3
BY PRODUCT/A4
ACROSS REGION/A6
ON CATEGORY SUBTOTAL
ON CATEGORY SUBHEAD
"CATEGORY IS <CATEGORY "
" "
ON TABLE SET PAGE NOPAGE
END
```

出力結果では、フォーマットの再設定は BY および ACROSS 行に表示されますが、中間見出し行および中間合計行には継承されません。

```
                    Region
                      Midwes       Northe       Southe       West
CATEGORY  PRODUCT
-------------------------------------------------------------------------
CATEGORY IS Coffee

Cof       Capu              .        44785        73264        72831
          Espr         101154        68127        68030        71675
          Latt         231623       226243       211063       213920

*TOTAL CATEGORY Coffee
                       332777       339155       352357       358426

CATEGORY IS Food

Foo       Bisc          90413       149793       119594        70569
          Croi         139881       137394       156456       197022
          Scon         116127        70732        73779        72776
*TOTAL CATEGORY Food
                       346421       357919       349829       340367

CATEGORY IS Gifts

Gif       Coff          54002        40977        50556        48081
          Coff          47156        46185        49922        47432
          Mug           86718        91497        88474        93881
          Ther          46587        48870        48976        45648
*TOTAL CATEGORY Gifts
                       234463       227529       237928       235042
```

```
TOTAL                  913661       924603       940114       933835
```

ソートグループでの表示フィールド値の操作

WITHIN 句を使用して、ソートグループで表示フィールドの値を合計する際に、その値を操作することができます。この方法は、演算接頭語と併用して、レポートの列ではなく、特定の集計フィールドで計算を行う場合に使用します。これに対して、SUM コマンドおよび COUNT コマンドを使用した場合は、列全体の値が集計されます。

WITHIN 句には、BY 句または ACROSS 句を使用する必要があります。表示コマンド 1 つにつき、最大で 2 つの WITHIN 句を使用することができます。WITHIN 句を 1 つのみ使用する場合は、BY 句とともに実行します。WITHIN 句を 2 つ使用する場合は、1 つ目は BY 句と、2 つ目は ACROSS 句とともに実行します。

WITHIN TABLE コマンドを使用して、リクエストコマンド内で元の値を返すこともできます。WITHIN TABLE コマンドは、BY 句を伴わない ACROSS 句のみが必要な場合に使用することもできます。それ以外は、1 つの WITHIN 句に対して、1 つの BY 句が必要になります。

WITHIN 句による表示フィールドの操作

```
{SUM|COUNT} display_field WITHIN by_sort_field [WITHIN across_sort_field]
   BY by_sort_field [ACROSS across_sort_field]
```

説明

display_field

SUM または COUNT 表示コマンドのオブジェクトです。

by_sort_field

BY 句のオブジェクトです。

across_sort_field

ACROSS 句のオブジェクトです。

ソートグループでの値の集計

次のレポートは、STORE_CODE および PROD_CODE ごとの販売総数とその販売率を表示します。

```
TABLE FILE SALES
SUM UNIT_SOLD AS 'UNITS'
AND PCT.UNIT_SOLD AS 'PCT,SOLD,WITHIN,TABLE'
AND PCT.UNIT_SOLD WITHIN STORE_CODE AS 'PCT,SOLD,WITHIN,STORE'
BY STORE_CODE SKIP-LINE BY PROD_CODE
END
```

出力結果は次のとおりです。

STORE_CODE	PROD_CODE	UNITS	PCT SOLD WITHIN TABLE	PCT SOLD WITHIN STORE
14B	B10	60	9	15
	B12	40	6	10
	B17	29	4	7
	C13	25	3	6
	C7	45	6	11
	D12	27	4	7
	E2	80	12	21
	E3	70	10	18
14Z	B10	30	4	18
	B17	20	3	12
	B20	15	2	9
	C17	12	1	7
	D12	20	3	12
	E1	30	4	18
	E3	35	5	21
77F	B20	25	3	38
	C7	40	6	61
K1	B10	13	2	30
	B12	29	4	69

マトリックスレポートの作成

行と列の両方をソートして、マトリックスレポートを作成することができます。リクエストに BY 句と ACROSS 句の両方を使用すると、レポートのデータが縦と横にソートされ、格子上に配置されたようなデータマトリックスになります。マトリックスレポートには、複数の BY フィールドおよび複数の ACROSS フィールドを指定することができます。

単純なマトリックスの作成

次のリクエストは、部署別の給与合計を横方向に、ジョブコード別の給与合計を縦方向に表示するマトリックスレポートを作成します。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL
ACROSS DEPARTMENT
BY CURR_JOBCODE
END
```

出力結果は次のとおりです。

```
                  DEPARTMENT
                  MIS              PRODUCTION
CURR_JOBCODE
-------------------------------------------------
A01                         .        $9,500.00
A07                  $9,000.00       $11,000.00
A15                         .       $26,862.00
A17                 $27,062.00       $29,700.00
B02                 $18,480.00       $16,100.00
B03                 $18,480.00                .
B04                 $21,780.00       $21,120.00
B14                 $13,200.00                .
```

複数のソートフィールドによるマトリックスの作成

次のリクエストは、複数の BY フィールドおよび ACROSS フィールドを使用してマトリックスレポートを作成します。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL
ACROSS DEPARTMENT ACROSS LAST_NAME
BY CURR_JOBCODE BY ED_HRS
WHERE DEPARTMENT EQ 'MIS'
WHERE CURR_SAL GT 21500
END
```

出力結果は次のとおりです。

```
                           DEPARTMENT
                           MIS
                           LAST_NAME
                           BLACKWOOD            CROSS
CURR_JOBCODE  ED_HRS
----------------------------------------------------
A17            45.00                .       $27,062.00
B04            75.00       $21,780.00                .
```

照合順序の制御

「照合」とは、言語要素すべての順位付けと一致に適用される一連の規則として定義され、2つの値の比較を伴います。さまざまな要素がこの機能の影響を受けます。これらの要素には、ソート、集計、WHERE 条件、スタイルシートなどがあります。デフォルト設定では、項目はバイナリ値に基づいてソートされます。照合順序設定の SRV_CI および SRV_CS は、それぞれ大文字と小文字を区別しない照合および大文字と小文字を区別する照合であり、これらは言語設定に基づいて照合を実装します。大文字と小文字を区別しない照合では、すべての WHERE 句とソートで、比較対象要素の大文字と小文字が無視されます。照合順序はセッションレベルの設定で、ON TABLE 句ではサポートされていないため、edasprof サーバプロファイルで設定する必要があります。

照合設定は、文字値にのみ適用されます。

バイナリまたは大文字と小文字を区別しない照合順序の設定

edasprof.prf サーバプロファイルに、次のコマンドを追加します。

```
SET COLLATION = {BINARY|SRV_CI|SRV_CS|CODEPAGE}
```

説明

BINARY

バイナリ値に基づいて、照合順序を設定します。

SRV_CI

言語の設定に基づいて、照合順序を設定します。大文字と小文字は区別されません。

SRV_CS

言語の設定に基づいて、照合順序を設定します。大文字と小文字は区別されます。

CODEPAGE

有効なコードページに基づいて、照合順序を設定します。大文字と小文字は区別されます。デフォルト値は CODEPAGE です。

SET COLLATION 使用上の注意

- JOIN フィールドの文字データに大文字と小文字が混在する場合、1 つのモードで作成された SUFFIX=FIX ファイルおよび SUFFIX=FOCUS/XFOCUS HOLD ファイルを、その他のモードの JOIN のクロスリファレンスフィールドとして使用することはできません。
- FIXRETRIEVE はバイナリデータのみでサポートされるため、COLLATION を BINARY 以外に設定すると、FIXRETRIEVE が OFF になり、JOIN のパフォーマンスが低下する場合があります。

ソートと集計の規則

- 順序と文字が同一のレコードは、大文字と小文字が異なる場合でも、同一レコードと見なされます。複数の入力レコードにこれらのバリエーションが存在する場合、最初のレコードの値が使用されます。
- 詳細レベルのレポートでは、ソート値は各出力レコードで同一になります。この値は、最初に照合される、値が最小の入力レコードになります。
- ソート値として指定された値を含む複数の文字表示フィールドの最大値または最小値が、大文字と小文字を区別しない照合で同一と見なされるが、これら 2 つの値のいずれかの文字種が異なる場合、UMPREF=LST に設定した場合は、入力ファイルで最後の

もの (入力レコード番号が最大)、SUMPREF=FST の場合は最初のもの (レコード番号が最小) が保持されます。

ソート順の指定

ソートフィールドの値は、自動的に昇順にソートされます。つまり、値の小さい方から大きい方へとデータが表示されます。デフォルトのソート順は、オペレーティングシステムの種類により異なります。UNIX および Windows では、文字フィールドは 0 から 9、A から Z、a から z の順に、数値フィールドは 0 から 9 の順になります。

オプションとして、ソート句に HIGHEST を使用してこのデフォルト設定を無効にし、値を大きい方から小さい方へ降順にソートすることもできます。

ソート順の指定

```
{BY|ACROSS} {LOWEST|HIGHEST} sortfield
```

説明

LOWEST

値の小さい方から大きい方へと昇順にソートします (文字フィールドは a から z、A から Z、0 から 9 の順、数値フィールドは 0 から 9 の順)。このオプションがデフォルトです。

HIGHEST

値の大きい方から小さい方へと降順にソートします。HIGHEST の同義語として、TOP を代用することもできます。

sortfield

ソートフィールドの名前です。

昇順によるソート

次のリクエストではソート順が指定されていないため、デフォルト設定により給与額の値が小さい方から大きい方へとソートされます。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
BY CURR_SAL
END
```

ソート句に LOWEST を含めると、これと同じ昇順のソートを明示的に指定することができます。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
BY LOWEST CURR_SAL
END
```

出力結果は次のとおりです。

CURR_SAL	LAST_NAME
$9,000.00	GREENSPAN
$9,500.00	SMITH
$11,000.00	STEVENS
$13,200.00	SMITH
$16,100.00	MCKNIGHT
$18,480.00	JONES
	MCCOY
$21,120.00	ROMANS
$21,780.00	BLACKWOOD
$26,862.00	IRVING
$27,062.00	CROSS
$29,700.00	BANNING

降順によるソート

次のリクエストでは、給与額の値が大きい方から小さい方へとソートされます。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
BY HIGHEST CURR_SAL
END
```

出力結果は次のとおりです。

CURR_SAL	LAST_NAME
$29,700.00	BANNING
$27,062.00	CROSS
$26,862.00	IRVING
$21,780.00	BLACKWOOD
$21,120.00	ROMANS
$18,480.00	JONES
	MCCOY
$16,100.00	MCKNIGHT
$13,200.00	SMITH
$11,000.00	STEVENS
$9,500.00	SMITH
$9,000.00	GREENSPAN

ソート順のユーザ定義

ソートフィールドの値は、自動的に昇順にソートされます。つまり、値の小さい方から大きい方へとデータが表示されます。

デフォルトのソート順を無効にし、ユーザ定義のソート順で値を表示することができます。ユーザ定義のソート順でソートするには、次の手順を実行します。

1. 対象となるソートフィールドの値を選択します。この場合、ソートフィールドの値をすべて指定したり、一部の値だけを指定したりすることができます。リクエストを発行すると、ここで指定した値を含むレコードのみがレポートに出力されます。
2. 値の表示順序を指定します。ここでは、任意の順序を指定することができます。たとえば、1 文字コードを含む A1 ソートフィールドを、A、Z、B、C、Y... のようにソートするよう指定することも可能です。

ユーザ定義のソート順を指定するには、BY 句で行をソートする場合と ACROSS 句でフィールドをソートする場合の 2 通りの方法があります。

- BY ROWS OVER 句を使用して、行のソート順を定義します。
- ACROSS COLUMNS AND 句を使用して、列のソート順を定義します。

ソート順のユーザ定義

```
BY sortfield AS 'coltitle' ROWS value1 [AS 'text1']
OVER value2 [AS 'text2']
[...OVER valuen [ AS 'textn']]
END
```

説明

sortfield

レポート内の最後の BY フィールドです。

coltitle

レポート出力に表示する BY フィールドのフィールドタイトルです。

value1

ソート順内での先頭のソートフィールド値です。

AS 'text1'

先頭行の代替テキストを指定します。出力結果のフィールド値は、このテキストで置換されます。テキストは一重引用符 (') で囲みます。

value2

ソート順内での 2 番目のソートフィールド値です。

AS 'text2'

2 行目の行の代替テキストを指定します。出力結果のフィールド値は、このテキストで置換されます。テキストは一重引用符 (') で囲みます。

valuen

ソート順内での最後のソートフィールド値です。

AS 'textn'

最終行の代替テキストを指定します。出力結果のフィールド値は、このテキストで置換されます。テキストは一重引用符 (') で囲みます。

行を基にした FOR レポート句を構文の代用にすることもできます。

```
FOR sortfield
value1 OVER value2 [...OVER valuen]
```

詳細は、財務レポートの作成を参照してください。

ソート順を定義する際の注意

- ソートフィールドの値を BY ROWS OVER 句で指定しない場合、その値はソート順には含まれず、レポートに出力されません。
- ソートフィールドの値にブランクが含まれている場合は、値全体を一重引用符 (') で囲む必要があります。
- ソートフィールドの値を BY ROWS OVER 句で指定する場合、データの有無に関係なく、ソートフィールドの値がレポートに出力されます。
- レポートにミッシングデータが含まれている場合、そのデータは最下位のソートレベルで挿入する必要があります。
- ソートフィールドの名前はレポートに出力されません。
- 1 つのレポートに含めることのできる BY ROWS OVER 句は 1 つだけです。FOR 句では BY ROWS OVER はサポートされません。FOR 句についての詳細は、財務レポートの作成を参照してください。

行のソート順の定義

次の例は、給与の自動振込み先となる銀行別に従業員をソートするとともに、銀行フィールドのソート順も定義しています。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
BY BANK_NAME ROWS 'BEST BANK' OVER STATE
   OVER ASSOCIATED OVER 'BANK ASSOCIATION'
END
```

出力結果は次のとおりです。

	LAST_NAME
BEST BANK	BANNING
STATE	JONES
ASSOCIATED	IRVING
ASSOCIATED	BLACKWOOD
ASSOCIATED	MCKNIGHT
BANK ASSOCIATION	CROSS

フィールドソート順の定義

```
ACROSS sortfield COLUMNS value1 AND value2 [...AND valuen]
```

説明

sortfield

ソートフィールドの名前です。

value1

ソート順内での先頭のソートフィールド値です。

value2

ソート順内での 2 番目のソートフィールド値です。

valuen

ソート順内での最後のソートフィールド値です。

ACROSS COLUMNS 句

- ソートフィールドの値を ACROSS COLUMNS AND 句で指定しない場合、その値はソート順のラベルには含まれず、レポートに出力されません。
- ソートフィールドの値にブランクが入る場合は、それを一重引用符 (') で囲む必要があります。
- ソートフィールドの値を ACROSS COLUMNS AND 句で指定する場合、データの有無に関係なく、ソートフィールドのすべての値がレポートに出力されます。

- COMPUTE に ACROSS COLUMNS 句を使用する場合は、COLUMNS を最後に指定する必要があります。

```
ACROSS acrossfield [AND] COMPUTE compute_expression; COLUMNS values
```

- 各リクエストに含めることができる ACROSS COLUMNS AND 句は、1 つのみです。

フィールドソート順の定義

次の例は、給与の自動振込み先となる銀行別に従業員の給与を合計するとともに、銀行フィールドのソート順も定義しています。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL
ACROSS BANK_NAME COLUMNS 'BEST BANK' AND STATE
   AND ASSOCIATED AND 'BANK ASSOCIATION'
END
```

出力結果は次のとおりです。

```
BANK_NAME
BEST BANK          STATE              ASSOCIATED         BANK ASSOCIATION
-------------------------------------------------------------------------
     $29,700.00         $18,480.00         $64,742.00         $27,062.00
```

ACROSS 値の列タイトルの選択および適用

ACROSS COLUMNS 句を使用してレポート出力に表示する ACROSS ソートフィールドの列を選択、順序指定する場合、AS 句を使用することで、選択した列に新しい列タイトルを割り当てることができます。

ACROSS 値への列タイトルの割り当て

```
ACROSS sortfield [AS title]
 COLUMNS aval1 [AS val1title] [{AND|OR} aval2 [AS val2title] [... {AND|OR}
avaln [AS valntitle]]]
```

説明

sortfield

ACROSS フィールド名です。

title

ACROSS フィールド名のタイトルです。

AND|OR

選択した ACROSS 値を区切るために必要です。この用途では、AND と OR は同義語です。

aval1, aval2, ... avaln

レポート出力に表示する ACROSS 値を選択します。

val1title, val2title ...valntitle

選択した ACROSS 値に割り当てる列タイトルです。

ACROSS 値に列タイトルを割り当てる際の注意

- ソート句で ACROSS 値として新しい値を指定すると、その値が IF または WHERE テストで除外された場合やデータソースに実際に存在しない場合でも、新しい値がレポート出力に表示されます。

  注意：スタイル出力フォーマットで SET HIDENULLACRS=ON を使用すると、レポート出力の ACROSS グループからブランクの列が削除されます。

- ACROSS フィールドの列タイトルは、レポート出力の単一行に表示されます。

- ACROSS 値の AS 名は、TABLE FILE コマンドでのみサポートされます。
- SET ASNAMES=ON を使用して HOLD ファイルを作成した場合、出力されるマスターファイルには、AS 名ではなく、元のフィールド名が継承されます。

ACROSS 値の列タイトルの選択および適用

次のリクエストは、GGSALES データソースに対して実行され、ACROSS フィールドとして指定した PRODUCT の Coffee Grinder、Latte、Coffee Pot を選択し、これらの値のそれぞれに新しい列タイトルを割り当てます。

```
 TABLE FILE GGSALES
 SUM
 DOLLARS/I8M AS ''
 BY REGION
 ACROSS PRODUCT  AS 'Products'
   COLUMNS 'Coffee Grinder' AS 'Grinder'
    OR Latte AS 'caffellatte'
    AND 'Coffee Pot' AS 'Carafe'
ON TABLE SET PAGE NOPAGE
END
```

出力結果は次のとおりです。

```
             Products
             Grinder          caffellatte      Carafe
Region
----------------------------------------------------------------
Midwest         $666,622       $2,883,566         $599,878
Northeast       $509,200       $2,808,855         $590,780
Southeast       $656,957       $2,637,562         $645,303
West            $603,436       $2,670,405         $613,624
```

ソートフィールド値の順位付け

BY 句を使用してレポートの行をソートする際に、それぞれの行に順位を付けることができます。ソートフィールド値の順位付けを行う機能は、その順位に基づいて表示するデータを限定する機能と組み合わせてよく使用されます。

ソートフィールドに同一の値が存在する場合は、同じ順位に複数のレポート行が表示される場合があるので留意してください。

RANKED BY に対するデフォルトのフィールドタイトルは RANK です。タイトルを変更するには、AS 句を使用します。RANK フィールドには I7 フォーマットが指定されています。そのため、レポートの RANK 列には最大で 7 桁を使用することができます。詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。

集計値に順位を付けるには、RANKED BY TOTAL 構文を使用します。詳細は、レポートフィールドのソートおよび集計を参照してください。

ソートフィールド値の順位付け

```
RANKED [AS 'name'] BY  {HIGHEST|LOWEST} [n]  sortfield [AS 'text']
```

説明

name

RANK フィールドタイトルの新しい名前です。

sortfield

ソートフィールドの名前です。フィールドには、数値または文字を使用することができます。

n

レポート出力に表示する順位付けカテゴリの数です。

text

レポート出力のソートフィールド列に使用する列見出しです。

ソートフィールド値の順位付け

次のリクエストは、給与額の順に従業員の名前をソートして表示するとともに、その給与額に基づいて順位を付けます。ここでは、従業員の JONES と MCCOY の現在の給与額が等しいため、同一の順位が付けられています。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
```

```
RANKED AS 'Sequence' BY CURR_SAL
END
```

出力結果は次のとおりです。

Sequence	CURR_SAL	LAST_NAME
1	$9,000.00	GREENSPAN
2	$9,500.00	SMITH
3	$11,000.00	STEVENS
4	$13,200.00	SMITH
5	$16,100.00	MCKNIGHT
6	$18,480.00	JONES
		MCCOY
7	$21,120.00	ROMANS
8	$21,780.00	BLACKWOOD
9	$26,862.00	IRVING
10	$27,062.00	CROSS
11	$29,700.00	BANNING

ソートフィールド値の順位付けと表示データの限定

次の例のように、ソートフィールド値に順位を付ける機能は、その順位に基づいて表示するデータを限定する機能と組み合わせてよく使用されます。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
RANKED BY HIGHEST 5 CURR_SAL
END
```

出力結果は次のとおりです。

```
RANK         CURR_SAL  LAST_NAME
----         --------  ---------
   1       $29,700.00  BANNING
   2       $27,062.00  CROSS
   3       $26,862.00  IRVING
   4       $21,780.00  BLACKWOOD
   5       $21,120.00  ROMANS
```

DENSE および SPARSE の順位付け

WebFOCUS ソート句の RANK BY および BY {HIGHEST|LOWEST} **n** は、レポート出力をソートし、一連のデータ値に順位を割り当てます。デフォルト設定では、WebFOCUS がデータ値に順位を割り当てる場合、順位を飛ばさずに番号が付けられます。つまり、複数のデータ値に同一順位が割り当てられた場合でも、次の値グループには次の整数値が連番で割り当てられます。この順位付け方法は、「DENSE」(連番) と呼ばれます。

一部のリレーショナルエンジンでは、「SPARSE」(非連番) と呼ばれる方法で順位が割り当てられます。SPARSE による順位付け方法では、複数のデータ値に同一順位が割り当てられた場合、次の順位は、前の順位に複数のデータ値の個数を加算した整数値になります。

WebFOCUS の RANK パラメータを使用して、WebFOCUS で使用する順位付けタイプを制御することができます。また、リレーショナルデータソースにアクセスする場合は、順位付け方法を、リレーショナルエンジンで使用されている順位付けタイプに設定することで順位計算を最適化することができます。一部のリレーショナルエンジンは、DENSE と SPARSE の両方の順位付け機能を備えています。この場合、いずれかの設定を最適化することができます。

順位付けの最適化

順位付け処理をリレーショナルエンジンに渡すには、リクエストで次のことを行う必要があります。

- SUM (または WRITE、ADD) コマンドを使用して値を集計する。
- 表示する順位カテゴリの個数を指定する。つまり、n の値を指定する必要があります。

```
[RANKED] BY [HIGHEST] n
```

順位付け方法の制御

```
SET RANK={DENSE|SPARSE}
```

説明

DENSE

連番による順位付け方法です。この方法では、複数のデータ値に同一順位が割り当てられた場合でも、次の順位には次の整数値が連番で割り当てられます。デフォルト値は DENSE です。

SPARSE

非連番による順位付け方法です。この方法では、複数のデータ値に同一順位が割り当てられた場合、次の順位は、前の順位に複数のデータ値の個数を加算した整数値になります。

次に、リクエストで BY 句の形式として次のいずれかを使用します。

```
RANKED BY {HIGHEST|LOWEST} [n] sortfield [AS 'text']
```

または

```
BY {HIGHEST|LOWEST} n sortfield [AS 'text']
```

説明

n

RANKED BY 句を使用した場合にレポート出力に表示する最上位の順位です。RANKED を使用しない場合、SET RANK=DENSE 設定ではレポート出力に表示するソートフィールド値の種類数、SET RANK=SPARSE 設定ではソートフィールドの出力行の合計を指定します。

sortfield

ソートフィールドの名前です。

text

レポート出力のソートフィールド列に使用する列見出しです。

SET RANK 使用時の注意

- RNK. 演算接頭語は、RANK パラメータに影響されません。
- 順位は、RANK パラメータの設定に基づいて HOLD ファイルに継承されます。

FOCUS データソースの値の順位付け

次のリクエストは、EMPDATA データソースに対して実行され、部門別の給与を降順で順位付けします。RANK パラメータは DENSE (デフォルト) に設定されています。

```
SET RANK = DENSE
TABLE FILE EMPDATA
PRINT LASTNAME FIRSTNAME
RANKED BY HIGHEST 12 SALARY
BY DIV
ON TABLE SET PAGE NOPAGE
END
```

出力結果では、第 6 位に 6 名の従業員が含まれています。順位付けが DENSE 場合、次の順位は第 7 位 (連番の次の整数値) になります。

```
RANK            SALARY  DIV   LASTNAME         FIRSTNAME
----            ------  ---   --------         ---------
   1       $115,000.00  CE    LASTRA           KAREN
   2        $83,000.00  CORP  SANCHEZ          EVELYN
   3        $80,500.00  SE    NOZAWA           JIM
   4        $79,000.00  CORP  SOPENA           BEN
   5        $70,000.00  WE    CASSANOVA        LOIS
   6        $62,500.00  CE    ADAMS            RUTH
                        CORP  CVEK             MARCUS
                              WANG             JOHN
                        NE    WHITE            VERONICA
                        SE    BELLA            MICHAEL
                              HIRSCHMAN        ROSE
   7        $58,800.00  WE    GOTLIEB          CHRIS
   8        $55,500.00  CORP  VALINO           DANIEL
                        NE    PATEL            DORINA
   9        $54,100.00  CE    ADDAMS           PETER
                        WE    FERNSTEIN        ERWIN
  10        $52,000.00  NE    LIEBER           JEFF
  11        $50,500.00  SE    LEWIS            CASSANDRA
  12        $49,500.00  CE    ROSENTHAL        KATRINA
                        SE    WANG             KATE
```

SET RANK=SPARSE を使用して同一のリクエストを実行すると、次の出力結果が生成されます。第 6 位には 6 名の従業員が含まれているため、次の順位は第 12 位 (6 + 6) になります。

```
RANK           SALARY  DIV   LASTNAME         FIRSTNAME
----           ------  ---   --------         ---------
   1      $115,000.00  CE    LASTRA           KAREN
   2       $83,000.00  CORP  SANCHEZ          EVELYN
   3       $80,500.00  SE    NOZAWA           JIM
   4       $79,000.00  CORP  SOPENA           BEN
   5       $70,000.00  WE    CASSANOVA        LOIS
   6       $62,500.00  CE    ADAMS            RUTH
                       CORP  CVEK             MARCUS
                             WANG             JOHN
                       NE    WHITE            VERONICA
                       SE    BELLA            MICHAEL
                             HIRSCHMAN        ROSE
  12       $58,800.00  WE    GOTLIEB          CHRIS
```

ソートフィールド値の個数の制限

次のリクエストは、EMPDATA データソースに対して実行され、部門別の給与を降順でソートし、上位 12 位までの給与を表示します。RANK パラメータは DENSE (デフォルト) に設定されています。

```
SET RANK = DENSE
TABLE FILE EMPDATA
PRINT LASTNAME FIRSTNAME
BY HIGHEST 12 SALARY
BY DIV
ON TABLE SET PAGE NOPAGE
END
```

出力結果では、複数の従業員の給与が同一の場合でも、異なる 12 個の給与値が表示されます。

```
     SALARY  DIV   LASTNAME         FIRSTNAME
     ------  ---   --------         ---------
$115,000.00  CE    LASTRA           KAREN
 $83,000.00  CORP  SANCHEZ          EVELYN
 $80,500.00  SE    NOZAWA           JIM
 $79,000.00  CORP  SOPENA           BEN
 $70,000.00  WE    CASSANOVA        LOIS
 $62,500.00  CE    ADAMS            RUTH
             CORP  CVEK             MARCUS
                   WANG             JOHN
```

```
             NE    WHITE           VERONICA
             SE    BELLA           MICHAEL
                   HIRSCHMAN       ROSE
 $58,800.00  WE    GOTLIEB         CHRIS
 $55,500.00  CORP  VALINO          DANIEL
             NE    PATEL           DORINA
 $54,100.00  CE    ADDAMS          PETER
             WE    FERNSTEIN       ERWIN
 $52,000.00  NE    LIEBER          JEFF
 $50,500.00  SE    LEWIS           CASSANDRA
 $49,500.00  CE    ROSENTHAL       KATRINA
             SE    WANG            KATE
```

SET RANK=SPARSE を使用して同一のリクエストを実行すると、次の出力結果が生成されます。出力結果では、従業員 6 名の給与が $62,500 のため、その値が 6 回カウントされ、12 行のみが表示されます (種類の異なる値は 7 個)。

```
      SALARY  DIV   LASTNAME        FIRSTNAME
      ------  ---   --------        ---------
$115,000.00  CE    LASTRA          KAREN
 $83,000.00  CORP  SANCHEZ         EVELYN
 $80,500.00  SE    NOZAWA          JIM
 $79,000.00  CORP  SOPENA          BEN
 $70,000.00  WE    CASSANOVA       LOIS
 $62,500.00  CE    ADAMS           RUTH
             CORP  CVEK            MARCUS
                   WANG            JOHN
             NE    WHITE           VERONICA
             SE    BELLA           MICHAEL
                   HIRSCHMAN       ROSE
 $58,800.00  WE    GOTLIEB         CHRIS
```

範囲指定による数値データのグループ化

数値ソートフィールドを使用してレポートをソートする場合、ソートフィールドの値をグループ化し、各グループの範囲を定義することができます。

グループを定義するにはいくつかの方法があります。定義できるグループには次のものがあります。

- IN-GROUPS-OF 句を使用して均等な範囲を指定する。

  1 つのリクエストには、IN-GROUPS-OF 句と IN-RANGES-OF 句を全部で 5 つ含めるこ

とができます。IN-GROUPS-OF 句は、1 つの BY フィールドに対して 1 回だけ使用することができます。最初のソートフィールド範囲は、IN-GROUPS-OF の対象となる値の中で最小の値から開始され、ここで表示された値がそれぞれの範囲の起点となります。

- IN-RANGES-OF 句を使用して均等な範囲を指定する。

  1 つのリクエストには、IN-GROUPS-OF 句と IN-RANGES-OF 句を全部で 5 つ含めることができます。IN-GROUPS-OF 句は、1 つの BY フィールドに対して 1 回だけ使用することができます。この句により、ソート句の総数に含める必要のある追加の内部ソート句が生成されます。最初のソートフィールド範囲は、IN-RANGES-OF の対象となる値の中で最小の値から始まります。範囲を 0 (ゼロ) に指定した場合はメッセージは生成されませんが、レポートに表示される値は予測できません。

- FOR 句を使用して不均等な範囲を指定する。
- 分位数を使用する。この方法には、100 分位数、4 分位数、10 分位数などがあります。詳細は、数値データのグループ配分表示を参照してください。

FOR 句は、FML (Financial Modeling Language) の一部で、通常はマトリックスレポートの作成に使用します。また、FOR 句を使用して、ソートフィールドの値を不均等な範囲でグループ化した表形式レポートを作成することもできます。

FOR 句は、個々の行に対してソート値を表示します。複数の範囲を定義する場合、それらの範囲を連続させる必要はありません。つまり、範囲と範囲の間に間隔を空けて定義することができます。FOR 句についての詳細は、財務レポートの作成を参照してください。

注意：グループにデータが存在しない場合でも、そのグループの行がレポートに表示されます。

均等範囲によるグループの定義

```
{BY|ACROSS} sortfield IN-GROUPS-OF value [TOP limit]
```

説明

sortfield

ソートフィールドの名前です。ソートフィールドは、数値フィールドでなければなりません。該当するフォーマットには、I (整数)、F (浮動小数点数)、D (倍精度浮動小数点数)、P (パック 10 進数) があります。

value

ソートフィールドの値をグループ化する範囲を示す正の整数です。

limit

オプションとして指定する値で、最上位グループを示すラベルを定義し、レポートに表示します。

均等範囲によるグループの定義

次の例は、各給与範囲に属する従業員の名前を表示するとともに、給与範囲を $5,000 単位で定義する方法を示しています。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
BY CURR_SAL IN-GROUPS-OF 5000
END
```

出力結果は次のとおりです。

```
  CURR_SAL  LAST_NAME
  --------  ---------
 $5,000.00  SMITH
            GREENSPAN
$10,000.00  STEVENS
            SMITH
$15,000.00  JONES
            MCCOY
            MCKNIGHT
$20,000.00  ROMANS
            BLACKWOOD
$25,000.00  BANNING
            IRVING
            CROSS
```

均等範囲の定義

```
{BY|ACROSS} sortfield IN-RANGES-OF value [TOP limit]
```

説明

sortfield

ソートフィールドの名前です。ソートフィールドは、数値フィールドでなければなりません。該当するフォーマットには、I (整数)、F (浮動小数点数)、D (倍精度浮動小数点数)、P (パック 10 進数) があります。

value

0 (ゼロ) より大きい整数で、ソートフィールド値をグループ化する範囲を表します。

limit

オプションとして指定する値で、最上位の範囲を示すラベルを定義し、レポートに表示します。この値より上位にあるすべてのデータ値が対象範囲になります。

注意：IN-GROUPS-OF 句により、ソート句の総数に含める必要のある追加の内部ソート句が生成されます。

均等範囲の定義

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
BY CURR_SAL IN-RANGES-OF 5000
END
```

出力結果は次のとおりです。

CURR SAL	LAST NAME
$5,000.00 - $9,999.99	SMITH
	GREENSPAN
$10,000.00 - $14,999.99	STEVENS
	SMITH
$15,000.00 - $19,999.99	JONES
	MCCOY
	MCKNIGHT
$20,000.00 - $24,999.99	ROMANS
	BLACKWOOD
$25,000.00 - $29,999.99	BANNING
	IRVING
	CROSS

データ値のカスタムグループの定義

```
FOR sortfield
begin1 TO end1 [OVER begin2 TO end2 ... ]
```

説明

sortfield

ソートフィールドの名前です。

begin

範囲の開始点を識別する値です。

end

範囲の終了点を識別する値です。

データ値のカスタムグループの定義

次のリクエストは、従業員の給与を表示しますが、任意の基準でグループ化されています。ここでは、各範囲の先頭の値がレポートに出力されます。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
FOR CURR_SAL
9000 TO 13500 OVER
14000 TO 19700 OVER
19800 TO 30000
END
```

出力結果は次のとおりです。

	LAST_NAME
9000	STEVENS
9000	SMITH
9000	SMITH
9000	GREENSPAN
14000	JONES
14000	MCCOY
14000	MCKNIGHT
19800	BANNING
19800	IRVING
19800	ROMANS
19800	BLACKWOOD
19800	CROSS

数値データのグループ配分表示

表形式のレポートでは、数値データを 100 分位数、10 分位数、4 分位数などのグループに配分して表示することができます。たとえば、生徒のテスト成績を 10 分位数にグループ化して、どの生徒がクラスの上位 10 パーセントに入るかを割り出したり、総売上高に基づいて、どの営業担当者が全体の上位半分に入るかを特定したりすることができます。

グループ化は、選択した BY フィールドの値に基づいて行われ、指定したグループ数にできるだけ均等にデータが配分されます。

データをグループに配分して表示する場合、次のようなことが実行されます。

- デフォルト設定で「TILE」というラベルの付いた新しいフィールドが、レポートに追

加され、このフィールドの各インスタンスに割り当てられた分位数が表示されます。フィールド見出しを変更するには、AS 句を使用します。AS 句についての詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。

- データを配分する計算は、リクエストの中で上位にあるすべてのソートフィールドで実行され、配分して表示するフィールド値より上位にあるソートフィールドが変更されるたびに再実行されます。
- インスタンスの個数は、TILE フィールドを基に集計されます。リクエストが下位に位置するセグメントのフィールドを表示する場合、TILE フィールドの 1 つのインスタンスに対して、レポートの行が複数存在する場合があります。
- TILE フィールドの値が等しいインスタンスは、同一のグループに配分されます。たとえば、3 つのグループに配分される次のデータについて考察します。

  1 5 5 5 8 9

  この場合、レコード数が等しくなるようインスタンスを配分してグループ化すると、次のような結果になります。

グループ	データ値
1	1,5
2	5,5
3	8,9

ここでは、等しいデータ値はすべて同一グループに入れる必要があるため、グループ 2 の 5 は、グループ 1 に移動され、グループ 2 はブランクになります。最終的に、次のようにグループ化されます。

分位番号	データ値
1	1,5,5,5
2	
3	8,9

数値データのグループ配分表示

```
BY [ {HIGHEST|LOWEST} [k] ] tilefield [AS 'head1']
          IN-GROUPS-OF n TILES [TOP m] [AS 'head2']
```

説明

HIGHEST

データを降順でソートします。これにより、データの最大値が TILE 1 に配置されます。

LOWEST

データを昇順でソートします。これにより、データの最小値が TILE 1 に配置されます。これがデフォルトのソート順です。

k

レポートに配分して表示するグループ数を表す正の整数です。たとえば、BY HIGHEST 2 は、データが存在し、最大値を持つ 2 つのグループを表示します。

tilefield

値が TILE 番号の割り当てに使用されるフィールドです。

head1

TILE ソートフィールドの値を表示するフィールドの見出しです。

n

データを配分するグループ数を指定する 32,767 以下の正の整数です。たとえば、100 を選択すると 100 分位数、10 を選択すると 10 分位数、というようになります。

m

レポートに表示するグループ数の最大値を表す正の整数です。たとえば、TOP 3 と指定した場合、割り当てられたグループ番号が 3 より大きいデータ行は表示されません。

head2

グループ数を表示するフィールドの新しい見出しです。

注意

- この構文は、k、n、m に整数以外の数字を受け入れます。UNIX および Windows で

は、小数点を含む値は切り捨てられます。マイナスまたは 0 (ゼロ) の数字を使用すると、エラーメッセージが表示されます。

- k と m を併用すると、レポートの各ソート区切り間に表示する行数が制限されます。両方を指定した場合、より厳密な値で表示が制御されます。k と m の両方がグループ数の n より大きい場合は、n が使用されます。

データの 5 グループ配分表示

次の例は、データを 5 つのグループに配分して表示する方法を示しています。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME
BY DEPARTMENT
BY CURR_SAL IN-GROUPS-OF 5 TILES
END
```

出力結果は次のとおりです。

DEPARTMENT	CURR_SAL	TILE	LAST_NAME	FIRST_NAME
MIS	$9,000.00	1	GREENSPAN	MARY
	$13,200.00	1	SMITH	MARY
	$18,480.00	2	JONES	DIANE
			MCCOY	JOHN
	$21,780.00	4	BLACKWOOD	ROSEMARIE
	$27,062.00	5	CROSS	BARBARA
PRODUCTION	$9,500.00	1	SMITH	RICHARD
	$11,000.00	1	STEVENS	ALFRED
	$16,100.00	2	MCKNIGHT	ROGER
	$21,120.00	3	ROMANS	ANTHONY
	$26,862.00	4	IRVING	JOAN
	$29,700.00	5	BANNING	JOHN

ここでは、上位に位置する DEPARTMENT ソートフィールドでデータがグループ化されています。MIS カテゴリには、TILE 3 に属するデータはありません。PRODUCTION カテゴリには、5 つすべての TILE にデータが存在します。

先頭 3 グループの配分表示

次の例では、給与額の低い方から上位 3 名の従業員が 5 つのグループに配分されています。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME
BY DEPARTMENT
BY LOWEST 3 CURR_SAL IN-GROUPS-OF 5 TILES
END
```

出力結果は次のとおりです。

DEPARTMENT	CURR_SAL	TILE	LAST_NAME	FIRST_NAME
MIS	$9,000.00	1	GREENSPAN	MARY
	$13,200.00	1	SMITH	MARY
	$18,480.00	2	JONES	DIANE
			MCCOY	JOHN
	$21,780.00	4	BLACKWOOD	ROSEMARIE
PRODUCTION	$9,500.00	1	SMITH	RICHARD
	$11,000.00	1	STEVENS	ALFRED
	$16,100.00	2	MCKNIGHT	ROGER
	$21,120.00	3	ROMANS	ANTHONY

このリクエストでは、各カテゴリで 3 つのグループが表示されています。MIS カテゴリには TILE 3 に属するデータがないため、このカテゴリには TILE 1、2、4 の 3 つが表示されています。

上位 3 グループの配分表示

次の例では、給与額の低い方から上位 3 名の従業員が 5 つ のグループに配分されています。ただし、配分したグループの上位 3 つ (Tile 1、2、3) に属するデータのみを表示します。また、AS 句を使用して TILES フィールドの見出しを DECILES に変更します。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME
BY DEPARTMENT
BY LOWEST 3 CURR_SAL IN-GROUPS-OF 5 TILES TOP 3 AS DECILES
END
```

出力結果は次のとおりです。

DEPARTMENT	CURR_SAL	DECILE	LAST_NAME	FIRST_NAME
MIS	$9,000.00	1	GREENSPAN	MARY
	$13,200.00	1	SMITH	MARY
	$18,480.00	2	JONES	DIANE
			MCCOY	JOHN
PRODUCTION	$9,500.00	1	SMITH	RICHARD
	$11,000.00	1	STEVENS	ALFRED
	$16,100.00	2	MCKNIGHT	ROGER
	$21,120.00	3	ROMANS	ANTHONY

MIS カテゴリには TILE 3 に属するデータがないため、このカテゴリには TILE 1 と 2 のみが表示されています。

データをグループに配分して表示する際の注意

- リクエストがセグメントからデータを取得し、そのセグメントが TILE フィールドを含むセグメントより下位にある場合は、TILE フィールドの 1 つのインスタンスに対して複数のレポート行が存在する場合があります。これら余分なレポート行は、データをグループに配分する際に使用するインスタンス数に影響は与えません。ただし、複数のセグメントからフィールドを取得して、単一セグメントの出力ファイルを作成する場合は、この Flat File には TILE フィールドのインスタンスが複数存在することになるため、このインスタンス数の増加分が割り当てられたグループの値に影響を与える場合があります。そのため、マルチレベルファイルおよび単一セグメントファイルに対して同一リクエストを実行すると、結果としてグループの割り当てが異なることもあります。
- データをグループに配分する場合、常にリクエストの BY フィールドで計算されます。
- 配分するグループの計算は、各リクエストにつき 1 回のみです。ただし、リクエストには TILES 句に加えて、最大で 5 つの TILE 以外の IN-GROUP-OF 句を含めることができます。
- TILE 番号を割り当てる際の値の比較には、その値の表示フォーマットに関係なく、正確なデータ値が使用されます。そのため、浮動小数点数の値を D7 で表示した場合、有効桁数が十分に表示されないため、それぞれの値がどのように各グループに配分されたかが分かりづらい場合があります。
- TILE フィールドには、DEFINIE コマンドまたはマスターファイルの DEFINIE で作成された実フィールドまたは一時項目 (DEFINE) のいずれかを指定することができます。

COMPUTE コマンドを使用して、TILE フィールドを作成することはできません。

- 配分されたデータがないグループは、レポート出力に表示されません。
- 複数のソートフィールドを含むリクエストでは、グループに配分して表示する機能は最下位で実行され、BY LOWEST 句と併用される場合にのみサポートされます。
- グループに配分して表示する機能は、出力ファイルでサポートされます。ただし、HOLDLIST を PRINTONLY に設定しない限り、グループに配分する計算に使用するフィールドは、3 つのフィールドを HOLD ファイルに継承します (実フィールドの値、TILE、順位フィールド)。
- グループに配分して表示する機能は、BY TOTAL、TABLEF、FML、GRAPH ではサポートされていません。

ソートフィールド値の順位による限定

BY 句を使用してレポートの行をソートする場合、ソートフィールドの値を順位の上位または下位に属するグループに限定することができます。レポートに含めるフィールド数を限定することができます。たとえば、BY HIGHEST (または LOWEST) を使用して、上位 (または下位) 10 個のソートフィールドの値のみを表示するように選択することができます。

BY HIGHEST または BY LOWEST は、最大で 5 つのソートフィールドに使用することができます。

ソートフィールド値の順位による限定

```
BY {HIGHEST n|LOWEST n} sortfield
```

説明

HIGHEST n

ソートフィールドの値の中で、上位 n 個のみがレポートに表示されます。TOP は HIGHEST の同義語です。

LOWEST n

ソートフィールドの値の中で、下位 n 個のみがレポートに表示されます。

sortfield

ソートフィールドの名前です。ソートフィールドには、数値または文字を使用することができます。

注意：HIGHEST n または LOWEST n は、レポートの行数ではなく、ソートフィールドの値の個数を表しています。HIGHEST n または LOWEST n の条件を満たすソートフィールドの値が複数のレコードに存在する場合は、すべての値がレポートに表示されます。

ソートフィールド値のグループの限定

次のリクエストは、給与額が上位 5 名に入る従業員の名前を表示します。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
BY HIGHEST 5 CURR_SAL
END
```

出力結果は次のとおりです。

CURR_SAL	LAST_NAME
$29,700.00	BANNING
$27,062.00	CROSS
$26,862.00	IRVING
$21,780.00	BLACKWOOD
$21,120.00	ROMANS

レポートフィールドのソートおよび集計

BY TOTAL 句を使用して、1 回のデータのやり取りで、レポートの数値フィールドに対してソートと集計を同時に実行することができます。BY TOTAL 句を正しく動作させるには、SUM などの集計を表示するコマンドが必要です。PRINT などの集計を表示しないコマンドは、データを集計せずに、単にデータの取得だけを行います。指定したクエリにより、レコードが昇順または降順でソートされます。デフォルトは昇順です。

また、BY TOTAL 句を使用して、COMPUTE コマンドで計算した一時的な値を基にソートを行うこともできます。

レポートフィールドのソートおよび集計

```
[RANKED] BY [HIGHEST|LOWEST [n] ]
         TOTAL {display_field|COMPUTE name/format=expression;}
```

または

```
[RANKED] BY TOTAL {[HIGHEST|LOWEST [n] ]
        display_field|COMPUTE name/format=expression;}
```

説明

RANKED

出力するレポートに新しいフィールドを追加し、集計後のソート値に対する順位をこのフィールドで割り当てます。複数の行に同一の順位が割り当てられた場合、その順位は先頭の行にのみ表示されます。

n

レポートに表示するソートフィールド値の個数です。n を省略すると、計算したソートフィールドの値がすべて表示されます。デフォルトのソート順は昇順です。

display_field

フィールド名、演算子に続くフィールド名 (接頭語、フィールド名)、一時項目 (COMPUTE) のいずれかです。

内部マトリックスが作成された場合は、BY TOTAL フィールドは表示フィールドのように扱われます。マトリックスが作成された場合、すべてのソートフィールドに基づいて出力の行が集計され、再ソートが実行されます。

レポートフィールドのソートおよび集計

次の例では、給与の平均を計算し、それをソートフィールドとして使用します。レポートには、給与額の上位 2 つの値が表示されます。

```
TABLE FILE EMPLOYEE
SUM SALARY CNT.SALARY
BY DEPARTMENT
BY HIGHEST 2 TOTAL AVE.SALARY AS 'HIGHEST,AVERAGE,SALARIES'
```

```
BY CURR_JOBCODE
END
```

出力結果は次のとおりです。

DEPARTMENT	HIGHEST AVERAGE SALARIES	CURR_JOBCODE	SALARY	SALARY COUNT
MIS	$26,418.50	A17	$52,837.00	2
	$21,780.00	B04	$21,780.00	1
PRODUCTION	$29,700.00	A17	$29,700.00	1
	$25,641.00	A15	$51,282.00	2

レポートフィールドのソート、集計、順位付け

次の例では、給与の平均を計算し、それをソートフィールドとして使用します。給与額の上位 2 つの値が順位付けされて表示されます。

```
TABLE FILE EMPLOYEE
SUM SALARY CNT.SALARY
BY DEPARTMENT
RANKED BY HIGHEST 2 TOTAL AVE.SALARY AS 'HIGHEST,AVERAGE,SALARIES'
BY CURR_JOBCODE
END
```

出力結果は次のとおりです。

DEPARTMENT	RANK	HIGHEST AVERAGE SALARIES	CURR_JOBCODE	SALARY	SALARY COUNT
MIS	1	$26,418.50	A17	$52,837.00	2
	2	$21,780.00	B04	$21,780.00	1
PRODUCTION	1	$29,700.00	A17	$29,700.00	1
	2	$25,641.00	A15	$51,282.00	2

COMPUTE によるレポートフィールドのソートと集計

次の例では、ソートフィールド内で COMPUTE を使用して月給額を計算します。月給額の上位 2 つの値が表示されます。

```
TABLE FILE EMPLOYEE
SUM SALARY CNT.SALARY
BY DEPARTMENT
BY HIGHEST 2 TOTAL COMPUTE MONTHLY_SALARY/D12.2M=SALARY/12;
AS 'HIGHEST,MONTHLY,SALARIES'
BY CURR_JOBCODE
END
```

出力結果は次のとおりです。

DEPARTMENT	HIGHEST MONTHLY SALARIES	CURR_JOBCODE	SALARY	SALARY COUNT
MIS	$4,403.08	A17	$52,837.00	2
	$3,019.17	B03	$36,230.00	2
PRODUCTION	$4,273.50	A15	$51,282.00	2
	$2,591.67	B02	$31,100.00	2

ACROSS 句を含むリクエストでの一時項目 (COMPUTE) に対する BY TOTAL の使用

次のリクエストでは、「PROFIT」という一時項目 (COMPUTE) を作成し、BY TOTAL 句でその一時項目 (COMPUTE) を使用します。このリクエストでは、ACROSS RATING 句も使用されています。

```
TABLE FILE MOVIES
SUM LISTPR WHOLESALEPR
COMPUTE
PROFIT = LISTPR - WHOLESALEPR;
BY CATEGORY
BY TOTAL PROFIT
ACROSS RATING
```

```
WHERE  RATING NE 'NR' OR 'R'
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *GRID = OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。

		RATING G			PG			PG13		
CATEGORY	PROFIT	LISTPR	WHOLESALEPR	PROFIT	LISTPR	WHOLESALEPR	PROFIT	LISTPR	WHOLESALEPR	PROFIT
ACTION	13.92	.	.	.	34.90	20.98	13.92	.	.	.
CHILDREN	47.40	101.89	54.49	47.40	.	.	.	.	.	.
CLASSIC	48.96	89.95	40.99	48.96	.	.	.	.	.	.
COMEDY	58.12	.	.	.	74.83	46.70	28.13	59.99	30.00	29.99
FOREIGN	64.63	19.98	13.25	6.73	119.90	62.00	57.90	.	.	.
MUSICALS	25.93	29.95	15.00	14.95	19.98	13.99	5.99	14.98	9.99	4.99
MYSTERY	32.94	.	.	.	39.96	18.00	21.96	19.98	9.00	10.98
SCI/FI	18.94	.	.	.	.	.	.	54.93	35.99	18.94

BY TOTAL 使用上の注意

- BY HIGHEST/LOWEST *n* を BY TOTAL HIGHEST/LOWEST *n* とともに使用する場合、BY TOTAL 句は、BY 句の結果に対して適用されます (BY 句の結果の *n* 行)。

レポートに含めるフィールド数の制限

ACROSS により作成されるフィールド数は、次の個数により異なります。

- レポートの最大サイズは 32 キロバイトであるため、レポートのフィールドが表示されます。
- これらの各フィールドの値
- リクエスト内の ACROSS および ACROSS-TOTAL 句

行合計、BY フィールド、中間見出し、中間脚注、および見出しと脚注に使用するフィールドは ACROSS フィールド数の計算に含まれません。

ソート値の非表示

レポートをソートする場合、NOPRINT 句を使用してソートフィールドの値自体をレポートに表示させないようにすることができます。これはさまざまな状況で役立ちます。たとえば、同一のフィールドをソートフィールドおよび表示フィールドとして使用する場合や、フィールド別にソートしてもその値をレポートには出力しない場合などに役立ちます。

ソート値の非表示

```
{BY|ACROSS} sortfield {NOPRINT|SUP-PRINT}
```

説明

sortfield

ソートフィールドの名前です。

SUP-PRINT は NOPRINT と同義なため、代用することができます。

ソート値の非表示

次のリクエストでは、従業員を入社日の順にソートして表示し、従業員の姓、名、入社日の順序でそれぞれレポートに出力しようとしていますが、これだけでは不十分です。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME HIRE_DATE
END
```

出力結果は次のとおりです。

LAST_NAME	FIRST_NAME	HIRE_DATE
STEVENS	ALFRED	80/06/02
SMITH	MARY	81/07/01
JONES	DIANE	82/05/01
SMITH	RICHARD	82/01/04
BANNING	JOHN	82/08/01
IRVING	JOAN	82/01/04
ROMANS	ANTHONY	82/07/01
MCCOY	JOHN	81/07/01
BLACKWOOD	ROSEMARIE	82/04/01
MCKNIGHT	ROGER	82/02/02
GREENSPAN	MARY	82/04/01
CROSS	BARBARA	81/11/02

入社日の順に従業員をリスト表示するには、HIRE_DATE フィールドでレポートをソートするとともに、NOPRINT 句を使用してそのソートフィールドの値を非表示にする必要があります。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME HIRE_DATE
BY HIRE_DATE NOPRINT
END
```

出力結果は次のとおりです。

LAST_NAME	FIRST_NAME	HIRE_DATE
STEVENS	ALFRED	80/06/02
SMITH	MARY	81/07/01
MCCOY	JOHN	81/07/01
CROSS	BARBARA	81/11/02
SMITH	RICHARD	82/01/04
IRVING	JOAN	82/01/04
MCKNIGHT	ROGER	82/02/02
BLACKWOOD	ROSEMARIE	82/04/01
GREENSPAN	MARY	82/04/01
JONES	DIANE	82/05/01
ROMANS	ANTHONY	82/07/01
BANNING	JOHN	82/08/01

ソート処理パフォーマンスの考慮事項

ソート処理では、ディスク I/O 数を減らすために処理対象リクエストおよび利用可能ソートメモリ量が分析されます。このソート手法は、リクエストの詳細と、SORTMATRIX および SORTMEMORY パラメータ値によって制御されます。

SORTMATRIX

SORTMATRIX パラメータは、外部メモリの使用を抑えてインメモリのソートを使用するかどうかを制御します。構文は次のとおりです。

```
SET SORTMATRIX = {SMALL|LARGE}
```

説明

SMALL

最大 2048 行の単一ソートマトリックスを作成し、挿入ソートに基づくバイナリサーチを取得時の集計に使用します。このマトリックスの最大行数は、このタイプのソートで最適パフォーマンスとなるよう指定されています。ソートマトリックスが最大行数に達すると、ディスク上の FOCSORT というファイルに書き込まれ、インメモリのマトリックスを空にして取得を継続します。FOCSORT への書き込みは必要に応じて何回でも行われます。データの最後が検出されると、残りの行が FOCSORT に書き込まれ、集計処理を完了すると同時に、マージルーチンによって、ソート文字列がすべてFOCSORT に結合されます。場合によっては、複数の結合フェーズが必要になります。

LARGE

SORTMEMORY パラメータで指定した適切なメモリ量が使用可能な場合に、メモリ内に大規模なマトリックスまたは複数の小規模なマトリックスを作成します。デフォルト値は LARGE です。この方法を使用する目的は、ディスクにレコードを書き込む前に、できるだけ多くのソートを内部メモリで実行することです。ディスク I/O がソート処理に必要かどうかは、ソートに割り当てられたメモリ量およびリクエスト出力のサイズによります。SORTMEMORY の容量が十分でなく、LARGE を有意義に利用できない場合、ソートのデフォルトは SMALL に設定されます。LARGE を使用することで、ディスク I/O の要件を大幅に抑えることができます。そのため、最終的にディスク I/O が必要な場合 (出力が非常に大きい場合) に、複数の結合フェーズを使用する必要が実質的になくなります。

SORTMEMORY

SORTMEMORY パラメータは、ソートに利用可能な内部メモリ量を制御します。構文は次のとおりです。

```
SET SORTMEMORY = {n|512}
```

説明

n

ソートに利用可能なメモリのメガバイト数 (正の値) です。デフォルト値は 512 です。

複数表示コマンドによるソート

1 つのリクエストに使用できる表示コマンド (動詞句とも呼ばれる) は最大で 64 種類で、各コマンドにはそれぞれ個別のソート条件を含めることができます。すべてのデータを正しく表示するには、それぞれのソート条件の間で何らかの意味のある関係が存在していなければなりません。次の規則が適用されます。

- 最大で 64 種類の表示コマンドおよびそれに関連するソート条件を使用することができます。先頭の表示コマンドにはソート条件を設定する必要はありません。PRINT または LIST などの詳細コマンドは、最終の表示コマンドとしてのみ使用することができます。その表示コマンドは集計コマンドでなければなりません。
- WHERE 条件および IF 条件は、レポート用に選択したレコード全体に適用される。WHERE および IF 条件についての詳細は、レコードの選択を参照してください。
- 表示コマンドにソート句を使用する場合、それに続く表示コマンドには同一のソート条件を同一の順序で使用する必要があります。以下はその例です。

  ```
  TABLE FILE EMPLOYEE
  SUM ED_HRS
  SUM CURR_SAL CNT.CURR_SAL
  BY DEPARTMENT
  PRINT FIRST_NAME
  BY DEPARTMENT
  BY LAST_NAME
  END
  ```

  先頭の SUM にはソート条件は設定されていません。2 つ目の SUM には、「BY DEPARTMENT」というソート条件が設定されています。ここでこのソート条件を指定したために、PRINT コマンドには最初のソート条件として BY DEPARTMENT が必要になります。必要に応じて、他のソート条件を追加することができます。

複数の表示フィールドとソートフィールドの使用

次のリクエストは、データソースの詳細を段階別に集計して表示します。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL
```

```
SUM CURR_SAL BY DEPARTMENT
SUM CURR_SAL BY DEPARTMENT BY LAST_NAME
END
```

SUM CURR_SAL コマンドで現在の給与総額を計算します。SUM CURR_SAL BY DEPARTMENT コマンドで現在の給与総額を部署別に計算します。SUM CURR_SAL BY DEPARTMENT BY LAST_NAME コマンドで現在の給与総額を従業員別に計算します。

出力結果は次のとおりです。

CURR_SAL	DEPARTMENT	CURR_SAL	LAST_NAME	CURR_SAL
$222,284.00	MIS	$108,002.00	BLACKWOOD	$21,780.00
			CROSS	$27,062.00
			GREENSPAN	$9,000.00
			JONES	$18,480.00
			MCCOY	$18,480.00
			SMITH	$13,200.00
	PRODUCTION	$114,282.00	BANNING	$29,700.00
			IRVING	$26,862.00
			MCKNIGHT	$16,100.00
			ROMANS	$21,120.00
			SMITH	$9,500.00
			STEVENS	$11,000.00

複数表示コマンドを含むレポートのフォーマット制御

SET DUPLICATECOL コマンドを使用して、複数の表示コマンドが含まれたリクエストのフォーマットを再設定し、集計フィールドを同一列内の表示フィールドの上部に表示することができます。

デフォルト設定では、リクエストで表示コマンドを追加するごとに、ソートフィールドおよび表示フィールドの新しい列が生成されます。DUPLICATECOL を OFF に設定すると、フィールドはリクエスト内で 1 列として処理され、表示コマンド値は直前の表示コマンド値の下に表示されます。

複数表示コマンドを含むレポートのフォーマット制御

```
SET DUPLICATECOL={ON|OFF}
```

説明

ON

フィールドごとに新しい列が作成されてレポートに表示されます。これがデフォルト値です。

OFF

共通フィールドはすべて、レポートの同一列内に表示されます。

複数表示コマンドを使用したレポートの表示

次のリクエストは、Employee データソースと DEPARTMENT について、CURR_SAL と ED_HRS の合計を算出します。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL ED_HRS
SUM CURR_SAL ED_HRS BY DEPARTMENT
END
```

DUPLICATECOL=ON に設定すると、出力の総合計と DEPARTMENT 合計に別の列が表示されます。

```
   CURR_SAL  ED_HRS  DEPARTMENT         CURR_SAL  ED_HRS
   --------  ------  ----------         --------  ------
$222,284.00  351.00  MIS             $108,002.00  231.00
                     PRODUCTION      $114,282.00  120.00
```

DUPLICATECOL=OFF に設定すると、各フィールドは出力に 1 列として表示されます。総合計はレポートの 1 行目、DEPARTMENT 合計は総合計の下に追加された複数行に表示されます。

```
DEPARTMENT         CURR_SAL  ED_HRS
----------         --------  ------
                $222,284.00  351.00
MIS             $108,002.00  231.00
PRODUCTION      $114,282.00  120.00
```

次のリクエストには、前のリクエストに DEPARTMENT 別、LAST_NAME でソートする PRINT コマンドが追加されています。

```
SET SPACES = 1
TABLE FILE EMPLOYEE
SUM CURR_SAL ED_HRS
SUM CURR_SAL ED_HRS BY DEPARTMENT AS 'DEPT'
PRINT FIRST_NAME CURR_SAL ED_HRS BY DEPARTMENT BY LAST_NAME
END
```

DUPLICATECOL=ON に設定すると、出力の総合計と DEPARTMENT 合計に別の列が表示されます。

```
   CURR_SAL ED_HRS DEPT             CURR_SAL  ED_HRS LAST_NAME       FIRST_NAME        CURR_SAL ED_HRS
   -------- ------ ----             --------  ------ ---------       ----------        -------- ------
$222,284.00 351.00 MIS           $108,002.00  231.00 BLACKWOOD       ROSEMARIE       $21,780.00  75.00
                                                     CROSS           BARBARA         $27,062.00  45.00
                                                     GREENSPAN       MARY             $9,000.00  25.00
                                                     JONES           DIANE           $18,480.00  50.00
                                                     MCCOY           JOHN            $18,480.00    .00
                                                     SMITH           MARY            $13,200.00  36.00
                 PRODUCTION      $114,282.00  120.00 BANNING         JOHN            $29,700.00    .00
                                                     IRVING          JOAN            $26,862.00  30.00
                                                     MCKNIGHT        ROGER           $16,100.00  50.00
                                                     ROMANS          ANTHONY         $21,120.00   5.00
                                                     SMITH           RICHARD          $9,500.00  10.00
                                                     STEVENS         ALFRED          $11,000.00  25.00
```

DUPLICATECOL=OFF に設定すると、各フィールドは出力に 1 列として表示されます。総合計はレポートの 1 行目、DEPARTMENT 合計は総合計の下に追加された複数行に、さらに LAST_NAME 合計はその下に追加された複数行に表示されます。

```
DEPT       LAST_NAME       FIRST_NAME        CURR_SAL ED_HRS
----       ---------       ----------        -------- ------
                                          $222,284.00 351.00
MIS                                       $108,002.00 231.00
           BLACKWOOD       ROSEMARIE       $21,780.00  75.00
           CROSS           BARBARA         $27,062.00  45.00
           GREENSPAN       MARY             $9,000.00  25.00
           JONES           DIANE           $18,480.00  50.00
           MCCOY           JOHN            $18,480.00    .00
           SMITH           MARY            $13,200.00  36.00
PRODUCTION                                $114,282.00 120.00
           BANNING         JOHN            $29,700.00    .00
           IRVING          JOAN            $26,862.00  30.00
```

```
MCKNIGHT        ROGER           $16,100.00  50.00
ROMANS          ANTHONY         $21,120.00   5.00
SMITH           RICHARD          $9,500.00  10.00
STEVENS         ALFRED          $11,000.00  25.00
```

SET DUPLICATECOL=ON でのレポートスタイル設定

スタイルシートで、行の作成元となっている表示コマンドを指定することによって、スタイル設定する行を特定することができます。

```
VERBSET = n
```

説明

n

リクエスト内の表示コマンドの順序を表す番号です。

特定表示コマンドに関連する行のスタイル設定

次のリクエストには、2 つの表示コマンドが指定されています。

1. SUM CURR_SAL ED_HRS BY DEPARTMENT
2. PRINT FIRST_NAME CURR_SAL ED_HRS BY DEPARTMENT BY LAST_NAME

```
SET DUPLICATECOL = OFF
TABLE FILE EMPLOYEE
SUM CURR_SAL ED_HRS BY DEPARTMENT
PRINT FIRST_NAME CURR_SAL ED_HRS BY DEPARTMENT BY LAST_NAME

ON TABLE SET STYLE *
TYPE = REPORT, COLUMN= P4, VERBSET = 1, STYLE = ITALIC,    COLOR=BLUE,$
TYPE = REPORT, COLUMN= B2, VERBSET = 2, STYLE = UNDERLINE, COLOR = RED,$
ENDSTYLE
END
```

下図は、出力結果を示しています。

- 4 つ目の表示列 (P4、CURR_SAL の部門合計) は、斜体の青色で表示されます。
- PRINT コマンドを表す 2 つ目の BY フィールド (LAST_NAME) は、下線付きの赤色で表示されます。

特定のフィールドにスタイルを設定する際、P 表記を使用すると、BY フィールドを含めて、レポート出力に表示されるすべてのフィールドが数えられます。そのため、P1 は DEPARTMENT フィールド、P2 は LAST_NAME フィールド (これは同時に 2 つ目の BY フィールド、すなわち B2 でもある)、P3 は FIRST_NAME フィールド、P4 は PRINT で指定された CURR_SAL フィールド (内部マトリックスには複数のCURR_SAL フィールドが存在する)、P5 は PRINT で指定された ED_HRS フィールド (内部マトリックスには複数の ED_HRS フィールドが存在) となります。

出力結果は次のとおりです。

PAGE 1				
DEPARTMENT	LAST_NAME	FIRST_NAME	CURR_SAL	ED_HRS
MIS			*$108,002.00*	231.00
	BLACKWOOD	ROSEMARIE	$21,780.00	75.00
	CROSS	BARBARA	$27,062.00	45.00
	GREENSPAN	MARY	$9,000.00	25.00
	JONES	DIANE	$18,480.00	50.00
	MCCOY	JOHN	$18,480.00	.00
	SMITH	MARY	$13,200.00	36.00
PRODUCTION			*$114,282.00*	120.00
	BANNING	JOHN	$29,700.00	.00
	IRVING	JOAN	$26,862.00	30.00
	MCKNIGHT	ROGER	$16,100.00	50.00
	ROMANS	ANTHONY	$21,120.00	5.00
	SMITH	RICHARD	$9,500.00	10.00
	STEVENS	ALFRED	$11,000.00	25.00

複数動詞リクエストでの AS 名の照合に基づく重複フィールドの積み上げ表示

SET DUPLICATECOL コマンドを使用して、複数の表示コマンドが含まれたリクエストのフォーマットを再設定し、集計フィールドを同一列内の表示フィールドの上部に表示することができます。

デフォルト設定では、リクエストで表示コマンドを追加するごとに、ソートフィールドおよび表示フィールドの新しい列が生成されます。DUPLICATECOL を OFF に設定すると、フィールドはリクエスト内で 1 列として処理され、表示コマンド値は直前の表示コマンド値の下に表示されます。

以前のバージョンでは、重複フィールドはフィールド名に基づいて照合されていました。新しいバージョンでは、重複フィールドは AS 名に基づく方法でも照合されます。ただし、AS 名はフィールド名とは照合されません。フィールドに AS 名が定義されている場合、そのフィールドは、同一の AS 名が定義された他のフィールドとのみ照合されます。

複数動詞リクエストでの AS 名の照合に基づく重複フィールドの積み上げ表示

次のリクエストでは、3 つの表示コマンドが使用されています。1 つ目のコマンドは、CURR_SAL フィールドを集計します。2 つ目のコマンドは、SALARY フィールドを部門別に集計します。3 つ目のコマンドは、GROSS フィールドを部門別および名前 (性) 別に表示します。CURR_SAL フィールドも含め、これらの 3 つのフィールドに同一の AS 名が割り当てられています。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL AS CURR_SAL  ED_HRS
SUM SALARY AS CURR_SAL ED_HRS BY DEPARTMENT AS 'DEPT'
PRINT FIRST_NAME GROSS AS CURR_SAL ED_HRS BY DEPARTMENT BY LAST_NAME
ON TABLE SET DUPLICATECOL OFF
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF, SIZE=10, $
VERBSET=1, COLOR=RED,$
VERBSET=2, COLOR=BLUE,$
VERBSET=3,COLOR=BLACK,$
```

```
ENDSTYLE
END
```

下図は、出力結果の一部を示しています。

DEPT	LAST_NAME	FIRST_NAME	CURR_SAL	ED_HRS
			$222,284.00	351.00
MIS			$160,177.00	231.00
	BLACKWOOD	ROSEMARIE	$1,815.00	75.00
		ROSEMARIE	$1,815.00	75.00
		ROSEMARIE	$1,815.00	75.00
		ROSEMARIE	$1,815.00	75.00
		ROSEMARIE	$1,815.00	75.00
	CROSS	BARBARA	$2,255.00	45.00
		BARBARA	$2,255.00	45.00
		BARBARA	$2,255.00	45.00
		BARBARA	$2,255.00	45.00
		BARBARA	$2,255.00	45.00
		BARBARA	$2,147.75	45.00
		BARBARA	$2,147.75	45.00
		BARBARA	$2,147.75	45.00
		BARBARA	$2,147.75	45.00
		BARBARA	$2,147.75	45.00
	GREENSPAN	MARY	$750.00	25.00
		MARY	$750.00	25.00
		MARY	$750.00	25.00
		MARY	$720.84	25.00
	JONES	DIANE	$1,540.00	50.00
		DIANE	$1,540.00	50.00
		DIANE	$1,540.00	50.00
		DIANE	$1,479.50	50.00
	MCCOY	JOHN	$1,540.00	.00
	SMITH	MARY	$1,100.00	36.00
		MARY	$1,100.00	36.00
		MARY	$1,100.00	36.00
		MARY	$1,100.00	36.00
		MARY	$1,100.00	36.00
		MARY	$1,100.00	36.00
		MARY	$1,100.00	36.00
		MARY	$1,100.00	36.00

外部ソートによる効率化

レポートを作成する場合、レポートのソートはデフォルト設定で内部ソートプロシジャを使用するようになっています。内部ソートプロシジャで処理できるデータサイズはさまざまな要因に左右されますが、このソート方法ではおよそ 180 キロバイトから 200 キロバイトまでのレポートサイズであれば最適化されます。

内部ソートに使用する FOCSORT ファイルのサイズは、オペレーティングシステムおよび使用可能なディスク領域で許可されている任意のサイズまで増やすことができます。ユーザは、サイズが大きいファイルに対応するためリクエストを分割する必要はありません。以前のバージョンでは、FOCSORT ファイルは 2 ギガバイトに制限されていたため、WebFOCUS の制限を超えた場合に FOC298 メッセージが表示されていました。WebFOCUS で制限が設定されていない場合は、オペレーティングシステムが制限を超える FOCSORT ファイルを管理し、警告およびエラー処理を行います。

外部ソートを使用するには、EXTSORT パラメータを ON に設定しておく必要があります。スタイルシートを使用すると、外部ソートは OFF になります。

外部ソート使用上の注意

次の場合には、外部ソートを使用しないことをお勧めします。

- リクエストにマトリックスを使用する必要がある (TABLEF リクエストには変換不可)。リクエストにマトリックスを使用する必要があり、かつ外部ソートを使用する場合は、外部および内部の両方のソートが実行されるため、パフォーマンスの向上は期待できません。

  レポートを TABLEF に変換できるかどうかを確認するには、?STAT (ソートタイプのクエリを参照) を使用するか、外部ソートを ON にし、レコードの制限を最小にした状態でリクエストの簡略バージョンを実行します。レポートの統計が TABLE 出力の後に表示された場合は、TABLEF として実行されたことになります。逆に、統計が TABLE 出力のより前に表示された場合は、TABLEF に変換できなかったため、TABLE 処理として実行されたことになります。

- 入力したものをすべてソートする、またはその大部分をソートする。
- 使用中のシステムが、多数の作業ファイルを処理できない。詳細は、ソート作業ファイルとリターンコードを参照してください。この場合は、内部ソートの方が効率よく処理を行える場合があります。それは、内部的に FOCSORT のスペースを共有しておよそ 60 個の論理作業ファイルを実行しているためです。

使用するソート方法を決定するには

どのソート方法を使用するかは、次の順序で基準が評価されます。

手順

1. **BINS**　レポート全体のソートを作業領域 (BINS) 内で行える場合は、EXTSORT の設定が ON になっていても外部ソートは呼び出されません。
2. **EXTERNAL**　レポート全体のソートを行えるだけの容量が作業領域 (BINS) に確保できず、EXTSORT の設定が ON の場合は、外部ソートユーティリティが呼び出されます。

外部ソートの制御

外部ソート機能の ON と OFF を切り替えるには、SET EXTSORT コマンドを使用します。

```
SET EXTSORT = {ON|OFF}
```

説明

ON

レポートのソートに外部ソートを使用できるようにします。

OFF

レポートのソートに内部ソートプロシジャを使用します。この値がデフォルトになります。

ソートタイプのクエリ

レポートにどのソート方法が使用されているかを確認するには、リクエストの後に次のコマンドを発行します。

```
?STAT
```

このコマンドにより、SORT USED パラメータの値が次のように表示されます。

FOCUS

内部ソートプロシジャを使用してレポート全体がソートされたことを表します。

SQL

リレーショナルデータソースを使用し、RDBMS からソート済みのデータが提供されたことを表します。

EXTERNAL

外部ソートを使用してレポートがソートされたことを表します。

NONE

レポートのソートを行う必要がなかったことを表します。

ソートに使用する入力レコードまたはレポートサイズの推定値の取得

入力サイズ (ESTRECORDS) またはレポートサイズ (ESTLINES) の推定値の取得には次の 2 つの利点があります。

- リクエストを TABLEF リクエストに変換できず、ファイルサイズの見積もりで外部ソートが必要と示された場合、FOCUS は外部ソートを即時に開始するため、FOCUS マージは不要になります。見積もりを取得しない場合、このリクエストでは常に FOCUS によるマージが実行されます。

ソートに使用する入力レコードまたはレポートサイズの推定値を取得するには

```
ON TABLE SET ESTRECORDS nON TABLE SET ESTLINES n
```

説明

n

ソートするレコード数または行数の推定値です。

ソート作業ファイルとリターンコード

外部ソートは一時作業ファイルを使用してソートの中間結果を保持します。異なるタイプの外部ソートを使用する場合は、それぞれのタイプでソート用の作業ファイルがどのように作成され、使用されるかを理解しておく必要があります。

UNIX、Windows でのソート用作業ファイル

内部ソートでは EDATEMP ディレクトリに割り当てられた FOCSORT 作業ファイルを 1 つだけ使用しますが、外部ソートでは 1 つまたは複数のディスクドライブ (スピンドル) またはディレクトリに割り当てられた最大で 31 個の作業ファイルを使用することができます。

警告：これらの作業ファイルのうち 1 つまたは複数のファイルがかなり大きくなる場合があります。FOCSORT に必要な全ディスク領域の数倍もの領域が使用されることを計算に入れておく必要があります。

デフォルト設定では、UNIX の /tmp ディレクトリ、または Windows の TMP 環境変数として指定されたディレクトリに、5 つの作業ファイルが 割り当てられます。これでもソート用の作業領域が不足する場合があります。ファイルがディレクトリに正しく割り当てられていても、通常は 5 つのファイルでは最適パフォーマンスを得るには不十分です。また、ソート用のすべての作業ファイルが同一ディスクにあると、パフォーマンスが低下する場合があります。

ここでは、次の 2 つのオプションがあります。

- TMPDIR シェル変数 (UNIX) または TMP 環境変数 (Windows) を定義し、適正な書き込みが可能なディレクトリを指定します。最適な結果を得るには、このディレクトリはデータソースまたは EDATEMP ディレクトリと同一のディスクではなく、利用可能な領域が十分にあるディスクに配置します。繰り返しになりますが、5 つの一時作業ファイルは同一のスピンドルに割り当てられるため、結果としてパフォーマンスが低下します。
- IBITMPDIR01 ... IBITMPDIR31 フォームの 1 から 31 までのシェル変数を定義し、1 つ

または複数の書き込み可能なディレクトリを指すようにします。

UNIX TMPDIR または Windows TMP 変数が設定されている場合、IBITMPDIRnn 変数を使用するには、これらの変数を未設定の状態にする必要があります。TMPDIR 変数を未設定の状態にする UNIX コマンドは次のとおりです。

```
```

TMP 変数を未設定の状態にする Windows コマンドは次のとおりです。

```
SET TMP=
```

異なる変数で同一ディレクトリを指定することができます。n 個の作業ファイルを割り当てる場合は、01 から n までの変数を定義する必要があります。どの変数が最初に環境から欠落しているかにより、使用する作業ファイル数が決定されます。定義する作業ファイルが 5 個未満の場合は、システムデフォルトロケーションにより追加ファイルが割り当てられ、その差分を補います。割り当てる作業ファイル数が多いほど、そして作業ファイルを異なるスピンドルに広く分散させるほどパフォーマンスが向上します。最大の制約は、使用できるディスク領域に制限があることです。

作業ファイルの名前は ANSI tempnam 関数により作成され、すべての名前に接頭文字として srtwk が付けられます。ソート処理が正常に終了した場合、またはエラー (通常はディスク領域のオーバフロー) が検知されて中断された場合は、割り当てられた作業ファイルはすべて削除されます。作業ファイルを明示的に保存する方法はありません。他の理由で異常な中断が発生した場合は、srtwk ファイルがディスクに残る場合があります。その場合は、ファイルを手動で消去することができます。また、消去する必要があります。

ibi™ WebFOCUS® 外部ソートのリターンコード

WebFOUCS エラーメッセージ FOC909 は、外部ソートのすべてのエラーに対して発行されます。この場合、3 桁のコードが追加され、そのうち末尾の 2 桁が重要な意味を持ちます。末尾のエラー番号の意味とその対処法は次のとおりです。

- 16 - 外部ソートに十分なメモリが割り当てられませんでした。この場合は、作業ファイル数を減らしてみます。
- 20 - I/O エラーが発生しました。ほとんどの場合は、ディスクのいずれかが書き込み不可であるか、ディスクがオーバーフローしたためです。この場合は、作業ファイルを別の場所に割り当てたり、作業ファイル数を減らしたりします。

- 28 - 作業ファイルのいずれかを開くことができませんでした。この場合は、パス名が正確に指定されていること、書き込みおよび読み出しのプロテクトがかかっていないことを確認します。
- 32 - ソート処理中に、内部論理エラーが検知されました。この問題を技術サポートに連絡してください。

階層レポート - BY HIERARCHY

Essbase や SAP BW などのキューブデータソースは、ディメンションとファクトで構成されています。多くの場合、ディメンションは階層構造を形成しています。キューブデータソースのシノニムには、ディメンション階層を定義する属性が含まれています。WebFOCUS の階層レポート構文を使用して、これらの階層から自動的にレポートを作成し、出力結果をインデント表示することで、階層関係を表すことができます

WebFOCUSシノニムでディメンション階層を定義した後、これらのディメンションに対して階層レポートリクエストを実行することができます。階層属性が定義された非キューブシノニムは「仮想キューブ」と呼ばれます。

地域や時間などのデータカテゴリは「ディメンション」と呼ばれ、このディメンションを操作してビジネスパフォーマンスの分析や比較を行います。ディメンションは、「メンバー」と呼ばれるデータ要素で構成されます。たとえば、「地域」というディメンションを「イギリス」と「フランス」というメンバーで構成することができます。

通常、ディメンションのメンバーは階層構造を形成します。階層は、各メンバーがノードとして表現されたツリー構造と見なすことができます。たとえば、「地域」ディメンションの最上位レベル (ルートノード) に「世界」という要素を追加します。「世界」という要素の下に、「大陸」という子ノード (メンバー) を追加します。同様に、「大陸」というノードの下に国名を表す子ノードを、国名のノードの下に都市名を表す子ノードを順に追加します。また、子ノードの存在しないノードは、「リーフノード」と呼ばれます。

売上高や純利益などの数値データは「メジャー」と呼ばれ、このメジャーを使用して業績を数値で評価します。

キューブは、「ファクト」(事実) から導かれたデータで構成されます。ファクトとは、特定のビジネストランザクションのレコードのことです。たとえば、セールストランザクションに、「日時」、「店舗名」、「製品名」、「販売価格」、「売上数量」という要素が関係する場合、1 つのファクトレコードが 1 つのセールストランザクションを表します。キューブには、メジャーおよび各種ディメンションのメンバーすべての組み合わせの集計ファクト値が格納されます。

シノニムでは、階層構造および階層メンバー間の関係を定義する一連のフィールドを使用して階層が記述されます。WebFOCUS には、階層からレポートを作成するための特別な階層レポート構文があります。

階層レポートリクエストは、複数のステップで構成されます。

- **ステップ 1 - 表示する階層メンバーの選択**

  階層レポートの BY 句および ON HIERARCHY 句は、階層のソートと親子関係を示すインデントを自動的に設定します。また、子メンバーのメジャー値が自動的に集約され、親メンバーのメジャー値が生成されます。

  階層全体を表示しない場合は、WHEN 句を使用することにより、表示する階層メンバーを選択することができます。この WHEN 句の式は、ディメンションプロパティやメジャーではなく、階層フィールドのみを参照する必要があります。

- **ステップ 2 - 取得されたディメンションデータの選別**

  WHERE 選択条件は、ステップ 1 で選択されたメンバーのリーフノードに適用されます。このため、ディメンションプロパティは WHERE テストで使用することもできます。これらのテストで、階層フィールドを参照することもできます。ただし、選択条件は常にリーフノードの値に適用されるため、上位の値に基づいてデータを選択することはできません。たとえば、大陸名、国名、都市名で構成されたディメンションで国名レベルを選択する WHERE 条件を使用しても行は表示されません。この場合、都市名レベルを選択する WHERE 条件を使用することができます。WHERE テストでは、メジャーを参照することもできます。

- **ステップ 3 - 集計値に基づく選別**

  集計値であるメジャーは、WHERE TOTAL テストおよび COMPUTE コマンドで参照することができます。これらのコマンドは、リクエスト内の階層の選択と集計句の後で処理されるためです。

マスターファイルでの階層の指定

データソースには、階層構造が定義されたディメンションが少なくとも 1 つ存在する必要があります。ディメンションの宣言は次のとおりです。

```
DIMENSION=dimname,CAPTION=dimcaption, $
```

説明

dimname

ディメンションの名前です。

dimcaption

ディメンションのラベルです。

ディメンション内の階層の宣言は次のとおりです。

```
HIERARCHY=hname,CAPTION='hcaption',HRY_DIMENSION=dimname,
HRY_STRUCTURE=RECURSIVE, $
```

説明

hname

階層の名前です。

hcaption

階層のラベルです。

dimname

この階層が定義されているディメンションの名前です。

親子階層は、複数のフィールドを使用して定義します。各フィールドの PROPERTY 属性により、フィールドの階層プロパティが定義されます。各階層には、ユニーク ID フィールドが存在する必要があります。このフィールドは「階層フィールド」と呼ばれます。シノニムが FOCUS データソースを表す場合、このフィールドはインデックスフィールドである必要があります (FIELDTYPE=I)。階層フィールドの宣言は次のとおりです。

```
FIELD=hfield,ALIAS=halias,USAGE= An, [ACTUAL=Am,]
WITHIN='*hierarchy',PROPERTY=UID, [TITLE='title1',] [FIELDTYPE=I,] $
```

説明

hfield

階層フィールドの名前です。

halias

階層フィールドのエイリアスです。リレーショナルデータソースの場合、このエイリアスはリレーショナル DBMS のフィールド名に一致させる必要があります。

hierarchy

このフィールドが属する階層の名前です。

USAGE= An, [ACTUAL=Am,]

USAGE フォーマットです。FOCUS データソース以外のデータソースの場合、フィールドの ACTUAL フォーマットです。

title1

フィールドのタイトルです (オプション)。

この階層で定義される他のフィールドとして、親フィールドとキャプションフィールドがあります。これらの各フィールドには、階層フィールド名に接尾語を追加した名前が付けられます。各フィールドの PROPERTY 属性で、階層内でのフィールドの役割を指定し、REFERENCE 属性で、対応する階層フィールドを指定します。

親フィールドの宣言は次のとおりです。親フィールドは、階層内の親子関係を定義するために必要です。

```
FIELD=hfield_PARENT,ALIAS=parentalias,USAGE=An,[ACTUAL=Am,] [TITLE=ptitle,]
      PROPERTY=PARENT_OF, REFERENCE=hfield, $
```

説明

hfield

階層フィールドです。

parentalias

親フィールドのエイリアスです。リレーショナルデータソースの場合、このエイリアスはリレーショナル DBMS のフィールド名に一致させる必要があります。

USAGE= An, [ACTUAL=Am,]

USAGE フォーマットです。FOCUS データソース以外のデータソースの場合、フィールドの ACTUAL フォーマットです。

ptitle

親フィールドのタイトルです (オプション)。

キャプションフィールドの宣言は次のとおりです。キャプションは、階層フィールドのそれぞれの値を説明するタイトルです。キャプションはデータの一部であるため、マスターファイルの TITLE 属性とは異なります。TITLE 属性は、レポート出力にフィールドタイトルとして表示されます。

```
FIELD=hfield_CAPTION,ALIAS=capalias,USAGE=Ann,[ACTUAL=Amm,] [TITLE=captitle,]
        PROPERTY=CAPTION, REFERENCE=hfield, $
```

説明

hfield

階層フィールドです。

capalias

キャプションフィールドのエイリアスです。リレーショナルデータソースの場合、このエイリアスはリレーショナル DBMS のフィールド名に一致させる必要があります。

USAGE= Ann, [ACTUAL=Amm,]

USAGE フォーマットです。FOCUS データソース以外のデータソースの場合、フィールドの ACTUAL フォーマットです。

captitle

キャプションフィールドのフィールドタイトルです。

ディメンション階層が定義されたサンプルマスターファイル

次のマスターファイルは、FML 階層が定義された CENTGL マスターファイルを基に作成されたものです。このマスターファイルの名前は NEWGL で、複数のアカウントがディメンション階層で定義されています。GL_ACCOUNT は階層フィールド、GL_ACCOUNT_PARENT は親フィールド、GL_ACCOUNT_CAPTION はキャプションフィールドです。階層に基づいた他のフィールドもあります (GL_ACCOUNT_LEVEL、GL_ROLLUP_OP、GL_ACCOUNT_TYPE)。また、メジャーフィールドも定義されています (GL_ACCOUNT_AMOUNT)。

```
FILE=NEWGL       ,SUFFIX=FOC,$
SEGNAME=ACCOUNTS   ,SEGTYPE=S01
DIMENSION=Accnt,CAPTION=Accnt, $
HIERARCHY=Accnt,CAPTION='Accnt',HRY_DIMENSION=Accnt,
HRY_STRUCTURE=RECURSIVE, $
FIELD=GL_ACCOUNT,GLACCT,A7,WITHIN='*Accnt',PROPERTY=UID,
            TITLE='Ledger,Account', FIELDTYPE=I, $
FIELD=GL_ACCOUNT_PARENT,GLPAR,A7, TITLE=Parent,
            PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT, $
```

```
FIELD=GL_ACCOUNT_TYPE,GLTYPE,A1, TITLE=Type,$
FIELD=GL_ROLLUP_OP,ROLL,A1, TITLE=Op, $
FIELD=GL_ACCOUNT_LEVEL,GLLEVEL,I3, TITLE=Lev, $
FIELDNAME=GL_ACCOUNT_AMOUNT,GLAMT,D12.2, TITLE=Amount, $
FIELD=GL_ACCOUNT_CAPTION,GLCAP,A30, TITLE=Caption,
            PROPERTY=CAPTION, REFERENCE=GL_ACCOUNT, $
FIELD=SYS_ACCOUNT,ALINE,A6, TITLE='System,Account,Line', MISSING=ON, $
```

このマスターファイルが WebFOCUS で使用可能な場合 (パス上に存在、または割り当て済み)、次のプロシジャは、このデータソースにデータをロードします。

```
CREATE FILE NEWGL NOMSG
-RUN
MODIFY FILE NEWGL
COMPUTE TGL_ACCOUNT_LEVEL/A3=;
COMPUTE TGL_ACCOUNT_AMOUNT/A12=;
FIXFORM GL_ACCOUNT/A4B X3 GL_ACCOUNT_PARENT/A4B X3 GL_ACCOUNT_TYPE/A1B
FIXFORM SYS_ACCOUNT/A4B GL_ROLLUP_OP/A1B
FIXFORM TGL_ACCOUNT_LEVEL/A3B GL_ACCOUNT_CAPTION/A30B
FIXFORM TGL_ACCOUNT_AMOUNT/A12B
COMPUTE GL_ACCOUNT_LEVEL = EDIT(TGL_ACCOUNT_LEVEL);
COMPUTE GL_ACCOUNT_AMOUNT = ATODBL(TGL_ACCOUNT_AMOUNT , '12', GL_ACCOUNT_
AMOUNT);

MATCH GL_ACCOUNT
   ON MATCH REJECT
   ON NOMATCH INCLUDE
```

```
DATA
1000          R.   +  1Profit Before Tax
2000   1000   R.   +  2Gross Margin
2100   2000   R.   +  3Sales Revenue
2200   2100   R.   +  4Retail Sales
2210   2200   R7001+  5Retail - Television                  505.00
2220   2200   R7002+  5Retail - Stereo                      505.00
2230   2200   R7003+  5Retail - Video Player               505.00
2240   2200   R7004+  5Retail - Computer                   505.00
2250   2200   R7005+  5Retail - Video Camera               505.00
2300   2100   R.   +  4Mail Order Sales
2310   2300   R7011+  5Mail Order - Television            505.00
2320   2300   R7012+  5Mail Order - Stereo                505.00
2330   2300   R7013+  5Mail Order - Video Player          505.00
2340   2300   R7014+  5Mail Order - Computer              505.00
2350   2300   R7015+  5Mail Order - Video Camera          505.00
2400   2100   R.   +  4Internet Sales
```

```
2410   2400   R7021+  5Internet - Television                505.00
2420   2400   R7022+  5Internet - Stereo                    505.00
2430   2400   R7023+  5Internet - Video Player             505.00
2440   2400   R7024+  5Internet - Computer                 505.00
2450   2400   R7025+  5Internet - Video Camera             505.00
2500   2000   E.   -  3Cost Of Goods Sold
2600   2500   E.   +  4Variable Material Costs
2610   2600   E7101+  5Television COGS                      505.00
2620   2600   E7102+  5Stereo COGS                          505.00
2630   2600   E7103+  5Video COGS                           505.00
2640   2600   E7104+  5Computer COGS                        505.00
2650   2600   E7105+  5Video Camera COGS                    505.00
2700   2500   E7111+  4Direct Labor                         404.00
2800   2500   E7112+  4Fixed Costs                          404.00
3000   1000   E.   -  2Total Operating Expenses
3100   3000   E.   +  3Selling Expenses
3110   3100   E.   +  4Advertising
3112   3110   E7202+  5TV/Radio                             505.00
3114   3110   E7203+  5Print Media                          505.00
3116   3110   E7206+  5Internet Advertising                 505.00
3120   3100   E7212+  4Promotional Expenses                 404.00
3130   3100   E7213+  4Joint Marketing                      404.00
3140   3100   E7214+  4Bonuses/Commisions                   404.00
3200   3000   E.   +  3General + Admin Expenses
3300   3200   E.   +  4Salaries-Corporate
```

```
3310   3300   E7301+  5Salaries-Corp Mgmt                   505.00
3320   3300   E7302+  5Salaries-Administration             505.00
3330   3300   E7303+  5IT Contractors                       505.00
3400   3200   E.   +  4Company Benefits
3410   3400   E7311+  5Social Security                      505.00
3420   3400   E7312+  5Unemployment                         505.00
3430   3400   E7313+  5Vacation Pay                         505.00
3440   3400   E7314+  5Sick Pay                             505.00
3450   3400   E.   +  5Insurances
3451   3450   E7321+  6Medical Insurance                    606.00
3452   3450   E7322+  6Dental Insurance                     606.00
3453   3450   E7323+  6Pharmacy Insurance                   606.00
3454   3450   E7324+  6Disability Insurance                 606.00
3455   3450   E7325+  6Life Insurance                       606.00
3500   3200   E.   +  4Depreciation Expenses
3510   3500   E7411+  5Equipment                            505.00
3520   3500   E7412+  5Building                             505.00
3530   3500   E7413+  5Vehicles                             505.00
3600   3200   R7414-  4Gain/(Loss) Sale of Equipment       404.00
3700   3200   E.   +  4Leasehold Expenses
```

```
3710   3700   E7421+  5Equipment                              505.00
3720   3700   E7422+  5Buildings                              505.00
3730   3700   R7429-  5Sub-Lease Income                       505.00
3800   3200   E7440+  4Interest Expenses                     404.00
3900   3200   E.   +  4Utilities
3910   3900   E7451+  5Electric                               505.00
3920   3900   E7452+  5Gas                                    505.00
3930   3900   E7453+  5Telephone                              505.00
3940   3900   E7454+  5Water                                  505.00
3950   3900   E7455+  5Internet Access                        505.00
5000   1000   E.   -  2Total R+D Costs
5100   5000   E7511+  3Salaries                               303.00
5200   5000   E7521+  3Misc.Equipment                        303.00
END
```

階層レポートの作成

階層レポートでは、子ディメンションメンバーのメジャー値が集約され、親メンバーの値が生成されます。データソースでは、そのメジャーに対して親メンバーの値は存在しません。

```
SUM measure_field ...
BY hierarchy_field [HIERARCHY [WHEN expression_using_hierarchy_fields;]
[SHOW [TOP|UP n] [TO {BOTTOM|DOWN m}] [byoption [WHEN condition] ...] ]
[WHERE expression_using_dimension_data]
[ON hierarchy_field HIERARCHY [WHEN expression_using_hierarchy_fields;]
[SHOW [TOP|UP n] [TO BOTTOM|DOWN m] [byoption [WHEN condition] ...]]
```

説明

measure_field

メジャーのフィールド名です。

BY hierarchy_field HIERARCHY

ソートに使用する階層です。このフィールドは階層フィールドです。

ON hierarchy_field HIERARCHY

ソートに使用する階層です。このフィールドは階層フィールドです。リクエストには、このフィールド名に対する BY 句または BY HIERARCHY 句のいずれかを使用する必要があります。

WHEN expression_using_hierarchy_fields;

階層メンバーを選択します。WHEN 句は、「HIERARCHY」という語句の直後に配置して、BY オプションの WHEN 句 (例、SUBFOOT) と区別する必要があります。ここでは、階層フィールドのみで構成された任意の式を使用することができます。WHEN 句は、BY HIERARCHY コマンドまたは ON HIERARCHY コマンドのいずれかに使用することができますが、これらの両方に使用することはできません。

SHOW

WHEN 句で選択したレベルとの相対値を使用して、レポート出力に表示するレベルを指定します。WHEN 句を使用しない場合、SHOW オプションは階層のルートノードに適用されます。SHOW オプションは、BY HIERARCHY 句または ON HIERARCHY 句のいずれかで使用することはできますが、これらの両方に使用することはできません。

n

メジャー値を持ち、選択したメンバーより上位にあるレベル数です。すべての上位レベルは、選択したメンバーの階層コンテキストとしてレポートに表示されます。ただし、SHOW 句に上位レベルを含めていない場合、その上位レベルのレポートカラムにはミッシングデータ記号が表示されます。n のデフォルト値は 0 (ゼロ) です。

TOP

階層のルートノードまでの上位レベルにメジャー値を表示します。

TO

SHOW オプションで下位レベルを指定する際に必要です。

BOTTOM

階層のリーフノードまでのすべての下位レベルにメジャー値を表示します。これがデフォルト値です。

m

表示対象として選択したレベルより下位にあるレベル数です。m のデフォルト値は BOTTOM です。この設定では、すべての下位レベルが表示されます。

byoption

ソートオプションの PAGE-BREAK、REPAGE、RECAP、RECOMPUTE、SKIP-LINE、SUBFOOT、SUBHEAD、SUBTOTAL、SUB-TOTAL、SUMMARIZE、UNDER-LINE のいずれかです。SUBHEAD または SUBFOOT を指定する場合は、見出しまたは脚注テキストの次の行に WHEN 句を記述する必要があります。

condition

論理式です。

expression_using_dimension_data

BY HIERARCHY 句、ON HIERARCHY 句、WHEN 句で選択した行をディメンションデータに基づいて選別します。式でディメンションプロパティおよび階層フィールドを使用することができますが、選択条件は常にリーフノードの値に適用されます。このため、WHERE を使用して、上位レベルの階層フィールド値に基づいて行を選択することはできません。たとえば、大陸名、国名、都市名で構成されたディメンションで国名レベルを選択する WHERE 条件を使用しても行は表示されません。この場合、都市名レベルを選択する WHERE 条件を使用することができます。

ディメンション階層レポートの作成

次のリクエストは、NEWGL データソースの GL_ACCOUNT 階層全体からレポートを作成します。

```
TABLE FILE NEWGL
SUM GL_ACCOUNT_AMOUNT
BY GL_ACCOUNT HIERARCHY
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
TYPE=REPORT,GRID=OFF,$
ENDSTYLE
END
```

下図は、出力結果の一部を示しています。Account フィールド値がインデントされ、階層関係を示しています。

Ledger Account	Amount
1000	27,169.00
2000	10,908.00
2100	7,575.00
2200	2,525.00
2210	505.00
2220	505.00
2230	505.00
2240	505.00
2250	505.00
2300	2,525.00
2310	505.00
2320	505.00
2330	505.00
2340	505.00
2350	505.00
2400	2,525.00
2410	505.00
2420	505.00
2430	505.00
2440	505.00
2450	505.00
2500	3,333.00
2600	2,525.00
2610	505.00
2620	505.00
2630	505.00
2640	505.00
2650	505.00

この例では、上記と同一のリクエストで GL_ACCOUNT_CAPTION フィールドを使用します。

```
TABLE FILE NEWGL
SUM GL_ACCOUNT_AMOUNT
BY GL_ACCOUNT_CAPTION HIERARCHY
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
TYPE=REPORT,GRID=OFF,$
```

```
ENDSTYLE
END
```

下図は、出力結果の一部を示しています。

Caption	Amount
Profit Before Tax	27,169.00
Gross Margin	10,908.00
Sales Revenue	7,575.00
Retail Sales	2,525.00
Retail - Television	505.00
Retail - Stereo	505.00
Retail - Video Player	505.00
Retail - Computer	505.00
Retail - Video Camera	505.00
Mail Order Sales	2,525.00
Mail Order - Television	505.00
Mail Order - Stereo	505.00
Mail Order - Video Player	505.00
Mail Order - Computer	505.00
Mail Order - Video Camera	505.00
Internet Sales	2,525.00
Internet - Television	505.00
Internet - Stereo	505.00
Internet - Video Player	505.00
Internet - Computer	505.00
Internet - Video Camera	505.00
Cost Of Goods Sold	3,333.00
Variable Material Costs	2,525.00
Television COGS	505.00
Stereo COGS	505.00
Video COGS	505.00
Computer COGS	505.00
Video Camera COGS	505.00

WHEN 句による階層メンバーの選択

次のリクエストでは、WHEN 句を使用して特定のアカウントを選択し、選択されたノードの 1 レベル上および 1 レベル下に値を挿入します。出力結果にはルートノードまでのレベルがすべて表示され、レベルの前後関係が示されますが、選択されたメンバーに属さないレベルがある場合、そのレベルのメジャー値は挿入されません。

```
TABLE FILE NEWGL
SUM GL_ACCOUNT_AMOUNT
BY GL_ACCOUNT_CAPTION HIERARCHY
WHEN GL_ACCOUNT GT '2000' AND GL_ACCOUNT LT '3000';
 SHOW UP 1 TO DOWN 1
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
TYPE=REPORT,GRID=OFF,$
ENDSTYLE
END
```

下図は、出力結果を示しています。

Caption	Amount
Profit Before Tax	.
Gross Margin	10,908.00
Sales Revenue	7,575.00
Retail Sales	2,525.00
Retail - Television	505.00
Retail - Stereo	505.00
Retail - Video Player	505.00
Retail - Computer	505.00
Retail - Video Camera	505.00
Mail Order Sales	2,525.00
Mail Order - Television	505.00
Mail Order - Stereo	505.00
Mail Order - Video Player	505.00
Mail Order - Computer	505.00
Mail Order - Video Camera	505.00
Internet Sales	2,525.00
Internet - Television	505.00
Internet - Stereo	505.00
Internet - Video Player	505.00
Internet - Computer	505.00
Internet - Video Camera	505.00
Cost Of Goods Sold	3,333.00
Variable Material Costs	2,525.00
Television COGS	505.00
Stereo COGS	505.00
Video COGS	505.00
Computer COGS	505.00
Video Camera COGS	505.00
Direct Labor	404.00
Fixed Costs	404.00

選択された階層メンバーの WHERE 句による選別

次のリクエストでは、WHEN 句を使用してメンバーを選択し、選択されたメンバーに WHERE 句を適用することで、出力結果を選別します。

```
TABLE FILE NEWGL
SUM GL_ACCOUNT_AMOUNT GL_ACCOUNT_TYPE
BY GL_ACCOUNT HIERARCHY
WHEN GL_ACCOUNT NE '3000';
 SHOW UP 0 TO DOWN 0
WHERE GL_ACCOUNT_TYPE NE 'E'   ;
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

下図は、出力結果を示しています。

Ledger Account	Amount	Type
1000	8,484.00	R
2000	7,575.00	R
2100	7,575.00	R
2200	2,525.00	R
2210	505.00	R
2220	505.00	R
2230	505.00	R
2240	505.00	R
2250	505.00	R
2300	2,525.00	R
2310	505.00	R
2320	505.00	R
2330	505.00	R
2340	505.00	R
2350	505.00	R
2400	2,525.00	R
2410	505.00	R
2420	505.00	R
2430	505.00	R
2440	505.00	R
2450	505.00	R
3000	. .	
3200	909.00	R
3600	404.00	R
3700	505.00	R
3730	505.00	R

レコードの選択

フィールドを選択してレポートを作成する場合、そのフィールドのすべてのインスタンスをレポートに表示させたくない場合があります。その場合は、選択条件を指定して、ユーザの要求を満たすフィールド値のみを表示させることができます。これにより、データのサブセットを選択できます。サブセットは、リクエストを発行するたびに、容易に再定義することができます。

レコード選択の概要

リクエストを作成する場合、さまざまな要因に基づいてレコードの選択条件を定義することができます。

- フィールドの値を個別に指定します。詳細は、特定値によるレコード選択を参照してください。
- フィールドの集計値を指定します (例、フィールド値の合計や平均)。詳細は、集計値による選択を参照してください。
- フィールドに存在するミッシング値、フィールドの値が存在する範囲、フィールドに存在しない値などを指定します。詳細は、レコード選択のタイプを参照してください。
- フィールドに存在するレコード数を指定します。フィールドの値ではなく、たとえば先頭の 50 件のレコードのみを指定するという方法です。詳細は、読み込みレコード数の制限を参照してください。
- グループキーを使用する、FOCUS 以外のデータソースでは、グループキーの値に基づいてレコードを選択します。詳細は、グループキー値による選択を参照してください。

また、フィルタ条件をデータソースに割り当てたり、ファイルから選択値を読み込んだりするなど、効率的にレコードを選択するためのさまざまな方法を活用することができます。

フィルタ条件の選択

レコードの選択には、WHERE と IF の 2 つの句を使用します。レコードを選択する場合は、WHERE の使用をお勧めします。IF では、WHERE の機能の一部が提供されます。IF で実行できることは、WHERE ですべて実行することができます。つまり、WHERE は IF で実行できないことも実行することができます。

これまで IF を使用してレコードを選択していた場合は、WHERE と IF は異なる句であるため、場合によっては同様の結果を得るために異なる構文が要求されることを理解しておく必要があります。

この章では、WHERE 構文について説明します。IF 構文についての詳細は、IF 句によるレコードの選択を参照してください。

特定値によるレコード選択

WHERE 句を使用して、レポートに表示するレコードをデータソースから選択します。データは、データソースから検索される前に選択条件に従って評価されます。

選択条件の定義には、必要な数の WHERE 句を任意に使用することができます。使用例については、Using Multiple WHERE Phrasesを参照してください。詳細は、複合式によるレコード選択を参照してください。

注意：マルチパスデータソースの異なるパスに存在するフィールド群で複数の選択テストを行う場合は、MULTIPATH パラメータの設定に基づいて、これらのフィールド群が AND または OR 演算子で結合されたように取り扱われます。詳細は、マルチパスデータソースのレコード選択制御を参照してください。

WHERE 句によるレコード選択

```
WHERE criteria [;]
```

説明

criteria

レポートに表示するレコードの選択条件です。この条件は、TRUE または FALSE で評価する有効な式 (つまり、ブール式) で定義する必要があります。式についての詳細は、式の使

用を参照してください。WHERE 句で使用できる演算子、たとえば CONTAINS、IS、GT についての詳細は、WHERE と IF で使用可能な演算子を参照してください。

;

オプションとして使用するセミコロンで、リクエストを読みやすくするために使用します。セミコロンはレポートに影響を与えません。

WHERE 句の使用上の注意

WHERE 句には、次のものを含めることができます。

- DEFINE 式の右側で有効となるほとんどの式。ただし、論理式の IF ... THEN ...ELSE は使用することができません。
- 実フィールド、一時項目、結合したファイルのフィールド。フィールド名が一重引用符 (') または二重引用符 (") で囲まれている場合、フィールド参照ではなくリテラル参照として扱われます。
- 演算子の EQ、NE、GE、GT、LT、LE、CONTAINS、OMITS、FROM ... TO、NOT-FROM ... TO、INCLUDES、EXCLUDES、LIKE、NOT LIKE 述語。
- すべての算術演算子 (+、-、*、/、**) および関数 (MIN、MAX、ABS、SQRT)。
- リテラルになることができる文字式。または、EDIT または DECODE を使用して結果として文字または数値が得られる関数。

  DECODE に使用するファイルには、2 つのフィールドを含めることができます。一方はフィールド値に、もう一方は数値デコード値に使用します。
- 一重引用符 (') で囲んだ文字と日付のリテラル、および DT フォーム (date-time literal) の日付時間リテラル。
- 日付リテラルを使用して日付フィールドに対する選択テストを行う場合、その日付リテラルに曜日の値を含めることはできません。
- テキストフィールド。ただし、テキストフィールドでサポートされる演算子は CONTAINS と OMITS のみです。
- すべての関数。

複雑な選択条件を作成するには、論理演算子の AND および OR を使用して単純な式を結合したり、オプションとして括弧を追加して評価順序を明示的に指定したりする方法があります。この方法は、同一の結果を得るために IF 句を使用するより簡単です。それは、IF 句を使用すると、DEFINE コマンドがさらに必要となる場合があるためです。詳細は、複合式によるレコード選択を参照してください。

単純な WHERE テストの使用

次のリクエストを発行して、年収が $20,000 を超える従業員のみの名前と給与額を表示します。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME AND FIRST_NAME AND CURR_SAL
BY LAST_NAME NOPRINT
WHERE CURR_SAL GT 20000
END
```

この例では、CURR_SAL フィールドを選択して、「CURR_SAL GT 20000」という選択条件を設定しています。現在の給与額が $20,000 を超えるレコードのみを取得し、その他のレコードはすべて除外します。

出力結果は次のとおりです。

LAST_NAME	FIRST_NAME	CURR_SAL
BANNING	JOHN	$29,700.00
BLACKWOOD	ROSEMARIE	$21,780.00
CROSS	BARBARA	$27,062.00
IRVING	JOAN	$26,862.00
ROMANS	ANTHONY	$21,120.00

複数の WHERE 句の使用

選択条件の定義には、必要な数の WHERE 句を任意に使用することができます。このリクエストでは、複数の WHERE 句を使用し、MIS 部または PRODUCTION 部の中で姓が Cross または Banning の従業員のみをレポートに表示します。

```
TABLE FILE EMPLOYEE
PRINT EMP_ID LAST_NAME
WHERE SALARY GT 20000
WHERE DEPARTMENT IS 'MIS' OR 'PRODUCTION'
WHERE LAST_NAME IS 'CROSS' OR 'BANNING'
END
```

出力結果は次のとおりです。

EMP_ID	LAST_NAME
119329144	BANNING
818692173	CROSS

詳細は、複合式によるレコード選択を参照してください。

分割 FOCUS データソースのレコードの選択

分割 FOCUS データソースからレポートを実行する際は、選択条件が分割データ配置のためフィールドの同一のフィールドに基づいている場合、関連付けられた分割データのみが検索用に開きます。

マルチパスデータソースのレコード選択制御

マルチパスデータソースからレポートを作成する際に、ある親セグメントのいくつかのパスには子セグメントが存在し、他のパスには子セグメントが存在しないという場合があります。MULTIPATH パラメータを使用すると、そのような親セグメントをレポート出力から除外するかどうかを制御することができます。

MULTIPATH の設定は、個々のパスで行う選択条件の処理にも影響を与えます。MULTIPATH を設定する際の注意点は次のとおりです。

- COMPOUND で WHERE、IF のいずれかを使用する場合、異なるパス上の選択条件は、これらの句が AND 演算子で結合されているように実行されます。つまり、レポート出力に親セグメントを含めるには、すべてのパスが選別テストの条件を満たす必要があります。
- SIMPLE で WHERE、IF のいずれかを使用する場合、異なるパス上の選択条件は、これらの句が OR 演算子で結合されているように独立して実行されます。そのため、少なくともパスの 1 つが選別テストを満たすと、親インスタンスがレポートに含まれます。リクエストがある 1 つのパスでの条件選択を含んでいる場合、他のパスからもデータを取得することを告げる警告メッセージが表示されます。条件選択を行ったパスで条件を満たすかどうかに関係なく、個別パスでレコードが取得されます。

MULTIPATH 設定は、すべてのデータソースタイプおよびすべてのレポート環境に適用されます (TABLE、TABLEF、MATCH、GRAPH、複数表示コマンドを使用するリクエスト)。また、MULTIPATH は、代替ビュー、インデックスビュー、フィルタ、DBA、JOIN 構造とも併用することができます。

マルチパスデータソースでのレコード選択の制御

コマンドレベルまたはストアドプロシジャから MULTIPATH を設定するには、次の構文を使用します。

```
SET MULTIPATH = {SIMPLE|COMPOUND}
```

リクエストで MULTIPATH を設定するには、次の構文を使用します。

```
ON TABLE SET MULTIPATH {SIMPLE|COMPOUND}
```

説明

SIMPLE

次の場合、レポート出力に親セグメントが表示されます。

- 親セグメントに属する子セグメントの中で、選別条件を満たす子セグメントが少なくとも 1 つ存在する。

  注意：ユニークセグメントは親セグメントの一部と見なされるため、個々のパス処理は呼び出されません。

- 親セグメントのパスに、参照された子セグメントは存在しないが、子セグメントがオプションで存在する。

  マルチパスレポートデータを選別する場合、警告メッセージ (FOC144) が表示されます。

  ```
  (FOC144) 警告。無関連のオカレンスのテストです
  ```

COMPOUND

必要となるすべての子セグメントが親セグメントに存在する場合、レポート出力に親セグメントが含められます。WHERE、IF のいずれかを使用する場合、異なるパス上の選択条件は、これらの句が AND 演算子で結合されているように実行されます。つまり、レポート出力に親セグメントを含めるには、すべてのパスが選別テストの条件を満たす必要があります。デフォルト値は COMPOUND です。

詳細は、MULTIPATH と SET ALL の組み合わせおよびセグメントの必須とオプションを決定する際の規則を参照してください。

MULTIPATH = COMPOUND の要件と使用上の注意

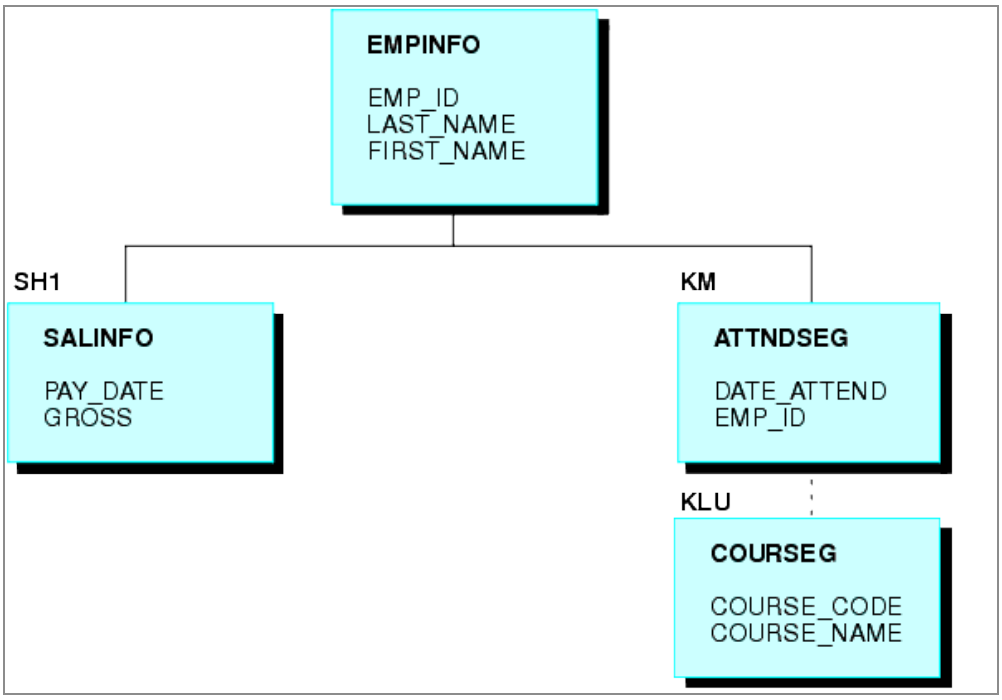
- MULTIPATH = COMPOUND の設定に必要な最小メモリは、有効なセグメント 1 つにつき 4 キロバイトです。メモリが不足すると、SIMPLE 設定が実行され、メッセージが表示されます。

  セグメントインスタンス (行) の数に制限はありませんが、1 つのセグメントインスタンスで、有効フィールドが 4 キロバイトを超えてはなりません (参照されたフィールドまたは参照フィールドを取得するために必要とされるフィールド)。この上限を超えると、SIMPLE 設定が実行され、メッセージが表示されます。

- OR 演算子を使用して複数パスで選別を行う WHERE 条件はサポートされていません。

複数パスからのデータ取得

この例では、EMPLOYEE データソースから次のセグメントを使用します。



次のリクエストは、MULTIPATH = SIMPLE を使用して両方のパスからデータを取得し、いずれかの条件を満たした場合にデータを表示します。

```
SET ALL = OFF
SET MULTIPATH = SIMPLE
TABLE FILE EMPLOYEE
PRINT GROSS DATE_ATTEND COURSE_NAME
BY LAST_NAME BY FIRST_NAME
WHERE PAY_DATE EQ 820730
WHERE COURSE_CODE EQ '103'
END
```

次の警告メッセージが表示されます。

```
(FOC144) 警告。無関連のオカレンスのテストです
```

コースをまったく受講しなかった従業員が何名か存在しますが、2 つのパスのうちいずれかにインスタンスが存在するため、その従業員の名前もレポートに表示されています。

出力結果は次のとおりです。

LAST_NAME	FIRST_NAME	GROSS	DATE_ATTEND	COURSE_NAME
BANNING	JOHN	$2,475.00	.	.
BLACKWOOD	ROSEMARIE	$1,815.00	.	.
CROSS	BARBARA	$2,255.00	.	.
GREENSPAN	MARY	$750.00	.	.
IRVING	JOAN	$2,238.50	.	.
JONES	DIANE	$1,540.00	82/05/26	BASIC REPORT PREP FOR PROG
MCCOY	JOHN	$1,540.00	.	.
MCKNIGHT	ROGER	$1,342.00	.	.
ROMANS	ANTHONY	$1,760.00	.	.
SMITH	MARY	$1,100.00	81/11/16	BASIC REPORT PREP FOR PROG
	RICHARD	$791.67	.	.
STEVENS	ALFRED	$916.67	.	.

MULTIPATH = COMPOUND を設定して上記と同一のリクエストを実行すると、COURSE_NAME に対してインスタンスが存在しない従業員はレポート出力から除外され、警告メッセージは表示されません。

出力結果は次のとおりです。

```
LAST_NAME       FIRST_NAME           GROSS  DATE_ATTEND  COURSE_NAME
---------       ----------           -----  -----------  -----------
JONES           DIANE            $1,540.00     82/05/26  BASIC REPORT PREP
FOR PROG
SMITH           MARY             $1,100.00     81/11/16  BASIC REPORT PREP
FOR PROG
```

MULTIPATH と SET ALL の組み合わせ

ALL パラメータは、個々のパスの処理に影響を与えます。下表は、EMPLOYEE データソースの例を使用して、ALL と MULTIPATH との関係について説明しています。

リクエスト	MULTIPATH=SIMPLE	MULTIPATH=COMPOUND

`SET ALL = OFF`
`PRINT EMP_ID`
`PAY_DATE DATE_`
`ATTEND`
	SALINFO または ATTNDSEG のいずれかのデータが存在する従業員を表示します。	SALINFO および ATTNDSEG の両方のデータが存在する従業員を表示します。

リクエスト	MULTIPATH=SIMPLE	MULTIPATH=COMPOUND

`SET ALL = ON`
`PRINT EMP_ID`
`PAY_DATE DATE_`
`ATTEND`
	SALINFO または ATTNDSEG のいずれかのデータが存在するか、子データがまったく存在しない従業員を表示します。	SIMPLE の場合と同一です。

`SET ALL = OFF`
`PRINT EMP_ID`
`PAY_DATE DATE_`
`ATTEND`
`WHERE PAY_DATE`
`EQ 980115`
	980115 に適合する SALINFO データまたは何らかの ATTNDSEG データが存在する従業員を表示します。
	メッセージ (FOC144) が表示されます。
		980115 に適合する SALINFO データおよび ATTNDSEG データの両方が存在する従業員を表示します。

`SET ALL = ON`
`PRINT EMP_ID`
`PAY_DATE DATE_`
`ATTEND`
`WHERE PAY_DATE`
`EQ 980115`
	980115 に適合する SALINFO データまたは何らかの ATTNDSEG データが存在する従業員を表示します。
	メッセージ (FOC144) が表示されます。
		980115 に適合する SALINFO データが存在する従業員を表示します。DATE_ATTEND データも表示されます。

`SET ALL = OFF`
`PRINT ALL.EMP_`
`ID DATE_ATTEND`
`WHERE PAY_DATE`
`EQ 980115`
	980115 に適合する SALINFO データまたは何らかの ATTNDSEG データが存在する従業員を表示します。
	メッセージ (FOC144) が表示されます。
		980115 に適合する SALINFO データが存在する従業員を表示します。DATE_ATTEND データも表示されます。

`SET ALL = ON または OFF`
`PRINT EMP_ID`
`PAY_DATE DATE_`
`ATTEND`
	980115 に適合する SALINFO データまたは COURSE 103 のいずれかが存在する従業員を表示します。	980115 に適合する SALINFO データおよび COURSE 103 の両方が存在する従業員を表示します。

リクエスト	MULTIPATH=SIMPLE	MULTIPATH=COMPOUND
`WHERE PAY_DATE EQ 980115 AND COURSE_CODE EQ '103'`	**注意：**WHERE 句の AND は、SIMPLE により OR として処理されます。 メッセージ (FOC144) が表示されます。	

注意：MULTIPATH = COMPOUND では、SET ALL = PASS はサポートされません。

ALL パラメータについての詳細は、ミッシングフィールド値を含むレコード処理を参照してください。

セグメントの必須とオプションを決定する際の規則

セグメントの規則は、データソース階層またはビュー階層においてレベルごとに降順に適用されます。つまり、データソースのツリー全体では、親セグメントの存在は子セグメントの存在の上に、子セグメントの存在は孫セグメントの存在の上にそれぞれ成立します。

セグメントが必須かオプションかを決定する場合、次の規則が適用されます。

- SET ALL が ON または OFF の場合、WHERE または IF 条件で指定されたセグメントはその親セグメントに対して必須になり、ルートセグメントまでのすべてのセグメントがそれぞれの親セグメントに対して必須になります。

  SET ALL = PASS の場合、WHERE または IF 条件で指定されたセグメントはオプションになります。

- SET ALL = ON または PASS の場合、WHERE や IF 条件が指定されていないすべての参照セグメントは、その親セグメントに対してオプションになります (OUTER JOIN)。
- SET ALL = OFF の場合、すべての参照セグメントが必須になります (INNER JOIN)。
- 親セグメントにフィールド演算接頭語の ALL. を使用する場合は、参照セグメントをオプションにすることもできます。

注意：アダプタの中には、ALL = PASS をサポートしないものがあります。サポートされている場合でも、動作が多少異なる場合があります。詳細は、該当するデータアダプタのマニュアルを参照してください。

ALL パラメータについての詳細は、ミッシングフィールド値を含むレコード処理を参照してください。

集計値による選択

フィールドの集計値に基づいてレコードを選択することができます。たとえば、WHERE TOTAL 句を使用することで、フィールド値の合計や平均に基づいてレコードを選択できます。WHERE TOTAL は、集計表示コマンドの SUM および COUNT を使用する場合に役立ちます。また、AVE. や PCT. などの演算接頭語が指定されたフィールドにはこの句を使用する必要があります。

WHERE では、データが評価された後でデータが取得されます。逆に、WHERE TOTAL テストでは、すべてのデータが取得、処理された後でそのデータが選択されます。この例は、WHERE TOTAL によるレコードの選択を参照してください。

WHERE TOTAL によるレコードの選択

```
WHERE TOTAL criteria[;]
```

説明

criteria

レポートに表示するレコードの選択条件です。この条件は、TRUE または FALSE で評価する有効な式 (つまり、ブール式) で定義する必要があります。式についての詳細は、式の使用を参照してください。WHERE 句で使用できる演算子 (例、IS、GT)。詳細は、WHERE と IF で使用可能な演算子を参照してください。

;

オプションとして使用するセミコロンで、リクエストを読みやすくするために使用します。セミコロンはレポートに影響を与えません。

WHERE TOTAL 使用上の注意

- 計算した値を参照したり、TOT.field や AVE.field などの値を集計する機能を使用したりする場合には、WHERE TOTAL を使用する必要があります。
- 演算接頭語を伴うフィールドには、WHERE TOTAL を使用する必要があります。
- WHERE TOTAL は、最下位のソートレベルで実行されます。
- 文字および日付のリテラルは、一重引用符 (') で囲む必要があります。日時リテラルは、DT (date-time literal) フォームに含める必要があります。
- WHERE TOTAL とともに ACROSS を使用する場合、選択条件を満たしていないデータは、「NODATA」という文字でレポートに表示されます。
- リクエストの出力を HOLD ファイルに保存する場合、WHERE TOTAL テストにより「WH$$$T1」という内部計算を含むフィールドが作成されます。複数の WHERE TOTAL テストがある場合、それぞれの TOTAL テストに連続した番号が付けられて WH$$$T フィールドが作成されます。

WHERE TOTAL によるレコードの選択

次の例では、現在の給与額を部署別に合計します。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL
BY DEPARTMENT
END
```

出力結果は次のとおりです。

```
DEPARTMENT         CURR_SAL
----------         --------
MIS             $108,002.00
PRODUCTION      $114,282.00
```

ここで、このリクエストに WHERE TOTAL 句を追加し、給与の総額が $110,000 を超える部署のみを表示するレポートを作成します。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL
```

```
BY DEPARTMENT
WHERE TOTAL CURR_SAL EXCEEDS 110000
END
```

部署別に値を計算し、計算結果の値を $110,000 と比較します。出力結果は次のとおりです。

```
DEPARTMENT         CURR_SAL
----------         --------
PRODUCTION      $114,282.00
```

WHERE TOTAL と WHERE の組み合わせによるレコードの選択

次のリクエストでは、MIS 部署のレコードが抽出されます。次に、各従業員の CURR_SAL の値を合計します。従業員の給与額が $20,000 を超える場合に、CURR_SAL の値が処理されてレポートに表示されます。つまり、WHERE TOTAL によるデータの選別は、レコードが選択された後で行われます。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL
BY LAST_NAME AND BY FIRST_NAME
WHERE TOTAL CURR_SAL EXCEEDS 20000
WHERE DEPARTMENT IS 'MIS'
END
```

出力結果は次のとおりです。

```
LAST_NAME        FIRST_NAME         CURR_SAL
---------        ----------         --------
BLACKWOOD        ROSEMARIE        $21,780.00
CROSS            BARBARA          $27,062.00
```

COMPUTE 処理前の内部マトリックスへの選択条件の適用

WHERE TOTAL テストは、レポートの出力段階で COMPUTE 演算が処理された後に、内部マトリックスの行に適用されます。WHERE_GROUPED テストは、COMPUTE 演算が処理される前に、内部マトリックスに適用されます。次に COMPUTE 演算が処理され、続いて WHERE TOTAL テストが適用されます。これにより、開発者が評価順序を制御することができます。この方法は、特に再帰的演算を使用する場合に役立ちます。

WHERE_GROUPED 選択条件の適用

```
WHERE_GROUPED expression
```

説明

expression

内部マトリックスの 1 行のみを参照する式 (複数行を参照しない式) です。たとえば、式に LAST 演算子を使用して、前レコードの値を参照または取得することはできません。

WHERE_GROUPED テストの使用

次のリクエストには、2 つの COMPUTE コマンドが使用されています。1 つ目の COMPUTE は、ビジネス地方区分の値が変化したかどうかを確認し、変化していない場合はカウンタの値を 1 つ増加させます。これにより、マトリックス内のレコードが順序どおりに配列されます。2 つ目の COMPUTE は、ビジネス地方区分ごとの遅延日数のローリング合計を作成します。

```
TABLE FILE WF_RETAIL_LITE
SUM  DAYSDELAYED AS DAYS
COMPUTE CTR/I3 = IF BUSINESS_REGION EQ LAST BUSINESS_REGION THEN CTR+1 ELSE
1;
COMPUTE NEWDAYS = IF BUSINESS_REGION EQ LAST BUSINESS_REGION THEN
NEWDAYS+DAYSDELAYED ELSE DAYSDELAYED;
BY BUSINESS_REGION AS Region
BY TIME_MTH
```

```
WHERE BUSINESS_REGION NE 'Oceania'
ON TABLE SET PAGE NOPAGE
END
```

出力結果は次のとおりです。

Region	Sale Month	DAYS	CTR	NEWDAYS
EMEA	1	191	1.00	191.00
	2	205	2.00	396.00
	3	224	3.00	620.00
	4	213	4.00	833.00
	5	185	5.00	1,018.00
	6	234	6.00	1,252.00
	7	223	7.00	1,475.00
	8	234	8.00	1,709.00
	9	249	9.00	1,958.00
	10	290	10.00	2,248.00
	11	1	11.00	2,249.00
North America	1	253	1.00	253.00
	2	205	2.00	458.00
	3	293	3.00	751.00
	4	319	4.00	1,070.00
	5	273	5.00	1,343.00
	6	268	6.00	1,611.00
	7	219	7.00	1,830.00
	8	248	8.00	2,078.00
	9	220	9.00	2,298.00
	10	326	10.00	2,624.00
	11	8	11.00	2,632.00
South America	1	64	1.00	64.00
	2	87	2.00	151.00
	3	29	3.00	180.00
	4	39	4.00	219.00
	5	29	5.00	248.00
	6	49	6.00	297.00
	7	16	7.00	313.00
	8	26	8.00	339.00
	9	54	9.00	393.00
	10	54	10.00	447.00

次のリクエストでは、WHERE TOTAL テストを追加し、DAYSDELAYED が 200 日を超えた月のみを選択します。

```
TABLE FILE WF_RETAIL_LITE
SUM  DAYSDELAYED AS DAYS
COMPUTE CTR/I3 = IF BUSINESS_REGION EQ LAST BUSINESS_REGION THEN CTR+1 ELSE
1;
COMPUTE NEWDAYS= IF BUSINESS_REGION EQ LAST BUSINESS_REGION THEN
NEWDAYS+DAYSDELAYED ELSE DAYSDELAYED;
BY BUSINESS_REGION AS Region
BY TIME_MTH
WHERE BUSINESS_REGION NE 'Oceania'
WHERE TOTAL DAYSDELAYED GT 200
ON TABLE SET PAGE NOPAGE
END
```

出力結果は次のとおりです。CTR および NEWDAYS の COMPUTE 演算は、TOTAL DAYSDELAYED が 200 日以下の行が削除される前に処理されたため、これらの値は元の出力結果と同一になります。この出力結果では、レコードの連続番号が反映されず、出力結果に実際に表示されている値のローリング合計が正しくありません。正しく出力するには、両方の COMPUTE 式が評価される前に、適切な月 (DAYSDELAYED GT 200) を選択する必要があります。この場合、WHERE_GROUPED を使用する必要があります。

Region	Sale Month	DAYS	CTR	NEWDAYS
EMEA	2	205	2.00	396.00
	3	224	3.00	620.00
	4	213	4.00	833.00
	6	234	6.00	1,252.00
	7	223	7.00	1,475.00
	8	234	8.00	1,709.00
	9	249	9.00	1,958.00
	10	290	10.00	2,248.00
North America	1	253	1.00	253.00
	2	205	2.00	458.00
	3	293	3.00	751.00
	4	319	4.00	1,070.00
	5	273	5.00	1,343.00
	6	268	6.00	1,611.00
	7	219	7.00	1,830.00
	8	248	8.00	2,078.00
	9	220	9.00	2,298.00
	10	326	10.00	2,624.00

次のリクエストでは、WHERE TOTAL テストの代わりに、WHERE_GROUPED テストを使用します。

```
TABLE FILE WF_RETAIL_LITE
SUM  DAYSDELAYED AS DAYS
COMPUTE CTR/I3 = IF BUSINESS_REGION EQ LAST BUSINESS_REGION THEN CTR+1 ELSE
1;
COMPUTE NEWDAYS= IF BUSINESS_REGION EQ LAST BUSINESS_REGION THEN
NEWDAYS+DAYSDELAYED ELSE DAYSDELAYED;
BY BUSINESS_REGION AS Region
BY TIME_MTH
WHERE BUSINESS_REGION NE 'Oceania'
WHERE_GROUPED DAYSDELAYED GT 200
ON TABLE SET PAGE NOPAGE
END
```

出力結果は次のとおりです。NEWDAYS の COMPUTE 演算は、TOTAL DAYSDELAYED が 200 日以下の行が削除された後に処理されたため、NEWDAYS の値は、元のリクエストの演算より少ない行に基づいて計算されています。このことは、CTR 値が連続番号で表示されていることで確認できます。ローリング合計には、レポート出力に実際に表示されている値が反映されています。

Region	Sale Month	DAYS	CTR	NEWDAYS
EMEA	2	205	1.00	205.00
	3	224	2.00	429.00
	4	213	3.00	642.00
	6	234	4.00	876.00
	7	223	5.00	1,099.00
	8	234	6.00	1,333.00
	9	249	7.00	1,582.00
	10	290	8.00	1,872.00
North America	1	253	1.00	253.00
	2	205	2.00	458.00
	3	293	3.00	751.00
	4	319	4.00	1,070.00
	5	273	5.00	1,343.00
	6	268	6.00	1,611.00
	7	219	7.00	1,830.00
	8	248	8.00	2,078.00
	9	220	9.00	2,298.00
	10	326	10.00	2,624.00

WHERE_GROUPED 使用上の注意

- 内部マトリックスの複数行を参照する式の場合、次のメッセージが生成され、処理が停止します。

  ```
  (FOC32692) WHERE_GROUPED でレポートの別の行を参照することはできません。
  ```

- 複数行を参照しない COMPUTE は、WHERE_GROUPED テストの前に評価されるため、その COMPUTE を式で使用することも、WHERE_GROUPED テストの一部として評価することもできます。
- WHERE_GROUPED を SQL データソース用に最適化するには、GROUP BY *fieldname* HAVING *expression* 句を作成します。ここで、expression は WHERE_GROUPED 選択条件を表します。

複合式によるレコード選択

演算子の AND と OR のいずれかまたはその両方を使用して、いくつかの単純な WHERE 式を組み合わせて複合式を作成することができます。

デフォルト設定では、複数の WHERE 句を評価する場合、論理 AND が論理 OR より先に処理されます。複合式では、括弧を使用して評価順序を変更することができます。括弧内にあるすべての AND および OR 演算子が、括弧外の AND および OR 演算子より先に評価されます。

複合式では、式が正しい順序で評価されるよう、括弧を適宜使用する必要があります。以下はその例です。

```
WHERE (SEATS EQ 2) AND (SEATS NOT-FROM 3 TO 4)
```

この方法は論理 AND または論理 OR にリテラル OR を組み合わせて使用する場合に特に役立ちます。

- 論理 AND または論理 OR では、フィールドのすべての名前、関係、値が、「OR」または「AND」という語句により明示的に参照され、結合されます。以下はその例です。

  ```
  WHERE (LAST_NAME EQ 'CROSS') OR (LAST_NAME EQ 'JONES')
  ```

  または

```
WHERE (CURR_SAL GT 20000) AND (DEPARTMENT IS 'MIS')
       AND (CURR_JOBCODE CONTAINS 'A')
```

- リテラル OR では、「OR」という語句は 1 つのフィールド名のテスト値の間では繰り返し使用されますが、フィールド名自体および結合用の関係演算子は繰り返し使用されません。以下はその例です。

  ```
  WHERE (LAST_NAME EQ 'CROSS' OR 'JONES')
  ```

AND と OR の組み合わせによるレコード選択

次の例では、リテラル OR と論理 AND の評価において括弧がどのような影響を及ぼすかを示しています。

このリクエストでは、括弧で囲まれた式が、それぞれ記述された順序で先に評価されます。ここでは、最初の式にリテラル OR が含まれています。次に、それぞれの式の結果が論理 AND で評価されます。

括弧をはずした場合、論理 AND がリテラル OR より先に評価されます。

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL BY LAST_NAME
WHERE (LAST_NAME EQ 'CROSS' OR 'JONES')
AND (CURR_SAL GT 22000)
END
```

出力結果は次のとおりです。

```
LAST_NAME               CURR_SAL
---------               --------
CROSS                 $27,062.00
```

演算子によるレコード選択

WHERE と IF の選択には、さまざなまな演算子を使用することができます。多くの演算子が WHERE と IF に共通して使用できますが、その中のいくつかは WHERE でのみサポートされます。

WHERE と IF で使用可能な演算子

次の演算子を使用して、WHERE および IF の選択条件を定義することができます。

WHERE 演算子	IF 演算子	説明
EQ IS	EQ IS	テスト値と等しい値を選択します。
NE IS-NOT	NE IS-NOT	テスト値と等しくない値を選択します。
GE	GE FROM IS-FROM	テスト値以上の値を選択します (数値では 0 から 9 まで、文字では A から Z、a から z までの値に基づく)。 テスト値は、フィールド値にすることも、式の結果にすることもできます。
GT EXCEEDS IS-MORE-THAN	GT EXCEEDS IS-MORE-THAN	テスト値より大きい値を選択します。
LT IS-LESS-THAN	LT IS-LESS-THAN	テスト値より小さい値を選択します。
LE	LE TO による範囲	テスト値以下の値を選択します。
GE lower AND ...		テスト値の範囲内にある値を選択します。

WHERE 演算子	IF 演算子	説明
`LE upper`		
`LT lower OR ...GT upper`		テスト値の範囲外にある値を選択します。
`FROM lower TO upper`		テスト値の範囲内にある値を選択します。
`IS-FROM lower TO upper`	`IS-FROM lower TO upper`	テスト値の範囲内にある値を選択します。WHERE では、FROM lower TO upper の代わりにこの構文を使用することができます。いずれの演算子も同一の結果となります。
`NOT-FROM lower TO upper`	`NOT-FROM lower TO upper`	テスト値の範囲外にある値を選択します。
`IS MISSING IS-NOT MISSING NE MISSING`	`IS MISSING IS-NOT MISSING NE MISSING`	フィールドに存在するミッシング値の有無をテストします。フィールドのインスタンスの中でデータが存在しないものがあれば、ミッシングデータがあると判断されます。ミッシングデータについての詳細は、ミッシングフィールド値を含むレコード処理を参照してください。
`CONTAINS LIKE`	`CONTAINS LIKE`	テスト値に一致する文字列を含む値を選択します。この文字列は、テスト値の任意の位置に存在することができます。CONTAINS は、

WHERE 演算子	IF 演算子	説明
		WHERE とともに使用した場合は文字フィールドのテストを行うことができます。また、IF とともに使用した場合は文字およびテキストの両フィールドのテストを行うことができます。
`OMITS` `NOT LIKE`	`OMITS` `UNLIKE[UNLIKE]`	テスト値と一致する文字列を含まない値を選択します。この文字列は、テスト値のどの位置にも存在することはできません。OMITS は、WHERE とともに使用した場合は文字フィールドのテストを行うことができます。また、IF とともに使用した場合は文字およびテキストの両フィールドのテストを行うことができます。
`INCLUDES`	`INCLUDES`	子セグメントにあるフィールドの一連の値が、リテラルリストにすべて含まれているかどうかをテストします。
`EXCLUDES`	`EXCLUDES`	子セグメントにあるフィールドの一連の値が、リテラルリストにすべて含まれていないかどうかをテストします。
`IN (z,x,y)`		順序不同のリストに存在する値に基づいてレコードを選択します。
`NOT ...IN` `(z,x,y)`		順序不同のリストに存在しない値に基づいてレコードを選択します。
`IN FILE`		シーケンシャルファイルに保存された値に基づいてレコードを選択

WHERE 演算子	IF 演算子	説明
		します。
`NOT ...IN FILE`		シーケンシャルファイルに保存されていないフィールド値を含むレコードを選択します。
`IF-THEN-ELSE`		IF-THEN-ELSE 句で記述された論理条件に基づいてレコードを選択します。

演算子によるフィールドと値の比較

次の例は、1 つまたは複数の値を使用して設定したフィールド選択条件を示しています。使用できる演算子には、EQ、IS、IS-NOT、EXCEEDS、IS-LESS-THAN、IN があります。

例 1：LAST_NAME フィールドの値が JONES と等しいことが条件です。

```
WHERE LAST_NAME EQ 'JONES'
```

例 2：LAST_NAME フィールドの値が「CR」または「MC」という語句で始まることが条件です。

```
WHERE EDIT (LAST_NAME, '99') EQ 'CR' OR 'MC'
```

例 3：AREA フィールドの値が EAST または WEST 以外であることが条件です。

```
WHERE AREA IS-NOT 'EAST' OR 'WEST'
```

例 4：AREA フィールドの値が REGION フィールドの値と等しいことが条件です。

```
WHERE AREA EQ REGION
```

IF では、1 つのフィールドを別のフィールドと比較することはできません。

例 5：RETAIL_COST と DEALER_COST の割合が 1.25 より大きいことが条件です。

```
WHERE RETAIL_COST/DEALER_COST GT 1.25
```

例 6：UNITS フィールドの値が 50 以下であり、AREA が NORTH EAST または WEST 以外であることが条件です。ここでは、NORTH EAST を一重引用符 (') で囲んでいます。すべての文字列は、一重引用符 (') で囲む必要があります。

```
WHERE UNITS LE 50 WHERE AREA IS-NOT 'NORTH EAST' OR 'WEST'
```

例 7：AMOUNT の値が 40 より大きいことが条件です。

```
WHERE AMOUNT EXCEEDS 40
```

例 8：AMOUNT の値が 50 より小さいことが条件です。

```
WHERE AMOUNT IS-LESS-THAN 50
```

例 9：SALES の値が、順序不同のリストに存在する数値のいずれかと等しいことが条件です。リストの値を区切るには、カンマ (,) またはブランクを使用します。

```
WHERE SALES IN (43000,12000,13000)
```

例 10：CAR の値が、順序不同のリストの文字のいずれかと等しいことが条件です。文字リストの値は、一重引用符 (') で囲む必要があります。

```
WHERE CAR IN ('JENSEN','JAGUAR')
```

WHERE 句での IF-THEN-ELSE ロジックの使用

次のリクエストでは、WHERE 句に IF-THEN-ELSE ロジックを使用し、WHOLESALEPR 値に基づいてレコードを選択します。この場合、選択条件に使用される値は、そのレコードの LISTPR 値に応じて変わります。

```
TABLE FILE MOVIES
PRINT COPIES
LISTPR
WHOLESALEPR
BY CATEGORY
WHERE WHOLESALEPR GT (IF LISTPR GT 20.00 THEN 15.00 ELSE 11.00)
```

```
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。選択された各レコードでは、LISTPR が $20.00 より大きい場合、「WHOLESALEPR が $15.00 より大きい」が選択条件になります。LISTPR が $20.00 以下の場合、「WHOLESALEPR が $11.00 より大きい」が選択条件になります。

CATEGORY	COPIES	LISTPR	WHOLESALEPR
ACTION	3	19.98	11.50
	4	19.99	11.99
CHILDREN	2	44.95	29.99
	1	29.95	15.99
CLASSIC	3	39.99	20.00
	1	29.95	15.99
	3	89.95	40.99
	2	89.99	40.99
COMEDY	4	19.95	12.55
	2	19.95	12.55
	3	59.99	30.00
	4	19.98	13.75
	4	19.98	13.75
FOREIGN	1	19.98	13.25
	1	59.95	30.00
	1	29.95	15.99
	1	39.98	25.00
	1	59.95	32.00
	1	59.95	30.00
MUSICALS	1	19.98	13.99
	1	19.98	13.99
MYSTERY	4	29.98	15.99
	2	25.99	15.99
	4	59.99	30.00
SCI/FI	1	19.98	14.99
	1	19.98	14.55
	1	19.95	13.99
	1	24.95	16.00
	3	29.98	19.99

変数によるレコード選択

次の例では、WHERE テストの変数として REGION フィールドを使用します。これにより、ユーザがレポートを作成する際に、REGION フィールドの値をリスト値 (CE、CORP、NE、SE、WE) から選択するよう指示されます。値の後に続けて記述したテキストが、出力結果のドロップダウンリストの先頭に表示されます。

```
TABLE FILE EMPDATA
SUM SALARY
BY DIV
BY DEPT
HEADING
"Current Salary Report"
"for the &REGION Division"
" "
WHERE ( DIV EQ
'&REGION.(CE,CORP,NE,SE,WE).Please select a Region.');
END
```

出力結果は次のとおりです。

ドロップダウンリストから地域を選択し、[実行] をクリックします。NE 地域を選択した場合の出力結果は次のとおりです。

Current Salary Report
for the NE Division

DIV	DEPT	SALARY
NE	CUSTOMER SUPPORT	$81,800.00
	MARKETING	$139,800.00
	SALES	$82,600.00

レコード選択のタイプ

レポートに表示するレコードを選択する場合、演算子を使用してさまざまなタイプのテストを実行することができます。詳細は、WHERE と IF で使用可能な演算子を参照してください。実行可能なテストには次のものがあります。

- 範囲内または範囲外に存在する値。詳細は、FROM ... TO による範囲および GE ... LE または GT ... LT による範囲を参照してください。
- ミッシングデータまたはミッシングではないデータ。詳細は、ミッシングデータのテストを参照してください。
- 文字列の有無。詳細は、CONTAINS と OMITS による文字列の選別を参照してください。
- データフィールドで部分的に定義された文字列。詳細は、マスクフィールドによる選別を参照してください。
- 親セグメント内のリテラル。詳細は、INCLUDES と EXCLUDES による親セグメントの選択を参照してください。

FROM ... TO による範囲

演算子の FROM ... TO および NOT-FROM ... を使用して、フィールドの値が所定の範囲内または範囲外に存在するかどうかをテストします。範囲の上限値と下限値を指定するには、値または式を使用します。範囲テストは、ソートフィールドにも適用することができます。この範囲テストは、ソート句のすぐ後に指定します。

式が範囲内または範囲外で失敗するかどうかをテストすることもできます。

範囲の指定 (FROM ... TO)

```
WHERE [TOTAL] {fieldname|expression} {FROM|IS-FROM} lower TO upper
WHERE [TOTAL] fieldname  NOT-FROM      lower TO upper
```

説明

fieldname

任意の有効フィールド名またはエイリアスです。

expression

任意の有効な式です。

lower

範囲の下限を表す数値、文字、式のいずれかです。式を括弧で囲んで、読みやすくすることもできます。

upper

範囲の上限を表す数値、文字、式のいずれかです。式を括弧で囲んで、読みやすくすることもできます。

FROM ... TO による範囲

次の例では、範囲に式を使用して範囲テストを行います。

```
WHERE SALES FROM (DEALER_COST * 1.4) TO (DEALER_COST * 2.0)
```

次の例では、比較値と範囲の両方に式を使用して範囲テストを行います。

```
WHERE SALES * 1.5 FROM (DEALER_COST * 1.4) TO (DEALER_COST * 2.0)
```

NOT-FROM ... TO による範囲

次の例は、範囲テストの NOT-FROM ... TO を使用して、指定した範囲外にあるレコードのみを表示する方法を示しています。この例では、給与額が $12,000 から $22,000 の範囲に該当

しない従業員を表示します。

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL
BY LAST_NAME
WHERE CURR_SAL NOT-FROM 12000 TO 22000
END
```

出力結果は次のとおりです。

```
LAST_NAME               CURR_SAL
---------               --------
BANNING               $29,700.00
CROSS                 $27,062.00
GREENSPAN              $9,000.00
IRVING                $26,862.00
SMITH                  $9,500.00
STEVENS               $11,000.00
```

FROM ... TO によるソートフィールドの範囲

次の例では、ソート句の BY または ACROSS を使用してフィールドをソートする際に、範囲テストを実行します。

```
BY MONTH FROM 4 TO 8
```

または

```
ACROSS MONTH FROM 6 TO 10
```

GE ... LE または GT ... LT による範囲

演算子の GE (以上)、LE (以下)、GT (より大きい)、LT (より小さい) のいずれかを使用して範囲を指定することができます。

GE ... LE を使用すると、所定の範囲以内の値を指定することができます。

LT ...GT を使用すると、所定の範囲より外にある値を指定することができます。

範囲の指定 (GE ... LE)

特定の範囲内にある値を選択するには、次の構文を使用します。

```
WHERE fieldname GE lower AND fieldname LE upper
```

所定の範囲内に存在しない値を含んだレコードを検索するには、次の構文を使用します。

```
WHERE fieldname LT lower OR fieldname GT upper
```

説明

fieldname

任意の有効フィールド名またはエイリアスです。

lower

範囲の下限を表す数値、文字、式のいずれかです。式を括弧で囲んで、読みやすくすることもできます。

upper

範囲の上限を表す数値、文字、式のいずれかです。式を括弧で囲んで、読みやすくすることもできます。

範囲内の値の選択

次の WHERE 句は、UNITS の値が 10,000 から 14,000 の範囲に存在するレコードを選択します。

```
WHERE UNITS GE 10000 AND UNITS LE 14000
```

上記の構文は、次のように記述することもできます。

```
WHERE UNITS GE 10000
WHERE UNITS LE 14000
```

範囲外の値の選択

次の例は、演算子の LT と GT を使用して、指定した値の範囲外に存在する値を選択する方法を示しています。この例では、給与額が $12,000 より小さい従業員および $22,000 より大きい従業員のみを出力結果に表示します。

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL
BY LAST_NAME
WHERE CURR_SAL LT 12000 OR CURR_SAL GT 22000
END
```

出力結果は次のとおりです。

```
LAST_NAME               CURR_SAL
---------               --------
BANNING               $29,700.00
CROSS                 $27,062.00
GREENSPAN              $9,000.00
IRVING                $26,862.00
SMITH                  $9,500.00
STEVENS               $11,000.00
```

ミッシングデータのテスト

リクエストを作成する際に、ミッシングデータをテストしておきたい場合があります。このタイプのテストは、マスターファイルでミッシングデータを含むフィールドの MISSING 属性が 「ON」に設定されているときに役立ちます。 ミッシングデータについての詳細は、ミッシングフィールド値を含むレコード処理を参照してください。

注意：文字フィールドの選別に使用するテスト値が変数で、ミッシングインスタンスを検索する必要がある場合、MISSING ではなく _FOC_MISSING を使用します。これは、テストでは文字リテラル値を一重引用符で囲む必要があるためで、'MISSING' はリテラル値の MISSING であり、ミッシング値ではないためです。_FOC_MISSING の値は、一重引用符に囲まれる場合も囲まれない場合も、ミッシング値 (MISSING) を表します。

ミッシングデータのテスト

```
{WHERE|IF} fieldname {EQ|IS} MISSING
```

説明

fieldname

任意の有効フィールド名またはエイリアスです。

EQ|IS

レコード選択の演算子です。EQ と IS は同義です。

ミッシングでないデータのテスト

```
{WHERE|IF} fieldname {NE|IS-NOT} MISSING
```

説明

fieldname

任意の有効フィールド名またはエイリアスです。

NE|IS-NOT

レコード選択の演算子です。NE と IS-NOT は同義です。

CONTAINS と OMITS による文字列の選別

演算子の CONTAINS および OMITS は、WHERE とともに使用して文字フィールドをテストしたり、IF とともに使用して文字とテキストフィールドの両方をテストしたりする場合に使用します。CONTAINS を使用する場合、1 つまたは複数のリテラルがフィールド値の文字内の任意の場所に存在する場合、そのテストの条件が満たされたことになります。

OMITS は、CONTAINS とは逆に機能します。1 つまたは複数のリテラルがフィールド値の文字内の任意の場所に存在する場合、そのテストの条件が満たされないことになります。

CONTAINS と OMITS のテストは、正確な値の綴りが分からない場合に役立ちます。特定の文字列が値の中に存在することが分かっている限り、そのデータを取得することができます。

CONTAINS と OMITS によるレコードの選択

次の例は、演算子の CONTAINS と OMITS を使用したさまざまな方法を示しています。テストするフィールド名は、演算子の CONTAINS または OMITS の左側に配置する必要があります。

- この例では、次の句により「JOHN」という文字列を含む、たとえば「JOHNSON」という文字が選択されます。

  ```
  WHERE LAST_NAME CONTAINS 'JOHN'
  ```

  この場合、「JOHN」という文字列は、LAST_NAME フィールドの任意の場所に存在することができます。

- 次の例では、「JOHN」という文字列を含まない姓がすべて選択されます。

  ```
  WHERE LAST_NAME OMITS 'JOHN'
  ```

- 次の例では、「ING」という文字を含むすべての名前が取得されます。

  ```
  TABLE FILE EMPLOYEE
  LIST LAST_NAME AND FIRST_NAME
  WHERE LAST_NAME CONTAINS 'ING'
  END
  ```

  出力結果は次のとおりです。

  ```
  LIST  LAST_NAME        FIRST_NAME
  ----  ---------        ----------
     1  BANNING          JOHN
     2  IRVING           JOAN
  ```

マスクフィールドによる選別

マスクとは、値の比較を行うために使用する文字のパターンのことです。データフィールドは、文字フォーマット (A) でなければなりません。演算子の LIKE と NOT LIKE または IS と IS-NOT を使用して、マスクフィールドにおいて選別を行うことができます。

マスクフィールドで選別を行う場合、使用できるワイルドカード文字には次のものがあります。

- 演算子の LIKE と NOT LIKE には、パーセント (%) およびアンダースコア (_) を使用します。パーセント (%) の後には文字を入れないか、任意の数の文字を入れることができます。アンダースコア (_) は、その位置に任意の文字が入ることを表します。LIKE 演算子は、DEFINE または COMPUTE のいずれかのコマンドで一時項目を作成する式でサポートされます。
- 演算子の IS (または EQ) および IS-NOT (または NE) には、ドル記号 ($) およびドル記号とアスタリスクの組み合わせ ($*) を使用します。ドル記号 ($) は、その位置に任意の文字が入ることを表します。ドル記号とアスタリスクの組み合わせ ($*) は、複数のドル記号を記述して文字数を指定しなくてもマスクの終点を簡単に指定できる方法です。この組み合わせは、マスクの終点にのみ使用することができます。

  IF 句、および 1 つまたは複数の IF 句に分割可能な WHERE 句では、SET EQTEST=EXACT コマンドを発行して、「$」と「$*」をワイルドカードではなく通常の文字として扱うことができます。

注意：演算子の IS (または EQ) および IS-NOT (または NE) は、固定長フォーマットに対してマスクによる選別を行う場合にのみサポートされます。たとえば AnV のような可変長フォーマットの場合、マスクによる選別には LIKE または NOT LIKE 演算子を使用します。

マスクによるフィールドの選別 (LIKE と NOT LIKE の使用)

LINK 演算子でレコードを検索するには次の構文を使用します。

```
WHERE field LIKE 'mask'
```

マスクの値に基づいてレコードを除外するには、次のいずれかの構文を使用します。

```
WHERE field NOT LIKE 'mask'
```

または

```
WHERE NOT field LIKE 'mask'
```

説明

field

任意の有効フィールド名またはエイリアスです。

mask

入力する文字またはテキストです。マスクに使用できるワイルドカード文字には 2 種類あります。アンダースコア (_) は、その位置に任意の 1 文字が入ることを表し、パーセント記号 (%) は、その後に 0 (ゼロ) または任意の数の文字を指定することができます。

詳細は、レコード選択のタイプを参照してください。

IF 句の LIKE および UNLIKE による選別

LINK 演算子でレコードを検索するには次の構文を使用します。

```
IF field LIKE 'mask1' [OR 'mask2'...]
```

マスクの値によりレコードを拒否するには、次のいずれかの構文を使用します。

```
IF field UNLIKE 'mask1' [OR 'mask2' ...]
```

説明

field

任意の有効フィールド名またはエイリアスです。

mask1, mask2

比較に使用する文字パターンです。マスクにブランクが含まれる場合、一重引用符 (') を使用する必要があります。マスクに使用できるワイルドカード文字には 2 種類あります。アンダースコア (_) は、その位置に任意の 1 文字が入ることを表し、パーセント記号 (%) は、その後に 0 (ゼロ) または任意の数の文字を指定することができます。その他の文字は、すべてその文字自体と一致した場合にのみパターンとの一致と見なされます。

マスクによるフィールドの選別 (IS と IS-NOT の使用)

IS 演算子を使用してレコードを検索するには、次の構文を使用します。

```
{WHERE|IF} field {IS|EQ} 'mask'
```

マスクの値によりレコードを拒否するには、次のいずれかの構文を使用します。

```
{WHERE|IF} field {IS-NOT|NE} 'mask'
```

説明

field

任意の有効フィールド名またはエイリアスです。

IS|IS-NOT

レコード選択の演算子です。EQ は IS と同義です。NE は IS-NOT と同義です。

mask

入力する文字またはテキストです。マスクに使用できるワイルドカード文字には、ドル記号 ($) およびドル記号とアスタリスクの組み合わせ ($*) があります。ドル記号 ($) は、その位置に任意の文字が入ることを表します。ドル記号とアスタリスクの組み合わせ ($*) の後には文字を入れないか、任意の数の文字を入れることができます。ドル記号とアスタリスクの組み合わせ ($*) は、複数のドル記号を記述して文字数を指定しなくてもマスクの終点を簡単に指定できる方法です。この組み合わせは、マスクの終点にのみ使用することができます。

詳細は、マスクに使用する文字の制限を参照してください。

マスクに使用する文字の制限

- IS 演算子とともに使用するワイルドカード文字のドル記号 ($) およびドル記号とアスタリスクの組み合わせ ($*) は、LIKE 演算子とともに使用するリテラルのように処理され

ます。

- ワイルドカード文字の $ および $* を使用したマスクは、論理演算子の AND または OR を使用した複合 WHERE 句ではサポートされません。

先頭文字による選別

次のリクエストを発行し、初級レベルのコースを受講した従業員をすべて表示します。ここでは、すべての初級レベルのコースの先頭に「BASIC」という文字が付いています。

```
TABLE FILE EMPLOYEE
PRINT COURSE_NAME COURSE_CODE
BY LAST_NAME BY FIRST_NAME
WHERE COURSE_NAME LIKE 'BASIC%'
END
```

出力結果は次のとおりです。

LAST_NAME	FIRST_NAME	COURSE_NAME	COURSE_CODE
BLACKWOOD	ROSEMARIE	BASIC REPORT PREP NON-PROG	102
CROSS	BARBARA	BASIC REPORT PREP DP MGRS	107
JONES	DIANE	BASIC REPORT PREP FOR PROG	103
SMITH	MARY	BASIC REPORT PREP FOR PROG	103
SMITH	RICHARD	BASIC RPT NON-DP MGRS	108

フィールド任意の位置での文字選別

次の例では、FOCUS コースを受講した従業員名を知りたいが、「FOCUS」という文字がコースタイトルのどの位置に含まれているか分からないため、「FOCUS」という文字をワイルドカードで囲んでいます (CONTAINS 演算子の使用と同等)。

```
TABLE FILE EMPLOYEE
PRINT COURSE_NAME COURSE_CODE
BY LAST_NAME BY FIRST_NAME
WHERE COURSE_NAME LIKE '%FOCUS%'
END
```

出力結果は次のとおりです。

```
LAST_NAME        FIRST_NAME  COURSE_NAME                    COURSE_CODE
---------        ----------  -----------                    -----------
```

```
BLACKWOOD        ROSEMARIE   WHAT'S NEW IN FOCUS            202
JONES            DIANE       FOCUS INTERNALS                203
```

次のリクエストを発行し、コースコードが 20x シリーズのコースを受講した従業員をすべて表示します。ここでは、コースコードの最後の桁以外はすべて共通の 20 であることが分かっています。

```
TABLE FILE EMPLOYEE
PRINT COURSE_NAME COURSE_CODE
BY LAST_NAME BY FIRST_NAME
WHERE COURSE_CODE LIKE '20_'
END
```

出力結果は次のとおりです。

```
LAST_NAME        FIRST_NAME  COURSE_NAME                    COURSE_CODE
---------        ----------  -----------                    -----------
BLACKWOOD        ROSEMARIE   WHAT'S NEW IN FOCUS            202
JONES            DIANE       FOCUS INTERNALS                203
                             ADVANCED TECHNIQUES            201
```

先頭文字と特定の文字数による選別

次の例では、先頭文字およびフィールド値の長さを指定して選別を行います。この例では、姓が「BAN」という文字で始まり、その全文字数が 7 バイトになることが WHERE 句に記述されています。つまり、先頭の 3 文字が BAN で、続く 4 文字がドル記号になっています。このフィールドの残りの文字 (8 桁目から 15 桁目まで) はブランクになります。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
WHERE LAST_NAME IS 'BAN$$$$'
END
```

出力結果は次のとおりです。

```
LAST_NAME
---------
BANNING
```

先頭文字と任意の文字数による選別

任意の文字数を含むレコードを取得するには、ドル記号とアスタリスクの組み合わせ ($*) を使用します。

```
WHERE LAST_NAME IS 'BAN$*'
```

この句により、名前の文字数に関係なく、「BAN」という文字で始まるすべての姓が検索されます。$* を使用することにより、入力する文字を減らし、取得するフィールドの正確な長さが分からなくても選別マスクを定義することが可能です。

ワイルドカード文字の無効化

```
SET EQTEST = {WILDCARD|EXACT}
```

説明

WILDCARD

「$」および「$*」を、ワイルドカード文字として扱います。デフォルト値は「WILDCARD」です。

EXACT

IF テスト、および IF テストに変換される WHERE テストで、「$」と「$*」をワイルドカードではなく通常の文字として扱います。

SET EQTEST によるレコード選択

次のリクエストは、VIDEOTR2 データソースが使用され、2 つの類似した Email アドレスが検索されます。

- ドル記号 ($) を含む handy$man@usa.com
- 上記 Email アドレスのドル記号 ($) を「i」という文字で置き換えた handyiman@usa.com

```
DEFINE FILE VIDEOTR2
SMAIL/A18= IF EMAIL EQ 'handyman@usa.com'
           THEN 'handyiman@usa.com'
           ELSE EMAIL;
SMAIL/A18 = STRREP(18,SMAIL,1,'_',1,'$',18,SMAIL);
END
TABLE FILE VIDEOTR2
PRINT SMAIL
BY LASTNAME BY FIRSTNAME
WHERE SMAIL EQ 'handy$man@usa.com'
ON TABLE SET EQTEST WILDCARD
END
```

デフォルト設定の SET EQTEST=WILDCARD では、「$」はワイルドカード文字として扱われ、その位置のあらゆる文字が選別テストをパスするため、「WHERE SMAIL EQ 'handy$man@usa.com'」という WHERE テストは、アドレスに「$」を含むレコードと「i」を含むレコードの両方を返します。

```
LASTNAME         FIRSTNAME   SMAIL
--------         ---------   -----
HANDLER          EVAN        handy$man@usa.com
                             handyiman@usa.com
```

ON TABLE SET コマンドを ON TABLE SET EQTEST EXACT に変更すると、ドル記号 ($) が通常の文字として扱われ、完全一致の場合にのみテストをパスするため、「$」を含む Email アドレスが 1 つだけ返されます。

```
LASTNAME         FIRSTNAME   SMAIL
--------         ---------   -----
HANDLER          EVAN        handy$man@usa.com
```

LIKE でのエスケープ文字の使用

LIKE 構文にエスケープ文字を使用して、マスク文字 (% と _) を検索パターン内でワイルドカードではなくリテラルとして扱うことができます。これにより、データ内でこれらの文字を検索できるようになります。詳細は、レコード選択のタイプを参照してください。

WHERE 句でのエスケープ文字の使用

任意の単一文字をエスケープ文字として使用するには、「ESCAPE」という語句をその前に置きます。

```
WHERE fieldname LIKE 'mask' ESCAPE 'c'
```

説明

fieldname

選択テストで評価する任意の有効フィールド名またはエイリアスです。

mask

検索パターンです。一重引用符 (') で囲む必要があります。

c

エスケープ文字として識別する任意の単一文字です。マスク内のパーセント記号 (%) またはアンダースコア (_) の前にエスケープ文字を埋め込むと、これらはワイルドカードではなく、リテラルとして扱われます。一重引用符 (') で囲む必要があります。

IF 句マスクへのエスケープ文字の指定

LINK 構文または UNLIKE 構文に「ESCAPE」という語を前置することにより、任意の単一文字をエスケープ文字に割り当てることができます。

```
IF field {LIKE|UNLIKE} 'mask1' ESCAPE 'a' [OR 'mask2' ESCAPE 'b' ...
```

説明

field

選択テストで評価する任意の有効フィールド名またはエイリアスです。

mask1, mask2

検索パターンです。一重引用符 (') で囲む必要があります。

a, b ...

エスケープ文字として使用する単一文字です。各マスクに独自のエスケープ文字を指定することも、他と同一の文字を使用することもできます。マスク内のパーセント記号 (%) またはアンダースコア (_) の前にエスケープ文字を埋め込むと、これらはワイルドカードではなく、リテラルとして扱われます。マスクにブランクが含まれる場合、一重引用符 (') を使用する必要があります。

エスケープ文字使用上の注意

- パーセント記号 (%) またはアンダースコア (_) 以外は、エスケープ文字の前に指定しても、その文字およびエスケープ自体が無視されます。
- エスケープ文字自体をエスケープさせることができます。たとえば、abc¥¥%¥¥ のように使用すると文字列内で普通の文字になります。
- WHERE 句内の 1 つの LIKE 句に 1 つのみのエスケープ文字を使用することができます。
- エスケープ文字が有効になるのは、ESCAPE 構文が LIKE 句に含まれている場合に限られます。
- それぞれの LIKE 句に個別のエスケープ文字を設定することができます。
- WHERE 条件をリテラル OR 句とともに使用する場合、ESCAPE は先頭の OR 句に配置する必要があります。その場合、ESCAPE は WHERE 式の残りのすべての句に適用されます。以下はその例です。

```
WHERE field LIKE 'ABCg_' ESCAPE 'g' OR 'ABCg%' OR 'g%ABC'
```

WHERE 句内のエスケープ文字の使用

VIDEOTR2 データソースには Email アドレス用のフィールドが定義されています。次のリクエストを発行して、「handy_」という文字を含む Email アドレスを検索します。

```
TABLE FILE VIDEOTR2
PRINT CUSTID LASTNAME FIRSTNAME EMAIL
WHERE EMAIL LIKE 'handy_%'
END
```

アンダースコア (_) はワイルドカード文字として機能するため、このリクエストで返された 2 つのインスタンスにはアンダースコアが付いていないものが 1 つ含まれています。

出力結果は次のとおりです。

```
CUSTID  LASTNAME         FIRSTNAME   EMAIL
------  --------         ---------   -----
0944    HANDLER          EVAN        handy_man@usa.com
0944    HANDLER          EVAN        handyman@usa.com
```

アンダースコア (_) を含んだインスタンスのみを取得するには、アンダースコアをワイルドカードではなく、普通の文字として扱うようにする必要があります。次のリクエストでは、Email フィールドでアンダースコア (_) を含むインスタンスのみが取得されます。

```
TABLE FILE VIDEOTR2
PRINT CUSTID LASTNAME FIRSTNAME EMAIL
WHERE EMAIL LIKE 'handy¥_%' ESCAPE '¥'
END
```

出力結果は次のとおりです。

```
CUSTID  LASTNAME         FIRSTNAME   EMAIL
------  --------         ---------   -----
0944    HANDLER          EVAN        handy_man@usa.com
```

IF 句内のエスケープ文字の使用

VIDEOTR2 データソースには Email アドレス用のフィールドが定義されています。次のリクエストを発行して、「handy_」という文字を含む Email アドレスを検索します。

```
TABLE FILE VIDEOTR2
PRINT CUSTID LASTNAME FIRSTNAME EMAI
IF EMAIL LIKE 'handy_%'
END
```

アンダースコア (_) はワイルドカード文字として機能するため、このリクエストで返された 2 つのインスタンスにはアンダースコアが付いていないものが 1 つ含まれています。

出力結果は次のとおりです。

```
CUSTID  LASTNAME         FIRSTNAME   EMAIL
------  --------         ---------   -----
0944    HANDLER          EVAN        handy_man@usa.com
0944    HANDLER          EVAN        handyman@usa.com
```

アンダースコア (_) を含んだインスタンスのみを取得するには、アンダースコアをワイルドカードではなく、普通の文字として扱うようにする必要があります。次のリクエストでは、Email フィールドでアンダースコア (_) を含むインスタンスのみが取得されます。

```
TABLE FILE VIDEOTR2
PRINT CUSTID LASTNAME FIRSTNAME EMAI
IF EMAIL LIKE 'handy¥_%' ESCAPE '¥'
END
```

出力結果は次のとおりです。

```
CUSTID  LASTNAME         FIRSTNAME   EMAIL
------  --------         ---------   -----
0944    HANDLER          EVAN        handy_man@usa.com
```

INCLUDES と EXCLUDES による親セグメントの選択

演算子の INCLUDES および EXCLUDES を使用して、列挙したすべてのリテラルが子セグメントのフィールドインスタンスに含まれているかどうかをテストすることができます。INCLUDES および EXCLUDES では、親レコードのみが取得されます。INCLUDES または EXCLUDES のテストでフィールドを指定した場合、そのフィールドと同一のセグメントのフィールドを表示 (PRINT または LIST) させることはできません。

注意：INCLUDES および EXCLUDES は、マルチセグメントの FOCUS データソースでのみ機能します。

INCLUDES および EXCLUDES の使用上の注意

- ブランクを含むリテラルは、一重引用符 (') で囲む必要があります。
- リテラルの総数は、31 個以下にする必要があります。
- 1 つのリクエストに複数の INCLUDES 句または EXCLUDES 句を使用する場合、それぞれの句を別の行に記述します。
- テストする複数のリテラルを AND および OR で結合することはできますが、OR は AND に変更されます。

INCLUDES および EXCLUDES によるレコードの選択

次のリクエストの選択条件には、INCLUDES 句が使用されています。

```
WHERE JOBCODE INCLUDES A01 OR B01
```

このリクエストでは、AND を使用した場合のように、A01 と B01 の両方の JOBCODE インスタンスの従業員レコードが取得されます。

次の例では、JOBCODE フィールドに A01 と B01 の両方の値が存在する場合にレコードが選択されます。

```
WHERE JOBCODE INCLUDES A01 AND B01
```

いずれかの値が存在しない場合、レポート出力用にこのレコードは選択されません。

次のリクエストの選択条件には、EXCLUDES 句が使用されています。

```
WHERE JOBCODE EXCLUDES A01 AND B01
```

ここでは、両方の値が存在しないすべてのレコードがレポート出力用に選択されます。

次の例では、CAR データソースを使用して、Jaguar と Jensen を生産している英国のみを表示しています。

```
TABLE FILE CAR
PRINT COUNTRY
WHERE CAR INCLUDES JAGUAR AND JENSEN
END
```

出力結果は次のとおりです。

```
COUNTRY
-------
ENGLAND
```

グループキー値による選択

データソースの中には、グループキーを使用するものがあります。グループキーは、複数のフィールドで構成された単一キーです。グループ名を使用して、そのグループキーのフィールドを参照することができます。

グループキーの値に基づいてレコードを選択するには、各フィールドの値を記述する必要があります。この場合、それぞれの値をスラッシュ文字 (/) で区別する必要があります。

なお、グループフィールドを参照する WHERE 句は、AND または OR とともに使用することはできません。詳細は、複合式によるレコード選択を参照してください。

グループキーによるレコードの選択

データソースに「PRODNO」という名前の 3 つの異なるフィールドで構成されるグループキーが存在することを想定します。最初のフィールドは文字、2 つ目はパック 10 進数、3 つ目は整数のフォーマットでそれぞれ保存されています。このグループでの選択条件の句は次のようになります。

```
WHERE PRODNO EQ 'RS/62/83'
```

読み込みレコード数の制限

レポートによっては、限られたレコード数で十分に目的を達する場合もあります。取得するレコード数を指定しておくと、レコード取得の処理を途中で終了させることができます。こ

の方法は、次の場合に役立ちます。

- レポートの新しいデザインを考案中に、実際のデータソースから何個かのレコードのみを取得してデザインのテストを行う。
- データベース管理者が、大容量のデータソースから取得するレコード数に上限を設定し、レポートのサイズを制限する必要がある。この上限は、ユーザのパスワードに付加されます。
- テスト条件を満たすレコード数がすでに分かっている。そのレコード数を指定しておくと、条件を満たす最終レコードに達すると検索処理を継続する必要がなくなります。たとえば、資金の電子振替を使用する従業員が 10 名しかいないことが分かっている場合、10 名分のレコードのみを取得します。レコードの上限を 10 にすれば、10 番目のレコードが取得された時点でその取得処理は終了します。それ以降のデータソースの検索は実行されません。

読み込みレコード数の制限

取得するレコード数を制限するには 2 つの方法があります。次の構文を使用することができます。

```
WHERE RECORDLIMIT EQ n
```

説明

n

0 (ゼロ) より大きい数で、取得するレコード数を表します。この構文は、FOCUS データソースと FOCUS 以外のデータソースの両方で使用することができます。

FOCUS 以外のすべてのデータソースには次の構文を使用することもできます。

```
WHERE READLIMIT EQ n
```

説明

n

0 (ゼロ) より大きい数で、実行する読み込み処理数 (レコード数ではない) を表します。詳細は、該当するデータアダプタのマニュアルを参照してください。

ヒント：FOCUS データソースに READLIMIT テストを適用しようとすると、そのリクエストは正しく処理されますが、READLIMIT 句は無視されます。

注意：SET RECORDLIMIT を使用すると、AUTOINDEX が無効になります。

読み込みレコード数の制限

次のリクエストは、4 つのレコードを取得して、4 行で構成されるレポートを作成します。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME AND FIRST_NAME AND EMP_ID
WHERE RECORDLIMIT EQ 4
END
```

出力結果は次のとおりです。

```
LAST_NAME        FIRST_NAME  EMP_ID
---------        ----------  ------
STEVENS          ALFRED      071382660
SMITH            MARY        112847612
JONES            DIANE       117593129
SMITH            RICHARD     119265415
```

IF 句によるレコードの選択

IF 句を使用してレポートに表示するレコードを選択することができます。IF 句では、WHERE 句の機能の一部が提供されます。使用可能な IF 演算子のリストは、演算子によるレコード選択を参照してください。

ヒント：IF 構文を明示的に使用する必要のある場合 (例、レガシーアプリケーションをサポートする) 以外は、WHERE 句を使用することをお勧めします。

IF 句によるレコードの選択

```
IF fieldname operator literal [OR literal]
```

説明

fieldname

テストを行うフィールド名です (テスト値)。

operator

使用する選択演算子のタイプです。有効な演算子についての詳細は、演算子によるレコード選択を参照してください。

literal

MISSING キーワード (レコード選択のタイプを参照)、またはデータソースに存在する文字または数値で、これらの値の間に OR を使用します。

ブランクを含むすべてのリテラル (例、New York City) およびすべての日付と日付時間のリテラルは、一重引用符 (') で囲む必要があります。

注意：IF 句を単独で使用して、論理演算子の AND および OR で単純な式を結合した複合式を作成することはできません。複合論理式には、DEFINE コマンドとともに IF 句を使用する必要があります。詳細は、式の使用を参照してください。WHERE を使用すると、より簡単にこの操作を行うことができます。詳細は、複合式によるレコード選択を参照してください。

複数 IF 句の使用

次の例のように、必要な数の IF 句を使用して、すべての選択条件を定義することができます。

```
TABLE FILE EMPLOYEE
PRINT EMP_ID LAST_NAME
IF SALARY GT 20000
IF DEPARTMENT IS MIS
IF LAST_NAME IS CROSS OR BANNING
END
```

これらの条件をすべて満たすレコードのみがレポートに表示されます。出力結果は次のとおりです。

```
EMP_ID     LAST_NAME
------     ---------
818692173  CROSS
```

ファイルからの選択値読み込み

WHERE 句または IF 句にリテラルテスト値を直接記述する代わりに、その値をファイルに保存しておき、リクエストでそのファイルを参照することができます。その後、ファイルに保存した値で等式または不等式のテストを実行し、その結果を基にレコードを選択することができます。

この方法には次のような利点があります。

- 多数の選択値を最初にコーディングしておき、これらの値一式をリクエストに何回も使用することで、時間を節約することができます。また、選択条件を 1 箇所で管理することにより、値の整合性を保つことができます。
- 選択値がデータソースにすでに存在する場合、選択値のファイルをすばやく作成するには、レポートを作成してからその出力結果を HOLD ファイルまたは SAVE ファイルに保存しておきます。その後、そのファイルから選択値を読み込むことができます。

  HOLD ファイル (データ説明を含む) の値は、BINARY フォーマット (デフォルト) または ALPHA (単純文字) フォーマットで保存することができます。SAVE ファイルを使用する場合は、ALPHA フォーマット (デフォルト) で保存する必要があります。SAVB ファイルは、文字値の場合にのみ使用することができます。HOLD および SAVE ファイルについての詳細は、レポート出力の保存と再利用を参照してください。
- 大文字と小文字が混在する文字および特殊文字で構成されるエントリを含めることができます。

ファイルからの選択値の読み込み - WHERE フィールド IN ファイル

```
WHERE [NOT] fieldname IN FILE file
```

説明

fieldname

選択フィールドの名前です。データソースの任意の実フィールドまたは一時項目です。

file

ファイルの名前です。

2 部構成名 (app/file) はサポートされません。

詳細は、ファイルからの選択値読み込みを参照してください。

ファイルからの選択値の読み込み - WHERE フィールド演算子 (ファイル)

```
WHERE field1 operator1 (file1) [{OR|AND} field2 operator2 (file2) ... ]
```

説明

field1, field2

有効な任意のフィールド名またはエイリアスです。

operator1, operator2

EQ、IS、NE、IS-NOT 演算子のいずれかです。

file1, file1

ファイルの名前です。

2 部構成名 (app/file) はサポートされません。

ファイルからの選択値の読み込み - IF

```
IF fieldname operator (file) [OR (file) ... ]
```

説明

fieldname

任意の有効フィールド名またはエイリアスです。

operator

演算子の EQ、IS、NE、IS-NOT のいずれかです。詳細は、演算子によるレコード選択を参照してください。

file

ファイルの名前です。

2 部構成名 (app/file) はサポートされません。

ファイルから選択値を読み込む際の注意

ファイルから選択条件を読み込む場合、そのファイルは次の規則に適合する必要があります。

- ファイル内ではそれぞれの値を別の行に記述する必要があります。

  IF では、複数の情報を 1 行に記述することができますが、使用されるのはその行の先頭のデータ値のみです。

- 選択値は、1 列目から開始する必要があります。
- ファイル名が HOLD 以外の場合は、値は文字フォーマットと見なされ、数値は必要に応じて内部計算用の数値に変換されます (例、バイナリ整数)。
- 値の最大数は 32,767 です。
- WHERE では、ブランクまたは数理演算子 (-、+、*、/) が含まれる文字は、一重引用符 (') で囲む必要があります。
- WHERE では、複合 WHERE 句に IN FILE を複数回使用する場合、指定した複数のファイルは同一のレコードフォーマットになっていなければなりません。

  リテラルリストが大きすぎる場合は、エラーが表示されます。

- 複数のファイル名を使用する場合、IF では「OR」という語句でファイル名を区切ります。WHERE では「AND」を使用します。ファイル名の前にアプリケーション名を指定することはできません。リテラルは、ファイル名と混在させることもできます。以下はその例です。

```
IF fieldname operator (filename) OR literal...etc...
```

ファイルからの選択値の読み込み (WHERE フィールド IN ファイル)

B141 と B142 の値を含む EXPER.txt という名のファイルを、baseapp フォルダに作成します。

次のリクエストでは、EXPER ファイルに保存されている選択条件を使用します。リクエストの実行前に EXPER ファイルを割り当てるか、FILEDEF コマンドを発行する必要があります。たとえば、このファイル名が exper.ftm で、baseapp アプリケーションディレクトリに保存されている場合、次の FILEDEF コマンドを発行することができます。

```
FILEDEF EXPER DISK baseapp/exper.ftm
```

PRODUCT_ID の値が B141 または B142 となるレコードをすべて選択します。

```
TABLE FILE GGPRODS
SUM UNIT_PRICE
BY PRODUCT_DESCRIPTION
WHERE PRODUCT_ID IN FILE EXPER
END
```

選択条件をリクエストに直接含める場合は、WHERE 句で値を明示的に指定します。

```
WHERE PRODUCT_DESCRIPTION EQ 'B141' or 'B142'
```

出力結果は次のとおりです。

```
                     Unit
Product              Price
-------              -----
French Roast         81.00
Hazelnut             58.00
```

ファイルからの選択値の読み込み - WHERE フィールド演算子 (ファイル)

次のリクエストは、GGPRODS データソースに対して実行され、「EXPER1」という HOLD ファイルを作成し、このファイルに製品 ID の B141、B142、B143、B144 を格納します。

```
TABLE FILE GGPRODS
BY PRODUCT_ID BY PRODUCT_DESCRIPTION
WHERE PRODUCT_ID EQ 'B141' OR 'B142' OR 'B143' OR 'B144'
ON TABLE HOLD AS EXPER1 FORMAT ALPHA
END
```

次のリクエストは、GGPRODS データソースに対して実行され、「EXPER2」という HOLD ファイルを作成し、このファイルに製品 ID の B144、F101、F102 を格納します。

```
TABLE FILE GGPRODS
BY PRODUCT_ID BY PRODUCT_DESCRIPTION
WHERE PRODUCT_ID EQ 'B144' OR 'F101' OR 'F102'
ON TABLE HOLD AS EXPER2 FORMAT ALPHA
END
```

次のリクエストは、EXPER1 と EXPER2 の両ファイルに存在する値を選択します。

```
TABLE FILE GGPRODS
SUM PRODUCT_DESCRIPTION
BY PRODUCT_ID
WHERE PRODUCT_ID EQ (EXPER1) AND PRODUCT_ID IS (EXPER2)
ON TABLE SET PAGE NOPAGE
END
```

出力結果は次のとおりです。

```
PRODUCT
Code     Product
-------  -------
B144     Kona
```

ファイルからの選択値の読み込み (IF)

B141 と B142 の値を含む EXPER.txt という名のファイルを、baseapp フォルダに作成します。

次のリクエストでは、EXPER ファイルに保存されている選択条件を使用します。PRODUCT_ID の値が B141 または B142 となるレコードをすべて選択します。

```
TABLE FILE GGPRODS
SUM UNIT_PRICE
BY PRODUCT_DESCRIPTION
IF PRODUCT_ID IS (EXPER)
END
```

選択条件をリクエストに直接含める場合は、IF 句で値を明示的に指定します。

```
IF PRODUCT_DESCRIPTION EQ 'B141' or 'B142'
```

出力結果は次のとおりです。

```
                     Unit
Product              Price
-------              -----
French Roast         81.00
Hazelnut             58.00
```

ファイルへの選別条件の割り当て

リクエストとは別にデータソースに選択条件を割り当てておくと、そのデータソースに対するリクエストで自動的に選択条件を有効にすることができます。

フィルタとは、WHERE および IF 条件のいずれかまたはその両方で構成される定義の集合体のことで、これはファイルレベルで管理されます。データソースに対してリクエストを発行する際は、そのデータソースに対して有効なすべてのフィルタが常に使用されます。リクエストで有効な WHERE または IF 構文は、フィルタでも有効になります。

リクエストを実行する前であれば、いつでもフィルタを宣言することができます。フィルタを実行しているセッション中であっても、そのフィルタを次のリクエストに使用することができます。詳細は、フィルタの宣言を参照してください。

フィルタにより、次のことが可能です。

- 共通の選別条件一式を宣言し、データソースからデータを取得する際に常にその選択条件一式を適用します。1 つのデータソースに対して、1 つまたは複数のフィルタを宣言することができます。
- 選別条件一式を宣言し、その条件のオンとオフを動的に切り替えることができます。
- マスターファイルで規則を指定せずにデータへのアクセスを制限します。

対話型環境では、フィルタを使用することで繰り返し入力する回数が減ります。

注意：データソースに対してフィルタを宣言するだけでは、そのフィルタは有効になりません。フィルタは SET コマンドを使用して有効にする必要があります。詳細は、フィルタの有効化と無効化を参照してください。

フィルタの宣言

フィルタは次のように宣言します。

```
FILTER FILE filename [CLEAR|ADD]
   [filter-defines;]
   NAME=filtername1 [,DESC=text]
   where-if phrases    .
   .
   .
   NAME=filternamen [,DESC=text]
   where-if phrases END
```

説明

filename

フィルタを適用するマスターファイル名です。

CLEAR

既存のフィルタ句をすべて削除します。削除の対象には、定義済みの一時項目 (DEFINE) も含まれます。

ADD

定義済みのフィルタをクリアせずに、既存のフィルタ宣言に新しいフィルタ句を追加します。

filter-defines

フィルタで使用するために宣言された一時項目 (DEFINE) です。詳細は、フィルタに一時項目 (DEFINE) を使用する際の注意を参照してください。

filtername1...filternamen

後に続く SET FILTER コマンドで参照するフィルタ名です。この名前の最大文字数は 66 バイトで、他のファイルと重複しない名前でなければなりません。

text

フィルタの説明を記述します。テキストは 1 行に収める必要があります。

where-if phrases

有効なすべての構文を含めることができる選別条件です。マスターファイル内のデータソースフィールドおよび一時項目 (DEFINE) を参照することはできますが、DEFINE コマンドで宣言された一時項目や他のフィルタ名を参照することはできません。

フィルタに一時項目 (DEFINE) を使用する際の注意

フィルタに使用する一時項目 (DEFINE) の条件は次のとおりです。

- 特定のフィルタ宣言内のフィルタに対してのみ使用することができます。
- DEFINE または TABLE コマンドで参照することはできません。
- DEFINE コマンドで一時項目に対して有効な構文はすべてサポートされます。
- DEFINE コマンドの一時項目は参照することができませんが、マスターファイルの一時項目 (DEFINE) は参照することができます。
- DEFINE コマンドの一時項目とは異なり、表示フィールドの制限には考慮されません。
- 指定される最初のフィルタより先にすべて宣言しておく必要があります。
- 各一時項目 (DEFINE) の後にはセミコロン (;) を入れて終了する必要があります。
- DEFINE FILE と END コマンドの間に入れることはできません。
- 同一ファイルに同一の一時項目 (DEFINE) 名を繰り返し使用することはできません。

フィルタの宣言

最初の例では、1 つの WHERE 条件で構成される UK という名前のフィルタを作成します。また、CAR データソースのフィルタですでに使用されている一時項目 (DEFINE) 群に、「MARK_UP」という一時項目の定義を追加します。

フィルタの UK を有効にした状態でリクエストを CAR に対して発行すると、「MARK_UP が 1000 より大きい場合」という WHERE 条件が自動的にリクエストに追加されます。

注意：一時項目 (DEFINE) の MARK_UP は、TABLE リクエストで明示的に表示したり、参照したりすることはできません。

```
FILTER FILE CAR ADD
MARK_UP/D7=RCOST-DCOST;
NAME=UK
WHERE MARK_UP GT 1000
END
```

2 つ目の例では、CAR データソースに対して、「ASIA」、「UK」、「LUXURY」という名前の 3 つのフィルタを宣言します。フィルタの ASIA には、説明用のテキストが記述されています。1 行目の CLEAR により、CAR データソースの既存フィルタおよびそこで使用されていた既存の定義済み一時項目 (DEFINE) がすべてクリアされ、その後で新しい定義が処理されます。

```
FILTER FILE CAR CLEAR
NAME=ASIA,DESC=Asian cars only
IF COUNTRY EQ JAPAN
NAME=UK
IF COUNTRY EQ ENGLAND
NAME=LUXURY
IF RETAIL_COST GT 50000
END
```

フィルタの有効化と無効化

次のコマンドを使用して、フィルタの有効と無効を切り替えることができます。

```
SET FILTER= {*|xx[yy zz]} IN {file|*} {ON|OFF}
```

説明

*****

宣言されたすべてのフィルタを示します。これがデフォルト値です。

xx, yy, zz

FILTER FILE コマンドの NAME = 構文で宣言するフィルタの名前です。

file

選別条件を割り当てるデータソース名です。アスタリスク (*) は、すべてのデータソースを表します。

ON

特定のデータソースまたはすべてのデータソース (*) を対象に、すべてのフィルタ (*) または指定したフィルタを有効にします。1 つのデータソースで有効にできるフィルタの最大数は、そのフィルタが含まれている WHERE/IF 句の数により限定されます。それは、1 つのリクエストの WHERE/IF 条件の制限数を超えてはなりません。

OFF

特定のデータソースまたはすべてのデータソース (*) を対象に、すべてのフィルタ (*) または指定したフィルタを無効にします。デフォルト値は OFF です。

注意：SET FILTER コマンドは、1 行内に収める必要があります。有効にする複数のフィルタが 1 行に収まらない場合は、新しい SET FILTER コマンドを追加発行します。ON を指定した場合は、有効なフィルタが累積されます。

フィルタの有効化と無効化

次のコマンドは、フィルタの A、B、C、D、E、F を有効にし、G を無効にする例です (すでに ON に設定されていると仮定)。

```
SET FILTER = A B C IN CAR ON
SET FILTER = D E F IN CAR ON
SET FILTER = G IN CAR OFF
```

次のコマンドは、いくつかのフィルタを有効にした上で、その他のフィルタを無効にします。

```
SET FILTER = UK LUXURY IN CAR ON
...
TABLE FILE CAR
PRINT COUNTRY MODEL RETAIL_COST
END
...
SET FILTER = LUXURY IN CAR OFF
TABLE FILE CAR
PRINT COUNTRY MODEL RETAIL_COST
END
```

この例では、最初の SET FILTER コマンドにより、CAR データソースに割り当てられた フィルタの UK と LUXURY が有効になり、それ以降の CAR データソースに対するすべてのリクエストにその選別条件が適用されます。

2 番目の SET FILTER コマンドにより、CAR データソースの LUXURY フィルタが無効になります。LUXURY を再度有効にしない限り、これ以降の CAR に対するリクエストは LUXURY の条件は適用されませんが、UK の条件は引き続き適用されます。

フィルタステータスのクエリ

既存フィルタのステータスを特定には、次の構文を使用します。

```
? FILTER [{file|*}] [SET] [ALL]]
```

説明

file

マスターファイル名です。

*****

フィルタが宣言されているすべてのマスターファイルのフィルタを表示します。

SET

有効なフィルタのみを表示します。

ALL

フィルタのすべての情報を表示します。たとえば、フィルタの説明や WHERE または IF の正確な定義などがあります。

フィルタのクエリ

フィルタのクエリを実行するには、次のコマンドを発行します。

```
FILTER FILE CAR CLEAR
NAME=BOTH, DESC=Asian and British cars only
IF COUNTRY EQ JAPAN AND ENGLAND
END
```

```
SET FILTER =BOTH IN CAR ON
TABLE FILE CAR
PRINT CAR RETAIL_COST
BY COUNTRY
END
```

出力結果は次のとおりです。

```
COUNTRY     CAR               RETAIL_COST
-------     ---               -----------
ENGLAND     JAGUAR                  8,878
            JAGUAR                 13,491
            JENSEN                 17,850
            TRIUMPH                 5,100
JAPAN       DATSUN                  3,139
            TOYOTA                  3,339
```

次の例は、データソースすべてのフィルタクエリを実行します。

```
? FILTER
```

フィルタが定義されていない場合は、次のメッセージが表示されます。

```
有効なフィルタはありません
```

フィルタが定義されている場合は、次のメッセージが表示されます。

```
Set File     Filter name Description
--- -------- ----------- ----------------------------------
    CAR      ROB         Rob's selections
*   CAR      PETER       Peter's selections for CAR
*   EMPLOYEE DAVE        Dave's tests
    EMPLOYEE BRAD        Brad's tests
```

CAR データソースのフィルタクエリを実行するには、次のコマンドを発行します。

```
? FILTER CAR
```

CAR データソースにフィルタが定義されていない場合は、次のメッセージが表示されます。

```
ファイル CAR にはフィルタは適用されていません。
```

CAR データソースにフィルタが定義されている場合は、次のメッセージが表示されます。

```
Set File     Filter name Description
--- -------- ----------- ----------------------------------
    CAR      ROB         Rob's selections
*   CAR      PETER       Peter's selections for CAR
```

有効なフィルタをすべて表示するには、次のコマンドを発行します。

```
? FILTER * SET
```

出力結果は次のとおりです。

```
Set File     Filter name Description
--- -------- ----------- ----------------------------------
*   CAR      PETER       Peter's selections for CAR
*   EMPLOYEE DAVE        Dave's tests
```

最初の列に表示されたアスタリスク (*) は、そのフィルタが有効であることを表しています。

JOIN 間のフィルタ保持

デフォルト設定では、ホストデータソースで定義したフィルタは、JOIN コマンドの実行後にクリアされます。ただし、SET KEEPFILTERS=ON コマンドを発行することにより、JOIN コマンドを発行した後でもフィルタを保持することができます。

KEEPFILTERS を ON に設定すると、JOIN コマンドの実行後に、フィルタ定義およびその定義を宣言したときのステータスが元の状態に戻ります。JOIN を発行する前に定義した一連のフィルタおよび一時項目 (DEFINE) は、「コンテキスト」と呼ばれます。なお、一時項目に関連するコンテキストについての詳細は、SET KEEPDEFINES および DEFINE FILE SAVE についての説明を参照してください。JOIN または DEFINE FILE コマンドを発行するたびに、新しいコンテキストが作成されます。

JOIN コマンドを発行した後で新しいフィルタを定義する場合、FILTER FILE コマンドの CLEAR オプションを発行しない限り、以前に定義したフィルタと同一名のフィルタを作成することはできません。CLEAR オプションを使用すると、すべてのコンテキストから指定したマスターファイルのフィルタ定義がすべてクリアされます。

JOIN をクリアすると、その JOIN コマンドを発行する前に有効であったフィルタ定義およびクリアされなかったフィルタ定義がすべて元のステータスに戻ります。JOIN CLEAR join_

name コマンドを発行して JOIN をクリアすると、JOIN join_name コマンドを発行した後で作成したコンテキストおよびフィルタ定義がすべて削除されます。

注意：元の FILTER FILE に存在しないフィールドへの参照があるためにエラーが発生した場合、KEEPFILTER が ON に設定されている場合でも、そのフィルタは無効になります。

KEEPFILTERS によるフィルタ定義の保持

```
SET KEEPFILTERS = {OFF|ON}
```

説明

OFF

JOIN の前に発行されたフィルタは保持されません。デフォルト値は OFF です。

ON

JOIN 間でのフィルタは保持されます。

KEEPFILTERS によるフィルタの保持

次の例では、最初のフィルタである UNITPR は、すべての JOIN が発行される前に定義されますが、KEEPFILTERS を ON に設定した後に定義しています。

```
SET KEEPFILTERS = ON
FILTER FILE VIDEOTRK
PERUNIT/F5 = TRANSTOT/QUANTITY;
NAME=UNITPR
WHERE PERUNIT GT 2
WHERE LASTNAME LE 'CRUZ'
END
```

? UNITPR の名のフィルタを作成したが有効になっていないことを示しています (有効になったことは SET フィールドのアスタリスク (*) によって示されます)。

```
? FILTER

SET FILE     FILTER NAME DESCRIPTION


```

```
--- -------- ----------- ---------------------------------
    VIDEOTRK UNITPR
```

次にこのフィルタを有効にします。

```
SET FILTER= UNITPR IN VIDEOTRK ON
```

? フィルタが有効化されたことを示しています。

```
? FILTER

SET FILE     FILTER NAME DESCRIPTION
--- -------- ----------- ---------------------------------
*   VIDEOTRK UNITPR
```

次の例は、フィルタを使用したデータソースに対して発行した TABLE リクエストです。

```
TABLE FILE VIDEOTRK
SUM QUANTITY TRANSTOT BY LASTNAME
END
```

出力結果を見ると、TABLE リクエストにより、UNITPR フィルタの条件を満たすデータのみが取得されたことが分かります。

```
テーブルのレコード数=        6  行数=       3
アクセスはフィルタにより制限されています

PAUSE..PLEASE ISSUE CARRIAGE RETURN WHEN READY

LASTNAME         QUANTITY  TRANSTOT
--------         --------  --------
CHANG                   3     31.00
COLE                    2     18.98
CRUZ                    2     16.00
```

次に、VIDEOTRK データソースを MOVIES データソースに結合します。?UNITPR フィルタが JOIN によってクリアされなかったことを示しています。

```
JOIN MOVIECODE IN VIDEOTRK TO ALL MOVIECODE IN MOVIES AS J1
```

? UNITPR フィルタがまだ存在し、有効であることを示しています。

```
? FILTER

SET FILE     FILTER NAME DESCRIPTION
--- -------- ----------- ---------------------------------
*   VIDEOTRK UNITPR
```

次に、「YEARS1」という名前の新しいフィルタを作成し、VIDEOTRK と MOVIES 間の結合に対してこのフィルタを有効にします。

```
FILTER FILE VIDEOTRK
YEARS/I5 = (EXPDATE - TRANSDATE)/365;
NAME=YEARS1
WHERE YEARS GT 1
END
SET FILTER= YEARS1 IN VIDEOTRK ON
```

?UNITPR および YEARS1 フィルタの両方が存在し、有効であることを示しています。

```
? FILTER

SET FILE     FILTER NAME DESCRIPTION
--- -------- ----------- ---------------------------------
*   VIDEOTRK UNITPR
*   VIDEOTRK YEARS1
```

次に、J1 をクリアします。? FILTER クエリの出力結果は JOIN コマンドが発行された後に作成された YEARS1 フィルタがもう存在しないことを示しています。JOIN コマンドを発行する前に作成した UNITPR フィルタは、元の状態に戻っています。

```
JOIN CLEAR J1
? FILTER

SET FILE     FILTER NAME DESCRIPTION
--- -------- ----------- ---------------------------------
*   VIDEOTRK UNITPR
```

一時項目の作成

レポートを作成する場合、使用可能なフィールドはデータソースに存在するフィールドに限定されません。既存のデータから必要な情報を出力できる場合は、作成した一時項目を評価し、レポートに表示することができます。一時項目には保存場所は不要です。保存場所は必要に応じて作成されます。

一時項目の概要

一時項目とは、その値自体はデータソースに保存されていないが、既存のデータから計算を行ったり、絶対値を割り当てたりできるフィールドです。一時項目には保存場所は不要です。保存場所は必要に応じて作成されます。

一時項目を作成する場合、式を記述してその値を生成します。1 つの値を生成するために、式にはフィールド、定数、演算子を組み合わせて使用することができます。たとえば、給与額および控除額で構成されるデータがある場合、次の式を使用して給与額に対する控除の割合を計算することができます。

```
deduction / salary
```

ユーザが独自に式を指定することもできますが、特定の計算や操作を行うために用意されたさまざまな関数の中から必要なものを選択することもできます。さらに、単純な式および関数を構成要素としてさらに複雑な式を作成したり、特定の一時項目を使用してその他の一時項目の評価を行ったりすることも可能です。

一時項目のタイプ

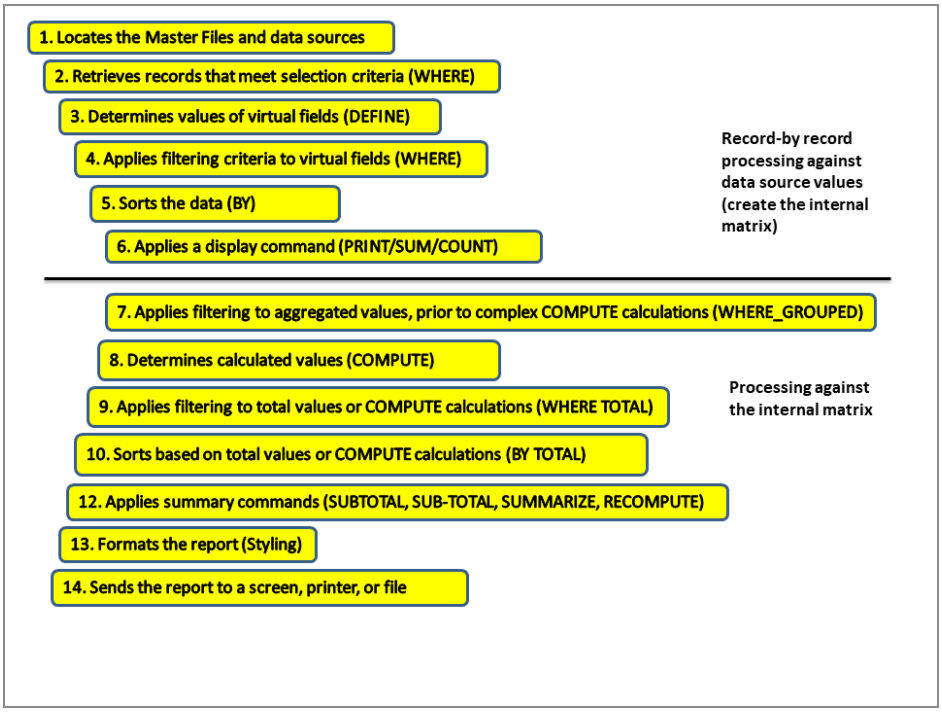
使用可能な一時項目のタイプは、その評価手法の違いから、「一時項目 (DEFINE)」と「一時項目 (COMPUTE)」という 2 種類に分類されます。

一時項目 (DEFINE) は、選択条件に一致するレコードをデータソースから取得する際に評価されます。式の結果は、データソースに実際に保存されている実フィールドのように扱われます。

一時項目 (COMPUTE) は、選択条件に一致するすべてのデータが検索、ソート、集計された後で評価されます。その結果、この計算はフィールドの集計値を使用して実行されます。

一時項目の評価

下図は、リクエストがどのように処理され、一時項目がそれぞれいつ評価されるかを示しています。



一時項目 (DEFINE) と一時項目 (COMPUTE) の区別

次の例は、DRATIO フィールド (一時項目 (DEFINE)) と CRATIO (一時項目 (COMPUTE)) の両方に 「DELIVER_AMT/OPENING_AMT」という同一の式を使用したにも関わらず、それぞれの結果が異なる様子を示しています。一時項目 (COMPUTE) の CRATIO は、すべてのレコードが選択、ソート、集計された後で計算が実行されます。一時項目 (DEFINE) の DRATIO は、取得した各レコードに対して計算が実行されます。

```
DEFINE FILE SALES
DRATIO = DELIVER_AMT/OPENING_AMT;
END
TABLE FILE SALES
SUM DELIVER_AMT AND OPENING_AMT AND DRATIO
COMPUTE CRATIO = DELIVER_AMT/OPENING_AMT;
END
```

出力結果は次のとおりです。

DELIVER_AMT	OPENING_AMT	DRATIO	CRATIO
760	724	28.41	1.05

一時項目の選択

次の説明を参考にして、一時項目のタイプを選択します。

一時項目 (DEFINE) 次の場合にこのタイプを選択します。

- 一時項目 (DEFINE) を使用してレポートに表示するデータを選択する。この場合、一時項目 (COMPUTE) を使用することはできません。一時項目 (COMPUTE) では、データが選択された後でデータの評価が行われるためです。
- 一時項目を使用してデータ値をソートする。一時項目 (COMPUTE) では、データがソートされた後にデータの評価が行われます。BY TOTAL 句を使用すると、このタイプのフィールドでソートを行うことができます。

一時項目 (COMPUTE) 次の場合にこのタイプを選択します。

- 合計値または演算接頭語 (合計値の演算を行う) を使用して一時項目を評価する。この場合、一時項目 (DEFINE) を使用することはできません。一時項目 (DEFINE) では、合計が計算される前に評価が行われるためです。
- データ構造内の異なるパスに存在するフィールドを使用して一時項目を評価する。この場合、一時項目 (DEFINE) を使用することはできません。一時項目 (DEFINE) では、異なるパスのデータ間で関係が確立される前に評価が行われるためです。

一時項目 (DEFINE) の定義

一時項目 (DEFINE) をリクエストで使用する場合、データソースの実フィールドのように扱うことができます。一時項目 (DEFINE) の値を決定する計算は、実フィールドにおいて選別条件を満たすレコードが取得されてから、その各レコードで実行されます。式の結果は、データソースに実際に保存されている実フィールドのように扱われます。

一時項目 (DEFINE) は、次の方法で定義することができます。

- **マスターファイル**　この一時項目 (DEFINE) は、レポートの作成にデータソースが使用されている限り、常に使用することができます。この一時項目 (DEFINE) は、JOIN または DEFINE FILE コマンドでクリアすることはできません。
- **プロシジャ**　プロシジャで作成された一時項目 (DEFINE) は、そのプロシジャでのみ持続します。

ヒント：現在の環境で KEEPDEFINES パラメータがサポートされている場合は、KEEPDEFINES を ON または ALL に設定しておくと、その後に使用する JOIN コマンドにより定義済みの一時項目 (DEFINE) がクリアされることを防ぐことができます。詳細は、データソースの結合 (JOIN)を参照してください。

一時項目 (DEFINE) 作成時の注意

- KEEPDEFINES パラメータを使用せずに JOIN コマンドを発行すると、マスターファイルで定義した一時項目 (DEFINE) を除いて、対象データソースで定義済みの一時項目 (DEFINE) がすべてクリアされます。
- ソースファイルの一時項目を使用してデータ構造を結合する場合は、JOIN コマンドの後に DEFINE コマンドを使用するようにします。そうしない場合は、JOIN コマンドで一時項目がクリアされます。結合したデータソースでレポートを作成する方法についての詳細は、データソースの結合 (JOIN)を参照してください。
- 式に使用するフィールドがマスターファイルに存在しない、または定義されていない場合は、WITH コマンドを使用して、定義した計算の論理ホームを識別します。
- WITH を使用すると、たとえば下位セグメントでインスタンスを集計する場合など、割り当て予定のセグメントより下位のセグメントに一時項目 (DEFINE) の論理ホームを移動させることができます。
- 複数のデータソースを使用する必要がある場合は、マスターファイルで定義したフィールドに加えて、これらの複数データソースのフィールドを同時に定義すること

ができます。すべての一時項目 (DEFINE) と実フィールドを合計した長さは、32,000 バイトを超えることはできません。

- リクエストで指定した一時項目 (DEFINE) は、表示フィールド上限数の対象になります。リクエストで使用可能な表示フィールドの最大数を決定する方法についての詳細は、レポートデータの表示を参照してください。
- 一時項目 (DEFINE) を使用できるのは、データソースを使用してレポートを作成する場合に限られます。
- DEFINE コマンドでは、式の左辺に修飾フィールド名を使用することはできません。複数のセグメントに同一のフィールド名が存在し、そのフィールドを再定義または再計算する必要がある場合は、REDEFINES コマンドを使用します。
- たとえば、x=x+1 などの自己参照の DEFINE を使用する場合は、AUTOPATH が無効になります。詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。
- 一時項目 (DEFINE) を定義する式に使用するフィールド名は、一重引用符 (') または二重引用符 (") で囲むことはできません。引用符で囲まれたすべての文字列は、フィールド参照ではなく、リテラル参照として扱われます。
- DEFINE FILE コマンドは、フォーマットを再定義した場合を除いて、マスターファイルに存在する同一名の DEFINE を上書きします。ただし、フォーマットの再定義は禁止されています。

一時項目 (DEFINE) の作成

リクエストを開始する前に、次の構文を記述します。

```
DEFINE FILE filename[.view_fieldname] [CLEAR|ADD]
fieldname[/format] [(GEOGRAPHIC_ROLE = georole]
         [,TITLE = 'line1[,line2 ...']]
         [,DESCRIPTION = 'description'])] = expression;
fieldname[/format][WITH realfield] = expression;
fieldname[/format] REDEFINES qualifier.fieldname = expression;
.
.
.
END
```

説明

filename

一時項目 (DEFINE) を定義するデータソース名です。

リクエストで代替ビューを指定する場合は、view_fieldname とともに filename を使用します。

一時項目 (DEFINE) の定義に使用するすべてのフィールドは、データソースの単一パスに存在していなければなりません。単一パスに存在しない場合は、代替ビューを使用することができます。この場合、代替ビューの DEFINE コマンドを使用する必要があります。代替ビューでは、一時項目 (DEFINE) に修飾フィールド名を使用することはできません。代替ビューについての詳細は、データ構造回転による検索速度の改善を参照してください。

DEFINE FILE コマンドラインは一時項目 (DEFINE) 定義とは別の行にする必要があります。

view_fieldname

リクエストにおいて、代替ビューの基準となるフィールドです。使用するフィールドが標準ビューの単一パスに存在しない場合は、代替ビューを使用する必要があることがあります。

CLEAR

指定したデータソースに関連する定義済みの一時項目 (DEFINE) をクリアします。デフォルト値は CLEAR です。

ADD

既存の一時項目 (DEFINE) をクリアせずに、データソースに新しい一時項目 (DEFINE) を追加します。ADD オプションを指定しない場合は、CLEAR オプションを指定した場合と同様の結果が得られます。

fieldname

WebFOCUS フィールドの名前規則に準拠する名前です。FOCUS データソースのインデックスフィールド名は、12 バイト以下にする必要があります。新しく定義する一時項目 (DEFINE) 名に使用したり、マスターファイルで宣言した既存フィールドを再定義する際の名前に使用したりすることができます。

この名前には、文字、数字、アンダースコア (_) を任意に組み合わせて使用することができますが、名前の先頭には必ず文字を使用します。

なお、Cn、En、Xn タイプのフィールド名を使用することはできません (ここで、n は 1 桁または 2 桁の連続番号を表します)。これらのフィールド名は、その他の目的のために予約されています。

format

フィールドのフォーマットです。デフォルト値は D12.2 です。

georole

有効な地理的役割です。以下は、デフォルト地理的役割のリストを示しています。

- **ADDRESS_FULL** 完全な住所です。
- **ADDRESS_LINE** 番地です。
- **CITY** 都市名です。
- **CONTINENT** 大陸名です。
- **COUNTY** 郡名です。
- **COUNTRY** 国名です。
- **GEOMETRY_AREA** 面ジオメトリです。
- **GEOMETRY_LINE** 線ジオメトリです。
- **GEOMETRY_POINT** 点ジオメトリです。
- **LATITUDE** 緯度
- **LONGITUDE** 経度
- **POSTAL_CODE** 郵便番号です。
- **STATE** 州名です。

WITH realfield

一時項目 (DEFINE) を、実フィールドが存在するデータソースセグメントに関連付けます。

line1, line2...

AS 句で上書きする場合を除き、一時項目 (DEFINE) のフィールドタイトルとして表示する行です。

description

一時項目 (DEFINE) に関連付ける説明です。一重引用符で囲みます。この説明は、マスターファイル参照ツールに表示されます。

REDEFINES qualifier.fieldname

フィールド名が複数のセグメントに存在する場合に、そのフィールドの再定義または再計算を行えるようにします。フィールドの再定義時にフォーマットを変更する場合、新しいフォーマットの長さは、元のフォーマットの長さ以下にする必要があります。また、文字データタイプと数値データタイプを切り替えることはできません。

expression

演算および論理による式または関数で、fieldname の値を確定するために評価されます。詳細は、式の使用を参照してください。複数の式を指定する場合は、それぞれの式の末尾にセミコロン (;) を挿入する必要があります。ただし、最後の式の末尾のセミコロン (;) は省略することもできます。

式に使用するフィールドとして、実データフィールド、クロスリファレンスまたは結合されたデータソースのデータフィールド、定義済みの一時項目 (DEFINE) のいずれかを指定することができます。

END

DEFINE FILE コマンドの終了に必要です。プロシジャでは、END コマンドは、独立した行として記述する必要があります。

注意：一時項目 (DEFINE) の MISSING 属性についての詳細は、ミッシングフィールド値を参照してください。

一時項目 (DEFINE) の定義

次のリクエストでは、DELIVER_AMT の値を OPENING_AMT で除算して RATIO の値を計算します。DEFINE コマンドにより一時項目 の RATIO が作成され、リクエストではこの一時項目がデータソースの実フィールドのように扱われます。

```
DEFINE FILE SALES
RATIO = DELIVER_AMT/OPENING_AMT;
END
TABLE FILE SALES
PRINT DELIVER_AMT AND OPENING_AMT AND RATIO
WHERE DELIVER_AMT GT 50
END
```

出力結果は次のとおりです。

```
DELIVER_AMT  OPENING_AMT           RATIO
-----------  -----------           -----
         80           65            1.23
        100          100            1.00
         80           90             .89
```

一時項目 (DEFINE) の複数定義

1 つのデータソースに対して複数の一時項目 (DEFINE) を定義し、そのいくつかまたはすべてをリクエストに使用したい場合があります。ADD オプションを使用すると、既存の一時項目 (DEFINE) をクリアせずに新しい一時項目 (DEFINE) を追加することができます。ADD オプションを使用しないと、データソースで定義済みの一時項目 (DEFINE) がクリアされます。

特定のデータソースの一時項目 (DEFINE) をクリアするには、CLEAR オプションを使用します。

一時項目 (DEFINE) の追加

```
DEFINE FILE filename ADD
```

説明

filename

データソースの名前です。

一時項目 (DEFINE) の追加

次の注釈付きの例は、一時項目 (DEFINE) の ADD および CLEAR オプションを使用する方法を示しています。

```
1.DEFINE FILE CAR
   ETYPE/A2=DECODE STANDARD (OHV O OHC O ELSE L);
   END
2.DEFINE FILE CAR ADD
```

```
TAX/D8.2=IF MPG LT 15 THEN .06*RCOST
   ELSE .04*RCOST;
FCOST = RCOST+TAX;
END
```

1. 最初の DEFINE コマンドにより、CAR データソースに対して一時項目 (DEFINE) の TYPE が作成されます。DECODE 関数についての詳細は、『ibi™ WebFOCUS® 関数リファレンス』を参照してください。
2. CAR データソースに対して、2 つ以上の一時項目 (DEFINE) TAX と FCOST を作成します。ADD オプションを使用することにより、これ以降のリクエストで参照できる一時項目 (DEFINE) は、ETYPE、TAX、FCOST の 3 つになります。

一時項目 (DEFINE) の表示

すべての一時項目 (DEFINE) を表示するには、? DEFINE コマンドを使用します。

一時項目 (DEFINE) を表示するには

```
?DEFINE
```

詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

一時項目 (DEFINE) を表示するには

DEFINE ツールで [一時項目 (DEFINE)] タブをクリックします。

一時項目 (DEFINE) のクリア

プロシジャで作成された一時項目 (DEFINE) は、次の方法でクリアすることができます。

- DEFINE FILE *filename* CLEAR コマンド。

- 同一データソースの後に続く ADD オプションを含まない DEFINE コマンド。
- JOIN を作成する。データソースに対して JOIN を作成すると、マスターファイルで定義された一時項目 (DEFINE) を除いて、そのデータソースの既存の一時項目 (DEFINE) がすべてクリアされます。この方法は、式で使用された一時項目 (DEFINE) に影響を及ぼす場合があります。
- FIELDNAME SET パラメータの値を変更する。

マスターファイルで指定した一時項目 (DEFINE) は、プロシジャで作成された一時項目 (DEFINE) とは異なり、上記の方法ではクリアされません。

すべてのデータソースの一時項目 (DEFINE) をクリアするには、次のコマンドを使用します。

```
DEFINE FILE * CLEAR
END
```

一時項目 (DEFINE) のクリア

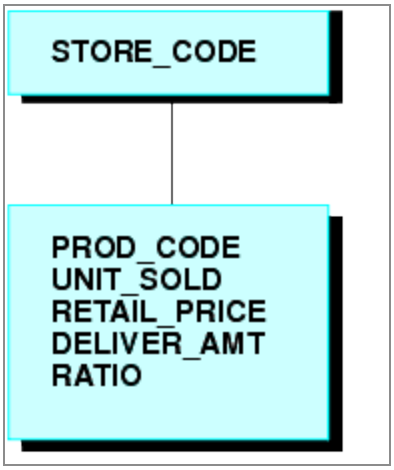
次の注釈付きの例は、一時項目 (DEFINE) の CLEAR オプションを使用する方法を示しています。

```
1.DEFINE FILE CAR
   ETYPE/A2=DECODE STANDARD (OHV O OHC O ELSE L);
   END
2.DEFINE FILE CAR CLEAR
   COST = RCOST-DCOST;
   END
```

1. 最初の DEFINE コマンドにより、CAR データソースに対して一時項目 (DEFINE) の TYPE が作成されます。DECODE 関数についての詳細は、『ibi™ WebFOCUS® 関数リファレンス』を作成してください。
2. CLEAR オプションにより、定義済みの一時項目 (DEFINE) がすべてクリアされ、DEFINE で最後に定義した一時項目 (DEFINE) の COST のみがこれ以降のリクエストで有効になります。

一時項目 (DEFINE) セグメント位置の設定

データソースの実フィールドと同様に、一時項目 (DEFINE) にはデータソース構造内で論理的な位置が定義されています。一時項目 (DEFINE) の論理ホームは最下位のセグメントにあり、このセグメントにアクセスして式が評価され、これによりフィールドの実行時間が決まります。ここでは、次のデータソース構造で DEFINE コマンドを使用する場合について考察します。




```
DEFINE RATIO = DELIVER_AMT/RETAIL_PRICE ;
```

RATIO の式には、データソースの実フィールドが 1 つ以上含まれています。レポートを作成する上では、RATIO フィールドはマスターファイルの実フィールドと同様に扱われ、その位置は最下位のセグメントです。

アプリケーションの中には、データソースの実フィールドを含まない式で一時項目 (DEFINE) を評価できるものもあります。そのような式は、一時項目またはリテラルのみを参照します。以下はその例です。

```
NCOUNT/I5 = NCOUNT+1;
```

または

```
DATE/YMD = '19990101';
```

いずれの式にもデータソースのフィールドが含まれていないため (NCOUNT とリテラルはともにマスターファイルに存在しない)、データソースでの一時項目 (DEFINE) の論理的な位置を特定することはできません。この場合は、式を配置するセグメントを指定する必要があります。一時項目 (DEFINE) を特定のセグメントに関連付けるには、WITH 句を使用します。

WITH の後に続けるフィールド名には、マスターファイルの任意の実フィールドを指定することができます。

FOCUS データソースでは、一時項目 (DEFINE) の外部インデックスを使用して、データの取得速度を上げることができます。この場合、一時項目 (DEFINE) を含むセグメントの外で、外部インデックスをターゲットセグメントに関連付けます。外部インデックスについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

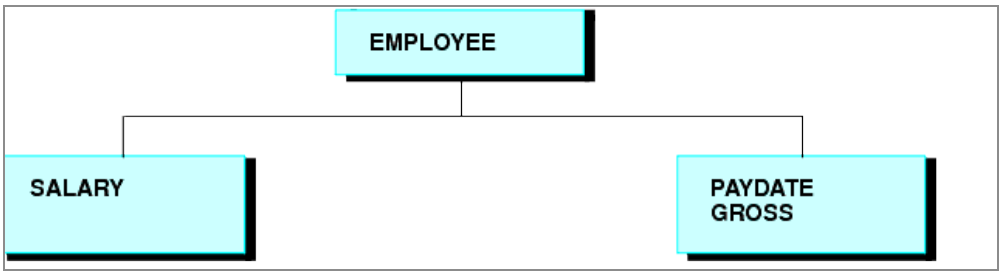
セグメント位置の設定

次の例では、NCOUNT フィールドが UNITS フィールドと同一のセグメントに配置されます。新しいセグメントインスタンスが取得されるたびに、NCOUNT が計算されます。

```
DEFINE FILE GGSALES
NCOUNT/I5 WITH UNITS = NCOUNT+1;
END
```

マルチパスデータソースでの一時項目 (DEFINE) の定義

一時項目 (DEFINE) の演算にはデータソース内のすべてのセグメントのフィールドを含めることができますが、これらのフィールドは最上位から最下位まで連続する一意のセグメントパスに存在する必要があります。もちろん、異なる一時項目 (DEFINE) は異なるパスに存在することができます。ここでは、次のデータソース構造について考察します。



このデータソース構造に対して次の式を記述することはできません。

```
NEWAMT = SALARY+GROSS;
```

この構造では、特定の EMPLOYEE に対して複数の SALARY セグメントが存在する可能性があり、SALARY と GROSS 間のセグメント関係が明確に特定されないため、この式は無効です。

このような演算を実行する場合は、代替ビューを使用することができます。詳細は、パフォーマンスの改善を参照してください。

一時項目 (DEFINE) 計算速度の向上

計算の速度を上げるために、一時項目 (DEFINE) はマシンコードにコンパイルされます。

DEFINE FILE SAVE と RETURN による一時項目 (DEFINE) の保持

場合によっては、新しいコードを既存のアプリケーションに追加する必要があります。コードを追加する際に、既存の一時項目 (DEFINE) の名前を使用して、その一時項目 (DEFINE) を不用意に上書きしてしまう可能性があります。

DEFINE FILE SAVE コマンドを使用すると、一時項目 (DEFINE) に対する新しいコンテキストが作成されます。新しいコンテキストでは、レイヤまたはコマンド環境が新しく作成されます。この新しい環境に最初に入った際に、既存のレイヤで定義されていたすべての一時項目 (DEFINE) を新しいレイヤでも使用できるようになります。一時項目 (DEFINE) の定義を上書きまたはクリアした場合、その影響を受けるのは現在のレイヤのみです。一時項目 (DEFINE) の定義をそのまま保持した状態でデフォルトのコンテキストに戻すには、DEFINE FILE RETURN コマンドを使用します。

これにより、新しいアプリケーションで作成したすべての一時項目 (DEFINE) を削除しても、既存の一時項目 (DEFINE) に影響を与えることなく呼び出しアプリケーションに戻ることができます。

DEFINE FILE SAVE および DEFINE FILE RETURN の例については、データソースの結合 (JOIN)を参照してください。

注意：DEFINE FILE SAVE コマンドの後に JOIN コマンドを発行することができます。ただし、JOIN がすでに有効になっている場合に JOIN コンテキストをクリアするには、JOIN CLEAR コマンドを発行する必要があります。DEFINE FILE SAVE コマンドを発行した後に一時項目 (DEFINE) および DEFINE FILE ADD コマンドのみを発行した場合は、DEFINE FILE RETURN コマンドを発行してこのコンテキストをクリアすることができます。

一時項目 (DEFINE) の上書き防止

```
DEFINE FILE filename SAVE
fld1/format1=expression1;
fld2/format2=expression2;
END
TABLE FILE filename ...
MODIFY FILE filename ...
DEFINE FILE filename RETURN
END
```

説明

SAVE

一時項目 (DEFINE) に対する新しいコンテキストを作成します。

filename

新しいコンテキストの対象となるマスターファイル名です。このマスターファイルでは、DEFINE FILE RETURN コマンドを発行する前に他の一時項目 (DEFINE) を連続的に指定します。

RETURN

DEFINE FILE SAVE で現在のコンテキストが作成された場合、そのコンテキストをクリアし、以前のコンテキストに戻します。

レポートフィールドへの動的フォーマットの適用

動的フォーマットにより、フィールド内の特定のデータにタイプ異なるフォーマットを適用することができます。この場合、動的なデータ設定を含む一時項目が使用されます。

動的フォーマットを使用してレポートフィールドにフォーマットを設定する場合は、あらかじめレポートを作成し、一時項目をそのレポートフィールドに適用しておく必要があります。たとえば、通貨の単位は国ごとに異なりますが、小数点を使用しない日本の通貨単位や小数点 2 桁を使用するイギリスの通貨単位など、さまざまなフォーマットを設定した一時項目を作成することができます。これらの通貨フォーマットは、動的フォーマットと見なされます。この動的フォーマットが設定された一時項目を、たとえば Sales フィールドに適用す

ることができます。レポート内の Sales フィールドには、国ごとに異なる通貨フォーマットが適用されます。

フォーマット指定を含めることのできるフィールドは次のとおりです。

- データソースの実フィールド。
- DEFINE コマンドで指定した一時項目。
- マスターファイルで指定した一時項目 (DEFINE)。
- COMPUTE コマンド。一時項目を COMPUTE コマンドで作成する場合は、リクエストで、フォーマットの再設定の前に、コマンドを指定する必要があります。

フォーマットを含むフィールドは文字で、長さは少なくとも 8 バイトの必要があります。フォーマット設定には、最初の 8 バイトのみが使用されます。

フィールドを基準とするフォーマットには、元のフィールドより長い値を指定をすることもできます。ただし、新しい長さが元の長さよりも 3 分の 1 以上長くなる場合は、レポートの列幅が値の表示に十分ではなくなることがあります。その場合、フィールドの値はアスタリスク (*) で表示されます。

フィールドを基準とするフォーマットは、あらゆるタイプのフィールドに適用することができます。ただし、新しいフォーマットは、元のフォーマットに一致させておく必要があります。

- 数値フィールドのフォーマットを再設定し、任意の編集フォーマットオプションを持つ別の数値フォーマットにすることができます。
- 文字フィールドのフォーマットを再設定し、異なる長さにすることができます。
- 日付フィールドのフォーマットは、他のタイプの日付フォーマットに再設定することができます。
- 日付時間フィールドのフォーマットは、他の日付時間フォーマットに再設定することができます。

フィールドを基準とするフォーマットが無効、または変換できないタイプを指定した場合は、レポート出力の該当フィールドに、プラス記号 (++++) が表示されます。

フォーマットフィールドの定義および適用

- DEFINE コマンドでは、次のように定義します。

```
DEFINE FILE filename
format_field/A8 = expression;
END
```

- マスターファイルでは、次のように定義します。

```
DEFINE format_field/A8 = expression; $
```

- リクエストでは、次の構文を使用します。

```
COMPUTE format_field/A8 = expression;
```

説明

format_field

行のフォーマットを設定するフィールド名です。

expression

フォーマット値をフォーマットフィールドに割り当てる式です。

フォーマットフィールドの定義後、それをリクエストに適用することができます。

```
TABLE FILE filename
display fieldname/format_field[/just]
END
```

説明

display

任意の有効な表示コマンドです。

fieldname

フォーマットを再設定するリクエストのフィールドです。

format_field

フォーマットを設定するフィールド名です。フォーマットを設定するフィールドの名前が明示フォーマットと重複する場合は、明示フォーマットが使用されます。たとえば、「I8」という名前のフィールドは明示的フォーマット I8 として解釈されるため、フィールドを基準とするフォーマットの再設定には使用できません。

just

位置揃えオプションの L、R または C です。位置揃えオプションはフォーマットフィールドの前または後ろにスラッシュ (/) でフォーマットと区別して配置することができます。

フィールドを基準とするフォーマット再設定を行う際の注意

- フィールドを基準とするフォーマット再設定は、TABLE および TABLEF でサポートされています。スタイルシート、JOIN、すべてのデータソースタイプで使用することができます。
- フィールドを基準とするフォーマット再設定は、MATCH、GRAPH、RECAP、FOOTING、HEADING、テキストフィールドではサポートされません。
- DEFINE または COMPUTE コマンドを使用してフォーマットフィールドを作成することはできますが、フィールドを基準とするフォーマットを一時項目 (DEFINE または COMPUTE) に適用することはできません。
- フィールドを基準とするフォーマット再設定は、BY フィールドで使用することはできません。ACROSS フィールドでは使用することができます。
- フィールドを基準とするフォーマット再設定を使用してレポートフィールドを作成する場合、列の合計または中間合計に使用するフォーマットは 1 つ前の詳細データの行から取得されます。
- 明示フォーマットの再設定では、フォーマットを再設定する各フィールドに対して内部的に 2 つの表示フィールドが作成されます。フィールドを基準とするフォーマット再設定では、3 つの表示フィールドが作成されます。
- フォーマットによって、レポート処理の最終出力段階で適用されるものと演算の実行前に適用されるものがありますが、フィールドを基準とするフォーマットは、レポート処理の最終出力段階で適用されます。そのため、動的にフォーマットが再設定されたフィールドは、元のフォーマットを使用して、集計を含む演算を実行しますが、特定のフォーマットでフォーマットが再設定されたフィールドは、演算時に新しいフォーマットを使用します。したがって、精度の低いパック 10 進数フィールドを使用した際に、端数処理により最終出力の数値に差異が生じる場合があります。
- フォーマットが再設定された各フィールドに対して 3 つの表示フィールドがファイルに保存されますが、フィールドを基準とするフォーマット再設定は HOLD ファイルの文字フィールドに対しても使用することができます。SET HOLDLIST=PRINTONLY を

指定して、HOLD ファイルに余分なフィールドが継承されないようにします。

- 小数の桁数が行ごとに異なる場合、レポート出力では小数点の位置は揃いません。

動的にフォーマットが設定されたフィールドの作成

次のリクエストは、CATEGORY フィールドの値に従って DOLLAR2 フィールドのフォーマットを設定し、フォーマットの再設定を動的に行った場合と静的に行った場合の数値の相違を示します。

```
DEFINE FILE GGSALES
MYFORMAT/A8=DECODE CATEGORY ('Coffee' 'P15.3' 'Gifts' 'P15.0' ELSE
'P15.2');
DOLLARS2/P15.2 = DOLLARS + .5;
END
```

```
TABLE FILE GGSALES
SUM DOLLARS2/MYFORMAT AS 'Dynamic' DOLLARS2/P10.2 AS 'Specific'
BY CATEGORY
ON TABLE SUBTOTAL
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
END
```

下図は、出力結果を示しています。

Category	Dynamic	Specific
Coffee	17232175.000	17232175.00
Food	17230053.00	17230053.00
Gifts	11696221.	11696220.50
TOTAL	46158449.	46158448.50

SQL.Function 構文による関数呼び出しのリレーショナルエンジンへの直接転送

SQL アダプタは、特定の SQL スカラ関数を呼び出す一時項目 (DEFINE) を、リレーショナルエンジンに渡して処理させることができます。これにより、SQL 関数と同等の関数が WebFOCUS 言語に存在しない場合でも、その SQL 関数をリクエストで使用することが可能になります。この関数は、行ベースの関数であり、フィールド名または定数のカンマ区切りリストで構成されるパラメータリストが含まれている必要があります。この関数を式で参照するには、関数名の先頭に「SQL」を追加します。

マスターファイルで一時項目 (DEFINE) が定義されている場合、自動パススルー (APT) の要件を満たす TABLE リクエストと SQL リクエストの両方でその一時項目 (DEFINE) を使用することができます。一時項目 (DEFINE) を DEFINE FILE コマンドで作成した場合は、TABLE リクエストでその一時項目 (DEFINE) を使用することができます。関数名およびパラメータは、変換されずにリレーショナルエンジンに渡されます。そのため、一時項目 (DEFINE) を作成する式を最適化しておく必要があります。最適化されていない場合、リクエストは失敗します。

直接呼び出しの SQL 関数を使用する際の注意

- SQL.function 呼び出しを含む式は、最適化する必要があります。最適化されていない場合、式は失敗して次のメッセージが表示されます。

  ```
  (FOC32605) SQL. に特化する構文の式は最適化できません
  ```

- この関数は、行ベースのスカラ関数であり、フィールド名または定数のカンマ区切りリストで構成されたパラメータリストが含まれている必要があります。関数でカンマ区切りの値リスト以外を使用する場合、SQL. 構文を使用して一時項目 (DEFINE) を渡すことはできません。
- 定数の一時項目 (DEFINE) には、WITH 句を使用してセグメントパスを割り当てる必要があります。
- 式は一時項目 (DEFINE) として宣言する必要があります。これは、SQL 関数のパラメータとしてサポートされます。

- データタイプは、SQL 関数のパラメータとしてサポートされません。データタイプ引数の一例として、CONVERT 関数に使用する CHAR および INT、CHAR 関数に使用する ISO、EUR、JIS、USA などがあります。

リクエストでの SQL CONCAT 関数の呼び出し

次の例では、「WebFOCUS Retail デモ」サンプルを使用します。このサンプルのリレーショナルアダプタ対応のデータソースを作成するには、このサンプルの格納先とするアプリケーションディレクトリを右クリックし、コンテキストメニューから [新規]、[チュートリアル] を順に選択します。次に、[チュートリアル] ドロップダウンリストから [WebFOCUS - Retail デモ] を選択し、[作成] をクリックします。

次のリクエストは、[WebFOCUS Retail デモ] データソースに対して実行され、SQL CONCAT 関数を使用して製品カテゴリと製品サブカテゴリを連結します。

```
SET TRACEUSER = ON
SET TRACEOFF = ALL
SET TRACEON = STMTRACE//CLIENT
SET TRACESTAMP=OFF
SET XRETRIEVAL = OFF

DEFINE FILE WF_RETAIL
CAT_SUBCAT/A50 = SQL.CONCAT(PRODUCT_CATEGORY, PRODUCT_SUBCATEG);
END

TABLE FILE WF_RETAIL
PRINT CAT_SUBCAT
BY PRODUCT_CATEGORY NOPRINT
END
```

トレース出力に示すように、SQL 関数呼び出しが RDBMS に渡されています。

```
SELECT
CONCAT(T2."PRODUCT_CATEGORY",T2."PRODUCT_SUBCATEG"),
T2."PRODUCT_CATEGORY",
T2."PRODUCT_SUBCATEG"
FROM
wfr_product T2
ORDER BY
T2."PRODUCT_CATEGORY"
FOR FETCH ONLY;
```

一時項目 (COMPUTE) の作成

一時項目 (COMPUTE) とは、選別条件を満たすすべてのデータが取得、ソート、集計された後に評価される一時項目のことです。一時項目 (COMPUTE) は、指定したリクエストでのみ使用することができます。

COMPUTE コマンドは、リクエストの本文で指定します。表示コマンドの後には、オプションとして AND を指定します。1 つの COMPUTE コマンドで複数フィールドの計算を行うことができます。

一時項目 (COMPUTE) 使用上の注意

一時項目 (COMPUTE) を使用する際には、次の規則が適用されます。

- オプションとして AS、IN、NORPINT などの COMPUTE 句を指定し、これらの句の後で別のフィールドの計算を行う場合は、そのフィールドを指定する前に COMPUTE コマンドまたは AND COMPUTE コマンドを繰り返し指定する必要があります。
- 列合計および行合計の名前や位置揃えを変更することができます。詳細は、合計と中間合計の追加を参照してください。
- COMPUTE コマンドの式には、演算接頭語を使用したフィールドを含めることができます。詳細は、演算接頭語による表示フィールドの操作を参照してください。有効な式についての詳細は、式の使用を参照してください。
- COMPUTE コマンドで参照するフィールドは、表示フィールド上限数の対象となり、内部マトリックスに表示されます。リクエストで使用可能な表示フィールドの最大数を決定する方法についての詳細は、レポートデータの表示を参照してください。
- 一時項目 (COMPUTE) を定義する式に使用するフィールド名は、一重引用符 (') または二重引用符 (") で囲むことはできません。引用符で囲まれたすべての文字列は、フィールド参照ではなく、リテラル参照として扱われます。
- COMPUTE に ACROSS COLUMNS 句を使用する場合は、COLUMNS を最後に指定する必要があります。

```
ACROSS acrossfield [AND] COMPUTE compute_expression; COLUMNS values
```

一時項目 (COMPUTE) の作成

```
COMPUTE fld [/format] [(GEOGRAPHIC_ROLE = georole)] = expression;
  [AS 'title'] [NOPRINT] [IN [+n]]
```

説明

fld

一時項目 (COMPUTE) の名前です。

この名前は、WebFOCUS フィールドの名前規則に準拠する任意の名前にすることができます。

なお、Cn、En、Xn タイプのフィールド名を使用することはできません (ここで、n は 1 桁または 2 桁の連続番号を表します)。これらのフィールド名は、他の目的のために予約されています。

format

フィールドのフォーマットです。デフォルト値は D12.2 です。

georole

有効な地理的役割です。以下は、デフォルト地理的役割のリストを示しています。

- **ADDRESS_FULL** 完全な住所です。
- **ADDRESS_LINE** 番地です。
- **CITY** 都市名です。
- **CONTINENT** 大陸名です。
- **COUNTY** 郡名です。
- **COUNTRY** 国名です。
- **GEOMETRY_AREA** 面ジオメトリです。
- **GEOMETRY_LINE** 線ジオメトリです。
- **GEOMETRY_POINT** 点ジオメトリです。
- **LATITUDE** 緯度
- **LONGITUDE** 経度

- **POSTAL_CODE**　郵便番号です。
- **STATE**　州名です。

expression

演算および論理による式または関数です。詳細は、式の使用を参照してください。式に使用する各フィールドは、リクエストの一部でなければなりません。それぞれの式の末尾にはセミコロン (;) を入れる必要があります。

NOPRINT

フィールドを表示しません。詳細は、レポートページのレイアウトを参照してください。

AS 'title'

一時項目 (COMPUTE) の名前を変更します。

IN [+n]

フィールドの位置を指定します。IN は、STYLEMODE SET パラメータが FIXED または OFF に設定されている場合に限り、HTML レポートで使用することができます。

計算を伴わない一時項目 (COMPUTE) の作成

```
COMPUTE fld [/format]= ;
```

説明

fld

一時項目 (COMPUTE) の名前です。

この名前は、WebFOCUS フィールドの名前規則に準拠する任意の名前にすることができます。

なお、Cn、En、Xn タイプのフィールド名を使用することはできません (ここで、n は 1 桁または 2 桁の連続番号を表します)。これらのフィールド名は、他の目的のために予約されています。

format

フィールドのフォーマットです。デフォルト値は D12.2 です。

一時項目 (COMPUTE) の作成

次の例では、COMPUTE コマンドを使用して UNIT_SOLD および RETAIL_PRICE の製品データを基に一時項目の REVENUE を作成し、New York City に関するデータを表示します。フォーマットの D12.2M により、REVENUE フィールドのフォーマットが設定され、AS コマンドにより UNIT_SOLD および RETAIL_PRICE のデフォルトのフィールドタイトルが変更されます。REVENUE は、このリクエストでのみ有効です。

```
TABLE FILE SALES
HEADING CENTER
"NEW YORK PROFIT REPORT"
" "
SUM UNIT_SOLD AS 'UNITS,SOLD' RETAIL_PRICE AS 'RETAIL,PRICE'
COMPUTE REVENUE/D12.2M = UNIT_SOLD * RETAIL_PRICE;
BY PROD_CODE AS 'PROD,CODE'
WHERE CITY EQ 'NEW YORK'
END
```

出力結果は次のとおりです。

```
       NEW YORK PROFIT REPORT

PROD  UNITS  RETAIL
CODE  SOLD   PRICE           REVENUE
----  -----  ------          -------
B10      30    $.85           $25.50
B17      20   $1.89           $37.80
B20      15   $1.99           $29.85
C13      15   $1.99           $29.85
C14      18   $2.05           $36.90
C17      12   $2.09           $25.08
D12      20   $2.09           $41.80
E1       30    $.89           $26.70
E2       33    $.99           $32.67
E3       35   $1.09           $38.15
```

一時項目 (COMPUTE) の位置参照

COMPUTE コマンドでは、フィールドの参照に、名前よりもレポートフィールドの位置を指定した方が便利な場合があります。複数のレポートフィールドに対して共通のフィールドが指定されている場合、このオプションは特に役立ちます。

フィールドの参照は、同一のフィールド名をさまざまな方法で使用している場合に必要不可欠です。下図のように、表示コマンドを使用してフィールドを生成する場合、そのフィールドを表示するかどうかに関係なく、先頭のフィールドは C1、2 つ目のフィールドは C2 のように、それぞれを順に参照することができます。BY フィールド列はその数には含まれません。

位置指定によるフィールド参照の使用

次の例は、COMPUTE コマンドで位置指定のフィールド参照を使用する方法を示しています。

```
TABLE FILE CAR
SUM AVE.DEALER_COST
SUM AVE.DEALER_COST AND COMPUTE RATIO=C1/C2;
BY COUNTRY
END
```

表示コマンドを使用してフィールドを生成する場合、先頭のフィールド (AVE.DEALER_COST) は C1、2 つ目のフィールド (AVE.DEALER_COST BY COUNTRY) は C2 のように、それぞれを順に参照することができます。BY フィールド列はその数には含まれません。

出力結果は次のとおりです。

```
AVE                     AVE
DEALER_COST  COUNTRY    DEALER_COST          RATIO
-----------  -------    -----------          -----
      7,989  ENGLAND          9,463            .84
             FRANCE           4,631           1.73
             ITALY           10,309            .77
             JAPAN            2,756           2.90
             W GERMANY        7,795           1.02
```

一時項目 (COMPUTE) での ACROSS 句の使用

COMPUTE コマンドを ACROSS 句の直後に発行すると、集計タイプの計算がすべてのフィールドに対して一度だけ実行されます。COMPUTE は表示コマンドの一部として使用されるため、新しいフィールドがそれぞれの値一式に対して計算されます。

表示コマンドの一部としての COMPUTE の使用

```
TABLE FILE SALES
SUM UNIT_SOLD
COMPUTE NEWVAL = UNIT_SOLD * RETAIL_PRICE;
ACROSS CITY
END
```

次は出力結果の 1 ページ目です。

```
CITY
NEW YORK             NEWARK               STAMFORD             UNIONDALE
UNIT_SOLD    NEWVAL UNIT_SOLD    NEWVAL UNIT_SOLD    NEWVAL UNIT_SOLD
NEWVAL
-------------------------------------------------------------------------------
---
      162  1,764.18        42    104.16       376  4,805.28        65
297.70
```

一時項目 (COMPUTE) での ACROSS 句の使用

次の例では、COMPUTE コマンドで 「C1」、「C2」、「C3」、「C4」、「C5」、「C6」という位置指定のフィールド列参照を使用し、COMPUTE コマンドを ACROSS 句の直後に発行します。このレポートで COMPUTE コマンドは一度だけ実行され、その結果がすべてのソートグループの右側に表示されます。

```
TABLE FILE SALES
SUM UNIT_SOLD AND RETURNS
WHERE DATE GE '010' AND DATE LE '1031'
```

```
ACROSS DATE
COMPUTE
TOT_UNITS/D5=C1 + C3 + C5;
TOT_RETURNS = C2 + C4 + C6;
END
```

出力結果は次のとおりです。

```
DATE
10/17       10/18     10/19       TOT_UNITS   TOT_RETURNS              TOT_
UNITS   TOT_RETURNS
UNIT_SOLD   RETURNS   UNIT_SOLD   RETURNS     UNIT_SOLD     RETURNS
-------------------------------------------------------------------------
-----------------
162         15        78          2           29            1          269
     18.00
```

一時項目のソート

一時項目 (DEFINE または COMPUTE) を基準にしてレポートをソートすることができます。一時項目 (COMPUTE) を基準にしてソートするには、リクエストに BY TOTAL 句を使用する必要があります。

一時項目 (COMPUTE) の選別

COMPUTE コマンドで作成した値を選別するには、WHERE TOTAL または WHERE_GROUPED テストを使用します。詳細は、レコードの選択を参照してください。

フィールド参照番号の割り当て

CNOTATION パラメータは、リクエストに対して作成された内部マトリックスの各列に連続列番号を割り当てます。レポートで使用する列に対して、列参照番号の作成を制御するには、CNOTATION 列表記コマンドを使用します。

列番号は、内部マトリックスのフィールドを参照するため、データの取得および集計が完了した後で割り当てられます。フィールドが作成され、レポートに表示される場合、その

フィールドは内部マトリックスに保存されます。また、レポートに表示されないフィールドでも、内部マトリックスに保存される場合もあります。内部マトリックスに保存されるフィールドには、一時項目、フォーマットが再設定されたフィールド値、BY フィールド、NOPRINT オプションが指定されたフィールド、FORECAST または REGRESS などの RECAP 計算があります。デフォルト設定では、内部マトリックスのその他すべてのフィールドに列番号が割り当てられるようになっています。つまり、リクエストでフィールド値を正しく参照するには、内部的に作成されたすべてのフィールドを考慮する必要があります。

SET CNOTATION コマンドを使用しフィールド参照番号のデフォルト割り当てを変更することができます。このコマンドはフィールド番号をレポート出力に表示するフィールドのみ、またはレポート出力で参照されるすべてのフィールドに割り当てることができます。このパラメータを COMPUTE および RECAP コマンドで使用すると、リクエスト内でそのフィールドを参照することができます。

作成する列番号の制御

```
SET CNOTATION={ALL|PRINTONLY|EXPLICIT}
```

説明

ALL

内部マトリックスのすべての列に列番号を割り当てます。デフォルト値は ALL です。

PRINTONLY

レポート出力に表示するフィールドのみに列番号を割り当てます。

EXPLICIT

表示または非表示に関係なく、リクエストで参照されるすべてのフィールドに列参照番号を割り当てます。

リクエストでの列表記の使用

次のいずれかの方法で、リクエストに列参照を作成することができます。

- FML 以外のリクエストでは、列番号の前に C を付けます。
- FML リクエストでは、行ラベルと連携させて列番号をインデックスとして使用しま

す。このタイプの表記では、特定の列、相対列番号、連続した列、一連の列を指定することができます。

- FML リクエストでは、表記 E(r, c) を使用して特定のセルを参照します。ここで、r は行番号、c が列番号を表します。

FML 以外のリクエストで CNOTATION=ALL を使用した列表記

次のリクエストには CNOTATION=ALL が設定されていますが、フォーマットの再設定で新しい列が作成されるため、C1 と C2 の積 (PRODUCT) は、TRANSTOT と QUANTITY の積にはなりません。

```
SET CNOTATION = ALL
TABLE FILE VIDEOTRK
SUM TRANSTOT/D12.2 QUANTITY/D12.2
AND COMPUTE
PRODUCT = C1 * C2;
BY TRANSDATE
END
```

出力結果は次のとおりです。

```
TRANSDATE        TRANSTOT        QUANTITY         PRODUCT
---------        --------        --------         -------
 91/06/17           57.03           12.00       3,252.42
 91/06/18           21.25            2.00         451.56
 91/06/19           38.17            5.00       1,456.95
 91/06/20           14.23            3.00         202.49
 91/06/21           44.72            7.00       1,999.88
 91/06/24          126.28           12.00      15,946.63
 91/06/25           47.74            8.00       2,279.11
 91/06/26           40.97            2.00       1,678.54
 91/06/27           60.24            9.00       3,628.85
 91/06/28           31.00            3.00         961.00
```

BY フィールドには列参照が適用されないため、最初の列参照は元のフォーマットの TRANSTOT、その後にフォーマットが再設定されたバージョンの TRANSTOT が続きます。その次は、元のフォーマットの QUANTITY 、さらにフォーマットが再設定されたバージョンの QUANTITY と続きます。最後は、一時項目 (COMPUTE) の PRODUCT になります。

FML 以外のリクエストで CNOTATION=PRINTONLY を使用した列表記

次の例では、CNOTATION=PRINTONLY が設定されています。これにより、列参照は、出力されるフィールドのみに割り当てられます。この場合、C1 と C2 の積 (PRODUCT) は、TRANSTOT と QUANTITY の積です。

```
SET CNOTATION = PRINTONLY

TABLE FILE VIDEOTRK
SUM TRANSTOT/D12.2 QUANTITY/D12.2
AND COMPUTE
PRODUCT = C1 * C2;
BY TRANSDATE
END
```

出力結果は次のとおりです。

```
TRANSDATE        TRANSTOT        QUANTITY         PRODUCT
---------        --------        --------         -------
 91/06/17           57.03           12.00          684.36
 91/06/18           21.25            2.00           42.50
 91/06/19           38.17            5.00          190.85
 91/06/20           14.23            3.00           42.69
 91/06/21           44.72            7.00          313.04
 91/06/24          126.28           12.00        1,515.36
 91/06/25           47.74            8.00          381.92
 91/06/26           40.97            2.00           81.94
 91/06/27           60.24            9.00          542.16
 91/06/28           31.00            3.00           93.00
```

FML リクエストで CNOTATION=PRINTONLY を使用した列番号

次のリクエストでは、フィールドのフォーマットを再設定することにより、新しい列が内部マトリックスに作成されます。2 つ目の RECAP 式では、CNOTATION の設定により次のようになります。

- TOTCASH(1) は、表示された列 1 の現金総額を参照します。

- TOTCASH(2) は、表示された列 2 の現金総額を参照します。
- 計算結果は、「CASH GROWTH(%)」というラベルが付いた行の列 2 に表示されます。
- RECAP 値は、指定した列に対してのみ計算されます。

```
SET CNOTATION=PRINTONLY
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END
TABLE FILE LEDGER
SUM CUR_YR/F9.2 AS 'CURRENT,YEAR'
LAST_YR/F9.2 AS 'LAST,YEAR'

FOR ACCOUNT
1010 AS 'CASH ON HAND'                                OVER
1020 AS 'DEMAND DEPOSITS'                             OVER
1030 AS 'TIME DEPOSITS'                               OVER
BAR                                                   OVER
RECAP TOTCASH/F9.2C= R1 + R2 + R3; AS 'TOTAL CASH' OVER
" "                                                   OVER
RECAP GROCASH(2)/F9.2C=100*TOTCASH(1)/TOTCASH(2) - 100;
AS 'CASH GROWTH(%)'
END
```

出力結果は次のとおりです。

	CURRENT YEAR	LAST YEAR
CASH ON HAND	8784.00	7216.00
DEMAND DEPOSITS	4494.00	3483.00
TIME DEPOSITS	7961.00	6499.00

	---------	---------
TOTAL CASH	21239.00	17198.00
CASH GROWTH (%)		23.50

FML リクエストで CNOTATION=PRINTONLY を使用した連続列の RECAP 計算

次の例では、ATOT に対する RECAP 計算は、リクエストで指定された表示列 2 および 3 に対してのみ実行されます。表示列 1 に対しては、計算は実行されません。

```
SET CNOTATION=PRINTONLY
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
NEXT_YR/I5C=1.13*CUR_YR + 222;
END
TABLE FILE LEDGER
SUM NEXT_YR/F9.2 CUR_YR/F9.2 LAST_YR/F9.2
FOR ACCOUNT
10$$ AS 'CASH'                       OVER
1100 AS 'ACCOUNTS RECEIVABLE'        OVER
1200 AS 'INVENTORY'                  OVER
BAR                                  OVER
RECAP ATOT(2,3)/I5C = R1 + R2 + R3;
AS 'ASSETS  ACTUAL'
END
```

出力結果は次のとおりです。

	NEXT_YR	CUR_YR	LAST_YR
CASH	25992.00	21239.00	17198.00
ACCOUNTS RECEIVABLE	21941.00	18829.00	15954.00
INVENTORY	31522.00	27307.00	23329.00
	--------	--------	--------
ASSETS ACTUAL		67,375	56,478

FML リクエストで CNOTATION=PRINTONLY を使用した相対列アドレス指定

次の例では、表示される列 1 および列 2 に対する現金の増減 (CHGCASH) を計算します。

```
SET CNOTATION=PRINTONLY
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
NEXT_YR/I5C=1.13*CUR_YR + 222;
END
TABLE FILE LEDGER
SUM NEXT_YR/F9.2 CUR_YR/F9.2 LAST_YR/F9.2
FOR ACCOUNT
10$$ AS 'TOTAL CASH' LABEL TOTCASH           OVER
" "                                          OVER
```

```
RECAP CHGCASH(1,2)/I5SC = TOTCASH(*) - TOTCASH(*+1); AS 'CHANGE IN CASH'
END
```

出力結果は次のとおりです。

	NEXT_YR	CUR_YR	LAST_YR
TOTAL CASH	25992.00	21239.00	17198.00
CHANGE IN CASH	4,752	4,044	

FML リクエストによる CNOTATION=PRINTONLY を使用した列表記

次のリクエストでは、2 つの RECAP 式を使用して、3 行目 (PROFIT) の 4 つの表示列 (1、2、3、4) で値の減算を行い、VARIANCE (EVAR と WVAR) の値を求めます。これらの値は、セル表記 (r、c) を使用して識別されます。

```
SET CNOTATION=PRINTONLY
TABLE FILE REGION
SUM E_ACTUAL/F9.2 E_BUDGET/F9.2 W_ACTUAL/F9.2 W_BUDGET/F9.2
FOR ACCOUNT
3000 AS 'SALES'                          OVER
3100 AS 'COST'                           OVER
BAR                                      OVER
RECAP PROFIT/I5C = R1 - R2;              OVER
" "                                      OVER
RECAP EVAR(1)/I5C = E(3,1) - E(3,2);
AS 'EAST  VARIANCE'                      OVER
```

```
RECAP WVAR(3)/I5C = E(3,3) - E(3,4);
AS 'WEST  VARIANCE'
END
```

出力結果は次のとおりです。

	E ACTUAL	E BUDGET	W ACTUAL	W BUDGET
SALES	6000.00	4934.00	7222.00	7056.00
COST	4650.00	3760.00	5697.00	5410.00
PROFIT	1,350	1,174	1,525	1,646
EAST VARIANCE	176			
WEST VARIANCE			-121	

NOPRINT、フィールドのフォーマット再設定、COMPUTE を使用した列表記

次のリクエストには、表示しないフィールド、フォーマットが再設定された複数のフィールド、3 つの一時項目 (COMPUTE) があります。SET CNOTATION=PRINTONLY の設定により、列参照が正しく出力されます。

```
SET CNOTATION = PRINTONLY
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
NEXT_YR/I5C=1.13*CUR_YR + 222;
END
TABLE FILE LEDGER
SUM NEXT_YR NOPRINT CUR_YR
COMPUTE AMT2/D6 = AMOUNT *2;
LAST_YR/D5   AMOUNT NEXT_YR
COMPUTE AMT3/D6  = AMOUNT*3;
COMPUTE AMT4/D6  = AMOUNT*4;
FOR ACCOUNT
10$$ AS 'CASH'                                  OVER
1100 AS 'ACCTS.REC.'                           OVER
1200 AS 'INVENTORY'                             OVER
BAR                                             OVER
RECAP ATOT/I8C = R1 + R2 + R3; AS 'TOTAL'       OVER
```

```
RECAP DIFF(2,10,2)/D8  = ATOT(*) - ATOT(*-1);
END
```

出力結果は次のとおりです。

	CUR_YR	AMT2	LAST_YR	AMOUNT	NEXT_YR	AMT3	AMT4
CASH	21,239	42,478	17,198	21,239	25,992	63,717	84,956
ACCTS. REC.	18,829	37,658	15,954	18,829	21,941	56,487	75,316
INVENTORY	27,307	54,614	23,329	27,307	31,522	81,921	109,228
TOTAL	67,375	134,750	56,481	67,375	79,455	202,125	269,500
DIFF		67,375		10,894		122,670	

FML リクエスト以外の NOPRINT への列表記の使用

次のリクエストは、TRANSTOT、QUANTITY、TRANSCODE を TRANSDATE 別に合計します。TRANSTOT には NOPRINT オプションが指定されているため、レポート出力には表示されません。このリクエストは、COMPUTE コマンドを使用して、次のフィールドも計算します。

- TTOT2 の値は TRANSTOT と同一で、レポート出力に表示されます。
- UNIT_COST1 は フィールド 1 をフィールド 2 で除算して取得されます。
- UNIT_COST2 はフィールド 1 を QUANTITY で除算して取得されます。

```
SET CNOTATION = ALL
TABLE FILE VIDEOTRK
SUM TRANSTOT/D7.2 NOPRINT QUANTITY/D7.2 TRANSCODE
  COMPUTE TTOT2/D7.2 = C1;
  COMPUTE UNIT_COST1/D7.2 = C1/C2;
  COMPUTE UNIT_COST2/D7.2 = C1/QUANTITY;
BY TRANSDATE
END
```

このリクエストにより CNOTATION=EXPLICIT は正しい出力結果を生成します。次の注釈は EXPLICIT 設定が必要なことを示しています。

CNOTATION=ALL により、内部マトリックスのすべてのフィールドに列番号が割り当てられます。このリクエストは、次の列参照を生成します。

- C1 - 元のフォーマットの TRANSTOT
- C2 - D7.2 フォーマットの TRANSTOT
- C3 - 元のフォーマットの QUANTITY
- C4 - D7.2 フォーマットの QUANTITY
- C5 - TRANSCODE

UNIT_COST1 は C1/C2 です。これらの列番号はともに TRANSTOT に割り当てられため、UNIT_COST1 は常に 1 です。UNIT_COST2 は C1 (TRANSTOT) を QUANTITY で除算して取得されます。出力結果は次のとおりです。

```
TRANSDATE  QUANTITY  TRANSCODE     TTOT2  UNIT_COST1  UNIT_COST2
---------  --------  ---------     -----  ----------  ----------
 91/06/17     12.00         10     57.03        1.00        4.75
 91/06/18      2.00          2     21.25        1.00       10.63
 91/06/19      5.00          4     38.17        1.00        7.63
 91/06/20      3.00          3     14.23        1.00        4.74
 91/06/21      7.00          6     44.72        1.00        6.39
 91/06/24     12.00          9    126.28        1.00       10.52
 91/06/25      8.00          7     47.74        1.00        5.97
 91/06/26      2.00          2     40.97        1.00       20.48
 91/06/27      9.00          7     60.24        1.00        6.69
 91/06/28      3.00          3     31.00        1.00       10.33
```

CNOTATION = PRINTONLY の場合、NOPRINT オプションが指定されている TRANSTOT フィールドには列番号は割り当てられません。レポート出力に表示されない元のフォーマットの QUANTITY には列番号は割り当てられません。フォーマットが再適用された QUANTITY フィールドは表示され、列番号が割り当てられます。したがって、リクエストは次の列参照を生成します。

- C1 - D7.2 フォーマットの QUANTITY
- C2 - TRANSCODE

UNIT_COST1 は C1/C2、QUANTITY/TRANSCODE です。UNIT_COST2 は C1 (QUANTITY) を QUANTITY で除算して取得されます。したがって、UNIT_COST2 は常に 1 です。出力結果は次のとおりです。

```
TRANSDATE  QUANTITY  TRANSCODE     TTOT2  UNIT_COST1  UNIT_COST2
---------  --------  ---------     -----  ----------  ----------
 91/06/17     12.00         10     12.00        1.20        1.00
 91/06/18      2.00          2      2.00        1.00        1.00
 91/06/19      5.00          4      5.00        1.25        1.00
```

```
 91/06/20      3.00          3      3.00        1.00        1.00
 91/06/21      7.00          6      7.00        1.17        1.00
 91/06/24     12.00          9     12.00        1.33        1.00
 91/06/25      8.00          7      8.00        1.14        1.00
 91/06/26      2.00          2      2.00        1.00        1.00
 91/06/27      9.00          7      9.00        1.29        1.00
 91/06/28      3.00          3      3.00        1.00        1.00
```

CNOTATION = EXPLICIT の場合、フォーマット再適用後の TRANSTOT フィールドはリクエスト内で明示的に参照されるため、表示されない場合も列番号が割り当てられます。ただし、元のフォーマットの TRANSTOT フィールドには列番号は割り当てられません。リクエスト内で明示的に参照されないため、元のフォーマットの QUANTITY には列番号は割り当てられません。フォーマット再適用後の QUANTITY フィールドには列番号が割り当てられます。したがって、リクエストは次の列参照を生成します。

- C1 - D7.2 フォーマットの TRANSTOT
- C2 - D7.2 フォーマットの QUANTITY
- C3 - TRANSCODE

UNIT_COST1 は C1/C2、TRANSCODE/QUANTITY です。UNIT_COST2 は C1 (TRANSTOT) を QUANTITY で除算して取得されます。したがって、UNIT_COST2 は常に UNIT_COST1 と同一です。出力結果は次のとおりです。

```
TRANSDATE  QUANTITY  TRANSCODE     TTOT2  UNIT_COST1  UNIT_COST2
---------  --------  ---------     -----  ----------  ----------
 91/06/17     12.00         10     57.03        4.75        4.75
 91/06/18      2.00          2     21.25       10.63       10.63
 91/06/19      5.00          4     38.17        7.63        7.63
 91/06/20      3.00          3     14.23        4.74        4.74
 91/06/21      7.00          6     44.72        6.39        6.39
 91/06/24     12.00          9    126.28       10.52       10.52
 91/06/25      8.00          7     47.74        5.97        5.97
 91/06/26      2.00          2     40.97       20.48       20.48
 91/06/27      9.00          7     60.24        6.69        6.69
 91/06/28      3.00          3     31.00       10.33       10.33
```

FML リクエストでのセル番号の使用

次のリクエストでは、CUR_YR に NOPRINT オプションが指定されています。CHGCASH RECAP 式は LAST_YR および NEXT_YR から CUR_YR を減算することが想定されます。

```
SET CNOTATION = ALL
DEFINE FILE LEDGER
CUR_YR/I7C = AMOUNT;
LAST_YR/I5C = .87*CUR_YR - 142;
NEXT_YR/I5C = 1.13*CUR_YR + 222;
END
TABLE FILE LEDGER
SUM CUR_YR/I5C NOPRINT LAST_YR NEXT_YR
FOR ACCOUNT
10$$ AS 'TOTAL CASH ' LABEL TOTCASH OVER
" " OVER
RECAP CHGCASH(1,3)/I5SC=(TOTCASH(*) - TOTCASH(1));
  AS 'CHANGE FROM CURRENT'
END
```

CNOTATION = ALL の場合は、C1 は元のフォーマットの CUR_YR フィールドを、C2 はフォーマットが再適用された値を、C3 は LAST_YR、C4 は NEXT_YR をそれぞれ参照します。列が追加され、RECAP は列 1 および 3 のみを参照するため、「NEXT_YR - CUR_YR」という演算は実行されません。出力結果は次のとおりです。

```
                     LAST_YR  NEXT_YR
                     -------  -------
TOTAL CASH            17,195   25,991

CHANGE FROM CURRENT   -4,044
```

CNOTATION = PRINTONLY の場合は、CUR_YR フィールドには列番号が割り当てられないことから、列 3 は存在しません。したがって、演算は実行されません。出力結果は次のとおりです。

```
                     LAST_YR  NEXT_YR
                     -------  -------
TOTAL CASH            17,195   25,991

CHANGE FROM CURRENT
```

CNOTATION = EXPLICIT の場合、フォーマットが再適用された CUR_YR フィールドは表示はされませんが、リクエスト内で明示的に参照されているため、C1 です。両方の演算は正しく実行されます。出力結果は次のとおりです。

```
                     LAST_YR  NEXT_YR
                     -------  -------
TOTAL CASH            17,195   25,991
```

```
CHANGE FROM CURRENT   -4,044    4,752
```

列番号使用上の注意

- BY フィールドには列番号は割り当てられません。
- ACROSS フィールドには列番号が割り当てられます。
- 一時項目 (COMPUTE)には列番号が割り当てられます。
- リクエストで生成される列の範囲外の列番号は、次の状況で許可され、値は 0 (ゼロ)として扱われます。
  - ACROSS 句の後で、COMPUTE コマンドで指定される。
  - FML RECAP コマンドのセル参照。

  これらの場合、構文が生成する列の数を事前に認識することはできません。その他のコンテキストで範囲外の列番号を使用すると、次のメッセージが表示されます。

  ```
  (FOC258) 演算式のフィールド名に誤りがあります :column
  ```

COMPUTE コマンドでの FORECAST の使用

FORECAST 機能は、RECAP コマンドで使用するバージョンがすでに実装されていました。ただし、RECAP での使用は、出力での FORECAST フィールドの配置およびソートフィールドの使用に制限があります。

FORECAST を COMPUTE コマンドで使用することで、これらの制限が解除され、マスターファイルに FORECAST 演算を配置することが可能になります。COMPUTE コマンドで使用する FORECAST の場合、演算タイプごとにそれぞれ独自の FORECAST 関数があります。

FORECAST による傾向値と予測値の計算

FORECAST 機能を使用して、数値データの傾向値を計算したり、データソースに保存された値の範囲外で値を予測したりすることができます。FORECAST は、レポートまたは GRAPH リクエストで使用可能です。

傾向値および予測値を特定するための計算には、次のものがあります。

- **単純移動平均 (FORECAST_MOVAVE)** フィールドから抽出する値の数を指定して、算術平均を計算します。詳細は、COMPUTE コマンドでの FORECAST の使用を参照してください。
- **指数移動平均** 前回計算した平均値と次のデータ点との間で加重平均を計算します。指数移動平均を使用する場合、次の 3 つの方法があります。
    - **単純指数平滑法 (FORECAST_EXPAVE)** 新しい値と古い値に適用する加重値を選択して平均を計算します。詳細は、COMPUTE コマンドでの FORECAST の使用を参照してください。
    - **二重指数平滑法 (FORECAST_DOUBLEXP)** 繰り返すことなく時間の変化とともに増減するデータの傾向を考慮する方法です。詳細は、COMPUTE コマンドでの FORECAST の使用を参照してください。
    - **三重指数平滑法 (FORECAST_SEASONAL)** 時間の変化とともに特定の間隔で繰り返すデータの傾向を考慮する方法です。詳細は、COMPUTE コマンドでの FORECAST の使用を参照してください。
- **線形回帰分析 (FORECAST_LINEAR)** すべてのデータ点に最適な直線の係数を導き出し、その一次方程式を使用して値を推定します。詳細は、COMPUTE コマンドでの FORECAST の使用を参照してください。

傾向値の計算に加えて値の予測を行う場合、FORECAST は、生成した傾向値を新しいデータ点として使用し、当初のデータ点の範囲外で同一の計算を継続します。線形回帰法では、計算した回帰方程式を使用して、傾向値と予測値を導き出します。

FORECAST は、指定されたデータに基づいて計算を行いますが、計算の使用およびその信頼性はユーザの判断に委ねられます。そのため、FORECAST による予測の信頼性については、予測の正確さを左右するさまざまな要因に基づき、ユーザの責任で判断します。

FORECAST の処理

COMPUTE コマンドに FORECAST 関数のいずれかを含めることにより、FORECAST プロセスを呼び出します。FORECAST は、すべての既存データ点に対して指定した演算を実行しま

す。その後、この計算を繰り返し実行し、要求された数の予測値を生成します。FORECAST では、計算に使用するフィールド、生成する予測値の数、既存のデータ点を表す行のレポート出力に入力フィールドの値を表示するか、一時項目 (COMPUTE) を表示するかを指定するパラメータが必要です。

FORECAST は、リクエストの最下位ソートフィールドで演算を実行します。これは、リクエストの最後の ACROSS フィールド、または、リクエストに ACROSS フィールドが含まれない場合は最後の BY フィールドのいずれかです。最上位のソートフィールドの値が変更されたところで、 FORECAST の演算は再実行されます。複数の表示コマンドが指定されたリクエストでは、FORECAST は、最後の ACROSS フィールドで演算を実行します (ACROSS フィールドが存在しない場合は、最後の BY フィールド)。FORCAST で ACROSS フィールドを使用する場合、表示コマンドは SUM または COUNT でなければなりません。

FORECAST の使用上の注意

- FORECAST で使用するソートフィールドは、数値または日付フォーマットでなければなりません。
- 単純移動平均および指数移動平均のいずれかの方法を使用する場合、有益な結果を得るためにはデータ値を等間隔にする必要があります。
- 列参照番号の使用は、FORECAST 式ではサポートされませんN。列参照番号は、この式以外では従来どおりサポートされます。FORECAST の値を生成する間に、レポート出力に表示されない追加のフィールドが生成されます。これらの追加フィールドの数および位置は、指定したリクエストにより異なります。
- ミッシング値により、予期できない結果や使用不可の結果が生成される場合があるため、FORECAST_LINEAR ではミッシング値を使用しないことをお勧めします。
- ESTRECORDS パラメータを使用して外部ソートによりソート作業領域の必要量を効率的に推定できる場合は、予測値の FORECAST で出力結果に新しいレコードが作成されることを考慮する必要があります。
- スタイルが設定されたレポートでは、スタイルシート属性の WHEN=FORECAST を使用して、特定の属性を FORECAST で予測した値に割り当てることができます。たとえば、予測値を赤で表示するには、TABLE リクエストで次の構文を使用します。

```
ON TABLE SET STYLE *
TYPE=DATA,COLUMN=MYFORECASTSORTFIELD,WHEN=FORECAST,COLOR=RED,$
ENDSTYLE
```

FORECAST の制限事項

FORECAST を使用する COMPUTE コマンドでは次のものはサポートされません。

- BY TOTAL コマンド
- MORE、MATCH、FOR、OVER 句

データ行と予測行の区別

レポート出力を分かりやすくするため、新しいフィールドを作成して、各行の FORECAST 値が予測値であるかどうかをそのフィールドに表すことができます。これを行うには、一時項目 (DEFINE) を定義して、その値に 0 (ゼロ) 以外の定数を割り当てます。データソースの実際のレコードを表すレポート出力の行には、0 (ゼロ) 以外の値が表示されます。予測値を表す行には 0 (ゼロ) が表示されます。また、このフィールドを HOLD ファイルに継承させることもできます。

データ行と予測行の区別

次の例では、データソースのデータ行に対しては、一時項目 (DEFINE) の DATA_ROW に「1」という値を割り当てます。また、予測行に対しては 0 (ゼロ) という値を割り当てます。PREDICT フィールドは、予測行に対しては YES、データを含む行に対しては NO として計算されます。また、スタイルシート属性の WHEN=FORECAST を使用して、FORTOT フィールドの予測値を赤色で表示することもできます。

```
DEFINE FILE VIDEOTRK
DATA_ROW/I1 = 1;
END
TABLE FILE VIDEOTRK
SUM TRANSTOT DATA_ROW
COMPUTE
PREDICT/A3 = IF DATA_ROW NE 0 THEN 'NO' ELSE 'YES' ;
FORTOT/D12.2=FORECAST_LINEAR(MODEL_DATA,TRANSTOT,1,3);
BY TRANSDATE
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
TYPE=DATA, COLUMN=FORTOT, WHEN=FORECAST, COLOR=RED,$
```

```
ENDSTYLE
END
```

下図は、出力結果を示しています。

TRANSDATE	TRANSTOT	DATA_ROW	PREDICT	FORTOT
06/17/91	57.03	10	NO	38.58
06/18/91	21.25	2	NO	40.32
06/19/91	38.17	4	NO	42.07
06/20/91	14.23	3	NO	43.81
06/21/91	44.72	6	NO	45.55
06/24/91	126.28	9	NO	50.78
06/25/91	47.74	7	NO	52.52
06/26/91	40.97	2	NO	54.26
06/27/91	60.24	7	NO	56.00
06/28/91	31.00	3	NO	57.74
06/29/91		0	YES	59.48
06/30/91		0	YES	61.23
07/01/91		0	YES	62.97

FORECAST による傾向値と予測値の計算

FORECAST 機能を使用して、数値データの傾向値を計算したり、データソースに保存された値の範囲外で値を予測したりすることができます。FORECAST は、レポートまたは GRAPH リクエストで使用可能です。

傾向値および予測値を特定するための計算には、次のものがあります。

- 単純移動平均 (MOVAVE)　フィールドから抽出する値の数を指定して、算術平均を計算します。
- **指数移動平均**　前回計算した平均値と次のデータ点との間で加重平均を計算します。指数移動平均を使用する場合、次の 3 つの方法があります。
    - **単純指数平滑法 (EXPAVE)**　新しい値と古い値に適用する加重値を選択して平均を計算します。
    - **二重指数平滑法 (DOUBLEXP)**　繰り返すことなく時間の変化とともに増減するデータの傾向を考慮する方法です。

    - **三重指数平滑法 (SEASONAL)** 時間の変化とともに特定の間隔で繰り返すデータの傾向を考慮する方法です。

- **線形回帰分析 (REGRESS)** すべてのデータ点に最適な直線の係数を導き出し、その一次方程式を使用して値を推定します。

傾向値の計算に加えて値の予測を行う場合、FORECAST は、生成した傾向値を新しいデータ点として使用し、当初のデータ点の範囲外で同一の計算を継続します。線形回帰法では、計算した回帰方程式を使用して、傾向値と予測値を導き出します。

FORECAST は、指定されたデータに基づいて計算を行いますが、計算の使用およびその信頼性はユーザの判断に委ねられます。そのため、FORECAST による予測は常に信頼できるとは限らず、予測の正確さはさまざまな要因により左右されます。

FORECAST の処理

RECAP コマンドに FORECAST を含めることにより、FORECAST プロセスを呼び出します。このコマンドでは、予測値の生成に必要な各種パラメータを指定します。これらのパラメータには、計算に使用するフィールド、使用する計算タイプ、生成する予測数などがあります。FORECAST の結果は RECAP フィールドに格納され、新しいフィールド (非再帰的) または FORECASTFORECAST の計算に使用した同一フィールド (再帰的) のいずれかとして使用することができます。

- 再帰的な FORECAST では、結果が格納された RECAP フィールドが、FORECAST の一時項目 (COMPUTE) の生成に使用するフィールドにもなります。この場合、元のフィールドが表示コマンドで参照されていても表示されず、RECAP フィールドには元のフィールド値およびその後に続けて FORECAST 構文で指定した予測値の数が格納されます。レポートには傾向値は表示されません。ただし、HOLDLIST の設定が PRINTONLY でない場合、元のフィールドは出力ファイルに保存されます。
- 非再帰的な FORECAST では、新しいフィールドに FORECAST の計算結果が格納されます。表示コマンドで参照されている場合、この新しいフィールドは元のフィールドとともにレポートに表示されます。指定された場合は、新しいフィールドに傾向値と予測値が格納されます。

FORECAST は、リクエストの最後の ACROSS フィールドで演算を実行します。リクエストに ACROSS フィールドが含まれていない場合は、最後の BY フィールドで演算を実行します。最上位のソートフィールドの値が変更されたところで、 FORECAST の演算は再実行されます。複数の表示コマンドが指定されたリクエストでは、FORECAST は、最後の ACROSS フィールドで演算を実行します (ACROSS フィールドが存在しない場合は、最後の BY フィールド)。FORCAST で ACROSS フィールドを使用する場合、表示コマンドは SUM または COUNT でなければなりません。

注意：FORECAST は括弧内の引数リストを使用してパラメータを渡しますが、関数にはなりません。同名の関数と共存させることができるのは、その関数が RECAP コマンドで指定されていない場合に限ります。

傾向値と予測値の計算

MOVAVE 計算

```
ON sortfield RECAP result_field[/fmt] = FORECAST(infield, interval,
 npredict, 'MOVAVE',npoint1)sendstyle
```

EXPAVE 計算

```
ON sortfield RECAP result_field[/fmt] = FORECAST(infield, interval,
 npredict, 'EXPAVE',npoint1);
```

DOUBLEXP 計算

```
ON sortfield RECAP fld1[/fmt] = FORECAST(infield,
interval, npredict, 'DOUBLEXP',npoint1, npoint2);
```

SEASONAL 計算

```
ON sortfield RECAP fld1[/fmt] = FORECAST(infield,
interval, npredict, 'SEASONAL', nperiod, npoint1, npoint2, npoint3);
```

REGRESS 計算

```
ON sortfield RECAP result_field[/fmt] = FORECAST(infield, interval,
 npredict, 'REGRESS');
```

説明

sortfield

リクエスト内の最後の ACROSS フィールドです。このフィールドは、数値または日付フォーマットでなければなりません。リクエストに ACROSS フィールドが指定されていない場合は、FORECAST は最後の BY フィールドで演算を実行します。

result_field

FORECAST の結果を格納するフィールドです。このフィールドは、新しいフィールドにすることも、infield と同一のフィールドにすることもできます。これは数値フィールドでなければなりません。実フィールド、一時項目 (DEFINE)、一時項目 (COMPUTE) のいずれかです。

注意：「FORECAST」という用語および開始括弧は、構文の sortfield= と同一の行に配置する必要があります。

fmt

result_field の表示フォーマットです。デフォルトフォーマットは D12.2 です。DEFINE または COMPUTE コマンドを使用して result_field のフォーマットを再設定した場合は、RECAP コマンドで指定したフォーマットが優先されます。

infield

任意の数値フィールドです。result_field と同一のフィールドにすることも、異なるフィールドにすることもできます。ただし、日付時間フィールドおよび日付表示オプションの数値フィールドにすることはできません。

interval

それぞれの sortfield 値に追加する増加値です。最後のデータ点の後に追加して次の値を作成します。この値は正の整数でなければなりません。降順にソートするには、BY HIGHEST 句を使用します。この数が sortfield の値に追加された場合、その結果は sortfield と同一のフォーマットに変換されます。

日付フィールドでは、フォーマットの最小単位の要素によりその数字の認識方法が決まります。たとえば、フォーマットが YMD、MDY、DMY のいずれかの場合、間隔値の 2 は 2 日と認識され、フォーマットが YM の場合は間隔値の 2 は 2 か月と認識されます。

npredict

計算する FORECAST の予測数です。この値は、0 (ゼロ) 以上の整数でなければなりません。この値の 0 (ゼロ) は予測値が必要ないことを表し、それは非再帰的の FORECAST でのみサポートされます。SEASONAL を使用した方法では、npredict は計算する periods の数になります。作成する points の数は次のとおりです。

```
nperiod * npredict
```

nperiod

SEASONAL を使用する場合、1 つの期間でのデータ点の数を指定する正の自然数です。

npoint1

MOVAVE を使用する場合、平均する値の数です。EXPAVE、DOUBLEXP、SEASONAL では、この数を使用して、平均の各要素に対する加重値を計算します。この値は正の自然数でなければなりません。次の計算式を使用して、加重値 k を計算します。

```
k=2/(1+npoint1)
```

npoint2

DOUBLEXP および SEASONAL では、この正の自然数を使用して、傾向の各項に対する加重値を計算します。次の計算式を使用して、加重値 g を計算します。

```
g=2/(1+npoint2)
```

npoint3

SEASONAL では、この正の自然数を使用して、季節調整の各項に対する加重値を計算します。次の計算式を使用して、加重値 p を計算します。

```
p=2/(1+npoint3)
```

FORECAST の使用上の注意

- FORECAST で使用するソートフィールドは、数値または日付フォーマットでなければなりません。
- 単純移動平均および指数移動平均のいずれかの方法を使用する場合、有益な結果を得るためにはデータ値を等間隔にする必要があります。
- FORECAST とともに RECAP コマンドを使用する場合、このコマンドには FORECAST 構文のみを含めることができます。FORECAST は、終了のセミコロン (;) の後に記述した構文を認識しません 。AS や IN などのオプションを指定する場合は、次の方法で行います。
    - 非再帰的な FORECAST リクエストでは、RECAP コマンドの前に空の COMPUTE コマンドを使用します。
    - 再帰的な FORECAST リクエストでは、リクエストで最初にフィールドを参照する際に、これらのオプションを指定します。

- FORECAST を含むリクエストでは、列表記の使用はサポートされません。FORECAST の値を生成する間に、レポート出力に表示されない追加のフィールドが生成されます。これらの追加フィールドの数および位置は、指定したリクエストにより異なります。
- リクエストには、FORECAST RECAP 以外のコマンドを最大で 7 つ、および追加の FORECAST RECAP コマンドを最大で 7 つ含めることができます。
- FORECAST に使用する RECAP コマンドの左側では、通貨フィールドを作成するための CURR 属性がサポートされます。
- 再帰的および非再帰的 REGRESS は、表示コマンドの SUM、ADD、WRITE のいずれかが使用されている場合、それと同一のリクエストではサポートされません。
- ミッシング値は、REGRESS ではサポートされません。
- ESTRECORDS パラメータを使用して外部ソートによりソート作業領域の必要量を効率的に推定できる場合は、予測値の FORECAST で出力結果に新しいレコードが作成されることを考慮する必要があります。
- スタイルが設定されたレポートでは、スタイルシート属性の WHEN=FORECAST を使用して、特定の属性を FORECAST で予測した値に割り当てることができます。たとえば、予測値を赤で表示するには、TABLE リクエストで次の構文を使用します。

```
ON TABLE SET STYLE
*TYPE=DATA,COLUMN=MYFORECASTSORTFIELD,WHEN=FORECAST,COLOR=RED,
$ENDSTYLE
```

FORECAST の制限事項

FORECAST を使用する RECAP コマンドでは次のものはサポートされません。

- BY TOTAL コマンド
- MORE、MATCH、FOR、OVER 句
- SUMMARIZE と RECOMPUTE は、FORECAST に使用するソートフィールドと同一のフィールドではサポートされません。
- MISSING 属性

単純移動平均の使用

単純移動平均は、フィールドから抽出する値の数を指定して計算した算術平均です。連続したデータ値から新しい平均を計算するには、前回の計算で使用した最初の値を除外し、次のデータ値を計算に追加します。

単純移動平均は、時間とともに変化する株価の傾向分析に使用することもあります。その場合、株価の期間数を指定して平均を計算します。この指標の難点は、先へ進むとともに一番古い値を計算から除外するため、時間とともに過去の記録を失うことです。さらに、この方法では各点に同一の加重が加えられるため、上下の極値により平均値に歪みが生じます。

データ値の範囲外で値を予測する場合は、計算した傾向値を新しいデータ点として扱う移動平均を使用してその予測値が計算されます。

最初の完全な移動平均は、計算に n 個の値が必要になることから、n 番目のデータ点で得られます。これは「ラグ (遅れ)」と呼ばれます。ラグの行に対する移動平均値は次のように計算されます。移動平均フィールドの最初の値は最初のデータ値と等しく、移動平均フィールドの 2 つ目の値は最初の 2 つのデータ値の平均値と等しくするというように、移動平均の計算に使用する値の数が指定した数に達する n 行目までこれを繰り返します。

新しい単純移動平均フィールドの計算

次のリクエストでは、「PERIOD」という名前の整数値を定義し、それを移動平均の独立変数として使用します。ここでは、取得したデータの範囲外に値が発生する 3 期間を予測します。

```
DEFINE FILE GGSALES
 SDATE/YYM = DATE;
 SYEAR/Y = SDATE;
 SMONTH/M = SDATE;
 PERIOD/I2 = SMONTH;
END
TABLE FILE GGSALES
  SUM UNITS DOLLARS
  BY  CATEGORY BY PERIOD
  WHERE SYEAR EQ 97 AND CATEGORY NE 'Gifts'
  ON PERIOD RECAP MOVAVE/D10.1= FORECAST(DOLLARS,1,3,'MOVAVE',3);
END
```

出力結果は次のとおりです。

Category	PERIOD	Unit Sales	Dollar Sales	MOVAVE
Coffee	1	61666	801123	801,123.0
	2	54870	682340	741,731.5
	3	61608	765078	749,513.7
	4	57050	691274	712,897.3
	5	59229	720444	725,598.7
	6	58466	742457	718,058.3
	7	60771	747253	736,718.0
	8	54633	655896	715,202.0
	9	57829	730317	711,155.3
	10	57012	724412	703,541.7
	11	51110	620264	691,664.3
	12	58981	762328	702,334.7
	13	0	0	694,975.6
	14	0	0	719,879.4
	15	0	0	705,729.9
Food	1	54394	672727	672,727.0
	2	54894	699073	685,900.0
	3	52713	642802	671,534.0
	4	58026	718514	686,796.3
	5	53289	660740	674,018.7
	6	58742	734705	704,653.0
	7	60127	760586	718,677.0
	8	55622	695235	730,175.3
	9	55787	683140	712,987.0
	10	57340	713768	697,381.0
	11	57459	710138	702,348.7
	12	57290	705315	709,740.3
	13	0	0	708,397.8
	14	0	0	707,817.7
	15	0	0	708,651.9

このレポートでは、平均の計算に使用した値の数は 3 で、生成された PERIOD の値に対して UNITS および DOLLARS の値は存在しません。

それぞれの平均 (MOVAVE の値) は、DOLLARS 値が存在する場合はその値を使用して計算されます。移動平均の計算は次のように開始されます。

- 最初の MOVAVE の値 (801,123.0) は、最初の DOLLARS の値に等しくなります。
- 2 つ目の MOVAVE の値 (741,731.5) は、最初の 2 つの DOLLARS 値の平均に等しくなります。その計算は、(801,123 + 682,340) /2 です。
- 3 つ目の MOVAVE の値 (749,513.7) は、最初の 3 つの DOLLARS 値の平均に等しくなります。その計算は、(801,123 + 682,340 + 765,078) / 3 です。
- 4 つ目の MOVAVE の値 (712,897.3) は、2 つ目から 4 つ目までの DOLLARS 値の平均に等しくなります。その計算は、(682,340 + 765,078 + 691,274) /3 です。

指定された値の範囲外で値を予測する場合は、計算された MOVAVE の値が新しいデータ点として使用され、移動平均が引き続き計算されます。MOVAVE の予測値 (PERIOD 13 の 694,975.6 から開始) は、前回の MOVAVE の値を新しいデータ点として使用して計算されます。たとえば、最初の予測値 (694,975.6) は、11 および 12 番目の期間のデータ点 (620,264 および 762,328) と 12 番目の期間の移動平均 (702,334.7) の平均になります。実際の計算は、(620,264 + 762,328 + 702,334.7)/3 = 694,975 のようになります。

単純移動平均フィールドとしての既存フィールドの使用

次のリクエストでは、「PERIOD」という名前の整数値を定義し、それを移動平均の独立変数として使用します。ここでは、取得したデータの範囲外に値が発生する 3 期間を予測します。RECAP フィールドには、FORECAST パラメータリストの最初の引数と同一の名前を使用します。傾向値はレポートに表示されません。レポートフィールドの予測値は、DOLLARS の実データ値の次に表示されます。

```
DEFINE FILE GGSALES
 SDATE/YYM = DATE;
 SYEAR/Y = SDATE;
 SMONTH/M = SDATE;
 PERIOD/I2 = SMONTH;
END
TABLE FILE GGSALES
  SUM UNITS DOLLARS
  BY  CATEGORY BY PERIOD
  WHERE SYEAR EQ 97 AND CATEGORY NE 'Gifts'
  ON PERIOD RECAP DOLLARS/D10.1 = FORECAST(DOLLARS,1,3,'MOVAVE',3);
END
```

出力結果は次のとおりです。

Category	PERIOD	Unit Sales	DOLLARS
Coffee	1	61666	801,123.0
	2	54870	682,340.0
	3	61608	765,078.0
	4	57050	691,274.0
	5	59229	720,444.0
	6	58466	742,457.0
	7	60771	747,253.0
	8	54633	655,896.0
	9	57829	730,317.0
	10	57012	724,412.0
	11	51110	620,264.0
	12	58981	762,328.0
	13	0	694,975.0
	14	0	719,879.0
	15	0	705,729.0
Food	1	54394	672,727.0
	2	54894	699,073.0
	3	52713	642,802.0
	4	58026	718,514.0
	5	53289	660,740.0
	6	58742	734,705.0
	7	60127	760,586.0
	8	55622	695,235.0
	9	55787	683,140.0
	10	57340	713,768.0
	11	57459	710,138.0
	12	57290	705,315.0
	13	0	708,397.0
	14	0	707,817.0
	15	0	708,651.0

単純指数平滑法の使用

単純指数平滑法では、新しい値と古い値に適用する加重値を選択して平均を計算します。

最新の値に割り当てる加重値は、次の計算式で求められます。

```
k = 2/(1+n)
```

説明

k

最新の値に割り当てる加重値です。

n

1 より大きい整数値です。n の値が増加するにつれて、これまでの観察値 (つまり、データインスタンス) に割り当てた加重値の方が最新のものよりも大きくなります。

次の計算式により、指数移動平均 (EMA) の値を得るための計算が導き出されます。

```
EMA = (EMA * (1-k)) + (datavalue * k)
```

この式は、データソースの最新の値に係数 k を乗算し、現在の移動平均に係数 (1-k) を乗算することを表しています。これらの数値を集計して、新しい EMA を算出します。

注意：データ値をすべて使用し終わった場合は、ソートグループの最後のデータ値が次のデータ値として使用されます。

単純指数平滑フィールドの計算

次の例では、「PERIOD」という名前の整数値を定義し、それを移動平均の独立変数として使用します。ここでは、取得したデータの範囲外に値が発生する 3 期間を予測します。

```
DEFINE FILE GGSALES
 SDATE/YYM = DATE;
 SYEAR/Y = SDATE;
 SMONTH/M = SDATE;
 PERIOD/I2 = SMONTH;
END
TABLE FILE GGSALES
  SUM UNITS DOLLARS
  BY  CATEGORY BY PERIOD
  WHERE SYEAR EQ 97 AND CATEGORY NE 'Gifts'
  ON PERIOD RECAP EXPAVE/D10.1= FORECAST(DOLLARS,1,3,'EXPAVE',3);
END
```

出力結果は次のとおりです。

Category	PERIOD	Unit Sales	Dollar Sales	EXPAVE
Coffee	1	61666	801123	801,123.0
	2	54870	682340	741,731.5
	3	61608	765078	753,404.8
	4	57050	691274	722,339.4
	5	59229	720444	721,391.7
	6	58466	742457	731,924.3
	7	60771	747253	739,588.7
	8	54633	655896	697,742.3
	9	57829	730317	714,029.7
	10	57012	724412	719,220.8
	11	51110	620264	669,742.4
	12	58981	762328	716,035.2
	13	0	0	716,035.2
	14	0	0	716,035.2
	15	0	0	716,035.2
Food	1	54394	672727	672,727.0
	2	54894	699073	685,900.0
	3	52713	642802	664,351.0
	4	58026	718514	691,432.5
	5	53289	660740	676,086.3
	6	58742	734705	705,395.6
	7	60127	760586	732,990.8
	8	55622	695235	714,112.9
	9	55787	683140	698,626.5
	10	57340	713768	706,197.2
	11	57459	710138	708,167.6
	12	57290	705315	706,741.3
	13	0	0	706,028.3
	14	0	0	705,671.6
	15	0	0	705,493.3

このレポートでは、CATEGORY 別に 3 つの EXPAVE の予想値が計算されます。データの範囲外の値に対して、前回の期間 (PERIOD) の値に間隔値 (1) が追加されて新しい期間 (PERIOD) の値が生成されます。

それぞれの平均 (EXPAVE の値) は、DOLLARS 値が存在する場合はその値を使用して計算されます。移動平均の計算は次のように開始されます。

- 最初の EXPAVE 値 (801,123.0) は、最初の DOLLARS 値と等しくなります。
- 2 つ目の EXPAVE 値 (741,731.5) は、次のように計算されます。なお、使用する小数の端数処理および小数点以下の桁数により、この例で得られた計算値がレポート出力に表示されるものと多少異なる場合があります。

```
n=3 (number used to calculate weights)
```

```
k = 2/(1+n) = 2/4 = 0.5
```

```
EXPAVE = (EXPAVE*(1-k))+(new-DOLLARS*k) = (801123*0.5) + (682340*0.50) =
400561.5 + 341170 = 741731.5
```

- 3 つ目の EXPAVE 値 (753,404.8) は、次のように計算されます。

```
EXPAVE = (EXPAVE*(1-k))+(new-DOLLARS*k) = (741731.5*0.5)+(765078*0.50) =
370865.75 + 382539 = 753404.75
```

指定された値の範囲外で値を予想する場合、最後の EXPAVE 値が指数平滑計算の新しいデータ点として使用されます。EXPAVE の予想値 (706,741.6 から開始) は、前回の平均と新しいデータ点を使用して計算されます。前回の平均は新しいデータ点としても使用されるため、予想値は常に最後の傾向値と等しくなります。たとえば、期間 13 に対する前回の平均は 706,741.6 ですが、この値を次のデータ点としても使用すると、平均の計算は (706,741.6 * 0.5) + (706,741.6 * 0.5) = 706,741.6 のようになります。

```
EXPAVE = (EXPAVE * (1-k)) + (new-DOLLARS * k) = (706741.6*0.5) +
         (706741.6*0.50) =  353370.8 + 353370.8 = 706741.6
```

二重指数平滑法の使用

二重指数平滑法は、繰り返すことなく時間の変化とともに増減するデータの傾向を考慮した指数移動平均です。この計算には、2 つの定数を用いた 2 つの方程式が使用されます。

- 最初の方程式は、現在の期間を考慮した式で、現在のデータ値と前回の平均の加重平均です。追加の要素 (b) は、前回期間の傾向値を表します。k は加重定数です。

```
DOUBLEXP(t) = k * datavalue(t) + (1-k) * ((DOUBLEXP(t-1) + b(t-1))
```

- 2 つ目の方程式は、傾向値を計算する式で、現在と前回の平均差と前期間の傾向値との加重平均です。b(t) は平均の傾向値を表します。g は加重定数です。g は加重定数です。

```
b(t) = g * (DOUBLEXP(t)-DOUBLEXP(t-1)) + (1 - g) * (b(t-1))
```

これら 2 つの方程式を解いて、平滑化した平均を求めます。平滑化した最初の平均が、最初のデータ値に設定されます。最初の傾向要素は 0 (ゼロ) に設定されます。一般的に、最善の

結果を得るには、2 つの定数を選定する際に、データ値と計算した平均値の間の平均平方誤差 (MSE) を最小にする定数を選択するようにします。最適な定数を得るために、非線形の最適化手法を使用する方がよい場合があります。

二重指数平滑法では、次の方程式を使用してデータ点の範囲外で値を予測します。

```
forecast(t+m) = DOUBLEXP(t) + m * b(t)
```

説明

m

予測を行う期間数です。

二重指数平滑フィールドの計算

次の例では、「PERIOD」という名前の整数値を定義し、それを移動平均の独立変数として使用します。二重指数平滑法は、単純指数平滑法に比べてデータ点における傾向値の推定に優れています。

```
SET SHOWBLANKS = OFF
TABLE FILE CENTSTMT
SUM ACTUAL_YTD
  BY PERIOD
  ON PERIOD RECAP EXP/D15.1 = FORECAST(ACTUAL_YTD,1,0,'EXPAVE',3);
  ON PERIOD RECAP DOUBLEXP/D15.1 = FORECAST(ACTUAL_YTD,1,0,
     'DOUBLEXP',3,3);
WHERE GL_ACCOUNT LIKE '3%%%'
END
```

出力結果は次のとおりです。

	YTD		
PERIOD	Actual	EXP	DOUBLEXP
2002/01	12,957,681.	12,957,681.0	12,957,681.0
2002/02	25,441,971.	19,199,826.0	22,439,246.3
2002/03	39,164,321.	29,182,073.5	34,791,885.1
2002/04	52,733,326.	40,957,699.8	48,845,816.0
2002/05	66,765,920.	53,861,809.9	63,860,955.9
2002/06	80,952,492.	67,407,150.9	79,188,052.9

三重指数平滑法の使用

三重指数平滑法は、時間とともに特定の間隔で繰り返される値の傾向を考慮した指数移動平均です。たとえば、現在の売上データが増加していることおよび毎年 12 月には常に 25 パーセントの売り上げが期待されることから、この売上データには傾向と季節性の両方の要因が重なっています。三重指数平滑法は、3 つの定数を用いた 3 つの方程式を使用して、傾向と季節性の両方の要因を考慮します。

三重指数平滑法では、各期間でのデータ点の数 (次の方程式の L) を知る必要があります。季節性を考慮するには、季節インデックスを計算します。データを前回の季節インデックスで除算し、そのデータを使用して平滑化した平均を計算します。

- 最初の方程式は、現在の期間を表した式で、現在のデータ値を季節係数で除算した値と前期間の傾向値で調整された前回平均との加重平均です。k は加重定数です。

  ```
  SEASONAL(t) = k * (datavalue(t)/I(t-L)) + (1-k) * (SEASONAL(t-1) +  b(t-1))
  ```

- 2 つ目の方程式は、傾向値を計算する式で、現在と前回の平均差と前期間の傾向値との加重平均です。b(t) は平均の傾向値を表します。g は加重定数です。g は加重定数です。

  ```
  b(t) = g * (SEASONAL(t)-SEASONAL(t-1)) + (1-g) * (b(t-1))
  ```

- 3 つ目の方程式は、季節インデックスを計算する式で、現在のデータを現在の平均で除算した値と前回の季節での季節インデックスとの加重平均です。I(t) は平均季節係数を表します。p は加重値の定数です。

  ```
  I(t) = p * (datavalue(t)/SEASONAL(t)) + (1 - p) * I(t-L)
  ```

これらの方程式を解いて、三重に平滑化された平均を求めます。平滑化した最初の平均が、最初のデータ値に設定されます。季節係数の初期値は、データソースに存在するデータの全期間の最大数に基づいて計算されます。それに対して、初期の傾向値は 2 期間のデータに基づいて計算されます。これらの値は次の手順で計算されます。

1. 次の計算式で初期の傾向係数を計算します。

   ```
   b(0) = (1/L) ((y(L+1)-y(1))/L + (y(L+2)-y(2))/L + ... + (y(2L) - y(L))/L )
   ```

2. 初期の季節係数は、各期間のデータ値の平均 A(j) (1<=j<=N) に基づいて計算します。

```
A(j) = ( y((j-1)L+1) + y((j-1)L+2) + ... + y(jL) ) / L
```

3. 次の計算式から初期の周期係数が得られます。ここで、N はデータの中で使用可能な全期間数、L は期間ごとのデータ点の数、n は期間内のデータ点 (1<= n <= L) を表します。

```
I(n) = ( y(n)/A(1) + y(L+n)/A(2) + ... + y((N-1)L+n)/A(N) ) / N
```

これら 3 つの定数は慎重に選定する必要があります。一般的に、最善の結果を得るには、データ値と計算した平均値の間の平均平方誤差 (MSE) を最小にする定数を選択するようにします。結果は、npoint1 と npoint2 の値の変化に影響され、値によっては精度の高い予測値を得ることがあります。精度の高い予測値を得るには、MSE を最小にする値を特定する必要があります。

三重指数平滑法を使用して最終のデータ点以降を予測するには、次の方程式を使用します。

```
forecast(t+m) = (SEASONAL(t) + m * b(t)) / I(t-L+MOD(m/L))
```

説明

m

予測する期間数です。

三重指数平滑フィールドの計算

次の例では、季節性は考慮するが、傾向値は考慮しないデータを取り扱います。そのため、npoint2 を高 (1000) に設定して、傾向係数を計算内で無視します。

```
SET SHOWBLANKS = OFF
TABLE FILE VIDEOTRK
SUM TRANSTOT
BY  TRANSDATE
ON TRANSDATE RECAP SEASONAL/D10.1 = FORECAST(TRANSTOT,1,3,'SEASONAL',
   3,3,1000,1);
WHERE TRANSDATE NE '19910617'
END
```

出力結果では、npredict は 3 です。つまり、3 期間分 (9 点 nperiod * npredict) が出力されます。

TRANSDATE	TRANSTOT	SEASONAL
91/06/18	21.25	21.3
91/06/19	38.17	31.0
91/06/20	14.23	34.6
91/06/21	44.72	53.2
91/06/24	126.28	75.3
91/06/25	47.74	82.7
91/06/26	40.97	73.7
91/06/27	60.24	62.9
91/06/28	31.00	66.3
91/06/29		45.7
91/06/30		94.1
91/07/01		53.4
91/07/02		72.3
91/07/03		140.0
91/07/04		75.8
91/07/05		98.9
91/07/06		185.8
91/07/07		98.2

線形回帰式の使用

線形回帰式では、従属変数 (新しく計算した値) と独立変数 (ソートフィールド値) が直線で表される関数で関係が成り立つと仮定して値を予測します。

```
y = mx + b
```

説明

y

従属変数です。

x

独立変数です。

m

直線の傾きです。

b

y 切片です。

REGRESS は「最小二乗推定法」と呼ばれる方法を使用して m および b の値を計算し、結果として得られた直線とデータ間の平方差の合計を最小にします。

m および b は、次の計算式で計算されます。

$$m = \frac{(\sum xy - (\sum x \bullet \sum y)/n)}{(\sum x^2 - (\sum x)^2/n)}$$

$$b = (\sum y)/n - (m \bullet (\sum x)/n)$$

説明

n

データ点の数です。

y

データ値 (従属変数) です。

x

ソートフィールド値 (独立変数) です。

傾向値および予測値は、回帰直線の方程式を使用して計算します。

新しい線形回帰フィールドの計算

```
TABLE FILE CAR
PRINT MPG
BY DEALER_COST
WHERE MPG NE 0.0
  ON DEALER_COST RECAP FORMPG=FORECAST(MPG,1000,3,'REGRESS');
END
```

出力結果は次のとおりです。

```
DEALER_COSTMPGFORMPG
      2,886       27           25.51
      4,292       25           23.65
      4,631       21           23.20
      4,915       21           22.82
      5,063       23           22.63
      5,660       21           21.83
                  21           21.83
      5,800       24           21.65
      6,000       24           21.38
      7,427       16           19.49
      8,300       18           18.33
      8,400       18           18.20
     10,000       18           16.08
     11,000       18           14.75
     11,194        9           14.50
     14,940       11            9.53
     15,940        0            8.21
     16,940        0            6.88
     17,940        0            5.55
```

注意

- FORMPG の 3 つの予測値が計算されます。データの範囲外の値では、前回の DEALER_COST の値に間隔値 (1,000) が追加されて新しい DEALER_COST の値が生成されます。
- 生成された DEALER_COST の値に対する MPG の値は存在しません。
- FORMPG のそれぞれの値は、MPG のすべての実データ値から計算された回帰直線を使用して算出されます。

  DEALER_COST は独立変数 (x) で、MPG は従属変数 (y) です。この方程式を使用して、MPGFORECAST の傾向値と予測値が計算されます。

  この場合、方程式はおおよそ次のようになります。

  ```
  FORMPG = (-0.001323 * DEALER_COST) + 29.32
  ```

  予測値は次のとおりです。ただし、これらの値は、端数処理のために FORECAST で計算した値と必ずしも一致しませんが、その計算式は下のとおりです。

DEALER_COST	計算式	FORMPG
15,940	(-0.001323 * 15,940) + 29.32	8.23

DEALER_COST	計算式	FORMPG
16,940	(-0.001323 * 16,940) + 29.32	6.91
17,940	(-0.001323 * 17,940) + 29.32	5.59

FORECAST によるレポート作成

FORECAST は、1 つのレポートで何回でも使用することができます。ただし、すべての FORECAST リクエストで、指定するソートフィールド、間隔、予測数は一致していなければなりません。変更できるものは、RECAP フィールド、計算方法、 値の計算に使用するフィールド、平均の計算に使用するデータ点の数に限られます。他のパラメータを変更した場合、変更後のパラメータは無視されます。

レポート出力で FORECAST フィールドの位置を変更するには、プレースホルダとして FORECAST フィールドに空の COMPUTE コマンドを使用します。データタイプ (I、F、P、D) は、COMPUTE コマンドおよび RECAP コマンドで一致していなければなりません。

レポート出力を分かりやすくするため、新しいフィールドを作成して、各行の FORECAST 値が予測値であるかどうかをそのフィールドに表すことができます。これを行うには、一時項目 (DEFINE) を定義して、その値に 0 (ゼロ) 以外の定数を割り当てます。データソースの実際のレコードを表示する際に、この定数がレポート出力の各行に表示されます。予測値を表す行には 0 (ゼロ) が表示されます。また、このフィールドを HOLD ファイルに継承させることもできます。

複数 FORECAST フィールドの生成

次の例では、GGSALES データソースの DOLLARS フィールドと BUDDOLLARS フィールドの両方に対して、移動平均および指数平均を計算します。ソートフィールド、間隔、予測数はすべての計算で共通です。

```
DEFINE FILE GGSALES
 SDATE/YYM = DATE;
 SYEAR/Y = SDATE;
 SMONTH/M = SDATE;
 PERIOD/I2 = SMONTH;
END
TABLE FILE GGSALES
```

```
  SUM DOLLARS AS 'DOLLARS' BUDDOLLARS AS 'BUDGET'
  BY CATEGORY NOPRINT BY PERIOD AS 'PER'
  WHERE SYEAR EQ 97 AND CATEGORY EQ 'Coffee'
  ON PERIOD RECAP DOLMOVAVE/D10.1= FORECAST(DOLLARS,1,0,'MOVAVE',3);
  ON PERIOD RECAP DOLEXPAVE/D10.1= FORECAST(DOLLARS,1,0,'EXPAVE',4);
  ON PERIOD RECAP BUDMOVAVE/D10.1 = FORECAST(BUDDOLLARS,1,0,'MOVAVE',3);
  ON PERIOD RECAP BUDEXPAVE/D10.1 = FORECAST(BUDDOLLARS,1,0,'EXPAVE',4);
END
```

出力結果は次のとおりです。

PER	DOLLARS	BUDGET	DOLMOVAVE	DOLEXPAVE	BUDMOVAVE	BUDEXPAVE
1	801123	801375	801,123.0	801,123.0	801,375.0	801,375.0
2	682340	725117	741,731.5	753,609.8	763,246.0	770,871.8
3	765078	810367	749,513.7	758,197.1	778,953.0	786,669.9
4	691274	717688	712,897.3	731,427.8	751,057.3	759,077.1
5	720444	739999	725,598.7	727,034.3	756,018.0	751,445.9
6	742457	742586	718,058.3	733,203.4	733,424.3	747,901.9
7	747253	773136	736,718.0	738,823.2	751,907.0	757,995.6
8	655896	685170	715,202.0	705,652.3	733,630.7	728,865.3
9	730317	753760	711,155.3	715,518.2	737,355.3	738,823.2
10	724412	709397	703,541.7	719,075.7	716,109.0	727,052.7
11	620264	630452	691,664.3	679,551.0	697,869.7	688,412.4
12	762328	718837	702,334.7	712,661.8	686,228.7	700,582.3

FORECAST フィールドの移動

次の例では、MOVAVE フィールドに対してブランクの COMPUTE コマンドをプレースホルダとして使用し、DOLLARS フィールドを MOVAVE フィールドの後に配置します。COMPUTE コマンドおよび RECAP コマンドの両方で (同一データタイプの) MOVAVE に対するフォーマットが指定されますが、RECAP コマンドのフォーマットが優先されます。

```
DEFINE FILE GGSALES
 SDATE/YYM = DATE;
 SYEAR/Y = SDATE;
 SMONTH/M = SDATE;
 PERIOD/I2 = SMONTH;
END
TABLE FILE GGSALES
SUM   UNITS
COMPUTE MOVAVE/D10.2 = ;
DOLLARS
  BY CATEGORY BY PERIOD
  WHERE SYEAR EQ 97 AND CATEGORY EQ 'Coffee'
  ON PERIOD RECAP MOVAVE/D10.1= FORECAST(DOLLARS,1,3,'MOVAVE',3);
END
```

出力結果は次のとおりです。

```
CategoryPERIODUnit SalesMOVAVEDollar Sales
Coffee             1       61666     801,123.0        801123
                   2       54870     741,731.5        682340
                   3       61608     749,513.7        765078
                   4       57050     712,897.3        691274
                   5       59229     725,598.7        720444
                   6       58466     718,058.3        742457
                   7       60771     736,718.0        747253
                   8       54633     715,202.0        655896
                   9       57829     711,155.3        730317
                  10       57012     703,541.7        724412
                  11       51110     691,664.3        620264
                  12       58981     702,334.7        762328
                  13           0     694,975.6             0
                  14           0     719,879.4             0
                  15           0     705,729.9             0
```

データ行と予測行の区別

次の例では、データソースのデータ行に対しては、一時項目 (DEFINE) の DATA_ROW に「1」という値を割り当てます。また、予測行に対しては 0 (ゼロ) という値を割り当てます。PREDICT フィールドは、予測行に対しては YES、データを含む行に対しては NO として計算されます。

```
DEFINE FILE CAR
DATA_ROW/I1 = 1;
END
TABLE FILE CAR
  PRINT DATA_ROW
COMPUTE PREDICT/A3 = IF DATA_ROW EQ 1 THEN 'NO' ELSE 'YES' ;
MPG
BY DEALER_COST
WHERE MPG GE 20
  ON DEALER_COST RECAP FORMPG/D12.2=FORECAST(MPG,1000,3,'REGRESS');
  ON DEALER_COST RECAP MPG         =FORECAST(MPG,1000,3,'REGRESS');
END
```

出力結果は次のとおりです。

```
DEALER_COSTDATA_ROWPREDICTMPGFORMPG
      2,886         1  NO                27.00          25.65
      4,292         1  NO                25.00          23.91
      4,631         1  NO                21.00          23.49
      4,915         1  NO                21.00          23.14
      5,063         1  NO                23.00          22.95
      5,660         1  NO                21.00          22.21
                    1  NO                21.00          22.21
      5,800         1  NO                24.20          22.04
      6,000         1  NO                24.20          21.79
      7,000         0  YES               20.56          20.56
      8,000         0  YES               19.32          19.32
      9,000         0  YES               18.08          18.08
```

重回帰 REGRESS による傾向値と予測値の計算

REGRESS による方法は、一連の数値データ点に最適な一次方程式を導き出し、その方程式を使用して、レポート出力に新しいフィールドを作成します。方程式は、1 つから 3 つまでの独立変数を使用して作成することができます。

この方法では、従属変数 (y、新しい一時項目 (COMPUTE)) と独立変数 (x1、x2、x3) が次のような一次方程式で関係が成り立つと仮定して値を予測します。

```
y = a1*x1 [+ a2*x2 [+ a3*x3]] + b
```

独立変数が 1 つの場合は、この方程式は直線を表します。その場合は、REGRESS を使用した FORECAST の場合と同一の値が生成されます。この方程式は、独立変数が 2 つの場合は面を表し、独立変数が 3 つの場合は超平面を表します。独立変数を線形に組み合わせることにより、従属変数の近似値を求めることができると考えられる場合は、その方法を使用するようにします。

REGRESS は「最小二乗推定法」と呼ばれる手法を使用して係数 (a1、a2、a3、a4) の値を計算し、結果として生成された行、面、超平面とデータ間の差の二乗和を最小にします。

線形重回帰フィールドの作成

```
ON {sortfield} RECAP y[/fmt] = REGRESS(n, x1, [x2, [x3,]] z);
```

説明

sortfield

データソースのフィールドです。REGRESS に使用するパラメータのいずれかと同一のフィールドにすることはできません。ソートフィールドの値が変更されるたびに、新しい線形回帰式が導き出されます。

y

回帰式を適用して計算する新しい数値フィールドです。この名前のフィールドに対して、DEFINE または COMPUTE コマンドを使用することはできません。

fmt

y の表示フォーマットです。省略する場合、デフォルトフォーマットは D12.2 です。

n

独立変数の個数を表す 1 から 3 までの自然数です。

x1, x2, x3

独立変数として使用するフィールド名です。これらの変数はすべて数値であり、それぞれが独立していなければなりません。

z

既存の数値フィールドで、独立変数に線形的にほぼ従属し、回帰式を導き出すために使用します。

REGRESS 使用上の注意

- REGRESS とともに使用するBY フィールドは、数値または日付フォーマットでなければなりません。
- REGRESS は、ACROSS フィールドで実行することはできません。

- 独立変数のいずれかがソートフィールドを兼ねている場合は、REGRESS ソートフィールドより前に、そのフィールドをリクエストで参照することはできません。
- 1 つのリクエストに FORECAST と REGRESS をともに使用することはできません。また、1 つのリクエストでサポートされる REGRESS は 1 つのみです。REGRESS RECAP 以外のコマンドはサポートされます。
- REGRESS とともに使用する RECAP コマンドには、REGRESS 構文のみを含めることができます。REGRESS は、終了のセミコロン (;) の後に続くすべての構文は認識しません。
- 括弧内の引数リストを使用してパラメータを REGRESS に渡した場合でも、REGRESS は関数にはなりません。ユーザが記述した同一名のサブルーチンと共存させることができるのは、そのサブルーチンが RECAP コマンドで指定されていない場合に限ります。
- BY TOTAL はサポートされません。
- MORE、MATCH、FOR、OVER はサポートされません。
- REGRESS の値が生成される過程で、レポート出力に表示されない新しいフィールドが作成されます。これらの追加フィールドの数および位置は、指定したリクエストにより異なります。そのため、列表記の使用は、REGRESS を含むリクエストではサポートされません。
- SUMMARIZE および RECOMPUTE は、REGRESS で使用するフィールドと同一のソートフィールドではサポートされません。
- REGRESS は、FOCUS GRAPH 機能ではサポートされません。
- REGRESS に使用する RECAP コマンドの左側では、通貨フィールドを作成するための CURR 属性がサポートされます。
- 回帰分析では、ミッシング値が存在するフィールドを使用することはできません。
- データ量が多いほど、有益な結果が得られます。

重回帰フィールドの作成

次のリクエストでは、GGSALES データソースを使用して、予測した DOLLARS フィールドを計算します。BUDUNITS、UNITS、BUDDOLLARS の各フィールドは独立変数です。実際の値が DOLLARS フィールドから入力されて予測計算に使用されます。

```
DEFINE FILE GGSALES
 YEAR/Y = DATE;
 MONTH/M = DATE;
 PERIOD/I2 = MONTH;
END
```

```
TABLE FILE GGSALES
PRINT BUDUNITS UNITS BUDDOLLARS DOLLARS
BY PERIOD
ON PERIOD
RECAP EST_DOLLARS/F8 = REGRESS(3, BUDUNITS, UNITS, BUDDOLLARS, DOLLARS);
WHERE CATEGORY EQ 'Coffee'
WHERE REGION EQ 'West'
WHERE UNITS GT 1600 AND UNITS LT 1700
END
```

出力結果は次のとおりです。

PERIOD	Budget Units	Unit Sales	Budget Dollars	Dollar Sales	EST_DOLLARS
1	1665	1678	21645	23492	0
	1725	1669	22425	21697	0
2	1613	1685	22582	18535	19065
	1568	1682	23520	25230	19089
	1847	1668	18470	25020	19725
3	1646	1656	23044	19872	19871
	1759	1615	17590	17765	17765
	1498	1637	16478	21281	21280
	1653	1694	21489	16940	16940
4	1457	1671	21855	20052	0
5	1662	1674	24930	18414	0
6	1825	1695	23725	25425	32228
	1870	1620	24310	24300	30866
	1712	1640	22256	16400	31859
7	1727	1623	24178	17853	-26821
	1733	1647	17330	24705	-13637
8	1830	1652	20130	23128	51170
	1451	1660	17412	19920	15755
	1556	1643	18672	18073	20517
9	1464	1663	14640	23282	18249
	1463	1663	21945	19956	23005
10	1464	1667	17568	25005	25005
	1711	1623	20532	22722	22722
	1701	1626	18711	21138	21138
	1473	1616	14730	16160	16160
11	1403	1601	21045	17611	0
12	1796	1696	17960	25440	0

DEFINE および COMPUTE でのテキストフィールドの使用

テキストフィールドを文字フィールドに割り当てることで、文字フィールドに割り当てられた値をテキストフィールドに渡すことができます。テキストフィールドの値を文字フィールドに割り当てる際に、その値が文字フィールドの長さを超える場合は、割り当て前に値の末尾が切り捨てられます。

文字式結果のテキストフィールドへの割り当て

次の例では、テキストフィールドが格納されている COURSE データソースを使用して、「ADESC」という文字フィールドを作成します。このフィールドでは、36 バイトを超えるテキストが切り捨てられます。また、ADESC 文字フィールドと同一長さの「NEWDESC」という新しいテキストフィールドを作成します。

```
DEFINE FILE COURSES
ADESC/A36   = DESCRIPTION;
NEWDESC/TX36 = ADESC;
END

TABLE FILE COURSES
PRINT ADESC NEWDESC
END
```

出力結果は次のとおりです。

```
ADESC                                NEWDESC
-----                                -------
This course provides the DP professi  This course provides the DP professi
Anyone responsible for designing FOC  Anyone responsible for designing FOC
This is a course in FOCUS efficienci  This is a course in FOCUS efficienci
```

マスターファイルから独立した一時項目の作成

DEFINE および COMPUTE コマンドで作成する一時項目は、特定のマスターファイルに関連付けられます。また、COMPUTE コマンドで計算した値は、特定のリクエストに関連付けられます。ただし、DEFINE FUNCTION コマンドを使用すると、マスターファイル、リクエストのどちらからも独立した一時項目を作成することができます。

DEFINE 関数は、複数の計算で構成された名前付きのグループで、任意の数の入力値を使用して値を返します。DEFINE 関数を呼び出す場合は、最初にその関数を定義しておく必要があります。

DEFINE 関数は、製品に付属の関数を使用できる状況のほとんどで呼び出すことができます。データタイプは、それぞれの引数と併せて定義します。引数の値を変更する場合は、そのフォーマットを定義済みのフォーマットと一致させる必要があります。文字の引数が指定

フォーマットより短い場合は、ブランクが追加されます。また、文字の引数が長い場合は、末尾が省略されます。

関数内のすべての計算は、倍精度で実行されます。フォーマットを変換するには、一時項目を定義する割り当て内で等号 (=) を使用します。

関数の定義

```
DEFINE FUNCTION name (argument1/format1,..., argumentn/formatn)
[tempvariablea/formata [TITLE 'line1[,line2 ...']
 [DESCRiption 'description'] = expressiona;]
   .
   .
   .
[tempvariablex/formatx = expressionx;]
name/format = [result_expression];
END
```

説明

name

関数名です。64 バイト以内で指定します。この名前は、この関数で計算する最後のフィールドにする必要があります。これを使用して、呼び出しプロシジャに関数の値を返します。

argument1...argumentn

引数の名前です。これらの名前は、WebFOCUS フィールドの名前規則に準拠する任意の名前にすることができます。

format1...formatn

関数に使用する引数のフォーマットです。

引数のフォーマットが文字の場合は、引数の値も文字でなければなりません。引数が短い場合は、右側にブランクが追加されます。引数が長い場合は、末尾が省略されます。

引数のフォーマットが数値の場合、引数の値も数値でなければなりません。期待どおりの結果を得るには、使用するデータタイプの整合を図る必要があります。

tempvariablea...tempvariablex

一時項目です。一時項目には、関数で使用する中間値が保管されます。一時項目は、必要な数だけ定義することができます。

tempformata...tempformatx

一時項目のフォーマットです。

line1,line2 ...

AS 句で上書きする場合を除き、一時項目 (DEFINE) のフィールドタイトルとして表示する行です。

description

一時項目 (DEFINE) に関連付ける説明です。一重引用符で囲みます。この説明は、マスターファイル参照ツールに表示されます。

expressiona...expressionx

一時項目の値を計算する式です。式に使用できるものには、パラメータ、定数、同一の関数で定義された他の一時項目があります。

format

関数が返す値のフォーマットです。

result_expression

関数が返す値を計算する式です。式に使用できるものには、パラメータ、定数、同一の関数で定義された他の一時項目があります。

関数内で定義したすべての名前は、その関数でのみ有効になります。関数の定義において END コマンドより前に定義した最後のフィールドは、その関数と同一の名前でなければなりません。このフィールドがこの関数の戻り値を表します。

DEFINE 関数の制限事項

- 1 つのセッションで定義および使用できる関数の数には概ね制限はありません。
- DEFINE 関数は、DEFINE FUNCTION CLEAR コマンドでクリアします。JOIN コマンドまたは WebFOCUS コマンドを発行してもクリアされません。
- クリアした関数を式で使用しようとすると、エラーが表示されます。

- DEFINE 関数が他の DEFINE 関数を呼び出すことはできますが、それ自体を呼び出すことはできません。
- DEFINE 関数を上書きまたはクリアした後、その関数を参照する一時項目を使用しようとすると、次の警告が表示されます。

```
(FOC03665) 外部関数 '%1' のロードエラーが発生しました。
```

関数の定義

次の例では、SUBTRACT 関数を作成して、その関数を呼び出します。SUBTRACT は、引数の VAL1 と VAL2 を使用して計算を実行します。

```
DEFINE FUNCTION SUBTRACT (VAL1/D8, VAL2/D8)
 SUBTRACT/D8.2 = VAL1 - VAL2;
END
```

```
TABLE FILE MOVIES
 PRINT TITLE LISTPR IN 35 WHOLESALEPR AND
 COMPUTE PROFIT/D8.2 = SUBTRACT(LISTPR,WHOLESALEPR);
 BY CATEGORY
   WHERE CATEGORY EQ 'MYSTERY' OR 'ACTION'
END
```

出力結果は次のとおりです。

CATEGORY	TITLE	LISTPR	WHOLESALEPR	PROFIT
action	JAWS	19.95	10.99	8.96
	ROBOCOP	19.98	11.50	8.48
	TOTAL	19.99	11.99	8.00

	RECALL			
	TOP GUN	14.95	9.99	4.96
	RAMBO III	19.95	10.99	8.96
MYSTERY	REAR WINDOW	19.98	9.00	10.98
	VERTIGO	19.98	9.00	10.98
	FATAL ATTRACTION	29.98	15.99	13.99
	NORTH BY NORTHWEST	19.98	9.00	10.98
	DEAD RINGERS	25.99	15.99	10.00
	MORNING AFTER, THE	19.95	9.99	9.96
	PSYCHO	19.98	9.00	10.98
	BIRDS, THE	19.98	9.00	10.98
	SEA OF LOVE	59.99	30.00	29.99

DEFINE 関数を表示するには

コマンドコンソールから次のコマンドを発行します。

```
? function
```

定義済み関数の表示

次のコマンドを発行します。

```
? function
```

このコマンドにより、次のようなデータが表示されます。

FUNCTIONS	CURRENTLY	ACTIVE	
name	FORMAT	parameter	FORMAT
----------	---------	---------	-------
SUBTRACT	D8.2	VAL1	D8
		VAL2	D8

関数を定義せずに ? FUNCTION コマンドを発行すると、次のように表示されます。

```
現在有効な関数はありません
```

DEFINE 関数のクリア

```
DEFINE FUNCTION {name|*} CLEAR
```

説明

name

クリアする関数名です。

*****

有効な DEFINE 関数をすべてクリアします。

合計と中間合計の追加

レポートの詳細情報を要約するために、行合計、列合計、総合計、中間合計を使用してデータを集計することができます。これらの集計行をレポートに使用して、レポートの情報を明確にしたり、強調したりすることができます。

行合計と列合計の計算

数値行または数値列を合計し、レポートに出力することができ、次の用途があります。

- ROW-TOTAL を使用して、同一行のすべての数値を合計し、合計した値を新しい列に表示する。
- COLUMN-TOTAL を使用して、同一列のすべての数値を合計し、合計した値をレポートの最終行に表示する。

行合計と列合計をマトリックスレポート (リクエストで BY と ACROSS の両方を使用して作成) に使用したり、行合計と列合計のタイトル名を変更したり、行合計と列合計に一時項目 (COMPUTE) を追加したりすることができます。また、ACROSS-TOTAL を使用して行合計を生成することもできます。

なお、レポートに合計を出力する際にフィールドの集計を実行すると、行合計のフォーマットはそのフィールドと同一のフォーマットになります。たとえば、CURR_SAL フィールドのフォーマットが D12.2M の場合、CURR_SAL の行合計のフォーマットも D12.2M になります。異なるフォーマットのフィールドを集計する場合は、デフォルトフォーマットの D12.2 が合計に適用されます。

AS 句を使用して、デフォルトのフィールドタイトルの名前を変更したり、行合計と列合計のラベルの位置を揃えたりすることができます。詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。

行合計と列合計の計算

```
display_command fieldname AND ROW-TOTAL [alignment][/format] [AS 'name']
display_command fieldname AND COLUMN-TOTAL [alignment][AS 'name']
```

説明

display_command

PRINT、LIST、SUM、COUNT のいずれかのコマンドです。

fieldname

行合計または列合計を計算するフィールドの名前です。

alignment

ROW-TOTAL または COLUMN-TOTAL のラベルの位置揃えを指定します。利用可能な値には、次のものがあります。

/R - ラベルを右揃えします。

/L - ラベルを左揃えします。

/C - ラベルを中央揃えします。

この位置揃えの設定は、HTML 出力では無視されます。WebFOCUS でこの位置揃え機能を活用するには、SET STYLE=OFF コマンドをリクエストに含めるか、レポートを PDF フォーマットまたはこの機能をサポートする別のフォーマットを使用します。

format

ROW-TOTAL のフォーマットを再設定します。

name

ROW-TOTAL または COLUMN-TOTAL のラベルです。

ON TABLE コマンドを使用して、行合計または列合計を指定することもできます。この場合、COLUMN-TOTAL ではオプションとしてフィールド名を指定することができますが、ROW-TOTAL ではフィールドを指定することはできません。使用する構文は次のとおりです。

```
ON TABLE COLUMN-TOTAL [alignment][AS 'name'][field field field]
ON TABLE ROW-TOTAL [alignment][/format] [AS 'name']
```

行合計と列合計の計算

次のリクエストでは、ROW-TOTAL および COLUMN-TOTAL を使用します。デフォルト設定により、行合計のラベルは「合計」、列合計のラベルは「縦計」になります。これを変更するには、AS 句を使用します。

```
TABLE FILE SALES
SUM RETURNS DAMAGED AND ROW-TOTAL AND COLUMN-TOTAL
BY PROD_CODE
END
```

出力結果は次のとおりです。

```
PROD_CODE  RETURNS  DAMAGED      TOTAL
---------  -------  -------  ---------
B10             13       10         23
B12              4        3          7
B17              4        2          6
B20              1        2          3
C13              3        0          3
C17              0        0          0
C7               5        4          9
D12              3        2          5
E1               4        7         11
E2               9        4         13
E3              12       11         23

TOTAL           58       45        103
```

ON TABLE による列合計の指定

次のリクエストでは、ON TABLE コマンドに COLUMN-TOTAL を使用します。

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL
BY LAST_NAME
ON TABLE COLUMN-TOTAL
END
```

出力結果は次のとおりです。

```
LAST_NAME               CURR_SAL
---------               --------
BANNING               $29,700.00
BLACKWOOD             $21,780.00
CROSS                 $27,062.00
GREENSPAN              $9,000.00
IRVING                $26,862.00
JONES                 $18,480.00
MCCOY                 $18,480.00
MCKNIGHT              $16,100.00
ROMANS                $21,120.00
SMITH                 $13,200.00
                       $9,500.00
STEVENS               $11,000.00

TOTAL                $222,284.00
```

マトリックスレポートでの行合計と列合計の使用

次のリクエストでは、BY および ACROSS 句の両方を使用して作成したマトリックスレポートに、ROW-TOTAL および COLUMN-TOTAL を使用します。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL AND ROW-TOTAL AND COLUMN-TOTAL
BY BANK_NAME
ACROSS DEPARTMENT
END
```

出力結果は次のとおりです。

```
                          DEPARTMENT
BANK_NAME                 MIS              PRODUCTION       TOTAL

------------------------------------------------------------------------
                          $40,680.00       $41,620.00       $82,300.00
ASSOCIATED                $21,780.00       $42,962.00       $64,742.00
BANK ASSOCIATION          $27,062.00                .       $27,062.00
BEST BANK                          .       $29,700.00       $29,700.00
STATE                     $18,480.00                .       $18,480.00
```

```
TOTAL                    $108,002.00      $114,282.00      $222,284.00
```

行合計と列合計への一時項目 (COMPUTE) の追加

次のリクエストでは、一時項目 (COMPUTE) である PROFIT が、行合計および列合計に追加されます。

```
TABLE FILE CAR
SUM DCOST RCOST
COMPUTE PROFIT/D12=RCOST-DCOST;
ROW-TOTAL/L/D12 AS 'TOTAL_COST'
BY COUNTRY
ON TABLE COLUMN-TOTAL/L AS 'FINAL_TOTAL'
END
```

出力結果は次のとおりです。

```
COUNTRY     DEALER_COST  RETAIL_COST          PROFIT       TOTAL_COST
-------     -----------  -----------          ------  ---------------
ENGLAND          37,853       45,319           7,466           90,638
FRANCE            4,631        5,610             979           11,220
ITALY            41,235       51,065           9,830          102,130
JAPAN             5,512        6,478             966           12,956
W GERMANY        54,563       64,732          10,169          129,464

FINAL_TOTAL     143,794      173,204          29,410          346,408
```

ROW-TOTAL の ACROSS および複数表示コマンドとの併用

リクエストに ACROSS フィールドが含まれているときは、レポート出力表示の各フィールド列には、この ACROSS 値が表示されます。たとえば、次のリクエストでは、各州に UNITS (U) および DOLLARS (D) の列が含まれています。

```
TABLE FILE GGSALES
SUM UNITS AS 'U' DOLLARS AS 'D' BY CITY
ACROSS ST
IF ST EQ 'CA'
IF BUDUNITS NE MISSING
END
```

出力結果は次のとおりです。

```
                        State
                        CA
City                          U        D
----------------------------------------
Los Angeles                298070  3772014
San Francisco              312500  3870258
```

ACROSS で行合計を指定するときは、行合計は ACROSS グループのフィールドごとに別々に計算されます。たとえば次のリクエストでは、行合計に UNITS (U) および DOLLARS (D) の列が含まれています。

```
TABLE FILE GGSALES
SUM UNITS AS 'U' DOLLARS AS 'D' BY CITY
ACROSS ST
IF ST EQ 'CA'
IF BUDUNITS NE MISSING
  ON TABLE ROW-TOTAL
END
```

出力結果は次のとおりです。

```
                        State
                        CA                            TOTAL
 City                          U        D         U        D
 ------------------------------------------------------------
 Los Angeles                298070  3772014    298070  3772014
 San Francisco              312500  3870258    312500  3870258
```

リクエストに他の表示コマンドが複数含まれるときは、その他のコマンドは、レポート出力の 各 ACROSS グループにそれぞれ別の列を追加します。

行合計グループの最初の列は、ACROSS 値ごとの表示コマンドに最初の列を追加、2 番目の列は表示コマンドに 2 番目の列を追加することにより計算されます。

たとえば次のリクエストには、UNITS および DOLLARS に対する SUM コマンド、さらに BUDUNITS と BUDDOLLARS に対する SUM コマンドが含まれています。行合計には、

UNITS と BUDUNITS の合計、および DOLLARS と BUDUNITS の合計を表す列が、それぞれ表示されます。

```
TABLE FILE GGSALES
SUM UNITS AS 'U' DOLLARS AS 'D'         BY CITY
SUM BUDUNITS AS 'BU' BUDDOLLARS AS 'BD' BY CITY
ACROSS ST
IF ST EQ 'CA'
IF BUDUNITS NE MISSING
ON TABLE ROW-TOTAL
END
```

出力結果は次のとおりです。

```
                   State
                   CA                                          TOTAL

City                    U         D        BU       BD       BU       BD
-------------------------------------------------------------------------
Los Angeles        298070   3772014    295637  3669484   593707  7441498
San Francisco      312500   3870258    314725  3916863   627225  7787121
```

複数の表示コマンドに、指定されるフィールド数が異なるものが含まれる場合は、列のいくつかは行合計の対象外になります。たとえば、次のリクエストの 2 つ目の SUM コマンドには、BUDUNITS フィールドは含まれていますが、BUDOLLARS フィールドは含まれていません。したがって、行合計グループには DOLLARS を表す列は表示されません。

```
TABLE FILE GGSALES
SUM UNITS AS 'U' DOLLARS AS 'D' BY CITY
SUM BUDUNITS AS 'BU'            BY CITY
ACROSS ST
IF ST EQ 'CA'
IF BUDUNITS NE MISSING
ON TABLE ROW-TOTAL
END
```

出力結果は次のとおりです。

```
                        State
                        CA                              TOTAL
City                         U         D        BU        BU
-------------------------------------------------------------
Los Angeles             298070   3772014    295637    593707
San Francisco           312500   3870258    314725    627225
```

この場合、列表記を使用すると、行合計を正しく計算することができます。たとえば次のリクエストでは、行合計フィールドは Units、Dollars、Budgeted Units フィールドを合計して計算されます。

```
TABLE FILE GGSALES
SUM UNITS AS 'U' DOLLARS AS 'D' BY CITY
SUM BUDUNITS AS 'BU'            BY CITY
ACROSS ST
COMPUTE TOTAL/I10 = C1 + C2 +C3; AS 'ROW-TOTAL'
IF ST EQ 'CA'
IF BUDUNITS NE MISSING
END
```

出力結果は次のとおりです。

```
                         State
                         CA                              ROW-TOTAL
City                          U         D        BU
-------------------------------------------------------------------
Los Angeles              298070   3772014    295637      4365721
San Francisco            312500   3870258    314725      4497483
```

ACROSS フィールド値に対する行合計の生成

ACROSS フィールド値に対する行合計を生成することができます。標準の行合計とは異なり、ACROSS-TOTAL で参照される横方向ソートフィールドの行合計には、ACROSS で参照される ACROSS フィールド値のみが含められます。整数、単精度浮動小数点数、倍精度浮動小数点数、パック 10 進数、長パック 10 進数のすべてを合計することができます。

ACROSS フィールド値に対する行合計の生成

```
ACROSS sortfield ACROSS-TOTAL [AS 'name'] [COLUMNS col1 AND col2 ...]
```

説明

sortfield

横方向にソートするフィールドの名前です。

name

ACROSS-TOTAL のフィールドタイトルの新しい名前です。

col1, col2

合計に含める ACROSS フィールドのタイトルです。

ACROSS フィールド値に対する行合計の生成

次の例は、ACROSS フィールド値に対する行合計を生成する方法を示しています。ここで、RATING の PG フィールドおよび R フィールドの値のみが、TOTAL TITLE COUNT 列の合計値に反映されます。COPIES フィールドの値は、ACROSS フィールドの値ではないため、合計には含まれません。

```
TABLE FILE MOVIES
SUM COPIES BY CATEGORY
COUNT TITLE BY CATEGORY
ACROSS RATING ACROSS-TOTAL
COLUMNS PG AND R
END
```

出力結果は次のとおりです。

```
                  RATING
                  PG     R      TOTAL
CATEGORY  COPIES
----------------------------------------
ACTION        14      2      3      5
COMEDY        16      4      1      5
DRAMA          2      0      1      1
FOREIGN        5      2      3      5
MUSICALS       2      1      1      2
MYSTERY       17      2      5      7
SCI/FI         3      0      3      3
```

ACROSS-TOTAL 使用上の注意

- ACROSS-TOTAL では見出しを積み重ねることはできません。
- 他のタイプのフィールド (文字、テキスト、日付) に ACROSS-TOTAL を使用すると、ブランクの列が生成されます。
- 複数の ACROSS フィールドで ACROSS-TOTAL を使用する場合は、新しいフィールドが追加されて中間合計値が生成されます。
- 表示フィールドが 1 つのみ存在する場合またはプロシジャに表示コマンドが 1 つのみ使用されている場合は、ROW-TOTAL と ACROSS-TOTAL のそれぞれの合計値は一致します。
- ACROSS-TOTAL コンポーネントの最大数は 5 です。
- ACROSS-TOTAL により、スタイルシートの ACROSSVALUE のコンポーネントに値が挿入されます。

セクション合計と総合計の追加

レポートに詳細情報を出力する際、単純な行合計と列合計をいくつかのセクションに分割し、より的確な集計結果を表示したい場合がよくあります。その場合は、特定のセクションの中間合計と総合計を表示する方法が適しています。

セクションの中間合計と総合計を生成するには、リクエストに次のコマンドを追加します。

- SUB-TOTAL および SUBTOTAL
- SUMMARIZE および RECOMPUTE (一時項目 (COMPUTE) とともに使用)
- RECAP および COMPUTE

これらのコマンドは、それぞれ異なる情報を使用して総合計と中間合計を生成します。中間合計は、指定したソートフィールド値が変更されるたびに合計を生成し、レコードの選択条件には依存しません。中間合計を生成する位置を制御するには、WHEN 条件を指定します。詳細は、集計行とテキストの条件付き表示を参照してください。中間合計をデータの上または下のどちらに表示するかを制御することができます。総合計を非表示にするには、NOTOTAL コマンドを使用します。詳細は、総合計の非表示を参照してください。

デフォルト設定では、ブランク行はレポート出力の中間合計の前に生成されます。これらの自動的なブランク行を削除するには、SET DROPBLNKLINE=ON コマンドを発行します。

注意：リクエストで PAGE-BREAK を使用すると、GRANDTOTAL が自動的にページに表示されます。

SUBTOTAL、SUB-TOTAL、SUMMARIZE、RECOMPUTE コマンドには、演算接頭語を使用することができます。また、1 つのリクエストに、異なる集計演算を組み合わせることができます。

セクション合計と総合計の使用

次のリクエストは、部署の値が変更されるたびに、中間合計を生成する方法を示しています。SUBTOTAL コマンドを使用することで、総合計は自動的に生成されます。

```
TABLE FILE EMPLOYEE
SUM DED_AMT BY DED_CODE BY DEPARTMENT
BY BANK_ACCT
WHERE BANK_ACCT NE 0
WHERE DED_CODE EQ 'CITY' OR 'FED'
ON DEPARTMENT SUBTOTAL
END
```

出力結果の最初と最後の部分は次のとおりです。

```
DED_CODE  DEPARTMENT  BANK_ACCT          DED_AMT
--------  ----------  ---------          -------
CITY      MIS          40950036           $14.00
                      122850108           $31.75
                      163800144           $82.70

*TOTAL DEPARTMENT MIS                     $128.45

          PRODUCTION     160633            $7.42
                      136500120           $18.25
                      819000702           $60.20

*TOTAL DEPARTMENT PRODUCTION               $85.87

FED       MIS          40950036        $1,190.77
                      122850108        $2,699.80
                      163800144        $7,028.30

*TOTAL DEPARTMENT MIS                 $10,918.87

          PRODUCTION     160633          $631.12
                      136500120        $1,552.10
```

```
                        819000702        $5,120.04

*TOTAL DEPARTMENT PRODUCTION             $7,303.26


TOTAL                                   $18,436.45
```

中間合計の追加

SUBTOTAL および SUB-TOTAL コマンドを使用して、指定したソートフィールドの値が変更されるたびに、数値列などの個別の値を合計することができます。

- SUB-TOTAL は、ソートフィールドの値が変わるたびに中間合計を表示します。また、そのソートフィールドより上位にあるソートフィールドの値が変わる場合にも中間合計を表示します。
- SUBTOTAL は、指定したソートフィールドの値が変更された場合のみ中間合計を表示します。上位のフィールドに対する中間合計は生成されません。

SUB-TOTAL および SUBTOTAL はともに総合計を生成します。NOTOTAL コマンドを使用して、総合計を非表示にすることもできます。詳細は、総合計の非表示を参照してください。

中間合計の計算は、ソートフィールドの値が変更されるたびに実行されます。また、ソートフィールドに WHEN 条件が適用されている場合は、その WHEN 条件が一致するたびに、計算が実行されます。

構文を初期化するには、BY 句、ACROSS 句、ON 句のいずれかが必要です。

中間合計の生成

```
{BY|ON} fieldname {SUB-TOTAL|SUBTOTAL} [MULTILINES]
      [field1 [AND] field2...] [AS 'text'][WHEN expression;]
```

説明

fieldname

SORT 句で指定したフィールド名を指定する必要があります。BY 句には、集計コマンドを含めることができます。中間合計を表示するフィールド数に中間合計のレベル数を乗算した値は、リクエストで使用可能な表示フィールドの上限数に含められます。リクエストで使用可能な表示フィールドの最大数を決定する方法についての詳細は、レポートデータの表示を参照してください。

SUB-TOTAL|SUBTOTAL

SUB-TOTAL は、BY|ON で指定したフィールドの値が変更されたところで中間合計を表示します。また、上位のソートフィールドの値が変更された場合に、それらのフィールドの中間合計を表示します。

SUBTOTAL は、指定したソートフィールドの値が変更された場合のみ中間合計を表示します。

MULTILINES

詳細データ行が 1 行しか存在しない場合、詳細データの 1 行と中間合計の値が常に等しくなるため、ソート区切りの中間合計行を非表示にします。なお、MULTI-LINES は MULTILINES と同義です。MULTILINES は ACROSS フィールドではサポートされません。

field1, field2, ...

中間合計を計算する特定のフィールド群を指定します。デフォルト設定では、すべての数値表示フィールドが対象になりますが、ここでフィールド名を指定すると、そのフィールド名が優先されます。フィールドリストには、数値フィールドおよび文字フィールドを含めることができます。

すべてのフィールド (数値および文字フィールド) を集計行に含める場合は、フィールドリストの代わりにアスタリスク (*) ワイルドカード文字を使用することができます。

AS 'text'

異なるラベルを指定することができます。

WHEN expression

ブール式で特定する中間合計の表示条件を指定します。式の末尾にセミコロン (;) を追加する必要があります。

集計行の配置制御

```
SET SUBTOTALS = {ABOVE|BELOW}
```

説明

ABOVE

集計行を詳細行の上部に配置し、レポート出力の各詳細行にソートフィールド値を表示します。

BELOW

集計行を詳細行の下部に配置します。デフォルト値は BELOW です。

注意：SET SUBTOTALS = ABOVE は、XLSX、EXL07、EXL2K FORMULA フォーマットではサポートされません。

データ上部への中間合計配置

次のリクエストは、EMPLOYEE データソースが使用され、控除額および給与総額が、部門別、控除コード別、姓別に集計されます。さらに、控除額および給与総額の中間合計が、部門別に集計されます。次のリクエストにより、中間合計が詳細行の下部に配置されます (デフォルト設定)。

```
TABLE FILE EMPLOYEE
SUM DED_AMT GROSS
BY DEPARTMENT
BY DED_CODE
  BY LAST_NAME
WHERE BANK_ACCT NE 0
WHERE DED_CODE EQ 'FICA' OR 'CITY'
  ON DEPARTMENT SUBTOTAL
  ON TABLE SET SUBTOTALS BELOW
  ON TABLE SET PAGE NOPAGE
END
```

出力結果は次のとおりです。

```
DEPARTMENT  DED_CODE  LAST_NAME               DED_AMT           GROSS
----------  --------  ---------               -------           -----
```

```
MIS         CITY      BLACKWOOD                $31.76        $9,075.00
                      CROSS                    $82.69       $22,013.77
                      JONES                    $14.01        $6,099.50
            FICA      BLACKWOOD             $2,223.37        $9,075.00
                      CROSS                 $5,788.01       $22,013.77
                      JONES                   $980.64        $6,099.50

*TOTAL DEPARTMENT MIS                       $9,120.47       $74,376.54

PRODUCTION  CITY      BANNING                   $7.42        $2,475.00
                      IRVING                   $60.24       $17,094.00
                      MCKNIGHT                 $18.26        $9,129.99
            FICA      BANNING                 $519.75        $2,475.00
                      IRVING                $4,216.53       $17,094.00
                      MCKNIGHT              $1,278.21        $9,129.99

*TOTAL DEPARTMENT PRODUCTION                $6,100.40       $57,397.98


TOTAL                                      $15,220.88      $131,774.52
```

次のリクエストは同一のものですが、中間合計は詳細行の上部に表示されます。

```
TABLE FILE EMPLOYEE
SUM DED_AMT GROSS
BY DEPARTMENT
BY DED_CODE
  BY LAST_NAME
WHERE BANK_ACCT NE 0
WHERE DED_CODE EQ 'FICA' OR 'CITY'
  ON DEPARTMENT SUBTOTAL
  ON TABLE SET SUBTOTALS ABOVE
  ON TABLE SET PAGE NOPAGE
END
```

出力では、最初に総合計行が表示され、次に MIS 部門の中間合計、続いて MIS 部門の詳細行、PRODUCTION 部門の中間合計および詳細行が表示されます。レポート出力の各行に、すべてのソートフィールド値が表示されます。

```
DEPARTMENT  DED_CODE  LAST_NAME              DED_AMT           GROSS
----------  --------  ---------              -------           -----
TOTAL                                      $15,220.88      $131,774.52
*TOTAL DEPARTMENT MIS                       $9,120.47       $74,376.54

MIS         CITY      BLACKWOOD                $31.76        $9,075.00
```

```
MIS          CITY       CROSS                    $82.69       $22,013.77
MIS          CITY       JONES                    $14.01        $6,099.50
MIS          FICA       BLACKWOOD             $2,223.37        $9,075.00
MIS          FICA       CROSS                 $5,788.01       $22,013.77
MIS          FICA       JONES                   $980.64        $6,099.50

*TOTAL DEPARTMENT PRODUCTION                  $6,100.40       $57,397.98

PRODUCTION   CITY       BANNING                   $7.42        $2,475.00
PRODUCTION   CITY       IRVING                   $60.24       $17,094.00
PRODUCTION   CITY       MCKNIGHT                 $18.26        $9,129.99
PRODUCTION   FICA       BANNING                 $519.75        $2,475.00
PRODUCTION   FICA       IRVING                $4,216.53       $17,094.00
PRODUCTION   FICA       MCKNIGHT              $1,278.21        $9,129.99
```

中間合計使用上の注意

- SUM または COUNT コマンドに SORT 句を 1 つのみ使用した場合、SUB-TOTAL および SUBTOTAL の計算結果は、SUM または COUNT コマンドで生成される値と等しくなります。ただし、SORT 句が 1 つのみの PRINT コマンドを使用する場合は、ソート区切り内に複数の値が存在する可能性があるため、SUBTOTAL が役立ちます。
- すべての SUB-TOTAL は、ソート区切りが発生する点を含めたそれ以前の点を表示するため、リクエストには中間合計の最も内側の点を指定するようにします。たとえば、次のように複数の BY フィールドを指定することを想定します。

  ```
  BY AREA
  BY PROD_CODE
  BY DATE SUB-TOTAL
  ```

  AREA が変更されると、DATE、PROD_CODE、AREA に対する中間合計が 3 行に (それぞれの下に) 表示されます。
- WHERE TOTAL または IF TOTAL テストを使用する場合、PRINTPLUS を ON にしない限り、中間合計行のソートフィールドの値は表示されません。WebFOCUS で PRINTPLUS を使用する方法についての詳細は、見出し、脚注、その内部の項目の配置を参照してください。
- 中間合計テキストが表示フィールド列の前の行に収まらない場合、中間合計はその次の行に表示されます。

- リクエストの複数の BY 句に SUBTOTAL/SUMMARIZE/RECOMPUTE/SUB-TOTAL が複数のレベルで使用され、そのいずれかのレベルで MULTILINES または MULTI-LINES が指定されている場合、すべてのレベルに MULTILINES または MULTI-LINES が適用されます。

  注意：ON BYfield SUBFOOT は、指定したレベルにのみ適用されます。

中間合計の生成

次のリクエストでは、国名の値が変更されるたびに、SALES に対する中間合計が生成されます。

```
TABLE FILE CAR
SUM AVE.MPG AND SALES AND AVE.RETAIL_COST
BY COUNTRY SUB-TOTAL SALES
BY BODYTYPE
END
```

出力結果は次のとおりです。

```
                           AVE           AVE
COUNTRY     BODYTYPE       MPG    SALES  RETAIL_COST
-------     --------       ----   -----  -----------
ENGLAND     CONVERTIBLE      16       0        8,878
            HARDTOP          25       0        5,100
            SEDAN            10   12000       15,671

*TOTAL ENGLAND                    12000

FRANCE      SEDAN            21       0        5,610

*TOTAL FRANCE                         0

ITALY       COUPE            11   12400       19,160
            ROADSTER         21   13000        6,820
            SEDAN            21    4800        5,925
*TOTAL ITALY

JAPAN       SEDAN            14   78030        3,239

*TOTAL JAPAN                      78030

W GERMANY   SEDAN            20   88190        9,247
```

```
*TOTAL W GERMANY                     88190

TOTAL                               208420
```

SUB-TOTAL と SUBTOTAL の比較

次のリクエストは、部署の値が変更されたところで、数値フィールド DED_AMT および GROSS の中間合計を作成します。また、上位ソートフィールド (DED_CODE) の値が変更されたところでは、それらのフィールドの中間合計を生成します。

```
TABLE FILE EMPLOYEE
SUM DED_AMT GROSS BY DED_CODE BY DEPARTMENT
BY BANK_ACCT
WHERE BANK_ACCT NE 0
ON DEPARTMENT SUB-TOTAL
END
```

SUB-TOTAL の代わりに SUBTOTAL を使用すると、DED_AMT および GROSS の合計は、DEPARTMENT 値が変更された場合にのみ表示されます。

下図は、出力結果の最初の部分です。

```
DED_CODE  DEPARTMENT  BANK_ACCT          DED_AMT            GROSS
--------  ----------  ---------          -------            -----
CITY      MIS          40950036           $14.00        $6,099.50
                      122850108           $31.75        $9,075.00
                      163800144           $82.70       $22,013.75

*TOTAL DEPARTMENT MIS                    $128.45       $37,188.25

          PRODUCTION     160633            $7.42        $2,475.00
                      136500120           $18.25        $9,130.00
                      819000702           $60.20       $17,094.00

*TOTAL DEPARTMENT PRODUCTION              $85.87       $28,699.00
*TOTAL DED_CODE CITY                     $214.32       $65,887.25
```

下図は、出力結果の最後の部分です。

```
DED_CODE  DEPARTMENT  BANK_ACCT          DED_AMT            GROSS
--------  ----------  ---------          -------            -----
STAT      MIS          40950036          $196.13         $6,099.50
                      122850108          $444.65         $9,075.00
                      163800144        $1,157.60        $22,013.75

*TOTAL DEPARTMENT MIS                  $1,798.38        $37,188.25

          PRODUCTION     160633          $103.95         $2,475.00
                      136500120          $255.65         $9,130.00
                      819000702          $843.32        $17,094.00

*TOTAL DEPARTMENT PRODUCTION           $1,202.92        $28,699.00
*TOTAL DED_CODE STAT                   $3,001.30        $65,887.25


TOTAL                                 $41,521.18       $461,210.75
```

中間合計行値の再計算

SUB-TOTAL と SUBTOTAL コマンドの代わりに SUMMARIZE と RECOMPUTE コマンドを使用して、COMPUTE コマンドで得られた結果を再計算することができます。SUMMARIZE は、ソート区切りごとに値を再計算するという点で SUB-TOTAL に類似しています。RECOMPUTE は、指定したソート区切りにおいてのみ値を再計算するという点で SUBTOTAL に類似しています。

SUMMARIZE は、レポート全体の総合計を再計算します。リクエストに NOTOTAL コマンドを使用して、総合計を非表示にすることができます。詳細は、総合計の非表示を参照してください。

一時項目 (COMPUTE) の中間合計

```
{BY|ON} fieldname {SUMMARIZE|RECOMPUTE} [MULTILINES]
      [field1 [AND] field2...] [AS 'text'][WHEN expression;]
```

説明

fieldname

SORT 句で指定したフィールド名を指定する必要があります。BY 句には、集計コマンドを含めることができます。集計を表示するフィールド数に集計コマンドのレベル数を乗算した値は、リクエストで使用可能な表示フィールドの上限数に含められます。リクエストで使用可能な表示フィールドの最大数を決定する方法についての詳細は、レポートデータの表示を参照してください。

SUMMARIZE

ソート区切りごとに値を再計算します。

RECOMPUTE

指定したソート区切りでのみ値を再計算します。

MULTILINES

詳細データ行が 1 行しか存在しない場合、詳細データの 1 行と中間合計の値が常に等しくなるため、ソート区切りの中間合計行を非表示にします。なお、MULTI-LINES は MULTILINES と同義です。MULTILINES は ACROSS フィールドではサポートされません。

NOTOTAL コマンドを使用して、総合計を非表示にすることもできます。詳細は、総合計の非表示を参照してください。

AS 'text'

異なるラベルを指定することができます。

field1, field2, ...

RECOMPUTE または SUMMARIZE の後に、中間合計を計算する特定のフィールドを指定します。デフォルト設定では、すべての数値表示フィールドが対象になりますが、ここでフィールド名を指定すると、そのフィールド名が優先されます。

すべてのフィールド (数値および文字フィールド) を集計行に含める場合は、フィールドリストの代わりにアスタリスク (*) ワイルドカード文字を使用することができます。アスタリスク (*) を使用してすべてのフィールドを表示することも、表示する特定のフィールド (数値および文字フィールド) を参照することもできます。

WHEN expression

ブール式に基づいて中間合計の表示条件を指定します。式の末尾にセミコロン (;) を追加する必要があります。

ON TABLE コマンドを使用して、再計算した値の中間合計を指定することもできます。使用する構文は次のとおりです。

```
ON TABLE SUMMARIZE
```

SUMMARIZE の使用

次のリクエストでは、SUMMARIZE を使用して、指定したソート区切りの DEPARTMENT およびその上位のソート区切りの PAY_DATE で DG_RATIO を再計算します。

```
TABLE FILE EMPLOYEE
SUM GROSS DED_AMT AND COMPUTE
DG_RATIO/F4.2=DED_AMT/GROSS;
BY HIGHEST PAY_DATE BY DEPARTMENT
BY BANK_ACCT
WHERE BANK_ACCT NE 0
ON DEPARTMENT SUMMARIZE
END
```

下図は、出力結果の最初の部分です。

```
PAY_DATE  DEPARTMENT  BANK_ACCT          GROSS          DED_AMT  DG_RATIO
--------  ----------  ---------          -----          -------  --------
82/08/31  MIS          40950036      $1,540.00          $725.34       .47
                      122850108      $1,815.00        $1,261.40       .69
                      163800144      $2,255.00        $1,668.69       .74

*TOTAL DEPARTMENT MIS                $5,610.00        $3,655.43       .65

          PRODUCTION     160633      $2,475.00        $1,427.24       .58
                      136500120      $1,342.00          $522.28       .39
                      819000702      $2,238.50        $1,746.03       .78

*TOTAL DEPARTMENT PRODUCTION         $6,055.50        $3,695.55       .61
*TOTAL PAY_DATE 82/08/31            $11,665.50        $7,350.98       .63
```

下図は、出力結果の最後の部分です。

```
PAY_DATE  DEPARTMENT  BANK_ACCT          GROSS          DED_AMT  DG_RATIO
--------  ----------  ---------          -----          -------  --------
82/01/29  PRODUCTION  819000702      $2,035.00        $1,241.33       .61

*TOTAL DEPARTMENT PRODUCTION         $2,035.00        $1,241.33       .61
*TOTAL PAY_DATE 82/01/29             $4,182.75        $2,648.12       .63
```

```
81/12/31  MIS         163800144      $2,147.75        $1,406.79       .66

*TOTAL DEPARTMENT MIS                $2,147.75        $1,406.79       .66
*TOTAL PAY_DATE 81/12/31             $2,147.75        $1,406.79       .66

81/11/30  MIS         163800144      $2,147.75        $1,406.79       .66

*TOTAL DEPARTMENT MIS                $2,147.75        $1,406.79       .66
*TOTAL PAY_DATE 81/11/30             $2,147.75        $1,406.79       .66


TOTAL                               $65,887.25       $41,521.18       .63
```

ヒント：SUMMARIZE の代わりに SUB-TOTAL または SUBTOTAL を使用する場合は、DG_RATIO の値が追加されます。

RECOMPUTE の使用

次のリクエストでは、RECOMPUTE を使用して、指定したソート区切りの DEPARTMENT でのみ DG_RATIO を再計算します。

```
TABLE FILE EMPLOYEE
SUM GROSS DED_AMT AND COMPUTE
DG_RATIO/F4.2=DED_AMT/GROSS;
BY HIGHEST PAY_DATE BY DEPARTMENT
BY BANK_ACCT
WHERE BANK_ACCT NE 0
ON DEPARTMENT RECOMPUTE
END
```

下図は、出力結果の最初の部分です。

```
PAY_DATE  DEPARTMENT  BANK_ACCT          GROSS          DED_AMT  DG_RATIO
--------  ----------  ---------          -----          -------  --------
82/08/31  MIS          40950036      $1,540.00          $725.34       .47
                      122850108      $1,815.00        $1,261.40       .69
                      163800144      $2,255.00        $1,668.69       .74

*TOTAL DEPARTMENT MIS                $5,610.00        $3,655.43       .65

          PRODUCTION     160633      $2,475.00        $1,427.24       .58
                      136500120      $1,342.00          $522.28       .39
```

```
                     819000702      $2,238.50        $1,746.03       .78

*TOTAL DEPARTMENT PRODUCTION        $6,055.50        $3,695.55       .61

82/07/30  MIS          40950036      $1,540.00          $725.34       .47
                     122850108      $1,815.00        $1,261.40       .69
```

下図は、出力結果の最後の部分です。

```
PAY_DATE  DEPARTMENT  BANK_ACCT          GROSS          DED_AMT  DG_RATIO
--------  ----------  ---------          -----          -------  --------
82/01/29  MIS         163800144      $2,147.75        $1,406.79       .66

*TOTAL DEPARTMENT MIS                $2,147.75        $1,406.79       .66

          PRODUCTION  819000702      $2,035.00        $1,241.33       .61

*TOTAL DEPARTMENT PRODUCTION         $2,035.00        $1,241.33       .61

81/12/31  MIS         163800144      $2,147.75        $1,406.79       .66

*TOTAL DEPARTMENT MIS                $2,147.75        $1,406.79       .66

81/11/30  MIS         163800144      $2,147.75        $1,406.79       .66

*TOTAL DEPARTMENT MIS                $2,147.75        $1,406.79       .66


TOTAL                               $65,887.25       $41,521.18       .63
```

文字フィールドの集計

デフォルト設定では、中間合計 (SUBTOTAL および SUB-TOTAL コマンドの使用) および再計算 (RECOMPUTE および SUMMARIZE コマンドの使用) は、数値レポートフィールドの値のみを表示します。ただし、集計行に表示するフィールドを指定するか、ワイルドカード文字のアスタリスク (*) を使用して集計行にすべてのフィールドを表示するかのいずれかの方法で、これらの集計行に文字フィールドを表示することができます。

SUBTOTAL または SUB-TOTAL 行に表示される文字値は、SUMPREFIX パラメータの値に基づいて、ソートグループ内の最初、最小、最大、最後の文字値のいずれかになります。RECOMPUTE または SUMMARIZE 行では、その行の集計値を使用して文字値が再計算されます。

集計行でのすべてのフィールドの表示

```
ON sortfieldsummarycommand *
```

説明

sortfield

値が変更されたときに集計行が開始されるソートフィールドです。

summarycommand

SUBTOTAL、SUB-TOTAL、RECOMPUTE、SUMMARIZE のいずれかです。

*****

このアスタリスク (*) は、すべてのフィールド、つまり数値フィールドおよび文字フィールドを集計行に含める必要があることを示しています。アスタリスク (*) を使用してすべてのフィールドを表示することも、表示する特定のフィールドを参照することもできます。

集計行での文字フィールドの表示

次のリクエストは、GGSALES データソースに対して実行され、DOLLARS および UNITS フィールドを文字に変換し、次に DOLLARS と UNITS の割合を計算し、最後に DOLLARS と UNITS の数値割合を計算します。RECOMPUTE * コマンドは、State ソートフィールドの値の変更時にすべての値を再計算します。

```
SET SUMPREFIX=FST
TABLE FILE GGSALES
SUM PRODUCT DOLLARS/I8M AS 'Dollars' IN 22 UNITS AS 'Units'
COMPUTE Formula/A19 = EDIT(DOLLARS)|'/'|EDIT(UNITS)|'=';
COMPUTE Ratio/F8    = DOLLARS/UNITS;
BY ST
BY CATEGORY NOPRINT
WHERE ST EQ 'CA' OR 'IL'
ON ST RECOMPUTE *
ON TABLE SET PAGE NOPAGE
END
```

出力結果では、文字の計算は、集計された数値フィールドを使用して再計算されます。ただし、Product フィールドの値は、このフィールドが再計算されず、SUMPREFIX=FST に設定されているため、各ソートフィールド値の最初の Product の値から取得されます。

```
State  Product           Dollars     Units  Formula                 Ratio
-----  -------           -------     -----  -------                 -----
CA     Capuccino      $2,957,852    237246  02957852/00237246=         12
       Biscotti       $2,770,508    222844  02770508/00222844=         12
       Coffee Grinder $1,935,863    152276  01935863/00152276=         13

*TOTAL CA
       Capuccino      $7,664,223    612366  07664223/00612366=         13

IL     Espresso       $1,398,779    109581  01398779/00109581=         13
       Biscotti       $1,561,904    120976  01561904/00120976=         13
       Coffee Grinder $1,050,243     83541  01050243/00083541=         13

*TOTAL IL
       Espresso       $4,010,926    314098  04010926/00314098=         13


TOTAL  Capuccino     $11,675,149    926464  11675149/00926464=         13
```

ここで、SUBTOTAL 集計コマンドを使用した場合は、この計算は再計算されず、各ソートグループの先頭行の値が表示されます。

文字フィールド集計の使用上の注意

- 日付フィールド、および日付フォーマットオプションが指定された数値フィールドと文字フィールドは、集計行に表示されません。
- フィールドの合計行は、集計行と同一の規則に従います。
- ACROSS ソートフィールドの集計値がサポートされます。

演算接頭語による集計値の操作

ON TABLE レベルで SUBTOTAL、SUB-TOTAL、RECOMPUTE、SUMMARIZE のいずれかのコマンドを使用する場合、集計演算のタイプを指定してレポートに総合計行を出力することができます。

また、ソート区切りと総合計の両レベルにおいて、集計オプションの SUBTOTAL、SUB-TOTAL、RECOMPUTE、SUMMARIZE に演算接頭語を使用することができます。SUM コマンドで複数の演算接頭語を使用して同一フィールドを集計した場合、演算接頭語とフィールド名を組み合わせることで、集計コマンドで複数の接頭語が指定された各フィールドを区別することができます。

集計行での接頭語による演算は、レポートの詳細行に表示される値の取得、選択、合計の後に実行されます。フィールドに基づいた演算接頭語とは異なり、この接頭語による演算は入力レコードごとには実行されません。

集計タイプによりその目的は異なるため、レポートに表示する集計情報のタイプに合わせて演算接頭語を操作します。たとえば、AVE. をソートフィールド区切りで使用すると、そのソートグループ内の平均が生成されます。

集計行に文字フィールドを表示することもできます。その場合は、集計コマンドで文字フィールド名を明示的に記述するか、ワイルドカードとしてアスタリスク (*) を使用してすべてのフィールドを表示させる必要があります。

1 つのソート区切りに対する 2 つの ON 句から異なる演算を実行すると、同一の集計行に表示され、集計行で異なる演算の組み合わせが可能になります。総合計を計算するコマンドでフィールドが指定されていない場合であっても、総合計行には集計コマンドで生成したすべてのフィールドの値が生成されます。

1 つのソート区切りに対して複数の ON 句で同一フィールドが参照されている場合、最後に指定された関数が適用されます。

数値フィールドでは、次の演算接頭語を使用することができます。

- ASQ.
- AVE.
- CNT.
- FST.
- LST.
- MAX.
- MIN.
- SUM.

文字フィールドでは、次の演算接頭語を使用することができます。

- FST.
- LST.

- MAX.
- MIN.
- SUM. (SUMPREFIX=LST または FST. で、SUMPREFIX=FST の場合は、LST. を意味します)

集計値への演算接頭語の使用

```
{BY|ON} breakfield [AS 'text1'] sumoption [MULTILINES]
        [pref. ] [*|[field1 [[pref2. ] field2 ...]]]
        [AS 'text2'] [WHEN expression;]
```

デフォルトの総合計を置換するには、次の構文を使用します。

```
ON TABLE sumoption [pref. ][field1 [[pref2. ]field2 ...]] [AS 'text2']
```

説明

breakfield

この値が変更された場合に集計演算を開始するソートフィールドです。BY 句には、集計コマンドを含めることができます。ソートフィールドの値が変更されたところで、集計演算が開始されます。

sumoption

SUBTOTAL、SUB-TOTAL、RECOMPUTE、SUMMARIZE のいずれかです。

'text1'

レポート出力の区切りフィールドに使用するフィールド見出しです。

MULTILINES

詳細データ行が 1 行しか存在しない場合は、ソート区切りごとの集計行を非表示にします。MULTILINES を使用すると、演算接頭語を使用して集計行に異なる演算を指定した場合でも、その集計行は非表示になります。MULTI-LINES は MULTILINES と同義です。MULTILINES は ACROSS フィールドではサポートされません。

pref.

演算接頭語です。フィールドリストを使用せずに演算接頭語を指定すると、その演算接頭語はレポート出力のすべての数値列に適用され、集計行のすべての数値列に値が生成されます。

*****

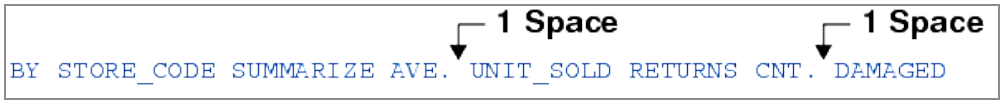
集計行のすべての表示フィールドが含められます。演算接頭語を指定した場合、すべてのフィールドに適用されます。演算接頭語が文字フィールドでサポートされていない場合は、その文字フィールドは集計行には含められません。

[field1 [field2 ... fieldn]]

指定したフィールド名に対して sumoption で指定したタイプの集計が生成されます。指定したフィールド名が存在しない場合は、レポート出力のすべての数値列に集計が生成されます。

pref. field1 [field2 ... fieldn] [pref2. fieldm ...]

最初の演算接頭語が、field1 から fieldn まで適用されます。2 つ目の演算接頭語は、fieldm に適用されます。集計行に値が表示されるのは、指定したフィールドのみです。演算接頭語とフィールド名との間にブランクを追加して両者を区別する必要があります。以下はその例です。



'text2'

集計行の左側に表示するテキストです。

expression

集計演算をソート区切りごとに実行するかどうかを決定する式です。

集計演算接頭語使用上の注意

- 演算接頭語は、COLUMN-TOTAL ではサポートされません。
- 演算接頭語の PCT.、RPCT.、TOT. はサポートされません。
- 二重演算接頭語 (例、PCT.CNT.) はサポートされません。
- リクエストで ACROSS フィールドを使用すると、同一のフィールド名がレポート出力

の複数フィールド (ACROSS グループ) に表示されます。集計行でそのフィールドに適用した演算接頭語は、それらのフィールドすべてに適用されます。

- SUM. 演算接頭語は、演算接頭語のない集計句と同一の集計値を生成します。
- SUMMARIZE および RECOMPUTE は、それに関連する COMPUTE コマンドで定義された計算を集計値に適用します。そのため、必要な計算を実行するには、SUMMARIZE または RECOMPUTE コマンドが、COMPUTE コマンドで参照されたすべてのフィールドを計算する必要があります。
- 1 つのフィールドが複数の集計演算で参照され、その集計演算の各レベルで異なる演算接頭語が使用されている場合は、デフォルトの総合計 (ON TABLE 集計コマンドを使用せずに生成された総合計) が、リクエストの最初の演算子で指定された演算に適用されます (レポート出力で一番左にあるソートフィールド)。

SUBTOTAL と演算接頭語の使用

次の例では、演算接頭語を使用して以下の計算を行います。

- 映画のランク付け別に、作品の価格を平均する。
- ランク付けの各フィールドで、カテゴリ別に作品数を合計する。

ここでは、ランク付けの中間合計行には LISTPR フィールド値のみが表示され、カテゴリの中間合計行には COPIES フィールド値のみが表示されます。総合計行には、中間合計を計算した列の値のみが表示されます。なお、演算接頭語とそれに続くフィールド名の間にブランクが配置されています。

```
TABLE FILE MOVIES
PRINT COPIES LISTPR WHOLESALEPR TITLE/A23
  BY RATING
  BY CATEGORY
  WHERE CATEGORY EQ 'CHILDREN' OR 'CLASSIC'
  WHERE RATING   EQ 'G' OR 'NR'
  ON RATING    SUBTOTAL AVE.LISTPR AS '*Ave:  '
  ON CATEGORY  SUBTOTAL SUM.COPIES AS '*Sum:  '
END
```

出力結果は次のとおりです。

```
RATING  CATEGORY  COPIES  LISTPR  WHOLESALEPR  TITLE
------  --------  ------  ------  -----------  -----
G       CHILDREN       2   44.95        29.99  SHAGGY DOG, THE
```

```
                        2   29.95        12.50  ALICE IN WONDERLAND
                        3   26.99        12.00  BAMBI

*Sum:   CHILDREN        7

        CLASSIC         3   89.95        40.99  GONE WITH THE WIND

*Sum:   CLASSIC         3
*Ave:   G                   47.96

NR      CHILDREN        1   19.95        10.00  SMURFS, THE
                        1   19.95         9.75  SCOOBY-DOO-A DOG IN THE
                        1   14.95         7.65  SESAME STREET-BEDTIME S
                        1   14.98         7.99  ROMPER ROOM-ASK MISS MO
                        1   29.95        15.99  SLEEPING BEAUTY

*Sum:   CHILDREN        5

        CLASSIC         1   24.98        14.99  EAST OF EDEN
                        3   39.99        20.00  CITIZEN KANE
                        1   29.95        15.99  CYRANO DE BERGERAC
                        1   19.99        10.95  MARTY
                        2   19.99        10.95  MALTESE FALCON, THE
                        2   19.95         9.99  ON THE WATERFRONT
                        2   89.99        40.99  MUTINY ON THE BOUNTY
                        2   19.99        10.95  PHILADELPHIA STORY, THE
                        2   19.98        10.99  CAT ON A HOT TIN ROOF
                        2   29.95        15.00  CASABLANCA

*Sum:   CLASSIC        18
*Ave:   NR                  27.64


TOTAL                  33   31.91
```

ソート区切りレベルと総合計レベルでの SUBTOTAL の使用

次の例では、「SUBTOTAL と演算接頭語の使用」で取り上げた前回のリクエストに、ソート区切りレベルで ON TABLE SUBTOTAL コマンドを追加し、作品数の最小値と作品価格の最大値を計算して、レポート全体の総合計行に表示します。

```
TABLE FILE MOVIES
PRINT COPIES LISTPR WHOLESALEPR TITLE/A23
  BY RATING
  BY CATEGORY
  WHERE CATEGORY EQ 'CHILDREN' OR 'CLASSIC'
  WHERE RATING   EQ 'G' OR 'NR'
  ON RATING    SUBTOTAL AVE.LISTPR AS '*Ave:  '
  ON CATEGORY  SUBTOTAL SUM.COPIES AS '*Sum:  '
  ON TABLE SUBTOTAL MIN.COPIES MAX.LISTPR
END
```

前回のリクエストと比べて出力結果が異なる箇所は、総合計行のみです。

```
RATING  CATEGORY  COPIES  LISTPR  WHOLESALEPR  TITLE
------  --------  ------  ------  -----------  -----
G       CHILDREN       2   44.95        29.99  SHAGGY DOG, THE
                       2   29.95        12.50  ALICE IN WONDERLAND
                       3   26.99        12.00  BAMBI

*Sum:   CHILDREN       7

        CLASSIC        3   89.95        40.99  GONE WITH THE WIND

*Sum:   CLASSIC        3
*Ave:   G                  47.96

NR      CHILDREN       1   19.95        10.00  SMURFS, THE
                       1   19.95         9.75  SCOOBY-DOO-A DOG IN THE
                       1   14.95         7.65  SESAME STREET-BEDTIME S
                       1   14.98         7.99  ROMPER ROOM-ASK MISS MO
                       1   29.95        15.99  SLEEPING BEAUTY

*Sum:   CHILDREN       5

        CLASSIC        1   24.98        14.99  EAST OF EDEN
                       3   39.99        20.00  CITIZEN KANE
                       1   29.95        15.99  CYRANO DE BERGERAC
                       1   19.99        10.95  MARTY
                       2   19.99        10.95  MALTESE FALCON, THE
                       2   19.95         9.99  ON THE WATERFRONT
                       2   89.99        40.99  MUTINY ON THE BOUNTY
                       2   19.99        10.95  PHILADELPHIA STORY, THE
                       2   19.98        10.99  CAT ON A HOT TIN ROOF
                       2   29.95        15.00  CASABLANCA

*Sum:   CLASSIC       18
```

```
*Ave:   NR                27.64

TOTAL                  1   89.99
```

複数の演算接頭語が指定されたフィールドの区別

次のリクエストでは、UNITS フィールドに MAX. および MIN. 演算接頭語を使用します。集計コマンドでは、「MAX.UNITS」および「MIN.UNITS」と参照することで、演算接頭語とフィールド名の組み合わせを区別します。

```
TABLE FILE GGSALES
   SUM MAX.UNITS MIN.UNITS
     BY REGION
   BY ST
   ON REGION RECOMPUTE MAX.MAX.UNITS MIN.MIN.UNITS
   WHERE DATE GE 19971001
   WHERE REGION EQ 'West' OR 'Northeast'
   ON TABLE RECOMPUTE MIN.MAX.UNITS MAX.MIN.UNITS
   ON TABLE SET PAGE NOPAGE
   END
```

出力結果では、地域ごとの集計行には、各州の最大値の中で最大の値、各州の最小値の中で最小の値が表示されています。レポート全体の集計行には、各州の最大値の中で最小の値、各州の最小値の中の最大の値が表示されています。下図は、レポート出力を示しています。

Region	State	MAX Unit Sales	MIN Unit Sales
Northeast	CT	1770	101
	MA	1780	146
	NY	1797	73
*TOTAL Northeast		1797	73
West	CA	1794	72
	WA	1787	257
*TOTAL West		1794	72
TOTAL		1770	257

集計行での文字フィールドの表示

次のリクエストは、映画の等級区分別に作品価格フィールド (LISTPR) の合計値と映画監督フィールド (DIRECTOR) の最小値を表示します。

```
TABLE FILE MOVIES
PRINT COPIES LISTPR WHOLESALEPR DIRECTOR
BY RATING
BY CATEGORY
WHERE CATEGORY EQ 'CHILDREN' OR 'CLASSIC'
WHERE RATING   EQ 'G' OR 'NR'
WHERE DIRECTOR NE ' '
ON RATING SUBTOTAL SUM.LISTPR MIN.DIRECTOR AS '*A/N:'
END
```

出力結果は次のとおりです。

```
RATING  CATEGORY  COPIES  LISTPR  WHOLESALEPR  DIRECTOR
------  --------  ------  ------  -----------  --------
G       CHILDREN       2   44.95        29.99  BARTON C.
                       2   29.95        12.50  GEROMINI
```

```
                        3   26.99        12.00  DISNEY W.
        CLASSIC         3   89.95        40.99  FLEMING V

*A/N:G                    191.84               BARTON C.

NR      CHILDREN        1   29.95        15.99  DISNEY W.
        CLASSIC         1   24.98        14.99  KAZAN E.
                        3   39.99        20.00  WELLES O.
                        1   29.95        15.99  GORDON M.
                        1   19.99        10.95  MANN D.
                        2   19.99        10.95  HUSTON J.
                        2   19.95         9.99  KAZAN E.
                        2   89.99        40.99  MILESTONE L.
                        2   19.99        10.95  CUKOR G.
                        2   19.98        10.99  BROOKS R.
                        2   29.95        15.00  CURTIZ M.

*A/N:NR                   344.71               BROOKS R.


TOTAL                      536.55               BARTON C.
```

集計行でのすべてのフィールドの表示

次のリクエストでは、中間合計行のすべての表示フィールドで合計値を表示します。
DIRECTOR フィールドは文字フィールドのため、最後の値が表示されます。

```
TABLE FILE MOVIES
PRINT COPIES LISTPR WHOLESALEPR DIRECTOR
BY RATING
BY CATEGORY
WHERE CATEGORY EQ 'CHILDREN' OR 'CLASSIC'
WHERE RATING   EQ 'G' OR 'NR'
WHERE DIRECTOR NE ' '
ON RATING SUBTOTAL SUM. * AS '*All:  '
END
```

出力結果は次のとおりです。

```
RATING  CATEGORY  COPIES  LISTPR  WHOLESALEPR  DIRECTOR
------  --------  ------  ------  -----------  --------
G       CHILDREN       2   44.95        29.99  BARTON C.
```

```
                          2   29.95        12.50  GEROMINI
                          3   26.99        12.00  DISNEY W.
         CLASSIC          3   89.95        40.99  FLEMING V

*All:    G               10  191.84        95.48  FLEMING V

NR       CHILDREN         1   29.95        15.99  DISNEY W.
         CLASSIC          1   24.98        14.99  KAZAN E.
                          3   39.99        20.00  WELLES O.
                          1   29.95        15.99  GORDON M.
                          1   19.99        10.95  MANN D.
                          2   19.99        10.95  HUSTON J.
                          2   19.95         9.99  KAZAN E.
                          2   89.99        40.99  MILESTONE L.
                          2   19.99        10.95  CUKOR G.
                          2   19.98        10.99  BROOKS R.
                          2   29.95        15.00  CURTIZ M.

*All:    NR              19  344.71       176.79  CURTIZ M.


TOTAL                    29  536.55       272.27  CURTIZ M.
```

集計行処理の制御

集計行を処理する際、SUBTOTAL コマンドと RECOMPUTE コマンドをレポートの総合計行に継承するかどうかを制御することができます。

集計行に演算接頭語を含むフィールドと含まないフィールドが存在する場合、演算接頭語を含まないフィールドは、SUM 演算接頭語が指定されたものとして処理されます。

SET SUMMARYLINES コマンドの機能を使用すると、総合計行での SUBTOTAL、SUB-TOTAL、SUMMARIZE、RECOMPUTE の処理を、ソートフィールド区切りでの動作と一致させることができます。このタイプの処理を呼び出す設定は、SET SUMMARYLINES=EXPLICIT です。

SUBTOTAL と RECOMPUTE をソート区切りレベルで使用する場合は、他のソート区切りには継承されません。SUB-TOTAL と SUMMARIZE は、上位のソート区切りすべてに継承されます。

総合計は、リクエストにおいて最上位のソートフィールドと見なすことができます。ただし、デフォルト設定では、SUB-TOTAL と SUMMARIZE に限定されずに、すべての集計オプションが総合計レベルに継承されます。

SET SUMMARYLINES=EXPLICIT コマンドを使用すると、SUBTOTAL と RECOMPUTE を総合計に継承しないようにすることができます。また、リクエストのすべての集計コマンドにフィールドリストが指定されている場合、指定したフィールドのみが集計されて、総合計行に表示されます。

リクエストで使用する集計コマンドが SUBTOTAL と RECOMPUTE のみの場合は、ON TABLE SUBTOTAL、SUB-TOTAL、RECOMPUTE、SUMMARIZE 句を使用してリクエストに明示的に指定した場合にのみ総合計行が生成されます。ON TABLE 句でフィールドリストが指定された場合は、指定されたフィールドのみが集計、表示されます。

なお、ON TABLE NOTOTAL コマンドを使用して、総合計をいつでも非表示にすることができます。

集計行処理の制御

```
SET SUMMARYLINES = {NEW|OLD|EXPLICIT}
```

説明

NEW

すべての集計演算が総合計行に継承されます。集計コマンドで指定したフィールドは、集計コマンドで作成された集計行および集計コマンドの継承で作成された集計行においてのみ値が生成されます。デフォルト値は NEW です。

SUBTOTAL または SUB-TOTAL 行に表示される文字値は、SUMPREFIX パラメータの値に基づいて、ソートグループ内の最初または最後の文字値になります。RECOMPUTE または SUMMARIZE 行では、その行の集計値を使用して文字値が再計算されます。

OLD

この値は使用できなくなり、NEW として処理されます。

EXPLICIT

SUBTOTAL および RECOMPUTE は、総合計行に継承されません。集計コマンドで指定したフィールドは、集計コマンドで作成された集計行および集計コマンドの継承で作成された集計行においてのみ値が生成されます。

注意：このコマンドは、ON TABLE SET 構文を使用したリクエストではサポートされません。

SET SUMMARYLINES 使用上の注意

- SET SUMMARYLINES は、TABLE リクエスト (ON TABLE) ではサポートされません。
- COLUMN-TOTAL がリクエストで指定されている場合、COLUMN-TOTAL 句で特定のフィールドが指定されていない限り、すべての数値フィールドは総合計行で合計されます。COLUMN-TOTAL 句で特定のフィールドが指定されている場合は、それらのフィールドおよび SUB-TOTAL または SUMMARIZE コマンドで継承されたすべてのフィールドが合計されます。
- 集計コマンドにフィールド名を指定すると、それに関連する集計行のみに値が生成されます。

以下はその例です。

```
TABLE FILE MOVIES
PRINT COPIES LISTPR WHOLESALEPR
 BY RATING
 BY CATEGORY
 WHERE CATEGORY EQ 'CHILDREN'
 WHERE RATING   EQ 'G'
 ON RATING    SUBTOTAL LISTPR AS '*LIST'
 ON CATEGORY  SUBTOTAL  COPIES AS '*COPY'
END
```

出力には、CATEGORY ソート区切りの COPIES、および RATING ソート区切りの LISTPR という中間合計がそれぞれ表示されます。総合計行の両方の列に、値が入力されます。WHOLESALEPR は、どの SUBTOTAL コマンドでも参照されていないため、集計行には表示されません。

```
RATING  CATEGORY  COPIES  LISTPR  WHOLESALEPR
------  --------  ------  ------  -----------
G       CHILDREN       2   44.95        29.99
                       2   29.95        12.50
                       3   26.99        12.00

*COPY CHILDREN         7
*LIST G                   101.89


TOTAL                  7  101.89
```

SET SUMMARYLINES と SUBTOTAL の併用

次のリクエストでは、MOVIES データソースを使用して、CATEGORY のソート区切りで LISTPR フィールドを中間合計し、RATING のソート区切りで COPIES フィールドを中間合計します。

```
TABLE FILE MOVIES
SUM COPIES LISTPR WHOLESALEPR
BY RATING
BY CATEGORY
WHERE CATEGORY EQ 'CHILDREN'
WHERE RATING   EQ 'G'
ON RATING SUBTOTAL COPIES
ON CATEGORY SUBTOTAL LISTPR
END
```

SUMMARYLINES=NEW の場合、RATING のソート区切りでは COPIES のみが中間合計され、CATEGORY のソート区切りでは LISTPR のみが中間合計されますが、それらの両方が総合計行に継承されます。

```
RATING  CATEGORY  COPIES  LISTPR  WHOLESALEPR
------  --------  ------  ------  -----------
G       CHILDREN       7  101.89        54.49

*TOTAL CHILDREN           101.89
*TOTAL G               7


TOTAL                  7  101.89
```

SUMMARYLINES=EXPLICIT の場合、RATING のソート区切りでは COPIES のみが中間合計され、CATEGORY のソート区切りでは LISTPR のみが中間合計されます。総合計行は生成されません。

```
RATING  CATEGORY  COPIES  LISTPR  WHOLESALEPR
------  --------  ------  ------  -----------
G       CHILDREN       7  101.89        54.49

*TOTAL CHILDREN           101.89
*TOTAL G               7
```

SUMMARYLINES=EXPLICIT の場合、ON TABLE SUBTOTAL WHOLESALEPR 句を追加すると、WHOLESALEPR フィールドが中間合計されて総合計行が生成されます。

```
RATING  CATEGORY  COPIES  LISTPR  WHOLESALEPR
------  --------  ------  ------  -----------
G       CHILDREN       7  101.89        54.49

*TOTAL CHILDREN           101.89
*TOTAL G               7


TOTAL                                    54.49
```

COLUMN-TOTAL と SET SUMMARYLINES=EXPLICIT の併用

次のリクエストでは、MOVIES データソースを使用して、CATEGORY のソート区切りで LISTPR フィールドを中間合計し、RATING のソート区切りで COPIES フィールドを中間合計します。ON TABLE COLUMN-TOTAL 句も使用されています。

```
SET SUMMARYLINES=EXPLICIT
TABLE FILE MOVIES
SUM COPIES LISTPR WHOLESALEPR
BY RATING
BY CATEGORY
WHERE CATEGORY EQ 'CHILDREN'
WHERE RATING   EQ 'G'
ON RATING SUBTOTAL COPIES
ON CATEGORY SUBTOTAL LISTPR
ON TABLE COLUMN-TOTAL
END
```

ON TABLE COLUMN-TOTAL 句が使用されているため、総合計行にはすべての数値列が表示されます。

```
RATING  CATEGORY  COPIES  LISTPR  WHOLESALEPR
------  --------  ------  ------  -----------
G       CHILDREN       7  101.89        54.49

*TOTAL CHILDREN           101.89
*TOTAL G               7
```

```
TOTAL                  7  101.89        54.49
```

次のリクエストでは、ON TABLE SUBTOTAL WHOLESALEPR コマンドが使用されています。ON TABLE COLUMN-TOTAL 句も使用されています。

```
SET SUMMARYLINES=EXPLICIT
TABLE FILE MOVIES
SUM COPIES LISTPR WHOLESALEPR
BY RATING
BY CATEGORY
WHERE CATEGORY EQ 'CHILDREN'
WHERE RATING   EQ 'G'
ON RATING SUBTOTAL COPIES
ON CATEGORY SUBTOTAL LISTPR
ON TABLE SUBTOTAL WHOLESALEPR
ON TABLE COLUMN-TOTAL
END
```

ON TABLE SUBTOTAL 句が使用されているため、総合計行には WHOLESALEPR 列の列合計のみが表示されます。

```
RATING  CATEGORY  COPIES  LISTPR  WHOLESALEPR
------  --------  ------  ------  -----------
G       CHILDREN       7  101.89        54.49

*TOTAL CHILDREN           101.89
*TOTAL G               7


TOTAL                                    54.49
```

SUBTOTAL の代わりに SUB-TOTAL を使用すると、COPIES および LISTPR が総合計行に集計されます。WHOLESALEPR は COLUMN-TOTAL 句で指定されているため、合計されます。LISTPR の中間合計は、総合計と RATING ソート区切りの両方に継承されます。

```
SET SUMMARYLINES=EXPLICIT
TABLE FILE MOVIES
SUM COPIES LISTPR WHOLESALEPR
BY RATING
BY CATEGORY
WHERE CATEGORY EQ 'CHILDREN'
WHERE RATING   EQ 'G'
```

```
ON RATING SUB-TOTAL COPIES
ON CATEGORY SUB-TOTAL LISTPR
ON TABLE COLUMN-TOTAL WHOLESALEPR
END
```

出力結果は次のとおりです。

```
RATING  CATEGORY  COPIES  LISTPR  WHOLESALEPR
------  --------  ------  ------  -----------
G       CHILDREN       7  101.89        54.49

*TOTAL CHILDREN           101.89
*TOTAL G               7  101.89


TOTAL                  7  101.89        54.49
```

一時項目 (COMPUTE) での演算接頭語の使用

リクエストに RECOMPUTE または SUMMARIZE コマンドが使用されている場合、関連する COMPUTE コマンドで指定した式が、集計行の値とともに適用されます。その式の再計算に使用するフィールドには、演算接頭語を使用することができます。再計算されたフィールドは、演算接頭語の指定の有無に関わらず、これらの入力値を COMPUTE コマンドで指定した式に適用します。そのため、再計算した値に影響を与えることなく、サポートされている任意の演算接頭語を再計算したレポートフィールドに指定することができます。

COMPUTE コマンドで使用されるすべてのフィールドは、値を生成するために RECOMPUTE または SUMMARIZE コマンドで表示する必要があります。式で使用されたフィールドに値が生成されない場合は、その式に対して返される一時項目 (COMPUTE) を予測することはできません。

RECOMPUTE での演算接頭語の使用

最初のリクエストでは、DOLLARS と BUDDOLLARS の差を表示する、DIFF という名前の一時項目 (COMPUTE) を作成します。さらに、演算接頭語を使用せずにこの値を地域別に再計算します。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS BUDDOLLARS
AND COMPUTE DIFF/I10 = DOLLARS-BUDDOLLARS;
  BY REGION
  BY CATEGORY
  WHERE CATEGORY EQ 'Food' OR 'Coffee'
  WHERE REGION EQ 'West' OR 'Midwest'
  ON REGION  RECOMPUTE
END
```

再計算した値が、DOLLARS と BUDDOLLARS の合計の差になります。

```
Region      Category    Unit Sales Dollar Sales Budget Dollars       DIFF
------      --------    ---------- ------------ --------------       ----
Midwest     Coffee          332777      4178513        4086032      92481
            Food            341414      4338271        4220721     117550

*TOTAL Midwest              674191      8516784        8306753     210031

West        Coffee          356763      4473517        4523963     -50446
            Food            340234      4202337        4183244      19093

*TOTAL West                 696997      8675854        8707207     -31353


TOTAL                      1371188     17192638       17013960     178678
```

次のリクエストでは、RECOMPUTE コマンドに演算接頭語を使用して、DOLLARS の最大値と BUDDOLLARS の最大値を計算し、続いて DIFF を再計算します。DIFF に対してどの演算接頭語を指定するかに関係なく、DOLLARS フィールド値と BUDDOLLARS フィールド値の差として計算されます。計算に使用するフィールド (DOLLARS、BUDDOLLARS、DIFF) はいずれも集計行に表示されないため、この計算を実行することはできません。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS BUDDOLLARS
AND COMPUTE DIFF/I10 = DOLLARS-BUDDOLLARS;
  BY REGION
  BY CATEGORY
  WHERE CATEGORY EQ 'Food' OR 'Coffee'
  WHERE REGION EQ 'West' OR 'Midwest'
  ON REGION RECOMPUTE MAX.DOLLARS MIN.BUDDOLLARS AVE.DIFF
END
```

出力結果は次のとおりです。

```
Region      Category    Unit Sales Dollar Sales Budget Dollars       DIFF
------      --------    ---------- ------------ --------------       ----
Midwest     Coffee          332777      4178513        4086032      92481
            Food            341414      4338271        4220721     117550

*TOTAL Midwest                          4338271        4086032     252239

West        Coffee          356763      4473517        4523963     -50446
            Food            340234      4202337        4183244      19093

*TOTAL West                             4473517        4183244     290273
```

ソート区切りレベルと総合計レベルでの RECOMPUTE の使用

次の例では、「SUBTOTAL と演算接頭語の使用」で取り上げた前回のリクエストに、ON TABLE SUBTOTAL コマンドを追加して、各列の平均値を計算します。ここでは、DIFF の値は、DOLLARS フィールド値と BUDDOLLARS フィールド値の差として計算されます。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS BUDDOLLARS
AND COMPUTE DIFF/I10 = DOLLARS-BUDDOLLARS;
  BY REGION
  BY CATEGORY
  WHERE CATEGORY EQ 'Food' OR 'Coffee'
  WHERE REGION EQ 'West' OR 'Midwest'
  ON REGION  RECOMPUTE MAX.DOLLARS MIN.BUDDOLLARS DIFF
 ON TABLE RECOMPUTE AVE.
END
```

出力結果は次のとおりです。

```
Region      Category    Unit Sales Dollar Sales Budget Dollars       DIFF
------      --------    ---------- ------------ --------------       ----
Midwest     Coffee          332777      4178513        4086032      92481
            Food            341414      4338271        4220721     117550

*TOTAL Midwest                          4338271        4086032     252239

West        Coffee          356763      4473527        4523963     -50436
            Food            340234      4202338        4183244      19094
```

```
*TOTAL West                          4473527        4183244     290283

TOTAL                      342797    4298162        4253490      44672
```

複数 SUB-TOTAL または SUMMARIZE コマンドと演算接頭語の併用

SUB-TOTAL および SUMMARIZE は、上位のソートフィールドすべてに演算を継承します。リクエストで SUB-TOTAL または SUMMARIZE を複数のソートレベルで使用する場合、複数の演算接頭語が 1 つのフィールドに適用されることがあります。

下位のソートフィールドの SUB-TOTAL または SUMMARIZE コマンドが上位に継承される場合、そのコマンドの演算接頭語は、異なる演算接頭語が指定されていない上位のフィールドのみに適用されます。上位で演算接頭語が指定されたフィールドでは、総合計行に異なる演算接頭語が指定されていない限り、元の演算接頭語が最初に指定されたレベルおよび総合計行に適用されます。

複数 SUB-TOTAL コマンドと演算接頭語の併用

次の例では、1 つのリクエストに複数の SUB-TOTAL コマンドを指定し、それぞれのコマンドに異なる演算接頭語を使用します。

```
DEFINE FILE GGSALES
YEAR/YY = DATE;
END

TABLE FILE GGSALES
SUM   UNITS DOLLARS/D10.2 BUDDOLLARS
  BY YEAR
  BY ST
  BY REGION
  BY CATEGORY
WHERE REGION EQ 'West' OR 'Midwest'
WHERE ST     EQ 'CA' OR 'IL'
WHERE YEAR EQ '1996' OR '1997'
  ON YEAR SUB-TOTAL CNT.UNITS AS '*CNT.UNITS:'
```

```
  ON ST SUB-TOTAL AVE.DOLLARS AS '*AVE. $:'
  ON REGION SUB-TOTAL MIN.AS '*MIN.:'
END
```

このレポートの出力結果では、斜体または太字を使用して特定の値を強調しています。

- 説明付きの行は、SUB-TOTAL コマンドで生成された行です。
- 標準書体で表示された中間合計値は、ON YEAR SUB-TOTAL CNT. UNITS. コマンドで生成された売上数量の合計です。これは最上位の集計コマンドであるため、他の集計行には継承されません。
- 斜体で表示された中間合計値は、ON ST SUB-TOTAL AVE. DOLLARS. コマンドで生成された売上高の平均です。これは 2 つ目の集計コマンドであるため、YEAR ソートフィールドに対する集計行の DOLLARS 列に継承されます。
- 太字で表示された中間合計値は、ON REGION SUB-TOTAL MIN. コマンドで生成されたソートグループの最小値です。これは最後の集計コマンドであるため、すべての集計行に継承されますが、数量の集計または平均が生成されていないフィールドの最小値のみを計算します。

YEAR	State	Region	Category	Unit Sales	Dollar Sales	Budget Dollars
1996	CA	West	Coffee	117539	1,484,873.00	1453548
			Food	116389	1,443,083.00	1414902
			Gifts	74948	947,783.00	946382
*MIN.: West				**74948**	**947,783.00**	**946382**
*AVE. $: CA				**74948**	*1,291,913.00*	**946382**
	IL	Midwest	Coffee	52348	693,559.00	628333
			Food	58777	768,715.00	742943
			Gifts	38405	481,515.00	487090
*MIN.: Midwest				**38405**	**481,515.00**	487090
*AVE. $: IL				**38405**	*644,596.33*	487090
*CNT. UNITS: 1996				6	*968,254.67*	**487090**
1997	CA	West	Coffee	118044	1,453,013.00	1507092
			Food	106322	1,325,429.00	1302582
			Gifts	77328	988,080.00	961841
*MIN.: West				**77328**	**988,080.00**	**961841**
*AVE. $: CA				**77328**	*1,255,507.33*	**961841**
	IL	Midwest	Coffee	57233	715,220.00	737931
			Food	59293	754,132.00	737912
			Gifts	41527	521,260.00	532647
*MIN.: Midwest				**41527**	**521,260.00**	**532647**
*AVE. $: IL				**41527**	*663,537.33*	**532647**
*CNT. UNITS: 1997				6	*959,522.33*	**532647**
TOTAL				12	*963,888.50*	**487090**

集計コマンドの組み合わせ

ソート区切り (BY または ACROSS フィールド) ごとに、異なる集計演算を指定することができます。

1 つのソートフィールドに複数の集計コマンドを指定した場合、次のメッセージが表示され、リクエストで最後に指定した集計コマンドが使用されます。

```
(FOC36359)  複数の SUBTOTAL/SUB-TOTAL/RECOMPUTE/SUMMARIZE があります。
```

1 つのキーフィールドに複数の SUBTOTAL、SUB-TOTAL、RECOMPUTE、SUMMARIZE を指定することはできません。最後に指定されたものが優先されます。

集計演算を上位のソート区切りに継承する SUMMARIZE および SUB-TOTAL は、上位のソート区切りで特定の集計コマンドが指定されたフィールドを無視します。集計演算の継承は、その集計行に使用された演算接頭語の有無により異なります。演算接頭語の有無により、次のように継承されます。

- 集計行で使用されない場合、集計コマンドでフィールドリストを指定すると、集計行のフィールドリストで指定されたフィールドのみの値がレポートに生成されます。
- 集計行で使用される場合、SUB-TOTAL および SUMMARIZE は、次のフィールドに継承されます。
    - 上位のソート区切りにあるフィールドで、特定の集計コマンドが指定されていないすべてのフィールド。
    - 特定の集計コマンドが指定された上位のソート区切りで、フィールドリストに指定されていないフィールド (空になる列)。なお、1 つのソート区切りで異なるフィールドに異なる集計オプションを指定する場合、これが唯一の方法です。

コマンドが RECOMPUTE と SUMMARIZE の場合でも、SUBTOTAL と SUB-TOTAL の場合でも、集計行の演算接頭語は同一の値を生成します。一時項目 (COMPUTE) の場合、演算接頭語は適用されず、COMPUTE コマンドの式および集計行の値を使用して値が再計算されます。

異なるソートフィールドに異なる集計コマンドを使用する場合、デフォルトの総合計行には、リクエストで指定した最初のソートフィールドに関連する集計コマンドが継承されます。総合計レベルで実行する処理を変更するには、ON TABLE 句を使用して特定の集計コマンドを指定します。

注意：総合計レベルは最上位のソートレベルと見なされます。そのため、SUMMARIZE または SUB-TOTAL コマンドを総合計レベルで使用することはできますが、これらのコマンドは総合計レベルにのみ適用され、レポートの他の列には継承されません。総合計レベルでは、SUMMARIZE は RECOMPUTE コマンドのように動作し、SUB-TOTAL は SUBTOTAL コマンドのように動作します。

リクエストでの SUBTOTAL と RECOMPUTE の使用

次のリクエストでは、最初に指定されたソートフィールドは COPIES で、このフィールドが RECOMPUTE コマンドと関連付けられています。そのため、総合計行では RATIO の値が正確に再計算され、LISTPR と WHOLESALEPR の値が合計されます (これが、フィールドの計算を COMPUTE コマンドで実行しない場合のデフォルト処理です)。

```
TABLE FILE MOVIES
PRINT DIRECTOR LISTPR WHOLESALEPR
COMPUTE RATIO = LISTPR/WHOLESALEPR;
BY COPIES
BY RATING
WHERE COPIES LT 3
WHERE DIRECTOR EQ 'DISNEY W.'OR 'HITCHCOCK A.'
ON COPIES RECOMPUTE AS '*REC: '
ON RATING SUBTOTAL AS '*SUB:  '
END
```

出力結果は次のとおりです。

```
COPIES  RATING  DIRECTOR           LISTPR  WHOLESALEPR           RATIO
------  ------  --------           ------  -----------           -----
     1  NR      DISNEY W.           29.95        15.99            1.87

*SUB:   NR                          29.95        15.99            1.87
*REC:    1                          29.95        15.99            1.87

     2  NR      HITCHCOCK A.        19.98         9.00            2.22

*SUB:   NR                          19.98         9.00            2.22

        PG      HITCHCOCK A.        19.98         9.00            2.22
                HITCHCOCK A.        19.98         9.00            2.22

*SUB:   PG                          39.96        18.00            4.44
     2  PG13    HITCHCOCK A.        19.98         9.00            2.22

*SUB:   PG13                        19.98         9.00            2.22

        R       HITCHCOCK A.        19.98         9.00            2.22
```

```
*SUB:   R                          19.98         9.00           2.22
*REC:    2                         99.90        45.00           2.22


TOTAL                             129.85        60.99           2.13
```

ここで、BY フィールドを入れ替えると、SUBTOTAL コマンドが総合計行を制御するため、総合計行では LISTPR と WHOLESALEPR の値だけではなく RATIO の値も合計されます。

```
TOTAL                             129.85        60.99          12.97
```

総合計レベルで実行する処理を変更するには、リクエストに次のコマンドを追加します。

```
ON TABLE RECOMPUTE
```

これにより、総合計行には再計算された値が表示されます。

```
TOTAL                             129.85        60.99           2.13
```

SUB-TOTAL と複数の集計コマンドとの併用

次のリクエストでは、SUB-TOTAL コマンドがソートフィールドの DIRECTOR に処理を継承します。ここでは、HITCHCOCK の合計の行から、RATIO 値は再計算されず、中間合計されていることが分かります。

ソートフィールドの RATING には特定の RECOMPUTE コマンドが指定されているため SUB-TOTALは継承されません。このソートフィールドでは、RATIO の値が再計算されます。RECOMPUTE が SUB-TOTAL より上位のソートフィールドで実行されるため、総合計行は再計算されます。

```
TABLE FILE MOVIES
PRINT LISTPR WHOLESALEPR
COMPUTE RATIO = LISTPR/WHOLESALEPR;
BY DIRECTOR
BY RATING
BY COPIES
WHERE COPIES LT 3
WHERE DIRECTOR EQ 'HITCHCOCK A.'
ON COPIES SUB-TOTAL AS '*SUB: '
```

```
ON RATING RECOMPUTE AS '*REC:  '
END
```

出力結果は次のとおりです。

```
DIRECTOR           RATING  COPIES  LISTPR  WHOLESALEPR           RATIO
--------           ------  ------  ------  -----------           -----
HITCHCOCK A.       NR           2   19.98         9.00            2.22

*SUB:    2                        19.98         9.00            2.22
*REC:   NR                        19.98         9.00            2.22

                   PG           2   19.98         9.00            2.22
                                    19.98         9.00            2.22

*SUB:    2                        39.96        18.00            4.44
*REC:   PG                        39.96        18.00            2.22

                   PG13         2   19.98         9.00            2.22
*SUB:    2                        19.98         9.00             2.2
*REC:   PG13                      19.98         9.00             2.2

HITCHCOCK A.       R            2   19.98         9.00             2.2

*SUB:    2                        19.98         9.00             2.2
*REC:   R                         19.98         9.00             2.2
*TOTAL DIRECTOR HITCHCOCK A.      99.90        45.00            11.1


TOTAL                             99.90        45.00             2.2
```

複数の集計コマンドと演算接頭語の使用

次のリクエストでは、ソートフィールドの RATING に関連する行に、LISTPR の平均値と RATIO の再計算値を表示します。ソートフィールドの COPIES に関連付けられた SUB-TOTAL コマンドは、ソートフィールドの DIRECTOR 行にあるすべてのフィールドおよびソートフィールドの RATING に関連する WHOLESALEPR と RATIO1 の各列に継承されます。このリクエストでは、総合計行は非表示になります。

```
TABLE FILE MOVIES
PRINT LISTPR WHOLESALEPR
COMPUTE RATIO/D6.2 = LISTPR/WHOLESALEPR;
```

```
COMPUTE RATIO1/D6.2 = LISTPR/WHOLESALEPR;
BY DIRECTOR
BY RATING
BY COPIES
WHERE COPIES LT 3
  WHERE DIRECTOR EQ 'KAZAN E.'
  ON RATING  RECOMPUTE  AVE.LISTPR  RATIO AS '*REC:  '
  ON COPIES  SUB-TOTAL                    AS '*SUB:  '
  ON TABLE NOTOTAL
END
```

下図は、出力結果を示しています。

- RECOMPUTE コマンドで指定されていないフィールドに SUB-TOTAL コマンドが継承されるため、「*REC」というラベルの付いた行の WHOLESALEPR と RATIO1 の値は、中間合計になります。
- LISTPR フィールドおよび RATIO フィールドは、RECOMPUTE コマンドで指定されているため、LISTPR の値は平均値で、RATIO の値 (RATIO1 と同一の定義) は再計算されます。
- SUB-TOTAL コマンドは、DIRECTOR の行に継承されます。

出力結果は次のとおりです。

```
DIRECTOR           RATING  COPIES  LISTPR  WHOLESALEPR   RATIO  RATIO1
--------           ------  ------  ------  -----------   -----  ------
KAZAN E.           NR           1   24.98        14.99    1.67    1.67

*SUB:    1                        24.98        14.99    1.67    1.67

                                2   19.95         9.99    2.00    2.00

*SUB:    2                        19.95         9.99    2.00    2.00
*REC:   NR                        22.46        24.98     .90    3.66
*TOTAL DIRECTOR KAZAN E.          44.93        24.98    3.66    3.66
```

フィールドリストを含む集計コマンドの継承

次のリクエストでは、RECOMPUTE コマンドにはフィールドリストが指定されています。

```
TABLE FILE MOVIES
PRINT LISTPR WHOLESALEPR
COMPUTE RATIO/D6.2 = LISTPR/WHOLESALEPR;
COMPUTE RATIO1/D6.2 = LISTPR/WHOLESALEPR;
BY DIRECTOR
BY RATING
BY COPIES
WHERE COPIES LT 3
  WHERE DIRECTOR EQ 'KAZAN E.'
  ON RATING RECOMPUTE LISTPR RATIO AS '*REC:  '
  ON COPIES SUB-TOTAL AS '*SUB:  '
END
```

値が生成されないすべての列にも SUB-TOTAL が継承されます。総合計行は、フィールドリストで指定されたフィールドに対して RECOMPUTE コマンドを継承し、その他の列には SUB-TOTAL コマンドが継承されます。

```
DIRECTOR           RATING  COPIES  LISTPR  WHOLESALEPR   RATIO  RATIO1
--------           ------  ------  ------  -----------   -----  ------
KAZAN E.           NR           1   24.98        14.99    1.67    1.67

*SUB:     1                       24.98        14.99    1.67    1.67

                                2   19.95         9.99    2.00    2.00

*SUB:     2                       19.95         9.99    2.00    2.00
*REC:   NR                        44.93        24.98    1.80    3.66
*TOTAL DIRECTOR KAZAN E.          44.93        24.98    3.66    3.66


TOTAL                             44.93        24.98    1.80    3.66
```

集計コマンドの組み合わせ時の注意

- SET SUMMARYLINES=EXPLICIT は、集計コマンドを総合計行に継承させる方法を、ソート区切りでの動作と一致させます。そのため、この設定を有効にすると、SUB-TOTAL と SUMMARIZE は総合計行に継承されますが、SUBTOTAL と RECOMPUTE は継承されません。

ACROSS フィールド集計列の生成

SUBTOTAL、SUB-TOTAL、SUMMARIZE、RECOMPUTE コマンドは、横方向ソート区切りに使用することができます。

リクエストに複数の表示フィールドと 1 つの ACROSS フィールドが含まれている場合、レポート出力には ACROSS 値ごとに複数のフィールドが表示されます。集計コマンドを各 ACROSS 値の特定のフィールドには適用するが、他のフィールドには適用しない場合、集計するフィールド名のみを指定することができます。この方法は、リクエストに OVER 句を使用して各フィールドをそれぞれ別の行に配置する際に役立ちます。

ACROSS フィールドの集計演算の実行

```
{ACROSS|ON} acrossfield [AS 'text1'] sumoption [AS 'text2']
             [COLUMNS c1 [AND c2 ...]]
```

または

```
ACROSS acrossfieldsumoption [field1field2 ... fieldn]
```

または

```
ACROSS acrossfield
```

```
ON acrossfieldsumoption [field1field2 ... fieldn]
```

説明

acrossfield

集計演算を実行する ACROSS フィールドです。集計演算は ACROSS フィールド値の最後に追加されます。

sumoption

次のいずれかの値です。SUBTOTAL、SUB-TOTAL、RECOMPUTE、SUMMARIZE のいずれかです。

'text1'

レポート出力の区切りフィールドに使用するフィールド見出しです。

'text2'

集計行の上部に表示するテキストです。

COLUMNSc1, c2 ...

レポート出力に特定の ACROSS 値を、指定の順序で一覧表示します。これらの値を ON 句で指定することはできません。ACROSS 句に指定する場合、ACROSS 句の末尾のオプションとして指定する必要があります。

field1 field2 ... fieldn

集計コマンドを適用するフィールドです。フィールド名を指定しない場合、すべてのフィールドが集計されます。

ACROSS フィールド集計の使用上の注意

- SUMMARIZE および SUB-TOTAL は、指定された ACROSS フィールドと、その上位の ACROSS フィールドの演算を実行します。BY フィールドの演算は実行しません。SUBTOTAL および RECOMPUTE は、指定された ACROSS フィールドのみの演算を実行します。ただし、上位の ACROSS フィールドの値が変わるまで集計は生成されません。
- BY フィールドに指定した SUMMARIZE および SUB-TOTAL コマンドは、BY フィールドと、その上位の BY フィールドの演算を実行します。ACROSS フィールドの演算は実行しません。
- ROW-TOTAL、ACROSS-TOTAL、SUBTOTAL、SUB-TOTAL は列の値を集計します。これらは、SUMMARIZE および RECOMPUTE とは異なり、集計以外の演算を再適用することはありません。
- ON TABLE 句で指定される集計コマンドは、行ではなく列の演算を実行します。
- ACROSS の場合、集計列は ACROSS グループの最後にのみ表示されます (上位の ACROSS フィールドの値が変化した時点)。
- 同一のソート区切りに 2 つの ON 句を使用してそれぞれ異なる演算を実行すると、演算結果が同一の集計列に表示されます。これにより、集計列に複数の演算結果を表示することが可能になります。

- 同一のソート区切りに複数の ON 句を使用して同一のフィールドを参照した場合、最後に指定された集計コマンドが適用されます。
- 各ソート区切りでそれぞれ異なる集計演算を指定することができます。
- SUMMARYLINES パラメータは ACROSS フィールドの処理に影響しません。
- OVER とともに使用すると、集計しないフィールドの行はブランクになります。
- 集計行では次の演算接頭語がサポートされます。
    - 数値フィールドでサポートされる演算接頭語は、ASQ.、AVE.、CNT.、FST.、LST.、MAX.、MIN.、SUM. です。
    - 演算接頭語の PCT.、RPCT.、TOT. はサポートされません。
    - 二重演算接頭語 (例、PCT.CNT.) はサポートされません。
    - SUM. 演算接頭語は、演算接頭語のない集計句と同一の集計値を生成します。
    - SUMMARIZE および RECOMPUTE は、それに関連する COMPUTE コマンドで定義された計算を集計値に適用します。そのため、必要な計算を実行するには、SUMMARIZE または RECOMPUTE コマンドに、COMPUTE コマンドで参照されているフィールドをすべて指定する必要があります。
    - 1 つのフィールドに対して複数の集計演算をそれぞれ別のレベルで実行する場合、指定した演算接頭語の計算が各レベルで実行されます。
    - SUB-TOTAL および SUMMARIZE は、上位のソートフィールドすべてに演算を継承します。リクエストで SUB-TOTAL または SUMMARIZE を複数のソートレベルで使用する場合、複数の演算接頭語が 1 つのフィールドに適用されることがあります。下位のソートフィールドで実行する SUB-TOTAL または SUMMARIZE コマンドが上位のソートフィールドに継承される場合、そのコマンドの演算接頭語は、上位で演算接頭語が指定されていないフィールドのみに適用されます。上位のフィールドで演算接頭語が指定されていた場合、元の演算接頭語が、最初に指定されたレベルで適用されます。
    - コマンドが RECOMPUTE と SUMMARIZE の場合でも、SUBTOTAL と SUB-TOTAL の場合でも、集計行の演算接頭語は同一の値を生成します。一時項目 (COMPUTE) の場合、演算接頭語は適用されず、COMPUTE コマンドの式および集計行の値を使用して値が再計算されます。
    - ACROSS フィールドに ACROSS-TOTAL 句および演算接頭語の集計コマンドが指定されている場合、演算接頭語は適用されますが、ACROSS-TOTAL は適用されません。

ACROSS での集計コマンドの使用

次のリクエストは、UNITS および DOLLARS を集計し、PRODUCT、REGION、MONTH ごとの UNIT COST を計算します。ACROSS MNTH RECOMPUTE コマンドは、UNITS および DOLLARS を集計し、選択した REGION と MONTH の値を再計算します。ACROSS REGION RECOMPUTE コマンドも、選択した地域の値を同様に計算します。ON TABLE SUMMARIZE コマンドは集計行を生成します。集計列は生成されません。

```
DEFINE FILE GGSALES
MNTH/MTr   = DATE;
END
TABLE FILE GGSALES
SUM
 UNITS/I5 AS 'UNITS'                   OVER
 DOLLARS/I6 AS 'DOLLARS'               OVER
 COMPUTE DOLLPER/I6 = DOLLARS/UNITS; AS 'UNIT COST'
BY PRODUCT
ACROSS REGION RECOMPUTE AS 'Region Sum' COLUMNS 'Northeast' AND 'West'
ACROSS MNTH   RECOMPUTE AS 'Month Sum' COLUMNS 'November' AND 'December'
WHERE DATE FROM '19971101' TO '19971231';
WHERE PRODUCT EQ 'Capuccino' OR 'Espresso';
ON TABLE SUMMARIZE AS 'Grand Total'

END
```

出力結果は次のとおりです。

PAGE 1								
		Region						
		Northeast			West			Region Sum
		MNTH						
		NOVEMBER	DECEMBER	Month Sum	NOVEMBER	DECEMBER	Month Sum	
Product								
Capuccino	UNITS	2282	1188	3470	2535	4051	6586	10056
	DOLLARS	25994	13668	39662	31153	57421	88574	128236
	UNIT COST	11	11	11	12	14	13	12
Espresso	UNITS	1947	2403	4350	3088	3732	6820	11170
	DOLLARS	23629	30605	54234	36123	51400	87523	141757
	UNIT COST	12	12	12	11	13	12	12
Grand Total	UNITS	4229	3591	7820	5623	7783	13406	21226
	DOLLARS	49623	44273	93896	67276	108821	176097	269993
	UNIT COST	11	12	12	11	13	13	12

ACROSS グループ内の単一フィールドの中間合計

次のリクエストは、GGSALES データソースに対して実行され、DOLLARS および UNITS フィールドを集計し、CATEGORY フィールドを BY で、REGION フィールドを ACROSS でソートします。ただし、中間合計は UNITS フィールドのみで計算します。

```
TABLE FILE GGSALES
SUM DOLLARS AS 'Dollars' OVER
UNITS AS 'Units'
  BY CATEGORY
  ACROSS REGION SUBTOTAL UNITS
WHERE REGION EQ 'Midwest' OR 'West'
ON TABLE SET PAGE NOPAGE
END
```

出力結果では、UNITS 値の行のみで中間合計が計算されています。

```
                     Region
                     Midwest      West         TOTAL
Category
---------------------------------------------------------------
Coffee       Dollars   4178513      4473517
```

```
             Units      332777       356763       689540
Food         Dollars   4338271      4202337
             Units      341414       340234       681648
Gifts        Dollars   2883881      2977092
             Units      230854       235042       465896
```

ACROSS グループでの一時項目 (COMPUTE) 値の集計

次のリクエストは、GGSALES データソースに対して実行され、DOLLARS および UNITS フィールドを集計し、DOLLARS PER UNIT を計算して、REGION フィールドを ACROSS でソートします。このリクエストでは、上位の ACROSS フィールドである CATEGORY が指定されているため、SUMMARIZE コマンドは両方の ACROSS フィールドに継承されます。

```
SET BYPANEL = ON
TABLE FILE GGSALES
SUM DOLLARS AS 'Dollars' OVER
UNITS AS 'Units'         OVER
AND COMPUTE DPERU/D9.2 = DOLLARS/UNITS;
  ACROSS CATEGORY
  ACROSS REGION
  ON REGION SUMMARIZE DPERU
  WHERE REGION EQ 'Midwest' OR 'West'
  WHERE CATEGORY EQ 'Food' OR 'Gifts'
  ON TABLE PCHOLD FORMAT PDF
END
```

出力結果の最初のページには、次の情報が表示されます。

- Food カテゴリでの Midwest および West 地域の DOLLARS、UNITS、DPERU の値。
- 集計列での DPERU 行のみの集計値。ACROSS の場合、REGION の集計列は、上位の ACROSS フィールドである CATEGORY の値が変化した後にのみ表示されます。
- Gifts カテゴリでの Midwest および West 地域の DOLLARS、UNITS、DPERU の値。

```
PAGE   1.1


         Category
         Food                                    Gifts
```

```
         Region
         Midwest      West         TOTAL        Midwest      West
-------------------------------------------------------------------------
Dollars      4338271      4202337                   2883881      2977092
Units         341414       340234                    230854       235042
DPERU          12.71        12.35        12.53        12.49        12.67
```

2 ページ目には、Gifts カテゴリの合計列、および総合計列が表示されます。それぞれの列には、DPERU 行の値のみが表示されます。

```
 PAGE   1.2


         Category
                       TOTAL
         Region
          TOTAL
 ----------------------------------
 Dollars
 units
 DPERU          12.58        12.55
```

集計コマンドの演算接頭語と ACROSS の使用

次のリクエストは、GGSALES データソースに対して実行され、DOLLARS および UNITS フィールドを集計し、CATEGORY および REGION フィールドを ACROSS でソートします。また、REGION フィールドに SUMMARIZE コマンドを使用します。このリクエストでは、上位の ACROSS フィールドである CATEGORY が指定されているため、SUMMARIZE コマンドは両方の ACROSS フィールドに継承されます。SUMMARIZE コマンドでは、AVE. 演算接頭語が DOLLARS フィールドに対して指定されています。

```
SET BYPANEL = ON
TABLE FILE GGSALES
SUM DOLLARS AS 'Dollars' OVER
UNITS AS 'Units'
  ACROSS CATEGORY
  ACROSS REGION
  ON REGION SUMMARIZE AVE.DOLLARS
  WHERE REGION EQ 'Midwest' OR 'West'
  WHERE CATEGORY EQ 'Food' OR 'Gifts'
```

```
  ON TABLE PCHOLD FORMAT PDF
END
```

出力結果の最初のページには、次の情報が表示されます。

- Food カテゴリでの Midwest および West 地域の DOLLARS および UNITS の値。
- 集計列での DOLLARS 行のみの集計値。ACROSS の場合、REGION の集計列は、上位の ACROSS フィールドである CATEGORY の値が変化した後にのみ表示されます。
- Gifts カテゴリでの Midwest および West 地域の DOLLARS および UNITS の値。

```
PAGE   1.1


          Category
          Food                                    Gifts
          Region
          Midwest      West          TOTAL        Midwest      West
---------------------------------------------------------------------------
Dollars    4338271      4202337      4270304      2883881      2977092
Units       341414       340234                    230854       235042
```

2 ページ目には、Gifts カテゴリの合計列、および総合計列が表示されます。それぞれの列には、DOLLARS 行の値のみが表示されます。

```
PAGE   1.2


         Category
                       TOTAL
         Region
          TOTAL
---------------------------------
Dollars    2930486      3600395
units
```

ACROSS 集計コマンドの組み合わせの使用

次のリクエストは、GGSALES データソースに対して実行され、DOLLARS および UNITS フィールドを集計し、CATEGORY および REGION フィールドを ACROSS でソートします。また、REGION フィールドに SUMMARIZE コマンド、CATEGORY フィールドに SUBTOTAL コマンドを使用します。SUMMARIZE コマンドでは、DOLLARS の平均値および UNITS の最

小値が指定されています。SUBTOTAL コマンドでは、DOLLARS の最小値が指定されています。

```
SET BYPANEL = ON
TABLE FILE GGSALES
SUM DOLLARS AS 'Dollars' OVER
UNITS AS 'Units'
  ACROSS CATEGORY
  ACROSS REGION
    ON CATEGORY SUBTOTAL MIN.DOLLARS
    ON REGION SUMMARIZE AVE.DOLLARS MIN.UNITS
  WHERE REGION EQ 'Midwest' OR 'West'
  WHERE CATEGORY EQ 'Food' OR 'Gifts'
  ON TABLE PCHOLD FORMAT PDF
END
```

出力結果では、すべての TOTAL 列に UNITS の最小値が表示されます。REGION ソートフィールドに関連する TOTAL 列には DOLLARS の平均値が表示されますが、CATEGORY ソートフィールドに関連する TOTAL 列には DOLLARS の最小値が表示されます。これは、上位のソートフィールドに関連付けられた演算接頭語が SUMMARIZE コマンドで変更されていないためです。

```
PAGE   1.1


          Category
          Food                                    Gifts
          Region
          Midwest      West         TOTAL        Midwest      West
---------------------------------------------------------------------------
Dollars    4338271      4202337      4270304      2883881      2977092
Units       341414       340234       340234       230854       235042
```

```
PAGE   1.2


         Category
                       TOTAL
         Region
          TOTAL
-----------------------------------
Dollars    2930486      2883881
Units       230854       230854
```

ソートフィールド区切りでの計算実行

RECAP および COMPUTE コマンドを使用して、計算時に中間合計の値を生成することができます。中間合計値は表示されません。計算結果のみがレポートに表示されます。

計算での中間合計の使用

RECAP および COMPUTE コマンドには、合計および中間合計を実行するコマンドに類似した構文があります。

```
{BY|ON} fieldname1 {RECAP|COMPUTE} fieldname2[/format] = expression;
                                        [WHEN expression;]
```

説明

fieldname1

BY 句で指定したフィールド名です。BY フィールドの値が変更されるたびに、新しい RECAP 値が計算されます。

fieldname2

式の結果を格納するフィールド名です。

/format

任意の有効フォーマットにすることができます。デフォルト値は D12.2 です。

expression

有効な任意の式にすることができます。詳細は、式の使用を参照してください。式の末尾にセミコロン (;) を追加する必要があります。

WHEN expression

RECAP との使用に限定されます。この句は、ブール式により決定される RECAP 行の条件付き表示を指定します。詳細は、集計行とテキストの条件付き表示を参照してください。式の末尾にセミコロン (;) を追加する必要があります。

RECAP および COMPUTE の使用上の注意

- RECAP が使用するものには、指定されたソートフィールドの現在値、表示フィールドとして表示される計算フィールドの現在の中間合計値、文字フィールドの最後の値があります。
- RECAP を使用すると、ページ下部にブランク領域が確保されるため、RECAP の計算対象となるデータが前ページに表示され、RECAP の計算結果のみが次ページの上部に単独で表示されることがなくなります。これと同一の手法が中間合計および総合計にも使用されますが、中間見出しおよび一時項目 (COMPUTE) には使用されません。
- 式に使用するフィールド名は、レポートに表示されるフィールドにする必要があります。つまり、表示フィールドまたはソート制御フィールドでなければなりません。
- RECAP の値は、それぞれ別の行に表示されます。ただし、リクエストに RECAP コマンドおよび SUBFOOT テキストが指定されている場合は、RECAP 値は SUBFOOT テキスト内にのみ表示されるため、スポットマーカーを使用してテキスト内で指定する必要があります。詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。
- RECAP および COMPUTE での計算は、テキストとともに制御区切り内の任意の位置に配置することができます。
- ON 句では、COMPUTE コマンドは RECAP コマンドと同様に機能します。
- ON *sortfield* RECAP 句の最大長は、各ソートフィールドで 64 です。
- 1 つの RECAP 句で複数の RECAP 計算を指定することができます。使用する構文は次のとおりです。

```
ON sortfield RECAP field1/format= ... ;field2/format= ... ;
.
.
.
```

RECAP の使用

次のリクエストでは、RECAP (DEPT_NET) を使用して、部署別の純利益を算出します。

```
TABLE FILE EMPLOYEE
SUM DED_AMT AND GROSS
```

```
BY DEPARTMENT BY PAY_DATE
ON DEPARTMENT RECAP DEPT_NET/D8.2M = GROSS-DED_AMT;
WHEN PAY_DATE GT 820101
END
```

出力結果は次のとおりです。

```
DEPARTMENT  PAY_DATE          DED_AMT            GROSS
----------  --------          -------            -----
MIS         81/11/30        $1,406.79        $2,147.75
            81/12/31        $1,406.79        $2,147.75
            82/01/29        $1,740.89        $3,247.75
            82/02/26        $1,740.89        $3,247.75
            82/03/31        $1,740.89        $3,247.75
            82/04/30        $3,386.73        $5,890.84
            82/05/28        $3,954.35        $6,649.50
            82/06/30        $4,117.03        $7,460.00
            82/07/30        $4,117.03        $7,460.00
            82/08/31        $4,575.72        $9,000.00

** DEPT_NET           $22,311.98

PRODUCTION  81/11/30          $141.66          $833.33
            81/12/31          $141.66          $833.33
            82/01/29        $1,560.09        $3,705.84
            82/02/26        $2,061.69        $4,959.84
            82/03/31        $2,061.69        $4,959.84
            82/04/30        $2,061.69        $4,959.84
            82/05/28        $3,483.88        $7,048.84
            82/06/30        $3,483.88        $7,048.84
            82/07/30        $3,483.88        $7,048.84
            82/08/31        $4,911.12        $9,523.84

** DEPT_NET           $27,531.14
```

複数 RECAP コマンドの使用

1 つのリクエストに複数の RECAP または COMPUTE コマンドを使用することができます。このオプションを使用すると、異なる制御区切りで異なる計算を実行することができます。

次のリクエストでは、複数の RECAP コマンドを使用します。

```
TABLE FILE SALES
SUM UNIT_SOLD AND RETURNS
WHERE AREA EQ 'U'
BY DATE BY AREA BY PROD_CODE
ON DATE RECAP
DATE_RATIO=RETURNS/UNIT_SOLD;
ON AREA UNDER-LINE RECAP
AREA_RATIO=RETURNS/UNIT_SOLD;
END
```

出力結果は次のとおりです。

```
DATE   AREA  PROD_CODE  UNIT_SOLD  RETURNS
----   ----  ---------  ---------  -------
10/17  U     B10               30        2
             B17               20        2
             B20               15        0
             C17               12        0
             D12               20        3
             E1                30        4
             E3                35        4

** AREA_RATIO                        .09

** DATE_RATIO                        .09

----------------------------------------
10/18  U     B10               13        1

** AREA_RATIO                        .08

** DATE_RATIO                        .08

----------------------------------------
10/19  U     B12               29        1

** AREA_RATIO                        .03

** DATE_RATIO                        .03

----------------------------------------
```

総合計の非表示

NOTOTAL コマンドを使用して、レポートの総合計を非表示にすることができます。

ソート区切りの値が 1 つの場合は、この値と総合計の値が等しくなるため、総合計を非表示にすると便利です。NOTOTAL コマンドを使用して、ソート区切りに詳細行が 1 つしか存在しない場合に、総合計行をレポートに表示しないようにします。MULTILINES コマンドを使用して、中間合計を非表示にすることもできます。

総合計の非表示

総合計を非表示にするには、リクエストに次の構文を追加します。

```
ON TABLE NOTOTAL
```

総合計の非表示

次のリクエストでは、NOTOTAL 句を使用して、CURR_SAL、GROSS、DED_AMT の総合計を非表示にします。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL AND GROSS AND DED_AMT
BY EMP_ID
BY BANK_ACCT
WHERE BANK_ACCT NE 0
ON BANK_ACCT SUB-TOTAL
ON TABLE NOTOTAL
END
```

出力結果は次のとおりです。

EMP_ID	BANK_ACCT	CURR_SAL	GROSS	DED_AMT
117593129	40950036	$18,480.00	$6,099.50	$2,866.18
*TOTAL	40950036	$18,480.00	$6,099.50	$2,866.18
*TOTAL	117593129	$18,480.00	$6,099.50	$2,866.18
119329144	160633	$29,700.00	$2,475.00	$1,427.24
*TOTAL	160633	$29,700.00	$2,475.00	$1,427.24
*TOTAL	119329144	$29,700.00	$2,475.00	$1,427.24
123764317	819000702	$26,862.00	$17,094.00	$11,949.44
*TOTAL	819000702	$26,862.00	$17,094.00	$11,949.44
*TOTAL	123764317	$26,862.00	$17,094.00	$11,949.44
326179357	122850108	$21,780.00	$9,075.00	$6,307.00
*TOTAL	122850108	$21,780.00	$9,075.00	$6,307.00
*TOTAL	326179357	$21,780.00	$9,075.00	$6,307.00
451123478	136500120	$16,100.00	$9,130.00	$3,593.92
*TOTAL	136500120	$16,100.00	$9,130.00	$3,593.92
*TOTAL	451123478	$16,100.00	$9,130.00	$3,593.92
818692173	163800144	$27,062.00	$22,013.75	$15,377.40
*TOTAL	163800144	$27,062.00	$22,013.75	$15,377.40
*TOTAL	818692173	$27,062.00	$22,013.75	$15,377.40

集計行とテキストの条件付き表示

集計行を使用してレポートの外観および内容を制御する以外に、WHEN 条件を指定して、BY フィールド値に対してどの集計行を表示するかの条件を制御することができます。WHEN は、SUBFOOT、SUBHEAD、SUBTOTAL、SUB-TOTAL、SUMMARIZE、RECOMPUTE、RECAP でサポートされます。

集計行とテキストの条件付き表示

たとえば、4 つの地域 (Midwest、Northeast、Southeast、West) を対象とした売上レポートを作成する場合、総売上高が $11,500,000 を超える場合のみ中間合計を表示したい場合が

あります。次のリクエストでは、売上が $11,500,000 を超える場合に中間合計を表示し、それ以下の場合は中間脚注のテキストを表示するという条件を指定します。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY REGION
BY CATEGORY
ON REGION SUBTOTAL
WHEN DOLLARS GT 11500000
SUBFOOT
"The total for the <REGION region is less than 11500000."
WHEN DOLLARS LT 11500000
END
```

出力結果は次のとおりです。

```
Region       Category     Unit Sales  Dollar Sales
------       --------     ----------  ------------
Midwest      Coffee           332777       4178513
             Food             341414       4338271
             Gifts            230854       2883881

The total for the Midwest region is less than 11500000.
Northeast    Coffee           335778       4164017
             Food             353368       4379994
             Gifts            227529       2848289

The total for the Northeast region is less than 11500000.
Southeast    Coffee           350948       4415408
             Food             349829       4308731
             Gifts            234455       2986240

*TOTAL Southeast              935232      11710379

West         Coffee           356763       4473517
             Food             340234       4202337
             Gifts            235042       2977092

*TOTAL West                   932039      11652946


TOTAL                        3688991      46156290
```

式の使用

式は、演算子、フィールド名、定数で構成され、計算を実行して 1 つの値を返します。さまざまなコマンドに式を使用して、一時項目またはダイアログマネージャ変数に値を割り当てたり、選別に使用したりします。単純な式を組み合わせて複雑な式を構築することができます。

式を作成する場合は、ユーザが独自に式を指定することもできますが、特定の計算や操作を行うために用意されたさまざまな関数の中から必要なものを選択することもできます。これらの関数は、1 つまたは複数の引数をとり、結果として 1 つの値を返します。関数は単に呼び出すだけで使用することができます。

演算式、文字式、論理式などの式の一部として IF-THEN-ELSE ロジックがサポートされます。IF-THEN-ELSE ロジックは、WHERE 句でもサポートされます。

コマンドおよび句での式の使用

さまざまなコマンドおよび句で式を使用することができます。式は 40 行以内で記述し、行の末尾にセミコロン (;) を配置します。ただし、WHERE および WHEN 句では、セミコロン (;) はオプションとして使用します。

ここでは、式をサポートするコマンドおよびその基本的な構文について概説します。構文についての詳細は、対応するマニュアルを参照してください。

次のような状況で式を使用することができます。

- 一時項目を作成し、そのフィールドに値を割り当てる。このフィールドは、マスターファイルで DEFINE 属性を使用したり、DEFINE または COMPUTE コマンドを使用したりして作成することができます。
    - DEFINE コマンドをリクエストの前に記述する。

```
DEFINE FILE filename
 fieldname  [/format] = expression;
   .
   .
```

```
   .
END
```

  - DEFINE 属性をマスターファイルに記述する。

```
DEFINE fieldname [/format] = expression;$
```

  - COMPUTE コマンドをリクエストに記述する。

```
COMPUTE fieldname  [/format] = expression;
```

- レコードの選択条件およびレポートのフォーマットを制御する条件を定義する。

```
{WHERE|IF} logical_expression[;]
      WHEN logical_expression[;]
```

- ダイアログマネージャでの分枝方法を決定したり、ダイアログマネージャ変数に値を割り当てたりする。

```
-IF logical_expression [THEN] GOTO label1 [ELSE GOTO label2];
```

```
-SET &name = expression;
```

- FML (Financial Modeling Language) で、RECAP コマンドを使用して計算を実行する。

```
RECAP name [(n)] [/format] = expression;
```

式のタイプ

次のいずれかのタイプの式を使用することができます。

- **数値**　数値式を使用して、数値定数 (整数または小数) とフィールドで構成される計算を実行します。たとえば、次のような式を記述して、各従業員の現在の給与額に所定の比率を乗算し、従業員賞与額を計算することができます。

```
COMPUTE BONUS/D12.2 = CURR_SAL * 0.05 ;
```

数値式は数値を返します。詳細は、数値式の作成を参照してください。

- **日付**　日付式を使用して、日付に関する数値計算を実行します。たとえば、次のような式を記述して、顧客が品物を注文した場合に、品物の発送日に配達日数を加算して顧客が品物を受け取る日を計算することができます。

  ```
  COMPUTE DELIVERY/MDY = SHIPDATE + 5 ;
  ```

  日付を計算する式には、次の 2 種類があります。

  - 日付式は、特定の日付、日付の構成要素、2 つの日付データ間の日数、月数、四半期数、年数を表す整数を返します。詳細は、日付式の作成を参照してください。
  - 日付時間式は、さまざまな日付時間関数を使用して作成し、それぞれ異なる値を返します。関数についての詳細は、『ibi™ WebFOCUS® 関数リファレンス』を参照してください。

- **文字**　文字式を使用して、文字定数または文字フィールドを操作します。たとえば、次のような式を記述して、文字フィールドから名前のイニシャルを抽出することができます。

  ```
  COMPUTE FIRST_INIT/A1 = EDIT (FIRST_NAME, '9$$$$$$$$$') ;
  ```

  文字式は文字の値を返します。詳細は、文字式の作成を参照してください。

  注意：テキストフィールドを文字フィールドに割り当てることで、文字フィールドに割り当てられた値をテキストフィールドに渡すことができます。テキストフィールドは、CONTAINS および OMITS 演算子を使用して式に含めたり、式から除外したりすることもできます。

- **論理**　論理式を使用して、2 つの値の関係を評価します。論理式は TRUE または FALSE を返します。詳細は、論理式の作成を参照してください。

- **条件**　条件式を使用して、論理式の結果に基づいて値を割り当てます。条件式 (IF ... THEN ... ELSE) は、数値または文字の値を返します。詳細は、条件式の作成を参照してください。

式とフィールドフォーマット

式を使用してフィールドに値を割り当てる場合、そのフィールドに設定したフォーマットと式が返す値のフォーマットが一致するようにします。たとえば、文字式を使用して氏名の姓と名を連結し、それを FULL_NAME フィールドに割り当てる場合、そのフィールドを文字として定義しておく必要があります。.

フィールドフォーマットの長さの確保

次の例では、文字式を使用して姓と名を連結し、氏名 (フルネーム) を出力します。ここでは、FULL_NAME フィールドに十分な長さの文字フォーマットを割り当てて、連結した氏名が正しく収まるようにします。

```
DEFINE FILE EMPLOYEE
FULL_NAME/A25 = FIRST_NAME | LAST_NAME;
END
TABLE FILE EMPLOYEE
PRINT FULL_NAME
WHERE LAST_NAME IS 'BLACKWOOD'
END
```

出力結果は次のとおりです。

```
FULL_NAME
---------
ROSEMARIE BLACKWOOD
```

数値式の作成

数値式は、数値定数、フィールド、演算子、関数を使用して計算を実行し、その結果として数値を返します。数値式を使用してフィールドに値を割り当てる場合、そのフィールドを数値フォーマットとして設定しておく必要があります。デフォルトフォーマットは D12.2 です。

数値式には次のような要素があり、次の例ではそれぞれの要素が太字で表示されています。

- 数値定数。以下はその例です。

```
COMPUTE COUNT/I2 = 1 ;
```

- 指数表示の数値定数。以下はその例です。

  ```
  COMPUTE COST/D12.2 = EXPN(8E+3);
  ```

  構文の使用方法についての詳細は、数値式の作成を参照してください。

- 数値フィールド。以下はその例です。

  ```
  COMPUTE RECOUNT/I2 = COUNT ;
  ```

- 2 つの数値定数またはフィールド。これらは、数値演算子で結合します。以下はその例です。

  ```
  COMPUTE BONUS/D12.2 = CURR_SAL * 0.05 ;
  ```

  数値演算子のリストは、数値式の作成を参照してください。

- 数値関数。以下はその例です。

  ```
  COMPUTE LONGEST_SIDE/D12.2 = MAX (WIDTH, HEIGHT) ;
  ```

- 複数の数値式。これらは、数値演算子で結合します。以下はその例です。

  ```
  COMPUTE PROFIT/D12.2 = (RETAIL_PRICE - UNIT_COST) * UNIT_SOLD ;
  ```

  なお、括弧を使用して、式の評価順序を変更することができます。数値演算の実行順序についての詳細は、数値式の作成を参照してください。

通常、数値は倍精度浮動小数点数フォーマットに変換されてから計算に使用されます。さらに、計算結果は指定されたフィールドフォーマットに変換されます。変換時の端数処理で、計算結果が若干異なる場合があります。なお、ネイティブモードの計算をサポートする環境では、端数処理の方法が異なります。

数字が大きすぎる場合 (10^{75} を超える) または小さすぎる場合 (10^{-75} 未満) は、オーバーフローまたはアンダーフローの警告が表示されるとともに、フィールド値にアスタリスク (*) が表示されます。

注意：オーバーフロー文字は、SET OVERFLOWCHAR コマンドを発行して変更することができます。

IF-THEN-ELSE ロジックは、数値式でサポートされます。

指数表記での数値の表示

IF 句には次の式を使用します。

```
IF fieldopn[.nn]{E|D|e|d}[±|-]p
```

WHERE 句には次の式を使用します。

```
WHERE fieldop EXPN(n[.nn{E|D|e|d}[±|-]p);
```

COMPUTE コマンドには次の式を使用します。

```
COMPUTE field[/format] = EXPN(n[.nn]{{E|D|e|d}[±|-]p);
```

DEFINE コマンドには次の式を使用します。

```
DEFINE FILE filename
field[/format] = EXPN(n[.nn]{E|D|e|d}[±|-]p);
END
```

マスターファイルの DEFINE には次の式を使用します。

```
DEFINE field[/format] = EXPN(n[.nn]{{E|D|e|d}[±|-]p);
```

説明

field

リクエストで指定するフィールドです。

/format

オプションとして指定するフィールドのフォーマットです。

op

リクエストで指定する関係演算子です。

n.nn

数値定数です。この定数は、整数、小数点、小数部分の順序で構成されます。

E, D, e, d

指数表記を表します。E と D は、相互に交換可能です。

+, -

p の正負を表します。デフォルトは正です。

p

10 の乗数です。桁数を増減します。p の値の範囲は -78 から +78 までです。

注意：F および D フォーマットのフィールドで計算を行う場合は、EXPN を使用すると便利です。一般に、P または I フォーマットのフィールドで計算を行う場合には適していません。

指数表記での数値の評価

IF または WHERE 句に指数表記を使用して、たとえば 8000 を 8E+03 のように表現することができます。

```
IF RCOST LT 8E+03
```

```
WHERE RCOST LT EXPN(8E+03)
```

算術演算子

式に使用可能な算術演算子には次のものがあります。

加算	+
減算	-
乗算	*
除算	/
指数演算	**

注意：除算の分母に 0 (ゼロ) を使用すると、その式の値は 0 (ゼロ) になります。すべてのタイプの日付式で、乗算および指数演算を使用することはできません。日付の一部を分離するには、単純な割り当てコマンドを使用します。

評価の順序

数値式は、次の順序で評価されます。

1. 指数演算
2. 除算および乗算
3. 加算および減算

演算子が同レベルの場合は、左から右の順序で評価されます。括弧内の式が他の式より先に評価されるため、括弧を使用することにより定義された順序を変更することができます。たとえば、次の式で括弧を使用すると、異なる結果が得られます。

```
COMPUTE PROFIT/D12.2  =  RETAIL_PRICE  -  UNIT_COST  *  UNIT_SOLD ;
COMPUTE PROFIT/D12.2  = (RETAIL_PRICE  -  UNIT_COST) *  UNIT_SOLD ;
```

1 つ目の式では、最初に UNIT_SOLD と UNIT_COST を乗算し、その結果を RETAIL_PRICE から減算します。2 つ目の式では、最初に UNIT_COST を RETAIL_PRICE から減算し、その結果に UNIT_SOLD を乗算します。

注意：2 つの演算子を連続して記述することはできません。次の式は無効です。

```
a * -1
```

この式を有効にするには、括弧を使用します。

```
a* (-1)
```

評価順序の変更

式を評価する順序は、その式の結果に影響を与えます。次の例では、ある製品に不良品が発生した場合に、その不良品の返品を含めた小売販売の損失額を割り出します。ここでは、次のリクエストを発行します。

```
TABLE FILE SALES
PRINT RETAIL_PRICE RETURNS DAMAGED
COMPUTE RETAIL_LOSS/D12.2 = RETAIL_PRICE * RETURNS + DAMAGED;
BY PROD_CODE
WHERE PROD_CODE IS 'E1';
END
```

次の式が計算に使用されます。

```
COMPUTE RETAIL_LOSS/D12.2 = RETAIL_PRICE * RETURNS + DAMAGED;
```

この場合、最初に RETAIL_PRICE に RETURNS を乗算し、その結果を DAMAGED に加算するため、誤った結果が得られます。正しい計算結果を得るには、最初に RETURNS を DAMAGED に加算し、その結果に RETAIL_PRICE を乗算する必要があります。

式を括弧で囲むことにより、評価の順序を変更することができます。括弧内の式は、他の式よりも先に評価されます。また、読みやすさを向上するために括弧を使用することもできます。

上記の計算を正しく実行するには、括弧を使用して次のような構文にします。

```
COMPUTE RETAIL_LOSS/D12.2 = RETAIL_PRICE * (RETURNS + DAMAGED);
```

出力結果は次のとおりです。

```
PROD_CODE  RETAIL_PRICE  RETURNS  DAMAGED     RETAIL_LOSS
---------  ------------  -------  -------     -----------
E1                $.89        4        7            9.79
```

演算式での IF-THEN-ELSE ロジックの使用

次のリクエストでは、演算式に IF-THEN-ELSE ロジックを使用して、LISTPR に追加する金額を特定し、NEWPRICE を計算します。

```
TABLE FILE MOVIES
SUM COPIES
LISTPR
COMPUTE
NEWPRICE = LISTPR + (IF COPIES GT 10 THEN 0.00 ELSE 25.00);
BY CATEGORY
```

```
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。個数が 10 を超える場合、NEWPRICE は LISTPR と同一の金額になります。それ以外の場合、NEWPRICE は、LISTPR に $25.00 を加算した金額になります。

CATEGORY	COPIES	LISTPR	WHOLESALEPR	NEWPRICE
ACTION	14	94.82	55.46	94.82
CHILDREN	12	201.67	105.87	201.67
CLASSIC	21	404.71	201.79	404.71
COMEDY	19	154.80	90.45	154.80
DRAMA	2	19.98	10.00	44.98
FOREIGN	6	269.76	146.24	294.76
MUSICALS	5	84.89	52.97	109.89
MYSTERY	21	235.81	116.97	235.81
SCI/FI	7	114.84	79.52	139.84
TRAIN/EX	10	119.87	60.98	144.87

ネイティブモード計算による数値式評価

ネイティブモードの計算を使用する際は、被演算子および演算子のフォーマットにより、数値式ごとに評価手順が異なります。すべての被演算子が同一フォーマットの場合、ほとんどの演算はそのフォーマットで実行されます。被演算子のフォーマットが異なる場合は、特定の優先順位で標準のフォーマットに変換されます。被演算子のフォーマットに関わらず、いくつかの演算子では、すべての被演算子を適切なフォーマットにするため特定のフォーマットへの変換が必要になる場合もあります。

ネイティブモード計算による同一被演算子フォーマットの使用

数値演算子のすべての被演算子が同一フォーマットの場合、下表を参照して、その演算がネイティブフォーマットで実行されるか、演算の実行前後で被演算子のフォーマットが変換されるかを特定することができます。変換が必要な場合、被演算子は演算フォーマットに変換され、中間結果がその演算フォーマットで返されます。結果のフォーマットがターゲット変数のフォーマットと異なる場合は、その結果がターゲット変数のフォーマットに変換されます。

操作		演算フォーマット
加算	+	ネイティブ
減算	-	ネイティブ
乗算	*	ネイティブ
完全な除算	/	単精度または倍精度の浮動小数点数が受容されます。その他すべてのフォーマットは倍精度浮動小数点数に変換されます。
指数演算	**	倍精度浮動小数点数

同一の被演算子フォーマットの使用 (ネイティブモード計算)

次の変数は、整数として定義されています。

```
COMPUTE OPERANDONE/I4  ;
        OPERANDTWO/I4  ;
        RESULT/I4  ;
```

必要となる乗算は、ネイティブモード計算 (整数計算) で実行されます。

```
COMPUTE RESULT/I4 = OPERANDONE * OPERANDTWO  ;
```

ネイティブモード計算による異なる被演算子フォーマットの使用

数値演算子に異なるフォーマットの被演算子を使用する場合、下表を使用して変換後の標準フォーマットを特定することができます。優先順位の低い被演算子は、演算が実行される前に優先度の高い被演算子のフォーマットに変換されます。

順位	フォーマット
1	16 バイトパック 10 進数
2	倍精度浮動小数点数
3	8 バイトパック 10 進数
4	単精度浮動小数点数
5	整数
6	文字 (文字およびテキスト)

たとえば、式に 16 バイトパック 10 進数の被演算子を使用すると、その他すべての被演算子が 16 バイトパック 10 進数フォーマットに変換されて評価が実行されます。一方、式に整数と文字の被演算子のみを使用すると、すべての文字被演算子は整数フォーマットに変換されます。

文字 (文字またはテキスト) の値が数値列の場合は、計算に使用することができます。文字被演算子は、式に使用された他の被演算子のフォーマットへの変換が試みられます。両方の被演算子が文字の場合は、倍精度への変換が試みられます。変換が不可能な場合は、エラーメッセージが表示されます。

整数に小数の値を割り当てると、小数点以下は切り捨てられます。

日付式の作成

日付式は、日付に関する数値計算を実行します。

日付式は、特定の日付、日付の構成要素、2 つの日付データ間の日数、月数、四半期数、年数を表す整数を返します。次の要素で構成される日付式を直接記述することができます。

- 日付定数。以下はその例です。

  ```
  COMPUTE END_DATE/MDYY = 'FEB 29 2000';
  ```

  日付定数は一重引用符 (') で囲む必要があります。

- 日付フィールド。以下はその例です。

  ```
  COMPUTE NEWDATE/YMD = START_DATE;
  ```

- 日付編集オプション付きの文字、整数、パック 10 進数のフィールド。たとえば、2 つ目の COMPUTE コマンドでは、OLDDATE が日付式になります。

  ```
  COMPUTE OLDDATE/I6YMD = 980307;
  COMPUTE NEWDATE/YMD DFC 19 YRT 10 = OLDDATE;
  ```

- 算術演算子または日付関数を使用して日付を返す計算。数値演算子は、日付フォーマット (従来の名称は SmartDate) でのみ使用します。次の例では、最初に整数の日付である HIRE_DATE (I6YMD フォーマット) を日付フォーマットの CONVERTED_HDT (YMD フォーマット) に変換します。次に、CONVERTED_HDT に 30 日を加算します。

  ```
  COMPUTE CONVERTED_HDT/YMD = HIRE_DATE;
  HIRE_DATE_PLUS_THIRTY/YMD = CONVERTED_HDT + 30;
  ```

- 数値演算子または日付関数を使用して 2 つの日付間の日数、月数、四半期数、年数を返す計算。次の例では、日付関数の YMD を使用して、従業員の雇用日と最初の昇給日の差 (昇給までの日数) を計算します。

  ```
  COMPUTE DIFF/I4 = YMD (HIRE_DATE,FST.DAT_INC);
  ```

日付値のフォーマット

日付を処理する場合、次の 2 つの方法があります。

- **日付フォーマット**　この値は、基準日から日付値までの日数を表す整数として処理されます。日付フォーマットには、2 つの基準日があります。
  - YMD および YYMD フォーマットの基準日は 1900/12/31 です。
  - YM および YYM フォーマットの基準日は、Windows および UNIX では 1900/12/31 です。

  整数の値を表示する場合は、フィールドの指定フォーマットに対応する日付に変換されます。このフォーマットは、マスターファイルで指定するか、フィールドに値を割り当てるための式を使用したコマンドで指定することができます。このフォーマットは、以前は「SmartDate が設定されたフォーマットフィールド」と呼ばれていました。
- **日付編集オプション付きの整数、パック 10 進数、文字フォーマット**　この値は、整数、パック 10 進数、文字のいずれかとして処理されます。この値を表示すると、日付フォーマットが設定されます。以前は、このフォーマットは「古い日付フォーマットが設定されたフィールド」と呼ばれていました。

日付のフォーマットを別のフォーマットに変換するには、単純に新しいフォーマットを割り当てます。次の例では、日付編集オプションでフォーマットが設定され、文字フィールドとして保存された日付を、日付変換により一時的な日付フィールドとして保存します。

```
COMPUTE ALPHADATE/A6MDY = '120599' ;
        REALDATE/MDY = ALPHADATE;
```

日付フォーマットの基準日

下表は、サポートされている日付フォーマットの基準日を示しています。

フォーマット	基準日
YMD、YYMD、MDYY、DMYY、MDY、DMY	1900/12/31

フォーマット	基準日
YM、YYM、MYY、MY	Windows および UNIX では 1900/12/31
YQ、YYQ、QYY、QY	1901 Q1
JUL、YYJUL	1900/365
D M Y、YY Q W	この値は日付ではなく、単なる数字であるため、これらのフォーマットには基準日は存在しません。

なお、関数の DA および DT には、基準日として 1899/12/31 が使用されます。日付関数についての詳細は、『ibi™ WebFOCUS® 関数リファレンス』を参照してください。

格納用と表示用の日付フォーマット

下表は、フィールドフォーマットの格納時および表示時のフォーマットを示しています。

	日付フォーマット (例、MDYY)		整数、パック 10 進数、文字フォーマット (例、A8MDYY)	
値	格納	表示	格納	表示
1999 年 2 月 28 日	35853	02/28/1999	02281999	02/28/1999
1999 年 3 月 1 日	35854	03/01/1999	03011999	03/01/1999

日付の計算

日付式でフィールドを使用する場合、その使用方法はフィールドフォーマットで決定されます。日付フォーマットの日付計算では、数値演算子および数値関数を組み込むことができます。整数、パック 10 進数、文字の各フォーマットの日付計算では、日付関数を使用する必要があります。数値演算を使用すると、エラーメッセージが表示されたり、誤った結果が返されたりします。

整数、パック 10 進数、文字のフォーマットで日付を処理することのできる関数の一式は製品に付属しています。日付関数についての詳細は、『ibi™ WebFOCUS® 関数リファレンス』を参照してください。

日付の計算

次の例では、SHIPPING データベースを保守する企業を想定します。ここでは、出荷部が品物の注文を受けてから出荷するまでの作業日数を、受注日の ORDER_DATE を出荷日の SHIPDATE から減算して求めます。

```
COMPUTE TURNAROUND/I4 = SHIP_DATE - ORDER_DATE;
```

品物の受注日が 1999/2/28 で、出荷日が 1999/3/1 の場合、その差は 1 日です。ただし、SHIP_DATE および ORDER_DATE フィールドが整数フォーマットの場合、数値演算子による減算 (-) を行えないため、「730000」という誤った計算結果が得られます。

下表は、フィールドフォーマットが結果に与える影響を示しています。

	日付フォーマットの値	整数フォーマットの値
SHIP_DATE = 1999 年 3 月 1 日	35854	03011999
ORDER_DATE = 1999 年 2 月 28 日	35853	02281999
TURNAROUND	1	730000

整数、パック 10 進数、文字の各フォーマットのフィールドで正確な結果を得るには、日付関数の MDY を使用して月-日-年フォーマットの 2 つの日付の差を計算します。次のように MDY 関数を使用して、TURNAROUND を計算することができます。

```
COMPUTE TURNAROUND/I4 = MDY(ORDER_DATE, SHIP_DATE);
```

DEFINE および COMPUTE による 4 桁の西暦の日付

DEFINE または COMPUTE コマンドに式を使用したり、マスターファイルの DEFINE 属性に式を使用したりして、4 桁の西暦年の処理を行うスライドウィンドウ手法を実装することができます。西暦年の最初の 2 桁が他の場所で使用されていない場合は、パラメータの DEFCENT および YRTHRESH で指定される西暦の変換を行うことができます。最初の 2 桁が指定された場合は、それがそのまま受け入れられます。

フィールドフォーマットの選択

日付式は常に数字を返します。その数字は、日付または 2 つの日付間の日数、月数、四半期数、年数のいずれかを表します。日付式を使用してフィールドに値を割り当てる場合、そのフィールドで選択したフォーマットにより、返される結果が決定されます。

返されるフィールドフォーマットの選択

次の例では、日付フォーマットが設定された SHIP_DATE および ORDER_DATE のフィールドに対してコマンドを実行します。1 つ目のコマンドは、品物の受注日の ORDER_DATE を出荷日の SHIP_DATE から減算して、出荷部が注文の処理に要する日数を計算します。2 つ目のコマンドは、品物の出荷日に 5 日を加算して、この品物の配達日を計算します。

```
COMPUTE TURNAROUND/I4 = SHIP_DATE - ORDER_DATE;
COMPUTE DELIVERY/MDY = SHIP_DATE + 5;
```

1 つ目のコマンドでは、日付式が返す値は注文の処理に要する日数です。そのため、関連するフィールドの TURNAROUND は整数フォーマットにする必要があります。2 つ目のコマン

ドでは、日付式が返す値は品物の配達日です。そのため、関連するフィールドの DELIVERY は日付フォーマットにする必要があります。

日付定数の使用

日付フォーマットのフィールドの計算に日付定数を使用する場合は、日付定数を一重引用符 (') で囲む必要があります。囲んでいない場合は、基準日 (1900/12/31) からその定数までの日数として認識されます。たとえば、「022899」という値を一重引用符で囲んでいない場合、この値は 1999/2/28 とは認識されず、1900/12/31 から 22,899 日後の日付として認識されます。

日付定数によるフィールドの初期化

次のコマンドは、START_DATE の初期値を日付定数の 1999/2/28 に設定します。

```
COMPUTE START_DATE/MDY = '022899';
```

次のコマンドは、1999/1/1 からの経過日数を計算します。

```
COMPUTE YEAR_TO_DATE/I4 = CURR_DATE - 'JAN 1 1999' ;
```

日付要素の抽出

日付要素には、日数、月数、四半期数、年数があります。日付フォーマットのフィールドから日付要素を抽出する式を記述することができます。ただし、日数、月数、四半期数の要素を含まない日付からこれらの値を抽出する式を記述することはできません。たとえば、年数のみを表す YY フォーマットの日付から月数を抽出することはできません。

日付から月数要素の抽出

次の例では、MDYY フォーマットの SHIP_DATE から月数要素を抽出します。

```
COMPUTE SHIP_MONTH/M = SHIP_DATE;
```

SHIP_DATE の値が 1999/3/1 の場合、SHIP_MONTH に対して上記の式が返す値は 03 です。

日付要素の計算は、目的の要素に対する有効な値を自動的に生成します。たとえば、SHIP_MONTH の現在値が 03 の場合、次の式では「06」という値が正しく返されます。

```
COMPUTE ADD_THREE/M = SHIPMONTH + 3;
```

月数を加算してその結果が 12 を超える場合は、加算後の月数が正しく調節されます。たとえば、11 + 3 = 14 ではなく、11 + 3 = 2 になります。

フォーマットが異なるフィールドの組み合わせ

日付フォーマットのフィールドを使用する場合、1 つの式に、日付時間要素の並び順が異なるフィールドを組み合わせて使用することができます。さらに、日付式の結果を、その式で使用したフィールドとは日付時間要素の並び順が異なるフィールドに割り当てることもできます。

ただし、整数、パック 10 進数、文字フォーマットの日付を組み合わせて式を記述することはできません。

YYMD と MDY フィールドフォーマットの組み合わせ

次の例では、DATE_PAID と DUE_DATE の 2 つのフィールドについて考察します。DATE_PAID のフォーマットは YYMD で、DUE_DATE のフォーマットは MDY です。1 つの式にこれら 2 つのフィールドを組み合わせて、支払い遅延日数を計算することができます。

```
COMPUTE DAYS_LATE/I4 = DATE_PAID - DUE_DATE;
```

並び順が異なる要素の結果フィールドへの割り当て

次の例では、DATE_SOLD フィールドについて考察します。このフィールドは YYMD フォーマットで、品物の販売日の値が格納されています。次の式を使用して、DATE_SOLD の値に 7 日を加算し、品物を返品できる最終日を計算します。続いて、その結果を DMY フォーマットのフィールドに割り当てます。

```
COMPUTE RETURN_BY/DMY = DATE_SOLD + 7;
```

日付時間式の作成

日付時間式は、日付と時間の要素を返します。付属の日付時間関数を使用して、これらの式を作成することができます。

日付および日付時間のフォーマット自動変換

以前のバージョンの日付時間フィールドでは、日付フォーマットと日付時間フォーマットの変換にはすべて日付時間関数を使用する必要がありました。これらの関数による変換は引き続きサポートされていますが、特定の演算では関数の使用が必要がなくなりました。

次の自動直接演算が、日付および日付時間フォーマットでサポートされます。

- **割り当て**

  日付時間フィールドへの日付フィールドまたは日付定数の割り当て　時間構成要素は、0 (午前零時) に設定されます。日付は、YYMD のように完全日付形式にすることも、YYM のような部分日付形式にすることもできます。ただし、Q など単一構成要素の日付形式にすることはできません。この日付タイプは、レポート内では日付として表示されますが、整数値として保存され、式では整数値として使用されるためです。

  日付フィールドへの日付時間フィールドまたは日付時間定数の割り当て　時間構成要素は削除されます。

- **比較と減算**

日付時間値が日付値と比較または日付値から減算される場合、または日付値が日付時間値と比較または日付時間値から減算される場合は、日付は日付時間に変換され、時間構成要素は 0 (午前零時) に設定されます。これらは、変換後に日付時間値として比較または減算されます。

- **関数パラメータ**

  簡略日付関数は、日付パラメータとして日付値または日付時間値を使用することができます。レガシーユーザ関数では、この新しい機能はサポートされません。日付時間関数 (H 関数) は、日付時間パラメータを使用しますが、新しい日付関数は、基準日からのオフセットとして格納された新しい日付を使用します。

日付または日付時間定数の認識と使用

- 定数は、DT 演算子を使用せず、文字列として表すことができます。
- 定数は、比較、減算、割り当ての対象となるフィールドフォーマットに従って、日付値および日付時間値間で変換されます。
- 適切な区切り文字を使用し、明確に変換またはフォーマット設定された文字列フォーマット (数値文字列または数値を除く) で表された場合を除き、定数の日付としての認識は、対応するフィールドのフォーマットに依存します。

  この場合、桁数の大きさは、完全構成要素の日付値または日付時間値の場合は 6 桁以上（4 桁の年を使用する場合は 8 桁以上）、部分的構成要素の日付の場合は 3 桁以上、単一構成要素の日付の場合は 1 桁以上に厳密に制限されます。
- 関数パラメータとして数値定数が使用され、対応するフィールドが存在しない場合は、YYMD または YMD フォーマットに基づいて認識されます。唯一の例外は、単一のブランクまたは数字の 0 (ゼロ) を含む文字列で、この場合、レポートにはブランクとして表示されます。日付オフセット定数は、使用できなくなりました。単一文字列内の各桁間のブランク区切り文字も使用できません。

日付値および日付時間値の割り当て

次のリクエストは、DT_CURRENT_DATETIME 関数を使用して日付時間値を生成します。次に、この値を日付フィールドに割り当て、さらにこの日付フィールドを日付時間フィールドに割り当てます。

```
TABLE FILE WF_RETAIL_LITE
PRINT QUANTITY_SOLD NOPRINT AND COMPUTE
 DATETIME1/HYYMDm =  DT_CURRENT_DATETIME(MILLISECOND);
     AS 'Date-Time 1'
```

```
COMPUTE
 DATE1/YYMD   = DATETIME1;
     AS 'Date'
COMPUTE
DATETIME2/HYYMDm = DATE1;
     AS 'Date-Time 2'
WHERE RECORDLIMIT EQ 20
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。元の日付時間フィールドには、0 (ゼロ) ではない時間構成要素が含まれています。日付フィールドに割り当てられると、時間構成要素は削除されます。この日付が、2 つ目の日付時間フィールドに割り当てられると、0 時の時間構成要素が追加されます。

Date-Time 1	Date	Date-Time 2
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.855000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.856000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.856000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.856000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.856000	2017/08/25	2017/08/25 00:00:00.000000
2017/08/25 09:10:06.856000	2017/08/25	2017/08/25 00:00:00.000000

日付値および日付時間値の比較

次のリクエストは、1 つの日付時間フィールドと 1 つの日付フィールドを生成します。QUANTITY_SOLD が 1 の場合、両方のフィールドに同一の日付値が表示され、日付時間フィールドには 0 時の時間構成要素が追加されます。QUANTITY_SOLD が 2 の場合、各フィールドには異なる日付値が表示され、日付時間フィールドには 0 時の時間構成要素が追加されます。それ以外の場合はすべて、日付時間フィールドには現在の日付と 0 (ゼロ) ではない時間構成要素、日付フィールドには現在の日付が表示されます。EQUAL1 フィールドは、両方のフィールドが等しいかどうかを比較します。

```
TABLE FILE WF_RETAIL_LITE
PRINT QUANTITY_SOLD AS Quantity AND COMPUTE
 DATETIME1/HYYMDm = IF QUANTITY_SOLD EQ 1 THEN '2017/06/05'
        ELSE IF QUANTITY_SOLD EQ 2 THEN '2016/02/29'
        ELSE DT_CURRENT_DATETIME(MILLISECOND);
        AS 'Date-Time'
COMPUTE
  DATE1/YYMD   = IF QUANTITY_SOLD EQ 1 THEN '2017/06/05'
        ELSE IF QUANTITY_SOLD EQ 2 THEN '2015/12/30'
        ELSE DT_CURRENT_DATE();
        AS 'Date'
COMPUTE
  EQUAL1/A1 = IF DATETIME1 EQ DATE1 THEN 'Y' ELSE 'N';
        AS 'Equal?'
WHERE RECORDLIMIT EQ 12
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。日付値を日付時間値と比較する場合、日付が、時間構成要素に 0 時が追加された日付時間値に変換された後、値が比較されます。したがって、QUANTITY_SOLD が 1 の場合、両方の日付構成要素は等しく、日付時間フィールドの時間構成要素が 0 (ゼロ) に設定されるため、日付が日付時間値に変換されると、両方のフィールドは等しくなります。QUANTITY_SOLD が 2 の場合は、日付構成要素が異なるため、両方のフィールドは等しくなりません。QUANTITY_SOLD が 3 の場合、日付構成要素は等しくなりますが、日付時間フィールドには、0 (ゼロ) 以外の時間構成要素が含まれます。したがって、日付フィールドが、時間構成要素に 0 時が追加された日付時間値に変換された後に比較された場合、各フィールドは等しくなりません。

Quantity	Date-Time	Date	Equal?
1	2017/06/05 00:00:00.000000	2017/06/05	Y
1	2017/06/05 00:00:00.000000	2017/06/05	Y
1	2017/06/05 00:00:00.000000	2017/06/05	Y
2	2016/02/29 00:00:00.000000	2015/12/30	N
1	2017/06/05 00:00:00.000000	2017/06/05	Y
1	2017/06/05 00:00:00.000000	2017/06/05	Y
2	2016/02/29 00:00:00.000000	2015/12/30	N
1	2017/06/05 00:00:00.000000	2017/06/05	Y
1	2017/06/05 00:00:00.000000	2017/06/05	Y
1	2017/06/05 00:00:00.000000	2017/06/05	Y
3	2017/08/25 09:24:45.203000	2017/08/25	N
1	2017/06/05 00:00:00.000000	2017/06/05	Y

日付時間フィールドの日付構成要素の順序を指定

```
SET DATEFORMAT = option
```

説明

option

MDY、DMY、YMD、MYD のいずれかを指定することができます。U.S. English フォーマットでは、デフォルト値は MDY です。

日付時間値の指定

外部の日付時間値は、次のいずれかのソースから取得される文字フォーマットの定数です。

- シーケンシャルデータソース。
- WHERE、IF、DEFINE、COMPUTE の式で使用。

日付時間定数、文字ファイルに表示される日付時間値には、次のいずれかのフォーマットが指定されます。

```
time_string  [date_string]
date_string  [time_string]
```

COMPUTE、DEFINE、WHERE 式の日付時間定数は、次のいずれかのフォーマットにする必要があります。

```
DT(time_string  [date_string])
DT(date_string  [time_string])
```

IF 式の日付時間定数は、次のいずれかのフォーマットにする必要があります。

```
'time_string  [date_string]'
'date_string  [time_string]'
```

値にブランクおよび特殊文字が含まれていない場合は、一重引用符 (') を使用する必要はありません。なお、IF 条件では、DT 接頭語はサポートされません。

説明

time_string

ブランクを含めることはできません。複数の時間要素はコロン (:) で区切ります。「AM」、「PM」、「am」、「pm」という語句をその後に配置することができます。以下はその例です。

```
14:30:20:99       (99 milliseconds)
14:30
14:30:20.99       (99/100 seconds)
14:30:20.999999   (999999 microseconds)
02:30:20:500pm
```

なお、秒数を小数点で表現したり、秒数の後にコロンを配置したりすることができます。

- 秒数の後にコロンを配置する場合、その後に続く値はミリ秒を表します。この表記を使用する場合、マイクロ秒およびナノ秒を表現する方法はありません。
- 秒数の値の小数点は、秒数の小数部を表します。マイクロ秒は、6 桁の小数を使用して表すことができます。ナノ秒は、9 桁で表すことができます。

date_string

次の 3 つのフォーマットのいずれかを使用することができます。

- **数値フォーマット**　正確に 4 桁、6 桁、8 桁のいずれかです。4 桁の文字列は年と認識されます (世紀の指定が必要)。月日は 1 月 1 日に設定されます。6 桁と 8 桁の文字列には、それぞれ 2 桁または 4 桁の西暦年が含まれ、その後ろに 2 桁の月と 2 桁の日が指定されます。このフォーマットでは構成要素の順序が固定されているため、DATEFORMAT 設定は無視されます。

  8 桁より長い数値列フォーマットは Hn フォーマットの組み合わせ日付時間列として処理されます。数列日付定数の例は次のとおりです。

  ```
  99
  1999
  19990201
  ```

- **書式設定された文字列フォーマット**　1 桁または 2 桁の日、1 桁または 2 桁の月、2 桁または 4 桁の年で構成され、それぞれをブランク、スラッシュ (/)、ハイフン (-)、ピリオド (.) のいずれかで区切ります。3 つの部分がすべて存在し、DATEFORMAT の設定に従う必要があります。3 つのフィールドの中で 4 桁のものは年と認識され、他の 2 つのフィールドは DATEFORMAT 設定で指定された順序に従う必要があります。次の例は、フォーマット設定済みの日付定数です。

  ```
  1999/05/20
  5 20 1999
  99.05.20
  1999-05-20
  ```

- **変換された文字列フォーマット**　完全な月名または省略された月名で構成されます。年は、4 桁または 2 桁のフォーマットで表示する必要があります。日の値が欠落している場合は、月の 1 日目と見なされます。日の値は、1 桁または 2 桁にすることができます。文字列に 2 桁の年と 2 桁の日の両方が含まれる場合は、DATEFORMAT で設定された順序に従う必要があります。以下はその例です。

  ```
  January 6 2000
  ```

注意

- 日付および時間の文字列は、1 つ以上のブランクで区切る必要があります。ブランクは、日付時間文字列の先頭または末尾に配置することも、am/pm 表記の直前に配置することもできます。
- 日付フォーマットでは、DEFCENT および YRTHRESH 設定を使用して、2 桁の年を認

識します。

日付時間リテラルの割り当て

日付時間リテラルを日付時間フィールドに割り当てるために、COMPUTE、DEFINE、または WHERE 式に接頭語 DT を使用できます (接頭語 DT の使用は必須ではなくなりました)。以下はその例です。

```
DT2/HYYMDS = DT(20051226 05:45);
```

```
DT3/HYYMDS = DT(2005 DEC 26 05:45);
```

```
DT4/HYYMDS = DT(December 26 2005 05:45);
```

日付時間フィールドの日付コンポーネントの順序指定

次のリクエストは、DATEFORMAT を MYD に設定します。

```
SET DATEFORMAT = MYD
DEFINE FILE EMPLOYEE
DTFLDYYMD/HYYMDI =  DT(APR 04 05);
END

TABLE FILE EMPLOYEE
PRINT CURR_SAL DTFLDYYMD
END
```

出力結果では、一般表記の日付リテラル「APR 04 05」が「1904/04/05」と解釈されています。

```
  CURR_SAL  DTFLDYYMD
  --------  ---------
$11,000.00  1904/04/05 00:00
$13,200.00  1904/04/05 00:00
$18,480.00  1904/04/05 00:00
```

```
 $9,500.00  1904/04/05 00:00
$29,700.00  1904/04/05 00:00
$26,862.00  1904/04/05 00:00
$21,120.00  1904/04/05 00:00
$18,480.00  1904/04/05 00:00
$21,780.00  1904/04/05 00:00
$16,100.00  1904/04/05 00:00
 $9,000.00  1904/04/05 00:00
$27,062.00  1904/04/05 00:00
```

トランザクションファイルからの日付時間値の読み取り

DTTRANS カンマ区切りトランザクションファイルでは、ID フィールド、日付 (8 バイト)、時間 (時:分:秒フォーマット) で構成された日付時間フィールドが使用されます。

```
01, 20000101 02:57:25,$
02, 19991231 14:05:35,$
```

トランザクションファイルでは数値フォーマットの日付が使用されるため、DATEFORMAT 設定は使用せず、日付を YMD の順序で入力します。

次のトランザクションファイルも有効です。ここでは、デフォルトの DATEFORMAT 設定である MDY に対応するフォーマットが設定された文字列日付が使用されています。

```
01, 01/01/2000 02:57:25,$
02, 12/31/1999 14:05:35,$
```

次のマスターファイルには、これらの値を受け取る「DATETIME」という名前の FOCUS データソースが記述されています。

```
FILE=DATETIME,    SUFFIX=FOC   ,$
SEGNAME=DATETIME, SEGTYPE=S0   ,$
FIELD=ID,  ID,    USAGE = I2   ,$
FIELD=DT1, DT1,   USAGE=HYYMDS ,$
```

COMPUTE コマンドでの日付時間値の使用

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME AND COMPUTE
NEWSAL/D12.2M = CURR_SAL + (0.1 * CURR_SAL);
RAISETIME/HYYMDIA = DT(20000101 09:00AM);
WHERE CURR_JOBCODE LIKE 'B%'
END
```

出力結果は次のとおりです。

```
LAST_NAME        FIRST_NAME           NEWSAL  RAISETIME
---------        ----------           ------  ---------
SMITH            MARY             $14,520.00  2000/01/01  9:00AM
JONES            DIANE            $20,328.00  2000/01/01  9:00AM
ROMANS           ANTHONY          $23,232.00  2000/01/01  9:00AM
MCCOY            JOHN             $20,328.00  2000/01/01  9:00AM
BLACKWOOD        ROSEMARIE        $23,958.00  2000/01/01  9:00AM
MCKNIGHT         ROGER            $17,710.00  2000/01/01  9:00AM
```

WHERE 条件での日付時間値の使用

WHERE 句では、日付時間定数には DT( ) フォーマットを使用する必要があります。

```
TABLE FILE VIDEOTR2
PRINT CUSTID TRANSDATE
WHERE TRANSDATE GT DT(2000/01/01 02:57:25)
END
```

出力結果は次のとおりです。

```
CUSTID  TRANSDATE
------  ---------
1118    2000/06/26 05:45
1237    2000/02/05 03:30
```

IF 条件での日付時間値の使用

IF 句でブランクを使用する場合は、日付時間定数を一重引用符 (') で囲む必要があります。

```
TABLE FILE VIDEOTR2
PRINT CUSTID TRANSDATE
IF TRANSDATE GT '2000/01/01 02:57:25'
END
```

注意：IF 句では、日付時間定数の DT 接頭語はサポートされません。

出力結果は次のとおりです。

```
CUSTID  TRANSDATE
------  ---------
1118    2000/06/26 05:45
1237    2000/02/05 03:30
```

世界標準時による日付入力値の指定

STANDARD および STANDARDU の DTSTANDARD 設定を使用して、次の日付時間値を入力値として読み取ることができます。

入力値	説明
`14:30[:20,99]`	時間要素を区切る場合は、ピリオド (.) の代わりにカンマ (,) を使用します。
`14:30[:20.99]Z`	世界標準時です。

`15:30[:20,99]+01`
`15:30`
`[:20,99]+0100`
`15:30`
`[:20,99]+01:00`
	中央ヨーロッパ標準時において、これらの入力値はすべて上記の値と等しくなります。

入力値	説明
`09:30[:20.99]-05`	東部標準時において、この入力値は上記の値と等しくなります。

なお、これらの値は内部的に STANDARDU 設定と同様に格納されます。STANDARD 設定を使用する場合、Z、+、- の後に来るものはすべて無視されます。

日付時間値の操作

2 つの日付時間値は、それぞれの値の長さが異なる場合でも比較することができます。

日付時間フィールドでミッシング値がサポートされる場合、ミッシング値が存在するフィールドには、日付時間フィールドのすべての値より大きい値が存在することになります。そのため、選択テストで GT または GE の演算子を使用した場合にレポート出力からミッシング値を除外するには、新しい定数フィールドの NE MISSING を選択テストに追加することをお勧めします。

```
date_time_field {GT|GE} date_time_value AND date_time_field NE MISSING
```

長さが異なる日付時間フォーマット間でも、値を割り当てることができます。10 バイトの日付時間値を 8 バイトの日付時間値に割り当てると、マイクロ秒の部分が切り捨てられます (端数処理は行われません)。短い値を長い値に割り当てると、マイクロ秒の順位の低い 3 桁が 0 (ゼロ) に設定されます。

他の演算子、たとえば連結、EDIT、日付時間の被演算子 LIKE などはサポートされていません。文字フィールドで使用できる演算接頭語はサポートされます。

日付時間ミッシング値のテスト

次の例では、DATETIM2 マスターファイルについて考察します。

```
FILE=DATETIM2,  SUFFIX=FOC                ,$
SEGNAME=DATETIME, SEGTYPE=S0              ,$
FIELD=ID, ID, USAGE = I2                  ,$
FIELD=DT1, DT1,   USAGE=HYYMDS, MISSING=ON,$
```

DT1 フィールドでは、ミッシング値がサポートされます。ここで、次のリクエストを発行します。

```
TABLE FILE DATETIM2
PRINT ID DT1
END
```

レポートの出力結果から、DT1 フィールドの ID=3 インスタンスにミッシング値が存在することが分かります。

```
ID  DT1
--  ---
 1  2000/01/01 02:57:25
 2  1999/12/31 00:00:00
 3  .
```

次のリクエストでは、DT1 の値の中から 2000/01/01 00:00:00 より大きい値を選択し、ミッシング値を除外します。

```
TABLE FILE DATETIM2
PRINT ID DT1
  WHERE DT1 NE MISSING AND DT1 GT DT(2000/01/01 00:00:00);
END
```

このレポート出力には、ミッシング値は表示されていません。

```
ID  DT1
--  ---
 1  2000/01/01 02:57:25
```

日付時間フィールドへの異なるフォーマットの割り当て

次の例では、VIDEOTR2 データソースを使用したリクエストについて考察します。

```
TABLE FILE VIDEOTR2
 PRINT CUSTID TRANSDATE AND COMPUTE
  DT2/HYYMDH = TRANSDATE;
  T1/HHIS = TRANSDATE;
```

```
WHERE DATE EQ 2000
END
```

出力結果は次のとおりです。

```
CUSTID  TRANSDATE         DT2             T1
------  ---------         ---             --
1118    2000/06/26 05:45  2000/06/26 05   05:45:00
1237    2000/02/05 03:30  2000/02/05 03   03:30:00
```

文字式の作成

文字式は、文字定数、フィールド、連結演算子、IF-THEN-ELSE ロジック、関数を使用して、文字の値を取得します。

テキストフィールドおよび文字フィールドの両方には、TABLE COMPUTE、DEFINE コマンドの文字フィールドまたは文字式に格納された値を割り当てることができます。テキストフィールドの値を文字フィールドに割り当てる際に、その値が文字フィールドの長さを超える場合は、割り当て前に値の末尾が切り捨てられます。

文字式は次の要素で構成されます。

- 一重引用符 (') で囲んだ文字定数 (文字列)。以下はその例です。

  ```
  COMPUTE STATE/A2 = 'NY';
  ```

- 連結演算子で結合した文字フィールドまたは定数。以下はその例です。

  ```
  DEFINE FILE EMPLOYEE TITLE/A19 = 'DR. ' | LAST_NAME;
  END
  ```

- 文字関数。以下はその例です。

  ```
  DEFINE FILE EMPLOYEE INITIAL/A1 = EDIT(FIRST_NAME, '9$$$$$$$$$$');
  END
  ```

- テキストフィールド。

注意

- 表示不可の文字は文字定数には使用できません。

- フィールドに MISSING ON が設定されている場合、連続する 2 つの一重引用符 ('') で、A1V フォーマットおよび実長 0 (ゼロ) の Null 値を表します。

引用符区切りリテラル列での引用符の埋め込み

状況によっては、引用符で区切られた文字列内に、さらに引用符を埋め込むことができます。文字列内で一重引用符 (') を表示するには、1 つの一重引用符または 2 つの連続した一重引用符を使用します。この両方が一重引用符として解釈されます。

引用符区切りの文字列は、次の場合に使用することができます。

- 複数の引用符を使用した WHERE および IF 条件
- *fieldname* {IS、IS-NOT、IN、IN FILE、または NOT IN FILE} を使用した WHERE 条件
- EDIT
- WHEN fieldname EQ リテラルに埋め込まれた引用符
- DEFINE コマンド
- マスターファイルの DEFINE 属性
- マスターファイルの DBA 属性 (例、VALUE = fieldname EQ リテラルに埋め込まれた引用符)
- マスターファイルの ACCEPT=、DESCRIPTION=、TITLE= 属性
- AS
- DECODE

引用符区切りリテラル列での O'BRIEN データ値の指定

次の例では、「O'BRIEN」というデータ値を正しく認識させるために引用符を使用します。

```
TABLE FILE VIDEOTRK
PRINT LASTNAME
WHERE LASTNAME IS 'O'BRIEN'
END
```

文字列の連結

式を記述して、複数の文字定数またはフィールドを連結して 1 つの文字列にすることができます。下表は、連結演算子の 2 つの形式を示しています。

シンボル	意味	説明
|	弱連結	末尾のブランクをそのまま保持します。
||	強連結	末尾のブランクを、連結した文字列の末尾へ移動します。

文字列の連結

次の例では、EDIT 関数を使用して、名前のイニシャルを抽出します。さらに、強連結および弱連結の両方を使用して、姓、カンマ (,)、イニシャル、ピリオド (.) の順に値を出力します。

```
DEFINE FILE EMPLOYEE
FIRST_INIT/A1 = EDIT(FIRST_NAME, '9$$$$$$$$$$');
NAME/A19 = LAST_NAME ||(', '| FIRST_INIT |'.');
END
```

```
TABLE FILE EMPLOYEE
PRINT NAME WHERE LAST_NAME IS 'BANNING'
END
```

出力結果は次のとおりです。

```
NAME
----
BANNING, J.
```

このリクエストでは、次のように式が評価されます。

1. EDIT 関数が、FIRST_NAME からイニシャルの J を抽出します。
2. 括弧内の式が値を返します。

   ```
   , J.
   ```

3. LAST_NAME が、手順 2 で取得した文字列に結合されます。

   ```
   Banning, J.
   ```

   LAST_NAME のフォーマットが EMPLOYEE マスターファイルで A15 に設定されているため、強連結により末尾のブランクが除外されます。ブランクを除外したり保持したりすることに関係なく、結果として生成されるフィールド名の NAME の長さは 19 バイト (A19) になります。

文字式での IF-THEN-ELSE ロジックの使用

次のリクエストでは、IF-THEN-ELSE ロジックを使用して、MOVIECODE に連結する文字を特定し、NEWCODE を計算します。

```
TABLE FILE MOVIES
PRINT COPIES
MOVIECODE
COMPUTE
NEWCODE/A20 =  MOVIECODE |(IF MOVIECODE CONTAINS 'DIS' THEN  'NEY;' ELSE
';');
BY CATEGORY
WHERE CATEGORY EQ 'CHILDREN'
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。MOVIECODE に「DIS」が含まれている場合、生成される NEWCODE に「NEY;」が連結されます。それ以外の場合、生成される NEWCODE に「;」が連結されます。

CATEGORY	COPIES	MOVIECODE	NEWCODE
CHILDREN	1	031KKV	031KKV;
	2	043DIS	043DISNEY;
	1	093WOR	093WOR;
	2	306DIS	306DISNEY;
	1	309RAN	309RAN;
	1	387PLA	387PLA;
	1	476DIS	476DISNEY;
	3	732DIS	732DISNEY;

可変長文字式の作成

文字タイプの AnV フィールドは、An タイプの場合と同様に、算術式および論理式で使用することができます。

- 式に AnV タイプのフィールドを使用する場合は、AnV タイプまたは An タイプのいずれかとして使用できます。
- 式の実行後のタイプは、指定した演算タイプにより異なります。詳細は、次のセクションで説明します。

注意：AnV フィールドには 2 バイトのオーバーヘッドがあり、それを削除するために追加処理が必要となるため、AnV フォーマットは非リレーショナルデータソースで使用しないことをお勧めします。

A*n*V フィールドでの連結の使用

2 つのフィールド間の連結で被演算子のいずれかが AnV フィールドの場合、この連結を実行する際に可変長文字の規則が適用されます。

- 連結した文字列のサイズは、各被演算子のサイズの合計になります。
- 弱連結では、連結した文字列の実際の長さは、2 つの入力文字列の実際の長さの合計になります。

- 強連結では、連結した文字列が AnV フィールドに格納された際の実際の長さは、1 つ目の入力文字列の実際の長さからブランク数を差し引き、それに 2 つ目の入力文字列の実際の長さを加えた合計になります。
- 連結を行う際の An フィールドでは、サイズと長さは一致します。
- フィールドに MISSING ON が設定されている場合、連続する 2 つの一重引用符 ('') で、A1V フォーマットおよび実長 0 (ゼロ) の Null 値を表します。

A*n*V フィールドでの EDIT 関数の使用

次の式では、x が AnV フォーマットである場合に限り、式の結果が AnV フォーマットになります。

```
EDIT(x,mask)
```

この結果の実際の長さは、ドル記号 ($) を除外して数えたマスクの文字数になります。

なお、結果として実際の長さが 0 (ゼロ) になることもあります。

EDIT(x) を使用して、x が AnV フォーマットの場合に、AnV フィールドを整数値に変換することができます。

*n*V フィールドでの CONTAINS および OMITS の使用

AnV フィールドで CONTAINS および OMITS 演算子を使用する場合、これらの演算子の評価において差が生じるのは、被演算子のいずれかの実際の長さが 0 (ゼロ) の場合のみです。

次の例では、Z フィールドの実際の長さは 0 (ゼロ) で、X および Y フィールドの実際の長さは 0 (ゼロ) 以外です。

式	結果
`Z CONTAINS Y`	FALSE

式	結果
`X CONTAINS Z`	TRUE
`Z CONTAINS Z`	TRUE
`Z OMITS Y`	TRUE
`X OMITS Z`	FALSE
`Z OMITS Z`	FALSE

A*n*V フィールドでの LIKE の使用

次の式を評価する場合、その評価において差が生じるのは、x の実際の長さが 0 (ゼロ) の場合のみです。

```
x LIKE mask ...
```

次の例では、フィールドインスタンスの Z には実際の長さが 0 (ゼロ) の値が存在します。

```
Z LIKE mask ...
```

この式は、マスクにパーセント記号 ('%') のみが存在する場合に評価が TRUE になります。

なお、空の文字列に対して評価を行えるマスクはほかにはありません。次の式では、マスクの長さが 1 であっても、評価が FALSE になります。

```
Z LIKE ''
```

A*n*V フィールドでの EQ、NE、LT、GT、LE、GE 演算子の使用

An タイプのフィールドと同様に、短い被演算子にはブランクが追加されることを前提に演算が評価されます。

そのため、空の AnV フィールドの Z は、すべてブランクで構成されたフィールドとして比較されます。

次の例では、Z が空の AnV フィールドインスタンス、X は空ではないブランク以外の文字を含む AnV フィールドインスタンスです。

式	結果

`Z EQ Z`
`Z GE Z`
`Z LE Z`
	TRUE

`Z NE Z`
`Z LT Z`
`Z GT Z`
	FALSE

`Z EQ X`	FALSE
`Z NE X`	TRUE
`Z LT X`	TRUE
`Z GT X`	FALSE

式	結果
`Z LE X`	TRUE
`Z GE X`	FALSE
`X EQ Z`	FALSE
`X NE Z`	TRUE
`X LT Z`	FALSE
`X GT Z`	TRUE
`X LE Z`	FALSE
`X GE Z`	TRUE

A*n*V フィールドでの DECODE 関数の使用

```
DECODE alphafield (value 'result'...
```

An または AnV フィールドのいずれかで DECODE を使用する場合、value と result 関係の result の部分が定数として指定される限り、この結果は An タイプになります。なお、定数は An タイプです。

A*n*V フィールドでの代入演算子の使用

AnV フォーマットで代入演算子を使用する場合、次の 3 つの状況が考えられます。代入演算子の右側にのみ AnV データタイプがある場合、両側に AnV データタイプがある場合、左側にのみ AnV データタイプがある場合です。

```
fld/An = AnV_type_expression;
```

- 評価する式の実際の長さは、An フィールドへの代入時に失われます。
- AnV 結果のサイズにより、短い An フォーマットフィールドへの代入が妨げられることはありません。
    - 代入演算子の左側にあるフィールドの長さより式の結果の実際の長さが短い場合は、その結果にブランクが追加されます。
    - 代入演算子の左側にあるフィールドの長さより式の結果の実際の長さが長い場合は、その結果の末尾が省略されます。

      ```
      fld/AnV = AnV_type_expression;
      ```

- 宣言したフィールドのサイズを超えない限り、式の結果の長さが代入演算子の左側のフィールドの長さに割り当てられます。この場合、割り当てられる長さは宣言したサイズ (n) になります。
- AnV 評価結果のサイズにより、短い AnV フォーマットフィールドへの割り当てが妨げられることはありません。
    - 代入演算子の左側にあるフィールドのサイズより式の結果の長さが短い場合は、その結果にブランクが追加されます。
    - 代入演算子の左側にあるフィールドのサイズより式の結果の実際の長さが長い場合は、その結果の末尾が省略されます。

      ```
      fld/AnV = An_type_expression;
      ```

- 代入演算子の左側にあるフィールドの長さには、サイズ (n) と等しいサイズが割り当てられます。
- 結果の実際の長さは、AnV フィールドで宣言したサイズ n に対して検証されます。結果が n より長い場合は、エラーが表示されます。

論理式の作成

論理式は、特定の条件が TRUE であるかどうかを特定します。論理式には、関係式とブール式の 2 種類があります。使用する式の種類は、比較する要素により決定されます。

- 関係式は、特定の 2 つの値 (フィールド値または定数のいずれか) を比較して、TRUE または FALSE を返します。
- ブール式は、複数の関係式の結果から TRUE または FALSE を返します。

論理式を使用して、数値フィールドに値を割り当てることができます。式が TRUE の場合は、フィールドに 1 の値を返します。式が FALSE の場合、フィールドには 0 (ゼロ) の値が返されます。

論理演算子

下表は、論理式で使用する標準的な演算子の一覧です。WHERE および IF を使用してレコードを選択する際に利用できる関係演算子および他の演算子についての詳細は、レコードの選択を参照してください。

演算子	説明
`EQ`	左辺の値が右辺の値と等しい場合、True を返します。
`NE`	左辺の値が右辺の値と等しくない場合、True を返します。
`GE`	左辺の値が右辺の値以上である場合、True を返します。
`GT`	左辺の値が右辺の値より大きい場合、True を返します。
`LE`	左辺の値が右辺の値以下である場合、True を返します。

演算子	説明
LT	左辺の値が右辺の値より小さい場合、True を返します。
AND	被演算子がいずれも True である場合、True を返します。
OR	被演算子のいずれかが True の場合、True を返します。
NOT	被演算子が False の場合、True を返します。
CONTAINS	指定した文字列を含みます。
OMITS	指定した文字列を含みません。
IS MISSING	フィールドがミッシングの場合、TRUE 値を返します。
IS-NOT MISSING	フィールドがミッシングでない場合、TRUE 値を返します。

関係式の記述

次の構文のすべての要素が、関係式で有効になります。

```
value {EQ|NE} valuevalue {LE|LT} valuevalue {GE|GT} valuecharacter_value
{CONTAINS|OMITS} character_value
```

説明

value

フィールド値または定数です。

character_value

文字列です。ブランクが含まれる場合は、文字列を一重引用符 (') で囲む必要があります。

ブール式の記述

次の構文のすべての要素が、ブール式で有効になります。

```
(relational_expression) {AND|OR} (relational_expression)
NOT (logical_expression)
```

説明

relational_expression

特定の 2 つの値 (フィールド値または定数のいずれか) を比較する式です。

logical_expression

評価結果が TRUE または FALSE の値になる式です。式が TRUE の場合は、フィールドに 1 の値を返します。式が FALSE の場合、フィールドには 0 (ゼロ) の値が返されます。この式は、括弧で囲む必要があります。

条件式の作成

条件式は、論理式の結果に基づいて値を割り当てます。数値または文字の値を割り当てることができます。

注意：IF を使用する選択条件とは異なり、条件式のすべての文字の値を一重引用符 (') で囲む必要があります。たとえば、IF COUNTRY EQ 'ENGLAND' のように記述します。

条件式の記述

```
IF expression1 THEN expression2 [ELSE expression3]
```

説明

expression1

この式を評価して、フィールドに割り当てた値が expression2 の値であるか、expression3 の値であるかを特定します。

expression2

この式では、結果のフォーマットがフィールドに割り当てたフォーマットに対応します。この式を条件式にすることができますが、その場合は括弧で囲む必要があります。

expression3

この式では、結果のフォーマットがフィールドに割り当てたフォーマットに対応します。オプションとして、式を括弧で囲むことができます。

ELSE

オプションとして、expression3 とともに使用します。ただし、ELSE 条件を指定せず、IF 条件が満たされていない場合は、最後に評価した条件から値が取得されます。そのため、ELSE 条件を指定しない場合は、期待した結果が得られない場合があります。

なお、最終的にソートしたレポートには、異なる種類の値が組み合わされて表示されることがあります。これは、DEFINE または COMPUTE を使用したかどうか、データレコードが集計の前後で評価が行われたかどうかにより異なります。

THEN および ELSE の後に記述する式は、フィールドに割り当てたフォーマットと対応するフォーマットでなければなりません。この式は、それ自体が条件式になることもあります。ただし、IF に続く式を IF ... THEN ... ELSE 式 (例、IF ... IF ...) にすることはできません。

条件式による値の指定

次の例では、データ値が欠落している場合 (つまり、フィールドにデータソースの値が存在しない場合) に、条件式を使用して NONE の値を BANK_NAME フィールドに割り当てます。

```
DEFINE FILE EMPLOYEE
BANK_NAME/A20 = IF BANK_NAME EQ ' ' THEN 'NONE'
```

```
ELSE BANK_NAME;
END

TABLE FILE EMPLOYEE
PRINT CURR_SAL AND BANK_NAME
BY EMP_ID BY BANK_ACCT
END
```

出力結果は次のとおりです。

```
EMP_ID     BANK_ACCT         CURR_SAL  BANK_NAME
------     ---------         --------  ---------
071382660                  $11,000.00  NONE
112847612                  $13,200.00  NONE
117593129   40950036       $18,480.00  STATE
119265415                   $9,500.00  NONE
119329144     160633       $29,700.00  BEST BANK
123764317  819000702       $26,862.00  ASSOCIATED
126724188                  $21,120.00  NONE
219984371                  $18,480.00  NONE
326179357  122850108       $21,780.00  ASSOCIATED
451123478  136500120       $16,100.00  ASSOCIATED
543729165                   $9,000.00  NONE
818692173  163800144       $27,062.00  BANK ASSOCIATION
```

TRUE または FALSE 条件の定義

TRUE または FALSE 条件を定義し、その条件をテストしてレポート出力を制御することができます。次の例では、括弧内の関係式のいずれかが TRUE の場合、MYTEST フィールドに TRUE の値を割り当てます。続いて、MYTEST の値をテストします。

```
DEFINE FILE EMPLOYEE
MYTEST= (CURR_SAL GE 11000) OR (DEPARTMENT EQ 'MIS');
END

TABLE FILE EMPLOYEE
PRINT CURR_SAL AND DEPARTMENT
BY EMP_ID
IF MYTEST IS TRUE
END
```

出力結果は次のとおりです。

```
EMP_ID            CURR_SAL  DEPARTMENT
------            --------  ----------
071382660       $11,000.00  PRODUCTION
112847612       $13,200.00  MIS
117593129       $18,480.00  MIS
119329144       $29,700.00  PRODUCTION
123764317       $26,862.00  PRODUCTION
126724188       $21,120.00  PRODUCTION
219984371       $18,480.00  MIS
326179357       $21,780.00  MIS
451123478       $16,100.00  PRODUCTION
543729165        $9,000.00  MIS
818692173       $27,062.00  MIS
```

注意：TRUE または FALSE 条件のテストは、IF コマンドでのみ有効になります。これは、WHERE を使用する際には無効になります。

レポート出力の保存と再利用

デフォルト設定では、リクエストが実行されてデータ値が収集されると、表示可能な形式のレポートが出力されます。この形式のレポートには、フィールドタイトルの追加やフォーマット属性の適用が行えます。データ値を表示する代わりに、特別なデータファイルに保存して、次の用途に使用することができます。

- Web ページ、印刷ドキュメント、またはテキストドキュメントとして表示する。
- スプレッドシート、データベース、ワードプロセッサ、3GL プログラムなど、他のアプリケーションで処理する。
- ブラウザや PC など、他の場所に送信する。
- 元のデータソースの一部を抽出し、マルチステップレポートを生成する。
- データソースを構造化抽出ファイルに抽出し、セグメントの関係についての情報を保持したまま、異なるオペレーティングシステム間でのデータソースとレポートのマイグレートを可能にする。

レポート出力の保存

次のコマンドは、さまざまなファイルフォーマットでレポートを保存または出力することができます。

- **HOLD** HOLD コマンドは、リクエストの出力結果を含むデータソースを作成します。デフォルト設定では、データはバイナリフォーマットで保存されますが、FOCUS、HTML、Excel など、他のフォーマットを指定することもできます。一部のフォーマットでは、HOLD コマンドによりデータソースに対応するマスターファイルも作成されます。その後、HOLD ファイルからデータを抽出または保存する別のリクエストを記述することができます。詳細は、HOLD ファイルの作成を参照してください。
- **SAVE および SAVB** SAVE コマンドは、マスターファイルを作成しない点、およびデフォルトフォーマットが ALPHA である点を除いて、HOLD コマンドと同一です。バイナリフォーマットで SAVE ファイルを作成するには、「SAVB」と呼ばれる SAVE コマンドのバリエーションを使用します。

  HOLD ファイル同様、他のソフトウェア製品での使用に適切な、さまざまなフォー

マットを指定することができます。詳細は、SAVE および SAVB ファイルの作成を参照してください。

- **PCHOLD** PCHOLD コマンドは、リクエストの出力結果を含むデータソースを作成し、HOLD データソースおよびオプションのマスターファイルをクライアントコンピュータまたはブラウザにダウンロードします。HOLD ファイル同様、さまざまなフォーマットを指定することができます。詳細は、PCHOLD ファイルの作成を参照してください。

ヒント：出力ファイルを保存する場合、ファイルのアロケーションを指定することをお勧めします。これは PCHOLD ファイルには適用されません。

レポート出力ファイル名の指定および保存

セッション中は、レポート出力ファイルは消去または上書きされるまで使用可能です。同一セッションで出力ファイルを作成すると、AS 句により他の名前を付けない限り、その前のバージョンを置き換えます。

出力ファイルを作成すると、FILEDEF コマンドが自動的に発行されます。ファイルの指定に使用する ddname は、割り当てない限り、レポート出力ファイル名 (HOLD、SAVE、SAVB または AS 句内の名前) と同一になります。

デフォルト設定では、HOLD、SAVE、または SAVB で作成したレポート出力ファイルは、一時領域に書き込まれます。セッションを終了すると、出力を特定の場所に保存しない限り、これらのファイルは使用できなくなります。

特定の場所にレポート出力を保存するには、FILEDEF コマンドを使用します。詳細は、HOLD ファイルの保存および検索を参照してください。

詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

ON TABLE HOLD AS *name* 構文を使用して HOLD ファイルを作成する場合、名前にはオペレーティングシステムでサポートされる最大バイト数までの文字を使用することができます。

数字の名前または数字で始まる名前は多くのオペレーティングシステムで有効ですが、次の理由により、そのような名前は使用しないことをお勧めします。

- リクエストでデータにアクセスする際に数字で始まるファイル名が指定された SUFFIX=EDA シノニムを使用すると、エラー (FOC14069) が発生します。SUFFIX=EDA シノニムは、リモートサーバのアダプタにより作成されます。
- 数字で始まるファイル名を受容しない外部アプリケーション (例、API アプリケーション) が、数字で始まるファイル名が指定された SUFFIX=EDA シノニムと通信すると、問

題が発生する場合があります。

- 他社製アプリケーションが数字のファイル名または数字で始まるファイル名で正しく動作しない場合に問題が発生することがあります。

HOLD ファイルの作成

HOLD コマンドを使用して、さまざまな用途のレポート出力ファイルを作成することができます。

- データの抽出ツールとして、HOLD コマンドはデータの検索と処理、および後の処理用に結果を抽出することが可能です。リクエストにより、新しいデータソースを作成することができます。データソースには、新しいレポートを生成するために必要なマスターファイルが含まれます。

  出力マスターファイルにはリクエストのフィールドのみが含まれています。HOLD ファイルのフィールド名は、マスターファイルで指定した元の名前と同一になります。レポートを表示または印刷する場合、これらの名前が抽出されます。AS 句と SET ASNAMES を組み合わせて使用すると、出力マスターファイルのフィールド名を変更することができます。

- HOLD コマンドでは、他のソフトウェアアプリケーションでレポート出力を表示または処理するために適切なフォーマットを指定することができます。詳細は、出力ファイルフォーマットの選択を参照してください。

- アプリケーションで HOLD オプションにないデータフォーマットが必要な場合は、サブルーチンを使用して、HOLD データソースに書き込まれる際に、各出力レコードを処理することができます。

使用する環境が SET パラメータの SAVEMATRIX をサポートする場合、リクエストの後にダイアログマネージャコマンドを発行して、最終レポートの内部マトリックスを保持することで、それに続く HOLD、SAVE、SAVEB コマンドで使用可能にすることができます。SAVEMATRIX についての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

HOLD ファイルを作成するには

リクエストでは、次の構文を使用します。

```
ON TABLE HOLD [AS filename] [FORMAT fmt] [DATASET dataset]
       [MISSING {ON|OFF}]
       [PERSISTENCE {STAGE|PERMANENT}]
```

または

```
hold_field HOLD [AS filename] [FORMAT fmt] [DATASET dataset]
       [MISSING {ON|OFF}]
       [PERSISTENCE {STAGE|PERMANENT}]
```

説明

HOLD

レポート出力を抽出、保存します。明示的なフォーマット指定をせずに HOLD コマンドを発行すると、BINARY がデフォルトフォーマットとして使用されます。レポート出力は、関連するマスターファイルとともに保存されます。

注意：デフォルト出力フォーマットを ALPHA に変更するには、SET HOLDFORMAT コマンドを発行します。

hold_field

リクエストの最終表示フィールドの名前です。

AS filename

HOLD ファイルの名前を指定します。ファイル名を指定しない場合、デフォルト名の「HOLD」になります。後に続く各 HOLD コマンドは前の HOLD ファイルを上書きするため、各リクエストで固有のファイル名を記述して抽出データを別のファイルに保存し、次の HOLD コマンドによる上書きを防止します。

この名前には、オペレーティングシステムでサポートされる最大バイト数までの文字を使用することができます。

FORMAT fmt

HOLD 出力ファイルのフォーマットを指定します。BINARY が Reporting Server のデフォルトフォーマットです。

- Web ページとして表示するには、HTML、HTMTABLE、DHTML を選択します。
- 印刷ドキュメントとして表示するには、PDF を選択します。
- テキストドキュメントとして使用するには、ALPHA、DOC、WP を選択します。
- スプレッドシートアプリケーションで使用するには、DIF、EXCEL、EXL97、EXL2K

[PIVOT]、LOTUS、SYLK を選択します。

- データベースアプリケーションで使用するには、COMMA、COM、COMT、DB2、DATREC、DFIX、FOCUS、INGRES、REDBRICK、SQL、SQLDBC、SQLORA、SQLINF、SQLMSS、SQLSYB、SQLODBC、TAB、TABT、XFOCUS を選択します。
- 3 GL プログラムで使用するには、INTERNAL を選択します。
- 別のレポートで使用するには、ALPHA、BINARY、FOCUS を選択します。
- データソースを変更するためのトランザクションファイルとして使用するには、ALPHA、BINARY を選択します。
- サーバに接続せずにインタラクティブ分析で使用するには、AHTML、APDF、FLEX を選択します。

使用可能なフォーマットについての詳細は、出力ファイルフォーマットの選択を参照してください。

dataset

完全修飾名のデータセット、ファイル、n 構成要素 (app/.../filename.ext) のいずれかにすることができます。

外部またはデフォルト設定に依存せず、リクエスト自体にレポート出力を含めるようにデータセットまたはファイルを指定することができます。これにより、書き込み可能な任意のフォルダ、ディレクトリ、またはデータセットに永続的な HOLD ファイルを格納することができます。この場合、その格納場所が APP PATHに含まれているかどうかは関係ありません。対応する HOLD マスターファイルには、生成されたファイルを指定する DATASET 属性が含まれます。

マスターファイルにデータセットの名前が大文字で追加されるか、小文字で追加されるかは、FILECASE パラメータの値によって異なります。デフォルト設定では、小文字が使用されます。実際のデータセットは、使用するオペレーティングシステムの標準に準拠した名前で作成されます。

注意

- DATASET 属性に USS 名前規則を使用して、USS 環境にファイルを保存することができます (テキスト出力タイプの場合)。

MISSING

マスターファイルの MISSING=ON 属性を含むフィールドを HOLD ファイルに継承するかどうかを制御します。MISSING ON がデフォルト属性です。HOLD コマンドにより MISSING OFF を指定すると、MISSING 属性を持つフィールドは継承されません。詳細は、ミッシングフィールド値を含むレコード処理を参照してください。SET HOLDMISS、SET NULL、SET HNODATA パラメータを使用してミッシング値を文字ファイルおよびカ

ンマ区切りファイルに継承する方法についての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

PERSISTENCE

リレーショナルフォーマットの HOLD ファイル (FORMAT *sqlengine*) にのみ適用されます。UPLOAD および EBL リクエストの実行中にのみ使用される中間テーブルの生成方法を指定します。この方法では、HOLD ファイルにデータをダウンロードする代わりに、すべての処理が DBMS サーバ上で継続するため、パフォーマンスが向上します。実際に生成される中間テーブルのタイプは、DBMS でサポートされる特定の機能と、使用されるデータ挿入メカニズムに基づいて実行時に決定されます。有効な値には、次のものがあります。

- **STAGE** この機能をサポートする DBMS では、一時テーブルまたはグローバル一時テーブルのいずれかを作成します。この機能をサポートしない DBMS の場合、メッセージが表示され、テーブルの作成は実行されません。
- **PERMANENT** 一意に生成される名前の付いた通常の SQL テーブルが作成されます。このテーブルがリクエストで使用されるとともに、リクエストの終了後にも使用可能になりますが、セッション終了時には削除されます。HOLD FORMAT *sqlengine* で PERSISTENCE 使用する際は、これがデフォルト値です。

ホットスクリーンからの HOLD ファイルの作成

FOCUS コマンドラインまたはホットスクリーンから、レポートの実行後に次を使用します。

```
HOLD [AS filename][FORMAT fmt][MISSING {ON|OFF}][VIAprogram]
```

説明

fmt

ALPHA、BINARY、DIF、SYLK、WP、INTERNAL、FOCUS のいずれかです。フォーマットを省略すると、デフォルトの HOLD フォーマット (BINARY) が使用されます。

レポートが表示されている間は、任意のレポートページでいつでもホットスクリーンコマンドラインからの HOLD の発行が可能です。どのページでコマンドを発行しても、全レポートが保存され、そのレポートのデータソースおよびマスターファイルが作成されます。これは、TABLE リクエスト内やホットスクリーン終了後に HOLD コマンドを発行した場合も同様です。

1 つの TABLE リクエストで複数の HOLD コマンドを発行することができますが、ホットスクリーンから HOLD コマンドで FOCUS フォーマットを指定する場合は、ホットスクリーンセッションの実行中は、追加の HOLD コマンドを発行することはできません。

ホットスクリーンからは、SAVE および SAVB コマンドを発行することはできません。これらのコマンドは、リクエストに追加するか、ホットスクリーンの終了後 FOCUS コマンドレベルで発行する必要があります。

HOLD での DATASET の使用

次のリクエストは、test1 という名前のアプリケーションに ggsum.ftm ファイルを生成します。

```
APP HOLD test1
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY CATEGORY
BY PRODUCT
ON TABLE HOLD AS GGCOMT FORMAT COMT
DATASET test1/ggsum.ftm
END
```

生成されたマスターファイルには、DATASET 属性が含まれます。

```
FILENAME=GGCOMT  , SUFFIX=COMT    ,
 DATASET=test1/ggsum.ftm (LRECL 80 RECFM V, $
  SEGMENT=GGCOMT, SEGTYPE=S2, $
    FIELDNAME=CATEGORY, ALIAS=E01, USAGE=A11, ACTUAL=A11, $
    FIELDNAME=PRODUCT, ALIAS=E02, USAGE=A16, ACTUAL=A16, $
    FIELDNAME=UNITS, ALIAS=E03, USAGE=I08, ACTUAL=A08, $
    FIELDNAME=DOLLARS, ALIAS=E04, USAGE=I08, ACTUAL=A08, $
```

次のリクエストは、z/OS で USER1.GGHOLD.FTM データセットを生成します。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY CATEGORY
BY PRODUCT
ON TABLE HOLD AS GGCOMT FORMAT COMT
DATASET user1.gghold.ftm
END
```

生成されたマスターファイルには、DATASET 属性が含まれます。

```
FILENAME=GGCOMT  , SUFFIX=COMT    ,
 DATASET=user1.gghold.ftm (LRECL 80 RECFM V, $
  SEGMENT=GGCOMT, SEGTYPE=S2, $
    FIELDNAME=CATEGORY, ALIAS=E01, USAGE=A11, ACTUAL=A11, $
    FIELDNAME=PRODUCT, ALIAS=E02, USAGE=A16, ACTUAL=A16, $
    FIELDNAME=UNITS, ALIAS=E03, USAGE=I08, ACTUAL=A08, $
    FIELDNAME=DOLLARS, ALIAS=E04, USAGE=I08, ACTUAL=A08, $
```

デフォルト HOLD フォーマットの設定

```
SET HOLDFORMAT = {BINARY|ALPHA}
```

または

```
ON TABLE SET HOLDFORMAT {BINARY|ALPHA}
```

説明

BINARY

デフォルトの HOLD ファイルフォーマットを BINARY に設定します。

ALPHA

デフォルトの HOLD ファイルフォーマットを ALPHA に設定します。

HOLD ファイルへのデータの抽出

次のリクエストは、データを EMPLOYEE データソースから抽出し、HOLD ファイルを作成します。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL AND ED_HRS
BY DEPARTMENT
LIST CURR_SAL AND ED_HRS AND BANK_ACCT
BY DEPARTMENT
BY LAST_NAME BY FIRST_NAME
ON TABLE HOLD
END
```

次のメッセージが表示されます。

```
NUMBER OF RECORDS IN TABLE=  12 LINES=     12
```

このリクエストにより作成されたレポートを表示するには、HOLD ファイルに対してリクエストを発行します。

ヒント：レポートを実行する前に、HOLD マスターファイルの情報を表示するには、? HOLD を発行します。

HOLD マスターファイルのクエリ

選択した HOLD フォーマットオプションによりマスターファイルが作成された場合、次のコマンドを発行して、HOLD マスターファイル内のフィールド、エイリアス、フォーマットを表示することができます。

```
?HOLD
```

このコマンドは、32 バイト以内のフィールド名を表示します。フィールド名が 32 バイトを超える場合は、32 番目の位置にキャレット記号 (>) が表示され、フィールド名が長いことが示されます。

AS *filename* を使用して HOLD ファイルの名前を変更した場合は、次の構文を使用します。

```
?HOLD filename
```

ヒント：? HOLD クエリは、HOLD ファイルを作成した同一セッション内で、発行する必要があります。

HOLD マスターファイルに対するレポートの実行

次の HOLD ファイルでは、表示フォーマットは FORMAT 属性の値です。任意のテキストエディタを使用して、HOLD マスターファイルおよび ACTUAL 属性の値を表示することができます。テキストフィールドの USAGE および ACTUAL フォーマットは、HOLD ファイルの各論理レコードの 1 行目の長さのみを指定します。USAGE フォーマットは元のマスターファイルのフィールドフォーマットと同一です。ACTUAL フォーマットは文字フィールドと同様に、4 バイト (4 の倍数) まで切り上げられます。

次のリクエストには、関連マスターファイル内のフィールド、エイリアス、およびフォーマットを表示する？HOLD コマンドが含まれてます。このリクエストは、HOLD ファイルを作成します。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL AND ED_HRS
BY DEPARTMENT
LIST CURR_SAL AND ED_HRS AND BANK_ACCT
BY DEPARTMENT
BY LAST_NAME BY FIRST_NAME
ON TABLE HOLD
END

?HOLD
```

出力結果は次のとおりです。

NUMBER OF RECORDS IN TABLE= 12			LINES=	12
DEFINITION OF HOLD FILE:HOLD				
FIELDNAME			ALIAS	FORMAT
DEPARTMENT			E01	A10
CURR_SAL			E02	D12.2M

ED_HRS			E03	F6.2
LAST_NAME			E04	A15
FIRST_NAME			E05	A10
LIST			E06	I5
CURR_SAL			E07	D12.2M
ED_HRS			E08	F6.2
BANK_ACCT			E09	I9S

HOLD ファイルに対して発行できるリクエストは次のとおりです。

```
TABLE FILE HOLD
PRINT E07 AS 'SALARY OF,EMPLOYEE' AND LAST_NAME AND FIRST_NAME
BY HIGHEST E03 AS 'TOTAL,DEPT,ED_HRS'
BY E01
BY HIGHEST E08 AS 'EMPLOYEE,ED_HRS'
END
```

出力結果は次のとおりです。

```
TOTAL
DEPT                EMPLOYEE        SALARY OF
ED_HRS  DEPARTMENT  ED_HRS          EMPLOYEE   LAST_NAME        FIRST_NAME
------  ----------  --------        ---------  ---------        ----------
231.00  MIS            75.00       $21,780.00  BLACKWOOD        ROSEMARIE
                       50.00       $18,480.00  JONES            DIANE
                       45.00       $27,062.00  CROSS            BARBARA
                       36.00       $13,200.00  SMITH            MARY
```

```
                        25.00        $9,000.00  GREENSPAN        MARY
                          .00       $18,480.00  MCCOY            JOHN
 120.00  PRODUCTION     50.00       $16,100.00  MCKNIGHT         ROGER
                        30.00       $26,862.00  IRVING           JOAN
                        25.00       $11,000.00  STEVENS          ALFRED
                        10.00        $9,500.00  SMITH            RICHARD
                         5.00       $21,120.00  ROMANS           ANTHONY
                          .00       $29,700.00  BANNING          JOHN
```

ibi FOCUS フォーマットによるレポート出力の保存

HOLD コマンドの発行がリクエスト内、リクエスト実行後のいずれの場合も、HOLD コマンドを使用して、リクエストで抽出されたデータから、FOCUS データソースおよび関連マスターファイルを作成することができます。この機能を使用して、次のものを作成することができます。

- サポートされる FOCUS 以外の任意のタイプのデータソースから FOCUS データソース
- 既存の FOCUS データソースのサブセット

注意：FOCUS フォーマットにファイルを保持すると、「警告。すでにファイルが存在します。CREATE しますか?」という警告 (FOC441) が表示されます。SET WARNING=OFF コマンドを発行すると、このメッセージを非表示にすることができます。

```
The file exists already. Create will write over it.
```

ibi FOCUS フォーマットによる HOLD ファイルの作成

リクエストでは、次の構文を使用します。

```
ON TABLE HOLD [AS filename] FORMAT FOCUS [INDEX field1 field2 ...]
```

説明

AS filename

HOLD ファイルの名前を指定します。ファイル名を指定しない場合、デフォルト名の「HOLD」になります。後に続く各 HOLD コマンドは前の HOLD ファイルを上書きするため、各リクエストで固有のファイル名を記述して抽出データを別のファイルに保存し、次の HOLD コマンドによる上書きを防止します。

名前は 64 バイト以内で指定します。

INDEX field1...

FOCUS フィールドにインデックスを付けることを可能にします。INDEX の後に指定されるフィールドは、すべてマスターファイルで「FIELDTYPE=I」と指定されます。最大で 4 つのフィールドにインデックスを付けることができます。

注意：インデックスフィールド名の数は変数であるため、HOLD コマンドに続き、フィールド名と同一文字列で始まるコマンド名が、別のインデックスフィールドとしてカウントされ、エラーが発生する場合があります。たとえば、HOLD の後に続くコマンドが、ON TABLE で始まり、フィールド名が文字列「ON」で始まる場合、このコマンドの「ON」がフィールド名の短縮形と見なされ、インデックスに追加されます。この問題を回避するため、FIELDNAME パラメータを NOTRUNC に設定し、コマンド名がフィールド名の短縮名と混同されないようにするか、HOLD コマンドを END コマンド直前のプロシジャの末尾に移動します。

ibi FOCUS ファイル構造の制御

FOCUS データソースの構造は、リクエストにより異なります。次の規則が適用されます。

- 各集計コマンド (SUM、COUNT、WRITE) は、リクエストの新しい BY フィールドそれぞれから 1 つのセグメントを作成します。これらのフィールドはキーフィールドになります。複数の表示コマンドを使用するリクエストでは、新しく作成したセグメントへのキーに親セグメントのキーは含まれません。
- PRINT または LIST コマンドを使用してセグメントを作成する場合は、すべての BY フィールドと内部 FOCLIST フィールドからキーが生成されます。
- INDEX 設定後に指定したすべてのフィールドにはインデックスが付けられます。つまり、マスターファイルに FIELDTYPE=I として指定されます。最大で 4 つのフィールドにインデックスを付けることができます。
- HOLD ファイルのデータが 1 ページを超える (FOCUS データソースでは 4 キロバイト、XFOCUS データソースでは 16 キロバイト) 場合、単一セグメントとして格納することはできません。単一セグメントとして大きすぎるデータは、ユニーク子セグメン

トを含む親セグメントになります。FOCUS データソースでは、フィールドは通常の FOCUS ページサイズセグメントにグループ化され、最大 32 キロバイトのユニークセグメントとしてデータに追加されます。XFOCUS データソースでは、ルートセグメントは最初の 16 キロバイトのデータを保持することができ、最大合計 32 キロバイトの追加データは一意の単一セグメントに追加されます。BY フィールドは、すべてルートセグメントに割り当てられるデータ部分に配置する必要があります。

HOLD ファイルに関連付けられたマスターファイルに ACCEPT 属性および TITLE 属性を継承させるかどうかを制御するには、SET HOLDATTR コマンドを使用します。HOLD マスターファイルの FIELDNAME 属性を制御するには、SET ASNAMES コマンドを使用します。HOLD マスターファイルでの TITLE、ACCEPT、および FIELDNAME 属性の制御方法についての詳細は、HOLD マスターファイルの属性制御を参照してください。

ibi FOCUS フォーマット HOLD ファイルの作成

次の例では、CAR データソースのサブセットを作成します。

```
TABLE FILE CAR
SUM SALES BY COUNTRY BY CAR BY MODEL
ON TABLE HOLD AS X1 FORMAT FOCUS
END
```

このリクエストは「X1」という名前の SEGTYPE S3 を持った単一セグメント FOCUS データソースを作成します。

リクエストにより、X1 マスターファイルが作成されます。

```
FILE=X1, SUFFIX=FOC
 SEGMENT=SEG01 ,SEGTYPE=S03
  FIELDNAME=COUNTRY      ,ALIAS=E01    ,USAGE=A10   ,$
  FIELDNAME=CAR          ,ALIAS=E02    ,USAGE=A16   ,$
  FIELDNAME=MODEL        ,ALIAS=E03    ,USAGE=A24   ,$
  FIELDNAME=SALES        ,ALIAS=E04    ,USAGE=I6    ,$
```

PRINT を使用した FOCLIST フィールドによる FOCUS データソースの作成

次の例では、3 つの BY フィールドと FOCLIST フィールドがあるため、SEGTYPE S4 の単一セグメント FOCUS データソースが作成されます。

```
TABLE FILE CAR
PRINT SALES BY COUNTRY BY CAR BY MODEL
ON TABLE HOLD AS X2 FORMAT FOCUS INDEX MODEL
END
```

このリクエストで作成されるマスターファイルは次のとおりです。

```
FILE=X2, SUFFIX=FOC
 SEGMENT=SEG01, SEGTYPE=S04
  FIELDNAME=COUNTRY      ,ALIAS=E01    ,USAGE=A10   ,$
  FIELDNAME=CAR          ,ALIAS=E02    ,USAGE=A16   ,$
  FIELDNAME=MODEL        ,ALIAS=E03    ,USAGE=A24   ,FIELDTYPE=I,$
  FIELDNAME=FOCLIST      ,ALIAS=E04    ,USAGE=I5    ,$
  FIELDNAME=SALES        ,ALIAS=E05    ,USAGE=I6    ,$
```

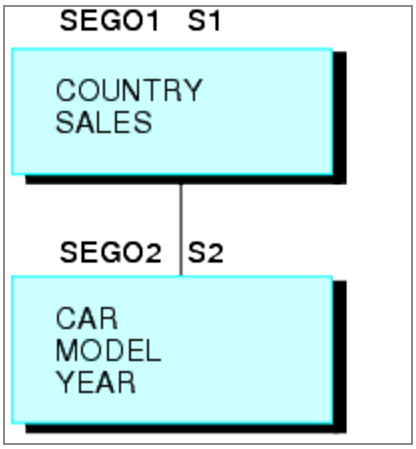
2 セグメント FOCUS データソースの作成

次のリクエストには 2 つの SUM コマンドが含まれています。まず、SUM SALES BY COUNTRY は COUNTRY をキーとし、SALES の集計値をデータフィールドとしたセグメントを作成します。次に、SUM SALES BY COUNTRY BY CAR BY MODEL は CAR および MODEL をキーとし、SALES をキーなしフィールドとした下位セグメントを作成します。

COUNTRY フィールドは 2 つ目のセグメントのキーの一部を形成しません。COUNTRY は 2 つ目のセグメントへのパスのキーで、この値の繰り返しは必要ありません。

```
TABLE FILE CAR
SUM SALES BY COUNTRY
SUM SALES BY COUNTRY BY CAR BY MODEL
ON TABLE HOLD AS X3 FORMAT FOCUS
END
```

これにより、2 セグメントで構成される FOCUS データソースが作成されます。



新しく作成された FOCUS データソースのマスターファイルは次のとおりです。

```
FILE=X3, SUFFIX=FOC
 SEGMENT=SEG01, SEGTYPE=S01
  FIELDNAME=COUNTRY       ,ALIAS=E01    ,USAGE=A10    ,$
  FIELDNAME=SALES         ,ALIAS=E02    ,USAGE=I6     ,$
 SEGMENT=SEG02, SEGTYPE=S02,PARENT=SEG01
  FIELDNAME=CAR           ,ALIAS=E03    ,USAGE=A16    ,$
  FIELDNAME=MODEL         ,ALIAS=E04    ,USAGE=A24    ,$
  FIELDNAME=SALES         ,ALIAS=E05    ,USAGE=I6     ,$
```

3 セグメント FOCUS データソースの作成

次の例では、表示コマンドごとに 1 つのセグメントが作成されます。

ルートセグメントへのキーは BY フィールドの COUNTRY で、下位セグメントへのキーは新しい BY フィールドです。表示コマンドが PRINT であるため、最終セグメントは内部 FOCLIST フィールドをキーの一部として使用します。

```
TABLE FILE CAR
SUM SALES BY COUNTRY BY CAR
SUM SALES BY COUNTRY BY CAR BY MODEL
PRINT SALES BY COUNTRY BY CAR BY MODEL BY BODY
ON TABLE HOLD AS X4 FORMAT FOCUS INDEX COUNTRY MODEL
END
```

マスターファイルは次のとおりです。

```
FILE=X4, SUFFIX=FOC
 SEGMENT=SEG01, SEGTYPE =S02
  FIELDNAME=COUNTRY    ,ALIAS=E01      ,USAGE=A10    ,FIELDTYPE=I,$
  FIELDNAME=CAR        ,ALIAS=E02      ,USAGE=A16    ,$
  FIELDNAME=SALES      ,ALIAS=E03      ,USAGE=I6     ,$
 SEGMENT=SEG02, SEGTYPE =S01 ,PARENT=SEG01
  FIELDNAME=MODEL      ,ALIAS=E04      ,USAGE=A24    ,FIELDTYPE=I,$
  FIELDNAME=SALES      ,ALIAS=E05      ,USAGE=I6     ,$
 SEGMENT=SEG03, SEGTYPE =S02 ,PARENT=SEG02
  FIELDNAME=BODYTYPE   ,ALIAS=E06      ,USAGE=A12    ,$
  FIELDNAME=FOCLIST    ,ALIAS=E07      ,USAGE=I5     ,$
  FIELDNAME=SALES      ,ALIAS=E08      ,USAGE=I6     ,$
```

HOLD マスターファイルの属性制御

SET ASNAMES、SET HOLDLIST、SET HOLDATTR コマンドにより、HOLD マスターファイルの FIELDNAME、TITLE、ACCEPT 属性を制御することができます。これらのコマンドはリクエストの前に発行し、変更しない限り、セッション中は有効です。

- SET ASNAMES コマンドは HOLD マスターファイルのフィールド名として AS 句で指定したテキストを割り当て、ACROSS 句で指定した 1つ目のフィールド名の先頭に連結します。
- SET HOLDLIST コマンドは HOLD ファイルおよび PCHOLD ファイルのフィールドをリクエストで表示されているものに制限します。つまり、リクエストで表示されていないフィールド (NOPRINT フィールドとして割り当てられているもの) は HOLD ファイルに含まれません。また、暗示的な非表示フィールドと明示的な非表示フィールドを区別することができます。詳細は、HOLD マスターファイルの属性制御を参照してください。
- SET HOLDATTR コマンドは、元のマスターファイルで使用されている TITLE および ACCEPT 属性を HOLD マスターファイルに継承します。詳細は、HOLD マスターファイルの属性制御を参照してください。

また、SET HOLDSTAT コマンドを使用して、HOLD マスターファイルにコメントおよび DBA 情報を含めることができます。SET コマンドについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

HOLD マスターファイルでのフィールド名の制御

SET ASNAMES を ON、MIXED、FOCUS のいずれかに設定すると、リクエストの AS 句で指定したリテラルが HOLD マスターファイルでフィールド名として使用されます。このコマンドにより、HOLD ファイルでの ACROSS フィールド名の付け方も制御します。

HOLD マスターファイルでのフィールド名の制御

```
SET ASNAMES = [ON|OFF|MIXED|FOCUS|FLIP]
```

説明

OFF

AS 句で指定したリテラルを HOLD ファイルのフィールド名として使用しないため、ACROSS フィールドの名前には影響しません。

ON

AS 句で指定したリテラルを大文字に変換し、それをフィールド名として HOLD マスターファイルに格納します。ACROSS フィールド値の先頭に AS 名を連結した ACROSS フィールド名を作成し、すべてのフォーマットの HOLD ファイルを対象として ACROSS フィールドの名前を制御します。

MIXED

AS 句で指定したリテラルをフィールド名として使用し、AS 名の大文字と小文字を区別したまま、ACROSS フィールド値の先頭に AS 名を連結した ACROSS フィールド名を作成します。

FOCUS

AS 句で指定したリテラルをフィールド名として使用し、FOCUS フォーマットの HOLD ファイルのみを対象として ACROSS フィールドの名前を制御します。デフォルト値は FOCUS です。

FLIP

元のマスターファイルのフィールド名を HOLD マスターファイルのエイリアス名に継承し、元のマスターファイルのエイリアス名を HOLD マスターファイルのフィールド名に継承します。

HOLD ファイルのフィールド名制御使用上の注意

- AS 句を使用してフィールドを指定しない場合は、元のマスターファイルのフィールド名が使用されます。マスターファイルで指定した TITLE 属性は、SET HOLDATTRS を事前に発行しない限り使用されません。
- リクエストでフィールドが複数回参照されている場合、HOLD マスターファイルでこれらのフィールドに一意の名前を割り当てるようにするには、SET ASNAMES を使用します。
- AS 句で使用した特殊文字およびブランクは、SET ASNAMES で作成したフィールド名に保持されます。新しく作成するマスターファイル内で参照する標準以外のフィールド名は、一重引用符 (') で囲みます。
- AS 句で指定したテキストが 66 バイトを超える場合、マスターファイルでは 66 バイトに切り捨てられます。
- ASNAMES パラメータのデフォルト設定で HOLD ファイルのフィールド名およびエイリアス名を生成する場合、リレーショナルデータソースの HOLD ファイルでは、元のマスターファイルのフィールド名およびエイリアス名が HOLD マスターファイルに継承されます。エイリアス名は、生成されたリレーショナルテーブルのフィールド名になります。AS 名は、HOLD マスターファイルの TITLE 属性になります。

  ASNAMES パラメータを FLIP に設定すると、リレーショナルデータソースの HOLD ファイルでは、元のマスターファイルのフィールド名またはリクエストで指定された AS 名が HOLD マスターファイルのエイリアス名になるとともに、生成されたリレーショナルテーブルのフィールド名、HOLD マスターファイルの TITLE 属性にもなります。元のマスターファイルのエイリアス名は HOLD マスターファイルのフィールド名になりますが、例外として、AS 名が指定されている場合は、元のフィールド名が HOLD ファイルのフィールド名になります。
- 非リレーショナル HOLD ファイルの場合、元のマスターファイルのフィールド名は HOLD マスターファイルに継承されますが、エイリアス名は継承されず、HOLD マス

ターファイルではデフォルトのエイリアス (E01、E02 の形式) が生成されます。

- SET ASNAMES=FLIP の場合、非リレーショナル HOLD ファイルでは、元のマスターファイルのフィールド名またはリクエストで指定された AS 名が HOLD マスターファイルのエイリアス名になり、デフォルトのフィールド名 (F01、F02 の形式) が生成されます。
- 切り捨ての結果または AS 句の指定方法により、新しく作成するマスターファイルにフィールド名の重複が発生することがあります。この場合は、エイリアス (E01、E02 など) を使用してフィールドを参照します。
- フィールド見出し行の区切り文字としてカンマ (,) を使用するときは、1 つ目のカンマ (,) までのリテラルのみがマスターファイルのフィールド名として使用されます。次の構文は、HOLD マスターファイルに「PLACE」というフィールド名を作成します。

  ```
  PRINT COUNTRY AS 'PLACE,OF,ORIGIN'
  ```

- リクエストで ACROSS を使用し、その結果を HOLD ファイルに抽出するときは、ACROSS 句で生成したフィールドはすべて HOLD マスターファイルの同一フィールド名になります。SET ASNAMES を発行すると、新しいフィールド名を、それぞれ一意にすることができます。この一意のフィールド名は、リクエストの表示コマンドで指定した ASNAME 値で構成され、ACROSS 句で使用したフィールド値の先頭に連結されます。複数のフィールド名が同一文字の場合、この方法は機能しません。

  ACROSS 句のフィールドに AS 句を使用すると、新しいフィールド名はそれぞれ ACROSS 句で使用した 1 つ目のフィールド値の先頭に連結される、AS 句のリテラルで構成される名前になります。

HOLD マスターファイルでのフィールド名の制御

次の例は、SET ASNAMES=ON により、AS 句のテキストが HOLD1 マスターファイルのフィールド名として使用されます。HOLD1 マスターファイルの 2 つのフィールドである NATION と AUTOMOBILE に、COUNTRY と CAR のデータがそれぞれ含まれます。

```
SET ASNAMES=ON
TABLE FILE CAR
PRINT CAR AS 'AUTOMOBILE'
BY COUNTRY AS 'NATION'
```

```
ON TABLE HOLD AS HOLD1
END
```

このリクエストにより作成されるマスターファイルは次のとおりです。

```
FILE=HOLD1, SUFFIX=FIX
 SEGMENT=HOLD1, SEGTYPE=S01,$
  FIELDNAME=NATION      ,ALIAS=E01   ,USAGE=A10  ,ACTUAL=A12     ,$
  FIELDNAME=AUTOMOBILE  ,ALIAS=E02   ,USAGE=A16  ,ACTUAL=A16     ,$
```

SET ASNAMES による一意のフィールド名の作成

次のリクエストで生成される HOLD マスターファイルでは、SALES および AVE.SALES のそれぞれに一意のフィールド名が付けられます。SET ASNAMES を使用しない場合、SALES および AVE.SALES の両方に「SALES」という名前が付けられます。

```
SET ASNAMES=ON
TABLE FILE CAR
SUM SALES AND AVE.SALES AS 'AVERAGESALES'
BY CAR
ON TABLE HOLD AS HOLD2
END
```

このリクエストにより作成されるマスターファイルは次のとおりです。

```
FILE=HOLD2, SUFFIX=FIX
 SEGMENT=HOLD2, SEGTYPE=S01,$
  FIELDNAME=CAR             ,ALIAS=E01  ,USAGE=A16 ,ACTUAL=A16   ,$
  FIELDNAME=SALES           ,ALIAS=E02  ,USAGE=I6  ,ACTUAL=I04   ,$
  FIELDNAME=AVERAGESALES    ,ALIAS=E03  ,USAGE=I6  ,ACTUAL=I04   ,$
```

SET ASNAMES の ACROSS 句への使用

次のリクエストは、リテラル CASH を COUNTRY の各値と連結して HOLD マスターファイルを作成します。

```
SET ASNAMES=ON
TABLE FILE CAR
SUM SALES AS 'CASH'
ACROSS COUNTRY
ON TABLE HOLD AS HOLD3
END
```

このリクエストにより作成されるマスターファイルは次のとおりです。

```
FILE=HOLD3, SUFFIX=FIX
 SEGMENT=HOLD3, SEGTYPE=S01,$
  FIELDNAME=CASHENGLAND     ,ALIAS=E01   ,USAGE=I6  ,ACTUAL=I04  ,$
  FIELDNAME=CASHFRANCE      ,ALIAS=E02   ,USAGE=I6  ,ACTUAL=I04  ,$
  FIELDNAME=CASHITALY       ,ALIAS=E03   ,USAGE=I6  ,ACTUAL=I04  ,$
  FIELDNAME=CASHJAPAN       ,ALIAS=E04   ,USAGE=I6  ,ACTUAL=I04  ,$
  FIELDNAME=CASHW GERMANY   ,ALIAS=E05   ,USAGE=I6  ,ACTUAL=I04  ,$
```

SET ASNAMES コマンドを使用しない場合、HOLD FILE の各フィールドは「COUNTRY」という名前になります。

ACROSS 値のフィールド名をフィールド値のみの名前にするには、次のように AS 句の後ろに 2 つの一重引用符 ('') を記述します。

```
SET ASNAMES=ON
TABLE FILE CAR
SUM SALES AS ''
ACROSS COUNTRY
ON TABLE HOLD AS HOLD4
END
```

結果のマスターファイルは次のとおりです。

```
FILE=HOLD4, SUFFIX=FIX
 SEGMENT=HOLD4, SEGTYPE=S0,$
  FIELDNAME=ENGLAND    ,ALIAS=E01   ,USAGE=I6  ,ACTUAL=I04    ,$
  FIELDNAME=FRANCE     ,ALIAS=E02   ,USAGE=I6  ,ACTUALI04     ,$
  FIELDNAME=ITALY      ,ALIAS=E03   ,USAGE=I6  ,ACTUALI04     ,$
  FIELDNAME=JAPAN      ,ALIAS=E04   ,USAGE=I6  ,ACTUALI04     ,$
  FIELDNAME=W GERMANY  ,ALIAS=E05   ,USAGE=I6  ,ACTUALI04     ,$
```

SET ASNAMES=FLIP による HOLD ファイルの生成

次のリクエストは、SET ASNAMES を OFF に設定し、HOLD ファイルを ALPHA フォーマットで生成します。このリクエストでは、CURR_SAL フィールドの AS 名として SALARY が指定されています。

```
SET ASNAMES=OFF
TABLE FILE EMPLOYEE
SUM CURR_SAL AS SALARY PCT_INC
BY DEPARTMENT
ON TABLE HOLD FORMAT ALPHA
END
```

HOLD マスターファイルでは、AS 名は継承されていません。フィールド名は元のマスターファイルから継承され、デフォルトのエイリアス名が生成されています。

```
FILENAME=HOLD    , SUFFIX=FIX     , IOTYPE=STREAM, $
  SEGMENT=HOLD, SEGTYPE=S1, $
    FIELDNAME=DEPARTMENT, ALIAS=E01, USAGE=A10, ACTUAL=A10, $
    FIELDNAME=CURR_SAL, ALIAS=E02, USAGE=D12.2M, ACTUAL=A12, $
    FIELDNAME=PCT_INC, ALIAS=E03, USAGE=F6.2, ACTUAL=A06, $
```

次のリクエストは、リレーショナルテーブルを生成します。

```
SET ASNAMES=OFF
TABLE FILE EMPLOYEE
SUM CURR_SAL AS SALARY PCT_INC
BY DEPARTMENT
ON TABLE HOLD FORMAT SQLMSS
END
```

元のマスターファイルのフィールド名が HOLD マスターファイルのフィールド名に継承され、元のマスターファイルのエイリアス名が HOLD マスターファイルに継承されています。CURR_SAL の AS 名は、HOLD マスターファイルの TITLE 属性として追加されています。

```
FILENAME=HOLD   , SUFFIX=SQLMSS  , $
  SEGMENT=SEG01, SEGTYPE=S0, $
    FIELDNAME=DEPARTMENT, ALIAS=DPT, USAGE=A10, ACTUAL=A10, $
```

```
    FIELDNAME=CURR_SAL, ALIAS=CSAL, USAGE=D12.2M, ACTUAL=D8,
      TITLE='SALARY', $
    FIELDNAME=PCT_INC, ALIAS=PI, USAGE=F6.2, ACTUAL=F4, $
```

SET ASNAMES を ON に変更すると、AS 名の SALARY が HOLD マスターファイルのフィールド名に継承されます。以下は、ALPHA フォーマットで生成された HOLD マスターファイルを示しています。

```
FILENAME=HOLD    , SUFFIX=FIX     , IOTYPE=STREAM, $
  SEGMENT=HOLD, SEGTYPE=S1, $
    FIELDNAME=DEPARTMENT, ALIAS=E01, USAGE=A10, ACTUAL=A10, $
    FIELDNAME=SALARY, ALIAS=E02, USAGE=D12.2M, ACTUAL=A12, $
    FIELDNAME=PCT_INC, ALIAS=E03, USAGE=F6.2, ACTUAL=A06, $
```

以下は、リレーショナルフォーマットで生成された HOLD マスターファイルを示しています。

```
FILENAME=HOLD   , SUFFIX=SQLMSS  , $
  SEGMENT=SEG01, SEGTYPE=S0, $
    FIELDNAME=DEPARTMENT, ALIAS=DPT, USAGE=A10, ACTUAL=A10, $
    FIELDNAME=SALARY, ALIAS=CURR_SAL, USAGE=D12.2M, ACTUAL=D8,
      TITLE='SALARY', $
    FIELDNAME=PCT_INC, ALIAS=PI, USAGE=F6.2, ACTUAL=F4, $
```

SET ASNAMES を FLIP に変更すると、AS 名の SALARY が HOLD マスターファイルのエイリアス名に継承されます。ALPHA フォーマットの HOLD ファイルでは、その他のフィールド名は HOLD マスターファイルのエイリアス名に継承され、デフォルトのフィールド名が生成されます。

```
FILENAME=HOLD    , SUFFIX=FIX     , IOTYPE=STREAM, $
  SEGMENT=HOLD, SEGTYPE=S1, $
    FIELDNAME=F01, ALIAS=DEPARTMENT, USAGE=A10, ACTUAL=A10, $
    FIELDNAME=F02, ALIAS=SALARY, USAGE=D12.2M, ACTUAL=A12, $
    FIELDNAME=F03, ALIAS=PCT_INC, USAGE=F6.2, ACTUAL=A06, $
```

リレーショナルフォーマットの HOLD ファイルでは、SET ASNAMES を FLIP に変更すると、AS 名の SALARY が HOLD マスターファイルのエイリアス名に継承されます。このフィールドの場合、元のマスターファイルのフィールド名が HOLD マスターファイルのフィールド名および TITLE 属性になります。その他のフィールド名は HOLD マスターファイルのエイリアス名に継承され、元のマスターファイルのエイリアス名は HOLD マスターファイルのフィールド名に継承されます。

```
FILENAME=HOLD   , SUFFIX=SQLMSS  , $
  SEGMENT=SEG01, SEGTYPE=S0, $
    FIELDNAME=DPT, ALIAS=DEPARTMENT, USAGE=A10, ACTUAL=A10, $
    FIELDNAME=CURR_SAL, ALIAS=SALARY, USAGE=D12.2M, ACTUAL=D8,
      TITLE='SALARY', $
    FIELDNAME=PI, ALIAS=PCT_INC, USAGE=F6.2, ACTUAL=F4, $
```

HOLD マスターファイルのフィールド制御

SET HOLDLIST コマンドを使用し、HOLD マスターファイルのフィールドに制限を加えてリクエスト内に表示することができます。

HOLD ファイルのフィールド制御

```
SET HOLDLIST = {PRINTONLY|ALL|ALLKEYS|EXPLICIT}
```

説明

PRINTONLY

生成する HOLD ファイルに、レポートに表示されるフィールドのみを含めることを指定します。リクエストの非表示フィールド (明示的または暗示的に NOPRINT フィールドとして割り当てたフィールド) は、HOLD ファイルには格納されません。

ALL

リクエストで参照したすべての表示フィールドが、一時項目 (COMPUTE) を含めて HOLD ファイルに格納されることを指定します。デフォルト値は ALL です。OLD は ALL の同義語として使用することができます。

注意：リクエストで指定された BY フィールドに NOPRINT オプションが使用されている場合、その BY フィールドは HOLDLIST=ALL に設定した場合でも HOLD ファイルに格納されません。

ALLKEYS

NOPRINT オプションが指定された BY フィールドを含むすべてのフィールドを継承します。

ALLKEYS 設定により、Analytic Document の操作に必要なデータすべてのキャッシュを有効にすることができます。

EXPLICIT

リクエストで NOPRINT オプションを使用してレポート出力から明示的に除外したフィールドを HOLD または PCHOLD ファイルに格納します。ただし、NOPRINT が暗示的に指定されているフィールドは格納されません。たとえば、リクエストでフィールドのフォーマットを再設定した場合、新しいフォーマットのフィールドと、元のフォーマットのフィールド (NOPRINT が暗示的に指定された) の 2 つのフィールドが存在します。

TABLE リクエストでは、SET HOLDLIST を発行することもできます。MATCH とともに使用する場合、SET HOLDLIST の動作は、HOLDLIST を ALL に設定した場合と常に同一になります。

HOLDLIST=ALL の使用

HOLDLIST を ALL に設定すると、それに続く TABLE リクエストは、NOPRINT フィールドおよび COMPUTE コマンドによる一時項目 (COMPUTE) を含む、指定フィールドをすべて含んだ HOLD ファイルを作成します。

```
SET HOLDLIST=ALL

TABLE FILE CAR
PRINT CAR MODEL NOPRINT
COMPUTE TEMPSEATS=SEATS+1;
BY COUNTRY
ON TABLE HOLD
END

?HOLD
```

出力結果は次のとおりです。

```
 NUMBER OF RECORDS IN TABLE=     18      LINE=        18
```

```
DEFINITION OF HOLD FILE:HOLD
```

fieldname			ALIAS	FORMAT
COUNTRY			E01	A10
CAR			E02	A16
MODEL			E03	A24
SEATS			E04	I3
TEMPSEATS			E05	D12.2

HOLDLIST= PRINTONLY の使用

HOLDLIST を PRINTONLY に設定すると、それに続く TABLE リクエストは、レポート出力内に表示されるフィールドのみを含んだ HOLD ファイルを作成します。

```
SET HOLDLIST=PRINTONLY

TABLE FILE CAR
PRINT CAR MODEL NOPRINT
COMPUTE TEMPSEATS=SEATS+1;
BY COUNTRY
```

```
ON TABLE HOLD
END

?HOLD
```

出力結果は次のとおりです。

NUMBER OF RECORDS IN TABLE= 18			LINES=	18
DEFINITION OF HOLD FILE:HOLD				
fieldname			ALIAS	FORMAT
COUNTRY			E01	A10
CAR			E02	A16
TEMPSEATS			E03	D12.2

異なる HOLDLIST 設定で作成したマスターファイルの比較

次のリクエストは、GGSALES データソースに対して実行され、フォーマットが再設定された 2 つの表示フィールド (DOLLARS および UNITS) が指定されています。DOLLARS フィールドは非表示 (NOPRINT) フィールドです。CATEGORY という BY フィールドも非表示 (NOPRINT) フィールドです。

```
SET HOLDLIST=ALL
TABLE FILE GGSALES
SUM UNITS/I5 DOLLARS/D12.2 NOPRINT
BY REGION BY CATEGORY NOPRINT
ON TABLE HOLD FORMAT FOCUS
END
```

このリクエストを HOLDLIST=ALL で実行すると、次の HOLD マスターファイルが生成されます。DOLLARS および UNITS フィールドは、元のフォーマットで 1 回、新しいフォーマットで 1 回の計 2 回書き込まれています (このレポートを HOLD ファイルに格納するのではなく、表示した場合は、元のフォーマットのフィールドは非表示になります)。ただし、非表示 (NOPRINT) BY フィールド (CATEGORY) は書き込まれていません。

```
FILENAME=HOLD, SUFFIX=FOC     , $
  SEGMENT=SEG01, SEGTYPE=S1, $
    FIELDNAME=REGION, ALIAS=E01, USAGE=A11,
      TITLE='Region', DESCRIPTION='Region code', $
    FIELDNAME=UNITS, ALIAS=E02, USAGE=I08,
      TITLE='Unit Sales', DESCRIPTION='Number of units sold', $
    FIELDNAME=UNITS, ALIAS=E03, USAGE=I5,
      TITLE='Unit Sales', $
    FIELDNAME=DOLLARS, ALIAS=E04, USAGE=I08,
      TITLE='Dollar Sales', DESCRIPTION='Total dollar amount of reported
sales', $
    FIELDNAME=DOLLARS, ALIAS=E05, USAGE=D12.2,
      TITLE='Dollar Sales', $
```

このリクエストを HOLDLIST=ALLKEYS で実行すると、次の HOLD マスターファイルが生成されます。DOLLARS および UNITS フィールドは、元のフォーマットで 1 回、新しいフォーマットで 1 回の計 2 回書き込まれています (このレポートを HOLD ファイルに格納するのではなく、表示した場合は、元のフォーマットのフィールドは非表示になります)。非表示 (NOPRINT) BY フィールド (CATEGORY) は書き込まれています。

```
FILENAME=HOLD, SUFFIX=FOC     , $
  SEGMENT=SEG01, SEGTYPE=S2, $
    FIELDNAME=REGION, ALIAS=E01, USAGE=A11,
      TITLE='Region', DESCRIPTION='Region code', $
    FIELDNAME=CATEGORY, ALIAS=E02, USAGE=A11,
      TITLE='Category', DESCRIPTION='Product category', $
    FIELDNAME=UNITS, ALIAS=E03, USAGE=I08,
      TITLE='Unit Sales', DESCRIPTION='Number of units sold', $
    FIELDNAME=UNITS, ALIAS=E04, USAGE=I5,
      TITLE='Unit Sales', $
    FIELDNAME=DOLLARS, ALIAS=E05, USAGE=I08,
      TITLE='Dollar Sales', DESCRIPTION='Total dollar amount of reported
sales', $
    FIELDNAME=DOLLARS, ALIAS=E06, USAGE=D12.2,
      TITLE='Dollar Sales', $
```

このリクエストを HOLDLIST=PRINTONLY で実行すると、次の HOLD マスターファイルが生成されます。レポート出力に実際に表示された REGION および UNITS フィールドのみが、新しいフォーマット (I5) で書き込まれます。非表示 (NOPRINT) BY フィールド (CATEGORY) を含めて、非表示 (NOPRINT) フィールドとフォーマット設定前のフィールドはすべて除外されます。

```
FILENAME=HOLD , SUFFIX=FOC     , $
  SEGMENT=SEG01, SEGTYPE=S1, $
    FIELDNAME=REGION, ALIAS=E01, USAGE=A11,
      TITLE='Region', DESCRIPTION='Region code', $
    FIELDNAME=UNITS, ALIAS=E02, USAGE=I5,
      TITLE='Unit Sales', $
```

このリクエストを HOLDLIST=EXPLICIT で実行すると、次の HOLD マスターファイルが生成されます。レポート出力に実際に表示されたフィールドとともに、非表示 (NOPRINT) フィールド (表示フィールドの DOLLARS および BY フィールドの CATEGORY) が書き込まれます。元のフォーマットの DOLLARS および UNITS は除外されます。

```
FILENAME=HOLD, SUFFIX=FOC     , $
  SEGMENT=SEG01, SEGTYPE=S2, $
    FIELDNAME=REGION, ALIAS=E01, USAGE=A11,
      TITLE='Region', DESCRIPTION='Region code', $
    FIELDNAME=CATEGORY, ALIAS=E02, USAGE=A11,
      TITLE='Category', DESCRIPTION='Product category', $
    FIELDNAME=UNITS, ALIAS=E03, USAGE=I5,
      TITLE='Unit Sales', $
    FIELDNAME=DOLLARS, ALIAS=E04, USAGE=D12.2,
      TITLE='Dollar Sales', $
```

HOLD マスターファイルでの属性の制御

SET HOLDATTR コマンドは、元のマスターファイルの TITLE 属性、ACCEPT 属性、およびその他の属性を HOLD マスターファイルに継承するかどうかを制御します。SET HOLDATTR は HOLD マスターファイルでのフィールド名の付け方には影響しません。

データソースのフィールドにマスターファイルで指定した TITLE 属性がなく、リクエストで指定した AS 句がある場合、HOLD ファイルの関連フィールド名は AS 句により付けられます。

TITLE および ACCEPT 属性の制御

```
SET HOLDATTR =[ON|OFF|FOCUS|CUBE]
```

説明

ON

元のマスターファイルで指定された TITLE 属性を、任意のフォーマットの HOLD ファイルで使用します。ACCEPT 属性は、FOCUS フォーマットの HOLD ファイルの場合にのみ HOLD マスターファイルに継承されます。この場合、PROPERTY 属性も継承されます。

OFF

元のマスターファイルの TITLE および ACCEPT 属性を HOLD マスターファイルで使用しません。

FOCUS

FOCUS フォーマットの HOLD ファイルの場合にのみ TITLE および ACCEPT 属性を使用します。この場合、PROPERTY 属性も継承されます。デフォルト値は FOCUS です。

CUBE

フォルダおよび DV_ROLE 属性を継承します。また、TITLE 属性が HOLD マスターファイルに継承されます。フィールド名は、エイリアス値として継承されます。

HOLD マスターファイルでの TITLE および ACCEPT 属性の制御

次の例は、CAR データソースのマスターファイルが TITLE および ACCEPT 属性を指定しています。

```
FILENAME=CAR2, SUFFIX=FOC
SEGNAME=ORIGIN, SEGTYPE=S1
  FIELDNAME =COUNTRY, COUNTRY, A10, TITLE='COUNTRY OF ORIGIN',
             ACCEPT='CANADA' OR 'ENGLAND' OR 'FRANCE' OR 'ITALY' OR
                    'JAPAN' OR 'W GERMANY',
             FIELDTYPE=I,$
SEGNAME=COMP, SEGTYPE=S1, PARENT=ORIGIN
  FIELDNAME=CAR, CARS, A16, TITLE='NAME OF CAR',$
.
.
.
```

次のリクエストでは、SET HOLDATTR=FOCUS を使用しています。

```
SET HOLDATTR = FOCUS
TABLE FILE CAR2
PRINT CAR
BY COUNTRY ON TABLE HOLD FORMAT FOCUS AS HOLD5
END
```

次の HOLD マスターファイルが作成されます。

```
FILE=HOLD5, SUFFIX=FOC
 SEGMENT=SEG01, SEGTYPE=S02
  FIELDNAME=COUNTRY ,USAGE=E01   ,ACTUAL=A10
      TITLE='COUNTRY OF ORIGIN',
      ACCEPT=CANADA ENGLAND FRANCE ITALY JAPAN 'W GERMANY',$
  FIELDNAME=FOCLIST ,USAGE=E02   ,ACTUAL=I5     ,$
  FIELDNAME=CAR     ,USAGE=E03   ,ACTUAL=A16    ,
      TITLE='NAME OF CAR' ,$
```

HOLD ファイルからのキー検索

キー検索は任意の単一セグメントの SUFFIX=FIX データソース、またはキーに基づいてソートされた HOLD ファイルでサポートされます。キー検索は、シーケンシャルファイルの論理キーを構成するフィールドの識別に、マスターファイルの SEGTYPE パラメータを使用するため、抽出ファイルの読み取り時に発生する入出力を低減することができます。FIXRETRIEVE の設定に基づいて、動作は次のようになります。

- ON の場合、キーの等価テストまたは範囲テストが TRUE のときに検索プロセスは停止します。
- OFF の場合、最終レポートの作成時に、シーケンシャルファイルのすべてのレコードの読み込みおよび選別条件の適用が実行されます。

ON TABLE HOLD コマンドにより、サポートされる多くのデータソースの読み込み、および抽出ファイルの作成が可能になります。その後、これら固定フォーマットのシーケンシャルファイルを他のデータソースに結合し、データビューを絞り込むことができます。固定フォーマットファイルでの論理キーの概念により、修飾語を含んだキー検索が可能になります。SEGTYPE 属性で識別される最初の *n* KEY フィールドに対する IF/WHERE テストに一致するすべてのレコードが検索されます。選別テストの実行中に IF/WHERE テストで指定した値より大きい値が検出された場合、検索は停止します。

抽出ファイルに対してマスターファイルが作成される場合、リクエストの BY フィールドに基づいた S*n* または SH*n* のいずれかの SEGTYPE が追加されます。たとえば、PRINT *field* BY *field* により、SEGTYPE=S1 を含む HOLD マスターファイルが作成されます。BY HIGHEST *field* を使用すると、SEGTYPE=SH1 を含むマスターファイルが作成されます。

HOLD ファイルのキー検索制御

```
SET FIXRET[RIEVE] = {ON|OFF}
```

説明

ON

キー検索を有効にします。ON がデフォルト設定です。

OFF

キー検索を無効にします。

HOLD ファイルからのキー検索のマスターファイル

次のマスターファイルは、昇順値でソートされた 1 から 100,000 までの SEQ_NO を含む、固定フォーマットのシーケンシャルファイルを記述します。

```
FILE=SORTED,SUFFIX=FIX,$
SEGNAME=ONE,SEGTYPE=S1,$
 FIELD=MYKEY,MK,I8,I8,$
 FIELD=MFIELD,MF,A10,A10,$

TABLE FILE SORTED
 PRINT MFIELD
 WHERE MYKEY EQ 100
END
```

このインスタンスでは FIXRETRIEVE=ON が設定されており、MYKEY が 101 に達したところで、100,000 のうち 101 件のレコードのみが読み出されるため、入出力数が大幅に減少します。

抽出ファイルからのキー検索の選択条件

等価値のリストを含む選択条件では、キー検索を使用します。以下はその例です。

```
{IF|WHERE} MYKEY EQ x OR y OR z
```

IF および WHERE テストには、範囲テストを含めることもできます。以下はその例です。

```
{IF|WHERE} MYKEY IS-FROM x TO y
```

BY フィールドの最大数は 32 です。

この機能を使用する際に、ソートしていないレコードをソート HOLD ファイルに追加すると、順序が不正確なレコードは検索されません。たとえば、ソートされたファイルに、次の 3 件のレコードが含まれることを想定します。

```
Key
```

```
1 1200
```

```
2 2340
```

```
3 4875
```

ファイルの最下部に次のレコードを追加します。

```
1 1620
```

FIXRETRIEVE=ON に設定した場合、キー値の 2 が検出されると検索は直ちに停止するため、1 のキー値を持つ新しいレコードは除外されます (1200 のレコードは検索されますが、1620 のレコードは検索されません)。

HOLD ファイルの保存および検索

WebFOCUS では、HOLD ファイルは処理中に一時ディレクトリに保存され、サーバへの接続が切断されると削除されます。これらのファイルを後で使用する場合は、FILEDEF コマンドを使用して HOLD データソースおよび関連マスターファイルを特定の場所に保存することができます。

HOLD マスターファイルに対してレポートを後から実行するには、アプリケーションを APP PATH に追加するか (存在しない場合)、TABLE FILE コマンドで 2 部構成名 (*appname*/*mastername*) を使用します。

保存済みの HOLD データファイルに対してレポートを実行するには、TABLE FILE コマンドを発行する前に、ファイルの検索場所を指定する FILEDEF コマンドを発行します。

HOLD マスターファイルの格納場所の指定 (Windows および UNIX)

```
APP MAP appnamepath_to_directory APP HOLDDATA appname APP HOLDMETA appname
```

説明

APP MAP

ディレクトリパスをアプリケーション名に関連付けます。

APP HOLDDATA

HOLD データファイルのアプリケーション名を指定します。

APP HOLDMETA

HOLD マスターファイルのアプリケーション名を指定します。

path_to_directory

ファイルを格納するディレクトリを指定します。

appname

アプリケーションフォルダ名です。

これらの APP コマンドは、サポートされているプロファイルに追加することをお勧めします。これにより、アプリケーションディレクトリにアクセスする必要があるプロシジャごとに APP コマンドを発行しなくても、すべてのプロシジャで APP コマンドが使用可能になります。また、プロシジャがアプリケーションの場所を特定できるように、使用するパスにアプリケーションを追加することをお勧めします。

HOLD データファイルおよびマスターファイルの格納場所の指定

次の例は、UNIX 上の WebFOCUS で「holdapp」というアプリケーションを作成し、そのアプリケーションにマッピングされた ¥ggtmp ディレクトリに HOLD データファイルおよびマスターファイルを格納する方法を示しています。

サポートされているプロファイルに次のコマンドを追加します。

```
APP MAP holdapp /ggtmp
APP HOLDDATA holdapp
APP HOLDMETA holdapp
```

次のリクエストは、¥ggtmp ディレクトリに格納する「sales.mas」という HOLD マスターファイルと「sales.ftm」という HOLD データファイルを作成します。

```
TABLE FILE GGSALES
PRINT SEQ_NO CATEGORY PRODUCT
ON TABLE HOLD AS SALES
END
```

HOLD ファイルに対して TABLE リクエストを発行するには、この HOLD データファイルを指定する FILEDEF コマンドを最初に発行する必要があります。FILEDEF コマンドの DDNAME は、HOLD コマンドで指定された AS 名です。AS 名を指定しない場合、DDNAME は「HOLD」になります。

```
FILEDEF SALES DIR ¥ggtmp¥sales.ftm
```

また、WebFOCUS がこのマスターファイルの場所を特定できるようにする必要があります。その 1 つの方法として、このアプリケーションをアプリケーションパスに追加します。アプリケーションをアプリケーションパスに追加するには、次のコマンドを使用します。

```
APP APPENDPATH holdapp
```

ここで、次のように HOLD ファイルに対してリクエストを発行します。

```
TABLE FILE SALES
PRINT *
END
```

別の方法として、2 部構成名 (アプリケーション名とマスターファイル名) を使用して、HOLD ファイルに対して TABLE リクエストを発行することができます。

```
TABLE FILE holdapp/sales
PRINT *
END
```

データベース一時テーブルの HOLD ファイルとしての使用

レポート出力ファイルである HOLD ファイルを、ネイティブデータベースの一時テーブルとして作成することが可能です。これにより、すべてのレポート処理が DBMS サーバ上に保持

され、データをローカルコンピュータにダウンロードしてから DBMS サーバに再びアップロードする必要がなくなるため、パフォーマンスが向上します。

たとえば、別のプロシジャで即座に使用するレポート出力を一時的に保存する場合は、標準的な HOLD ファイルを作成する代わりに、一時テーブルとして保存しておくと、中間データをローカルコンピュータに転送する手間を省くことができます。

一時テーブルのフィールドは、次のレポート要素から作成されます。

- 表示フィールド
- BY フィールド
- COMPUTE フィールド

NOPRINT が指定されているフィールドは対象外です。

レポートから作成する一時テーブルは、レポートを作成したデータソースと同一データソースタイプ (同一 DBMS) になります。レポート作成元のデータソースに複数のテーブルが含まれている場合は、すべてのテーブルのデータソースタイプが同じで、DBMS サーバの同じインスタンス上に存在している必要があります。

テーブルの持続性は、数種類の中から選択できます。

次のアダプタを使用すると、ネイティブ DBMS テーブルとして抽出ファイルを作成することができます。

- Db2 (UNIX、Windows 上)
- Informix
- Microsoft SQL Server
- MySQL
- Oracle
- Teradata

コマンドによるネイティブ一時テーブルとしてのレポート出力の保存

レポート出力をネイティブ DBMS 一時テーブルとして保存する構文は、次のとおりです。

```
ON TABLE HOLD [AS filename] FORMAT SAME_DB [PERSISTENCE persistValue]
```

説明

filename

HOLD ファイル名を指定します。AS *filename* を省略すると、一時テーブルの名前はデフォルトの「HOLD」になります。

新しく発行される HOLD コマンドは 1 つ前の HOLD ファイルを上書きするため、各リクエストで名前を指定することをお勧めします。これにより、抽出されたデータは個別のファイルとして作成され、以前のファイルが新しいファイルにより上書きされる事態を回避できます。

PERSISTENCE

テーブルの持続タイプと関連したテーブルプロパティを指定します。これは、一時テーブルをサポートする DBMS ではオプションです。それ以外では必須です。特定の DBMS の一時テーブルのサポートについての詳細は、データベース一時テーブルの HOLD ファイルとしての使用および DBMS のマニュアルを参照してください。

persistValue

次のいずれかです。

VOLATILE

テーブルが DBMS セッションにローカルであることを指定します。一時的なシノニム (マスターファイルとアクセスファイル) が自動的に作成されます。このシノニムは、サーバセッションの終了とともに無効になります。

これは、一時テーブルをサポートする DBMS すべてに適用されるデフォルトの持続設定です。

一時設定のサポート、持続性と他のテーブルプロパティ、および特定の DBMS についての詳細は、データベース一時テーブルの HOLD ファイルとしての使用および DBMS ベンダーのマニュアルを参照してください。

GLOBAL_TEMPORARY

テーブルが存在する間、その定義が他のデータベースセッションやユーザに表示されます。テーブル内のデータは表示されません。永続的なシノニム (マスターファイルとアクセスファイル) が自動的に作成されます。

グローバル一時設定のサポート、持続性と他のテーブルプロパティ、および特定の DBMS についての詳細は、データベース一時テーブルの HOLD ファイルとしての使用および DBMS のマニュアルを参照してください。

PERMANENT

標準の永続テーブルの作成を指定します。永続的なシノニム (マスターファイルとアクセスファイル) が自動的に作成されます。

一時テーブル の SAME_DB PERSISTENCE プロパティ

下表は、HOLD フォーマット SAME_DB とともに使用可能なデータソースタイプ別に、一時テーブルの持続性とその他のプロパティの詳細を示しています。

DBMS	VOLATILE	GLOBAL_TEMPORARY
Db2	Linux、UNIX および Windows 上の Db2 です。一時テーブルは、DECLARE GLOBAL TEMPORARY TABLE コマンドを使用し、ON COMMIT PRESERVE ROWS オプションで作成されます。宣言されたグローバル一時テーブルは持続し、現在のセッション (接続) でのみ表示可能です。宣言されたすべてのグローバル一時テーブルに対して、「SESSION」というスキーマ名が適用されます。	このタイプのテーブルは Db2 ではサポートされません。
Informix	一時テーブルは、CREATE TEMP TABLE コマンドを使用し、WITH NO LOG オプションで作成されます。定義とデータは持続し、表示可能ですが、これはカレントセッションでのみ表示可能です。	このタイプのテーブルは Informix ではサポートされません。

DBMS	VOLATILE	GLOBAL_TEMPORARY
Microsoft SQL Server	一時テーブルはローカル一時テーブルとして作成されます。このテーブル名には、シャープ記号が 1 つ (#) 接頭語として追加されます。このため、HOLD ファイルとして使用される一時テーブル名には、HOLD 句により指定されたテーブル名に、接頭語としてシャープ記号が 1 つ (#) 追加されます。テーブルの定義とデータは、現在のセッションが継続する間のみ持続して表示可能になります。	グローバル一時テーブル名には、接頭語としてシャープ記号が 2 つ (##) 追加されます。このため、HOLD ファイルとして使用されるグローバルテーブル名には、HOLD 句により指定されるテーブル名に、接頭語としてシャープ記号が 2 つ (##) 追加されます。テーブルは、そのテーブルを作成したセッションが終了し、他のタスクがテーブルへの参照を停止すると、自動的に無効になります。テーブルの定義とデータは、他のセッションでも表示可能です。
MySQL	一時テーブルは、CREATE TEMPORARY TABLE コマンドにより作成されます。一時テーブルは持続し、現在のセッション (接続) でのみ表示可能です。一時テーブル名が永続テーブル名と同一の場合、永続テーブルは非表示になります。	このタイプのテーブルは MySQL ではサポートされません。
Oracle	このタイプのテーブルは Oracle ではサポートされません。	テーブルの定義はすべてのセッションで表示可能ですが、テーブル内のデータは、テーブルにデータを挿入したセッションでのみ表示可能です。テーブルの定義は、標準のテーブルの定義と同じ期間持続します。
Teradata	一時テーブルの定義とデータは、テーブルを作成し、データを挿入したセッション中にのみ表示可能です。一時テーブルは、ON COMMIT	グローバル一時テーブルは、永続テーブルと同じ期間持続します。定義はすべてのセッションで表示可能ですが、データは、そのデータを挿入したセッションにのみ表示可能です。グローバ

DBMS	VOLATILE	GLOBAL_TEMPORARY
	PRESERVE ROWS オプションで作成されます。	ルー時テーブルは、ON COMMIT PRESERVE ROWS オプションで作成されます。

HOLD ファイル内のフィールド名

HOLD ファイルの各フィールドには、次のように名前が割り当てられます。

1. リクエスト内のフィールドに指定された AS 名から。
2. AS 名が指定されていない場合は、シノニムで指定されたエイリアスから (エイリアスは元のリレーショナルテーブルのフィールド名と同一です)。
3. それ以外の場合は、シノニムで指定されたとおりのフィールド名から。

HOLD ファイル内の主キーとインデックス

HOLD テーブルには、主キーまたはインデックスが作成されます。キーまたはインデックスの定義は、TABLE コマンドの非表示のソートキー以外の BY ソートキーから生成されます (非表示のソートキーには、NOPRINT が指定されています)。次の基準により、主キーとインデックスのいずれかが作成されます。

1. ソートキーが一意性を持ち、ミッシングを許可しない場合 (つまり、シノニムのフィールドの MISSING 属性は選択されないか、オフ)、および作成されるテーブルタイプ上の主キーを DBMS がサポートする場合は、主キーが作成されます。
2. これらのソートキーは一意性を保証しますが、以下が該当する場合があります。
    a. フィールドのミッシングを許可する。
    b. 作成されるテーブルのタイプでは DBMS が主キーをサポートせず、ユニークインデックスが作成される。
3. これらのソートキーが一意性を保証しない場合は、一意ではないインデックスが作成されます。
4. ソートキーが表示されていない場合 (つまり、NOPRINT が指定されていないソート

キーがない場合)、主キーとインデックスのいずれも作成されません。

SAVE および SAVB ファイルの作成

デフォルト設定では、SAVE コマンドは、レポート出力を見出しや脚注のない単純なシーケンシャルデータソースとして、ALPHA フォーマットで取得します。ただし、これ以外にも、SAVE ファイルには、多くのソフトウェア製品と互換性のある、さまざまなフォーマットを指定することができます。たとえば、SAVE フォーマットを指定して、レポート出力を Web ページ、テキストドキュメント、スプレッドシート、ワードプロセッサアプリケーションに表示する、または他のプログラム言語への入力として使用することができます。サポートされるフォーマットのリストは、出力ファイルフォーマットの選択を参照してください。

フォーマットに関わらず、SAVE コマンドはマスターファイルを作成しません。

SAVB コマンドは SAVE コマンドのバリエーションです。SAVB は、マスターファイルを持たないデータソースを作成しますが、数値フィールドは BINARY フォーマットとして格納されます。SAVB ファイルは、さまざまなアプリケーションの入力として使用することができます。SAVB 出力は、HOLD コマンドにより生成される、デフォルト出力と同一です。

SAVE ファイルの作成

リクエストでは、次の構文を使用します。

```
ON TABLE SAVE [AS filename] [FORMAT fmt] [MISSING {ON|OFF}]
```

または

```
save_field SAVE [AS filename] [FORMAT fmt] [MISSING {ON|OFF}]
```

説明

save_field

BY フィールドおよび ACROSS フィールドを除く、リクエストの最終フィールド名です。

AS filename

SAVE ファイルの名前を指定します。ファイル名を指定しない場合、デフォルト名の「SAVE」が使用されます。後に続く SAVE コマンドにより前の SAVE ファイルが上書きさ

れるため、各リクエストで固有のファイル名を記述して抽出データを別のファイルに保存し、次の SAVE コマンドによる上書きを防止します。

パスを一重引用符 (') で囲むことで、SAVE ファイルの保存場所を示すこともできます。以下はその例です。

```
ON TABLE SAVE FILENAME 'install_dir:¥dir¥filename.ext' FORMAT fmt
```

FORMAT fmt

SAVE ファイルのフォーマットを指定します。ALPHA がデフォルト値です。

- Web ページ内、または Web ページとしての表示フォーマットは次のとおりです。

  ```
  HTML、HTMTABLE、DHTML
  ```

- テキストドキュメントのフォーマットは次のとおりです。

  ```
  ALPHA, DOC, PDF, WP, Text
  ```

- スプレッドシートアプリケーションで使用するフォーマットは次のとおりです。

  ```
  DIF, EXCEL, EXL2K, LOTUS, (WK1), SYLK
  ```

- データベースアプリケーションとして使用するのは次のとおりです。

  ```
  COMMA、COM、COMT
  ```

使用可能なフォーマットについての詳細は、出力ファイルフォーマットの選択を参照してください。

MISSING

MISSING 属性が ON に設定されているフィールドを SAVE ファイルに継承させます。MISSING OFF がデフォルト属性です。詳細は、ミッシングフィールド値を含むレコード処理を参照してください。

SAVE ファイルの作成

次のリクエストは、EMPLOYEE データソースからデータを抽出し、SAVE ファイルを作成します。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME AND FIRST_NAME
BY DEPARTMENT
ON TABLE SAVE
END
```

ALPHA (デフォルトの SAVE フォーマット) ファイルレイアウトの記述はレコードの検索後に表示されます。

出力結果は次のとおりです。

```
 NUMBER OF RECORDS IN TABLE=       12  LINES=     12


ALPHANUMERIC RECORD NAMED  SAVE
FIELDNAME                       ALIAS          FORMAT         LENGTH

DEPARTMENT                      DPT            A10             10
LAST_NAME                       LN             A15             15
FIRST_NAME                      FN             A10             10

TOTAL                                                          35
```

SAVB ファイルの作成

リクエストでは、次の構文を使用します。

```
ON TABLE SAVB [AS filename] [MISSING {ON|OFF}]
```

または

```
save_field SAVB [AS filename] [MISSING {ON|OFF}]
```

説明

save_field

BY フィールドおよび ACROSS フィールドを除く、リクエストの最終フィールド名です。

AS filename

SAVB ファイルの名前を指定します。ファイル名を指定しない場合、デフォルト名の「SAVB」が使用されます。後に続く SAVB コマンドにより前の SAVB ファイルがすべて

上書きされるため、各リクエストで固有のファイル名を記述して抽出データを別のファイルに保存し、次の SAVB コマンドによる上書きを防止します。

パスを一重引用符 (') で囲むことで、SAVB ファイルの保存場所を示すこともできます。以下はその例です。

```
ON TABLE SAVB FILENAME 'c:¥dir¥filename.ext '
```

MISSING

MISSING 属性が ON に設定されているフィールドを SAVE ファイルに継承させます。デフォルト値は MISSING OFF です。詳細は、ミッシングフィールド値を含むレコード処理を参照してください。

SAVB ファイルの作成

次のリクエストは、データを SALES データソースから抽出して SAVB ファイルを作成します。

```
TABLE FILE SALES
PRINT PROD_CODE AND AREA
BY DATE
WHERE CITY IS 'STAMFORD' OR 'UNIONDALE'
ON TABLE SAVB
END
```

BINARY ファイルの記述はレコードの検索後に表示されます。

出力結果は次のとおりです。

```
 NUMBER OF RECORDS IN TABLE=       10  LINES=      10


INTERNAL RECORD NAMED  SAVB
FIELDNAME                       ALIAS          FORMAT        LENGTH

DATE                            DTE            A4MD           4
PROD_CODE                       PCODE          A3             4
AREA                            LOC            A1             4

TOTAL                                                         12
```

PCHOLD ファイルの作成

PCHOLD コマンドを使用して、データを抽出することができます。データは WebFOCUS Client 経由で WebFOCUS Reporting Server から抽出され、ブラウザに HTML フォーマットで自動的に表示されます。

さらに、ヘルパーアプリケーションを設定した場合、ON TABLE PCHOLD コマンドを使用してヘルパーアプリケーション内のビューアにデータを表示することができます。たとえば、プロシジャに ON TABLE PCHOLD FORMAT EXCEL コマンドが含まれている場合は、データが HTML フォーマットでブラウザに返されることはありません。その代わりに、データは Excel ワークシートまたはブラウザで指定した他のスプレッドシートプログラムに返され、インポートされます。

対照的に、データアクセスが (WebFOCUS Client を経由せずに) Reporting Server により直接処理される場合は、データは PCHOLD ファイルに抽出され、ローカルでのレポート実行のために、自動的に PC に送信されます。

注意：使用する環境が SET パラメータの SAVEMATRIX をサポートする場合、リクエストの後にダイアログマネージャコマンドを発行して、最終レポートの内部マトリックスを保持することで、それに続く HOLD、SAVE、SAVEB コマンドで使用可能にすることができます。SAVEMATRIX についての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

PCHOLD ファイルの作成

リクエストでは、次の PCHOLD 構文を使用します。

```
ON TABLE PCHOLD [AS filename] [FORMAT fmt] [DATASET name.ext]
```

説明

PCHOLD

データの抽出、およびブラウザでの自動的な HTML フォーマットでの表示を可能にします。HOLD AT CLIENT は PCHOLD と同義です。PCHOLD コマンドにはデフォルトフォーマットが設定されてないため、PCHOLD の使用時にフォーマットを指定する必要があります。出力結果はマスターファイルとともに保存されます。PCHOLD の動作についての詳細は、HOLD ファイルの作成を参照してください。

FORMAT を使用せずに ON TABLE PCHOLD コマンドを指定すると、XML/HTML コードがブラウザに返されます。

AS filename

PCHOLD ファイルの名前を指定します。ファイル名を指定しない場合、デフォルト名の「PCHOLD」になります。後続の PCHOLD コマンドにより前の PCHOLD ファイルが上書きされるため、各リクエストで固有のファイル名を記述して抽出データを別のファイルに保存し、次の PCHOLD コマンドによる上書きを防止します。

FORMAT fmt

PCHOLD ファイルのフォーマットを指定します。

- Web ページとして表示するか、Web ページ内に表示するには、以下を選択します。

  ```
  HTML、HTMTABLE、DHTML、VISDIS
  ```

- 印刷ドキュメントとして表示するには、以下を選択します。

  ```
  PDF
  ```

- テキストドキュメントとして使用するには、以下を選択します。

  ```
  ALPHA、DOC、WP
  ```

- スプレッドシートアプリケーションで使用するには、以下を選択します。

  ```
  DIF、XLSX、EXL2K [PIVOT]、LOTUS
  ```

- 別のレポートで使用するには、以下を選択します。

  ```
  ALPHA、DFIX、COM、COMT、TAB、TABT
  ```

- 別のアプリケーションで使用するには、以下を選択します。

  ```
  XML
  ```

- Active 出力を生成するには、以下を選択します。

```
AHTML、APDF、AFLEX
```

使用可能なフォーマットについての詳細は、出力ファイルフォーマットの選択を参照してください。

DATASET name.ext

レポート出力のファイル名および拡張子を指定するために FORMAT ALPHA とともに使用します。結果は、 WebFOCUS 管理コンソールの [出力先変更設定] で指定した拡張子の設定によって異なります。拡張子が WebFOCUS 管理コンソールで定義され、[保存レポート] の値が [はい] に設定されている場合、レポート出力は、指定したファイル名と拡張子で [ダウンロード] ディレクトリに保存されます。

出力ファイルフォーマットの選択

レポートの出力を保存する際にさまざまな出力フォーマットを選択して、次のいずれかの用途に使用することができます。

- Web ページとして (Web ページ内で) 表示、または印刷ドキュメントやテキストドキュメント内で使用する。
- スプレッドシート、データベース、ワードプロセッサ、3GL プログラムなどの他のアプリケーションで処理する。
- ブラウザや PC などの他の場所に送信する。
- マルチステップレポートを作成するために元のデータソースの一部を抽出する。

使用可能なコマンド (HOLD、PCHOLD、SAVE) および動作環境を含む、サポートされる各フォーマットについての詳細は、次のフォーマットの説明を参照してください。

AHTML	DFIX	GIF	PPT	SQLORA
AHTMLTAB	DHTML	HTML	PPTX	SQLPSTGR
ALPHA	DIF	HTMTABLE	REDBRICK	SQLSYB
APDF	DOC	INGRES	SQL	SYLK
BINARY	EXL2K	INTERNAL	SQL_SCRIPT	TAB
COMMA	EXL2K	JPEG	SQLDBC	TABT

COM COMT DATREC DB2 DBASE	FORMULA EXL2K PIVOT EXL97 FLEX FOCUS	JSCHART JSON LOTUS PDF	SQLINF SQLMAC SQLMSS SQLODBC	WK1 WP XFOCUS XLSX XML

FORMAT AHTML

説明：レポート出力を Analytic Document として保存します (サーバに接続することなく、オフライン分析およびインタラクティブ機能の使用が可能な HTML ファイル)。すべてのデータおよび JavaScript コードは、HTML ファイルに格納されます。これにより、Email 配信用に出力の高圧縮化が可能になるほか、セキュリティシステムに対しても透過的になります。

用途：データのオフライン分析に使用します。

コマンドサポート：HOLD、PCHOLD、SAVE

使用可能な製品：WebFOCUS、FOCUS

FORMAT AHTMLTAB

説明：Analytic Document で使用するデータとパラメータのみが格納された出力ファイルを作成します。生成される出力結果は、完全な Analytic Document ではありません。ただし、ダイアログマネージャコマンドの -HTMLFORM を使用して、このファイルを別の HTML ドキュメントに含めることができます。詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』の「ダイアログマネージャ」の項目を参照してください。

注意：HOLD AS *name* FORMAT AHTMLTAB を発行して Analytic Document を別の HTML ドキュメントに埋め込む場合、次のように Active テクノロジ JavaScript コードを HTML BODY に追加する必要があります。

```
<BODY>
!IBI.OBJ.ACTIVEREPORTJS;
```

用途：既存の HTML ドキュメントに Analytic Document を埋め込むために使用します。

コマンドサポート：HOLD、SAVE

使用可能な製品：WebFOCUS

FORMAT ALPHA

説明：レポート出力を固定フォーマット文字データとして保存し、HOLD ファイルとして作成することができます。

ALPHA がデフォルトの SAVE フォーマットです。出力ファイルには、データのみが保存されます。

テキストフィールドは ALPHA フォーマットが設定されたファイルでサポートされます。

MISSING=ON 属性のフィールドに継承されるミッシングデータ文字を制御するには、SET HNODATA コマンドを使用します。詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

用途：テキストドキュメントで表示します。FOCUS、WebFOCUS、または App Studio でのレポート作成に使用します。データソースを変更するためのトランザクションファイルとして使用します。

コマンドサポート：HOLD、PCHOLD、SAVE

使用可能な製品：WebFOCUS、FOCUS

FORMAT APDF

説明：レポート出力を、PDF ファイルに埋め込まれた Adobe Flex レポート (Adobe Flash ファイル) として保存します。出力結果の Analytic PDF は、サーバに接続することなく、オフライン分析およびインタラクティブ機能に使用することができます。

APDF フォーマットで作成されたレポートを実行するには、Adobe Reader 9 以降が必要です。これにより、Acrobat クライアントに含まれている Flash ランタイムコードで Flash コンテンツの表示が可能になります。ただし Adobe Reader 10.1.5 以降、Flash ランタイムとして、Flash Player NPAPI が必要です。

用途：データのオフライン分析に使用します。

コマンドサポート：HOLD、PCHOLD、SAVE

使用可能な製品：FOCUS、WebFOCUS

FORMAT BINARY

説明：レポートデータの数値フィールドを 2 進数で保存します。HOLD ファイルとして作成したときはマスターファイルも作成されます。

BINARY は HOLD ファイルのデフォルトフォーマットです。BINARY フォーマットで作成すると次のとおりになります。

- HOLD ファイルは単一セグメントのシーケンシャルデータソースです。HOLD マスターファイルは元のマスターファイルの一部で、COMPUTE または DEFINE コマンド、あるいは ACROSS 句により生成されたフィールドが含まれることもあります。
- デフォルト設定では、I フォーマットのフィールドは 4 バイトのバイナリ整数のまま、F フォーマットのフィールドは 4 バイトの浮動小数点数のまま、D フォーマットのフィールドは 8 バイトの倍精度浮動小数点数のまま、P フォーマットのフィールドはパック 10 進表記のままで 8 バイト (8 バイト以下の長さのフィールドの場合)、または 16 バイトを占めます (9 バイト以上のパック 10 進数フィールドの場合)。文字フィールド (A フォーマット) は文字フォーマットで保存されます。

  連続抽出レコードの各データフィールドは 4 バイトの先頭に配列されます。このため、フォーマットが A1 の場合、フィールドの右側にブランクの 3 バイトが追加されます。この配列により、ユーザ定義のサブルーチンによるこれらのデータフィールドの処理がしやすくなります。整数およびパック 10 進数のフィールドでブランクの追加を防止する場合は、HOLD FORMAT INTERNAL を使用します。詳細は、INTERNAL フォーマットでのレポート出力の保存を参照してください。

出力ファイルには、データのみが保存されます。

用途：FOCUS、WebFOCUS、または App Studio でのレポート作成に使用します。データソースを変更するためのトランザクションファイルとして使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT COMMA

説明：カンマ区切りのフィールドおよび二重引用符 (") で囲んだ文字値を含む可変長テキストファイルとしてデータ値を保存します。フィールドのブランクはすべて保持されます。このフォーマットが業界標準のカンマ区切りフォーマットです。

このフォーマットには、二重引用符 (") を含むテキストフィールド内に二重引用符 (") を追加する安全機能はありません。

このフォーマットの拡張子は PRN です。このフォーマットタイプは、マスターファイルを作成しません。

注意

- SMARTDATE フィールドおよび日付フォーマットオプション付きの I または P フォーマットの日付フィールドは数値として扱われ、出力ファイル内では二重引用符 (") で囲まれません。日付は、文字フィールドフォーマットが設定され、日付フォーマットオプションは文字として扱われるため、二重引用符 (") で囲まれます。
- コンチネンタル式 10 進表記 (CDN=ON|SPACE|QUOTEP) はサポートされません。間にカンマ (,) のある数値は、Microsoft Access などのアプリケーションでは、2 つのフィールドとして認識されます。

出力ファイルには、データのみが保存されます。

用途：データベースアプリケーションでの処理に使用します。このフォーマットタイプは、Excel や Lotus などのアプリケーションにインポートすることができます。

コマンドサポート：HOLD、SAVE

使用可能な製品：WebFOCUS、FOCUS

FORMAT COM

説明：カンマ区切りのフィールドおよび二重引用符 (") で囲んだ文字値を含む可変長テキストファイルとしてデータ値を保存します。先頭のブランクは数値フィールドから削除され、末尾のブランクは文字フィールドから削除されます。このデータソースに対してリクエストを発行するには、PCOMMA=ON 設定が必要です。

このフォーマットには文字フィールド内での引用符の埋め込みを可能にする機能も含まれます。2 つ目の二重引用符 (") は 1 つ目に隣接して挿入されます。たとえば、「Joe "Smitty" Smith」と入力すると、出力結果は「Joe ""Smitty"" Smith」となります。

このフォーマットの拡張子は CSV です。出力ファイルの作成にHOLD コマンドを使用するときは、このフォーマットタイプに対するマスターファイルが作成されます。作成されるマスターファイルの SUFFIX は COM です。

注意

- SMARTDATE フィールドおよび日付フォーマットオプション付きの I または P フォーマットの日付フィールドは数値として扱われ、出力ファイル内では二重引用符 (") で囲まれません。日付は、文字フィールドフォーマットが設定され、日付フォーマットオプションは文字として扱われるため、二重引用符 (") で囲まれます。

- コンチネンタル式 10 進表記 (CDN=ON|SPACE|QUOTEP) はサポートされません。間にカンマ (,) のある数値は、Microsoft Access などのアプリケーションでは、2 つのフィールドとして認識されます。
- ミッシング値とブランク列または 0 (ゼロ) 値を区別する、可変長カンマ (タブ) 区切りの HOLD ファイルを作成するには、SET NULL=ON コマンドを使用します。詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

用途：データベースアプリケーションでの処理に使用します。このフォーマットタイプは、Excel や Lotus などのアプリケーションにインポートすることができます。

コマンドサポート：HOLD、SAVE、PCHOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT COMT

説明：出力ファイルの 1 行目にフィールド見出しを保存します。カンマ区切りのフィールドおよび二重引用符 (") で囲んだ文字値を含む可変長テキストファイルを作成します。先頭のブランクは数値フィールドから削除され、末尾のブランクは文字フィールドから削除されます。このフォーマットは、Microsoft Access などの特定のソフトウェアパッケージを使用する際に必要です。

このフォーマットには文字フィールド内での引用符の埋め込みを可能にする機能も含まれます。2 つ目の二重引用符 (") は 1 つ目に隣接して挿入されます。たとえば、「Joe "Smitty" Smith」と入力すると、出力結果は「Joe ""Smitty"" Smith」となります。

このフォーマットの拡張子は CSV です。出力ファイルの作成にHOLD コマンドを使用するときは、このフォーマットタイプに対するマスターファイルが作成されます。作成されるマスターファイルの SUFFIX は COMT です。

注意

- SMARTDATE フィールドおよび日付フォーマットオプション付きの I または P フォーマットの日付フィールドは数値として扱われ、出力ファイル内では二重引用符 (") で囲まれません。日付は、文字フィールドフォーマットが設定され、日付フォーマットオプションは文字として扱われるため、二重引用符 (") で囲まれます。
- コンチネンタル式 10 進表記 (CDN=ON|SPACE|QUOTEP) はサポートされません。間にカンマ (,) のある数値は、Microsoft Access などのアプリケーションでは、2 つのフィールドとして認識されます。
- ミッシング値とブランク列または 0 (ゼロ) 値を区別する、可変長カンマ (タブ) 区切りの HOLD ファイルを作成するには、SET NULL=ON コマンドを使用します。詳細は、

『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

用途：データベースアプリケーションでの処理に使用します。このフォーマットタイプは、Excel や Lotus などのアプリケーションにインポートすることができます。

コマンドサポート：HOLD、SAVE、PCHOLD

使用可能な製品：FOCUS、WebFOCUS

FORMAT DATREC

説明：レポート出力をマスターファイル付きシーケンシャルファイルとして保存し、数値フィールドを 2 進数で保存します。全語境界での位置合わせは行われません。最終フィールドは、マスターファイルの他のフィールドに対しそれぞれ 1 バイトで構成され、対応するフィールドがミッシングフィールドであるかどうかを示します。

用途：FOCUS、WebFOCUS、または App Studio でのレポート作成に使用します。データソースを変更するためのトランザクションファイルとして使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT DB2

説明：Db2 対応アダプタおよびテーブルの作成権限がある場合、Db2 テーブルを作成します。

用途：データベースアプリケーションでの処理に使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT DBASE

説明：1 行目にフィールド見出しを含む dBase フォーマットの抽出ファイルを作成します。

注意：ブランクのフィールド名はブランクのフィールドタイトルとして表示されます。このファイルをアプリケーションへの入力ファイルとして使用しようとすると、さまざまなアプリケーションで、エラーが発生する場合があります。

用途：Access や Excel などの Windows ベースのアプリケーションへのデータのインポートに使用します。

コマンドサポート：HOLD

使用可能な製品：App Studio

FORMAT DFIX

説明：区切り文字付き出力ファイルを作成します。区切り文字を指定できるほか、文字フィールドを二重引用符 (") などの特殊文字で囲むかどうか、フィールド名を含む見出しレコードを追加してファイルを生成するかどうかを指定することができます。

用途：Access や Excel などの Windows ベースのアプリケーションへのデータのインポートに使用します。

コマンドサポート：HOLD、PCHOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT DHTML

説明：PDF などの印刷用の出力フォーマットに関連したほとんどの機能を HTML 出力に追加します。HTML ファイル (.htm) または Web アーカイブファイル (.mht) を作成することができます。作成される出力ファイルのタイプは、HTMLARCHIVE パラメータにより異なります。HTMLARCHIVE=ON により、Web アーカイブファイルが作成されます。

DHTML フォーマット でサポートされる機能には、次のものがあります。

- **絶対配置**　テキストおよびイメージを HTML レポート内に正確に配置します。HTML のレイアウトには、PDF 出力と同一のスタイルシート構文を使用することができます。
- **Web ビューア**　Web ビューアは SET HTMLARCHIVE=OFF で利用することができます。
- **PDF スタイルシート機能**　たとえば、罫線、背景色、OVER、バースト、複合レポート

機能がサポートされます。

注意

- DHTML レポートのフォントマッピングファイルは dhtml.fmp です。
- レガシー複合レポートはサポートされません。
- ドリルスルーレポートはサポートされません。

用途：Web ページとして表示します。

コマンドサポート：HOLD、PCHOLD、SAVE

使用可能な製品：WebFOCUS、FOCUS

FORMAT DIF

説明：見出し、脚注、中間見出しおよび中間脚注を除くレポート出力全体を取得し、文字ファイルを作成します。このファイルは、ほとんどのスプレッドシート製品と簡単に統合することができます。

たとえば、HEADING/FOOTING および ON TABLE PCHOLD を使用して TABLE リクエストを実行すると、 FORMAT DIF は見出しおよび脚注を含むレポート出力を表示しません。これを回避するには、別のフォーマット (HTML、PDF、EXL2K など) を使用します。

注意：HOLD FORMAT DIF では、Microsoft Excel SR-1 はサポート対象外です。これらのレポートを開くには、Microsoft Excel SR-2 と Microsoft Excel 2000 のいずれかを使用します。

用途：スプレッドシートアプリケーションでの表示および処理に使用します。

コマンドサポート：HOLD、PCHOLD、SAVE

PCHOLD のバリエーションによりデータを Web サーバからブラウザへ転送します。

使用可能な製品：WebFOCUS、FOCUS

FORMAT DOC

説明：見出し、脚注、中間合計を含むレポート出力全体を取得し、レイアウトと改行付きのテキストファイルを作成します。このファイルは、ほとんどのワードプロセッサ製品と簡単に統合することができます。DOC フォーマットではページ制御情報を示すフォームフィード

文字を使用します。

注意：ON TABLE PCHOLD FORMAT DOC を含むリクエストを Microsoft Office XP で表示すると、レポートの 1 ページ目がブランクになります。これを回避するには、リクエストに SET PAGE=NOPAGE を含めます。

用途：テキストドキュメントで表示します。

コマンドサポート：HOLD、PCHOLD、SAVE

PCHOLD のバリエーションによりデータを Web サーバからブラウザへ転送します。

使用可能な製品：WebFOCUS、FOCUS

FORMAT XLSX

説明：完全にスタイルが設定された Excel 2007 XML フォーマットレポートが作成されます。この出力フォーマットを使用するには、Excel 2007 以降をインストールしておく必要があります。

Excel 2007 ワークシートには、最大で 1,048,576 行、16,384 列を含めることができます。WebFOCUS は、これらの定義済み制限を超えるワークシートを生成しますが、生成されたブックを Excel 2007 で開くと、制限を超えるデータが切り捨てられます。

行数が多すぎるために単一ワークシートに収まらない場合は、超過した行を表示するオーバフローワークシートを作成することができます。

用途：スプレッドシートアプリケーションでの表示および処理に使用します。

コマンドサポート：HOLD、PCHOLD、SAVE

Internet Explorer では、PCHOLD を実行すると、ブラウザ上で Excel 2007 を開始します。

使用可能な製品：WebFOCUS

FORMAT EXL2K

説明：完全スタイル設定された Excel 2000 HTML フォーマットレポートが作成されます。この出力フォーマットを使用するには、Excel 2000 をインストールしておく必要があります。

EXL2K 出力フォーマットでは、ACROSS フィールドタイトルがサポートされます。

EXL2K 出力フォーマットでは、レポートに 65,536 行、256 列を含めることができます。これらの上限を超える行および列は、Excel で開いたときにワークシートから除外されます。

行数が多すぎるために単一ワークシートに収まらない場合は、超過した行を表示するオーバフローワークシートを作成することができます。

用途：スプレッドシートアプリケーションでの表示および処理に使用します。

コマンドサポート：HOLD、PCHOLD、SAVE

Internet Explorer では、PCHOLD を実行すると、ブラウザ上で Excel 2000 を開始します。

使用可能な製品：WebFOCUS

FORMAT EXL2K FORMULA

説明：レポートを Excel 2000 ワークシートとして表示します。WebFOCUS の合計をはじめとする一時項目 (COMPUTE) は、Excel の数式に変換されます。

用途：スプレッドシートアプリケーションでの表示および処理に使用します。

コマンドサポート：HOLD、PCHOLD、SAVE

使用可能な製品：WebFOCUS、FOCUS

FORMAT EXL2K PIVOT

説明：Excel 2000 HTML フォーマットの完全スタイルレポートに加え、PivotTable 機能が提供されます。PC 上に Excel 2000 が必要です。

用途：スプレッドシートアプリケーションでの表示および処理に使用します。

コマンドサポート：HOLD、PCHOLD、SAVE

Internet Explorer では、PCHOLD のバリエーションによりデータが Web サーバからブラウザに転送され、Excel 2000 が起動します。

使用可能な製品：WebFOCUS、FOCUS

FORMAT EXL97

説明：完全スタイル設定を含む Excel 97 レポートとして表示、保存します。

FORMAT EXL97 では、先頭の 0 (ゼロ) は表示されません。

用途：スプレッドシートアプリケーションでの表示および処理に使用します。

コマンドサポート：HOLD、PCHOLD、SAVE

使用可能な製品：FOCUS、WebFOCUS

FORMAT FLEX

説明：レポート出力を Analytic Document として、Adobe Flash Player と互換性のある .SWF ファイルに保存します。このファイルは、サーバに接続することなく、オフライン分析やインタラクティブ機能に使用することができます。Internet Explorer、Mozilla Firefox、Opera インターネットブラウザでは、Adobe Flex フォーマットは Shockwave Flash Object として認識されます。

Active テクノロジフォーマット (FLEX) で作成したレポートを実行するには、Adobe Flash Player 9.0.28 以降が必要です。

用途：データのオフライン分析に使用します。

コマンドサポート：HOLD、SAVE

使用可能な製品：FOCUS、WebFOCUS

FORMAT FOCUS

説明：FOCUS データソースを作成します。2 つのファイル (HOLD データファイル、HOLD マスターファイル) が作成されます。詳細は、ibi FOCUS フォーマットによるレポート出力の保存を参照してください。

テキストフィールドは FOCUS 出力ファイルでサポートされます。

用途：データベースアプリケーションでの処理に使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT GIF

説明：GRAPH リクエストの出力を GIF ファイルとして保存します。

用途：結果の GIF ファイルを PDF または HTML レポート見出し、脚注、または本文に埋め込むことができます。この方法は、TABLE リクエストおよび GRAPH リクエストからの出力などの複数の出力を含んだ単一 PDF または HTML レポートを作成するときに役立ちます。この種のレポートは、ReportCaster を使用して配信することができます。

UNIX プラットフォーム上で HEADLESS 構成を使用して GIF フォーマットのグラフを実行すると、グラフが正しく表示されない場合があります。イメージの代わりに赤い X が表示される場合があります。これはヘッダなしでイメージを作成する SUN の仕様によるもので、現在、GIF イメージはサポートされていません。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS

詳細は、グラフの作成を参照してください。

FORMAT HTML

説明：Web ブラウザで表示可能な完全な HTML ドキュメントを作成します。 PCHOLD のバリエーションによりデータを Web サーバからブラウザへ転送します。

HTML でサポートされるスタイルシート機能は、JAVASCRIPT (JavaScript へのドリルダウン)、TARGET、COLSPAN、HEADALIGN、IMAGEALIGN、IMAGEBREAK、GUTTER、BACKIMAGE、BACKCOLOR、IMAGE、GRIDS です。

用途：Web ページとして表示します。

コマンドサポート：HOLD、PCHOLD、SAVE

使用可能な製品：WebFOCUS、FOCUS

FORMAT HTMTABLE

説明：HTML テーブルのみを含む出力ファイルを作成します。生成される出力結果は完全な HTML ドキュメントではありません。

ただし、ダイアログマネージャコマンドの -HTMLFORM を使用して、このファイルを別の HTML ドキュメントに含めることができます。ダイアログマネージャについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

注意：HOLD AS *name* FORMAT HTMTABLE を発行する場合、-HTMLFORM 名で使用した .htm ファイル名とは異なる名前を指定する必要があります。

HTML でサポートされるスタイルシート機能は、JAVASCRIPT (JavaScript へのドリルダウン)、TARGET、COLSPAN、HEADALIGN、IMAGEALIGN、IMAGEBREAK、BACKCOLOR、IMAGE、GRID です。

内部カスケードスタイルシート (CSS) は FORMAT HTMTABLE でサポートされます。CSS コードは TABLE コマンドの直前に配置します。

用途：既存の HTML ドキュメントにレポートやグラフを埋め込むために使用します。

コマンドサポート：HOLD、PCHOLD、SAVE

PCHOLD のバリエーションによりデータを Web サーバからブラウザへ転送することもできます。

使用可能な製品：WebFOCUS、FOCUS

FORMAT INGRES

説明：Ingres 対応アダプタおよびテーブルの作成権限がある場合、Ingres テーブルを作成します。

用途：データベースアプリケーションでの処理に使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS (WebFOCUS Reporting Server のクライアントとして使用)

FORMAT INTERNAL

説明：整数値およびパックフィールドを追加せずにレポートの出力結果を保存します。詳細は、INTERNAL フォーマットでのレポート出力の保存を参照してください。

用途：3GL プログラムによる高精度処理に使用します。

コマンドサポート：HOLD、SAVB

使用可能な製品：WebFOCUS、FOCUS

FORMAT JPEG

説明：GRAPH リクエストの出力結果を JPEG ファイルとして保存します。

用途：結果の JPEG ファイルは PDF または HTML レポート見出し、脚注、本文に埋め込むことができます。この方法は、TABLE リクエストおよび GRAPH リクエストからの出力などの複数の出力を含んだ単一 PDF または HTML レポートを作成するときに役立ちます。この種のレポートは、ReportCaster を使用して配信することができます。JPEG (.jpg) ファイルは、GIF ファイルと同様に動作します。

注意：JPEG ファイルの場合、.jpg 拡張子のみがサポートされます。.jpeg 拡張子はサポートされません。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS

詳細は、グラフィックイメージからのリンクおよび レポートへのイメージの追加を参照してください。

FORMAT JSCHART

説明：グラフリクエストの出力を HTML5 グラフとして保存します。

用途：このグラフは、ビルトイン JavaScript エンジンを使用して、高画質でインタラクティブなベクタグラフィックスとしてブラウザに表示されます。古いブラウザでは、HTML5 標準の一部の機能がサポートされません。HTML5 グラフを作成するには、次のコマンドを含める必要があります。

```
ON GRAPH PCHOLD FORMAT JSCHART
```

コマンドサポート：ON GRAPH PCHOLD、ON GRAPH HOLD

使用可能な製品：WebFOCUS

FORMAT JSON

説明：JSON (JavaScript Object Notation) は、名前と値の組み合わせで構成される JavaScript ベースのテキストフォーマットです。さまざまなタイプのソフトウェアで解析で

き、データの保存と交換によく使用されます。リクエストのフィールド名と値は、FORMAT JSON ファイルに保存されます。中間合計、見出し、脚注、ACROSS フィールドは、HOLD ファイルに継承されません。シノニムは、HOLD ファイルに対して生成されます。STRUCTURE HIERARCHY 構文を HOLD コマンドに追加すると、リクエストのソートフィールドと表示フィールドを識別する JSON 出力ファイルが生成されます。

用途：システムおよびアプリケーション間のデータ交換に使用されます。

コマンドサポート：HOLD、PCHOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT LOTUS

説明：LOTUS 1-2-3 に後でインポート可能な文字ファイルとして、レポートのすべてのフィールドを取得します。すべての文字フィールドは引用符で囲みます。フィールドはカンマによって区切られます。

用途：スプレッドシートアプリケーションでの表示および処理に使用します。

コマンドサポート：HOLD、PCHOLD、SAVE (WebFOCUS のみ)

使用可能な製品：WebFOCUS、FOCUS

FORMAT PDF

説明：レポートを PDF (Adobe Portable Document Format) として保存します。PDF フォーマットは、出力結果 (見出し、脚注、タイトルなどのすべてのフォーマットオプション) を印刷したときと同様、物理ページ上に正確に配置することができます。

WebFOCUS では、PDF にスタイルシートを使用することができます。これには、レポート、ページ、ソート項目を指定している見出しおよび脚注への、ドリルダウンパラメータ、任意の URL へのリンク、GIF および JPEG イメージの埋め込みなどがあります。

PDF ドキュメントに Java アプレットを埋め込むことはできないため、PDF フォーマットでは OLAP レポートはサポートされません。ただし、OLAP レポートから PDF フォーマットレポートへドリルダウンを設定することは可能です。PDF では、横方向ソートに対するドリルダウンはサポートされません。レポートに別のレポートへのドリルダウンを設定するときは、プラス記号 (+)、シャープ記号 (#)、アンパサンド (&)、円記号 (¥) は使用できません。

PDF オブジェクトはページ、ハイパーリンクまたはイメージです。PDF には、ドキュメントに含めることのできるオブジェクト数に制限があります。WebFOCUS による各 PDF レポートのオブジェクトの制限は次のとおりです。

オブジェクト	制限値
ページ数	10,000
イメージ	900
1 ページ内のハイパーリンク数	500
ハイパーリンクを含むページの総数	100
ハイパーリンクの総数	44,500

PDF フォーマットでは、見出し、脚注、タイトルなどのすべてのフォーマットオプションが保持されます。

サポートされるフォントは、Windows 英字フォントでは、Courier (固定幅)、Times (プロポーショナル)、Helvetica (プロポーショナル)、日本語フォントでは MS ゴシック (固定幅)、MS P ゴシック (プロポーショナル)、MS 明朝 (固定幅)、MS P 明朝 (プロポーショナル)です。使用可能なフォントスタイルは、Normal (デフォルト)、Bold、Italic、Underline、およびこれらの組み合わせです。

PDF でサポートされるスタイルシート機能は、PAGESIZE、ORIENTATION、UNITS、TOPMARGIN、BOTTOMMARGIN、LEFTMARGIN、RIGHTMARGIN、POSITION、SQUEEZE、FOCEXEC (プロシジャへのドリルダウン)、URL (URL へのドリルダウン)、HGRID、VGRID、BACKCOLOR です。PDF レポート内で BACKCOLOR を使用するときは、レポート内のデータの各セルの上部、下部、右側、および左側に余分なスペースが追加されます。これは読みやすさを向上し、データが切り捨てられることを避けるためです。

用途：印刷ドキュメントとして表示します。

コマンドサポート：HOLD、PCHOLD、SAVE (WebFOCUS のみ)

使用可能な製品：WebFOCUS、FOCUS

FORMAT PDF OPEN/CLOSE

説明：複数のレポートを単一の PDF レポートに保存します。

用途：複数のドキュメントを単一の PDF ファイルに統合するために使用します。「複合レポート」とも呼ばれます。詳細は、レポートページのレイアウトを参照してください。

コマンドサポート：PCHOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT PPT

説明：PowerPoint フォーマットの出力ファイルを作成します。レポート出力の各ページはスタイルが適用された個別のスライドになります。

用途：スライドプレゼンテーションで使用します。

コマンドサポート：HOLD、SAVE

使用可能な製品：WebFOCUS、FOCUS

FORMAT PPTX

説明：PowerPoint フォーマットの出力ファイルを作成します。レポート出力の各ページはスタイルが適用された個別のスライドになります。

用途：スライドプレゼンテーションで使用します。

コマンドサポート：HOLD、SAVE

使用可能な製品：WebFOCUS、FOCUS

FORMAT REDBRICK

説明：Redbrick 対応アダプタおよびテーブルの作成権限がある場合、Red Brick テーブルを作成します。

用途：データベースアプリケーションでの処理に使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS (WebFOCUS Reporting Server のクライアントとして使用)

FORMAT SQL_SCRIPT

説明：SQL サブクエリファイルまたはデータ値のファイルとともに、そのデータを記述したシノニムを作成します。

リレーショナルデータソースに対するリクエストで HOLD FORMAT SQL_SCRIPT コマンドを使用すると、現在のクエリの実行に必要な SQL SELECT ステートメントが生成されます。この SQL SELECT ステートメントは、.sql 拡張子のファイルとしてアプリケーションフォルダに格納されます。また、SQL アンサーセットを記述した一組のマスターファイルとアクセスファイルも同時に生成されます。

その他のタイプのデータソースに対するリクエストで HOLD FORMAT SQL_SCRIPT コマンドを使用すると、現在のクエリが実行され、取得された値が .ftm 拡張子のシーケンシャルファイルとしてアプリケーションフォルダに格納されます。また、取得されたデータを記述したマスターファイルも同時に生成されます。

用途：HOLD FORMAT SQL_SCRIPT から取得された出力は、DB_INFILE 関数のターゲットファイルとして使用することができます。DB_INFILE 関数についての詳細は、『ibi™ WebFOCUS® 関数リファレンス』を参照してください。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT SQLDBC

説明：Teradata 対応アダプタおよびテーブルの作成権限がある場合、Teradata テーブルを作成します。

用途：データベースアプリケーションでの処理に使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT SQLINF

説明：Informix 対応アダプタおよびテーブルの作成権限がある場合、Informix テーブルを作成します。

用途：データベースアプリケーションでの処理に使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS (WebFOCUS Reporting Server のクライアントとして使用)

FORMAT SQLMAC

説明：Microsoft Access 対応アダプタおよびテーブルの作成権限がある場合、Microsoft Access テーブルを作成します。

用途：データベースアプリケーションでの処理に使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS

FORMAT SQLMSS

説明：Microsoft SQL 対応アダプタおよびテーブルの作成権限がある場合、Microsoft SQL Server テーブルを作成します。

用途：データベースアプリケーションでの処理に使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS (WebFOCUS Reporting Server のクライアントとして使用)

FORMAT SQLODBC

説明：最新の ODBC 対応アダプタおよびテーブルの作成権限がある場合、SQLODBC テーブルを作成します。

用途：データベースアプリケーションでの処理に使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS (WebFOCUS Reporting Server のクライアントとして使用)

FORMAT SQLORA

説明：Oracle 対応アダプタ およびテーブルの作成権限がある場合、Oracle テーブルを作成します。

用途：データベースアプリケーションでの処理に使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT SQLPSTGR

説明：PostgreSQL 対応アダプタおよびテーブル作成の権限がある場合、PostgreSQL テーブルを作成します。

用途：データベースアプリケーションでの処理に使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS (WebFOCUS Reporting Server のクライアントとして使用)

FORMAT SQLSYB

説明：Sybase 対応アダプタおよびテーブルの作成権限がある場合、Sybase テーブルを作成します。

用途：データベースアプリケーションでの処理に使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS (WebFOCUS Reporting Server のクライアントとして使用)

FORMAT SYLK

説明：リクエストのフィールドをすべて Multiplan (Microsoft のスプレッドシートプログラム) の文字ファイルとして取得します。作成したファイルに 9,999 行以上を含めることはできません。

用途：スプレッドシートアプリケーションでの表示および処理に使用します。

コマンドサポート：HOLD、SAVE

使用可能な製品：WebFOCUS、FOCUS

FORMAT TAB

説明：タブ区切りフォーマットの出力ファイルを作成します。TAB フォーマットには、文字フィールド内での引用符の埋め込みを可能にする機能が含まれます。2 つ目の二重引用符 (") は 1 つ目に隣接して挿入されます。たとえば、「Joe "Smitty" Smith」と入力すると、出力結果は「Joe ""Smitty"" Smith」となります。TAB フォーマットには、次のような機能もあります。

- 末尾のブランクは、すべて ALPHA [An] フィールドから削除されます。
- 先頭のブランクのすべては数値 [/Dx.y、/Fx.y、/Px.y および /In] フィールドから削除されます。
- 出力ファイルには、32 キロバイトというレコード長制限があります。
- HOLD コマンドを使用して出力ファイルを作成するとは、マスターファイルが作成さ

れます。マスターファイルは二重引用符 (") を含める点を除いて FORMAT ALPHA と同様に動作します。

注意：ミッシング値とブランク列または 0 (ゼロ) 値を区別する、可変長カンマ (タブ) 区切りの HOLD ファイルを作成するには、SET NULL=ON コマンドを使用します。詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

用途：Access や Excel などの Windows ベースのアプリケーションへのデータのインポートに使用します。

コマンドサポート：HOLD、SAVE、PCHOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT TABT

説明：1 行目にフィールド見出しを含むタブ区切りフォーマットの出力ファイルを作成します。TABT フォーマットには、文字フィールド内での引用符の埋め込みを可能にする機能が含まれます。2 つ目の二重引用符 (") は 1 つ目に隣接して挿入されます。たとえば、「Joe "Smitty" Smith」と入力すると、出力結果は「Joe ""Smitty"" Smith」となります。TABT フォーマットには、次のような機能もあります。

- 1 行目にフィールド名が含まれます。
- 末尾のブランクは、すべて ALPHA [An] フィールドから削除されます。
- 先頭のブランクのすべては数値 [/Dx.y、/Fx.y、/Px.y および /In] フィールドから削除されます。
- 出力ファイルには、32 キロバイトというレコード長制限があります。
- HOLD コマンドを使用して出力ファイルを作成するとは、マスターファイルが作成されます。マスターファイルは二重引用符 (") を含める点を除いて FORMAT ALPHA と同様に動作します。

注意

- ブランクのフィールド名はブランクのフィールドタイトルとして表示されます。このファイルをアプリケーションへの入力ファイルとして使用しようとすると、さまざまなアプリケーションで、エラーが発生する場合があります。
- ミッシング値とブランク列または 0 (ゼロ) 値を区別する、可変長カンマ (タブ) 区切りの HOLD ファイルを作成するには、SET NULL=ON コマンドを使用します。詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

用途：Access や Excel などの Windows ベースのアプリケーションへのデータのインポートに使用します。

コマンドサポート：HOLD、SAVE、PCHOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT WK1

説明：LOTUS 1-2-3 Release 2 にインポート可能な文字ファイルとして、レポートのすべてのフィールドを取得します。

用途：スプレッドシートアプリケーションでの表示および処理に使用します。

コマンドサポート：HOLD、PCHOLD

使用可能な製品：App Studio

FORMAT WP

説明：見出し、脚注、中間合計を含むレポート出力全体を取得し、テキストファイルを作成します。このファイルは、ほとんどのワードプロセッサ製品に簡単に統合することができます。

テキストフィールドは WP フォーマットでサポートされます。

改行コードをレポート出力の各ページの 1 列目に含めるかどうかは、次のように制御します。

```
[ON TABLE] HOLD AS filename FORMAT WP [CC|NOCC]
```

NOCC は、改行コードを除外します。これらの文字用に確保された位置は、ファイルにブランクとして残ります。CC は、改行コードを挿入します。WP ファイルにページ制御情報を含めるには、見出しまたは SET PAGE=OFF コマンドで TABPAGENO オプションを指定することもできます。1 列目の 1 つ目の文字は、新しいページの開始を示します。

FORMAT WP 改行コード制御オプションの規則の概要は次のとおりです。

- CC オプションは、常に改行コードを挿入する。
- NOCC オプションは、常に改行コードを除外する。

- CC および NOCC オプションなしで HOLD FORMAT WP を発行する場合は、次の規則が適用されます。
    - SET PAGE-NUM=OFF および SET PAGE-NUM=TOP は、常に改行コードを挿入する。
    - SET PAGE-NUM=NOPAGE は、常に改行コードを除外する。
    - SET PAGE-NUM=ON に設定する場合、見出しに TABPAGENO を含めるときは改行コードは挿入され、含めないときは除外されます。

ヒント：HOLD FORMAT WP は、ページごとの行番号は変更しません。これを実行するには、SET PRINT=OFFLINE、SET SCREEN=PAPER、SET SCREEN=OFF コマンドをそれぞれ単独で、または組み合わせて発行します。

HOLD FORMAT WP の最長レコードは 358 バイトであり、そのうちデータを表すことができるのは 356 バイトです。それ以上の出力には、PCHOLD FORMAT WP を使用します。

出力ファイルを後から処理するために、バージョンが変更された場合でもレポート幅を固定した状態にするには、SET WPMINWIDTH コマンドを使用して幅を設定します。このパラメータでは、出力ファイルの最小幅を指定します。リクエストの出力ファイルに継承されたフィールドがこの設定幅に収まらない場合は、幅が自動的に広くなります。レコードの長さは WPMIDWIDTH の値になります。

FORMAT WP は、見出し、脚注、中間合計を保持します。

用途：テキストドキュメントで表示します。

コマンドサポート：HOLD、PCHOLD、SAVE

使用可能な製品：WebFOCUS、FOCUS

FORMAT XFOCUS

説明：XFOCUS データソースを作成します。

用途：データベースアプリケーションでの処理に使用します。

コマンドサポート：HOLD

使用可能な製品：WebFOCUS、FOCUS

FORMAT XML

説明：最終レポート出力に表示される行が記述された内部 DTD に基づいて、XML 出力ファイルを作成します。このフォーマットでは、HOLDLIST 設定は適用されません。このフォーマットでは、ASNAMES 設定は適用されません。HOLDATTRS ON パラメータが自動的に呼び出されます。XML ファイルは、レポートリクエストの構造およびレイアウトに基づいて生成されます。DTD に記述されるメタデータは、各フィールドの FIELDNAME、ALIAS、データタイプ、長さ、FOCUS フォーマット、DESCRIPTION、ACCEPTS、HELP_MESSAGE、TITLE、WITHIN、PROPERTY、REFERENCE、VALIGN、フィールドタイトルです。FORMAT XML では、マスターファイルは作成されません。

注意：HTML オートプロンプトページ ([新規ウィンドウで実行] を選択)、BI Portal、HTML キャンバスで作成した HTML ページを使用する場合、iframe で出力を表示する標準モードで実行するブラウザでは、XML は表示されません。PCHOLD FORMAT XML が指定された WebFOCUS リクエストを実行すると、出力結果が新しいブラウザウィンドウに表示されます。

用途：後続の処理に使用します。

コマンドサポート：HOLD、PCHOLD

使用可能な製品：WebFOCUS、FOCUS

ON TABLE MERGE による既存データソースへのデータの結合

TABLE コマンドおよび TABLEF コマンドで ON TABLE MERGE 句を使用すると、既存の単一セグメントデータソースへのレコードの挿入、更新、削除が可能になります。

データソースが個別レコードまたはバルクロードによる挿入、更新、削除操作をサポートする場合、ON TABLE MERGE のすべてのオプションを利用することができます。リレーショナルデータソースでは、これらすべてのオプションがサポートされます。区切り付きフォーマット、固定フォーマットのシーケンシャルデータソースなど、その他のデータソースでは、既存データの末尾へのレコードの追加のみが行えます。

ON TABLE MERGE 処理

ON TABLE MERGE リクエストは、2 つのセクションで構成されます。

- ターゲットファイルに結合するソースファイルからデータを取得する TABLE リクエストまたは TABLEF リクエスト

  任意のタイプのデータソースと対応するシノニムを使用する、すべての TABLE 構文がサポートされます。リクエストには、JOIN、DEFINE、COMPUTE、およびサポートされるその他の構文を含めることができます。取得されたレコードは、SQLリクエストでリレーショナルデータベースに直接渡される (リクエストが最適化できる場合) か、SQL MERGE コマンドを使用してロードされ (リクエストは最適化できないが、データベースがSQL MERGE コマンドをサポートしている場合)ます。

- 結合で既存のターゲットデータソースを指定する ON TABLE MERGE 句、UPDATE、DELETE、INSERT コマンド、および更新または挿入するフィールド値の演算。

  ターゲットは単一セグメントファイルであり、対応するシノニムおよび WebFOCUS が更新可能な書き込みアダプタが必要です。

サポートされる結合句

- **MATCHING 式**　UPDATE、DELETE、または INSERT に渡すソースレコードおよびターゲットレコードを選択する式を指定します。レコードの末尾への追加のみを行う場合は、MATCHING を使用する必要はありません。
- **WHEN MATCHED [AND expression] UPDATE**　一致レコードを更新します。必要に応じて、別のターゲットレコードセットに対して異なる更新を実行するその他の式を追加します。複数の更新セットを実行するには、複数の WHEN MATCHED UPDATE 句を含めます。

  新しいターゲットフィールド値は、WHEN MATCHED UPDATE 句の後に指定された式を使用して計算されます。

- **WHEN MATCHED [AND expression] DELETE**　一致レコードを削除します。必要に応じて、複数のターゲットレコードセットに対するその他の式を追加します。複数のレコードセットを削除するには、複数の WHEN MATCHED DELETE 句を含めます。
- **WHEN NOT MATCHED INSERT**　MATCHING を満たすフィールド値がない場合に、新しいレコードを挿入するために使用します。

  新しいターゲットフィールド値は、WHEN MATCHED INSERT 句の後に指定された式を使用して計算されます。

- **INSERT**　MATCHING 句および WHEN NOT MATCHED 句を使用しない場合は、INSERT を使用して新しいレコードを末尾に追加することができます。

新しいターゲットフィールド値は、INSERT 句の後に指定された式を使用して計算されます。

生成された MERGE リクエストのプロパティ

MERGE リクエストを生成するには、いくつかの方法があります。

- **最適化済みリクエスト**　ソースとターゲットのデータベースが同一で (接頭語と接続が同一)、データベースで MERGE (INSERT の場合 INSERT from SELECT) がサポートされる場合は、SQL リクエストでリクエスト全体が DBMS に渡されます。

  最適化済みリクエストでは、MATCHING 式を任意の式にすることができ、MERGE リクエストでレコードの失敗が発生した場合、MERGE 全体が拒否されます。

- **非最適化リクエスト**　ターゲットとソースのデータベースや使用する接続が異なる場合や、TABLE リクエストに最適化できない構文が含まれている場合など、何らかの理由でリクエストを最適化できない場合は、次の方法で結合を実行することができます。

  - ターゲットデータベースが一時テーブルへのバルクロードおよび MERGE をサポートする場合は、バルクロードでターゲットデータベースを一時テーブルとして使用してデータをロードした上で、SQL MERGE コマンドで一時テーブルを入力として指定することで、ターゲットテーブルを更新します。
  - ターゲットデータベースが MERGE をサポートするがバルクロードをサポートしない場合は、レコードごとに 1 つのリクエストを使用して、データを 1 件ずつロードします。
  - MATCHING 式は、すべてのキーフィールドに一致する必要があり、処理はレコードごとに実行されるため、MERGE リクエストでレコードが失敗した場合は、リクエストは次のレコードに対して継続されます。

ON TABLE MERGE リクエストでは、ログは生成されません。したがって、実稼働環境ファイルを処理する際は、リクエストを実行する前に、ソースファイルのバックアップを作成することをお勧めします。MERGE 処理の最後に、結合が成功したかどうかと影響を受けたレコード数を示すメッセージが生成されます。下図は、最適化済みリクエストで表示される、このメッセージの例を示しています。

```
(FOC2659) FULL OPTIMIZATION OF MERGE WITH SUBSELECT HAS BEEN DONE
(FOC2661) TARGET FILE ibisamp/dmrpts
(FOC2665) MERGE PROCESS STARTED AT 10.11.22
(FOC1796) ROWS AFFECTED BY MERGE STATEMENT: 19
```

下図は、最適化されていないリクエストで表示される、このメッセージの例を示しています。

```
(FOC2644) FULL OPTIMIZATION OF MERGE WITH SUBSELECT CANNOT BE DONE
(FOC2625) SOURCE AND TARGET RESIDE ON DIFFERENT CONNECTIONS
(FOC2664) PARAMETRIZED PER RECORD STATEMENT WILL BE GENERATED
(FOC2661) TARGET FILE ibisamp/dmrpts

(FOC2665) MERGE PROCESS STARTED AT 13.22.42
(FOC1796) ROWS AFFECTED BY MERGE STATEMENT: 182
(FOC2666) MERGE PROCESS ENDED AT 13.22.42, ELAPSED TIME = 00:00:00.037
```

レコードの一部が処理されず、どのレコードが処理されなかったかを特定する必要がある場合は、トレースを有効にするか、セッションログを確認します。

ON TABLE MERGE による既存データソースへのデータの結合

次の構文では、最低でも 1 つの UPDATE、DELETE、INSERT コマンドを指定する必要があります。複数の式で個別の UPDATE または DELETE 条件およびターゲットフィールド値を指定する場合は、複数の UPDATE コマンドおよび DELETE コマンドがサポートされます。

```
{TABLEF|TABLE} FILE
...
ON TABLE MERGE INTO FILE target_mf
[MATCHING matching_expression;]

[
   WHEN MATCHED [AND matching_expression2;] UPDATE
     target_field1=expression1;
     target_field2=expression2; ...
   ...
]

[
   WHEN MATCHED [AND matching_expression3;] DELETE
   ...
]



[
 [WHEN NOT MATCHED] INSERT
  target_field3=expression3;
  target_field4=expression4; ...
]
END
```

説明

target_mf

更新するターゲットデータソースのマスターファイル名です。

MATCHING matching_expression

INSERT、UPDATE、DELETE コマンドを適用するための一致式です。複雑な式を指定することもできます。INSERT コマンドでレコードを末尾に追加する場合は、この句を省略します。ソースとターゲットでフィールド名が同一の場合は、ソースファイルのフィールド参照として SRC という接頭語を使用し、ターゲットファイルのフィールド参照として TRG という接頭語を使用します。

matching_expression2

UPDATE コマンドを適用するための追加の一致式です。ソースとターゲットでフィールド名が同一の場合は、ソースファイルのフィールド参照として SRC という接頭語を使用し、ターゲットファイルのフィールド参照として TRG という接頭語を使用します。

target_field1, target_field2 ...

ターゲットデータソースで更新する 1 つまたは複数のフィールドです。

expression1, expression2

既存のターゲットフィールド値の更新に使用するターゲットフィールド値を計算する式です。これらの式では、TABLE リクエストおよびターゲットデータソースのフィールドを使用することができます。ソースとターゲットでフィールド名が同一の場合は、ソースファイルのフィールド参照として SRC という接頭語を使用し、ターゲットファイルのフィールド参照として TRG という接頭語を使用します。

matching_expression3

DELETE コマンドを適用するための追加の一致式です。ソースとターゲットでフィールド名が同一の場合は、ソースファイルのフィールド参照として SRC という接頭語を使用し、ターゲットファイルのフィールド参照として TRG という接頭語を使用します。

target_field3, target_field4 ...

ターゲットデータソースで *matching_expression* が見つからない場合に、挿入する新しいレコードの値です。

expression3, expression4, expression5

新しいレコードを挿入する際に、ターゲットフィールド値の計算に使用する式です。これらの式では、TABLE リクエストのフィールドを使用することができます。これらのフィールドを参照するには、接頭語 SRC を使用します。

ON TABLE MERGE によるレコードの更新および挿入

この例では、[ibi™ Data Migrator - 全般] チュートリアルのソースデータおよびターゲットデータを使用します。チュートリアルにアクセスするには、[アプリケーション] ページのリボンで [新規] ボタン (+) をクリックして [チュートリアル] を選択するか、アプリケーションフォルダを右クリックして [新規]、[チュートリアル] を順に選択します。[Data Migrator - 全般] まで下方向へスクロールします。DBMS 接続、接頭語、アプリケーションの値を選択または入力し、[作成] をクリックします。

次のリクエストは、dminv テーブルのレコード (SRC.PROD_NUM) のフィールド値 PROD_NUM が dmrpts (TRG.PROD_NUM) のフィールド値 PROD_NUM に一致した場合に、dmrpts テーブルのレコードを dminv テーブルのレコードに更新します。PROD_NUM フィールドが一致しない場合は、レコードが挿入されます。

```
TABLE FILE dminv
PRINT
   PROD_NUM
 COMPUTE QUANTITY =  SUM.QTY_IN_STOCK;
 COMPUTE LINEPRICE =  SUM.PRICE;
 COMPUTE LINECOST = SUM.COST;
 COMPUTE LINECOGS = LINECOST * QUANTITY;

ON TABLE MERGE INTO FILE dmrpts
MATCHING TRG.PROD_NUM EQ SRC.PROD_NUM;WHEN MATCHED UPDATE
  QUANTITY=SRC.QUANTITY;
  YRMTH = TRG.YRMTH+1;
  LINEPRICE=SRC.LINEPRICE;
  LINECOGS=SRC.LINECOGS;
  PROFIT=SRC.LINECOGS - SRC.LINECOST;

WHEN NOT MATCHED INSERT
  STORE_CODE='1004NY';
  PROD_NUM=SRC.PROD_NUM;
  YRMTH=202101;
  QUANTITY=SRC.QUANTITY;
```

```
  LINEPRICE=SRC.LINEPRICE;
  LINECOGS=SRC.LINECOGS;
  PROFIT=SRC.LINECOGS - SRC.LINECOST;
END
```

次の TABLE リクエストは、更新された dmrpts ターゲットの値を表示します。

```
TABLE FILE DMRPTS
PRINT *
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

下図は、出力の最終ページを示しています。既存のレコードで更新された値および挿入されたレコード (最後の 2 行) が表示されています。

Product Number	Store Code	Year Month	Quantity	Line Total	Line Cost	Item Profit
1034	1003GA	200504	21,000	$499.00	7,329,000.00	7,328,651.00
1034	1003MA	200503	21,000	$499.00	7,329,000.00	7,328,651.00
1034	1003MD	200503	21,000	$499.00	7,329,000.00	7,328,651.00
1034	1003MO	200504	21,000	$499.00	7,329,000.00	7,328,651.00
1034	1003SC	200503	21,000	$499.00	7,329,000.00	7,328,651.00
1034	2011MN	200504	21,000	$499.00	7,329,000.00	7,328,651.00
1034	2011TX	200503	21,000	$499.00	7,329,000.00	7,328,651.00
1034	2012CA	200504	21,000	$499.00	7,329,000.00	7,328,651.00
1034	2012MI	200503	21,000	$499.00	7,329,000.00	7,328,651.00
1034	2012MI	200504	21,000	$499.00	7,329,000.00	7,328,651.00
1034	2021DC	200503	21,000	$499.00	7,329,000.00	7,328,651.00
1034	3002FL	200503	21,000	$499.00	7,329,000.00	7,328,651.00
1034	3002GA	200504	21,000	$499.00	7,329,000.00	7,328,651.00
1034	3002IL	200503	21,000	$499.00	7,329,000.00	7,328,651.00
1034	4003NJ	200503	21,000	$499.00	7,329,000.00	7,328,651.00
1036	2012MI	200503	33,000	$299.00	8,217,000.00	8,216,751.00
1036	2012MI	200504	33,000	$299.00	8,217,000.00	8,216,751.00
1036	3002FL	200503	33,000	$299.00	8,217,000.00	8,216,751.00
1036	3002GA	200504	33,000	$299.00	8,217,000.00	8,216,751.00
1036	3002IL	200503	33,000	$299.00	8,217,000.00	8,216,751.00
1036	4003NJ	200503	33,000	$299.00	8,217,000.00	8,216,751.00
2002	1004NY	202102	1,000	$80.00	40,000.00	39,960.00
2003	1004NY	202102	1,000	$200.00	100,000.00	99,900.00

ON TABLE MERGE によるレコードの追加

この例では、ソースとして GGSALES、ターゲットとして DATREC フォーマットの GGPRODS である GGDATREC を使用します。下図は、元の状態の GGDATREC を示しています。

Product Code	Product	Vendor ID	Vendor Name	Package	Size	Unit Price
B141	Hazelnut	V082	Coffee Connection	Pounds	16	58.00
B142	French Roast	V083	European Specialities,	Pounds	12	81.00
B144	Kona	V081	Evelina Imports, Ltd	Pounds	12	76.00
F101	Scone	V102	Ridgewood Bakeries	Case	20	13.00
F102	Biscotti	V100	Delancey Bakeries	Case	24	17.00
F103	Croissant	V103	West Side Bakers	Case	20	28.00
G100	Mug	V305	NY Ceramic Supply	Case	24	26.00
G104	Thermos	V305	ThermoTech, Inc	Case	16	96.00
G110	Coffee Grinder	V303	Appliance Craft	Case	12	125.00
G121	Coffee Pot	V304	Appliance Craft	Case	8	140.00

次のリクエストは、ON TABLE MERGE を使用して GGDATREC ターゲットの末尾に 1 件のレコードを追加します。MATCHING 句および WHEN NOT MATCHED 句は使用されていません。

```
TABLE FILE ggsales
SUM
   FST.PCD
   FST.PRODUCT
   FST.UNITS
   FST.DOLLARS

ON TABLE MERGE INTO FILE GGDATREC
INSERT
  PRODUCT_ID=SRC.PCD;
  PRODUCT_DESCRIPTION=SRC.PRODUCT;
  VENDOR_CODE='V400';
  VENDOR_NAME='Acme Foods';
  PACKAGE_TYPE='Case';
  SIZE = 18;
  UNIT_PRICE = SRC.DOLLARS/SRC.UNITS;
END
```

次の TABLE リクエストは、更新された GGDATREC 内の値を表示します。

```
TABLE FILE GGDATREC
PRINT *
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

下図は、追加されたレコード (最終行) を示しています。

Product Code	Product	Vendor ID	Vendor Name	Package	Size	Unit Price
B141	Hazelnut	V082	Coffee Connection	Pounds	16	58.00
B142	French Roast	V083	European Specialities,	Pounds	12	81.00
B144	Kona	V081	Evelina Imports, Ltd	Pounds	12	76.00
F101	Scone	V102	Ridgewood Bakeries	Case	20	13.00
F102	Biscotti	V100	Delancey Bakeries	Case	24	17.00
F103	Croissant	V103	West Side Bakers	Case	20	28.00
G100	Mug	V305	NY Ceramic Supply	Case	24	26.00
G104	Thermos	V305	ThermoTech, Inc	Case	16	96.00
G110	Coffee Grinder	V303	Appliance Craft	Case	12	125.00
G121	Coffee Pot	V304	Appliance Craft	Case	8	140.00
C144	Capuccino	V400	Acme Foods	Case	18	15.00

ON TABLE MERGE によるレコードの削除

次のリクエストでは、[Data Migrator - 全般] チュートリアルで生成された dminv テーブルおよび単一セグメントの dmord を使用します。

次のリクエストは、dmordsgl データソースに存在するすべての製造所を表示します。

```
TABLE FILE DMORDSGL
SUM QUANTITY
BY PLANT
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
END
```

出力結果は次のとおりです。

Plant Location	Quantity
BOS	16,046
DAL	9,756
LA	3,605
ORL	11,970
STL	26,198

次の ON TABLE MERGE リクエストは、ソースとターゲットの製品番号が一致し、かつ製造所が 'LA' であるレコードを削除します。

```
TABLE FILE dminv
PRINT
PROD_NUM
ON TABLE MERGE INTO FILE dmordsgl
MATCHING TRG.PROD_NUM EQ SRC.PROD_NUM;
WHEN MATCHED AND PLANT EQ 'LA'; DELETE
END
```

このリクエストを実行し、さらに製造所を表示するリクエストを実行すると、下図のように、製造所が 'LA' であるレコードが削除されていることが分かります。

Plant Location	Quantity
BOS	16,046
DAL	9,756
ORL	11,970
STL	26,198

ON TABLE MERGE のベストプラクティス

- ターゲットデータソースを元のコンテンツに戻す必要が生じる可能性があるため、ON TABLE MERGE を使用する前に、このファイルのバックアップを作成することをお勧めします。
- 安全性と明確性を確保するため、結合式内では、常に SRC (ソース) および TRC (ターゲット) 接頭語を使用してください。
- ON TABLE MERGE 句の前の TABLE コマンドは、内部的に HOLD ファイルを作成し、この HOLD ファイルからターゲットファイルへの結合が実行されます。HOLD ファイルは、ソースファイルのエイリアス名 (例、E01、E02...) を生成します。TABLE リクエ

ストのフィールド名は HOLD ファイルに重複するフィールド名を生成する可能性がある一方で、これらの名前は各フィールドで固有になります。フィールド名が重複する場合は、エイリアス名を使用するか AS 句により新しい名前を付けることで、固有名を使用する必要があります。たとえば、次のリクエストは DOLLARS という名前のフィールドを 3 つ生成します。

```
TABLE FILE GGSALES
SUM DOLLARS MAX.DOLLARS MIN.DOLLARS
BY UNITS
ON TABLE HOLD
END
```

生成されたマスターファイルには、重複するフィールド名および固有のエイリアス名が表示されます。

```
FILENAME=HOLD, SUFFIX=FIX     , IOTYPE=BINARY, $
  SEGMENT=HOLD, SEGTYPE=S1, $
    FIELDNAME=UNITS, ALIAS=E01, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=DOLLARS, ALIAS=E02, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=DOLLARS, ALIAS=E03, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=DOLLARS, ALIAS=E04, USAGE=I08, ACTUAL=I04, $
```

次のリクエストは、AS 句を使用して固有のフィールド名を生成します。

```
TABLE FILE GGSALES
SUM DOLLARS AS TOTDOLL
MAX.DOLLARS AS MAXDOLL
MIN.DOLLARS AS MINDOLL
BY UNITS
ON TABLE SET ASNAMES ON
ON TABLE HOLD AS UNIQNAME
END
```

このリクエストは、次のマスターファイルを生成します。

```
FILENAME=UNIQNAME, SUFFIX=FIX     , IOTYPE=BINARY, $
  SEGMENT=UNIQNAME, SEGTYPE=S1, $
    FIELDNAME=UNITS, ALIAS=E01, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=TOTDOLL, ALIAS=E02, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=MAXDOLL, ALIAS=E03, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=MINDOLL, ALIAS=E04, USAGE=I08, ACTUAL=I04, $
```

このマスターファイルを使用する際は、フィールド名とエイリアス名のいずれかを使

用することができます。また、ON TABLE MERGE リクエストでは、AS 名を HOLD マスターファイルに継承するための ON TABLE SET ASNAMES ON コマンドは必要ありません。

- ON TABLE MERGE リクエストを最適化する場合、SQL リクエストが生成されて DBMS に送信されます。たとえば、次のリクエストは dminv テーブルと dmrpts テーブルが、同一のアダプタおよび接続を使用する場合に最適化されます。

```
TABLE FILE dminv
PRINT
 PROD_NUM
 COMPUTE QUANTITY =  SUM.QTY_IN_STOCK;
 COMPUTE LINEPRICE =  SUM.PRICE;
 COMPUTE LINECOST = SUM.COST;
 COMPUTE LINECOGS = LINECOST * QUANTITY
ON TABLE MERGE INTO FILE dmrpts
MATCHING TRG.PROD_NUM EQ SRC.PROD_NUM;
 WHEN MATCHED UPDATE
  YRMTH=TRG.YRMTH + 1;
  QUANTITY=SRC.QUANTITY;
  LINEPRICE=SRC.LINEPRICE;
  LINECOGS=SRC.LINECOGS;
  PROFIT=SRC.LINECOGS - SRC.LINECOST;

 WHEN NOT MATCHED INSERT
  STORE_CODE='1004NY';
  PROD_NUM=SRC.PROD_NUM;
  YRMTH=202101;
  QUANTITY=SRC.QUANTITY;
  LINEPRICE=SRC.LINEPRICE;
  LINECOGS=SRC.LINECOGS;
  PROFIT=SRC.LINECOGS - SRC.LINECOST;
END
```

次のメッセージは、リクエストが最適化されたことを示しています。

```
(FOC2659) SUBSELECT 付き % の完全最適化が完了しました。
(FOC2661) TARGET FILE ibisamp/dmrpts

(FOC2665) MERGE PROCESS STARTED AT 15.33.19
(FOC1796) MERGE ステートメントの影響を受ける行: 182
```

Reporting Server Web コンソールの [ユーザオプション] メニューから [セッションログ] を開くことができます。セッションログを確認すると、次の SQL リクエストが生成されたことが分かります。

```
SELECT
  T1."PROD_NUM" AS "E01",
  T1."QTY_IN_STOCK" AS "E02",
  T1."PRICE" AS "E03",
  T1."COST" AS "E04",
  (T1."COST" * T1."QTY_IN_STOCK") AS "E05",
  '1004NY' AS "E06",
  202101 AS "E07",
  T1."QTY_IN_STOCK" AS "E08",
  ((T1."COST" * T1."QTY_IN_STOCK") - T1."COST") AS "E09"
   FROM
  dminv T1;
```

```
MERGE INTO dmrpts AS T3
  USING ( SELECT
  T1."PROD_NUM" AS "E01",
  T1."QTY_IN_STOCK" AS "E02",
  T1."PRICE" AS "E03",
  T1."COST" AS "E04",
  (T1."COST" * T1."QTY_IN_STOCK") AS "E05",
  '1004NY' AS "E06",
  202101 AS "E07",
  T1."QTY_IN_STOCK" AS "E08",
  ((T1."COST" * T1."QTY_IN_STOCK") - T1."COST") AS "E09"
   FROM
  dminv T1) T2
  ON
  (T3."PROD_NUM" =
  T2."E01")
```

```
 WHEN MATCHED THEN UPDATE SET
  "YRMTH" = ("YRMTH" + 1),
  "QUANTITY" =
  T2."E08",
  "LINEPRICE" =
  T2."E03",
  "LINECOGS" =
  T2."E05",
  "PROFIT" =
  T2."E09"
```

```
WHEN NOT MATCHED THEN INSERT (
```

```
   "PROD_NUM",
   "STORE_CODE",
   "YRMTH",
   "QUANTITY",
   "LINEPRICE",
   "LINECOGS",
   "PROFIT")
   VALUES (
   T2."E01",
   T2."E06",
   T2."E07",
   T2."E08",
   T2."E03",
   T2."E05",
   T2."E09");
```

ON TABLE MERGE でのエイリアス名の使用

次のリクエストは、フィールドをターゲットファイルに挿入する式で、ソース HOLD ファイルのエイリアス名を使用します。

```
TABLE FILE dminv
PRINT
   PROD_NUM
   COMPUTE YM = 202102;
   COMPUTE QUANTITY = QTY_IN_STOCK;
   COMPUTE LINEPRICE = PRICE;
   COMPUTE LINECOGS = LINEPRICE * QUANTITY;
ON TABLE MERGE INTO FILE ibisamp/dmrpts
 INSERT
  STORE_CODE='AAAA';
  PROD_NUM=SRC.E01;
  YRMTH=SRC.E02;
  QUANTITY=SRC.E03;
  LINEPRICE=SRC.E04;
  LINECOGS=SRC.E05;
  PROFIT=SRC.E04 - SRC.E05;
END
```

出力ファイルでのテキストフィールドの使用

テキストフィールドは、ALPHA、WP、FOCUS、XFOCUS フォーマットの HOLD および SAVE ファイルに継承することができます。ただし、ALPHA フォーマットではマスターファイルが生成されますが、テキストフィールドが含まれた ALPHA フォーマットの HOLD ファイルに対して TABLE リクエストを発行することはできません。

出力ファイルでのテキストフィールドの規則

- 必要に応じて、このファイルに任意の数のテキストフィールドを含めることができます。ただし、テキストフィールドは、表示リスト (PRINT...、SUM... など) のテキスト以外のフィールドの後に指定する必要があります。
- リクエストに ACROSS 句が含まれる場合は、表示リストのテキストフィールドは 1 つだけ指定することができ、テキスト以外のフィールドを指定することはできません。

HOLD および SAVE ファイルでは、テキストフィールドのミッシング値には、次の規則が適用されます。

- ブランク行は有効なデータです。テキストの末尾の記号はフィールドの終わりを示します。
- フィールドにテキストがない場合、HOLD または SAVE ファイルに単一のピリオド (.) と複数のブランクが表示されます。
- データ抽出時に MISSING=ON の場合、HOLD または SAVE ファイルにピリオド (.) が書き出されます。
- データ抽出時に MISSING=OFF の場合、HOLD または SAVE ファイルにブランクが書き出されます。

詳細は、ミッシングフィールド値を含むレコード処理を参照してください。

HOLD ファイルでのテキストフィールド規則の適用

次のリクエストは、「CRSEHOLD」という名前の HOLD ファイルにデータを抽出します。

```
TABLE FILE COURSE
PRINT COURSECODE DESCRIPTN1
ON TABLE HOLD AS CRSEHOLD
END
```

次のリクエストは、HOLD ファイルのデータを表示します。

```
TABLE FILE CRSEHOLD
PRINT *
END
```

以下は、出力結果の一部を示しています。

COURSECODE	DESCRIPTN1
AMA130	FOR PROJECT LEADERS AND SYS ANALYSTS.
AMA680	FOR INDUSTRIAL MARKETERS.
AMA800	FOR MANAGERS AND HUMAN RESOURCE STAFF.
BIT420	FOR SENIOR MANAGERS.ANALYZE AND IMPROVE

BIT640	FOR ADMINISTRATOR OF LABOR CONTRACTS.
BIT650	FOR PROGRAMMERS, PROJECT LEADERS AND
EDP090	FOR PRODUCTION MANAGERS.TO DELEGATE AND
EDP130	FOR PROGRAMMERS, PROJECT LEADERS AND
EDP390	FOR MANAGERS.HELP APPLY MANAGERIAL
EDP690	FOR EXEC MANAGERS AND MKTG RESEARCH
EDP750	FOR MARKETING MANAGERS.ENABLE TO
MC230	FOR PROGRAMMERS, PROJECT LEADERS AND
MC90	FOR VICE PRES OF SALES AND MARKETING.
NAMA40	FOR MARKETING VICE PRES AND MANAGERS.

NAMA730	FOR MANAGERS OF SALESPEOPLE.TO ENHANCE
NAMA930	FOR FINANCIAL MANAGERS.EXPLORE CONCEPTS
PDR330	FOR EXECUTIVE SECRETARIES.TO KNOW THE
PDR740	FOR EXPERIENCED MARKETING MANAGERS.HELP
PDR870	FOR DESIGNERS AND TRAINING COORDINATORS.
PU168	FOR MARKETING, PRODUCT, ADVERTISING
PU440	FOR GENERAL MANAGERS.TO EXPLORE
SFC280 . . .	FOR MANAGERS AND SECRETARIES.HELP

区切り付きシーケンシャルファイルの作成

HOLD FORMAT DFIX コマンドを使用して、任意の文字、または文字の組み合わせで区切られた文字フォーマットのシーケンシャルファイルを作成することができます。また、文字値を引用符またはその他の囲み文字で囲むかどうか、フィールド名を列記したヘッダレコードを

含めるかどうか、文字データの先頭および末尾のブランクを保持するかどうか、出力ファイルのレコード間に区切り文字を挿入するかどうかを指定することもできます。RDELIMITER を含めた場合、RECFM は UB です。

生成される区切り付きシーケンシャルファイルの記述であるマスターファイルとアクセスファイルが作成されます。マスターファイルの SUFFIX は DFIX です。アクセスファイルでは、区切り文字および囲み文字 (存在する場合) を指定するほか、文字データの先頭および末尾のブランクを保持するかどうか、ヘッダレコードが存在するかどうか、レコード区切り文字挿入するかどうかを指定します。マスターファイルとアクセスファイルは、後で WebFOCUS を使用してシーケンシャルファイルを読み取る場合に役立ちます。

区切り付きシーケンシャルファイルの作成

```
ON TABLE {HOLD|PCHOLD} [AS filename] FORMAT DFIX
 DELIMITER delimiter
 [ENCLOSURE enclosure] [HEADER {YES|NO}]
 [PRESERVESPACE {YES|NO}]  [RDELIMITER rdelimiter]
```

説明

filename

作成されるファイルの名前です。名前を指定しない場合、デフォルト値の HOLD が使用されます。

delimiter

最大で 30 バイトの Null 以外の表示可能文字または非表示文字で構成された区切り文字です。ここでは、文字セマンティクスが適用されます。たとえば、DBCS 文字を使用する場合、最大で 60 バイトの区切り文字を使用することができます。文字は、0x 表記の 16 進数値で表すこともできます。非表示文字の場合、この方法で文字を指定する必要があります。表示可能文字と非表示文字を併用する場合は、すべてを 16 進数の値として入力する必要があります。タブ区切りファイルを作成するには、DELIMITER 値を「TAB」またはその文字に相当する 16 進数の値で指定します (ASCII プラットフォームでは 0x09、EBCDIC プラットフォームでは 0x05)。一重引用符区切りファイルを作成するには、一重引用符 (,) の DELIMITER 値を、その文字に相当する 16 進数の値で指定する必要があります (ASCII プラットフォームでは 0x27、EBCDIC プラットフォームでは 0x7D)。このように指定しない場合、リクエストが誤って解釈され、使用不可能な HOLD ファイルが生成されます。

enclosure

囲み文字です。ファイル内の各文字の値を囲むための、最大で 4 バイトの表示可能文字または非表示文字を指定することができます。ここでは、文字セマンティクスが適用されます。たとえば、DBCS 文字を使用する場合、最大で 8 バイトの囲み文字を使用することができます。囲み文字では、すべてまたはその一部として、ほとんどの文字を使用することができます。ただし、囲み文字において、ピリオド (.)、プラス符号 (+)、マイナス符号 (-) などの数値に関する符号を使用することはできません。また、囲み文字として一重引用符 (') を指定する場合は、一重引用符を続けて 4 回入力する必要があります。最も一般的な囲み文字は、二重引用符 (") です。

表示可能文字と非表示文字を併用する場合は、すべてを 16 進数の値として入力する必要があります。表示可能文字の場合、実際の文字を使用することも、その文字に相当する 16 進数の値を使用することもできます。たとえば、アンパサンド (&) は、囲み文字の一部としてではなく、変数名の先頭文字として解釈される場合があります。

HEADER {YES|NO}

リクエストで生成された区切り付きシーケンシャルファイルに、フィールド名が格納されたヘッダレコードを含めるかどうかを指定します。

PRESERVESPACE {YES|NO}

文字データの先頭および末尾のブランクを保持するかどうかを指定します。YES を指定すると、先頭および末尾のブランクが保持されます。NO を指定すると、囲み文字で囲まれている先頭および末尾のブランクのみが保持されます。デフォルト値は NO です。

注意：PRESERVESPACE は、ENCLOSURE オプションにより上書きされます。そのため、PRESERVESPACE 設定を有効にするには、ENCLOSURE オプションを除外します。

rdelimiter

最大で 30 バイトの Null 以外の表示可能文字または非表示文字で構成されたレコード区切り文字です。ここでは、文字セマンティクスが適用されます。たとえば、DBCS 文字を使用する場合、最大で 60 バイトの区切り文字を使用することができます。非表示文字の場合、その文字を表す 16 進数の値を入力します。表示可能文字と非表示文字を併用する場合は、すべてを 16 進数の値として入力する必要があります。表示可能文字の場合、実際の文字を使用することも、その文字に相当する 16 進数の値を使用することもできます。たとえば、アンパサンド (&) は、区切り文字の一部としてではなく、変数名の先頭文字として解釈される場合があります。レコード区切り文字としてタブ文字を使用するには、区切り文字の値を「TAB」またはその文字に相当する 16 進数の値で指定します (ASCII プラットフォームでは 0x09、EBCDIC プラットフォームでは 0x05)。レコード区切り文字としてカンマ (,) はサポートされません。

区切り文字には、ピリオド (.)、プラス符号 (+)、マイナス符号 (-) などの数値に関する符号を使用することはできません。RDELIMITER を含めた場合、RECFM は UB です。

HOLD FORMAT DFIX 使用上の注意

- ミッシングデータに、値は表示されません。このため、囲み文字を使用する場合、ミッシングの文字フィールドは、2 つの囲み文字で表され、ミッシングの数値フィールドは 2 つの区切り文字で表されます。
- HOLD FORMAT DFIX で、テキストフィールドはサポートされません。
- HOLD FORMAT DFIX は単一のセグメントファイルを作成しますが、生成されるマスターファイルとアクセスファイルに、セグメントを手動で追加することができます。アクセスファイルで、セグメントごとに異なる区切り文字や囲み文字を指定することができます。
- 生成されたシーケンシャルファイルの拡張子は .ftm です。
- HOLD FORMAT DFIX で PRESERVESPACE YES オプションを使用すると、文字データの先頭および末尾のブランクが保持された状態で、ファイルが作成されます。このオプションを使用すると、アクセスファイルに PRESERVESPACE=YES 属性が追加されます。この属性が追加されると、FORMAT DFIX ファイルを読み取る際に、先頭および末尾のブランクが保持されます。

パイプ区切りファイルの作成

次のリクエストは、CENTORD データソースに対して実行され、フィールドをパイプ文字 (|) で区切った「PIPE1」というシーケンシャルファイルを作成します。文字の値は、引用符では囲まれず、ヘッダレコードは存在しません。

```
TABLE FILE CENTORD
SUM QUANTITY LINEPRICE BY REGION BY YEAR
ON TABLE HOLD AS PIPE1 FORMAT DFIX DELIMITER |
END
```

PIPE1 マスターファイルでは、SUFFIX 値に DFIX が指定されます。

```
FILENAME=PIPE1   , SUFFIX=DFIX    , $
  SEGMENT=PIPE1, SEGTYPE=S2, $
    FIELDNAME=REGION, ALIAS=E01, USAGE=A5, ACTUAL=A05, $
    FIELDNAME=YEAR, ALIAS=E02, USAGE=YY, ACTUAL=A04, $
    FIELDNAME=QUANTITY, ALIAS=E03, USAGE=I8C, ACTUAL=A08, $
    FIELDNAME=LINEPRICE, ALIAS=E04, USAGE=D12.2MC, ACTUAL=A12, $
```

PIPE1 アクセスファイルでは、区切り文字が指定されます。

```
SEGNAME=PIPE1, DELIMITER=|, HEADER=NO, $
```

PIPE1 シーケンシャルファイルに含まれるデータは次のとおりです。各データ値は、パイプ文字 (|) で次の値と分離されます。

```
EAST|2000|3907|1145655.77
EAST|2001|495922|127004359.88
EAST|2002|543678|137470917.05
NORTH|2001|337168|85750735.54
NORTH|2002|370031|92609802.80
SOUTH|2000|3141|852550.45
SOUTH|2001|393155|99822662.88
SOUTH|2002|431575|107858412.0
WEST|2001|155252|39167974.18
WEST|2002|170421|42339953.45
```

次の HOLD コマンドは、区切り文字 (|) と囲み文字 (“) を指定します。

```
ON TABLE HOLD AS PIPE1 FORMAT DFIX DELIMITER | ENCLOSURE "
```

マスターファイルに変化はありませんが、アクセスファイルには、囲み文字が指定されます。

```
SEGNAME=PIPE1, DELIMITER=|, ENCLOSURE=", HEADER=NO, $
```

作成される区切り付きファイルでは、各データ値は、パイプ文字 (|) で分離され、文字の値は二重引用符 (") で囲まれます。

```
"EAST"|2000|3907|1145655.77
"EAST"|2001|495922|127004359.88
"EAST"|2002|543678|137470917.05
"NORTH"|2001|337168|85750735.54
"NORTH"|2002|370031|92609802.80
"SOUTH"|2000|3141|852550.45
"SOUTH"|2001|393155|99822662.88
"SOUTH"|2002|431575|107858412.01
"WEST"|2001|155252|39167974.18
"WEST"|2002|170421|42339953.45
```

次の HOLD コマンドは、生成されるファイルにヘッダレコードを追加します。

```
ON TABLE HOLD AS PIPE1 FORMAT DFIX DELIMITER |  ENCLOSURE "  HEADER YES
```

マスターファイルに変化はありませんが、アクセスファイルには、生成されたシーケンシャルファイルの最初のレコードとして、フィールド名のヘッダレコードを格納することが指定されます。

```
SEGNAME=PIPE1, DELIMITER=|, ENCLOSURE=", HEADER=YES, $
```

作成される区切り付きファイルでは、各データ値は、パイプ文字 (|) で分離され、文字の値は二重引用符 (") で囲まれます。最初のレコードには、フィールド名が格納されます。

```
"REGION"|"YEAR"|"QUANTITY"|"LINEPRICE"
"EAST"|2000|3907|1145655.77
"EAST"|2001|495922|127004359.88
"EAST"|2002|543678|137470917.05
"NORTH"|2001|337168|85750735.54
"NORTH"|2002|370031|92609802.80
"SOUTH"|2000|3141|852550.45
"SOUTH"|2001|393155|99822662.88
"SOUTH"|2002|431575|107858412.01
"WEST"|2001|155252|39167974.18
"WEST"|2002|170421|42339953.45
```

タブ区切りファイルの作成

次のリクエストは、CENTORD データソースに対して実行され、フィールドをパイプ文字 (|) で区切った「TAB1」というシーケンシャルファイルを作成します。

```
TABLE FILE CENTORD
SUM QUANTITY LINEPRICE BY REGION BY YEAR
ON TABLE HOLD AS TAB1 FORMAT DFIX DELIMITER TAB
END
```

タブは非表示の文字であるため、TAB1 アクセスファイルでは、この区切り文字を 16 進数の値で表示します。

以下は、ASCII 環境のアクセスファイルを示しています。

```
SEGNAME=TAB1, DELIMITER=0x09, HEADER=NO, $
```

区切り付きファイルの作成とブランクの保持

次のリクエストは、GGSALES データソースに対して実行され、カンマ区切りファイルを作成します。元の文字データの末尾にはブランクが含まれています。HOLD コマンドで PRESERVESPACE YES オプションを使用すると、これらの末尾のブランクが保持されます。

```
APP HOLDDATA APP1
APP HOLDMETA APP1
TABLE FILE GGSALES
SUM DOLLARS UNITS
BY REGION
BY CATEGORY
BY PRODUCT
ON TABLE HOLD AS DFIX1 FORMAT DFIX DELIMITER , PRESERVESPACE YES
END
```

次のマスターファイルが生成されます。

```
FILENAME=DFIX1   , SUFFIX=DFIX    , $
  SEGMENT=DFIX1, SEGTYPE=S3, $
    FIELDNAME=REGION, ALIAS=E01, USAGE=A11, ACTUAL=A11, $
    FIELDNAME=CATEGORY, ALIAS=E02, USAGE=A11, ACTUAL=A11, $
    FIELDNAME=PRODUCT, ALIAS=E03, USAGE=A16, ACTUAL=A16, $
    FIELDNAME=DOLLARS, ALIAS=E04, USAGE=I08, ACTUAL=A08, $
    FIELDNAME=UNITS, ALIAS=E05, USAGE=I08, ACTUAL=A08, $
```

次のアクセスファイルが生成されます。

```
SEGNAME=DFIX1, DELIMITER=',', HEADER=NO, PRESERVESPACE=YES, $
```

DFIX1 ファイルの文字フィールドには、元のファイルに存在するブランクがすべて保持されます。

```
Midwest    ,Coffee     ,Espresso        ,1294947,101154
Midwest    ,Coffee     ,Latte           ,2883566,231623
Midwest    ,Food       ,Biscotti        ,1091727,86105
Midwest    ,Food       ,Croissant       ,1751124,139182
Midwest    ,Food       ,Scone           ,1495420,116127
Midwest    ,Gifts      ,Coffee Grinder  ,619154,50393
Midwest    ,Gifts      ,Coffee Pot      ,599878,47156
Midwest    ,Gifts      ,Mug             ,1086943,86718
```

```
Midwest    ,Gifts      ,Thermos          ,577906,46587
Northeast  ,Coffee     ,Capuccino        ,542095,44785
Northeast  ,Coffee     ,Espresso         ,850107,68127
Northeast  ,Coffee     ,Latte            ,2771815,222866
Northeast  ,Food       ,Biscotti         ,1802005,145242
Northeast  ,Food       ,Croissant        ,1670818,137394
Northeast  ,Food       ,Scone            ,907171,70732
Northeast  ,Gifts      ,Coffee Grinder   ,509200,40977
Northeast  ,Gifts      ,Coffee Pot       ,590780,46185
Northeast  ,Gifts      ,Mug              ,1144211,91497
Northeast  ,Gifts      ,Thermos          ,604098,48870
Southeast  ,Coffee     ,Capuccino        ,944000,73264
Southeast  ,Coffee     ,Espresso         ,853572,68030
Southeast  ,Coffee     ,Latte            ,2617836,209654
Southeast  ,Food       ,Biscotti         ,1505717,119594
Southeast  ,Food       ,Croissant        ,1902359,156456
Southeast  ,Food       ,Scone            ,900655,73779
Southeast  ,Gifts      ,Coffee Grinder   ,605777,47083
Southeast  ,Gifts      ,Coffee Pot       ,645303,49922
Southeast  ,Gifts      ,Mug              ,1102703,88474
Southeast  ,Gifts      ,Thermos          ,632457,48976
West       ,Coffee     ,Capuccino        ,895495,71168
West       ,Coffee     ,Espresso         ,907617,71675
West       ,Coffee     ,Latte            ,2670405,213920
West       ,Food       ,Biscotti         ,863868,70436
West       ,Food       ,Croissant        ,2425601,197022
West       ,Food       ,Scone            ,912868,72776
West       ,Gifts      ,Coffee Grinder   ,603436,48081
West       ,Gifts      ,Coffee Pot       ,613624,47432
West       ,Gifts      ,Mug              ,1188664,93881
West       ,Gifts      ,Thermos          ,571368,45648
```

PRESERVESPACE NO オプションを使用して同一のファイルを作成すると、末尾のブランクが削除されます。

```
Midwest,Coffee,Espresso,1294947,101154
Midwest,Coffee,Latte,2883566,231623
Midwest,Food,Biscotti,1091727,86105
Midwest,Food,Croissant,1751124,139182
Midwest,Food,Scone,1495420,116127
Midwest,Gifts,Coffee Grinder,619154,50393
Midwest,Gifts,Coffee Pot,599878,47156
Midwest,Gifts,Mug,1086943,86718
Midwest,Gifts,Thermos,577906,46587
Northeast,Coffee,Capuccino,542095,44785
Northeast,Coffee,Espresso,850107,68127
```

```
Northeast,Coffee,Latte,2771815,222866
Northeast,Food,Biscotti,1802005,145242
Northeast,Food,Croissant,1670818,137394
Northeast,Food,Scone,907171,70732
Northeast,Gifts,Coffee Grinder,509200,40977
Northeast,Gifts,Coffee Pot,590780,46185
Northeast,Gifts,Mug,1144211,91497
Northeast,Gifts,Thermos,604098,48870
Southeast,Coffee,Capuccino,944000,73264
Southeast,Coffee,Espresso,853572,68030
Southeast,Coffee,Latte,2617836,209654
Southeast,Food,Biscotti,1505717,119594
Southeast,Food,Croissant,1902359,156456
Southeast,Food,Scone,900655,73779
Southeast,Gifts,Coffee Grinder,605777,47083
Southeast,Gifts,Coffee Pot,645303,49922
Southeast,Gifts,Mug,1102703,88474
Southeast,Gifts,Thermos,632457,48976
West,Coffee,Capuccino,895495,71168
West,Coffee,Espresso,907617,71675
West,Coffee,Latte,2670405,213920
West,Food,Biscotti,863868,70436
West,Food,Croissant,2425601,197022
West,Food,Scone,912868,72776
West,Gifts,Coffee Grinder,603436,48081
West,Gifts,Coffee Pot,613624,47432
West,Gifts,Mug,1188664,93881
West,Gifts,Thermos,571368,45648
```

レコード区切り文字の指定

次のリクエストは、GGSALES データソースに対して実行され、フィールド区切り文字をカンマ (,)、囲み文字を一重引用符 (') に指定します。また、表示可能文字と非表示文字の両方で構成されたレコード区切り文字を、次の 16 進数シーケンスとして指定します。

- 0x - 区切り文字を 16 進数文字コードとして識別する文字シーケンス
- 2C - カンマ (,) の 16 進数値
- 24 - ドル記号 ($) の 16 進数値
- 0D - キャリッジリターンの 16 進数値
- 0A - 新規行の 16 進数値

```
TABLE FILE GGSALES
PRINT DOLLARS UNITS CATEGORY REGION
ON TABLE HOLD AS RDELIM1 FORMAT DFIX DELIMITER , ENCLOSURE ''''
HEADER NO RDELIMITER 0x2C240D0A
END
```

生成されたマスターファイルは次のとおりです。

```
FILENAME=RDELIM1 , SUFFIX=DFIX    , $
  SEGMENT=RDELIM1, SEGTYPE=S0, $
    FIELDNAME=DOLLARS, ALIAS=E01, USAGE=I08, ACTUAL=A08, $
    FIELDNAME=UNITS, ALIAS=E02, USAGE=I08, ACTUAL=A08, $
    FIELDNAME=CATEGORY, ALIAS=E03, USAGE=A11, ACTUAL=A11, $
    FIELDNAME=REGION, ALIAS=E04, USAGE=A11, ACTUAL=A11, $
```

アクセスファイルには、上記の区切り文字および囲み文字が記述されます。

```
SEGNAME=RDELIM1,
  DELIMITER=',',
  ENCLOSURE='''',
  HEADER=NO,
  RDELIMITER=0x2C240D0A,
  PRESERVESPACE=NO, $
```

出力された DFIX ファイルの各行は、カンマ (,) とドル記号 ($) の組み合わせ、キャリッジリターン、ラインフィードで終了します。以下は、リストの一部を示しています。

```
20805,1387,'Coffee','Northeast',$
20748,1729,'Coffee','Northeast',$
20376,1698,'Coffee','Northeast',$
20028,1669,'Coffee','Northeast',$
19905,1327,'Coffee','Northeast',$
19470,1770,'Coffee','Northeast',$
19118,1738,'Coffee','Northeast',$
18720,1560,'Coffee','Northeast',$
18432,1536,'Coffee','Northeast',$
17985,1199,'Coffee','Northeast',$
17630,1763,'Coffee','Northeast',$
16646,1189,'Coffee','Northeast',$
15650,1565,'Coffee','Northeast',$
15450,1545,'Coffee','Northeast',$
15435,1029,'Coffee','Northeast',$
14270,1427,'Coffee','Northeast',$
```

HOLD ファイルのミッシングデータ

次のリクエストは、CENTORD データソースに対して実行され、結果のカンマ区切り HOLD ファイルに、ミッシングの文字の値と数値を作成します。

```
DEFINE FILE CENTORD
AREA/A5 MISSING ON = IF REGION EQ 'EAST' THEN MISSING ELSE REGION;
MQUANTITY/I9 MISSING ON = IF REGION EQ 'WEST' THEN MISSING ELSE 200;
END

TABLE FILE CENTORD
SUM QUANTITY MQUANTITY LINEPRICE BY AREA BY YEAR
WHERE AREA NE 'NORTH' OR 'SOUTH'
  ON TABLE HOLD AS MISS1 FORMAT DFIX DELIMITER , ENCLOSURE "
END
```

MISS1 HOLD ファイルでは、ミッシング文字値は連続する 2 つの囲み文字 (““) で表され、ミッシング数値は連続する 2 つの区切り文字 (,,) で表されます。

```
"",2000,3907,600,1145655.77
"",2001,495922,343000,127004359.88
"",2002,543678,343000,137470917.05
"WEST",2001,155252,,39167974.18
"WEST",2002,170421,,42339953.45
```

INTERNAL フォーマットでのレポート出力の保存

HOLD ファイルは、バイナリ整数およびパック 10 進数データ値を 4 バイト (4 の倍数)までパディングします。たとえば、3 桁の整数フィールド (I3) は HOLD ファイルに 4 バイトとして保存されます。COBOL などの 3 GL プログラムにより HOLD ファイルを正確に読み出すためには、パディングを追加せずに HOLD に保存する必要が生じる場合があります。

HOLD ファイルのフィールドパディングを抑制するには、次のように、リクエストでフィールドのフォーマットを再設定して、マスターファイルの USAGE フォーマットに対応するデフォルトの ACTUAL フォーマットを上書きする必要があります。

- HOLD ファイルでパディングを抑制する整数フィールドおよびパックフィールドのフォーマットを編集し、表示する長さを修正する。

- レポート出力に対して、HOLD FORMAT INTERNAL を指定する。

HOLD ファイルでのフィールドパディングの抑制

```
SET HOLDLIST = PRINTONLY
TABLE FILE filename
display_command fieldname/[In|Pn.d]
.
.
ON TABLE HOLD AS name FORMAT INTERNAL
END
```

説明

PRINTONLY

リクエストは、レポート出力で表示されている指定フィールドのみを含む HOLD ファイルを作成します。この設定をしないと、パディングされたフィールドの長さを含む、余分なフィールドが HOLD ファイルに追加されます。詳細は、HOLD マスターファイルの属性制御を参照してください。

fieldname/[In|Pn.d]

パディングを抑制する整数フィールドおよびパックフィールドの、フォーマットの正確な長さを指定します。これらのフォーマットは、マスターファイルの表示フォーマットに使用した ACTUAL フォーマットを上書きします。詳細は、INTERNAL フォーマットでのレポート出力の保存を参照してください。

倍精度浮動小数点数 (D) および単精度浮動小数点数 (F) は HOLD FORMAT INTERNAL により影響されません。

FORMAT INTERNAL

指定の整数フィールドおよびパック 10 進数フィールドにパディングを追加せずに HOLD ファイルを保存します。

HOLD ファイルでフィールドパディングを抑制する際の注意

- 1、2、3、または 4 バイトの整数フィールド (I) は、HOLD FORMAT INTERNAL を使用しないと 4 バイトの整数を生成します。
- パック 10 進数フィールド (P*n.d*) では、*n* が桁数の合計で、*d* が小数点の右側の桁数です。*n* を 2 で除算し、1 を加算することによりバイト数を計算します。

  構文は次のとおりです。

  ```
  bytes = INT (n/2) + 1
  ```

  説明

  INT (n/2)

  2 で除算した後の最大整数です。
- HOLD FORMAT INTERNAL により、倍精度浮動小数点数 (D) および単精度浮動小数点数 (F) フィールドは影響されません。D は 8 バイト、F は 4 バイトで保持されます。
- 文字フィールドにはソースマスターファイルからその長さが自動的に継承され、4 バイト (4 の倍数) までパディングされることはありません。
- 上書きしたフォーマットよりデータ値が長い場合、その値は切り捨てられます。HOLD ファイルのデータが整数の場合、切り捨てにより不正確になることがあります。パック 10 進数および整数の場合、下位の桁を正確な値の桁と同等にするために上位の桁で切り捨てられます。

  不正確な結果を避けるためには、データ値を格納するのに十分な長さのフォーマットを指定します。
- HOLDMISS=ON 設定で、ミッシング値を HOLD ファイルに生成する場合、短いパックフィールド、および I1、I2 、I3 フォーマットを含むフィールドはミッシング値を保持するには十分な長さではありません。

HOLD FORMAT INTERNAL を使用しない HOLD ファイルの作成

次の例では、RETAIL_COST、DEALER_COST、SEATS の ACTUAL 値がすべて完全なサイズになるようパディングされます。文字フィールドも完全なサイズを占めます。

```
TABLE FILE CAR
PRINT CAR COUNTRY RETAIL_COST DEALER_COST SEATS
ON TABLE HOLD AS DJG
END
```

このリクエストにより作成されるマスターファイルは次のとおりです。

```
FILE=DJG, SUFFIX=FIX
 SEGMENT=DJG, SEGTYPE=S0
  FIELDNAME=CAR          ,ALIAS=E01  ,USAGE=A16  ,ACTUAL=A16    ,$
  FIELDNAME=COUNTRY      ,ALIAS=E02  ,USAGE=A10  ,ACTUAL=A12    ,$
  FIELDNAME=RETAIL_COST  ,ALIAS=E03  ,USAGE=D7   ,ACTUAL=D08    ,$
  FIELDNAME=DEALER_COST  ,ALIAS=E04  ,USAGE=D7   ,ACTUAL=D08    ,$
  FIELDNAME=SEATS        ,ALIAS=E05  ,USAGE=I3   ,ACTUAL=I04    ,$
```

HOLD FORMAT INTERNAL を使用した HOLD ファイルの作成

次の例では、DEALER_COST および RETAIL_COST がマスターファイルに D フィールドとして定義され、リクエストにより RETAIL_COST が I2 フィールドに、DEALER_COST が P3 フィールドに上書きされます。

```
SET HOLDLIST=PRINTONLY
TABLE FILE CAR
PRINT CAR COUNTRY RETAIL_COST/I2 DEALER_COST/P3 SEATS/I1
ON TABLE HOLD AS HINT3 FORMAT INTERNAL
END
```

これにより作成されるマスターファイルは次のとおりです。

```
FILE=HINT3, SUFFIX=FIX
 SEGMENT=HINT3, SEGTYPE=S0
  FIELDNAME=CAR          ,ALIAS=E01   ,USAGE=A16  ,ACTUAL=A16   ,$
  FIELDNAME=COUNTRY      ,ALIAS=E02   ,USAGE=A10  ,ACTUAL=A10   ,$
  FIELDNAME=RETAIL_COST  ,ALIAS=E03   ,USAGE=I6   ,ACTUAL=I02   ,$
  FIELDNAME=DEALER_COST  ,ALIAS=E04   ,USAGE=P4   ,ACTUAL=P02   ,$
  FIELDNAME=SEATS        ,ALIAS=E05   ,USAGE=I4   ,ACTUAL=I01   ,$
```

上書きされたフィールドの ACTUAL フォーマットは I2、P2、I1 です。DEALER_COST の ACTUAL は P2 になります。これは、フォーマットによる上書きの P3、つまり ACTUAL の 2 桁に 3 つの表示桁を保存できるという意味です。文字フィールドも、パディングは追加されません。

HOLD FORMAT SQL_SCRIPT によるサブクエリまたはシーケンシャルファイルの作成

リレーショナルデータソースに対するリクエストで HOLD FORMAT SQL_SCRIPT コマンドを使用すると、現在のクエリの実行に必要な SQL SELECT ステートメントが生成され、.sql 拡張子のファイルとしてアプリケーションフォルダに格納されます。また、SQL アンサーセットを記述した一組のマスターファイルとアクセスファイルも同時に生成されます。

その他のタイプのデータソースに対するリクエストで HOLD FORMAT SQL_SCRIPT コマンドを使用すると、現在のクエリが実行され、取得された値が .ftm 拡張子のシーケンシャルファイルとしてアプリケーションフォルダに格納されます。また、取得されたデータを記述したマスターファイルも同時に生成されます。

HOLD FORMAT SQL_SCRIPT から取得された出力は、DB_INFILE 関数のターゲットファイルとして使用することができます。

注意：.sql ファイルおよび付属のマスターファイルの生成後、グローバルダイアログマネージャ変数を使用して、その .sql ファイルをカスタマイズすることができます。これらのグローバル変数は、マスターファイルで宣言する必要があります。

HOLD FORMAT SQL_SCRIPT による SQL スクリプトファイルまたはシーケンシャルファイルの作成

```
ON TABLE HOLD AS script_name FORMAT SQL_SCRIPT
```

説明

script_name

HOLD FORMAT SQL_SCRIPT コマンドの結果として作成される .sql ファイルまたは .ftm ファイルの名前です。

HOLD FORMAT SQL_SCRIPT による SQL スクリプトファイルの作成

次のリクエストは、WF_RETAIL リレーショナルデータソースを使用し、baseapp アプリケーション内に SQL スクリプトファイルを作成します。

```
APP HOLD baseapp
TABLE FILE WF_RETAIL_LITE
SUM BUSINESS_REGION STATE_PROV_CODE_ISO_3166_2
BY BUSINESS_REGION NOPRINT BY STATE_PROV_CODE_ISO_3166_2 NOPRINT
WHERE BUSINESS_REGION EQ 'North America' OR 'EMEA'
WHERE STATE_PROV_CODE_ISO_3166_2 EQ 'AR' OR 'IA' OR 'KS' OR 'KY' OR 'WY' OR
'CT' OR 'MA' OR '04' OR '11' OR '14'
OR 'NJ' OR 'NY' OR 'RI'
ON TABLE HOLD AS RETAIL_SCRIPT FORMAT SQL_SCRIPT
END
```

WF_RETAIL は、ユーザが作成可能なサンプルデータソースです。このデータソースを作成するには、Reporting Server の Web コンソールでアプリケーションを左クリックし、リボンメニューから [チュートリアル]を選択します。チュートリアルフレームワークの作成画面で [作成]、[OK] の順に選択します。

このリクエストの結果は、retail_script.sql というシーケンシャルファイルと、対応するマスターファイルおよびアクセスファイルです。

retail_script.sql ファイルには、次の SQL SELECT ステートメントが格納されています。

```
 SELECT          MAX(T3."BUSINESS_REGION") AS "VB001_MAX_BUSINESS_REGION",       MAX
(T3."STATE_PROV_CODE_ISO_3166_2")
AS "VB002_MAX_STATE_PROV_CODE_ISO_"     FROM   wrd_wf_retail_geography T3
WHERE  (T3."STATE_PROV_CODE_ISO_3166_2" IN('AR', 'IA', 'KS', 'KY', 'WY',
'CT', 'MA', '04', '11', '14', 'NJ', 'NY', 'RI'))
AND    (T3."BUSINESS_REGION" IN('North America', 'EMEA'))       GROUP BY
T3."BUSINESS_REGION",   T3."STATE_PROV_CODE_ISO_3166_2"
```

retail_script.mas マスターファイルは次のとおりです。

```
FILENAME=RETAIL_SCRIPT, SUFFIX=MSODBC  , $
  SEGMENT=RETAIL_SCRIPT, SEGTYPE=S0, $
    FIELDNAME=BUSINESS_REGION, ALIAS=VB001_MAX_BUSINESS_REGION, USAGE=A15V,
ACTUAL=A15V,
      MISSING=ON,
      TITLE='Customer,Business,Region', $
    FIELDNAME=STATE_PROV_CODE_ISO_3166_2, ALIAS=VB002_MAX_STATE_PROV_CODE_
ISO_, USAGE=A5V, ACTUAL=A5V,
      MISSING=ON,
      TITLE='Customer,State,Province,ISO-3166-2,Code', $
```

retail_script.acx アクセスファイルは次のとおりです。

```
SEGNAME=RETAIL_SCRIPT,
   CONNECTION=CON01,
   DATASET=RETAIL_SCRIPT.SQL,
   SUBQUERY=Y, $
```

HOLD FORMAT SQL_SCRIPT によるシーケンシャルファイルの作成

次のリクエストは、EMPLOYEE データソースを使用し、このリクエストで取得された値を格納するシーケンシャルファイルと、対応するマスターファイルを作成します。

```
APP HOLD baseapp
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME
WHERE DEPARTMENT EQ 'MIS'
```

```
ON TABLE HOLD AS EMPVALUES FORMAT SQL_SCRIPT
END
```

empvalues.ftm シーケンシャルファイルには、次のデータが格納されています。

```
SMITH           MARY        JONES           DIANE       MCCOY           JOHN
      BLACKWOOD       ROSEMARIE   GREENSPAN       MARY        CROSS
BARBARA
```

empvalues.mas マスターファイルは次のとおりです。

```
FILENAME=EMPVALUES, SUFFIX=DATREC  , IOTYPE=BINARY, $
  SEGMENT=EMPVALUE, SEGTYPE=S0, $
    FIELDNAME=LAST_NAME, ALIAS=E01, USAGE=A15, ACTUAL=A15, $
    FIELDNAME=FIRST_NAME, ALIAS=E02, USAGE=A10, ACTUAL=A10, $
    FIELDNAME=NULLFLAG, ALIAS=__NULLFLAG__, USAGE=A2, ACTUAL=A2B, ACCESS_
PROPERTY=(INTERNAL), $
```

構造化 HOLD ファイルの作成

構造化 HOLD ファイル は、オペレーティングシステム間でのデータソースおよびレポートのマイグレート機能を提供します。

その他の HOLD フォーマットは、元のソースからデータを取得し、リクエスト自体からの暗黙的な構造化要素を保持することができますが、アクセスした元のデータソースについてのほとんどの情報、および HOLD マスターファイルやデータソースとの内部関係が継承されることはありません。これに反して、構造化 HOLD ファイルは、元のデータソースと同等の構造にデータを抽出します。この HOLD ファイルに対するリクエストは、保持されたデータ関係を使用して、環境やタイプの異なるデータソースと同種の関係を再作成することができます。

構造化 HOLD ファイルは、ALPHA、BINARY、FOCUS フォーマットで作成することができます。

- ALPHA および BINARY で生成された構造化 HOLD ファイルは Flat File で、TABLE リクエストの条件を満たすデータを含むセグメントインスタンスを保存します。TABLE リクエストにより読み出された元の構造に基づいて、複数のセグメントが生成されます。セグメントの相違は、RECTYPE を割り当てることにより識別されます。元のデータソースの子セグメントは、HOLD ファイルのユニークセグメントになります。
- FOCUS フォーマットの構造化 HOLD ファイルは、標準 FOCUS セグメントを使用し、

元の構造を保持します。

いずれの場合も、HOLD ファイルには TABLE リクエスト自体に基づく完全なレポートの提供に必要な、元のセグメントインスタンスがすべて格納されます。元のリクエストが使用する表示コマンド (PRINT、LIST、SUM、COUNT) とは関係なく、構造化 HOLD ファイルは PRINT が使用されたときのように作成されます。集計は無視されます。

HOLD ファイルは、リクエストを満足させるためにリクエストが特定する構造内のすべてのフィールド、またはすべての表示フィールドと BY フィールドを格納します。このファイルには、リクエストで明確に参照されない一時項目 (DEFINE) は格納されません。リクエストで参照される 一時項目 (DEFINE) の評価に必要なフィールドは、すべて格納されます。

構造化 HOLD ファイルは、TABLE および TABLEF コマンドでのみサポートされます。HOLD ファイルがサポートされる任意の場所で作成することができます。構造化 HOLD ファイルを作成する前に、特定のリクエストで ON TABLE SET EXTRACT コマンドを発行し、構造化 HOLD ファイルを有効にする必要があります。

リクエストの構造化 HOLD ファイルの有効化

```
ON TABLE SET EXTRACT {ON|*|OFF}
```

説明

ON

このリクエストの構造化 HOLD ファイルを有効にし、リクエストに記述されたすべてのフィールドを抽出します。

*****

このリクエストの構造化 HOLD ファイルを有効にし、その後に一連の抽出オプションが続くことを示します。たとえば、構造化 HOLD ファイルから特定のフィールドを除外することができます。詳細は、構造化 HOLD ファイルの作成を参照してください。

OFF

このリクエストの構造化 HOLD ファイルを無効にします。デフォルト値は OFF です。

構造化 HOLD ファイルの作成

HOLD コマンドを発行する前に、ON TABLE SET EXTRACT コマンドを発行して、このリクエストに対する構造化 HOLD ファイルを有効にします。詳細は、構造化 HOLD ファイルの作成を参照してください。次に、HOLD コマンドを発行し、構造化 HOLD ファイルを作成します。

```
[ON TABLE] {HOLD|PCHOLD} [AS name] FORMAT {ALPHA|BINARY|FOCUS}
```

説明

name

HOLD ファイル名です。これを省略すると、デフォルト名の HOLD になります。

FORMAT

ALPHA、BINARY、または FOCUS です。

注意：SET コマンドを発行し、デフォルトの HOLD フォーマットを ALPHA または BINARY に設定することができます。

```
SET HOLDFORMAT=ALPHA
SET HOLDFORMAT=BINARY
```

構造化 HOLD ファイル生成オプションの設定

特定のフィールドを除外するなど、抽出ファイルの作成オプションを設定するには、SET EXTRACT コマンドの * オプションを使用します。

```
ON TABLE SET EXTRACT *
EXCLUDE = (fieldname1, fieldname2, fieldname3 , ..., fieldnamen),$
FIELDS={ALL|EXPLICIT},$
ENDEXTRACT
ON TABLE HOLD AS name FORMAT {ALPHA|BINARY|FOCUS}
```

説明

EXCLUDE=(fieldname1, fieldname2, fieldname3,..., fieldnamen)

HOLD ファイルから特定のフィールドを除外します。

,$

抽出ブロックの要素を区切るために必要な構文です。

ALL

実行するリクエストで使用するすべての実フィールド、およびすべての一時項目 (DEFINE) を含めます。

EXPLICIT

表示リスト、または BY フィールドリストにある実フィールドおよび一時項目 (DEFINE) のみを含めます。明示的に参照されていない一時項目 (DEFINE)、および DEFINE の評価のためのフィールドは含められません。

ENDEXTRACT

抽出ブロックを終了します。

ALPHA フォーマット構造化 HOLD ファイルの作成

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME JOBCODE ED_HRS
BY DEPARTMENT
BY HIGHEST SALARY
ON TABLE SET EXTRACT ON
ON TABLE HOLD FORMAT ALPHA
END
```

このリクエストは、次の HOLD マスターファイルを作成します。

```
FILENAME=HOLD    , SUFFIX=FIX     , $
  SEGMENT=EMPINFO, SEGTYPE=S0, $
    FIELDNAME=RECTYPE, ALIAS=R, USAGE=A3, ACTUAL=A3, $
    FIELDNAME=LAST_NAME, ALIAS='LN', USAGE=A15, ACTUAL=A15, $
    FIELDNAME=FIRST_NAME, ALIAS='FN', USAGE=A10, ACTUAL=A10, $
    FIELDNAME=DEPARTMENT, ALIAS='DPT', USAGE=A10, ACTUAL=A10, $
    FIELDNAME=ED_HRS, ALIAS='OJT', USAGE=F6.2, ACTUAL=A06, $
```

```
  SEGMENT=PAYINFO, SEGTYPE=S0, PARENT=EMPINFO, $
    FIELDNAME=RECTYPE, ALIAS=1, USAGE=A3, ACTUAL=A3, $
    FIELDNAME=SALARY, ALIAS='SAL', USAGE=D12.2M, ACTUAL=A12, $
    FIELDNAME=JOBCODE, ALIAS='JBC', USAGE=A3, ACTUAL=A03, $
```

ALPHA または BINARY 構造化 HOLD ファイルの RECTYPE フィールドが生成されます。HOLD ファイルの各レコードは、RECTYPE で開始し、元の構造でレコードが属するセグメントを示します。ルートセグメントには RECTYPE=R が割り当てられ、その他のセグメントの RECTYPE には、上から下、左から右の順序で連続番号が割り当てられます。

次は HOLD ファイルの最初のいくつかのレコードを示しています。

```
R  STEVENS        ALFRED    PRODUCTION 25.00
1      11000.00A07
1      10000.00A07
R  SMITH          MARY      MIS        36.00
1      13200.00B14
R  JONES          DIANE     MIS        50.00
1      18480.00B03
1      17750.00B02
R  SMITH          RICHARD   PRODUCTION 10.00
1       9500.00A01
1       9050.00B01
```

ibi FOCUS フォーマット構造化 HOLD ファイルの作成

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME JOBCODE ED_HRS
BY DEPARTMENT
BY HIGHEST SALARY
ON TABLE SET EXTRACT ON
ON TABLE HOLD FORMAT FOCUS
END
```

このリクエストは、次の HOLD マスターファイルを作成します。

```
FILENAME=HOLD    , SUFFIX=FOC     , $
  SEGMENT=EMPINFO, SEGTYPE=S0, $
```

```
    FIELDNAME=LAST_NAME, ALIAS='LN', USAGE=A15, $
    FIELDNAME=FIRST_NAME, ALIAS='FN', USAGE=A10, $
    FIELDNAME=DEPARTMENT, ALIAS='DPT', USAGE=A10, $
    FIELDNAME=ED_HRS, ALIAS='OJT', USAGE=F6.2, $
  SEGMENT=PAYINFO, SEGTYPE=S0, PARENT=EMPINFO, $
    FIELDNAME=SALARY, ALIAS='SAL', USAGE=D12.2M, $
    FIELDNAME=JOBCODE, ALIAS='JBC', USAGE=A3, $
```

構造化 HOLD ファイルの再構築

次のリクエストは、「ALPHA フォーマット構造化 HOLD ファイルの作成」 の例で生成された構造化 HOLD ファイルから、元の FOCUS データソースを再構築します。

```
TABLE FILE HOLD
PRINT LAST_NAME FIRST_NAME JOBCODE ED_HRS
BY DEPARTMENT
BY HIGHEST SALARY
ON TABLE SET EXTRACT ON
ON TABLE HOLD AS RECONST FORMAT FOCUS
END
```

このリクエストは、次のマスターファイルを作成します。

```
FILENAME=RECONST    , SUFFIX=FOC     , $
  SEGMENT=EMPINFO, SEGTYPE=S0, $
    FIELDNAME=LAST_NAME, ALIAS='LN', USAGE=A15, $
    FIELDNAME=FIRST_NAME, ALIAS='FN', USAGE=A10, $
    FIELDNAME=DEPARTMENT, ALIAS='DPT', USAGE=A10,
    FIELDNAME=ED_HRS, ALIAS='OJT', USAGE=F6.2, $
  SEGMENT=PAYINFO, SEGTYPE=S0, PARENT=EMPINFO, $
    FIELDNAME=SALARY, ALIAS='SAL', USAGE=D12.2M, $
    FIELDNAME=JOBCODE, ALIAS='JBC', USAGE=A3, $
```

次のリクエストは、レポート出力を表示します。

```
TABLE FILE RECONST
PRINT LAST_NAME FIRST_NAME JOBCODE ED_HRS
BY DEPARTMENT
BY HIGHEST SALARY
END
```

出力結果は次のとおりです。

```
DEPARTMENT           SALARY  LAST_NAME       FIRST_NAME  JOBCODE  ED_HRS
----------           ------  ---------       ----------  -------  ------
MIS              $27,062.00  CROSS           BARBARA     A17       45.00
                 $25,775.00  CROSS           BARBARA     A16       45.00
                 $21,780.00  BLACKWOOD       ROSEMARIE   B04       75.00
                 $18,480.00  JONES           DIANE       B03       50.00
                             MCCOY           JOHN        B02         .00
                 $17,750.00  JONES           DIANE       B02       50.00
                 $13,200.00  SMITH           MARY        B14       36.00
                  $9,000.00  GREENSPAN       MARY        A07       25.00
                  $8,650.00  GREENSPAN       MARY        B01       25.00
PRODUCTION       $29,700.00  BANNING         JOHN        A17         .00
                 $26,862.00  IRVING          JOAN        A15       30.00
                 $24,420.00  IRVING          JOAN        A14       30.00
                 $21,120.00  ROMANS          ANTHONY     B04        5.00
                 $16,100.00  MCKNIGHT        ROGER       B02       50.00
                 $15,000.00  MCKNIGHT        ROGER       B02       50.00
                 $11,000.00  STEVENS         ALFRED      A07       25.00
                 $10,000.00  STEVENS         ALFRED      A07       25.00
                  $9,500.00  SMITH           RICHARD     A01       10.00
                  $9,050.00  SMITH           RICHARD     B01       10.00
```

構造化 HOLD ファイルのフィールドの除外

次のリクエストは、ソートに使用する SALARY フィールドを除外します。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME JOBCODE ED_HRS
BY DEPARTMENT
BY HIGHEST SALARY
ON TABLE SET EXTRACT *
EXCLUDE=(SALARY),$
ENDEXTRACT
ON TABLE HOLD FORMAT FOCUS
END
```

このリクエストは、次の HOLD マスターファイルを作成します。

```
FILENAME=HOLD   , SUFFIX=FOC     , $
  SEGMENT=EMPINFO, SEGTYPE=S0, $
    FIELDNAME=LAST_NAME, ALIAS='LN', USAGE=A15, $
    FIELDNAME=FIRST_NAME, ALIAS='FN', USAGE=A10, $
```

```
    FIELDNAME=DEPARTMENT, ALIAS='DPT', USAGE=A10, $
    FIELDNAME=ED_HRS, ALIAS='OJT', USAGE=F6.2, $
SEGMENT=PAYINFO, SEGTYPE=S0, PARENT=EMPINFO, $
    FIELDNAME=JOBCODE, ALIAS='JBC', USAGE=A3, $
```

構造化 HOLD ファイルに格納される要素

構造化 HOLD ファイルには、TABLE または TABLEF リクエストを完了するために必要なすべての元のセグメントインスタンスが格納されます。元のリクエストで使用された表示コマンド (PRINT、LIST、SUM、COUNT) とは無関係に、構造化 HOLD ファイルは PRINT コマンドが使用されたときのように作成されます。

抽出ファイルには次の要素が格納されます。

- 表示オブジェクト、ソートフィールド、選択条件 (WHERE/IF テスト) に使用するフィールドなど、リクエストで指定されるすべての実フィールド。

  リクエストで複数回参照されるフィールドは、HOLD ファイルに 1 回だけ追加されます。

- FILTER FILE 条件に使用されるフィールド。
- ALL 以外の演算接頭語は無視されます。これらは、収集されるデータ量には影響しますが、演算を意味するものではありません。
- フィールドベースでフォーマットを再適用する (FIELD1/FIELD2=) と、元のフィールドおよびフォーマットフィールドが格納されます。
- 明示的、またはリクエストですべてのメンバーが参照されている GROUP フィールド。

  注意：特定のグループメンバーが EXCLUDE で除外されている場合、または参照されていない場合、そのメンバーが属する GROUP は抽出マスターファイルに追加されません。これは、ネストされているグループおよび重複するグループにも適用されます。GROUP およびその要素がすべてリクエストで指定される場合、この GROUP は抽出 HOLD ファイルの実フィールドに追加されません。

- FIELDS=ALL に設定すると、リクエストで使用するすべての一時項目 (DEFINE) は、構造化 HOLD ファイルの実フィールドになり、DEFINE 式のその他のフィールドとともに追加されます (その他の 一時項目 (DEFINE) を含む)。EXCLUDE を使用し、EXTRACT 出力に追加するフィールド数を減らします。
- FIELDS=EXPLICIT に設定すると、表示オブジェクトおよびソートフィールドが格納さ

れます。リクエストで参照される場合、一時項目 (DEFINE) は実フィールドになりますが、それらを作成するために使用したフィールドは、明示的に参照されない限り格納されません。これにより、リクエストで返されるフィールド数が減少します。

構造化 HOLD ファイルに格納されない要素

- WHERE フィールドの演算接頭語は、データ選択時に評価されますが、抽出時の出力には格納されません。
- ALL を除き、表示オブジェクトの演算接頭語は無視されます。
- MATCH、MORE、GRAPH リクエストで構造化 HOLD 構文を使用すると、エラーメッセージが表示され、プロシジャは終了します。
- WHERE/IF TOTAL テストは、構造化 HOLD ファイルリクエストではサポートされず、実行するとリクエストが停止します。
- 実フィールドへのフォーマットの再適用は無視され、実フィールドのみが格納されます。
- 一時項目 (COMPUTE) は含められませんが、COMPUTE 式に使用するフィールドは抽出ファイルに格納されます。

構造上および動作上の注意

- 構造化 HOLD ファイルリクエストには、TABLE リクエストと同様にフィールドの数およびサイズの制限が適用されます。

構造上の注意

- 次の SET パラメータは、構造化 HOLD ファイルリクエストではオフに設定または無視されます。
  - AUTOINDEX
  - AUTOPATH
  - AUTOSTRATEGY
  - EXTHOLD

    - EXTSORT
    - HOLDATTR
    - 出力フォーマットを制御する SET および ON TABLE SET コマンドは、すべて抽出ファイルの作成時に無視されます。
- 代替ビューは保持され、抽出ファイルの構造に反映されます。
- リクエストのインデックスビューは保持され、出力ファイルの構造に反映されます。
- 親セグメントが明示的に除外されることにより、リクエストが 2 つの孤立セグメントを含むファイルを作成する場合、2 つの無関係のセグメントの親として機能するダミーシステムセグメントが作成されます。このセグメントのデータインスタンスは 1 つのみです。両方の孤立セグメントは、このシステムセグメントを親として参照します。リクエストで指定されないため親が存在しない場合、システムセグメントはリクエストの実行中に有効になり、そのセグメントの親として含められます。
- 親セグメントのない 2 つのユニーク (U) セグメントが含まれる場合、これらのユニークセグメントはシステムセグメントを親として参照する SEGTYPE S0 のセグメントに変換されます。
- JOIN および JOIN WHERE 構造はサポートされます。

SQL 最適化の注意

- 集計時の SQL 最適化は、EXTRACT リクエストではオフにする必要があります。

BY/ACROSS/FOR の注意

- BY および ACROSS フィールドは、追加の表示オブジェクトになります。
- BY. . .ROWS および ACROSS . . .COLUMNS 関数は、追加するフィールド値を制限する暗黙的な WHERE としてのみ機能します。
- FOR フィールドは含められます。
- RECAP フィールドは除外されます(COMPUTE と同様)。
- 以前に特定されたフィールドを参照する集計フィールドは、構造化 HOLD ファイルの作成時に無視されます。この種の集計フィールドには、SUMMARIZE、RECOMPUTE、SUBTOTAL、SUB-TOTAL、ACROSS-TOTAL、ROW-TOTAL、COLUMN-TOTAL があります。

フォーマットの注意

- 構造化 HOLD ファイル処理は、IN、OVER、NOPRINT、SUP-PRINT、FOLD-LINE、SKIP-LINE、UNDER-LINE、PAGE-BREAK、TABPAGENO、AS、およびフィールドタイ

トルの配置などのすべてのフォーマット要素を無視します。ただし、HEADING、FOOTING、SUBHEAD、SUBFOOT などのフォーマットコマンドで参照されるフィールド、およびこれらに関連付けられたすべての WHEN 式は含められます。

- STYLE および STYLESHEET コマンドは、抽出出力生成時にすべて無視されます。
- AnV および AnW フィールドはサポートされます。TX フィールドは FOCUS ファイルのみにエクスポートされます。
- 実フィールドの FIELDNAME および ALIAS が同一で、このフィールドが (別フォーマットの) それ自体として再定義 される場合 、HOLD マスターファイルにフィールド名とエイリアスが同一の 2 つのフィールドが作成されます。この場合、名前で参照されると、2 つ目のフィールドがアクセスされることはありません。2 つ目のフィールドのみを含めるには、FIELDS=EXPLICIT を使用します。次の DEFINE は、重複するフィールド名およびエイリアスの生成の例を示しています。

```
DEFINE FILE CAR
COUNTRY/A25=COUNTRY;
END
```

DBA の注意

- RESTRICT=NOPRINT 以外の 構造化 HOLD ファイルリクエストを実行する際、ソースファイルの DBA 制御は保持されます。この場合、リクエスト内のフィールド名は表示されません。これらのフィールドはエクスポートすることができないため、EXCLUDED を使用して明示的にエクスポートする必要があります。
- DBA 制限は HOLD マスターファイルには継承されません。

抽出ファイルの再構築

- 構造化 HOLD ファイルから FOCUS または Flat File を再構築するには、構造化 HOLD ファイルの生成に使用した構文と同一のものを使用します。複数パス構造を保持するためには、ON TABLE SET EXTRACT 構文を使用する必要があります。
- 再構築された FOCUS セグメントは、KEY および INDEX 情報が保持されないため、すべて SEGTYPE=S0 になります。REBUILD INDEX を使用して INDEX を再挿入することができます。

表示フォーマットの選択

レポートを画面に表示する際、数種類の表示フォーマットから選択することができます。最適な表示フォーマットは用途ごとに異なります。レポートの代表的な表示フォーマットには次のものがあります。

- HTML ページ - Web ブラウザでの表示に最適です。
- PDF ドキュメント - 画面上に表示したレポートを同一の外観で印刷する場合に適しています。
- DHTML ファイル - PDF などの印刷用の出力フォーマットで使用可能なほとんどの機能を備えた HTML 出力です。
- Excel 2007 ワークシート - Excel 2007 以降でデータを操作できます。
- Excel 2000 ワークシート - Excel 2000 または 2003 でデータを操作できます。

レポートを画面ではなくファイルに出力することもできます。使用可能なフォーマットについては、レポート出力の保存と再利用を参照してください。

レポート表示フォーマット

レポートを表示する際に、複数の表示フォーマットの中からいずれかを選択することができます。

- **Web フォーマット**　HTML のフォーマットです。
- **印刷フォーマット**　PDF のフォーマットです。
- **ワードプロセッサフォーマット**　WP および DOC のフォーマットです。
- **ワークシートフォーマット**　Excel 2007/2010 XML ベースのフォーマット、PivotTable や FORMULA を含む Excel 2000/2003 HTML ベースのフォーマット、Excel 97 HTML ベースのフォーマット、Excel バイナリのフォーマットです。

DHTML および HTML に関する注意：DHTML は、絶対位置での配置が可能な HTML です。HTML フォーマットは、出力を表ベースのフォーマットで生成しますが、正確な位置の調整はレポートの表示先となるブラウザに依存するよう設計されています。一方、DHTML フォーマットは、PDF の場合と同様、ユーザが定義した位置に表示するよう設計されています。つまり、各要素がレポートプロシジャで定義されたとおりにページ上の正確な位置に配

置されます。PDF、DHTML、PPT、PPTX は、位置ベースのフォーマットです。HTML および EXL2K は、表ベースまたはセルベースのフォーマットです。そのため、DHTML 出力は、HTML のような外観ではなく、PDF に似た外観になります。

レポートデータの表示ではなく、保存と再利用に使用できるフォーマットについての詳細は、レポート出力の保存と再利用を参照してください。

注意：スタイル出力フォーマットでは、LINES パラメータを 999 以上に設定すると連続用紙が生成されます。連続用紙が指定され、出力フォーマットのページに物理サイズが使用される場合 (例、PDF 出力)、列タイトルが物理ページの上部に繰り返し表示されます。この場合、ページ番号は表示されません。

PCHOLD による表示フォーマットの選択

リクエストで ON TABLE PCHOLD コマンドを使用し、画面にレポートを表示することができます。

```
ON TABLE PCHOLD FORMAT formatname
```

説明

formatname

次のいずれかの値です。

DOC	レポートを Web ブラウザ内の Microsoft Word で、改ページ付きテキストフォーマットのワードプロセッサドキュメントとして表示します。
EXCEL	レポートを Excel ワークシートとして表示します。
XLSX	レポートを Excel 2007/2010 ワークシートとして表示します。
EXL2K	レポートを Excel 2000/2003 ワークシートとして表示します。

EXL2K FORMULA	レポートを Excel 2000/2003 ワークシートとして表示します。WebFOCUS の合計をはじめとする一時項目 (COMPUTE) は、Excel の数式に変換されます。EXL2K FORMULA は処理に時間を要するため、レポートに数式が含まれない場合は、EXL2K フォーマットの使用をお勧めします。
EXL2K PIVOT	レポートを Excel 2000/2003 PivotTable として表示します。
EXL97	レポートを Excel 97 ワークシートとして表示します。
HTML	レポートを HTML ページとして表示します。
PDF	レポートを PDF (Portable Document Format) ドキュメントとして表示します。
WP	レポートを Web ブラウザで、テキストフォーマットのワードプロセッサドキュメントとして表示します。

SET ONLINE-FMT による表示フォーマットの選択

表示フォーマットの種類によっては、レポートを画面に表示するために SET コマンドの ONLINE-FMT パラメータを使用することができます。

リクエスト外部で次の構文を使用し、プロシジャ内のリクエストすべてに対するフォーマットを指定します。

```
SET ONLINE-FMT = formatname
```

リクエスト内部で次の構文を使用し、そのリクエストのみに対するフォーマットを指定します。

```
ON TABLE SET ONLINE-FMT formatname
```

説明

formatname

次のいずれかの値です。

HTML (デフォルト)	レポートを HTML ページとして表示します。
PDF	レポートを PDF (Portable Document Format) ドキュメントとして表示します。
XLSX	レポートを Excel 2007/2010 ワークシートとして表示します。
EXL2K	レポートを Excel 2000/2003 ワークシートとして表示します。
EXL97	レポートを Excel 97 ワークシートとして表示します。
STANDARD	レポートを、文字と行を基準とする等幅フォントのレガシーフォーマットで表示します。

ヒント：WebFOCUS の今後のリリースでは ONLINE-FMT 構文は PCHOLD 構文で代替される予定です。現在は、相互に置き換えて使用することができます。

WebFOCUS レポートの MIME タイプの指定

Web ブラウザでの表示用である HTML フォーマットのほかに、ブラウザに送信してデスクトップアプリケーションやヘルパーアプリケーションで開くことのできるレポートを作成することができます。ブラウザによるデスクトップアプリケーションの認識および呼び出しを正確に実行するためには、レポートの MIME (Multipurpose Internet Mail Extension) タイプを特定のアプリケーションに関連付ける必要があります。

詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

レポート出力での先頭および内部ブランクの保持

HTML ブラウザ出力および Excel 出力では、デフォルト設定で先頭および末尾のブランクはテキストから削除され、内部の複数のブランクは 1 つに統合されます。

HTML および EXL2K レポート出力の先頭および内部のブランクを保持する場合、SET SHOWBLANKS=ON を発行することができます。

このコマンドを発行する場合でも、配置がデフォルト設定の見出しまたは脚注を使用する見出し、中間見出し、脚注、中間脚注行を除き、末尾のブランクは保持されません。

HTML および EXL2K レポートの先頭および内部のブランクの保持

プロシジャまたはプロファイルでは、次の構文を使用します。

```
SET SHOWBLANKS = {OFF|ON}
```

リクエストでは、次の構文を使用します。

```
ON TABLE SET SHOWBLANKS {OFF|ON}
```

説明

OFF

HTML および EXL2K レポート出力の先頭のブランクを削除し、内部のブランクを 1 つにします。

ON

HTML および EXL2K レポート出力の先頭および内部のブランクを保持します。配置がデフォルト設定の見出しまたは脚注を使用する見出し、中間見出し、脚注、中間脚注行のブランクも保持されます。

HTML および EXL2K レポート出力の先頭および内部ブランクの保持

次のリクエストは、ACTION の値の先頭、および CATEGORY フィールドの TRAIN/EX と SCI/FI の値の先頭と内部にブランクを追加して一時項目 (DEFINE) を作成します。また、COMEDY の値に末尾のブランクも追加します。

```
SET SHOWBLANKS = OFF
DEFINE FILE MOVIES
NEWCAT/A30 = IF CATEGORY EQ 'ACTION' THEN '  ACTION'
        ELSE IF CATEGORY EQ 'SCI/FI' THEN 'SCIENCE   FICTION'
        ELSE IF CATEGORY EQ 'TRAIN/EX' THEN '   TRAINING    EXERCISE'
        ELSE IF CATEGORY EQ 'COMEDY' THEN 'COMEDY     '
        ELSE                'GENERAL';
END
TABLE FILE MOVIES
SUM CATEGORY LISTPR/D12.2 COPIES
BY NEWCAT

ON TABLE SET STYLE *
GRID=OFF,$
TYPE=REPORT, FONT=COURIER NEW,$
ENDSTYLE
END
```

SHOWBLANKS OFF では、これらのブランクは削除されます。

NEWCAT	CATEGORY	LISTPR	COPIES
TRANING EXERCISE	TRAIN/EX	119.87	10
ACTION	ACTION	94.82	14
COMEDY	COMEDY	154.80	19
GENERAL	MYSTERY	1,216.82	67
SCIENCE FICTION	SCI/FI	114.84	7

SHOWBLANKS ON では、先頭と内部に追加したブランクが保持されます。末尾のブランクは保持されません。

NEWCAT		CATEGORY	LISTPR	COPIES
TRANING	EXERCISE	TRAIN/EX	119.87	10
ACTION		ACTION	94.82	14
COMEDY		COMEDY	154.80	19
GENERAL		MYSTERY	1,216.82	67
SCIENCE	FICTION	SCI/FI	114.84	7

Web 表示フォーマットの使用 - HTML

レポートを HTML ページとして表示することができます。HTML フォーマットではスタイルシートオプションのほとんどがサポートされるため (特にカスケードスタイルシートとともに使用する場合)、完全なレポートフォーマットの設定が可能になります。

デフォルト設定では、先頭と内部の複数のブランクは、レポート出力で 1 つになります。

詳細は、レポートフォーマット設定の管理を参照してください。

WebFOCUS をインストールした直後のデフォルト表示フォーマットは HTML です。HTML レポートは Web ブラウザで開きます。

デフォルト設定に関わらず、レポートの実行時に HTML ページとして表示するよう指定することができます。使用可能なコマンドは次のとおりです。

- **PCHOLD コマンド**
- **SET コマンドの ONLINE-FMT パラメータ**

HTML 表示フォーマットでは SET コマンドのスタイルシートパラメータを OFF 以外の値に設定する必要があります。有効な値には、ON (デフォルト)、スタイルシートファイル名、インラインスタイルシート(*) があります。

HTML でサポートされるレポートとフォーマットのオプションについての詳細は、『ibi™ WebFOCUS® Language リファレンス』を参照してください。

HTML レポートの表示は、JavaScript または VBScript の JSURL SET パラメータによりカスタマイズすることもできます。詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

HTML レポート表示のカスタマイズ

次の例では、製品に付属の JavaScript 関数に加え、独自の JavaScript 関数を呼び出すことにより、HTML レポートの表示をカスタマイズします。ここでは JSURL SET パラメータを使

用します。

ここで説明する JavaScript 関数は、レポートの実行時に右クリックメニューを無効にします。

1. js ファイルを作成し、Web サーバからアクセスできる場所に保存します。

   たとえば、次の例は HTML レポートで右クリックメニューを無効にします。

   ```
   function setnocontextclick () {
      if (document.body != null) {
         document.body.oncontextmenu=new Function("return false");
      }
      else
         window.setTimeout("setnocontextclick()",100);
   }
   function killmenuOnLoadFunc(arrayofonloads,currentindex) {
        setnocontextclick();
   }
   ```

   このファイルを ibi_apps/ibi_html ディレクトリに「killmenu.js」という名前で保存します。

   注意：OnLoad 関数名のフォーマットは次のとおりです。

   ```
   customfunctionnameOnLoadFunc
   ```

   説明

   customfunctionname

   関数コードを含む JavaScript ファイル名です。

2. TABLE リクエストに JSURL パラメータを追加します。このコマンドを edasprof.prf ファイルに追加しておくと、サーバでの各 HTML レポートの実行時にこの js ファイルを実行することができます。

3. レポートを実行します。

   ```
   SET JSURL=/ibi_apps/ibi_html/killmenu.js
   TABLE FILE CENTORD
   SUM QUANTITY
   BY PLANTLNG
   END
   ```

   このレポート出力では右クリックメニューオプションは使用できません。

WebFOCUS デフォルト JavaScript 関数の無効化

JSURL SET パラメータを使用して、デフォルトの WebFOCUS JavaScript 関数を無効化または変更することができます。次の例では、すべての WebFOCUS デフォルト関数を警告ボックスに表示し、無効にする方法について説明します。

1. js ファイルを作成し、Web サーバからアクセスできる場所に保存します。

   このファイルを ibi_apps/ibi_html ディレクトリに「disable.js」という名前で保存します。arrayofonloads 配列は str1 と str2 の 2 つの文字列パラメータで構成されます。str1 はロード時に呼び出す関数名です。str2 は str1 で記述される動作を実行するかどうかを示すブール値 (TRUE/FALSE) です。currentindex パラメータはページを表示する際の関数のロード順序を定義する連続番号です。

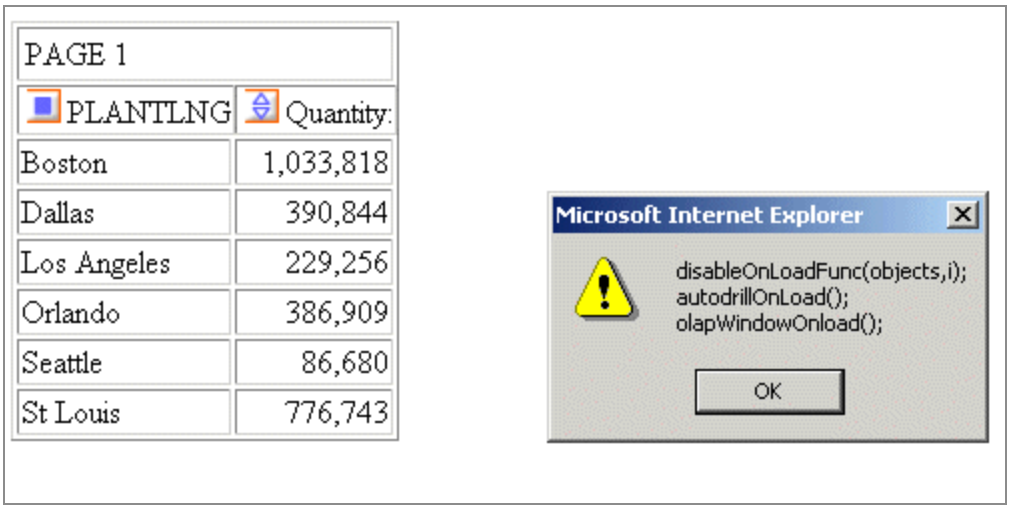
   ```
   function disableOnLoadFunc(arrayofonloads,currentindex) {
     buffer ="";
      for (var index=0;index<arrayofonloads.length;index++) {
         buffer += arrayofonloads[index].str1+"¥n" ;
         arrayofonloads[index].str2=false;
      }
      alert(buffer);
   }
   ```

2. TABLE リクエストに JSURL パラメータを追加します。

3. レポートを実行します。

   ```
   -OLAP ON
   SET AUTODRILL = ON
   SET JSURL=/ibi_apps/ibi_html/disable.js
   TABLE FILE CENTORD
   SUM QUANTITY
   BY PLANTLNG
   END
   ```

   出力結果は次のとおりです。



HTML レポート出力の使用上の注意

- 境界線がオンの場合、HTML フォーマットのデフォルト動作では、フィールドタイトルが下線なしで表示されます。境界線がオンの場合にフィールドタイトルを下線付きで表示するには、GRID OFF に設定します。
- AUTOFIT パラメータを使用すると、HTML レポート出力がコンテナ内 (ウィンドウまたはフレーム) に収まるようサイズが自動的に変更されます。複数のレポートを出力するプロシジャでは、レポート出力プロシジャのいずれかで AUTOFIT が ON に設定されている場合、その設定がページ上のレポート出力のすべてに適用されます。

  アコーディオン、Web ビューア、HTML TOC、HFREEZE インタラクティブレポートの機能を使用する場合、AUTOFIT はサポートされます。

  AUTOPATH パラメータについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

絶対配置による HTML レポートの作成

DHTML フォーマットは、PDF などの印刷用の出力フォーマットに関連したほとんどの機能を HTML 出力に追加します。HTML ファイル (.htm) または Web アーカイブファイル (.mht) を作成することができます。作成される出力ファイルのタイプは、HTMLARCHIVE パラメータにより異なります。

DHTML フォーマット でサポートされる機能には、次のものがあります。

- **絶対配置**　テキストおよびイメージを HTML レポート内に正確に配置します。HTML のレイアウトには、PDF 出力と同一のスタイルシート構文を使用することができます。
- **Web ビューア**　Web ビューアは SET HTMLARCHIVE=OFF で利用することができます。
- **PDF スタイルシート機能**　たとえば、罫線、背景色、OVER の機能がサポートされます。

Web アーカイブレポート出力の作成

```
SET HTMLARCHIVE = {ON|OFF}
```

説明

ON

Web アーカイブレポート出力を作成します。出力ファイルのファイルタイプは MHT です。

OFF

HTML フォーマットで出力を作成します。出力ファイルのファイルタイプは HTM です。デフォルト値は OFF です。

DHTML レポートの作成

```
[ON TABLE] HOLD [AS name] FORMAT DHTML
```

説明

name

出力ファイル名を指定します。SET HTMLARCHIVE が OFF の場合、拡張子は HTML です。SET HTMLARCHIVE が ON の場合、拡張子は MHT です。

DHTML フォーマットの使用上の注意

- DHTML レポートのフォントマップファイルは、ERRORS データセットの DHTML メンバーです。
- レガシー複合レポートはサポートされません。

DHTML レポートの作成

次の例では、絶対配置のイメージを含む DHTML ファイルを作成します。

```
SET HTMLARCHIVE = OFF
TABLE FILE GGSALES
SUM UNITS BY CATEGORY BY PRODUCT
ON TABLE SUBHEAD
"Report on Units Sold"
" "
" "
" "
" "
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE HOLD FORMAT DHTML
ON TABLE SET STYLE *
TYPE=TABHEADING,IMAGE=c:¥images¥GOTHAM.GIF, POSITION=(.25 .25),
 SIZE=(.5 .5), $
ENDSTYLE
END
```

出力結果の外観および配置は、PDF レポートと同一です。

Report on Units Sold

Category	Product	Unit Sales
Coffee	Capuccino	189217
	Espresso	308986
	Latte	878063
Food	Biscotti	421377
	Croissant	630054
	Scone	333414
Gifts	Coffee Grinder	186534
	Coffee Pot	190695
	Mug	360570
	Thermos	190081

印刷表示フォーマットの使用 - PDF

PDF (Portable Document Format) は Web を介した電子ドキュメントの配信および共有において最も使用頻度の高いフォーマットです。このフォーマットは、ブラウザやプリンタの種類に左右されずにレポートの外観およびレイアウトを保持したい場合に役立ちます。

レポートの実行時にレポート表示を PDF ドキュメントとして指定することができます。使用可能なコマンドは次のとおりです。

- **PCHOLD コマンド**
- **SET コマンドの ONLINE-FMT パラメータ**

1 つの PDF ファイルに複数のスタイルレポートを組み合わせることができます。詳細は、レポートページのレイアウトを参照してください。

複合レポートを含む PDF レポートは、ReportCaster で配信することができます。詳細は、『ibi™ WebFOCUS® ReportCaster 利用ガイド』を参照してください。

PDF 表示フォーマットの使用

レポートを PDF ファイルとして表示することができます。PDF (Portable Document Format) ではスタイルシート属性のほとんどがサポートされているため、完全なレポート

フォーマットの設定が可能です。PDF でサポートされるさまざまなスタイルシート機能については、このマニュアルで説明しています。

PDF は、作成や表示に使用するアプリケーションソフトウェア、ハードウェア、オペレーティングシステムに依存することなく、表示と実行が常に行えます。

レポートは Web ブラウザの Adobe Acrobat または Adobe Reader で開くことができます。PDF レポートを表示するには、コンピュータに Adobe Reader がインストールされている必要があります。Adobe Reader の無料ダウンロードについては、http://www.adobe.com を参照してください。

制限：PDF フォーマットではドキュメントのページ、ハイパーリンク、およびイメージの数に制限があります。WebFOCUS レポートを PDF フォーマットで作成する際の制限事項についての詳細は、レポート出力の保存と再利用を参照してください。

PDF 出力ファイルの圧縮

ファイルの圧縮を使用することで、PDF 出力ファイルの物理サイズを縮小することができます。この PDF に固有の機能を使用すると、サイズの小さい PDF ファイルを生成できるため、ファイル内のレポートのフォーマット設定や内容に影響を与えることなく、より簡単にファイルの格納、配信が可能になります。

```
SET FILECOMPRESS = {ON|OFF}
```

説明

ON

PDF 出力ファイルを圧縮します。

OFF

PDF 出力ファイルを圧縮しません。デフォルト値は OFF です。

このコマンドは、PDF 出力にのみ適用されます。HTML や Excel などの他の出力フォーマットでは無視されます。

PDF 出力での透かしの表示

透かしは、ドキュメントの最下位レイヤに配置され、透明なレイヤコンテンツを通して表示されるイメージまたはテキストです。

WebFOCUS の背景色では、透明度はサポートされません。そのため、ページ上で標準のイメージを背景色の背後に配置した場合、イメージが不鮮明になる場合があります。WebFOCUS では、PDF レポートでこの問題を解決するために、標準のプリンタで使用される手法と同様に、不透明イメージをドキュメントレイヤの最上面に配置します。この手法により、ドキュメントの各レイヤが、透明な透かしイメージの背後に隠れなくなります。

透かしイメージは、レポートの作成者が準備します。透明イメージを作成する際は、イメージを GIF フォーマットで作成し、透明の背景色を使用する必要があります。

背景色を使用した PDF レポートでのイメージ挿入

透かしは、複合レポートおよび単一 TABLE リクエストの PDF 出力でサポートされます。各ドキュメントでサポートされるアクティブ透かしイメージは 1 つです。このイメージを透かしイメージとして指定するには、Z-INDEX 属性でイメージの配置順序を定義します。

Z-INDEX 値が指定された最初のイメージが、現在のドキュメントでのアクティブ透かしイメージと見なされます。Z-INDEX または OPACITY のスタイルシート属性で定義された残りのイメージはすべて、標準の WebFOCUS イメージとして表示されます。

透かしイメージを指定するには、スタイルシートで透明の GIF イメージに対して次の属性を定義します。

Z-INDEX=TOP	イメージを透かしイメージとして処理し、そのイメージを常にページ上の他のすべてのオブジェクトの最上面に配置します。今後のリリースでは、この値は最上位レイヤとして使用され、他のレイヤとともにサポートされます。
OPACITY=*n*	ここで、n はイメージに適用する OPACITY のパーセント (%) を表します。OPACITY の値を大きくすると、イメージの透明度が下がります。イメージの背後にあるレポートは、より不鮮明になります。n の値は、0 (ゼロ) から 100 までの任意の数値にすることができます。値を指定しない場合、デフォルト値の 100% が使用され、完全に不透明なイメージが表示されます。

単一 TABLE リクエストでは、次のように指定します。

```
TYPE=<REPORT|HEADING>, OBJECT=IMAGE, IMAGE=<image.gif>,
Z-INDEX=TOP, OPACITY=15, POSITION=(.25 .25), DIMENSION=(8 10.5),$
```

COMPOUND 構文では、次のように指定します。

ページマスターで指定

```
PAGELAYOUT=ALL, NAME='Page Master', $
OBJECT=IMAGE, IMAGE= internalonlyport.GIF, Z-INDEX=TOP, OPACITY=15,
POSITION=(.25 .25), DIMENSION=(8 10.5),$
```

ページレイアウトで指定

```
PAGELAYOUT=1, NAME='Page Layout1', $
OBJECT=IMAGE, IMAGE= internalonlyport.GIF, Z-INDEX=TOP, OPACITY=15,
POSITION=(.25 .25), DIMENSION=(8 10.5),$
```

PDF レポートへの透明イメージの挿入

次のリクエストは、GGSALES データソースに対して実行され、ページ上にコーヒーイメージ (coffee.gif) を配置し、透かしイメージ (internalonlyport.gif) をレイヤの最上面に挿入します。これらのイメージは、レポートの各ページに表示されます。

```
TABLE FILE GGSALES
SUM
     GGSALES.SALES01.DOLLARS/D12CM
     GGSALES.SALES01.UNITS/D12C
     GGSALES.SALES01.BUDDOLLARS/D12CM
     GGSALES.SALES01.BUDUNITS/D12C
BY GGSALES.SALES01.REGION
BY GGSALES.SALES01.CATEGORY
BY GGSALES.SALES01.PRODUCT
HEADING
"Gotham Grinds"
"Product Sales By Region"
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
   INCLUDE = endeflt,
```

```
   TOPMARGIN=.5,
   BOTTOMMARGIN=.5,
   LEFTMARGIN=1,
   RIGHTMARGIN=1,
$
TYPE=REPORT,
    OBJECT=IMAGE,
    IMAGE=internalonlyport.gif,
    POSITION=(+0.70000 +0.70000),
    SIZE=(7 7.5),
    Z-INDEX=TOP, OPACITY=15,
$
TYPE=REPORT,
    OBJECT=IMAGE,
    IMAGE=coffee.gif,
    POSITION=(+1.0 +0.5),
    SIZE=(.5 .5),
$ENDSTYLE
END
```

出力結果は次のとおりです。

Gotham Grinds
Product Sales By Region

Region	Category	Product	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
Midwest	Coffee	Espresso	$1,294,947	101,154	$1,258,232	101,869
		Latte	$2,883,566	231,623	$2,827,800	233,657
	Food	Biscotti	$1,091,727	86,105	$1,067,629	85,839
		Croissant	$1,751,124	139,182	$1,708,733	139,648
		Scone	$1,495,420	116,127	$1,444,359	113,776
	Gifts	Coffee Grinder	$619,154	50,393	$613,453	50,628
		Coffee Pot	$599,878	47,156	$614,007	47,779
		Mug	$1,086,943	86,718	$1,096,150	87,092
		Thermos	$577,906	46,587	$564,010	46,819
Northeast	Coffee	Capuccino	$542,095	44,785	$561,491	44,432
		Espresso	$850,107	68,127	$872,902	69,776
		Latte	$2,771,815	222,866	$2,818,069	221,712
	Food	Biscotti	$1,802,005	145,242	$1,848,682	145,152
		Croissant	$1,670,818	137,394	$1,739,522	137,864
		Scone	$907,171	70,732	$865,703	68,415
	Gifts	Coffee Grinder	$509,200	40,977	$511,642	41,297
		Coffee Pot	$590,780	46,185	$573,349	45,404
		Mug	$1,144,211	91,497	$1,170,314	90,540
		Thermos	$604,098	48,870	$615,247	49,767
Southeast	Coffee	Capuccino	$944,000	73,264	$956,661	75,353
		Espresso	$853,572	68,030	$849,465	66,785
		Latte	$2,617,836	209,654	$2,625,303	213,555
	Food	Biscotti	$1,505,717	119,594	$1,512,019	120,549
		Croissant	$1,902,359	156,456	$1,969,906	157,148
		Scone	$900,655	73,779	$927,363	73,812
	Gifts	Coffee Grinder	$605,777	47,083	$569,585	46,784
		Coffee Pot	$645,303	49,922	$654,579	50,637
		Mug	$1,102,703	88,474	$1,124,345	89,371
		Thermos	$632,457	48,976	$618,745	48,253
West	Coffee	Capuccino	$895,495	71,168	$877,304	70,585
		Espresso	$907,617	71,675	$923,941	72,927
		Latte	$2,670,405	213,920	$2,722,718	215,272
	Food	Biscotti	$863,868	70,436	$861,804	67,780
		Croissant	$2,425,601	197,022	$2,406,554	195,329
		Scone	$912,868	72,776	$914,886	72,252
	Gifts	Coffee Grinder	$603,436	48,081	$571,316	47,397
		Coffee Pot	$613,624	47,432	$630,196	49,208
		Mug	$1,188,664	93,881	$1,156,976	93,629
		Thermos	$571,368	45,648	$575,818	46,402

Confidential - For internal use only

サポートされる機能

透かしを使用する場合、次の主要 PDF 機能がサポートされます。

- 標準の TABLE リクエスト (マルチパネル表示のレポートを含む)
- 複合レポート (MERGE=OFF)
- 統合された複合レポート (MERGE=ON)

- 従来の COMPOUND 構文 (OPEN/CLOSE)
- ドリルダウン
- ドリルスルー
- しおり
- 境界線/背景色

制限事項

- OPACITY は、0 (ゼロ) から 100 までの値にする必要があります。
- 単一ドキュメントでサポートされる透かしイメージは 1 つです。

使用上の注意

- 新しい COMPOUND 構文の場合、構文で最初に指定された透かしイメージがレポートに使用されます。構文で指定された他の透かしイメージは無視され、標準の WebFOCUS イメージとして表示されます。
- 従来の COMPOUND 構文の場合、透かしイメージは最初のレポートに配置する必要があります。最初のレポートに配置されていない場合、FOC3362 メッセージが表示されます。
- ブラウザの組み込み PDF ビューアでは、NLS 文字が正しく表示されない場合があります。NLS 文字が正しく表示されない場合、Adobe Reader を使用するようブラウザを構成します。

ページ幅に合わせた PDF レポート出力の拡大縮小

デフォルト設定では、PDF レポート出力の幅が広すぎて単一ページに収まらない場合、収まりきらない列を表示するために同一ページに複数のパネルが生成されます。ページ番号は、ページ番号とパネル番号を指定します。たとえば、ページ番号が 1.1 および 1.2 の場合、ページ 1/パネル 1、ページ 1/パネル 2 をそれぞれ表します。

PAGE-SCALE スタイルシート属性または PAGE-SCALE SET パラメータを使用して、ページ幅に合わせて出力を拡大縮小することができます。

ページ幅に合わせた PDF レポート出力の拡大縮小

スタイルシートでは次の構文を使用します。

```
TYPE=REPORT, PAGE-SCALE={OFF|AUTO}, $
```

プロシジャまたはプロファイルでは、次の構文を使用します。

```
SET PAGE-SCALE = {OFF|AUTO}
```

リクエストでは次の構文を使用します。

```
ON TABLE SET PAGE-SCALE {OFF|AUTO}
```

説明

OFF

PDF レポート出力でのページの拡大縮小表示を無効にします。これがデフォルト値です。

AUTO

PDF レポート出力でのページの拡大縮小表示を有効にします。

PAGE-SCALE 使用上の注意

- PAGE-SCALE は、PDF レポート出力のみでサポートされます。
- ページ上のコンテンツをより多く収めるよう、横方向にページを拡大した場合、生成される縦方向ページの数も少なくなります。

ページ幅に合わせた PDF レポート出力の拡大縮小

次のリクエストは、ページの拡大縮小を使用せずに PDF レポート出力を生成します。

```
SET SQUEEZE=ON
DEFINE FILE WF_RETAIL_LITE
SHOWPIC/A100='C:¥ibi¥WebFOCUS93¥samples¥web_
resource¥signin¥images¥favicon.jpg';
END

TABLE FILE WF_RETAIL_LITE
PRINT  PRODUCT_CATEGORY
COGS_US REVENUE_US MSRP_US DISCOUNT_US GROSS_PROFIT_US QUANTITY_SOLD
BY SHOWPIC NOPRINT
BY CONTINENT_NAME
BY COUNTRY_NAME
WHERE COUNTRY_NAME EQ 'FRANCE' OR 'ITALY'
WHERE RECORDLIMIT=3000;
ON TABLE SUBHEAD
" "
" "
" Report Without PDF Scaling "
" "
" "
ON COUNTRY_NAME SUBHEAD
" "
" "
ON TABLE PCHOLD FORMAT PDF

ON TABLE SET STYLE *
TYPE=DATA, COLUMN=CONTINENT_NAME, FONT=COMIC SANS MS,
  COLOR=BLUE, STYLE=BOLD+ITALIC, $
TYPE=DATA, COLUMN=PRODUCT_CATEGORY, COLOR=FUSCHIA, $
TYPE=HEADING, STYLE=BOLD, COLOR=RGB(0 35 95), SIZE=12, JUSTIFY=CENTER, $
TYPE=SUBHEAD, SIZE=18, STYLE=BOLD, COLOR=RED, $
TYPE=SUBHEAD, IMAGE=(SHOWPIC), SIZE=(.5 .5), $
TYPE=TABHEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER, $
ENDSTYLE
END
```

注意：中間見出しに表示されるイメージは、WebFOCUS に同梱されています。イメージのパスは、使用するプラットフォームおよびインストールオプションによって異なります。この

リクエストのパスには、Windows のデフォルトインストールディレクトリが使用されています。

ページ幅に対して出力の幅が広すぎるため、出力がパネル化されています。ページ 1.1 には、下図のように、ページ幅に収まる列が表示されます。

PAGE 1.1

Report Without PDF Scaling

Customer Continent	Customer Country	Product Category	Cost of Goods	Revenue	MSRP
Europe	France	Camcorder	$82.00	$179.00	179.00
		Camcorder	$60.00	$169.99	169.99
		Video Production	$380.00	$598.00	598.00
		Media Player	$260.00	$379.98	379.98
		Accessories	$80.00	$169.00	169.00
		Computers	$167.00	$280.00	280.00
		Accessories	$500.00	$699.99	699.99
		Video Production	$234.00	$399.00	399.00
		Stereo Systems	$122.00	$199.99	199.99
		Televisions	$730.00	$998.00	998.00
		Stereo Systems	$96.00	$199.98	199.98
		Media Player	$87.00	$179.99	179.99
		Accessories	$160.00	$253.50	338.00
		Computers	$167.00	$280.00	280.00
		Stereo Systems	$145.00	$119.99	199.99
		Camcorder	$398.00	$449.99	599.98
		Camcorder	$87.00	$179.00	179.00
		Accessories	$380.00	$499.95	499.95
		Stereo Systems	$400.00	$559.98	559.98
		Media Player	$420.00	$529.99	529.99
		Camcorder	$205.00	$299.00	299.00
		Media Player	$92.00	$159.98	159.98
		Accessories	$398.00	$599.98	599.98
		Camcorder	$500.00	$524.25	699.00
		Camcorder	$2,870.00	$3,499.00	3,499.00
		Video Production	$190.00	$299.00	299.00
		Media Player	$220.00	$279.99	279.99
		Stereo Systems	$310.00	$499.00	499.00
		Stereo Systems	$312.00	$478.00	478.00
		Stereo Systems	$61.00	$80.99	89.99
		Video Production	$190.00	$179.40	299.00
		Stereo Systems	$122.00	$199.99	199.99
		Stereo Systems	$121.00	$219.99	219.99

ページ 1.2 には、下図のように、残りの列が表示されます。

PAGE 1.2

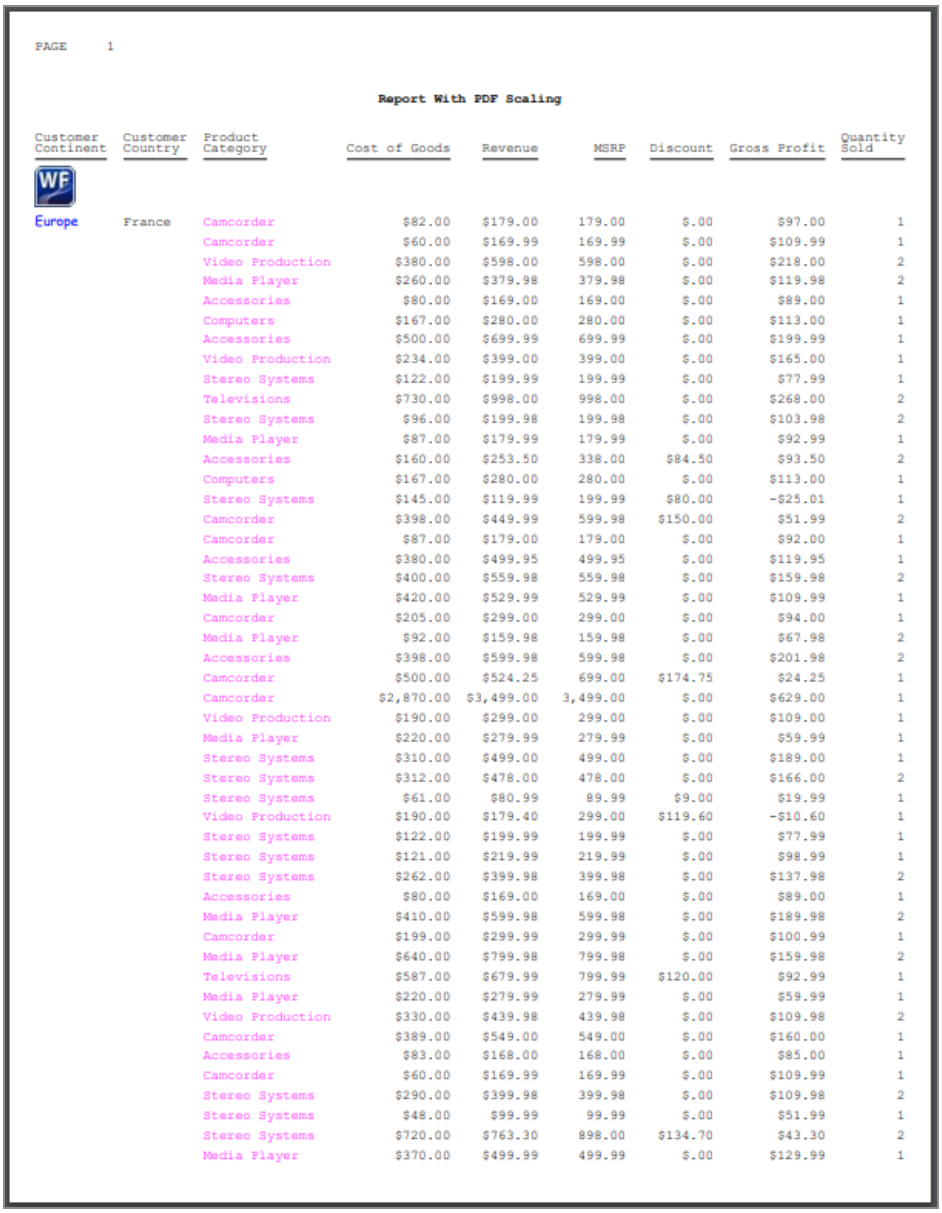
Discount	Gross Profit	Quantity Sold
$.00	$97.00	1
$.00	$109.99	1
$.00	$218.00	2
$.00	$119.98	2
$.00	$89.00	1
$.00	$113.00	1
$.00	$199.99	1
$.00	$165.00	1
$.00	$77.99	1
$.00	$268.00	2
$.00	$103.98	2
$.00	$92.99	1
$84.50	$93.50	2
$.00	$113.00	1
$80.00	-$25.01	1
$150.00	$51.99	2
$.00	$92.00	1
$.00	$119.95	1
$.00	$159.98	2
$.00	$109.99	1
$.00	$94.00	1
$.00	$67.98	2
$.00	$201.98	2
$174.75	$24.25	1
$.00	$629.00	1
$.00	$109.00	1
$.00	$59.99	1
$.00	$189.00	1
$.00	$166.00	2
$9.00	$19.99	1
$119.60	-$10.60	1
$.00	$77.99	1
$.00	$98.99	1

次のリクエストでは、ページの拡大縮小を使用します。

```
SET SQUEEZE=ON
DEFINE FILE WF_RETAIL_LITE
SHOWPIC/A100='C:¥ibi¥WebFOCUS93¥samples¥web_
resource¥signin¥images¥favicon.jpg';
END

TABLE FILE WF_RETAIL_LITE
PRINT  PRODUCT_CATEGORY
COGS_US REVENUE_US MSRP_US DISCOUNT_US GROSS_PROFIT_US QUANTITY_SOLD
```

```
BY SHOWPIC NOPRINT
BY CONTINENT_NAME
BY COUNTRY_NAME
WHERE COUNTRY_NAME EQ 'FRANCE' OR 'ITALY'
WHERE RECORDLIMIT=3000;
ON TABLE SUBHEAD
" "
" "
" Report With PDF Scaling "
" "
" "
ON COUNTRY_NAME SUBHEAD
" "
" "
ON TABLE PCHOLD FORMAT PDF

ON TABLE SET STYLE *
TYPE=REPORT, PAGE-SCALE=AUTO, $
TYPE=DATA, COLUMN=CONTINENT_NAME, FONT=COMIC SANS MS,
  COLOR=BLUE, STYLE=BOLD+ITALIC, $
TYPE=DATA, COLUMN=PRODUCT_CATEGORY, COLOR=FUSCHIA, $
TYPE=HEADING, STYLE=BOLD, COLOR=RGB(0 35 95), SIZE=12, JUSTIFY=CENTER, $
TYPE=SUBHEAD, SIZE=18, STYLE=BOLD, COLOR=RED, $
TYPE=SUBHEAD, IMAGE=(SHOWPIC), SIZE=(.5 .5), $
TYPE=TABHEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER, $
ENDSTYLE
END
```

出力結果は次のとおりです。すべての列がページ幅に収まり、パネル化されません。



PAGE 1

Report With PDF Scaling

Customer Continent	Customer Country	Product Category	Cost of Goods	Revenue	MSRP	Discount	Gross Profit	Quantity Sold
Europe	France	Camcorder	$82.00	$179.00	179.00	$.00	$97.00	1
		Camcorder	$60.00	$169.99	169.99	$.00	$109.99	1
		Video Production	$380.00	$598.00	598.00	$.00	$218.00	2
		Media Player	$260.00	$379.98	379.98	$.00	$119.98	2
		Accessories	$80.00	$169.00	169.00	$.00	$89.00	1
		Computers	$167.00	$280.00	280.00	$.00	$113.00	1
		Accessories	$500.00	$699.99	699.99	$.00	$199.99	1
		Video Production	$234.00	$399.00	399.00	$.00	$165.00	1
		Stereo Systems	$122.00	$199.99	199.99	$.00	$77.99	1
		Televisions	$730.00	$998.00	998.00	$.00	$268.00	2
		Stereo Systems	$96.00	$199.98	199.98	$.00	$103.98	2
		Media Player	$87.00	$179.99	179.99	$.00	$92.99	1
		Accessories	$160.00	$253.50	338.00	$84.50	$93.50	2
		Computers	$167.00	$280.00	280.00	$.00	$113.00	1
		Stereo Systems	$145.00	$119.99	199.99	$80.00	-$25.01	1
		Camcorder	$398.00	$449.99	599.98	$150.00	$51.99	2
		Camcorder	$87.00	$179.00	179.00	$.00	$92.00	1
		Accessories	$380.00	$499.95	499.95	$.00	$119.95	1
		Stereo Systems	$400.00	$559.98	559.98	$.00	$159.98	2
		Media Player	$420.00	$529.99	529.99	$.00	$109.99	1
		Camcorder	$205.00	$299.00	299.00	$.00	$94.00	1
		Media Player	$92.00	$159.98	159.98	$.00	$67.98	2
		Accessories	$398.00	$599.98	599.98	$.00	$201.98	2
		Camcorder	$500.00	$524.25	699.00	$174.75	$24.25	1
		Camcorder	$2,870.00	$3,499.00	3,499.00	$.00	$629.00	1
		Video Production	$190.00	$299.00	299.00	$.00	$109.00	1
		Media Player	$220.00	$279.99	279.99	$.00	$59.99	1
		Stereo Systems	$310.00	$499.00	499.00	$.00	$189.00	1
		Stereo Systems	$312.00	$478.00	478.00	$.00	$166.00	2
		Stereo Systems	$61.00	$80.99	89.99	$9.00	$19.99	1
		Video Production	$190.00	$179.40	299.00	$119.60	-$10.60	1
		Stereo Systems	$122.00	$199.99	199.99	$.00	$77.99	1
		Stereo Systems	$121.00	$219.99	219.99	$.00	$98.99	1
		Stereo Systems	$262.00	$399.98	399.98	$.00	$137.98	2
		Accessories	$80.00	$169.00	169.00	$.00	$89.00	1
		Media Player	$410.00	$599.98	599.98	$.00	$189.98	2
		Camcorder	$199.00	$299.99	299.99	$.00	$100.99	1
		Media Player	$640.00	$799.98	799.98	$.00	$159.98	2
		Televisions	$587.00	$679.99	799.99	$120.00	$92.99	1
		Media Player	$220.00	$279.99	279.99	$.00	$59.99	1
		Video Production	$330.00	$439.98	439.98	$.00	$109.98	2
		Camcorder	$389.00	$549.00	549.00	$.00	$160.00	1
		Accessories	$83.00	$168.00	168.00	$.00	$85.00	1
		Camcorder	$60.00	$169.99	169.99	$.00	$109.99	1
		Stereo Systems	$290.00	$399.98	399.98	$.00	$109.98	2
		Stereo Systems	$48.00	$99.99	99.99	$.00	$51.99	1
		Stereo Systems	$720.00	$763.30	898.00	$134.70	$43.30	2
		Media Player	$370.00	$499.99	499.99	$.00	$129.99	1

PDF レポートのページ位置揃え

JUSTIFYREPORT スタイルシート属性を使用することで、ページ内で PDF レポート全体を左揃え、中央揃え、右揃えにすることができます。

PDF レポートを左揃え、中央揃え、右揃えにするには、プロシジャに次の構文を追加します。

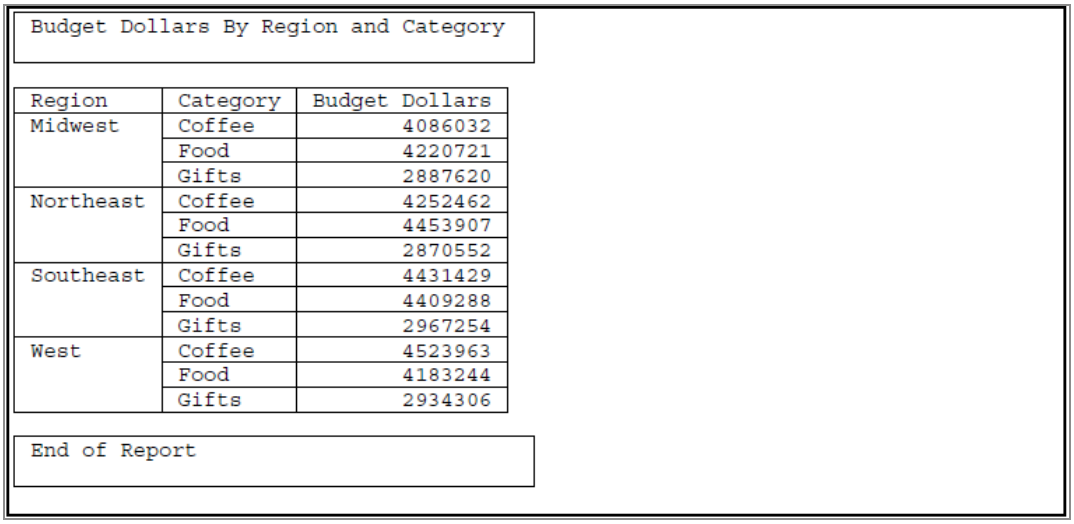
```
TYPE=REPORT, JUSTIFYREPORT={LEFT|CENTER|RIGHT},$
```

PDF レポートのページ左揃え

PDF レポートを左揃えにするには、次のプロシジャのように、TYPE=REPORT, JUSTIFYREPORT=LEFT 属性を追加します。

```
TABLE FILE GGSALES
SUM BUDDOLLARS
BY REGION
BY CATEGORY

HEADING
"Budget Dollars By Region and Product Category "
" "
FOOTING
"End of Report "
" "
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE PCHOLD FORMAT PDF

ON TABLE SET STYLE *
SQUEEZE=ON, GRID=ON, $
TYPE=REPORT,
JUSTIFYREPORT=LEFT,$
ENDSTYLE

END
```

出力結果は次のとおりです。



Budget Dollars By Region and Category

Region	Category	Budget Dollars
Midwest	Coffee	4086032
	Food	4220721
	Gifts	2887620
Northeast	Coffee	4252462
	Food	4453907
	Gifts	2870552
Southeast	Coffee	4431429
	Food	4409288
	Gifts	2967254
West	Coffee	4523963
	Food	4183244
	Gifts	2934306

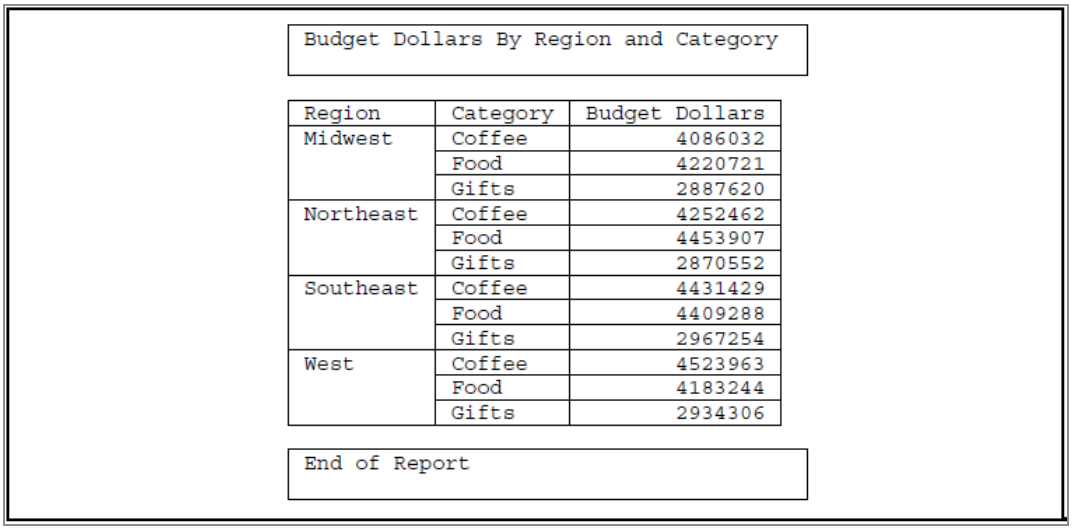
End of Report

PDF レポートのページ中央揃え

PDF レポートを中央揃えにするには、次のプロシジャのように、TYPE=REPORT, JUSTIFYREPORT=CENTER 属性を追加します。

```
TABLE FILE GGSALES
SUM BUDDOLLARS
BY REGION
BY CATEGORY

HEADING
"Budget Dollars By Region and Product Category "
" "
FOOTING
"End of Report "
" "
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE PCHOLD FORMAT PDF

ON TABLE SET STYLE *
SQUEEZE=ON, GRID=ON, $
TYPE=REPORT,
JUSTIFYREPORT=CENTER,$
ENDSTYLE

END
```

出力結果は次のとおりです。



Budget Dollars By Region and Category

Region	Category	Budget Dollars
Midwest	Coffee	4086032
	Food	4220721
	Gifts	2887620
Northeast	Coffee	4252462
	Food	4453907
	Gifts	2870552
Southeast	Coffee	4431429
	Food	4409288
	Gifts	2967254
West	Coffee	4523963
	Food	4183244
	Gifts	2934306

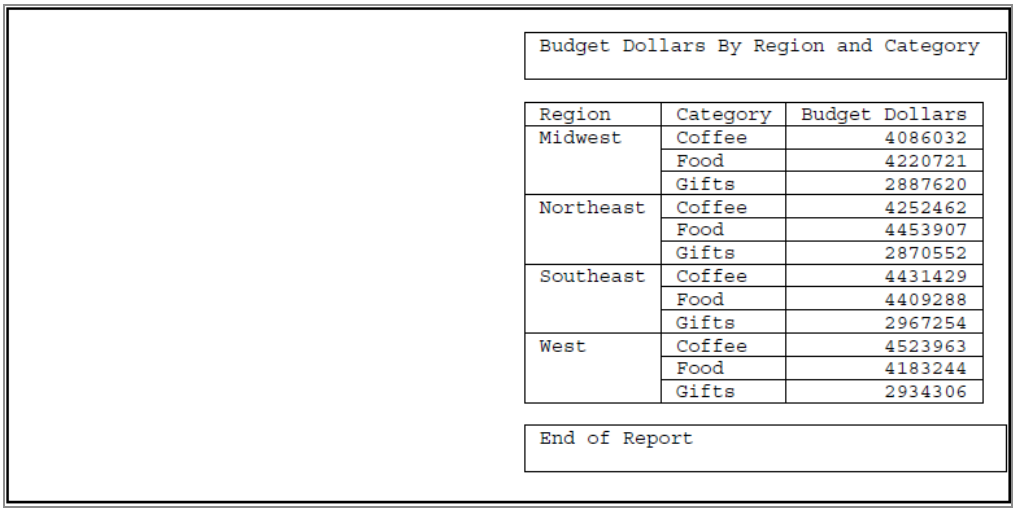
End of Report

PDF レポートのページ右揃え

PDF レポートを右揃えにするには、次のプロシジャのように、TYPE=REPORT, JUSTIFYREPORT=RIGHT 属性を追加します。

```
TABLE FILE GGSALES
SUM BUDDOLLARS
BY REGION
BY CATEGORY

HEADING
"Budget Dollars By Region and Product Category "
" "
FOOTING
"End of Report "
" "
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE PCHOLD FORMAT PDF

ON TABLE SET STYLE *
SQUEEZE=ON, GRID=ON, $
TYPE=REPORT,
JUSTIFYREPORT=RIGHT,$
ENDSTYLE

END
```

出力結果は次のとおりです。



Budget Dollars By Region and Category

Region	Category	Budget Dollars
Midwest	Coffee	4086032
	Food	4220721
	Gifts	2887620
Northeast	Coffee	4252462
	Food	4453907
	Gifts	2870552
Southeast	Coffee	4431429
	Food	4409288
	Gifts	2967254
West	Coffee	4523963
	Food	4183244
	Gifts	2934306

End of Report

WebFOCUS PDF レポートのアクセシビリティサポート

WebFOCUS PDF レポートのアクセシビリティ機能では、スクリーンリーダーなどの支援テクノロジがサポートされます。

注意：アクセシビリティの原則およびフォントタイプと使用についての詳細は、WebAIM の Web サイト (https://webaim.org/techniques/fonts/#intro) を参照してください。

WebFOCUS PDF レポート出力は、アクセシビリティの基準に準拠するよう次の機能を提供します。

- アクセシビリティを有効にする SET コマンドの WebFOCUS PDF 出力コードへの変更。
- バージョン 8.2.06 で導入された、複合レポートのメインドキュメントの見出しを識別する DisplayOn=DOC-HEADING スタイルシート属性。
    - 複合ドキュメントではない、単一のスタンドアロンレポートでは、1 つ目の ON TABLE SUBHEAD 文字列が自動的に <H1> としてタグ付けされます。他のページ見出しおよび脚注はすべて、<H2> としてタグ付けされます。
    - 複合レポートでは、ON TABLE SUBHEAD を含む単一の固定位置コンポーネントに、DisplayOn=DOC-HEADING 属性を追加する必要があります。DOC-HEADING 属性は、固定位置コンポーネントに挿入された ON TABLE SUBHEAD 文字列が <H1> としてタグ付けされ、メインドキュメントの見出しを表すことを示します。メインドキュメントの見出しは、ドキュメントの物理ページ 1 ページ目に 1 回だけ表示される見出しです。他のページ見出しおよび脚注はすべて、<H2> としてタグ付けされます。
- 見出しにカスタムフィールドタイトルを配置し、位置揃えする USEASTITLES スタイルシート属性。
- ドキュメント内の目的の場所への直接移動を可能にする BOOKMARK スタイルシート属性。
- レポート内に埋め込まれたイメージの説明を提供する ALT スタイルシート属性。
- ドリルダウンコンポーネントの説明を提供する ALT スタイルシート属性。
- ドキュメントのデフォルト言語を指定する LANG 属性。
- レポート出力。SET ACCESSPDF コマンドを有効にし、BYDISPLAY=ON 設定で作成された場合、ソート (BY) フィールドの各セルに値が生成されます。

注意：セクション 508 (米国リハビリテーション法 508 条) に準拠したレポートを作成するには、一般的なアクセシビリティ標準に従う必要があります。

アクセシビリティ対応 PDF コードの制御

SET ACCESSPDF コマンドを使用して、PDF レポートのアクセシビリティ機能を有効にすることができます。

PDF コードアクセシビリティの制御

- プロシジャまたはプロファイルでのすべてのリクエストでは、次の構文を使用します。

```
SET ACCESSPDF = {508|OFF}
```

- 単一リクエストでは、次の構文を使用します。

```
ON TABLE SET ACCESSPDF {508|OFF}
```

説明

508

セクション 508 アクセシビリティ要件に準拠した PDF ファイルを生成します。

OFF

セクション 508 アクセシビリティ要件に準拠しない PDF ファイルを生成します。これがデフォルト値です。

アクセシビリティ対応 PDF コードの制御

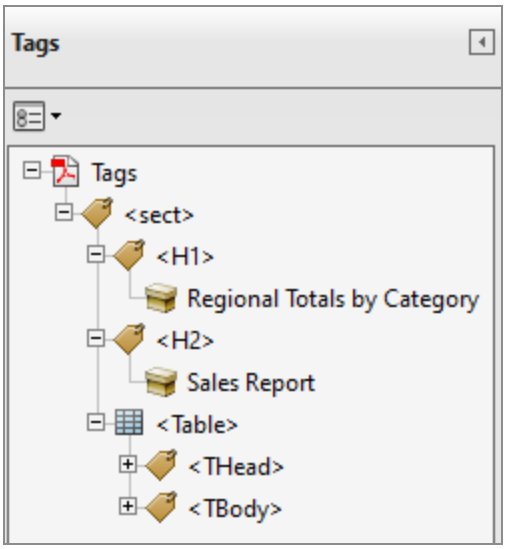
次のリクエストは、アクセシビリティ要件に準拠した PDF レポート出力を生成します。ここでは、アクセシビリティ機能を有効にする SET および SUBHEAD コマンドが太字で表示されています。

```
TABLE FILE GGSALES
SUM
   DOLLARS/D12M
BY REGION
BY CATEGORY
ON REGION SUBTOTAL AS 'Total for '
HEADING
" "
"Sales Report"
" "
ON TABLE SET ACCESSPDF 508
ON TABLE SET PAGE-NUM OFF
ON TABLE COLUMN-TOTAL AS 'Grand Total'
ON TABLE SUBHEAD "Regional Totals by Category"
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT='ARIAL',$
TYPE=TITLE,  COLUMN=N1, FONT='ARIAL', STYLE=BOLD,$
TYPE=TITLE,  COLUMN=N2, FONT='ARIAL', STYLE=BOLD,$
TYPE=TITLE,  COLUMN=N3, FONT='ARIAL', STYLE=BOLD,$
TYPE=HEADING, SIZE=14, STYLE=BOLD,$
TYPE=TABHEADING, SIZE=16, STYLE=BOLD,$
TYPE=SUBTOTAL, BY=1, STYLE=BOLD,$
TYPE=GRANDTOTAL, FONT='ARIAL', STYLE=BOLD,$
ENDSTYLE
END
```

WebFOCUS は、SET ACCESSPDF が 508 に設定された場合、次のタグを生成します。

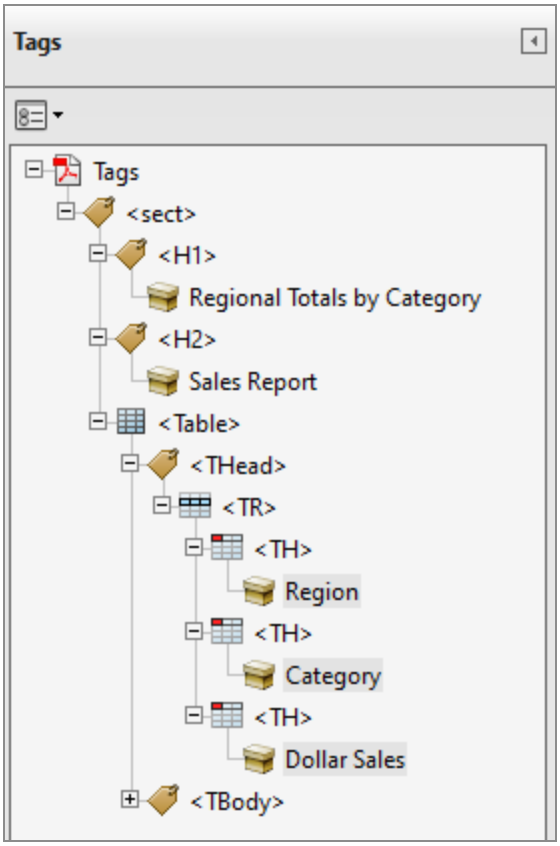
ページ見出し

見出しタグの <H1> および <H2> は、WebFOCUS で自動的に生成されます。



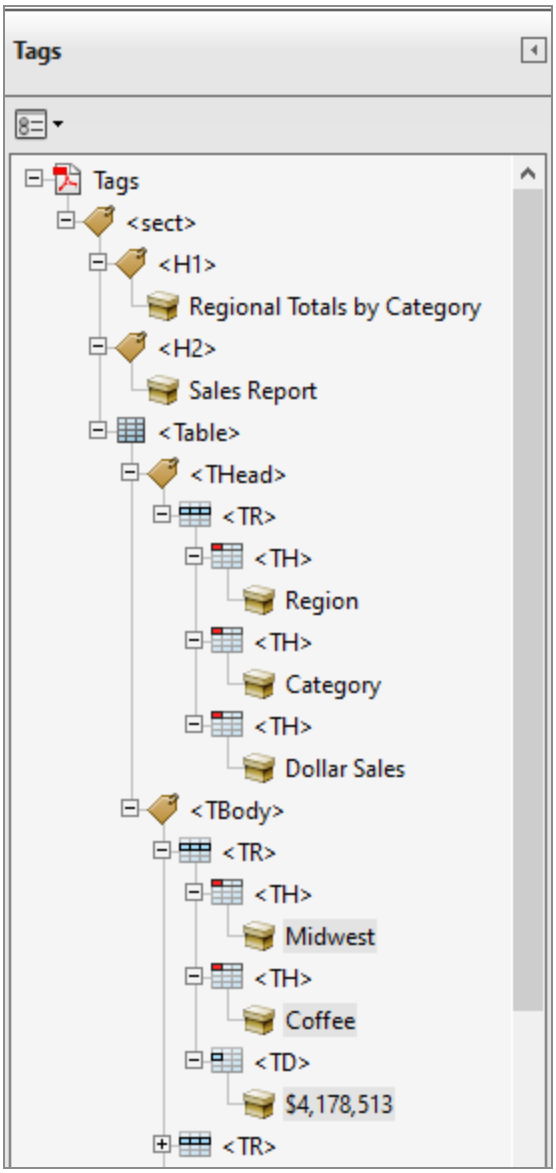
フィールドタイトル

行タグの <TR> およびセルタグの <TH> は、WebFOCUS で自動的に生成されます。フィールドタイトルは、適切なデータ列に合わせて位置揃えされます。



データセル

行タグの <TR> およびセルタグの <TH> は、WebFOCUS で自動的に生成されます。



出力結果は次のとおりです。

Regional Totals by Category

Sales Report

Region	Category	Dollar Sales
Midwest	Coffee	$4,178,513
Midwest	Food	$4,338,271
Midwest	Gifts	$2,883,881
Total for Midwest		**$11,400,665**
Northeast	Coffee	$4,164,017
Northeast	Food	$4,379,994
Northeast	Gifts	$2,848,289
Total for Northeast		**$11,392,300**
Southeast	Coffee	$4,415,408
Southeast	Food	$4,308,731
Southeast	Gifts	$2,986,240
Total for Southeast		**$11,710,379**
West	Coffee	$4,473,517
West	Food	$4,202,337
West	Gifts	$2,977,092
Total for West		**$11,652,946**
Grand Total		**$46,156,290**

ページ見出しの各要素と列データとの位置揃え

PDF レポートで USEASTITLES スタイルシート属性を使用することで、デフォルト設定のフィールドタイトルを使用する代わりに、見出しにカスタムフィールドタイトル (AS='text') を配置し、位置揃えすることができます。USEASTITLES 属性は、見出しのフィールドタイトルを適切なデータ列に関連付けます。

ページ見出しの各要素と列データとの位置揃え

```
TYPE=HEADING, USEASTITLES=ON, HEADALIGN=BODY,$
```

USEASTITLES=ON は、HEADING 全体にのみ設定できます。また、HEADALIGN=BODY も HEADALIG に対して設定する必要があります。

App Studio でこの属性を使用するための要件は、次のとおりです。

- レポート出力フォーマットを PDF に設定します。
- [ページ見出し] に対して [グリッド] を有効にします。
    - レポートにページ見出しを追加します。
    - [ページ見出し] を右クリックし、[グリッド] を選択します。
    - [グリッドの挿入] ダイアログボックスで、[データに合わせる] オプションを選択し、[OK] をクリックします。
    - ページ見出しでグリッドを右クリックし、[タイトルに合わせる (セクション 508)] を選択します。

ページ見出しの各要素と列データとの位置揃え

見出し要素の位置揃えを有効にする USEASTITLES 属性は、太字で表示されています。また、HEADALIGN=BODY も太字で表示されています。

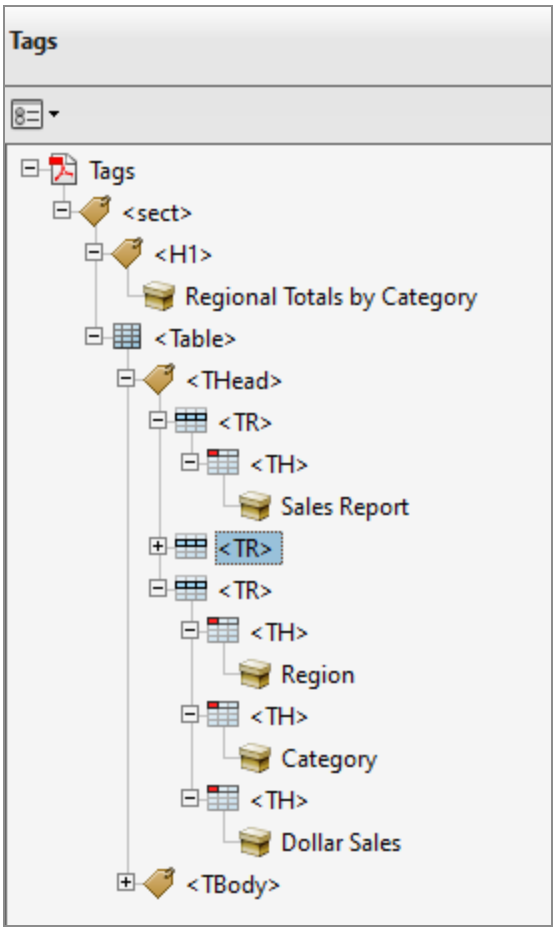
```
TABLE FILE GGSALES
SUM
   DOLLARS/D12M AS ''
BY REGION AS ''
BY CATEGORY AS ''
ON REGION SUBTOTAL AS 'Total for '
HEADING
" "
"Sales Report"
" <+0> <+0> "
"Region<+0>Category<+0>Dollar Sales"
```

```
ON TABLE SET ACCESSPDF 508
ON TABLE SET PAGE-NUM OFF
ON TABLE COLUMN-TOTAL AS 'Grand Total'
ON TABLE SUBHEAD "Regional Totals by Category <+1"
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
$
TYPE=REPORT, FONT='ARIAL',$
TYPE=TITLE, COLUMN=N1, FONT='ARIAL', STYLE=BOLD,$
TYPE=TITLE, COLUMN=N2, FONT='ARIAL', STYLE=BOLD,$
TYPE=TITLE, COLUMN=N3, FONT='ARIAL', STYLE=BOLD,$
TYPE=HEADING, SIZE=14, STYLE=BOLD,
HEADALIGN=BODY, USEASTITLES=ON,$
TYPE=HEADING, LINE=2, OBJECT=TEXT, ITEM=1, COLSPAN=3, JUSTIFY=LEFT,$
TYPE=HEADING, LINE=3, OBJECT=TEXT, ITEM=1, COLSPAN=1, JUSTIFY=LEFT,$
TYPE=HEADING, LINE=3, OBJECT=TEXT, ITEM=2, SIZE=10, COLSPAN=1,
     JUSTIFY=RIGHT,$
TYPE=HEADING, LINE=3, OBJECT=TEXT, ITEM=3, SIZE=10, COLSPAN=1,
     JUSTIFY=RIGHT,$
TYPE=HEADING, LINE=4, OBJECT=TEXT, ITEM=1, SIZE=11, COLSPAN=1,
     JUSTIFY=LEFT,$
TYPE=HEADING, LINE=4, OBJECT=TEXT, ITEM=2, SIZE=11, COLSPAN=1,
     JUSTIFY=LEFT,$
TYPE=HEADING, LINE=4, OBJECT=TEXT, ITEM=3, SIZE=11, COLSPAN=1,
     JUSTIFY=RIGHT,$
TYPE=TABHEADING, SIZE=16, STYLE=BOLD,$
TYPE=SUBTOTAL, BY=1, STYLE=BOLD,$
TYPE=GRANDTOTAL, FONT='ARIAL', STYLE=BOLD,$
ENDSTYLE
END
```

WebFOCUS は、USEASTITLES 属性が使用された場合に次のタグを生成します。

ページ見出しおよびフィールドタイトル

行タグの <TR> およびセルタグの <TH> は、WebFOCUS で自動的に生成されます。ページ見出しとともにフィールドタイトルも、ヘッダセル <TH> としてタグ付けされます。



出力結果は次のとおりです。

Regional Totals by Category

Sales Report

Region	Category	Dollar Sales
Midwest	Coffee	$4,178,513
Midwest	Food	$4,338,271
Midwest	Gifts	$2,883,881
Total for Midwest		**$11,400,665**
Northeast	Coffee	$4,164,017
Northeast	Food	$4,379,994
Northeast	Gifts	$2,848,289
Total for Northeast		**$11,392,300**
Southeast	Coffee	$4,415,408
Southeast	Food	$4,308,731
Southeast	Gifts	$2,986,240
Total for Southeast		**$11,710,379**
West	Coffee	$4,473,517
West	Food	$4,202,337
West	Gifts	$2,977,092
Total for West		**$11,652,946**
Grand Total		**$46,156,290**

しおりの追加

しおりは、ナビゲーションウィンドウ内のリンクで、ドキュメント内の目的の場所に直接移動することを可能にします。しおりを使用することで、ナビゲーションウィンドウ内のタグ付けされた項目から、この項目が存在するページに直接移動することができます。スクリーンリーダーと併せてしおりを使用するメリットは、ドキュメント全体の読み取りを待たずに指定された階層にナビゲートできることです。PDF でのしおりのナビゲーションについての詳細は、Adobe Web サイト (www.adobe.com) のマニュアルを参照してください。

注意

- PDF ドキュメントは完全なファイルであるため、しおりとスクロールを使用してファイル内をナビゲートすることができます。ドキュメント、特に 2 ページ以上の長さのドキュメントを簡単にナビゲートできます。
- 複合レイアウト構文に BOOKMARKS および TOC を追加することで、WebFOCUS 内でブックマーク (しおり) を生成することができます。さらに、コンポーネント定義で、

目次レベル (TOC レベル) と説明を指定します。必要に応じて、BYTOC レベルを指定することもできます。

- セクション 508 に準拠する PDF を生成する場合は、BOOKMARKS オブジェクトの使用をお勧めします。

複合レイアウトへのしおりの追加

位置揃えを有効にする BOOKMARK 属性は、太字で表示されています。両方のレポートの TOC-LEVEL=1 および BYTOC=2 属性も、太字で表示されています。これらの属性の説明は次のとおりです。

- TOC-LEVEL=*n*

  レポートレイアウトオブジェクトの目次レベルを、「n」として指定します。このオプションは、オブジェクトの目次内での階層順序を指定します。

  0 = オブジェクトは、目次に表示されません。

  1 = オブジェクトは、目次の第 1 レベルに表示されます。

  2 = オブジェクトは、目次の第 2 レベルに表示されます (以下同様)。

- BYTOC=*m*

  現在のコンポーネントエントリに含める BY フィールド数を、「m」として指定します。

次のリクエストは、しおりを追加し、複合レポートのメインドキュメント見出しを指定します。report0 コンポーネントは、DisplayOn=DOC-HEADING 属性を使用して、report0 コンポーネントの ON TABLE SUBHEAD 文字列が、PDF 出力のメインドキュメント見出しとしてタグ付けされていることを示します。

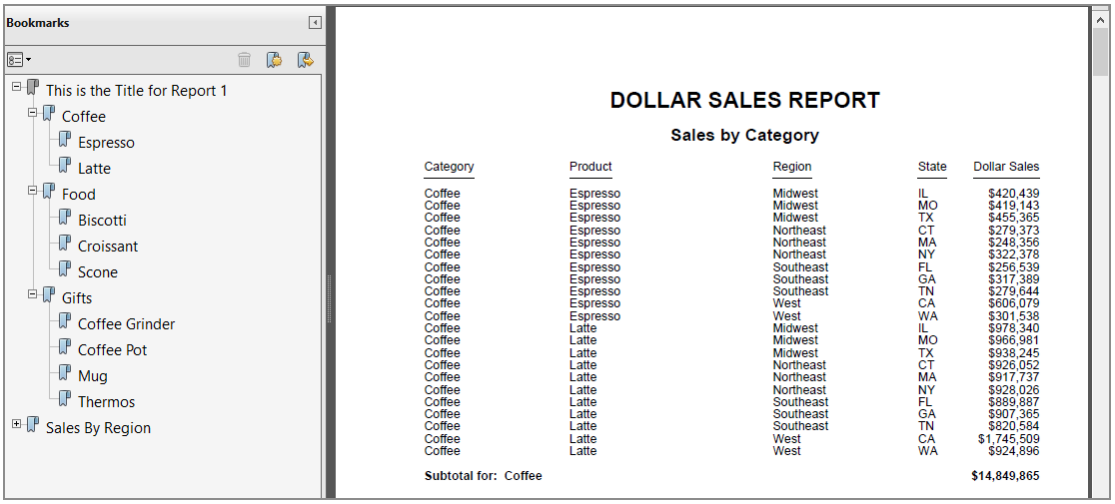
```
SET ACCESSPDF = 508
COMPOUND LAYOUT PCHOLD FORMAT PDF
OBJECT=BOOKMARKS,$
SECTION=S1, LAYOUT=ON, MERGE=OFF, ORIENTATION=PORTRAIT,$
PAGELAYOUT=1,$
COMPONENT=report0, position=(1 0.5), Dimension=(* *),
DisplayOn=DOC-HEADING,
$
COMPONENT=report1, TEXT='This is the Title for Report 1',
TOC-LEVEL=1,
BYTOC=2,
POSITION=(1 1), DIMENSION=(* *),$
COMPONENT=report2, TEXT='Sales By Region',
```

```
TOC-LEVEL=1,
BYTOC=2,
POSITION=(+0.00 +0.519), DIMENSION=(* *), RELATIVE-TO='report1',
REQUIRED-SPACE=(*3.5), RELATIVE-POINT=BOTTOM-LEFT, POSITION-POINT=TOP-LEFT,$
END
```

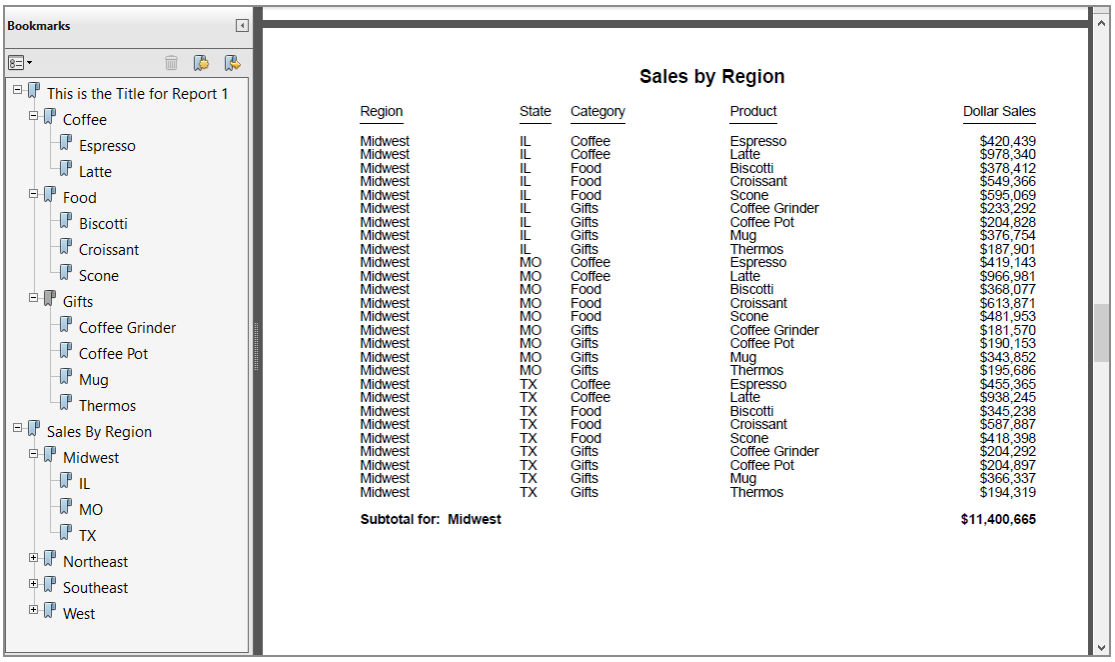
```
SET COMPONENT=report0
TABLE FILE GGSALES
BY CATEGORY NOPRINT
ON TABLE SUBHEAD
" "
"DOLLAR SALES REPORT"
" "
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=TABHEADING , SIZE=18, FONT=ARIAL, STYLE=BOLD, JUSTIFY=CENTER,$
ENDSTYLE
END

SET COMPONENT=report1
TABLE FILE GGSALES
SUM DOLLARS/F8M
BY CATEGORY
BY PRODUCT
BY REGION
BY ST
ON CATEGORY PAGE-BREAK
WHERE PRODUCT NE 'Capuccino'
ON CATEGORY SUBTOTAL AS 'Subtotal for: '
HEADING
"Sales by Category"
" "
ON TABLE HOLD FORMAT PDF
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=10, FONT=ARIAL,$
TYPE=HEADING, SIZE=14, FONT=ARIAL, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=SUBTOTAL, SIZE=10, STYLE=BOLD,$
ENDSTYLE
END
```

```
SET COMPONENT=report2
TABLE FILE GGSALES
SUM DOLLARS/F8M
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
ON REGION PAGE-BREAK
WHERE PRODUCT NE 'Capuccino'
ON REGION SUBTOTAL AS 'Subtotal for: '
HEADING
"Sales by Region"
" "
ON TABLE HOLD FORMAT PDF
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=10, FONT=ARIAL,$
TYPE=HEADING, SIZE=14, FONT=ARIAL, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=SUBTOTAL, SIZE=10, STYLE=BOLD,$
ENDSTYLE
END
COMPOUND END
```

下図は、しおり付きのレポート出力を示しています。

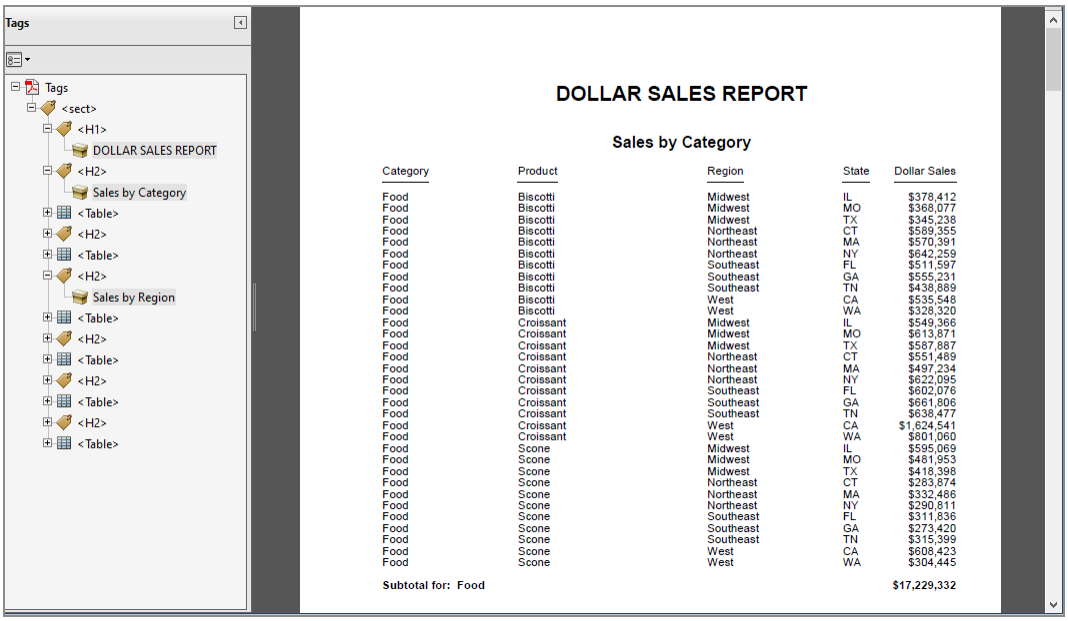


[しおり] パネルの階層ツリーには、レポートごとに 2 つのレベルがあり、BYTOC 属性で定義されています。下図のように、[しおり] ツリーの各レポートのタイトルは、TEXT 属性で指定されています。

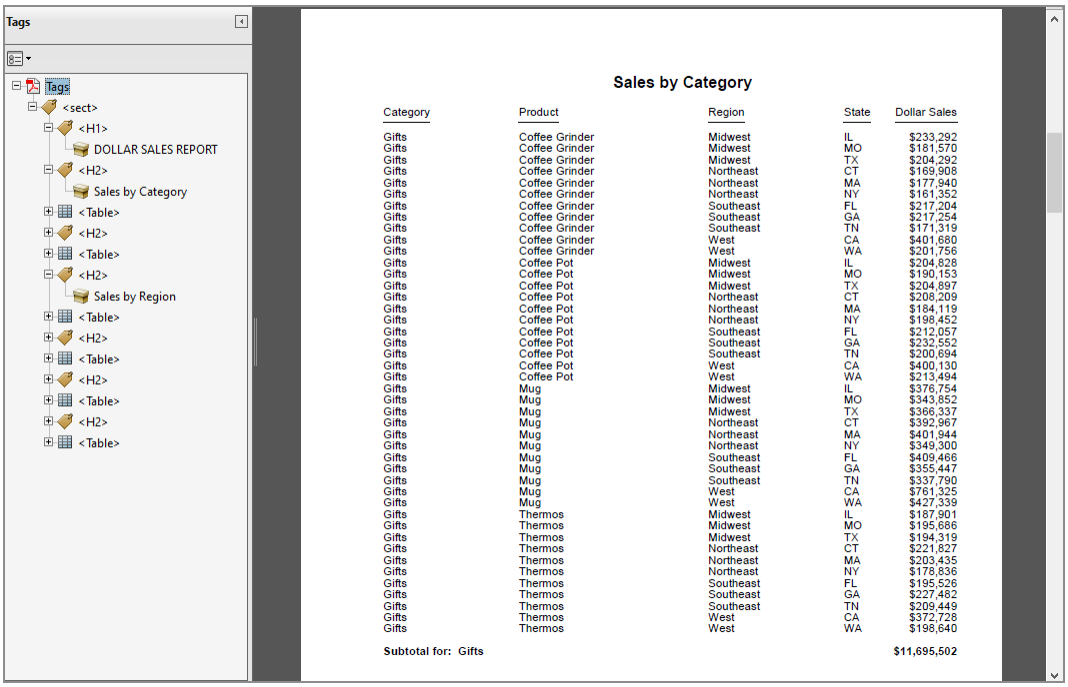


下図は、[タグ] パネルが表示された PDF 出力の最初の 3 ページを示しています。出力の最初のページにのみ <H1> 見出しの「DOLLAR SALES REPORT」が表示されていることに注意してください。これ以外の見出しは、 <H2> でタグ付けされています。

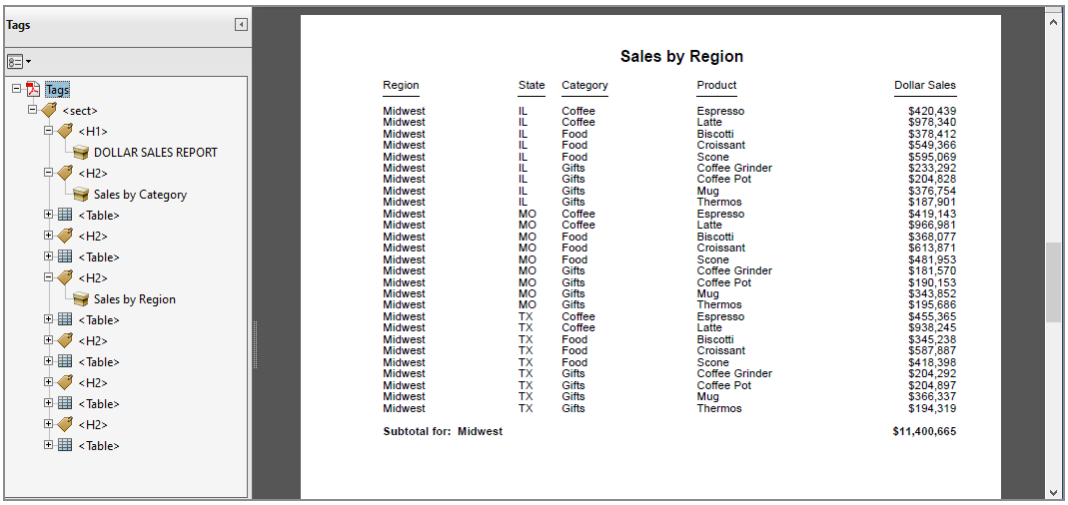
ページ 1



ページ 2



ページ 3



イメージへの説明テキストの追加

WebFOCUS スタイルシートの ALT 属性は、PDF レポートに埋め込まれたイメージに説明テキストを追加します。スタイルシートの ALT 属性は、 <IMG> タグに PDF ALT 属性を生成します。

ALT タグは、PDF 内のイメージ上にマウスポインタを置くと表示されます。

イメージに説明テキストを追加するには

```
ALT='description'
```

説明

description

イメージの簡単な説明です。一重引用符 (') で囲みます。最大長は 256 バイトです。

スタイルシート構文を使用してイメージを追加する方法についての詳細は、レポートへのイメージの追加を参照してください。

イメージへの説明テキストの追加

次のリクエストは、レポートの見出しに TIBCO ロゴを追加します。ここでは、WebFOCUS スタイルシートの ALT 属性を使用して、イメージを特定する説明テキスト (TIBCO ロゴ) を追加します。

```
TABLE FILE GGSALES
SUM
   DOLLARS/D12N
BY REGION
BY CATEGORY
ON REGION SUBTOTAL AS 'Total sales: '
ON REGION PAGE-BREAK
HEADING
"Sales Report"
" "
" "
" "
" "
" "
ON TABLE SET ACCESSPDF 508
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PDF
ON TABLE SUBHEAD "Regional Totals by Category"
ON TABLE SET STYLE *
     UNITS=IN,
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,
```

```
$
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9,$
TYPE=TITLE, STYLE=BOLD,$
TYPE=HEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=TABHEADING, SIZE=14, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=SUBTOTAL, BACKCOLOR=RGB(210 210 210), STYLE=BOLD,$
TYPE=REPORT, IMAGE=tibco.png, ALT='TIBCO logo',
POSITION=(0 .5),$
ENDSTYLE
END
```

レポートは次のとおりです。

リクエストの実行時に、イメージ上にマウスポインタを置くと説明テキストが表示されます。このテキストは、JAWS などのアクセシビリティツールで読み上げられます。

ドリルダウン情報の説明

ACCESSPDF を有効にすると、開発者は、WebFOCUS スタイルシートの ALT 属性を使用して、ドリルダウンの説明を追加することができます。PDF レポートでは、JAWS により、ドリルダウンコンポーネントの値が ALT テキストとともに読み上げられます。

ドリルダウンレポートの詳細コンポーネントの説明が追加されることにより、アクセシビリティがサポートされます。

ドリルダウン情報の説明の追加

```
ALT='description'
```

説明

description

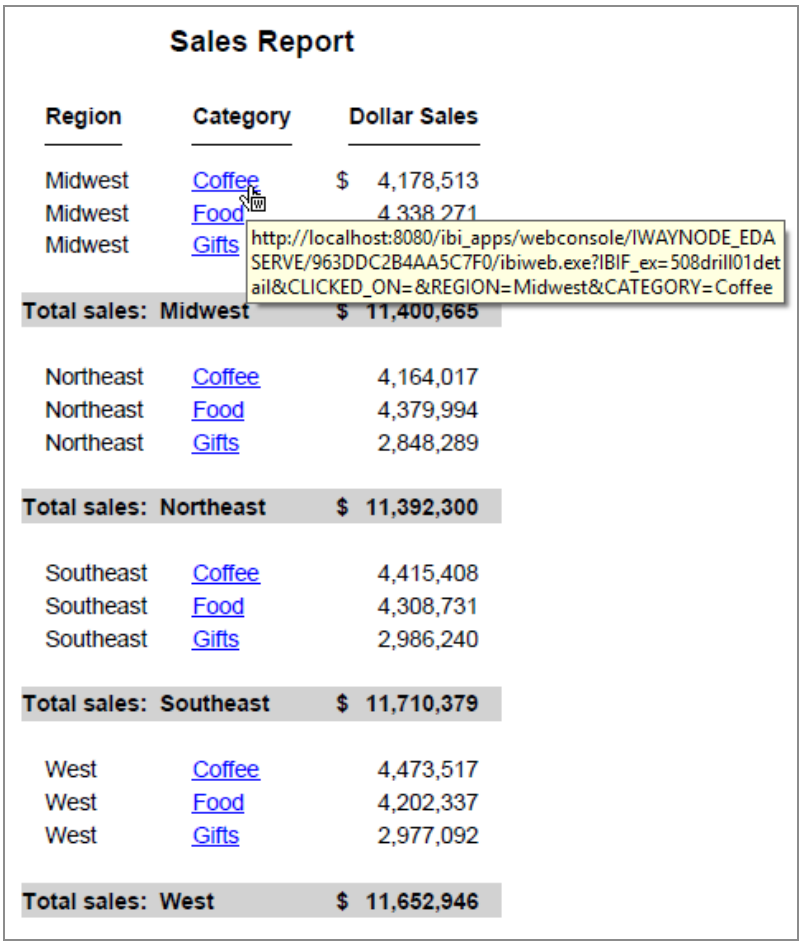
レポート内のドリルダウン情報の説明です。一重引用符 (') で囲みます。最大長は 256 バイトです。

ドリルダウンレポートのスタイルシート構文についての詳細は、レポートの別リソースへのリンクを参照してください。

```
TABLE FILE GGSALES
SUM
   DOLLARS/D12N
BY REGION
BY CATEGORY
ON REGION SUBTOTAL AS 'Total sales: '
HEADING
" "
ON TABLE SET ACCESSPDF 508
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE SUBHEAD "Sales Report"
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
     UNITS=IN,
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,
     SUMMARY='508 Sales report example',
     TITLETEXT='508 Sales report example',
$
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9,$
TYPE=DATA, COLUMN=N2,
ALT='Drill Down to detail report.', ¥
     FOCEXEC=508drill01detail( REGION=N1 CATEGORY=N2 ),$
TYPE=TITLE, STYLE=BOLD,$
TYPE=TABHEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=HEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=SUBTOTAL, BACKCOLOR=RGB(210 210 210), STYLE=BOLD,$
ENDSTYLE
END
```

ALT 属性および関連するテキストは、太字で表示されています。

レポートは次のとおりです。



[Midwest] の [Coffee] にマウスポインタを置くと、ドリルダウンの URL 情報がボックスに表示されます。スクリーンリーダー (例、JAWS) は、このボックス内のテキストを読み上げません。ACCESSPDF を有効にすると、ALT テキスト ('Drill Down to detail report') がドリルダウン (この例では、'Coffee') の値に追加されます。[Coffee] に移動すると、スクリーンリーダーが、「Coffee Drill Down to detail report」と応答します。

注意

- スクリーンリーダー (例、JAWS) を使用するユーザは、このドリルダウンリンクについてさらに情報が必要な場合があります。上記の例では、「Coffee」の値に複数のハイパーリンクが追加されています。この機能および 'read current cell' コマンドにより、ユーザは各値の相違を理解することができます。JAWS でのスクリーンリーダーコマンド 'read current cell' のショートカットは、Ctrl+Alt+Num5 です。この例では、JAWS は 'read current cell' に対して、「column two, row two, region Midwest, link Coffee Drill Down to detail report」と応答します。

- Adobe Readerで出力を開く場合は、FOCEXURL 設定を指定して、ドリルダウンがコールバックする WebFOCUS 環境の URL コンテキストを提供します。また、FOCEXURL の値には、リクエストがドリルダウンリクエストであることを指定するパラメーターが必要です。以下はその例です。

```
-SET &FOCEXURL='http://host:port/ibi_apps/WFServlet?IBIF_webapp=
   /ibi_apps' | '&';
-SET &FOCEXURL=&FOCEXURL | 'IBIMR_drill=IBFS,RUNFEX,IBIF_ex,true' | '&';
-SET &FOCEXURL=&FOCEXURL | 'IBIC_server=EDASERVE' | '&';
-SET &FOCEXURL=&FOCEXURL | 'IBIAPP_app=ibisamp' | '&';
SET FOCEXURL='&FOCEXURL'
```

アクセシビリティの制限事項

次の要素を使用するレポートは、アクセシビリティ要件に準拠しません。

- プロシジャに SET PAGE-NUM=OFF または ON TABLE SET PAGE-NUM OFF コマンドが含まれない場合、スクリーンリーダーによってページ番号が 2 回 読み上げられます。
- ACROSS レポートおよび OVER レポート。
- Financial Modeling Language (FML)。
- **FOLD-LINE**　レポートに FOLD-LINE が使用されている場合、アクセシビリティ機能は無効になります。
- **クレジットネガティブ (CR) を含む固定通貨編集オプション (N, !d, !e, !l、!y)**　指定された通貨オプションまたは CR オプションに対して一意のセルが作成されます。
- FOOTING、SUBHEAD、SUBFOOT、RECAP、SUMMARIZE、RECOMPUTE の各コマンドに追加または設定されたテキストでは、行見出しタグが生成されません。

ワードプロセッサ表示フォーマットの使用 - DOC、WP

次のフォーマットを使用して、レポートをワードプロセッサドキュメントテキストとして表示することができます。

- **DOC フォーマット**　Web ブラウザ内の Microsoft Word でレポートを開きます。Word でレポートを開く際、テキスト変換についての情報の入力が要求される場合は、デフォルトの選択を受け入れます。レポートを表示するコンピュータに Word がインストールされている必要があります。

  DOC フォーマットには改ページが含まれます。これにはレポートの先頭よりも前の初期改ページも含まれます。改ページを除外するには、プロシジャのはじめに SET PAGE = NOPAGE コマンドを発行します。

  DOC フォーマットでは、スタイルシートはサポートされません。

- **WP フォーマット**　レポートは Web ブラウザでテキストとして開きます。

  SET PAGE = OFF コマンドを発行するか、見出しまたは脚注に TABPAGENO を含めると、各区切りの先頭フィールドに「1」の文字を含めることで、改ページが示されます。これは S/390 環境では改ページコントロールとして認識されます。WP フォーマットでは、ほとんどのブラウザまたはワードプロセッサプログラムにより認識される改ページは含まれません。

  WP フォーマットではスタイルシートはサポートされません。

WebFOCUS でのレポートの実行時に PCHOLD コマンドを使用すると、レポートをテキストフォーマットのワードプロセッサドキュメントとして表示することを指定できます。

Excel XLSX フォーマットでのレポート出力の保存

Excel 2007 では、Microsoft から導入された新しいブックファイル形式でスプレッドシート機能が強化されました。WebFOCUS を使用して、WebFOCUS でサポートされている任意のデータソースからデータを取得し、データ分析と配信に使用可能な XLSX フォーマット (Excel 2007、Excel 2010、Excel 2013) のブックを生成することができます。以下の説明は、特に明記されていない限り、Excel 2007、Excel 2010、Excel 2013 に適用されます。

WebFOCUS の XLSX/EXL07 フォーマットは、次の Microsoft Office ソフトウェア製品をサポートします。

- Microsoft Office 2013/2010/2007、および Microsoft Office 互換性パックを使用した Microsoft Office 2000/2003
- Open Office サポート (FORMAT EXL07/XLSX)。WebFOCUS バージョン 8 では、EXL07/XLSX フォーマットで生成される主要な Excel 機能が Open Office でサポートされます。Open Office についての詳細は、「http://www.openoffice.org/」を参照してください。

- MAC Office 2008 および 2011。FORMAT EXL07/XLSX は、WebFOCUS バージョン 8 で動作保証されています。

WebFOCUS は、Microsoft XLSX 標準に基づいて XLSX ブックを生成します。これらのブックには、Microsoft XLSX ファイルをサポートするブラウザおよびモバイルアプリケーションのすべてからアクセス可能です。

- Microsoft Office 365　Microsoft Office 365 は、Microsoft Excel 2013 のローカルインストール機能を提供します。Office 2010 に対応し、Office 2007 の一部機能にも対応しています。Microsoft Office 365 では、Office Online を使用してほとんどのデバイスからアクセス可能なクラウドに Microsoft Excel のファイルをアップロードすることもできます。Microsoft Office 365 についての詳細は、「Office 365 for business FAQ」を参照してください。

  Microsoft Office 365 を使用して、WebFOCUS XLSX レポートにアクセスすることができます。最初に PCHOLD コマンドを使用して画面に XLSX レポートを表示し、次にこのレポートを OneDrive for Business に保存します。ファイルがクラウドに保存された後は、Office Online を使用してこのファイルにアクセスできます。

  Excel Online および Microsoft Office 2013 で使用可能な機能の相違についての詳細は、「Office Online Service Description」を参照してください。

  Office Online および OneDrive for Business の使用についての詳細は、「Using Office Online in OneDrive」を参照してください。

EXL07/XLSX フォーマットの概要

FORMAT EXL07 と FORMAT XLSX は同義語で、相互に置き換えて使用することができます。指定した構文に関係なく、ファイルの保存メッセージには常に「XLSX ファイルを保存しました」と表示されます。

WebFOCUS プロシジャは、単一ワークシートが含まれた新しいブックを生成します。このワークシートに、定義されたレポート要素 (例、見出し、中間合計) およびスタイルシート構文 (例、条件付きスタイル設定、ドリルダウン) を含むレポートが出力されます。

- fontuser.xml ファイルで、XLSX の新しいデフォルトフォントを定義することができます。This allows ease in customizing the look and feel of XLSX workbooks.WebFOCUS では、デフォルトフォントとして Arial が使用されます。ただし、デフォルトフォントは、Microsoft Office 標準フォント (Calibri) または社内の標準フォントに一致させることができます。
- XLSX では、フォーマット設定された数値、文字、日付がそれぞれのフォーマットで正確に表示されます。

- XLSX FORMULA を使用して、集計情報 (例、列合計、行合計、一時項目 (COMPUTE)) を Excel 数式に変換することができます。その結果、Excel ワークシートで値を編集した際に、集計情報が自動的に更新されるようになります。
- ReportCaster では、XLSX ブックおよび XLSX FORMULA ブックの配信がサポートされます。
- 生成されたワークシートでは、レポートの列に最大長の値が収まるよう列幅が自動的に調整されます (SQUEEZE=ON)。WebFOCUS は、PDF や DHTML などのスタイルフォーマット用に作成されたフォントメトリックスを使用して、各データ列のすべてのセルのフォントとサイズの要件に基づいて列幅を計算します。列幅は、レポートのデータおよびタイトル要素に基づいて計算されます。見出しおよび脚注要素は列幅計算には使用されず、これらの要素はデータ列の要件に基づいてサイズが調整されます。
- デフォルト設定では、データ行および各種タイトル行に標準の高さが使用されます。見出し、脚注、中間見出し、中間脚注の行は、データ行より高く設定され、テキストの折り返しが可能になるとともに、見出しとデータが区別しやすくなります。
- TITLETEXT スタイルシート属性を使用して、ブック内のタブ名をカスタマイズし、ワークシートのコンテンツを分かりやすくすることができます。
- HTML ベースのフォーマット (EXL2K) では、ワークシートのセル内の先頭、内部、末尾のブランクはすべて削除されますが、XLSX フォーマットでは、デフォルト設定でこれらのブランクは保持されます。
- XLSX ワークシートには、最大で 1,048,576 行、16,384 列を含めることができます。WebFOCUS は、この既定の制限値を超えるワークシートを生成しますが、このブックを Excel で開くことはできません。
- 新しい XLSX ファイルフォーマットが ZIP 圧縮形式のため、キャレット (<) などのネイティブ HTML 記号はタグ文字としてサポートされません。XLSX では、他の出力フォーマットと異なり、HTMLENCODE のデフォルト値が ON に設定されています。HTMLENCODE を OFF に設定すると、HTML タグ文字が含まれたデータがセルから除外されます。SET HTMLENCODE コマンドについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

.xlsx ブックファイルの作成

Excel 2007 では、Excel ブックのフォーマットおよび構造が変更されています。新しい .xlsx ファイルは、複数の xml ファイルを含むバイナリ圧縮ファイルです。WebFOCUS でこの新しいファイルフォーマットを生成する処理は、レポート出力を含む xml ファイルを生成する処理と、xml ドキュメントをバイナリ .xlsx フォーマットに圧縮する処理の 2 つで構成され

ます。xml 生成処理は、Reporting Server が実行します。圧縮処理は、クライアント (WebFOCUS Servket)、サーバ (JSCOM3)、または C ベースの ZIP ローカルモードのいずれかで実行することができます。

- **WebFOCUS Servlet**　Application Server 内の WebFOCUS Client が圧縮処理を実行します。この処理は、ローカルクライアントで実行することも、リモートアクセスするクライアントで実行することもできます。Servlet を使用するオプションは、各 WebFOCUS Client で定義されているデフォルトの処理方法です。デフォルト設定では、クライアントの指定先がそのクライアント自体になっています。
- **JSCOM3**　Reporting Server の Java 層が圧縮処理を実行します。このオプションは、セキュリティの設定された Web サーバまたは Application Server で WebFOCUS Servlet が構成されている場合に使用する必要があります。これは、JSCOM3 がリモート WebFOCUS Client への URL アクセスを必要としないためです。
- **C ベースの ZIP ローカルモード**　.xlsx ファイルの圧縮および解凍はローカルで処理され、サーバとの通信は行われません。C ベースの ZIP ローカルモードの実行は、パフォーマンスが向上し、Server または WebFOCUS Servlet URL による直接呼び出しでも使用できます。

EXCELSERVERURL による .xlsx ファイルの圧縮方法の選択

圧縮を実行する方法と場所を指定するには、EXCELSERVURL を URL (WebFOCUS Servlet の場合) に設定するか、ブランク (JSCOM3 の場合) に設定します。この値を特定のプロシジャに対して設定することも、環境全体に対して設定することもできます。

- **プロシジャの場合**　プロシジャで SET EXCELSERVURL コマンドを発行します。
- **環境全体の場合**　WebFOCUS 管理コンソールで IBIF_excelservurl 変数を編集します。

  IBIF_excelservurl 変数を指定するには、[構成] タブで [アプリケーションの設定]、[Client 設定] を順に選択し、[Client 設定] ページの [Excel Server URL] ドロップダウンリストから値を選択します。

  WebFOCUS 管理コンソールへのアクセスおよび IBIF_excelservurl 変数の設定についての詳細は、『ibi™ WebFOCUS® セキュリティ管理ガイド』を参照してください。

EXCELSERVURL に割り当てた値に基づいて、WebFOCUS Servlet または JSCOM3 のどちらが圧縮処理を実行するかが決定されます。

- **Servlet の指定**　WebFOCUS Servlet を使用するよう指定するには、EXCELSERVURL

パラメータまたは IBIF_excelservurl 変数をこの URL に設定します。以下はその例です。

プロシジャで指定する場合

```
SET EXCELSERVURL = http://servername:8080/ibi_apps
```

WebFOCUS 管理コンソールで指定する場合

```
IBIF_excelservurl = http://servername:8080/ibi_apps
```

- **JSCOM3 の指定**　現在の Reporting Server で JSCOM3 を使用するよう指定するには、EXCELSERVURL をブランクに設定します。

  プロシジャで指定する場合

  ```
  SET EXCELSERVURL = ''
  ```

  WebFOCUS 管理コンソールで指定する場合

  ```
  IBIF_excelservurl = ''
  ```

デフォルト設定で、各 WebFOCUS Client には、その Client 自体を指定する次の URL が定義されています。

```
&URL_PROTOCOL://&servername:&server_port&IBIF_webapp
```

.xlsx ファイル圧縮での C ベース ZIP ローカルモードの指定

.xlsx ファイルの ZIP に C ベース ZIP ローカルモードを指定するには、プロシジャで次のコマンドを発行します。

```
SET XLSXLOCALZIP = ON
```

Excel XLSX ブックの生成

レポートを XLSX ブックに保存するか、ブラウザまたは Excel アプリケーションに表示するかを指定することができます。

```
ON TABLE {PCHOLD|HOLD} AS name FORMAT XLSX
```

説明

PCHOLD

デスクトップ設定に基づいて、生成されたブックをブラウザまたは Excel アプリケーションのいずれかに表示します。

HOLD

.xlsx 拡張子のブックを、指定された場所に保存します。

name

生成されるブックの名前を指定します。

注意：生成されるブックにファイル名を割り当てるには、WebFOCUS Client 出力先変更設定で、.xlsx ファイル拡張子の [保存レポート] オプションを [はい] に設定します。Excel アプリケーションで開いた場合、生成されたブックには、指定した AS 名が保持されます。詳細は、『ibi™ WebFOCUS® セキュリティ管理ガイド』を参照してください。

XLSX レポート出力を開く

XLSX ブックを開くには、デスクトップに Excel 2013、2010、2007 のいずれかがインストールされている必要があります。

Excel 2000/2003 で XLSX レポート出力を開く

Excel 2000 および Excel 2003 は、Microsoft Office 互換機能パックを使用することで、Excel XLSX ブックを読み取れるよう更新することができます。この互換機能パックは、

Microsoft ダウンロードサイト (http://www.microsoft.com/downloads/en/default.aspx) から入手することができます。拡張子が .xlsx (XLSX ブック) のファイルを開く際に、Microsoft Office 互換機能パックが、このファイルを Excel 2000/2003 で読み取り、開くために必要な変換を実行します。

Microsoft Office 互換機能パックを使用すること以外に、WebFOCUS Client 出力先変更設定で [保存レポート] オプションを有効にしておくことが重要です。これにより、ユーザがレポート出力を .xlsx ファイル拡張子でマシンに保存しなくても、Excel 2000/2003 で XLSX レポート出力を開けるようになります。WebFOCUS Client 出力先変更設定の [保存レポート] オプションを使用して、ファイルタイプごとに WebFOCUS Client がレポート出力をどのようにユーザのマシンに送信するかを構成します。このオプションは次のように設定することができます。

- **[保存レポート] オプションを無効 (いいえ) に設定する**　WebFOCUS Client 出力先変更設定の [保存レポート] オプションは、デフォルト設定で無効 (いいえ) になっています。[保存レポート] オプションが [いいえ] に設定されている場合、WebFOCUS Client がユーザのマシンに送信するレポート出力には、WebFOCUS Client 出力先変更設定の構成ファイル (mime.wfs) でレポートフォーマットに対して指定されたアプリケーション関連付け情報が付加されます。

  ユーザのマシンに Excel 2007/2010 がインストールされていない場合、Excel 2007/2010 のアプリケーション関連付け情報は認識されず、Excel のメッセージが表示されます。

  Excel 2000/2003 ユーザは、[保存] を選択し、拡張子が .xlsx のファイル名を入力して、レポート出力をユーザのマシンに保存することができます。これで、ユーザは Excel 2000/2003 で .xlsx ファイルを直接開くことができます。

- **[保存レポート] オプションを有効 (はい) に設定する**　[保存レポート] オプションが [はい] に設定されている場合、WebFOCUS Client がユーザに送信するレポート出力は、WebFOCUS Client 出力先変更設定の構成ファイル (mime.wfs) で指定された拡張子のファイルになります。

  ファイルを受信した時点で [ファイルのダウンロード] ダイアログボックスが表示され、指定されたアプリケーションタイプでファイルを開くか、保存するかの選択がユーザに要求されます。[ファイルのダウンロード] ダイアログボックスには、Excel XLSX ファイルタイプとして認識されたレポート出力の名前が .xlsx 拡張子とともに表示されます。

  注意：[ファイルのダウンロード] ダイアログボックスは、Excel 2007/2010 がマシンにインストールされているユーザも含めて、すべてのユーザに表示されます。

  Excel 2000/2003 ユーザがファイルを開くよう選択した場合、Microsoft Office 互換機能パックが .xlsx ファイル拡張子を認識し、必要な変換を実行します。これにより、Excel 2000/2003 で Excel XLSX ブックの読み取りが可能になります。

Excel 2007/2010 ユーザがファイルを開くよう選択した場合、Excel が .xlsx ファイル拡張子を認識し、Excel XLSX ブックを読み取ります。

WebFOCUS Client 出力先変更設定についての詳細は、『ibi™ WebFOCUS® セキュリティ管理ガイド』を参照してください。

Excel ブックのブラウザ表示と Excel アプリケーション表示

クライアントに送信された Excel 出力が Internet Explorer ブラウザウィンドウと Excel アプリケーションのどちらに表示されるかは、オペレーティングシステムおよびデスクトップの設定に基づいて決定されます。Windows 環境で Excel 出力が [同じウィンドウで開く] に設定されている場合、WebFOCUS リクエストで生成されたブックは Internet Explorer ブラウザウィンドウに表示されます。[同じウィンドウで開く] オプションのチェックをオフにした場合、レポート出力はスタンドアロン Excel アプリケーションウィンドウに表示されるため、WebFOCUS によって作成されたブラウザウィンドウはブランクになります。

- Windows XP 以前のバージョンでは、ファイルタイプの設定はデスクトップの Windows エクスプローラで管理します。この設定を行うには、エクスプローラの [ツール] メニューから [フォルダオプション] を選択し、[ファイルタイプ] タブをクリックした後、拡張子 (.xls または .xlsx) を選択し、[詳細設定] ボタンをクリックして [同じウィンドウで開く] のチェックをオンにします。
- Windows 7 では、ワークシートをブラウザ内で開くようにするデスクトップ設定が削除されました。そのため、[フォルダオプション] ダイアログボックスを表示してこの動作を変更することができなくなり、代わりにレジストリ設定を変更する必要があります。

注意：この変更方法は、EXL2K および XLSX フォーマットで共通です。異なる点は、生成する Excel 出力のバージョンに基づいて選択するファイルタイプのみです。

XLSX レポート出力のセル値のフォーマット設定

マスターファイルまたはプロシジャで定義された WebFOCUS フォーマットは、生成された Excel XLSX ワークシートのセルに反映されます。可能な場合、WebFOCUS フォーマットは

カスタム Excel フォーマットに変換され、生データとして転送された値に適用されます。Excel のセルに転送される各データ値は、値とフォーマットマスクの組み合わせで定義されます。データフォーマットは、値に埋め込まれるのではなく、セルに関連付けられます。この方法により、WebFOCUS で生成されたワークシートの編集作業が強化されます。既存のセルに新しい値を入力した場合でもセルのフォーマットが保持され、これらの値はレポート内の列に定義されているスタイルで表示されます。

Excel に転送可能なデータのタイプは次のとおりです。

- **数値**　対応する Excel フォーマットマスクを定義できる場合、数値は、関連するフォーマットマスクが適用された生データ値として転送されます。同等のフォーマットマスクを定義できない場合、数値はテキストとして転送されます。
- **文字**　文字フォーマットは、[標準] フォーマットに設定されたテキストとして Excel に転送されます。デフォルト設定で、[標準] フォーマットではすべてのテキストフィールドが左揃えで表示されます。配置およびその他のスタイル属性をこれらのセルに適用することで、デフォルト設定を上書きすることができます。
- **日付**　有効な Excel 日付フォーマットを定義するための十分な要素がデータに含まれている場合、そのデータは日付値の生データとして転送され、WebFOCUS フォーマットが Excel 日付フォーマットマスクに変換されます。有効な Excel 日付値を作成するための十分な情報が WebFOCUS フォーマットに含まれていない場合、日付はテキストに変換されます。
- **日付時間**　データは日付時間値の生データとして転送され、WebFOCUS フォーマットが [カスタム] フォーマットの Excel 日付時間フォーマットマスクに変換されます。
- **テキスト**　テキスト値は、[標準] フォーマットに設定された文字列として転送されます (文字データと同様)。

注意：この動作は、EXL2K フォーマットで変更された点です。以前は、日付および複雑な数値フォーマットを含むセルは、フォーマット設定されたテキストとして転送されていました。

XLSX レポート出力でのフォーマット設定済み数値の表示

WebFOCUS の数値フォーマットは、Excel のカスタム数値フォーマットに変換されます。数値は、選択したセルの Excel 数式バーに表示されます。実際のセルには、フォーマットマスクが適用された値が表示されます。

次の数値データタイプの WebFOCUS フォーマットは、Excel XLSX フォーマットマスクに変換されるため、生成されたブックで完全な編集作業がサポートされます。

- データタイプ E、F、D、I、P
- カンマ編集オプション (C)
- ゼロサプレス (S)
- リーディングゼロ (L)
- 浮動通貨記号 (M)
- カンマサプレス (c)
- 右側マイナス記号 (-)
- クレジットネガティブ (CR)
- ブラケットネガティブ (B)
- 固定拡張通貨記号 (!d、!e、!l、!y)
- 浮動拡張通貨記号 (!D、!E、!L、!Y)
- パーセント (%)

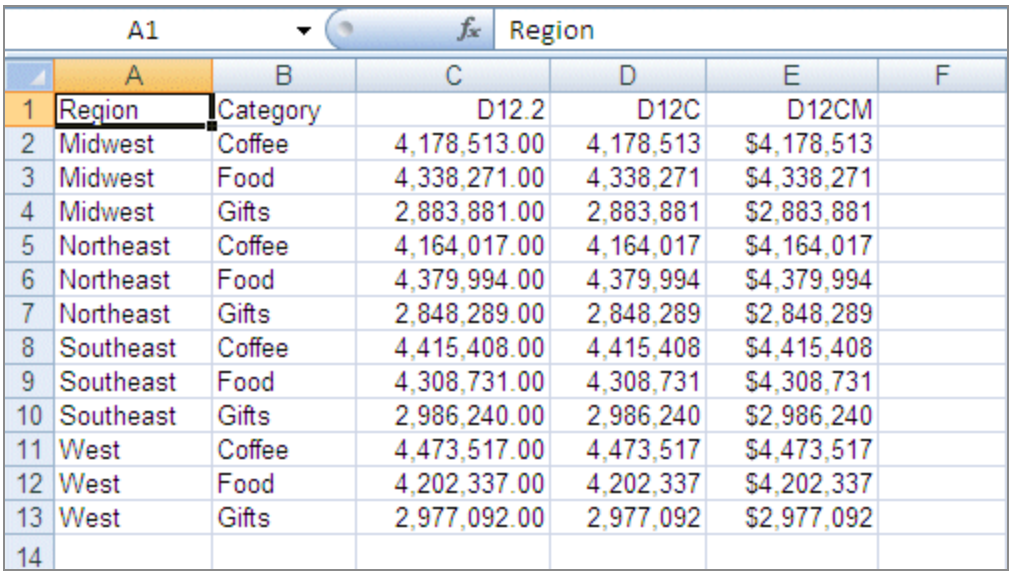
XLSX レポート出力への数値フォーマットの転送

次の例では、DOLLARS フィールドに複数の異なる数値フォーマットを割り当てることで、使用可能な各オプションの相違点を示します。フィールドタイトルは、適用された WebFOCUS フォーマットオプションが表示されるよう編集されています。

```
TABLE FILE GGSALES
SUM DOLLARS/D12.2 AS 'D12.2'
    DOLLARS/D12C  AS 'D12C'
    DOLLARS/D12CM AS 'D12CM'
BY REGION
BY CATEGORY
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET BYDISPLAY ON
END
```

生成されたワークシートでは、Midwest Coffee の DOLLAR 値が格納された C2 セルに、WebFOCUS フォーマット D12.2 (カンマ (,) および小数点以下 2 桁) の値が表示されています。数式バーには、フォーマットが設定されていない実際の値が表示されています。各行の個々の DOLLAR 値を確認すると、数式バーに表示された値は変更されておらず、各セルに表示された表示値のみが変更されていることが分かります。

また、SET BYDISPLAY ON の設定により、BY フィールド値がワークシートの各行に繰り返し表示されています。この設定により、重要な情報を失うことなく、Excel のさまざまなデータソート、フィルタ、テーブル機能で使用可能な完全なデータ行が作成されます。この設定は、すべてのワークシートに使用することをお勧めします。



A1 | Region

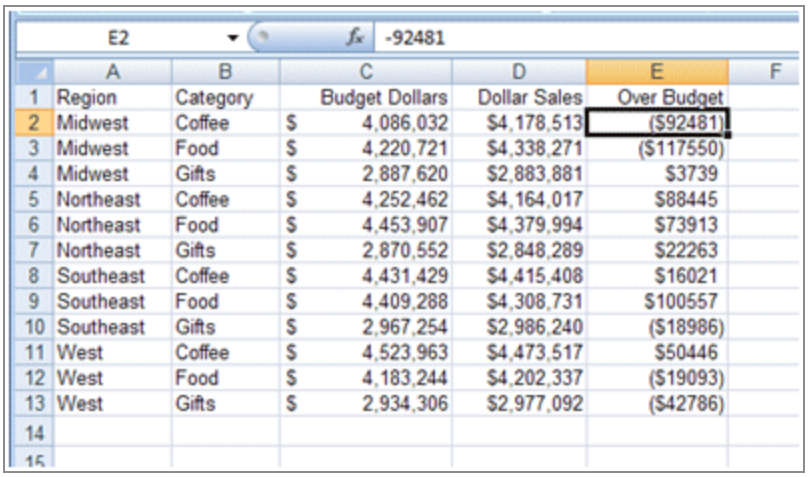
	A	B	C	D	E	F
1	Region	Category	D12.2	D12C	D12CM	
2	Midwest	Coffee	4,178,513.00	4,178,513	$4,178,513	
3	Midwest	Food	4,338,271.00	4,338,271	$4,338,271	
4	Midwest	Gifts	2,883,881.00	2,883,881	$2,883,881	
5	Northeast	Coffee	4,164,017.00	4,164,017	$4,164,017	
6	Northeast	Food	4,379,994.00	4,379,994	$4,379,994	
7	Northeast	Gifts	2,848,289.00	2,848,289	$2,848,289	
8	Southeast	Coffee	4,415,408.00	4,415,408	$4,415,408	
9	Southeast	Food	4,308,731.00	4,308,731	$4,308,731	
10	Southeast	Gifts	2,986,240.00	2,986,240	$2,986,240	
11	West	Coffee	4,473,517.00	4,473,517	$4,473,517	
12	West	Food	4,202,337.00	4,202,337	$4,202,337	
13	West	Gifts	2,977,092.00	2,977,092	$2,977,092	
14						

次の例では、固定ドル (N) フォーマット、および複数のフォーマットオプションの組み合わせを使用します。WebFOCUS フォーマットの各オプションは、それぞれ適切な Excel XLSX フォーマットマスクに変換され、セル値に割り当てられます。

```
TABLE FILE GGSALES
SUM BUDDOLLARS/D12N
    DOLLARS/D12M
COMPUTE OVERBUDGET/D12BMc = BUDDOLLARS-DOLLARS; AS 'Over Budget'
BY REGION
BY CATEGORY
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET BYDISPLAY ON
END
```

BUDDOLLARS フィールド (C 列) に定義された固定数値フォーマットでは、データ値の長さに関係なく、各セル内の固定位置にローカル通貨記号が表示されています。数式バーでは、一時項目 (COMPUTE) の「Over Budget」の値が、必要に応じて負の値として転送されてい

ます。実際のセルでは、カスタム Excel XLSX フォーマットマスクの一部として、負の値に括弧付きスタイルが適用されています。



E2 -92481

	A	B	C	D	E	F
1	Region	Category	Budget Dollars	Dollar Sales	Over Budget	
2	Midwest	Coffee	$ 4,086,032	$4,178,513	($92481)	
3	Midwest	Food	$ 4,220,721	$4,338,271	($117550)	
4	Midwest	Gifts	$ 2,887,620	$2,883,881	$3739	
5	Northeast	Coffee	$ 4,252,462	$4,164,017	$88445	
6	Northeast	Food	$ 4,453,907	$4,379,994	$73913	
7	Northeast	Gifts	$ 2,870,552	$2,848,289	$22263	
8	Southeast	Coffee	$ 4,431,429	$4,415,408	$16021	
9	Southeast	Food	$ 4,409,288	$4,308,731	$100557	
10	Southeast	Gifts	$ 2,967,254	$2,986,240	($18986)	
11	West	Coffee	$ 4,523,963	$4,473,517	$50446	
12	West	Food	$ 4,183,244	$4,202,337	($19093)	
13	West	Gifts	$ 2,934,306	$2,977,092	($42786)	
14						
15						

レポート見出しおよび脚注での数値フォーマットの使用

デフォルト設定では、見出しおよび脚注は単一文字列として Excel に転送されます。各行での位置を調整するためのスポットマーカーはサポートされません。見出しまたは脚注内の数値フィールドおよび日付は、見出しまたは脚注のコンテンツ全体に含まれるテキストとして転送されます。

見出しおよび脚注内の数値フィールドおよび日付を数値または日付値として表示するには、スタイルシートに HEADALIGN=BODY を使用して、見出し内の各項目をそれぞれ個別のセルとして定義します。数値または日付値が格納された各セルは、関連するフォーマットマスクが適用された値として転送されます。

見出しおよび脚注での数値フォーマット区切り文字の使用

データ列では、すべての通貨フォーマットが Excel XLSX フォーマットマスクに基づいて変換され、デスクトップの地域設定で定義された区切り文字ルールが使用されます。

コンチネンタル式 10 進表記を使用する言語では、通貨の定義により、小数点にはカンマ (,)、3 桁ごとの区切り文字にはピリオド (.) を使用するよう指定されています。そのため、D12.2CM では、値が英語表記 (米国) の「$ 9,999.99」ではなく、「$ 9.999,99」と表示されることがあります。見出しおよび脚注では、SET CDN=ON コマンドを発行することにより、区切り文字をコンチネンタル式 10 進表記に変換されるよう指定することができます。この設定を有効にすると、見出しおよび脚注テキスト内に埋め込まれたデータには、変換後の区切り文字を使用したフォーマットが設定されます。各項目をそれぞれ個別のセルに表示し、フィールド内で数値フォーマットを保持するには、HEADALIGN=BODY を指定します。これにより、データ列内でレポートデータと同一の規則が適用されます。

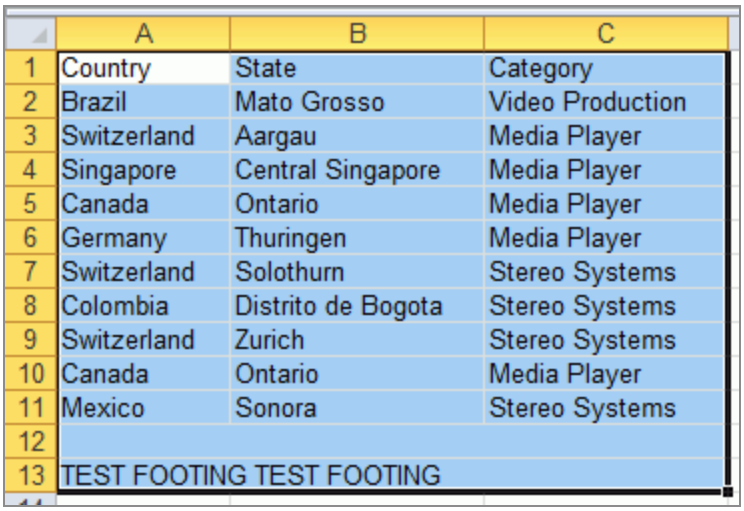
脚注を含む XLSX レポートのソートに関する使用上の注意

XLSX フォーマットでは、レポート脚注がデータテーブルの一部として Excel に含まれます。これは、EXL2K の場合と動作が異なります。EXL2K フォーマットでは、脚注はテーブルの一部として含まれません。

たとえば、次のプロシジャを実行し、データテーブルをソートした場合、リクエスト以下の図に示すように、レポート脚注がデータテーブルの一部となっています。

```
TABLE FILE WF_RETAIL_LITE
PRINT COUNTRY_NAME AS Country
STATE_PROV_NAME AS State
PRODUCT_CATEGORY AS Category
WHERE RECORDLIMIT EQ 10
FOOTING
""
"TEST FOOTING TEST FOOTING"
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
ENDSTYLE
END
```

出力結果は次のとおりです。



	A	B	C
1	Country	State	Category
2	Brazil	Mato Grosso	Video Production
3	Switzerland	Aargau	Media Player
4	Singapore	Central Singapore	Media Player
5	Canada	Ontario	Media Player
6	Germany	Thuringen	Media Player
7	Switzerland	Solothurn	Stereo Systems
8	Colombia	Distrito de Bogota	Stereo Systems
9	Switzerland	Zurich	Stereo Systems
10	Canada	Ontario	Media Player
11	Mexico	Sonora	Stereo Systems
12			
13	TEST FOOTING TEST FOOTING		

この問題を回避するには、次のプロシジャのように、プロシジャに名前付きセル範囲を追加します。

```
TABLE FILE WF_RETAIL_LITE
PRINT COUNTRY_NAME AS Country
STATE_PROV_NAME AS State
PRODUCT_CATEGORY AS Category
WHERE RECORDLIMIT EQ 10
FOOTING
""
"TEST FOOTING TEST FOOTING"
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=DATA, IN-RANGES=DATA, $TYPE=TITLE, IN-RANGES=DATA, $
ENDSTYLE
END
```

下図では、レポート脚注がデータテーブルの一部として含まれていません。

	A	B	C
1	Country	State	Category
2	Brazil	Mato Grosso	Video Production
3	Switzerland	Aargau	Media Player
4	Singapore	Central Singapore	Media Player
5	Canada	Ontario	Media Player
6	Germany	Thuringen	Media Player
7	Switzerland	Solothurn	Stereo Systems
8	Colombia	Distrito de Bogota	Stereo Systems
9	Switzerland	Zurich	Stereo Systems
10	Canada	Ontario	Media Player
11	Mexico	Sonora	Stereo Systems
12			
13	TEST FOOTING TEST FOOTING		

XLSX レポート出力への日付の転送

ほとんどの変換済み日付および SmartDate は、フォーマットマスクを適用して、標準の日付値として Excel に転送することができます。これにより、その日付値を Excel の関数、式、ソート順序に使用することができます。

Excel 2007 では先頭大文字の日付テキストのみがサポートされるため、WebFOCUS フォーマットで指定された大文字と小文字の区別に関係なく、すべての月名および曜日名は先頭大文字で表示されます。たとえば、WebFOCUS 日付フォーマットが WRYMTRD の場合、Excel フォーマット以外のすべてのフォーマットでは、日付テキスト情報は大文字で表示されます。Excel では、この値が先頭大文字に自動的に変換されます。

HTML では、日付フォーマットは次のように表示されます。

XLSX では、日付フォーマットは次のように表示されます。

WebFOCUS 日付の Excel XLSX 日付への変換

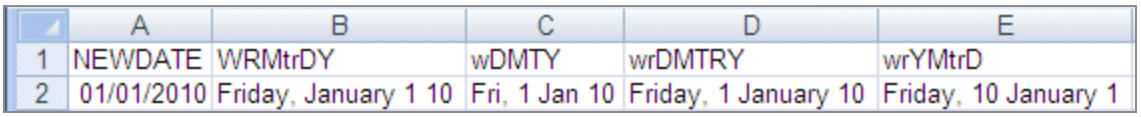
次のリクエストは、GGSALES データソースに対して実行され、「January 1, 2010」という日付を作成し、変換済みテキストを含む 4 つの日付フォーマットに変換します。

```
DEFINE FILE GGSALES
NEWDATE/MDYY = '01/01/2010';
WRMtrDY/WRMtrDY = NEWDATE;
wDMTY/wDMTY = NEWDATE;
wrDMTRY/wrDMTRY = NEWDATE;
wrYMtrD/wrYMtrD = NEWDATE;
END
TABLE FILE GGSALES
SUM DATE NOPRINT
NEWDATE WRMtrDY wDMTY wrDMTRY wrYMtrD
ON TABLE PCHOLD FORMAT XLSX
END
```

下表は、日付がどのように表示されるかを示しています。

WebFOCUS フォーマット	WebFOCUS 表示	XLSX 表示	XLSX 値
WRMtrDY	FRIDAY, January 1 10	Friday, January 1 10	2010/1/1
wDMTY	Fri, 1 JAN 10	Fri, 1 Jan 10	2010/1/1
wrDMTRY	Friday, 1 JANUARY 10	Friday, 1 January 10	2010/1/1
wrYMtrD	FRIDAY, 10 JANUARY 1	Friday, 10 January 1	2010/1/1

Excel 2007/2010 では、すべてのセルにフォーマットマスクが適用された日付値が配置され、WebFOCUS フォーマットで指定された大文字と小文字の区別に関係なく、すべての月名および曜日名は先頭大文字で表示されます。出力結果は次のとおりです。



	A	B	C	D	E
1	NEWDATE	WRMtrDY	wDMTY	wrDMTRY	wrYMtrD
2	01/01/2010	Friday, January 1 10	Fri, 1 Jan 10	Friday, 1 January 10	Friday, 10 January 1

日構成要素を含まない日付の転送

日付フォーマットで日の値を明示的に指定しない場合、その値は月の初日の日付値として定義されます。そのため、セルに配置された値がデータソースフィールドの日構成要素の値と異なる場合があり、この値を Excel 数式でのソートまたは日付計算に使用すると、予期しない結果になる可能性があります。

下表は、WebFOCUS 日付フォーマットが XLSX でどのように表されるかを示しています。この表から、WebFOCUS 日付フォーマットに適用するフォーマットマスクを使用することで、値がセル内でどのように保持され、表示がどのように生成されるかが分かります。

```
DATEFLD/MDYY = '01/02/2010'
```

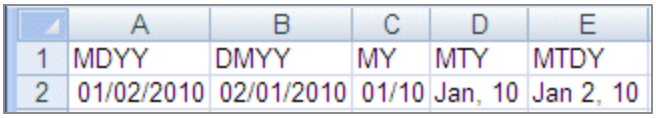
WebFOCUS フォーマット	XLSX 表示	XLSX 値
DMYY	02/01/2010	2010/1/2
MY	01/10	2010/1/1
MTY	Jan, 10	2010/1/1
MTDY	Jan 2, 10	2010/1/2

日構成要素を含まない WebFOCUS 日付の XLSX レポート出力への転送

次のリクエストは、GGSALES データソースに対して実行され、「January 2, 2010」という日付を作成し、その日付を MDYY、DMYY、MY、MTDY フォーマットで Excel に転送します。

```
DEFINE FILE GGSALES
NEWDATE/MDYY = '01/02/2010';
END
TABLE FILE GGSALES
SUM DATE NOPRINT
NEWDATE AS 'MDYY' NEWDATE/DMYY AS 'DMYY' NEWDATE/MY AS 'MY'
        NEWDATE/MTY AS 'MTY' NEWDATE/MTDY AS 'MTDY'
ON TABLE PCHOLD FORMAT XLSX
END
```

D 列および E 列にフォーマットマスクが適用された実際の日付値が挿入され、Excel 2007/2010 では先頭大文字で表示されます。MTY フォーマットには日構成要素が含まれていないため、格納される日付値は、January 2, 2010 (2010/1/2) ではなく、January 1, 2010 (2010/1/1) になります。



	A	B	C	D	E
1	MDYY	DMYY	MY	MTY	MTDY
2	01/02/2010	02/01/2010	01/10	Jan, 10	Jan 2, 10

Excel 数式で使用する日構成要素の転送

個々の構成要素 (例、D、Y、M、W) でフォーマット設定された日付は、数値として Excel に転送され、その数値を Excel の日付関数のパラメータとして使用することができます。これらの値は、Excel で数値として認識される [標準] フォーマットとして転送されます。

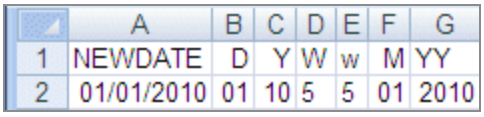
XLSX レポート出力への数値日付構成要素の転送

次のリクエストは、GGSALES データソースに対して実行され、「January 1, 2010」という日付を作成し、数値日付構成要素を抽出した後、Excel 2017/2010 に転送します。

```
DEFINE FILE GGSALES
NEWDATE/MDYY = '01/01/2010';
D/D = NEWDATE;
Y/Y = NEWDATE;
W/W = NEWDATE;
w/w = NEWDATE;
M/M = NEWDATE;
```

```
YY/YY = NEWDATE;
END
TABLE FILE GGSALES
SUM DATE NOPRINT
NEWDATE D Y W w M YY
ON TABLE PCHOLD FORMAT XLSX
END
```

出力結果は次のとおりです。



	A	B	C	D	E	F	G
1	NEWDATE	D	Y	W	w	M	YY
2	01/01/2010	01	10	5	5	01	2010

四半期フォーマットの転送

Excel では四半期フォーマットがサポートされないため、四半期構成要素を含む日付フォーマットは、常にテキストとして Excel に転送されます。

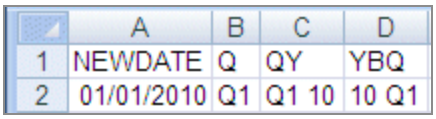
四半期構成要素を含む日付の XLSX レポート出力への転送

次のリクエストは、GGSALES データソースに対して実行され、「January 1, 2010」という日付を作成し、この日付を四半期構成要素を含む日付フォーマットに変換します。

```
DEFINE FILE GGSALES
NEWDATE/MDYY = '01/01/2010';
Q/Q = NEWDATE;
QY/QY = NEWDATE;
YBQ/YBQ = NEWDATE;
END
TABLE FILE GGSALES
SUM DATE NOPRINT
NEWDATE Q QY YBQ
ON TABLE PCHOLD FORMAT XLSX
END
```

XLSX では、四半期構成要素を含む日付のセルは [標準] フォーマットになります。これを確認するには、[セルの書式設定] ダイアログボックスを開きます。

出力結果は次のとおりです。



	A	B	C	D
1	NEWDATE	Q	QY	YBQ
2	01/01/2010	Q1	Q1 10	10 Q1

変換済みテキストとして定義された日付構成要素の転送

年や月の情報が含まれていないフォーマットなど、日付フォーマットに有効な日付を表すための十分な情報が含まれていない場合、Excel ではその日付は値に変換されません。これらの日付は、テキストとして Excel に転送されます。必要な情報が不足している場合、年はデフォルト値の現在の年になります。そのため、このタイプのフォーマットを日付値として転送すると、日付値は誤った値になります。次のフォーマットは、値として転送されません。

- MT、MTR、Mt、Mtr
- W、w、WR、wr

フォーマットマスクが適用された日付フォーマットが XLSX に転送されると、WebFOCUS フォーマットで指定された大文字と小文字の区別に関係なく、すべての月名および曜日名は先頭大文字で表示されます。ただし、この例では値が常にテキストとして転送されるため、WebFOCUS フォーマットで定義された大文字と小文字の区別が結果のセルに適用されます。

変換済みテキストとして定義された日付構成要素の XLSX レポート出力への転送

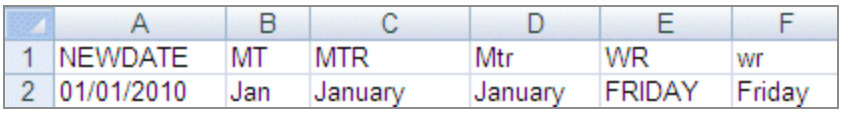
次のリクエストは、GGSALES データソースに対して実行され、「January 1, 2010」という日付を作成し、この日付を月名または曜日名として定義する日付フォーマットに変換します。

```
DEFINE FILE GGSALES
NEWDATE/MDYY = '01/01/2010';
MT/MT = NEWDATE;
MTR/MTR = NEWDATE;
Mtr/Mtr = NEWDATE;
WR/WR = NEWDATE;
```

```
wr/wr = NEWDATE;
END
TABLE FILE GGSALES
SUM DATE NOPRINT
NEWDATE MT MTR Mtr WR wr
ON TABLE PCHOLD FORMAT XLSX
END
```

Excel 2007 または 2010 では、日付を含むセルは [標準] フォーマットになります。これを確認するには、[セルの書式設定] ダイアログボックスを開きます。

出力結果は次のとおりです。



	A	B	C	D	E	F
1	NEWDATE	MT	MTR	Mtr	WR	wr
2	01/01/2010	Jan	January	January	FRIDAY	Friday

XLSX レポート出力での日付値の使用上の注意

- XLSX では、次の日付フォーマットはサポートされません。これらの日付フォーマットは、Excel の [標準] フォーマットに変換され、予期しない結果になる可能性があります。
  - JUL、YYJUL、I2MT
  - 日付表示オプションを使用して、パック 10 進数フィールドまたは文字フィールドとして格納された日付。

XLSX への日付時間値の転送

ほとんどの WebFOCUS 日付時間フォーマットは、フォーマットマスクを適用して、標準の日付時間値として XLSX に転送することができます。これにより、その日付時間値を Excel の関数、数式、ソート順序に使用することができます。

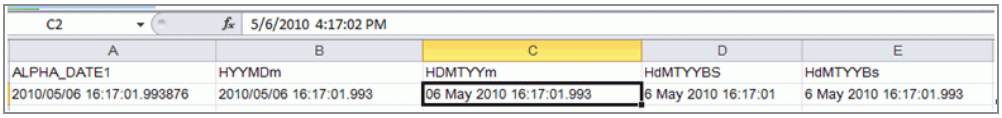
日付フォーマットと同様に、Excel では日付時間フィールドに先頭大文字のみがサポートされるため、Excel でサポートされる日付時間フォーマットにテキストが含まれている場合、WebFOCUS フォーマットで定義された大文字小文字の区別に関係なく、そのテキストが先頭大文字になります。

XLSX への日付時間値の転送

次の例は、GGSALES データソースに対して実行するリクエストを示しています。

```
DEFINE FILE GGSALES
DT1/HYYMDm WITH REGION = DT(20100506 16:17:01.993876);
DPT1/HDMTYYm = DT1;
ALPHA_DATE1/A30 = HCNVRT(DT1,'(HYYMDm)',30,'A30');
END
TABLE FILE GGSALES
PRINT
ALPHA_DATE1
DT1 AS 'HYYMDm'
DPT1 AS 'HDMTYYm'
DT1/HdMTYYBS   AS 'HdMTYYBS'
DT1/HdMTYYBs   AS 'HdMTYYBs'
ON TABLE SET SPACES 1
IF RECORDLIMIT EQ 1
ON TABLE PCHOLD FORMAT XLSX
END
```

出力結果は次のとおりです。



注意：Excel では分構成要素のみのフォーマットはサポートされず、値は [カスタム] フォーマットの整数として XLSX に送信されます。

Excel の時間フォーマットでは、ミリ秒までがサポートされます。マイクロ秒を表示する WebFOCUS フォーマットの場合、その値は Excel に送信されますが、ワークシート内でミリ秒にまで切り捨てられます。

下表は、日付時間値がどのように表示されるかを示しています。

WebFOCUS フォーマット	XLSX 表示	XLSX 値
HYYMDm	2010/05/06 16:17:01.993	5/6/2010 4:17:02 PM

WebFOCUS フォーマット	XLSX 表示	XLSX 値
HDMTYYm	06 May 2010 16:17:01.993	5/6/2010 4:17:02 PM
HdMTYYBS	6 May 2010 16:17:01	5/6/2010 4:17:01 PM
HdMTYYBs	6 May 2010 16:17:01.993	5/6/2010 4:17:02 PM

XLSX レポート出力での Excel 数式の生成

XLSX FORMULA を使用して表形式リクエストを実行し、レポート出力を表示または保存すると、ワークシートに Excel 数式が挿入されます。この数式は、静的な数値ではなく、任意のタイプの集計演算 (例、列合計、行合計、中間合計、一時項目 (COMPUTE)) を実行して結果を表示する式です。一時項目 (COMPUTE) の数式は、WebFOCUS 数式の内部形式を Excel 式に変換することにより生成されます。XLSX FORMULA フォーマットで保存したワークシートはインタラクティブ機能を備えており、データを自由に追加、変更し、その結果を即座に反映させることができます。

数式および値の理解

XLSX FORMULA フォーマットでは、WebFOCUS TABLE コマンド (ROW-TOTAL、COLUMN-TOTAL、SUB-TOTAL、SUBTOTAL、SUMMARIZE) で値が集計される場合、および関数で値が計算される場合、値の代わりに数式が生成されます。

- DEFINE コマンドで作成される一時項目 (DEFINE) の場合、数式ではなく、定数値が常に生成されます。
- COMPUTE コマンドで作成される一時項目 (COMPUTE) の場合、数式が生成されます。ただし、COMPUTE が単一変数の場合は例外です。この場合、数式ではなく、定数値が挿入されます。
- COMPUTE または RECOMPUTE コマンドで生成された一時項目 (COMPUTE) がレポートに含まれている場合、セル参照を数式に含めるには、一時項目 (COMPUTE) で参照されているフィールドすべてをレポートに表示する必要があります。参照されているフィールドがブックに表示されていない場合、数式にはセル参照ではなく、データ値が挿入されます。また、Excel に転送される情報に基づいて値を正確に計算できない場

合は、数式の代わりに値が使用されます。たとえば、WebFOCUS で LAST 関数を使用した場合、この関数を正しく変換して Excel に転送することはできません。この場合、セル参照ではなく、LAST 値が式に使用されます。

レポートキャンバスまたは FML 言語で作成される財務レポートでは、XLSX FORMULA はサポートされません。

FORMAT XLSX FORMULA の変換サポート

ここでは、FORMAT XLSX FORMULA の変換サポートについて説明します。サポートされていない WebFOCUS 機能を使用すると、不正確な結果になる場合があります。

- 標準の演算子はすべてサポートされます。これには、算術演算子、関係演算子、文字列演算子、IF/THEN/ELSE、論理演算子が含まれます。ただし、列参照番号はサポートされません。

  IS-PRESENT、IS-MISSING、IS-FROM、FROM、NOT-FROM、IS-MORE-THAN、IS-LESS-THAN、CONTAINS、OMITS 演算子はサポートされません。

  条件式 (IF-THEN-ELSE) および論理式では、AND および OR 論理演算子はサポートされません。

- 次の関数がサポートされています。

  ABS、ARGLEN、ATODBL、BYTVAL、CHARGET、CTRAN、DMOD、DOWK、DOWK、DOWKL、EXP、FMOD、HEXBYT、HHMMSS、IMOD、LCWORD、LOCASE、LOG、MAX、MIN、OVRLAY、POSIT、RDUNIF、SQRT、SUBSTR、TODAY、UPCASEEDIT 関数は、フォーマットの変換でサポートされます (1 つの引数型)。この関数は、文字列の編集ではサポートされません。

  CTRFLD、LJUST、RJUST 関数は、Excel 列内のデータを位置揃えする目的で使用することはお勧めしません。Excel でプロポーショナルフォントを使用する場合は、スタイルシートの JUSTIFY 属性を使用する方法が適しています。

  関数の引数に小数値が使用される場合は注意が必要です (BYTVAL、CTRAN、HEXBYT)。オペレーティングシステム環境 (EBCDIC または ASCII) に応じて、結果が異なる場合があります。

- 次の WebFOCUS コマンドおよび句では、XLSX FORMULA はサポートされません。
  - DEFINE
  - OVER
  - FOR

    - NOPRINT
    - 複数の表示コマンド (PRINT、LIST、SUM、COUNT)
    - SEQUENCE スタイルシート属性
    - RECAP
    - SET HIDENULLACRS
    - SET SUBTOTALS = ABOVE
    - LAST
- BYDISPLAY ON 設定を使用して、再計算する行のすべてにソートフィールド値を表示することをお勧めします。
- 式の長さが 1024 バイトを超える場合、WebFOCUS は、数式の代わりに値自体をセルに配置します。
- 条件スタイルは元のレポートの値を基準に設定されます。ワークシートの値を変更して数式を再計算した場合でも、更新後の情報はスタイル設定に反映されません。

FORMAT XLSX FORMULA としてのレポート保存

Excel 数式をブックで使用するには、リクエストに次の構文を追加します。

```
ON TABLE {PCHOLD|HOLD} FORMAT XLSX FORMULA
```

説明

PCHOLD

XLSX ブックに出力結果を表示します。

HOLD

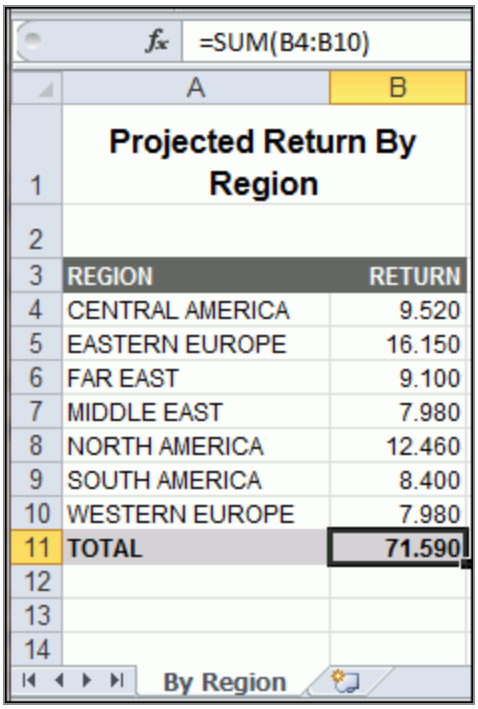
出力結果を保存し、Excel ワークシートとして再利用できるようにします。詳細は、レポート出力の保存と再利用を参照してください。

列合計の Excel 数式の生成

次の例では、XLSX FORMULA フォーマットの使用時に、リクエストの列合計を Excel の数式に変換する方法を示します。列合計 (TYPE=GRANDTOTAL) のフォーマットが Excel ブックで保持されます。レポートで合計を選択すると、「=SUM(B4:B10)」という式が数式バーに表示され、列合計はセル範囲の集計として表されます。

```
TABLE FILE SHORT
HEADING
"Projected Return By Region"
" "
SUM PROJECTED_RETURN AS 'RETURN'
BY REGION AS 'REGION'
ON TABLE COLUMN-TOTAL
ON TABLE PCHOLD FORMAT XLSX FORMULA
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9, TITLETEXT='By Region',$
TYPE=TITLE, BACKCOLOR=RGB(102 102 102), COLOR=RGB(255 255 255),$
TYPE=HEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=GRANDTOTAL, BACKCOLOR=RGB(210 210 210), STYLE=BOLD,$
END
```

出力結果は次のとおりです。



WebFOCUS は、任意の合計 (中間合計、行合計、列合計) を Excel の数式に変換することができます。

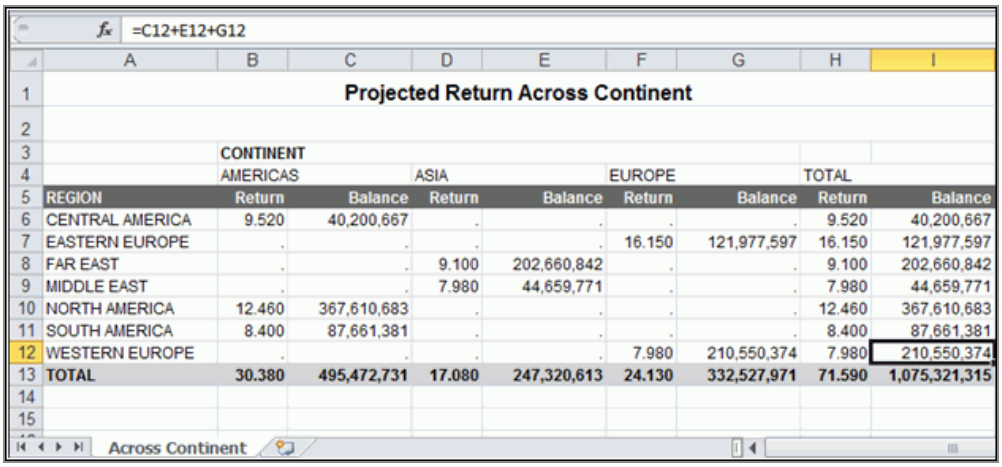
行合計の Excel 数式の生成

次のリクエストは、すべての大陸の収益合計と残高合計を計算します。行合計はセル範囲の集計として表示されます。

```
TABLE FILE SHORT
HEADING
"Projected Return Across Continent"
" "
SUM PROJECTED_RETURN AS 'Return' AND BALANCE AS 'Balance'
ACROSS CONTINENT AS 'CONTINENT'
BY REGION AS 'REGION'
ON CONTINENT ROW-TOTAL AS 'TOTAL'
ON TABLE COLUMN-TOTAL AS 'TOTAL'
ON TABLE PCHOLD FORMAT XLSX FORMULA
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9, TITLETEXT='Across Continent',$
TYPE=TITLE, BACKCOLOR=RGB(102 102 102), COLOR=RGB(255 255 255),$
TYPE=HEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=ACROSSTITLE, STYLE=BOLD,$
```

```
TYPE=GRANDTOTAL, BACKCOLOR=RGB(210 210 210), STYLE=BOLD,$
END
```

下図の出力結果では、セルの行合計を計算する数式 (I12=C12+E12+G12) が数式バーに表示されています。



=C12+E12+G12

	A	B	C	D	E	F	G	H	I
1	Projected Return Across Continent								
2									
3		CONTINENT							
4		AMERICAS		ASIA		EUROPE		TOTAL	
5	REGION	Return	Balance	Return	Balance	Return	Balance	Return	Balance
6	CENTRAL AMERICA	9.520	40,200,667	.	.	.	.	9.520	40,200,667
7	EASTERN EUROPE	.	.	.	.	16.150	121,977,597	16.150	121,977,597
8	FAR EAST	.	.	9.100	202,660,842	.	.	9.100	202,660,842
9	MIDDLE EAST	.	.	7.980	44,659,771	.	.	7.980	44,659,771
10	NORTH AMERICA	12.460	367,610,683	.	.	.	.	12.460	367,610,683
11	SOUTH AMERICA	8.400	87,661,381	.	.	.	.	8.400	87,661,381
12	WESTERN EUROPE	.	.	.	.	7.980	210,550,374	7.980	210,550,374
13	TOTAL	30.380	495,472,731	17.080	247,320,613	24.130	332,527,971	71.590	1,075,321,315
14									
15									

Across Continent

一時項目 (COMPUTE) の Excel 数式の生成

次のリクエストは、経費 (BUDDOLLARS) および売上 (DOLLARS) フィールドの合計を計算し、売上から経費を減算して利益 (PROFIT) フィールドの値を計算します。

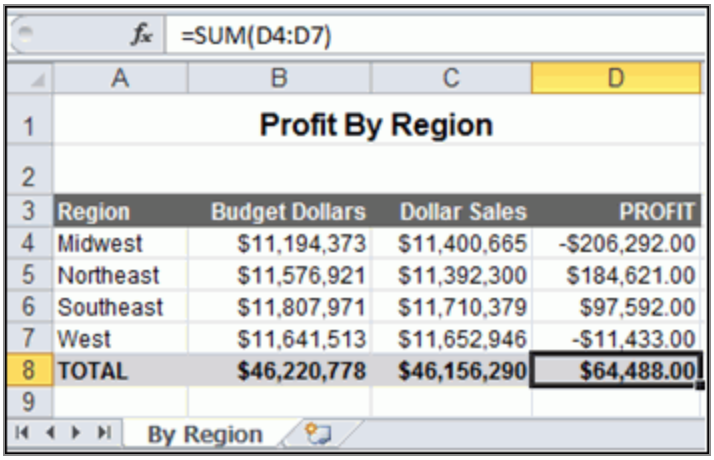
一時項目 (COMPUTE) の数式は、WebFOCUS の式 (PROFIT/D12.2MC = BUDDOLLARS - DOLLARS;) の内部形式を Excel の数式に変換することで生成されます。この例では、数式はセル B8、C8、D8 に表示されます。

セル参照を使用した有効な数式を作成するには、演算で参照されているすべてのフィールドがレポートに表示されている必要があります。すべてのフィールドが表示されていないと、レポートに存在しない値に基づいて数式が作成される場合があります。演算で使用されるフィールドがレポートに存在せず、演算の後に再計算が存在する場合、その再計算によって誤った数式が作成されます。

```
TABLE FILE GGSALES
ON TABLE SET PAGE-NUM OFF
SUM BUDDOLLARS/I8MC AND DOLLARS/I8MC
COMPUTE PROFIT/D12.2MC = BUDDOLLARS - DOLLARS;
BY REGION
HEADING
"Profit By Region"
```

```
" "
ON TABLE COLUMN-TOTAL
ON TABLE PCHOLD FORMAT XLSX FORMULA
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9, TITLETEXT='By Region',$
TYPE=TITLE, BACKCOLOR=RGB(102 102 102), COLOR=RGB(255 255 255),$
TYPE=HEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=GRANDTOTAL, BACKCOLOR=RGB(210 210 210), STYLE=BOLD,$
END
```

次の出力は、PROFIT の列合計を計算する数式を示しています (D8=SUM(D4:D7))。



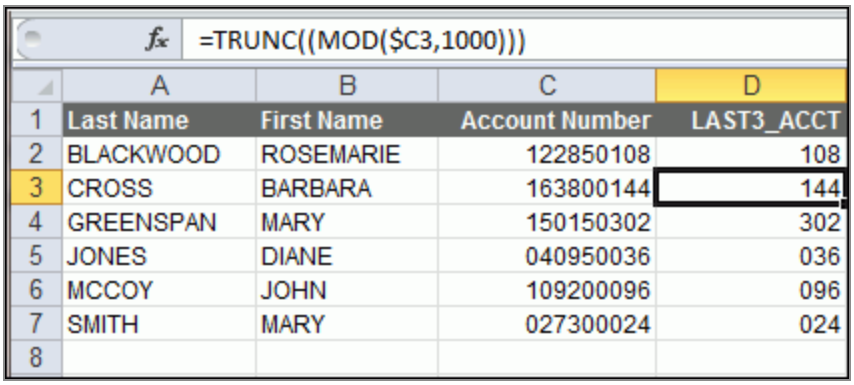
WebFOCUS 関数の Excel 数式の生成

次の例は、関数が Excel レポートにどのように変換されるかを示しています。IMOD 関数は ACCTNUMBER を 1000 で除算し、剰余を LAST3_ACCT に返します。この変換に該当する Excel 数式は「=TRUNC((MOD($C3,(1000))))」です。TRUNC は、数式から返された結果が整数フィールドに配置される場合に使用され、これにより小数点以下の桁数が切り捨てられます。

```
TABLE FILE EMPLOYEE
PRINT ACCTNUMBER AS 'Account Number'
COMPUTE LAST3_ACCT/I3L = IMOD(ACCTNUMBER, 1000, LAST3_ACCT);
BY LAST_NAME AS 'Last Name'
BY FIRST_NAME AS 'First Name'
WHERE (ACCTNUMBER NE 000000000) AND (DEPARTMENT EQ 'MIS');
ON TABLE PCHOLD FORMAT XLSX FORMULA
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9,$
```

```
TYPE=TITLE, BACKCOLOR=RGB(102 102 102), COLOR=RGB(255 255 255), STYLE=BOLD,$
END
```

出力結果は次のとおりです。



	A	B	C	D
1	Last Name	First Name	Account Number	LAST3_ACCT
2	BLACKWOOD	ROSEMARIE	122850108	108
3	CROSS	BARBARA	163800144	144
4	GREENSPAN	MARY	150150302	302
5	JONES	DIANE	040950036	036
6	MCCOY	JOHN	109200096	096
7	SMITH	MARY	027300024	024
8				

再計算値を含む数式の生成

COMPUTE または RECOMPUTE コマンドで生成された一時項目 (COMPUTE) がレポートに含まれている場合、セル参照を数式に含めるには、一時項目 (COMPUTE) で参照されているフィールドすべてをレポートに表示する必要があります。参照されているフィールドがブックに表示されていない場合、数式にはセル参照ではなく、データ値が挿入されます。この場合、ソート区切りの最後の詳細レコードの値が、再計算に誤って使用される場合があります。

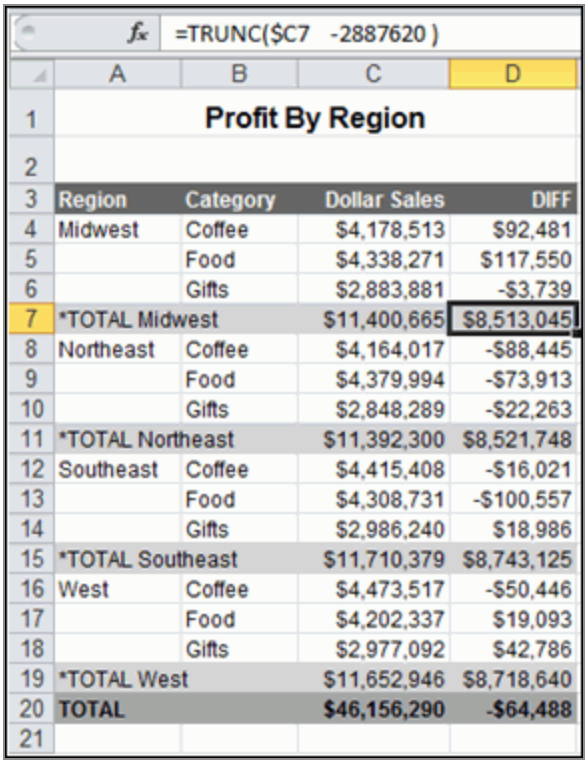
再計算値を含む数式の生成

次のリクエストは、売上高 (DOLLARS) から販売目標額 (BUDDOLLARS) を減算することで、その差 (DIFF) を計算します。式に使用される BUDDOLLARS フィールドは、SUM コマンドには含まれていません。DIFF の値は、REGION レベルで再計算されます。

```
TABLE FILE GGSALES
HEADING
"Profit By Region"
" "
SUM DOLLARS/I8CM
COMPUTE DIFF/I8CM=DOLLARS - BUDDOLLARS;
BY REGION
```

```
BY CATEGORY
ON REGION RECOMPUTE
ON TABLE PCHOLD FORMAT XLSX FORMULA
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9, TITLETEXT='By Region',$
TYPE=TITLE, BACKCOLOR=RGB(102 102 102), COLOR=RGB(255 255 255),$
TYPE=HEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=SUBTOTAL, BACKCOLOR=RGB(210 210 210),$
TYPE=GRANDTOTAL, BACKCOLOR=RGB(166 166 166), STYLE=BOLD,$
END
```

出力結果では、ワークシートに表示されていないデータ値が数式によって減算されています。セル参照が存在しないため、このデータ値には、現在のハードコード値から取得された BUDDOLLARS 値が使用されています。

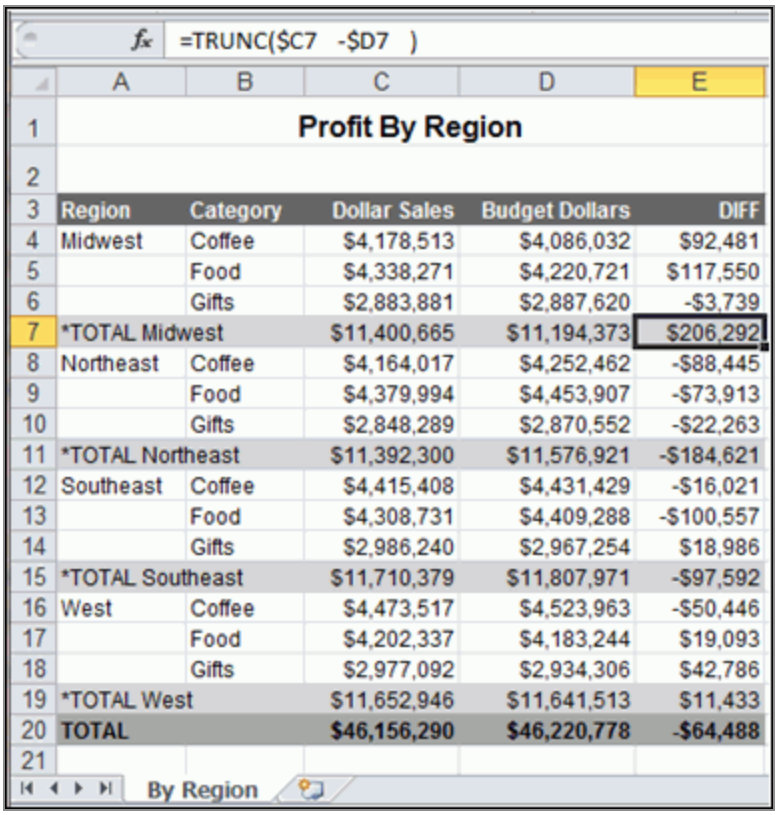


	A	B	C	D
1	Profit By Region			
2				
3	Region	Category	Dollar Sales	DIFF
4	Midwest	Coffee	$4,178,513	$92,481
5		Food	$4,338,271	$117,550
6		Gifts	$2,883,881	-$3,739
7	*TOTAL Midwest		$11,400,665	$8,513,045
8	Northeast	Coffee	$4,164,017	-$88,445
9		Food	$4,379,994	-$73,913
10		Gifts	$2,848,289	-$22,263
11	*TOTAL Northeast		$11,392,300	$8,521,748
12	Southeast	Coffee	$4,415,408	-$16,021
13		Food	$4,308,731	-$100,557
14		Gifts	$2,986,240	$18,986
15	*TOTAL Southeast		$11,710,379	$8,743,125
16	West	Coffee	$4,473,517	-$50,446
17		Food	$4,202,337	$19,093
18		Gifts	$2,977,092	$42,786
19	*TOTAL West		$11,652,946	$8,718,640
20	**TOTAL**		**$46,156,290**	**-$64,488**
21				

上記のリクエストに BUDDOLLARS フィールドを追加すると、数式が正しく再計算されます。

```
SUM DOLLARS/I8MC BUDDOLLARS/I8MC
```

新しい SUM コマンドで生成される数式には、計算に使用される両方のフィールドのセル参照が含まれています。



=TRUNC($C7 -$D7)

	A	B	C	D	E
1	Profit By Region				
2					
3	Region	Category	Dollar Sales	Budget Dollars	DIFF
4	Midwest	Coffee	$4,178,513	$4,086,032	$92,481
5		Food	$4,338,271	$4,220,721	$117,550
6		Gifts	$2,883,881	$2,887,620	-$3,739
7	*TOTAL Midwest		$11,400,665	$11,194,373	$206,292
8	Northeast	Coffee	$4,164,017	$4,252,462	-$88,445
9		Food	$4,379,994	$4,453,907	-$73,913
10		Gifts	$2,848,289	$2,870,552	-$22,263
11	*TOTAL Northeast		$11,392,300	$11,576,921	-$184,621
12	Southeast	Coffee	$4,415,408	$4,431,429	-$16,021
13		Food	$4,308,731	$4,409,288	-$100,557
14		Gifts	$2,986,240	$2,967,254	$18,986
15	*TOTAL Southeast		$11,710,379	$11,807,971	-$97,592
16	West	Coffee	$4,473,517	$4,523,963	-$50,446
17		Food	$4,202,337	$4,183,244	$19,093
18		Gifts	$2,977,092	$2,934,306	$42,786
19	*TOTAL West		$11,652,946	$11,641,513	$11,433
20	TOTAL		$46,156,290	$46,220,778	-$64,488
21					

By Region

XLSX FORMULA と演算接頭語の使用

XLSX FORMULA 出力では、WebFOCUS コマンド (例、SUBTOTAL、RECOMPUTE) で生成される集計行に演算接頭語を使用することができます。演算接頭語に対応する数式が Excel に存在する場合、これらの演算接頭語は同等の Excel 集計式に変換されます。取得データに対して直接使用した演算接頭語の結果は、数式ではなく、値として Excel に渡されます。

下表は、XLSX FORMULA の集計行でサポートされる演算接頭語、および生成されるワークシートに配置される同等の Excel 数式を示しています。

演算接頭語	同等の Excel 数式
SUM.	=SUM()
AVE.	=AVERAGE()
CNT.	=COUNT()

演算接頭語	同等の Excel 数式
MIN.	=MIN()
MAX.	=MAX()

次の演算接頭語を XLSX FORMULA の集計行に使用した場合、数式には変換されません。

- ASQ.
- FST.
- LST.

注意：

- 取得するデータに対して直接指定したフィールドに演算接頭語を使用する場合は、演算接頭語と演算対象のフィールドとの間にブランクは含めません。

  たとえば、次の集計表示コマンドでは、AVE. 演算接頭語が DOLLARS フィールドに対して演算を実行します。

  ```
  SUM AVE.DOLLARS
  ```

- 集計行に演算接頭語を使用する場合は、演算接頭語と演算対象の集計フィールドとの間にブランクを含める必要があります。

  演算接頭語は、REGION ソート区切りが挿入された位置で DOLLARS フィールドに対して演算を実行します。演算接頭語とフィールド名との間に、次のようにブランクを含める必要があります。

  ```
  ON REGION RECOMPUTE MAX.DOLLARS
  ```

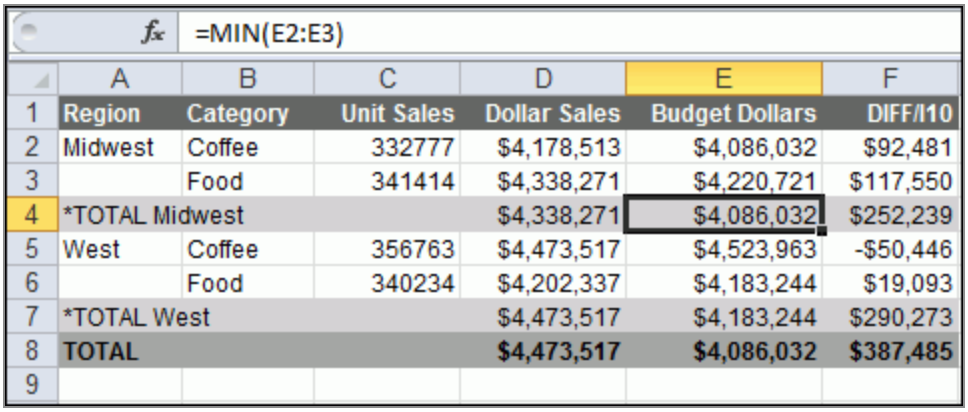
FORMAT XLSX FORMULA での集計演算接頭語の使用

次のリクエストは、GGSALES データソースに対して実行され、REGION ソートフィールドの RECOMPUTE コマンドにより、集計された DOLLARS フィールドの最大値と、集計された BUDDOLLARS フィールドの最小値が計算されます。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS/I8MC BUDDOLLARS/I8MC
AND COMPUTE DIFF/I8MC= DOLLARS-BUDDOLLARS;
BY REGION
BY CATEGORY
WHERE CATEGORY EQ 'Food' OR 'Coffee'
WHERE REGION EQ 'West' OR 'Midwest'
ON REGION RECOMPUTE MAX.DOLLARS MIN.BUDDOLLARS DIFF
ON TABLE PCHOLD FORMAT XLSX FORMULA
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9,$
TYPE=TITLE, BACKCOLOR=RGB(102 102 102), COLOR=RGB(255 255 255),$
TYPE=SUBTOTAL, BACKCOLOR=RGB(210 210 210),$
TYPE=GRANDTOTAL, BACKCOLOR=RGB(166 166 166), STYLE=BOLD,$
END
```

下図の出力結果では、Midwest 地域の BUDDOLLARS の再計算値を表すセルには、次の数式が生成されています。

```
=MIN(E2:E3)
```



fx =MIN(E2:E3)

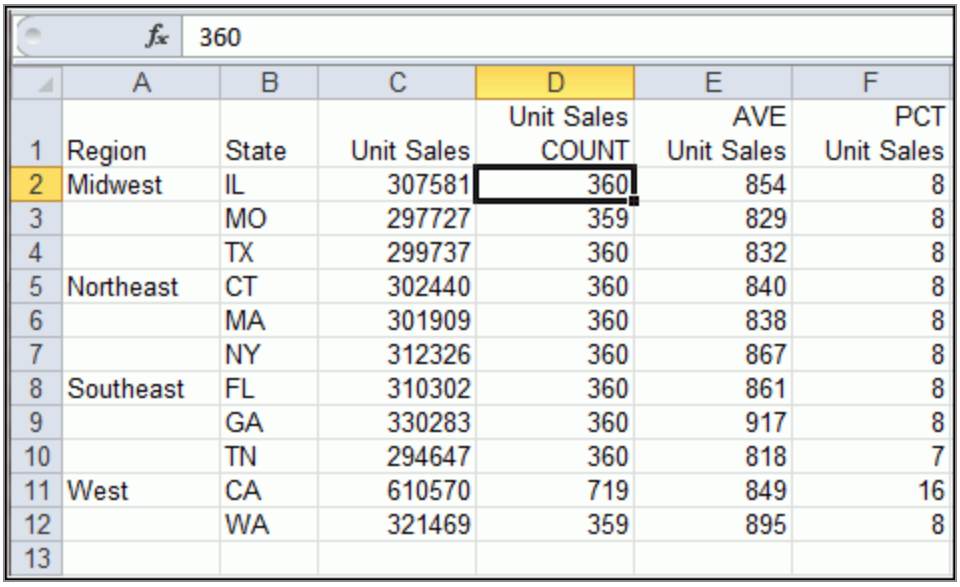
	A	B	C	D	E	F
1	Region	Category	Unit Sales	Dollar Sales	Budget Dollars	DIFF/I10
2	Midwest	Coffee	332777	$4,178,513	$4,086,032	$92,481
3		Food	341414	$4,338,271	$4,220,721	$117,550
4	*TOTAL Midwest			$4,338,271	$4,086,032	$252,239
5	West	Coffee	356763	$4,473,517	$4,523,963	-$50,446
6		Food	340234	$4,202,337	$4,183,244	$19,093
7	*TOTAL West			$4,473,517	$4,183,244	$290,273
8	TOTAL			$4,473,517	$4,086,032	$387,485
9						

FORMAT XLSX FORMULA での表示コマンドと演算接頭語の使用

次のリクエストは、GGSALES データソースに対して実行され、SUM 表示コマンドで CNT.、AVE.、PCT. 演算接頭語が使用されています。

```
TABLE FILE GGSALES
SUM UNITS
CNT.UNITS
AVE.UNITS
PCT.UNITS
BY REGION
BY ST
ON TABLE PCHOLD FORMAT XLSX FORMULA
END
```

出力結果では、これらの演算接頭語が数式としてではなく、データ値として Excel に渡されています。



360

	A	B	C	D	E	F
1	Region	State	Unit Sales	Unit Sales COUNT	AVE Unit Sales	PCT Unit Sales
2	Midwest	IL	307581	360	854	8
3		MO	297727	359	829	8
4		TX	299737	360	832	8
5	Northeast	CT	302440	360	840	8
6		MA	301909	360	838	8
7		NY	312326	360	867	8
8	Southeast	FL	310302	360	861	8
9		GA	330283	360	917	8
10		TN	294647	360	818	7
11	West	CA	610570	719	849	16
12		WA	321469	359	895	8
13						

数式での NODATA の使用

完全な Excel 機能をサポートするには、数式で参照されるセルに有効な数値のみが配置されている必要があります。

演算では NULL 値 (NODATA='') がサポートされます。数式に使用されるセルにデフォルトの NODATA 記号 (.) が含まれている場合、数式エラーが発生します。

以下はその例です。

```
SET NODATA=''
TABLE FILE GGSALES
```

```
SUM DOLLARS/D12CM UNITS/D12C AND ROW-TOTAL AND COLUMN-TOTAL
COMPUTE REVENUE/D12CM=DOLLARS*UNITS; AS 'Revenue'
BY LOWEST GGSALES.SALES01.CATEGORY
BY GGSALES.SALES01.PRODUCT
ACROSS REGION
ON TABLE PCHOLD FORMAT XLSX FORMULA
END
------------------------
SET NODATA=''
DEFINE FILE GGSALES
DOLLARMOD/D12CM MISSING ON=IF REGION GT 'V' THEN MISSING ELSE DOLLAR;
END
TABLE FILE GGSALES
SUM DOLLARMOD/D12CM UNITS/D12C AND ROW-TOTAL AND COLUMN-TOTAL
COMPUTE REVENUE/D12CM=DOLLARMOD*UNITS; AS 'Revenue'
BY REGION
BY LOWEST GGSALES.SALES01.CATEGORY
BY GGSALES.SALES01.PRODUCT
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE PCHOLD FORMAT XLSX FORMULA
END
```

XLSX 数式の使用上の注意

- 数式は、単一のワークシート内で定義されます。数式は、複数のワークシートにわたって割り当てられることはありません。
- BYTOC 複合ブックには、数式を含めることができます。主キーの値ごとに個別のワークシート (タブ) が生成され、各シート内の参照に基づいて数式が定義されます。
  BYTOC 複合ブックでは、全体の総合計が別のタブに表示されます。これらの総合計には、各コンポーネントワークシートへの数式参照は含まれません。

XLSX レポート出力での列幅および折り返しの制御

- FORMAT XLSX を使用する際は、Excel ワークシートの列幅およびデータ折り返しを制御することができます。

- データを折り返さずに列幅を変更する場合、正確な列幅を定義するには、SQUEEZE=ON を使用します。指定した列幅よりデータ値が長い場合、データの一部は隠れますが、数式バーにはデータ全体が表示されます。列幅は、ワークシートの生成後、Excel で調節することができます。
- デフォルトの動作では、定義された列幅内ですべてのデータが折り返されます。また、WRAP=ON を使用して正確な列幅を指定することもできます。
- 日付フォーマットのフィールドでは WRAP はサポートされません。

XLSX レポート出力での列幅の設定

```
TYPE=REPORT, [COLUMN=column,] SQUEEZE=value,$
```

説明

column

列を指定します。構文中に COLUMN が含まれていない場合、デフォルトの SQUEEZE 動作はレポート全体に適用されます。

value

次のいずれかです。

ON

列の最大長のデータ値に基づいて列幅を自動的に調整します。これがデフォルトの動作です。

OFF

マスターファイルまたは DEFINE で定義された最大フィールド長に基づいて列幅を設定します。

n

列幅を設定する特定の数値を表します。この値は、UNITS パラメータ (デフォルトはインチ) で指定した測定単位で指定されます。これが XLSX レポートで最もよく使用される SQUEEZE 設定です。この設定は、データの折り返しをオフにします。

注意

- ON TABLE SET SQUEEZE ON コマンドを使用することで、レポート全体に SQUEEZE

を適用することができます。

- SQUEEZE は、OVER 句または TABLEF を使用して作成されたフィールドではサポートされません。

XLSX レポート出力でのデータの折り返し

```
TYPE=REPORT, [COLUMN=column,] WRAP=value,$
```

説明

column

折り返し動作を指定するフィールドを指定します。構文中に COLUMN が含まれていない場合、折り返しはレポート全体に適用されます。

value

次のいずれかです。

ON

データの折り返しをオンにします。デフォルト値は ON です。この設定を使用すると、列幅はクライアント (Excel) が決定します。データは列幅を超えると折り返し、行の高さはデータ折り返し後の新しい高さに拡張されます。

OFF

データの折り返しをオフにします。列内のセルのデータはどれも折り返されません。

n

列幅を設定する特定の数値を表します。この値は、UNITS パラメータ (デフォルトはインチ) で指定した測定単位で指定されます。

この設定にはオン (ON) の意味が含まれています。ただし、データが列幅を超えない限り、列幅は指定幅に設定されます。列幅を超えると、オン (ON) のときと同様、折り返しが発生します。

注意：日付フォーマットのフィールドでは WRAP はサポートされません。

XLSX レポート出力での列幅および折り返しの制御

次の例では、データ列の折り返しのオンとオフ、および特定の列幅を設定します。この例の UNITS はインチに設定されています。インチがデフォルト値です。

```
DEFINE FILE GGSALES
PROFIT/D14.3 = BUDDOLLARS-DOLLARS;
DESCRIPTION/A80 = 'Subtract Total Sales Quota from Reported Sales to
calculate profit.';
END
```

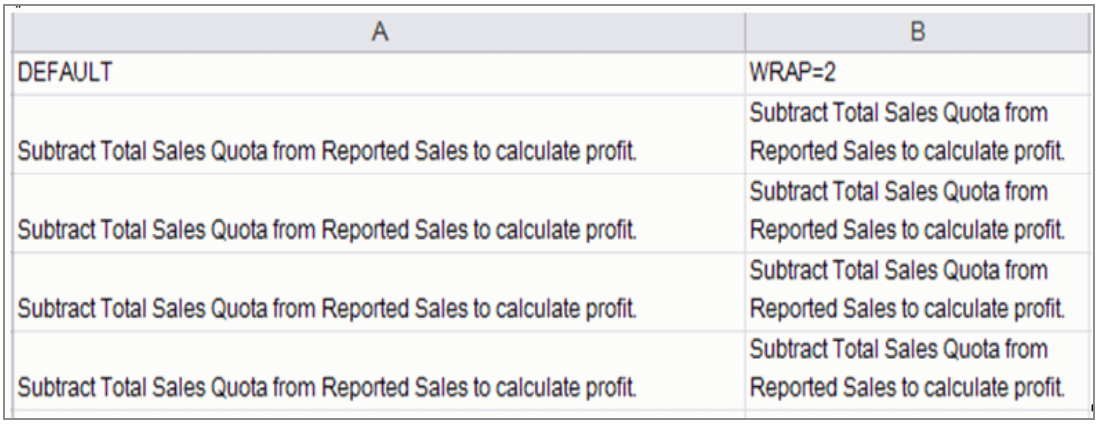
```
TABLE FILE GGSALES
SUM
DESCRIPTION AS 'DEFAULT'
DESCRIPTION AS 'WRAP = 2'
DESCRIPTION AS 'WRAP = OFF'
DESCRIPTION AS 'SQUEEZE = 1.5'
PROFIT
BY REGION NOPRINT
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, COLUMN=DESCRIPTION(2), WRAP=2,$
TYPE=REPORT, COLUMN=DESCRIPTION(3), WRAP=OFF,$
TYPE=REPORT, COLUMN=DESCRIPTION(4), SQUEEZE=1.5,$
END
```

説明

1. 「DEFAULT」というタイトルのフィールドには、デフォルト設定の列幅と折り返し動作を使用します。
2. 「WRAP=2」というタイトルのフィールドでは、列幅を 2 インチ、データの折り返しをオンに設定します。
3. 「WRAP=OFF」というタイトルのフィールドでは、データの折り返しをオフに設定します。
4. 「SQUEEZE=1.5」というタイトルのフィールドでは、列幅を 1.5 インチ、データの折り返しをオフに設定します。

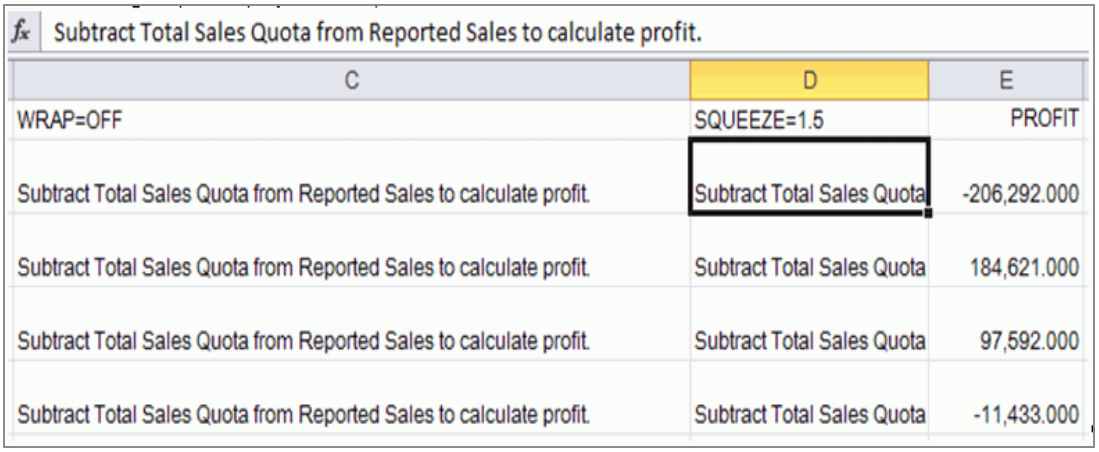
出力結果は 2 ページにわたって表示されるため、下図では各ページを 2 つのイメージで示しています。

下図の XLSX 出力結果は、[DEFAULT] 列と [WRAP=2] 列の動作を示しています。



A	B
DEFAULT	WRAP=2
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota from Reported Sales to calculate profit.
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota from Reported Sales to calculate profit.
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota from Reported Sales to calculate profit.
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota from Reported Sales to calculate profit.

下図の XLSX 出力結果は、[WRAP=OFF] 列と [SQUEEZE=1.5] 列の動作を示しています。



fx Subtract Total Sales Quota from Reported Sales to calculate profit.

C	D	E
WRAP=OFF	SQUEEZE=1.5	PROFIT
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota	-206,292.000
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota	184,621.000
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota	97,592.000
Subtract Total Sales Quota from Reported Sales to calculate profit.	Subtract Total Sales Quota	-11,433.000

XLSX レポートの列の固定

XLSX レポート出力で、列を固定することができます。これにより、右側にスクロールしてワークシートのデータを表示する際に、指定した列が画面上に保持されるため、コンテキストが失われることがありません。

XLSX レポートでの列の固定

XLSX レポートで固定する列を設定するには、次のスタイルシート構文を使用します。

```
TYPE=REPORT, FREEZE-COLUMN={Ncolumn|fieldname} ,$
```

説明

column

固定を開始する列番号を指定します。

fieldname

固定を開始する列のフィールド名を識別します。

列の固定使用上の注意

- レポートでの列の固定は、XLSX フォーマットでのみサポートされます。
- FREEZE-COLUMN 属性には、次のことが適用されます。
  - ブック内の個別のワークシートに適用されます。
  - ソートフィールドに基づく BYTOC を使用してブック内に複数のワークシートを生成した場合は、すべてのワークシートに適用されます。
  - HFREEZE = ON により行および列の両方を固定している場合でも、同時に使用することができます。

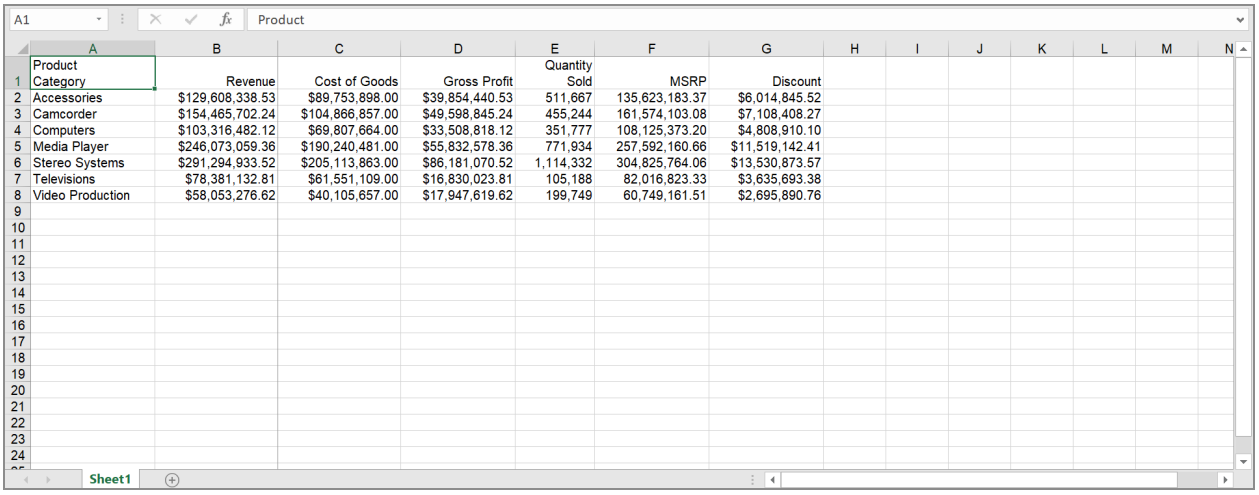
XLSX レポートの列の固定

次のリクエストは、WF_RETAIL データソースに対して実行され、FREEZE-COLUMN スタイルシート属性を使用して、最初の 2 列の製品カテゴリと収益を固定します。

```
TABLE FILE WF_RETAIL
SUM
     REVENUE_US
     COGS_US
```

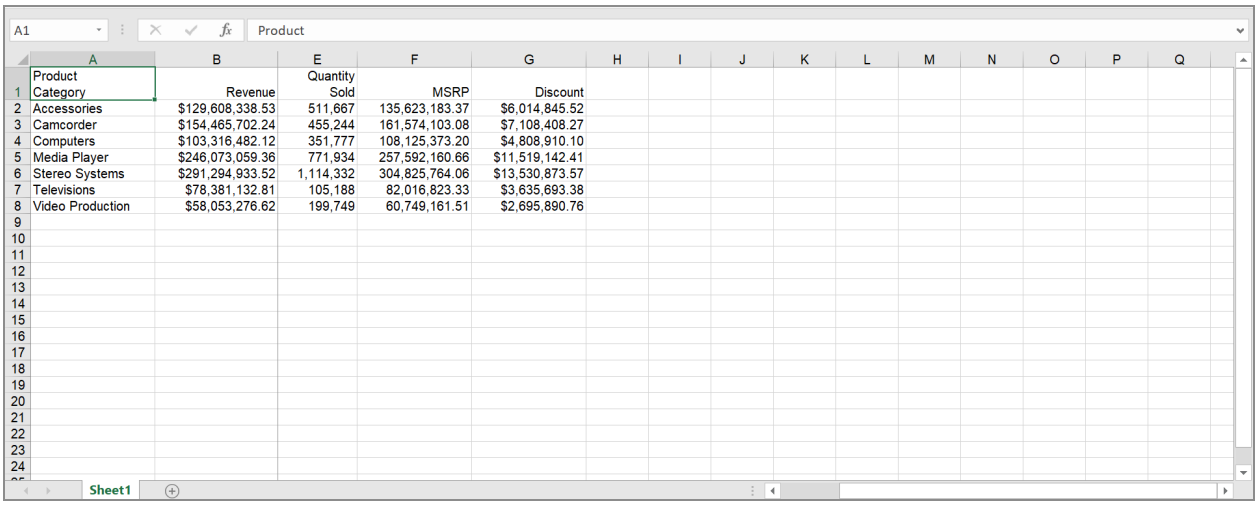
```
      GROSS_PROFIT_US
      QUANTITY_SOLD
      MSRP_US
      DISCOUNT_US
BY PRODUCT_CATEGORY
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE SET ASNAMES ON
ON TABLE SET BYDISPLAY ON
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT,
FREEZE-COLUMN=N2,
$
ENDSTYLE
END
```

出力結果は次のとおりです。



	Product Category	Revenue	Cost of Goods	Gross Profit	Quantity Sold	MSRP	Discount
2	Accessories	$129,608,338.53	$89,753,898.00	$39,854,440.53	511,667	135,623,183.37	$6,014,845.52
3	Camcorder	$154,465,702.24	$104,866,857.00	$49,598,845.24	455,244	161,574,103.08	$7,108,408.27
4	Computers	$103,316,482.12	$69,807,664.00	$33,508,818.12	351,777	108,125,373.20	$4,808,910.10
5	Media Player	$246,073,059.36	$190,240,481.00	$55,832,578.36	771,934	257,592,160.66	$11,519,142.41
6	Stereo Systems	$291,294,933.52	$205,113,863.00	$86,181,070.52	1,114,332	304,825,764.06	$13,530,873.57
7	Televisions	$78,381,132.81	$61,551,109.00	$16,830,023.81	105,188	82,016,823.33	$3,635,693.38
8	Video Production	$58,053,276.62	$40,105,657.00	$17,947,619.62	199,749	60,749,161.51	$2,695,890.76

下図のように、2 つの列を右側にスクロールすると、製品カテゴリ列と収益列は固定され、その他の 3 列は左側にスクロールします。



	A	B	E	F	G
1	Product Category	Revenue	Quantity Sold	MSRP	Discount
2	Accessories	$129,608,338.53	511,667	135,623,183.37	$6,014,845.52
3	Camcorder	$154,465,702.24	455,244	161,574,103.08	$7,108,408.27
4	Computers	$103,316,482.12	351,777	108,125,373.20	$4,808,910.10
5	Media Player	$246,073,059.36	771,934	257,592,160.66	$11,519,142.41
6	Stereo Systems	$291,294,933.52	1,114,332	304,825,764.06	$13,530,873.57
7	Televisions	$78,381,132.81	105,188	82,016,823.33	$3,635,693.38
8	Video Production	$58,053,276.62	199,749	60,749,161.51	$2,695,890.76

WebFOCUS の改ページと Excel の改ページとの同期

BY_field PAGE-BREAK 句を使用すると、WebFOCUS の改ページが、Microsoft Excel の改ページと自動的に同期されます。

XLSX レポート出力フォーマットでの WebFOCUS 改ページと Excel 改ページの同期

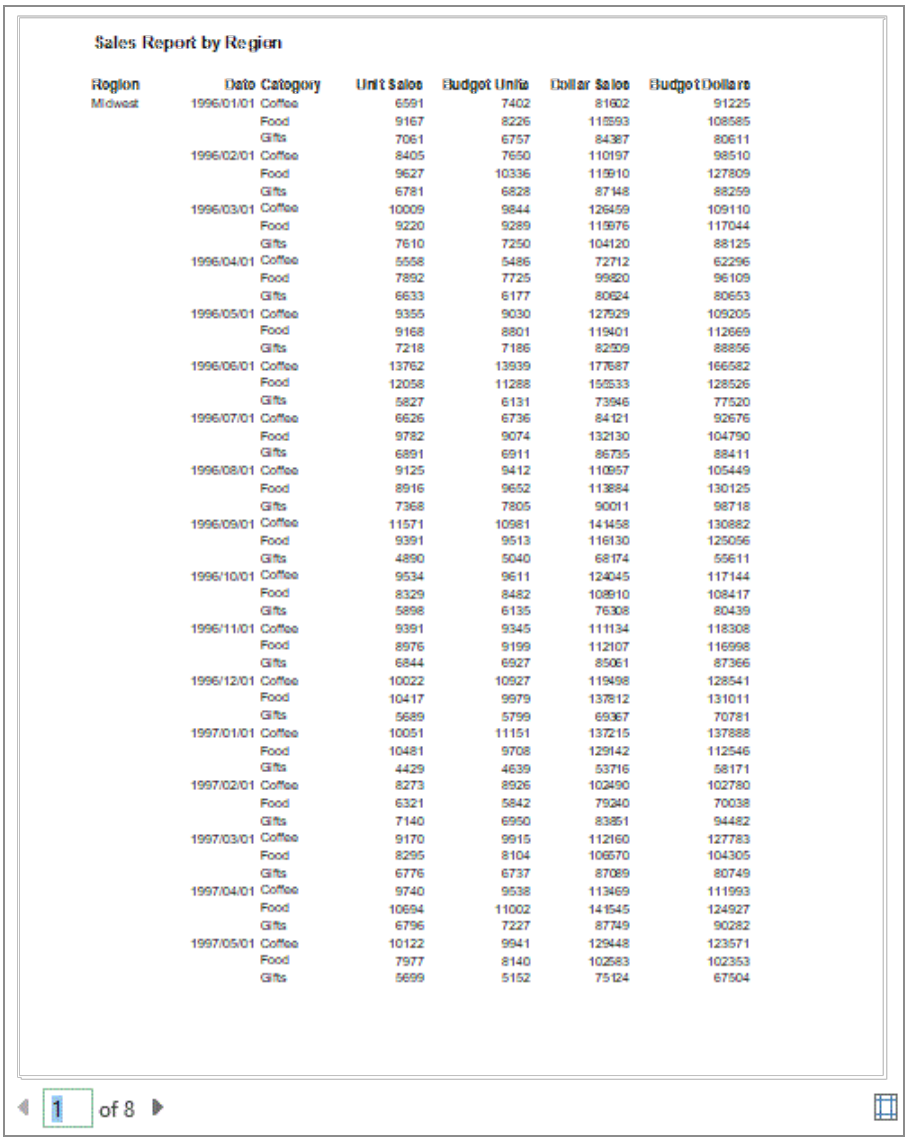
次のリクエストは、BY REGION PAGE-BREAK 句を使用して挿入された WebFOCUS 改ページで、XLSX フォーマットのレポート出力を生成します。

```
TABLE FILE GGSALES
HEADING
"Sales Report by Region"
" "
SUM UNITS BUDUNITS DOLLARS BUDDOLLARS
BY REGION PAGE-BREAK
BY DATE
BY CATEGORY
WHERE CITY LE 'Memphis'
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
```

```
TOPMARGIN=1.25, BOTTOMMARGIN=1, $
TYPE=REPORT, FONT=ARIAL, SIZE=9, $
TYPE=TITLE, STYLE=BOLD, SIZE=10, $
TYPE=HEADING, STYLE=BOLD, SIZE=12, $
ENDSTYLE
END
```

下図は、Midwest Region の出力の 1 ページ目と 2 ページ目を Excel の [印刷プレビュー] で示しています。デフォルト設定の Excel 改ページは、WebFOCUS リクエストで指定した改ページと同期されています。ページ見出しとフィールドタイトルは、BY の値が変更された場合のみ表示されます。

ページ 1 出力

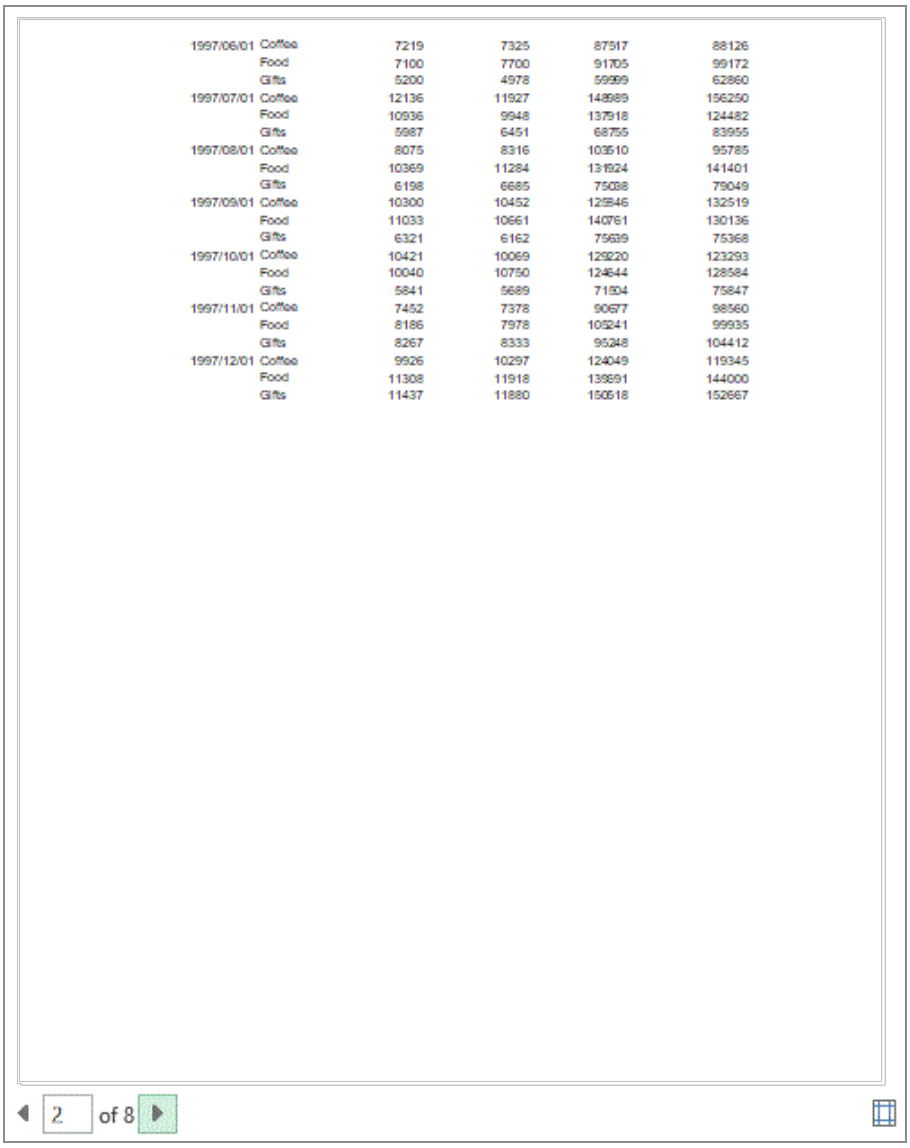


Sales Report by Region

Region	Date	Category	Unit Sales	Budget Units	Dollar Sales	Budget Dollars
Midwest	1996/01/01	Coffee	6591	7402	81602	91225
		Food	9167	8226	115593	108585
		Gifts	7061	6757	84387	80611
	1996/02/01	Coffee	8405	7650	110197	98510
		Food	9627	10336	115910	127809
		Gifts	6781	6828	87148	88259
	1996/03/01	Coffee	10009	9844	126459	109110
		Food	9220	9289	115976	117044
		Gifts	7610	7250	104120	88125
	1996/04/01	Coffee	5558	5486	72712	62296
		Food	7892	7725	99820	96109
		Gifts	6633	6177	80624	80653
	1996/05/01	Coffee	9355	9030	127929	109205
		Food	9168	8801	119401	112669
		Gifts	7218	7186	82509	88856
	1996/06/01	Coffee	13762	13939	177687	166582
		Food	12058	11288	156533	128526
		Gifts	5827	6131	73946	77520
	1996/07/01	Coffee	6626	6736	84121	92676
		Food	9782	9074	132130	104790
		Gifts	6891	6911	86735	88411
	1996/08/01	Coffee	9125	9412	110957	105449
		Food	8916	9652	113884	130125
		Gifts	7368	7805	90011	98718
	1996/09/01	Coffee	11571	10981	141458	130882
		Food	9391	9513	116130	125056
		Gifts	4890	5040	68174	55611
	1996/10/01	Coffee	9534	9611	124045	117144
		Food	8329	8482	108910	108417
		Gifts	5898	6135	76308	80439
	1996/11/01	Coffee	9391	9345	111134	118308
		Food	8976	9199	112107	116998
		Gifts	6844	6927	85061	87366
	1996/12/01	Coffee	10022	10927	119498	128541
		Food	10417	9979	137812	131011
		Gifts	5689	5799	69367	70781
	1997/01/01	Coffee	10051	11151	137215	137888
		Food	10481	9708	129142	112546
		Gifts	4429	4639	53716	58171
	1997/02/01	Coffee	8273	8926	102490	102780
		Food	6321	5842	79240	70038
		Gifts	7140	6950	83851	94482
	1997/03/01	Coffee	9170	9915	112160	127783
		Food	8295	8104	106570	104305
		Gifts	6776	6737	87089	80749
	1997/04/01	Coffee	9740	9538	113469	111993
		Food	10694	11002	141545	124927
		Gifts	6796	7227	87749	90282
	1997/05/01	Coffee	10122	9941	129448	123571
		Food	7977	8140	102583	102353
		Gifts	5699	5152	75124	67504

1 of 8

ページ 2 出力



1997/06/01	Coffee	7219	7325	87517	88126
	Food	7100	7700	91705	99172
	Gifts	5200	4978	59999	62860
1997/07/01	Coffee	12136	11927	148989	156250
	Food	10936	9948	137918	124482
	Gifts	5987	6451	68755	83955
1997/08/01	Coffee	8075	8316	103510	95785
	Food	10369	11284	131924	141401
	Gifts	6198	6685	75038	79049
1997/09/01	Coffee	10300	10452	125846	132519
	Food	11033	10661	140761	130136
	Gifts	6321	6162	75639	75368
1997/10/01	Coffee	10421	10069	129220	123293
	Food	10040	10750	124644	128584
	Gifts	5841	5689	71504	75847
1997/11/01	Coffee	7452	7378	90677	98560
	Food	8186	7978	105241	99935
	Gifts	8267	8333	95248	104412
1997/12/01	Coffee	9926	10297	124049	119345
	Food	11308	11918	139591	144000
	Gifts	11437	11880	150518	152667

各印刷ページにページ見出しとフィールドタイトルを繰り返し表示するには、**BY_field PAGE-BREAK** 句を XLSXPAGETITLES=ON スタイルシート属性と併せて使用します。以下はその例です。

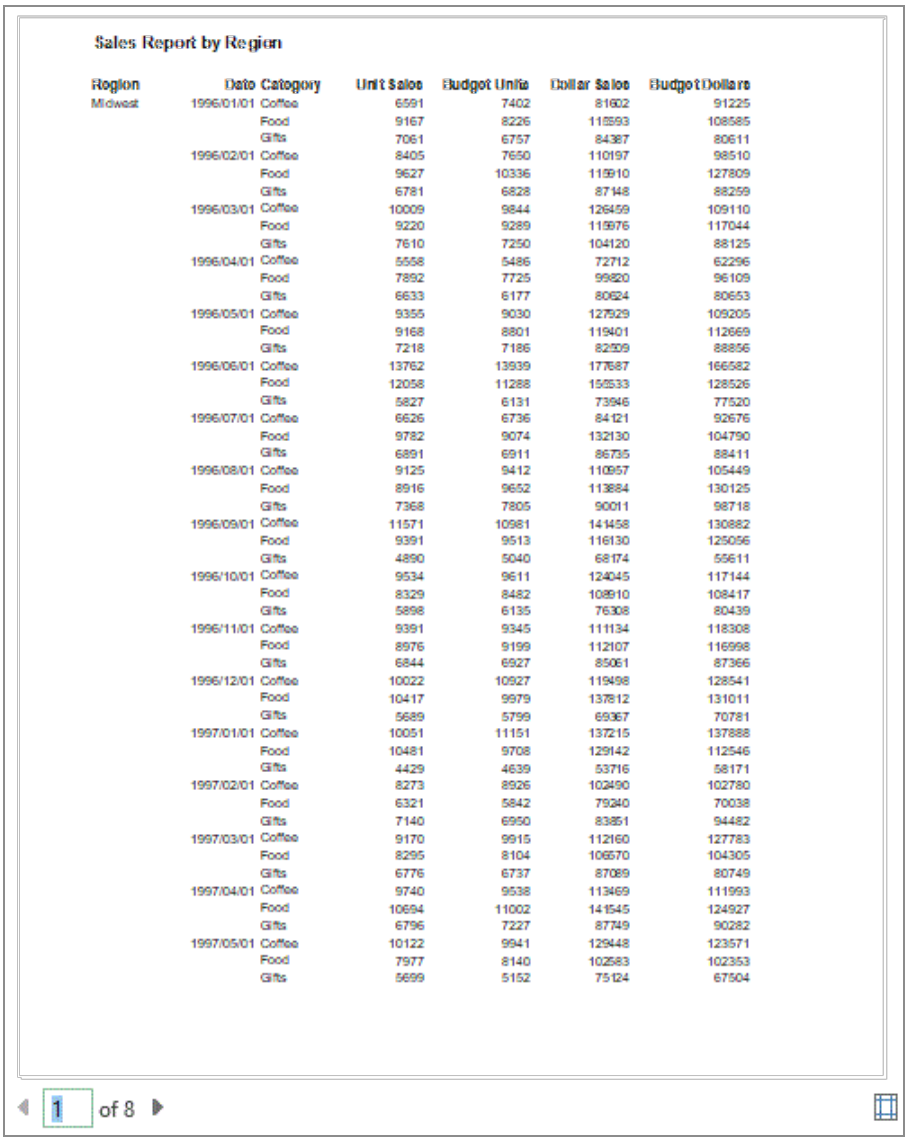
注意：XLSXPAGETITLES を SET コマンドとして使用することもできます。

```
TABLE FILE GGSALES
HEADING
"Sales Report by Region"
" "
SUM UNITS BUDUNITS DOLLARS BUDDOLLARS
BY REGION PAGE-BREAK
BY DATE
```

```
BY CATEGORY
WHERE CITY LE 'Memphis'
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
XLSXPAGETITLES=ON,
TOPMARGIN=1.25, BOTTOMMARGIN=1, $
TYPE=REPORT, FONT=ARIAL, SIZE=9, $
TYPE=TITLE, STYLE=BOLD, SIZE=10, $
TYPE=HEADING, STYLE=BOLD, SIZE=12, $
ENDSTYLE
END
```

下図は、Midwest Region の出力の 1 ページ目と 2 ページ目を Excel の [印刷プレビュー] で示しています。ページ見出しとフィールドタイトルは、各印刷ページに繰り返し表示されています。

ページ 1 出力

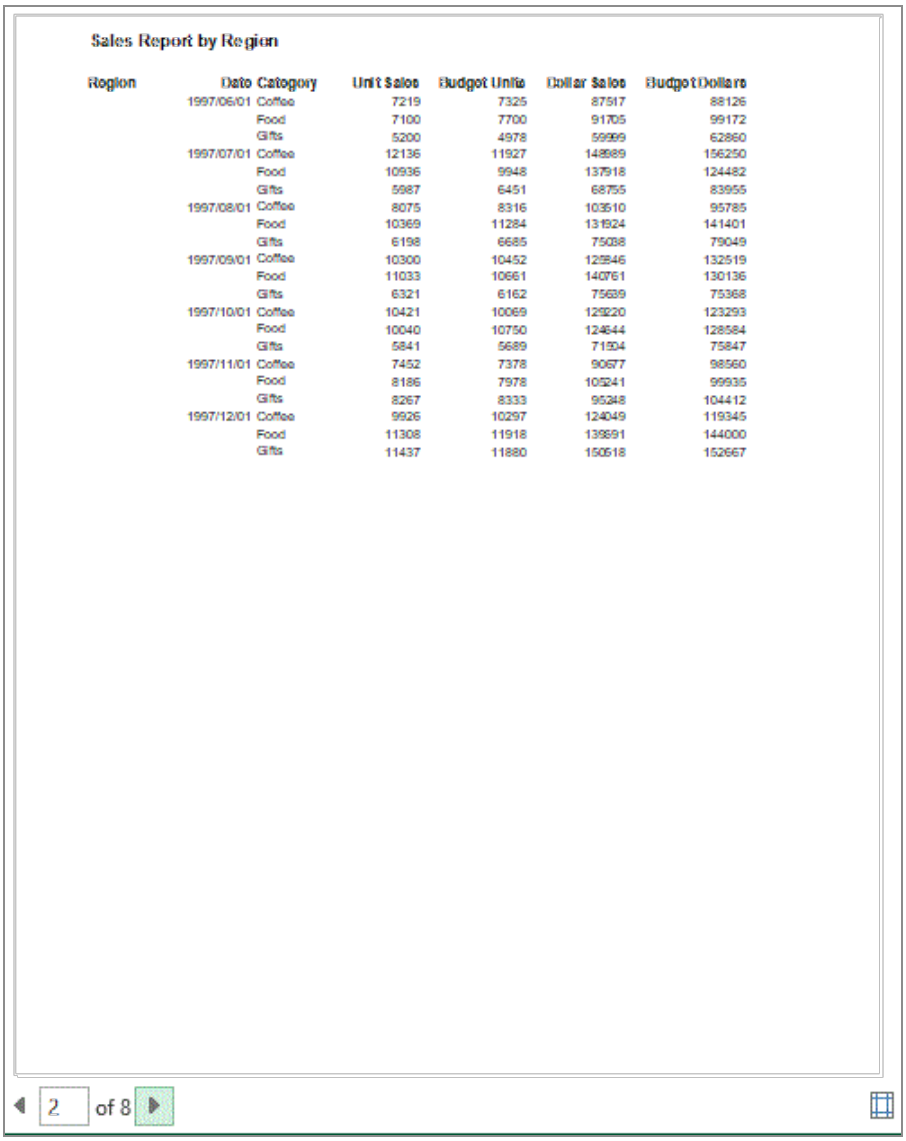


Sales Report by Region

Region	Date	Category	Unit Sales	Budget Units	Dollar Sales	Budget Dollars
Midwest	1996/01/01	Coffee	6591	7402	81602	91225
		Food	9167	8226	115593	108585
		Gifts	7061	6757	84387	80611
	1996/02/01	Coffee	8405	7650	110197	98510
		Food	9627	10336	115910	127809
		Gifts	6781	6828	87148	88259
	1996/03/01	Coffee	10009	9844	126459	109110
		Food	9220	9289	115976	117044
		Gifts	7610	7250	104120	88125
	1996/04/01	Coffee	5558	5486	72712	62296
		Food	7892	7725	99820	96109
		Gifts	6633	6177	80624	80653
	1996/05/01	Coffee	9355	9030	127929	109205
		Food	9168	8801	119401	112669
		Gifts	7218	7186	82509	88856
	1996/06/01	Coffee	13762	13939	177687	166582
		Food	12058	11288	156533	128526
		Gifts	5827	6131	73946	77520
	1996/07/01	Coffee	6626	6736	84121	92676
		Food	9782	9074	132130	104790
		Gifts	6891	6911	86735	88411
	1996/08/01	Coffee	9125	9412	110957	105449
		Food	8916	9652	113884	130125
		Gifts	7368	7805	90011	98718
	1996/09/01	Coffee	11571	10981	141458	130882
		Food	9391	9513	116130	125056
		Gifts	4890	5040	68174	55611
	1996/10/01	Coffee	9534	9611	124045	117144
		Food	8329	8482	108910	108417
		Gifts	5898	6135	76308	80439
	1996/11/01	Coffee	9391	9345	111134	118308
		Food	8976	9199	112107	116998
		Gifts	6844	6927	85061	87366
	1996/12/01	Coffee	10022	10927	119498	128541
		Food	10417	9979	137812	131011
		Gifts	5689	5799	69367	70781
	1997/01/01	Coffee	10051	11151	137215	137888
		Food	10481	9708	129142	112546
		Gifts	4429	4639	53716	58171
	1997/02/01	Coffee	8273	8926	102490	102780
		Food	6321	5842	79240	70038
		Gifts	7140	6950	83851	94482
	1997/03/01	Coffee	9170	9915	112160	127783
		Food	8295	8104	106570	104305
		Gifts	6776	6737	87089	80749
	1997/04/01	Coffee	9740	9538	113469	111993
		Food	10694	11002	141545	124927
		Gifts	6796	7227	87749	90282
	1997/05/01	Coffee	10122	9941	129448	123571
		Food	7977	8140	102583	102353
		Gifts	5699	5152	75124	67504

1 of 8

ページ 2 出力



Sales Report by Region

Region	Date	Category	Unit Sales	Budget Units	Dollar Sales	Budget Dollars
	1997/06/01	Coffee	7219	7325	87517	88126
		Food	7100	7700	91705	99172
		Gifts	5200	4978	59999	62860
	1997/07/01	Coffee	12136	11927	148989	156250
		Food	10936	9948	137918	124482
		Gifts	5987	6451	68755	83955
	1997/08/01	Coffee	8075	8316	103510	95785
		Food	10369	11284	131924	141401
		Gifts	6198	6685	75038	79049
	1997/09/01	Coffee	10300	10452	128846	132519
		Food	11033	10661	140761	130136
		Gifts	6321	6162	75639	75368
	1997/10/01	Coffee	10421	10069	129220	123293
		Food	10040	10750	124644	128584
		Gifts	5841	5689	71504	75847
	1997/11/01	Coffee	7452	7378	90677	98560
		Food	8186	7978	105241	99935
		Gifts	8267	8333	95348	104412
	1997/12/01	Coffee	9926	10297	124049	119345
		Food	11308	11918	139891	144000
		Gifts	11437	11880	150518	152667

注意：レポートに OVER または ACROSS 句が含まれている場合は、**BY_field** PAGE-BREAK を XLSXPAGETITLES=ALL スタイルシートまたは SET コマンドと併せて使用します。

レポート出力での先頭および内部ブランクの保持

SHOWBLANKS コマンドを使用して、XLSX レポートのデータセルおよび見出し内の先頭ブランクを保持することができます。XLSX では、内部ブランクは常に保持されますが、データフィールドの先頭および末尾のブランクは削除されます。SHOWBLANKS コマンドを使用して、先頭および末尾のブランクを保持することができます。

XLSX は HTML ベースのフォーマット (例、EXL2K) ではないため、SHOWBLANKS OFF 設定は内部ブランクに影響しません。デフォルト設定では、EXL2K はすべての内部ブランクを 1 つのブランクにしますが、XLSX はすべての内部ブランクを保持します。ブランクの処理に関するこの違いが、セル内部のフィールドの折り返しの違いに影響する場合があります。

SET SHOWBLANKS	XLSX (非 HTML ベース)	EXL2K (HTML ベース)
SET SHOWBLANKS = ON	先頭ブランクおよび内部ブランクは保持されます。	先頭ブランクおよび内部ブランクは保持されます。
SET SHOWBLANKS = OFF	先頭ブランクは削除されますが、内部ブランクは保持されます。	先頭ブランクおよび内部ブランクは削除されます。

見出し内ブランクについては、処理が異なります。

- デフォルト設定では、複数の項目を含む標準見出しでは (HEADALIGN=BODY 指定なし)、各項目は単一のテキストオブジェクトに連結されます。すべてのブランクは保持されます。
- HEADALIGN=BODY を指定した見出しに配置されたフィールドは、データ要素と同様に動作します。
- HEADALIGN=BODY を指定した見出しに配置された変数は、先頭、内部、末尾のブランクがすべて保持されます。SHOWBLANKS=OFF を指定した場合、内部ブランクのみが保持されます。SHOWBLANKS=ON に指定した場合、先頭、内部、末尾のブランクがすべて保持されます。

XLSX レポート出力の先頭および内部ブランクの保持

プロシジャまたはプロファイルでは、次の構文を使用します。

```
SET SHOWBLANKS = {OFF|ON}
```

リクエストでは、次の構文を使用します。

```
ON TABLE SET SHOWBLANKS {OFF|ON}
```

説明

OFF

XLSX レポート出力の先頭ブランクを削除し、内部ブランクを保持します。デフォルト値は OFF です。

ON

XLSX レポート出力の先頭ブランクおよび内部ブランクを保持します。また、見出し、中間見出し、脚注、中間脚注の行にデフォルトの位置揃えを使用した場合は、末尾ブランクも保持されます。

XLSX レポート出力の先頭および内部ブランクの保持

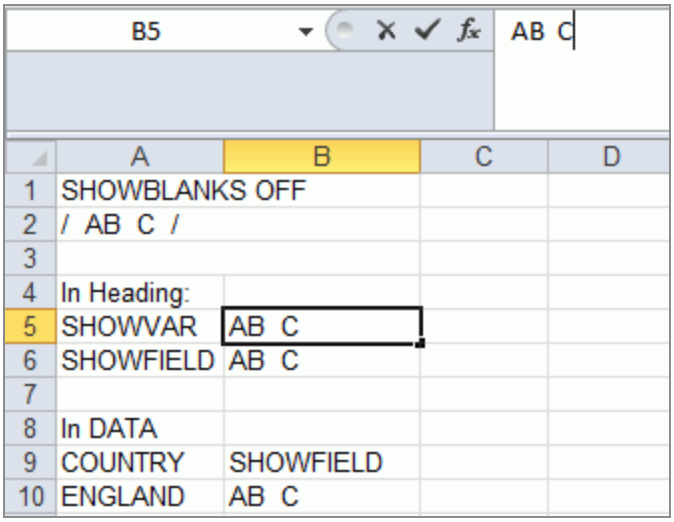
次のリクエストは、先頭、内部、末尾のブランクを含む SHOWVAR という変数を作成します。

```
SET SHOWBLANKS = OFF

-SET &SHOWVAR= '  AB  C  ';
DEFINE FILE CAR
SHOWFIELD/A9 = '  AB  C  ';
END

TABLE FILE CAR
ON TABLE SUBHEAD
"SHOWBLANKS OFF"
"/&SHOWVAR/"
""
HEADING
"In Heading:"
"SHOWVAR<+0>&SHOWVAR"
"SHOWFIELD<+0><SHOWFIELD"
""
"In DATA":
PRINT SHOWFIELD
BY COUNTRY
WHERE RECORDLIMIT EQ 1;
```

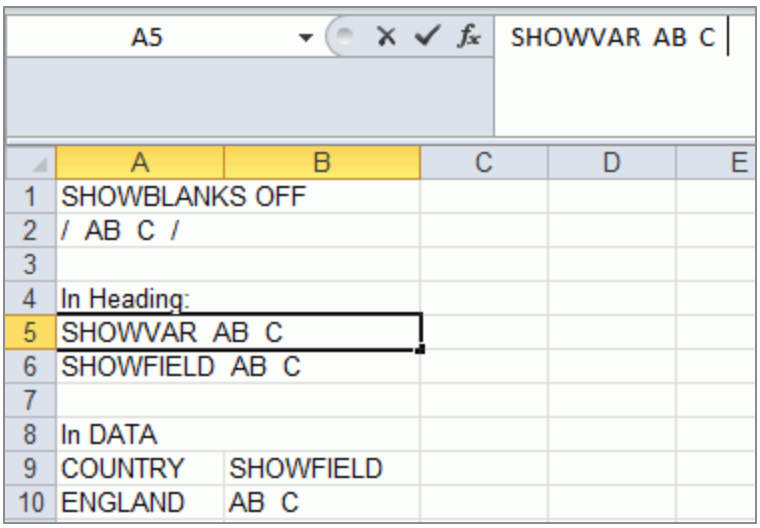
```
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
HEADALIGN=BODY,SQUEEZE=ON,$
TYPE=TABHEADING,COLSPAN=2,$
END
```

下図の各出力は、SET SHOWBLANKS = OFF および SET SHOWBLANKS = ON を使用して生成された XLSX レポートの相違点を示しています。

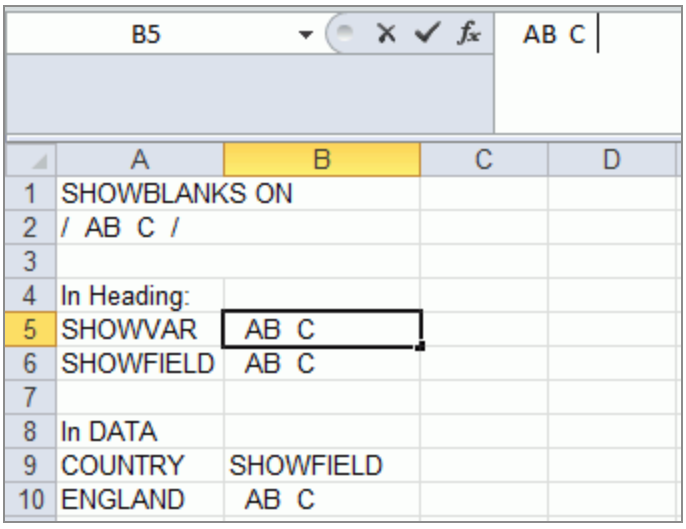
SET SHOWBLANKS = OFF および HEADALIGN=BODY を指定した場合 (先頭または末尾のブランクが保持されません)



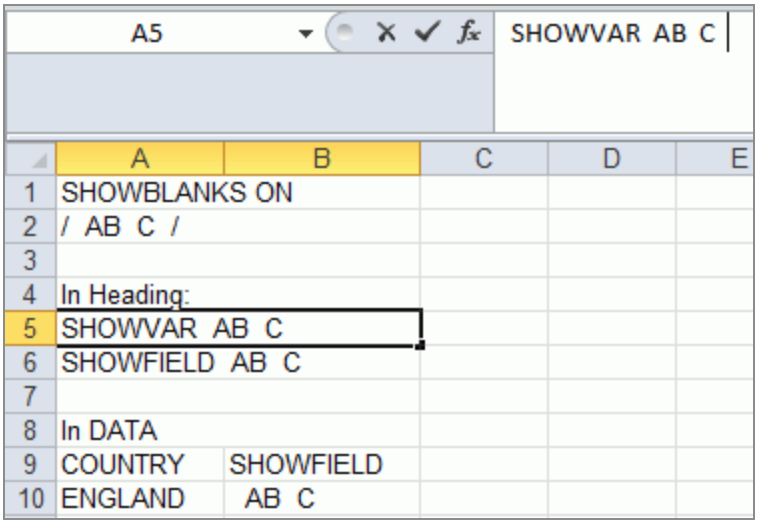
SET SHOWBLANKS = OFF を指定し、 HEADALIGN=BODY を指定しない場合 (ブランクは保持され、見出し項目は連結されます)



SET SHOWBLANKS = ON および HEADALIGN=BODY を指定した場合 (先頭または末尾のブランクが保持されます)



SET SHOWBLANKS = ON を指定し、 HEADALIGN=BODY を指定しない場合 (ブランクは保持され、見出し項目は連結されます)



XLSX レポート出力でのドリルダウンのサポート

XLSX フォーマットでは、EXL2K フォーマットでのサポートと同様の方法で、レポートのデータ要素内でドリルダウンがサポートされます。ハイパーリンク接続は、任意のデータ

フィールドのスタイルシート宣言で定義することができます。ハイパーリンクを使用することで、任意の外部 Web ソースへのアクセスや、プロシジャの実行が可能になります。プロシジャへのドリルダウンにデータ駆動パラメータを追加して、サポートされている任意の出力フォーマット (例、XLSX、PDF、HTML、DHTML、PPT) を生成することができます。

今後のリリースでは、見出し、中間見出し、脚注、中間脚注に埋め込まれたテキスト内でのドリルダウンが実装される予定です。

注意

- ワークシートで使用可能な 65530 個のハイパーリンク制限数に到達すると、警告メッセージが表示され、それ以上のリンクの挿入ができなくなります。ドリルダウンについての詳細は、スタイルシートを使用したリンクを参照してください。
- JAVASCRIPT ドリルダウンオプションおよび IMAGE ドリルダウンオプションは、FORMAT XLSX ではサポートされません。

リダイレクトと Excel ドリルダウンレポート

WebFOCUS Client は、レポート出力をクライアントアプリケーションに転送する際にリダイレクトを使用することができます。リダイレクトが有効になっている場合、リクエストが実行されると、WebFOCUS Client はレポート出力を一時ディレクトリに保存します。次に、ブラウザから HTTP コールが送信され、一時的に保存された出力が取得、表示されます。リダイレクトが無効になっている場合、レポート出力は一時的に保存されず、ブラウザに直接送信されます。

.xlsx ファイル拡張子のリダイレクトは、デフォルト設定で無効になっています。これは、リダイレクトを無効にしておくと、ユーザマシンの構成で Excel をブラウザ内で起動するか、ブラウザ外部のアプリケーションとして起動するかの設定に関係なく、ドリルダウンが正しく実行されるためです。

リダイレクトが有効になっている場合、Excel レポートでのドリルダウンの動作は、ブックをブラウザで開くか (Windows XP にのみ適用)、Excel アプリケーションで開くかに応じて異なります。

- **ブックをブラウザ外部の Excel アプリケーションで開いた場合**　現在のセキュリティコンテキストおよび設定済みのセッション関連 Cookie は保持されず、ユーザ認可が変更されるため、ドリルダウンレポートには、リダイレクトされたファイルへのアクセスに必要な情報が保持されません。最初のブックは Excel 内で開きますが、ドリルダウンのターゲットブックは開かず、「このビューアファイルへのアクセスが許可され

ていません。」というメッセージが表示されます。WebFOCUS バージョン 7.7 では Microsoft Office 製品でのドリルダウン機能は正常に機能していました。これは、ドリルダウンの匿名アクセスが許可されていたためです。

WebFOCUS バージョン 8 でこの機能を有効にするには、次の方法があります。

- 匿名アクセスを許可するよう WebFOCUS 認証を構成する。詳細は、『ibi™ WebFOCUS® セキュリティ管理ガイド』を参照してください。
- SSO を IIS/Tomcat Windows 統合認証と併用します。再ネゴシエーションが自動的に発生し、Excel および PowerPoint レポートが正しく表示されます。
- WebFOCUS バージョン 8 では、[ログイン] ページの [ユーザを記憶する] 機能を有効にすることができます。エンドユーザが [ユーザを記憶する] 機能を使用した場合、永続 Cookie が使用されます。

- **ブックをブラウザで開いた場合 (Windows XP にのみ適用)** リダイレクトが有効になっている場合でもドリルダウンは動作します。これは、リダイレクトされたターゲットブックファイルを取得するには、HTTP リクエストとともに HTTP ヘッダおよび Cookie を WebFOCUS Client に送信する必要がありますが、ブラウザセッション中はこれらの HTTP ヘッダおよび Cookie にアクセスできるためです。

  注意：Windows 7 の場合、Excel アプリケーションはブラウザウィンドウ内で起動されません。

リダイレクトオプションについての詳細は、『ibi™ WebFOCUS® セキュリティ管理ガイド』の「WebFOCUS 管理コンソールによる WebFOCUS Client の設定変更」を参照してください。

Excel ページ設定

XLSX ブックの Excel ページ設定には、デフォルト設定で次の WebFOCUS 標準が適用されます。

- 方向 - 縦
- ページサイズ - Letter
- 上マージン - 0.75 インチ (Excel デフォルト)
- 下マージン - 0.75 インチ (Excel デフォルト)

これらのページ設定をカスタマイズするには、XLSXPAGESETS 属性を ON に変更し、各属性を定義します。

XLSXPAGESETS 属性が ON であっても、ページマージン属性がプロシジャ内で定義されていない場合、これらの値は WebFOCUS デフォルト値の 0.25 インチに設定されます。

Excel ページ設定の定義

```
[TYPE=REPORT,] XLSXPAGESETS={ON|OFF} [,PAGESIZE={pagesize|LETTER}]
 [,ORIENTATION={PORTRAIT|LANDSCAPE}] [,TOPMARGIN=n] [,BOTTOMMARGIN=m],$
```

説明

XLSPAGESETS={ON|OFF}

ON に設定すると、WebFOCUS リクエストで定義されたページ設定が Excel ワークシートのページ設定に適用されます。OFF に設定すると、標準の Excel ブックで定義されたデフォルトのページ設定がそのまま使用されます。デフォルト値は OFF です。

n

UNITS パラメータで指定された単位 (デフォルト単位はインチ) でワークシートの上マージンを定義します。デフォルト値は 0.25 です。

m

UNITS パラメータで指定された単位 (デフォルト単位はインチ) でワークシートの下マージンを定義します。デフォルト値は 0.25 です。

pagesize

WebFOCUS スタイルシートでサポートされる PAGESIZE の値のいずれかです。デフォルトのページサイズは LETTER です。

PORTRAIT|LANDSCAPE

PORTRAIT に設定すると、縦方向が横方向より長いページが作成され、縦長のページにレポートが表示されます。デフォルト値は PORTRAIT です。

LANDSCAPE に設定すると、横方向が縦方向より長いページが作成され、横長のページにレポートが表示されます。

レポートへのイメージの追加

WebFOCUS では、ワークシート上のレポートの各領域または要素にイメージを配置することができます。ロゴなどのイメージは、レポートに企業イメージの統一感や視覚的訴求力を与えます。データ固有のイメージを見出し、脚注、データ列に配置することで、データの意味を明確にしたり、レポートの外観を整えたりできます。

イメージは、WebFOCUS Reporting Server 上の EDAPATH または APPPATH ディレクトリに存在する必要があります。ファイルが検索パスにない場合、完全なパス名を指定します。

Excel XLSX レポートへのイメージの挿入

イメージは、ワークシート上の WebFOCUS レポートの任意の領域または要素に配置することができます。サポートされるイメージフォーマットは、.gif および .jpg です。

使用上の注意

- すべてのイメージは、上端間隔および左端間隔に基づいて、定義された領域の先頭セルの左上隅に配置されます。現時点では、明示的な位置指定および位置揃えは実装されていません。
- 標準のページ設定キーワードを XLSXPAGESETS とともに使用して、標準レポート (複合レポート以外) のページレイアウトを制御することができます。
- 行または列のレポートセル内に配置されたイメージは、そのセルの左上隅に固定されます。最大サイズのイメージに合わせて、このセルの高さと幅が自動的に変更されます (SQUEEZE=ON)。
- 見出し、脚注、中間見出し、中間脚注に配置したイメージが正しく表示されるように、これらの要素内に行の追加が必要になる場合があります。

XLSX レポートの WebFOCUS レポート要素へのイメージの追加

```
TYPE={REPORT|heading|data}, IMAGE={url|file|(column)} [,BY=byfield] [,SIZE=(wh)] ,$
```

説明

REPORT

レポート本文にイメージを埋め込みます。イメージはレポートの背景に表示されます。REPORT がデフォルト値です。

heading

見出しまたは脚注にイメージを埋め込みます。有効な値は、TABHEADING、TABFOOTING、FOOTING、HEADING、SUBHEAD、および SUBFOOT です。見出しまたは脚注内に十分な余白をとり、見出しや脚注のテキストと重ならないようにします。また、スポットマーカーを使用して、イメージの右側に見出しまたは脚注のテキストを配置することもできます。

data

イメージを配置するデータ列内のセルを定義します。COLUMNS= 属性とともに使用して、イメージを固定する特定のレポート列を識別する必要があります。

url

イメージファイルの URL です。

file

イメージファイル名です。このファイルは、EDAPATH または APPPATH で指定された、WebFOCUS Reporting Server 上のディレクトリに存在する必要があります。ファイルが検索パスにない場合、完全なパス名を指定します。GIF を指定する場合は、ファイルの拡張子を省略できます。

column

イメージファイル名を含むデータソース内の文字フィールドです。フィールドは括弧で囲みます。フィールドにファイル名またはイメージが含まれている場合、そのフィールドはリクエスト内で参照されている表示フィールドまたは BY フィールドにする必要があります。フィールドの値は、スタイルシート内のイメージの URL として入力されたときと同様に解釈されます。接尾語を省略した場合、デフォルト設定で「.GIF」 が指定されます。SET BASEURL コマンドを使用して、イメージのベース URL を指定することができます。この方法では、フィールドの値に完全な URL を含める必要はありません。たとえば、SUBHEAD にイメージを埋め込み、BY フィールドの値が変わるたびに SUBHEAD に異なるイメージを表示する場合にこの構文が役立ちます。

byfield

中間見出しまたは中間脚注を生成するソートフィールドです。

SIZE

イメージのサイズです。デフォルト設定では、イメージは元のサイズで追加されます。

w

UNITS パラメータで指定された測定単位でイメージの幅を表します。*w* および *h* の値は括弧で囲みます。その間にカンマ (,) は挿入しません。

h

UNITS パラメータで指定された測定単位でイメージの高さを表します。

XLSX レポートの見出しまたは脚注へのイメージの配置

XLSX レポートの CELL-POSITION プロパティを使用すると、見出しまたは脚注に、イメージを挿入して配置することができます。このプロパティについての注意点は、次のとおりです。

- CELL-POSITION プロパティは、見出しまたは脚注の左上隅を基準に適用されます。たとえば、脚注の CELL-POSITION(+2, +1) は、脚注エリアの左上隅から 2 セル右、1 セル下の位置を表します。
- 画像を配置する位置によっては、テキストに重なる可能性があります。この問題が発生しないよう、正しいイメージサイズを使用する必要があります。
- CELL-DIMENSION プロパティを使用すると、イメージサイズを指定したセル範囲に収めることができます。
- CELL-DIMENSION を指定しない場合、SIZE プロパティを使用することができます。CELL-POSITION、SIZE のどちらも使用しない場合、イメージは元のサイズで表示されます。

```
TYPE=[HEADING | FOOTING], IMAGE=_image.gif_, CELL-POSITION=(x y), CELL-
DIMENSION(x y)$
```

説明

heading

イメージを Excel レポートの見出しに埋め込みます。

footing

イメージを Excel レポートの脚注に埋め込みます。

image

Excel レポートに挿入して配置するイメージです。

cell-position

見出しまたは脚注の左上隅を基準とする、イメージの位置を記述します。

cell-dimension

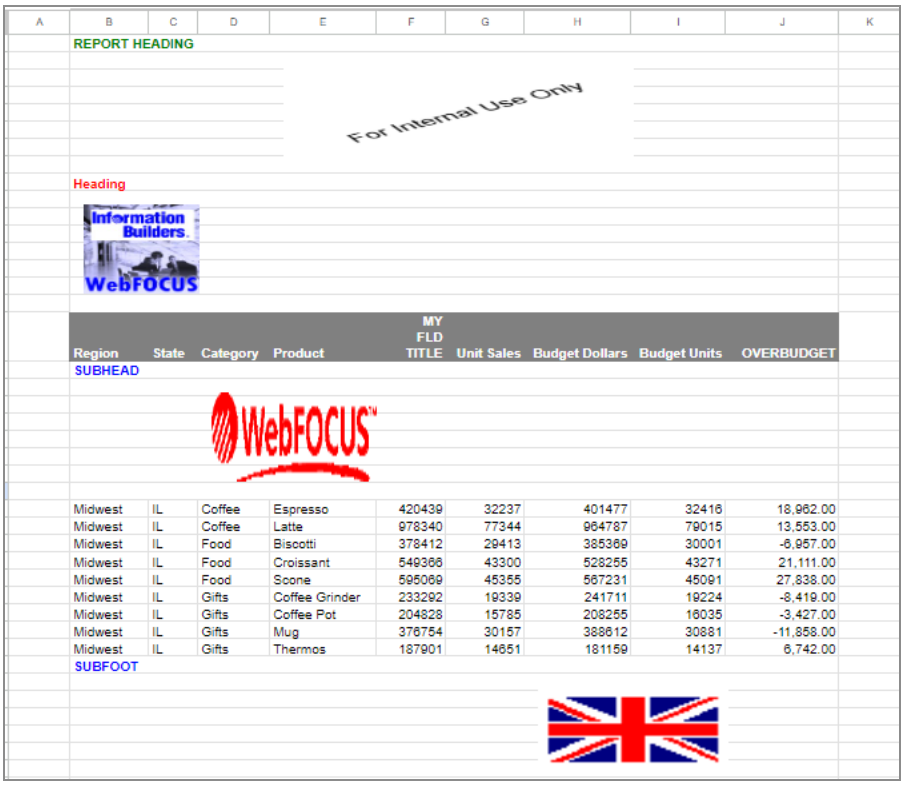
イメージのサイズをセル単位 (縦方向または横方向) で記述します。

次のプロシジャは、Excel レポートの見出しまたは脚注にイメージを挿入する例を示しています。

```
-INCLUDE STIM001A
-RUN

COMPOUND LAYOUT HOLD AS foccache/all_types_cmpd_1img FORMAT XLSX
SECTION=Section_1, LAYOUT=ON,  $
  PAGELAYOUT=1, $
    COMPONENT=Rpt_1, COMPONENT-TYPE=REPORT, POSITION=(.5 1), CELL-POSITION=(2
1), DIMENSION=(4 5), $
END

SET COMPONENT=Rpt_1
-*SET LAYOUTRTL=ON
TABLE FILE GGSALES
HEADING
"Heading"
"</6"
ON TABLE SUBHEAD
"REPORT HEADING"
"</6"
SUM DOLLARS AS 'MY,FLD,TITLE' UNITS BUDDOLLARS BUDUNITS
COMPUTE OVERBUDGET/D12.2=DOLLARS-BUDDOLLARS;
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
ON REGION SUBHEAD
"SUBHEAD"
"</6"
ON REGION SUBFOOT
```

```
"SUBFOOT"
"</6"
FOOTING
"Footing"
"</6"
ON TABLE SUBFOOT
"REPORT FOOTING"
"</6"
ON TABLE SET BYDISPLAY ON
ON TABLE SET STYLE *
TYPE=TITLE,      STYLE=BOLD, COLOR=WHITE, BACKCOLOR=GREY, $
TYPE=HEADING,    STYLE=BOLD, COLOR=RED, $
TYPE=FOOTING,    STYLE=BOLD, COLOR=RED, $
TYPE=TABHEADING, STYLE=BOLD, COLOR=GREEN, $
TYPE=TABFOOTING, STYLE=BOLD, COLOR=GREEN, $
TYPE=SUBHEAD,    STYLE=BOLD, COLOR=BLUE, $
TYPE=SUBFOOT,    STYLE=BOLD, COLOR=BLUE, $
TYPE=HEADING,  IMAGE=stim0017.jpg, position=(.25 .25), cell-position=(+0 +1),
cell-dimension=(2 6), $
TYPE=TABHEADING, IMAGE=stii008a.jpg, position=(.25 .25), cell-position=(+3
+1), cell-dimension=(4 7), $
TYPE=SUBHEAD, IMAGE=blob0011.gif, position=(4.25 .25), cell-position=(+2 +1),
cell-dimension=(2 6), $
TYPE=FOOTING, IMAGE=stim0017.jpg, position=(8.25 .25), cell-position=(+7 +1),
cell-dimension=(2 6), $
TYPE=TABFOOTING, IMAGE=stii008a.jpg, position=(8.25 .25), cell-position=(+3
+1), cell-dimension=(4 7), $
TYPE=SUBFOOT, IMAGE=blob0005.gif, position=(4.25 .25), cell-position=(+6 +1),
cell-dimension=(2 6), $
END

COMPOUND END
```

下図は、レポートの見出しおよび脚注セクションに配置されたイメージを示しています。



単一テーブルリクエストへの GIF イメージの追加

次のリクエストでは、参照されているイメージが既存の GGSALES テーブルの一部でないため、最初にイメージファイル (.gif) が DEFINE で作成され、次に TABLE リクエストで参照されています。フィールドをレポートの列として表示しない場合は NOPRINT で非表示にすることができますが、フィールドを内部マトリックスに含めるには、TABLE リクエストでそのフィールドが参照されている必要があります。これにより、見出し、脚注、データセルへのイメージの配置が可能になります。フィールド値に基づいてイメージを配置するために、スタイルシート定義を使用して特定の位置が指定されています。

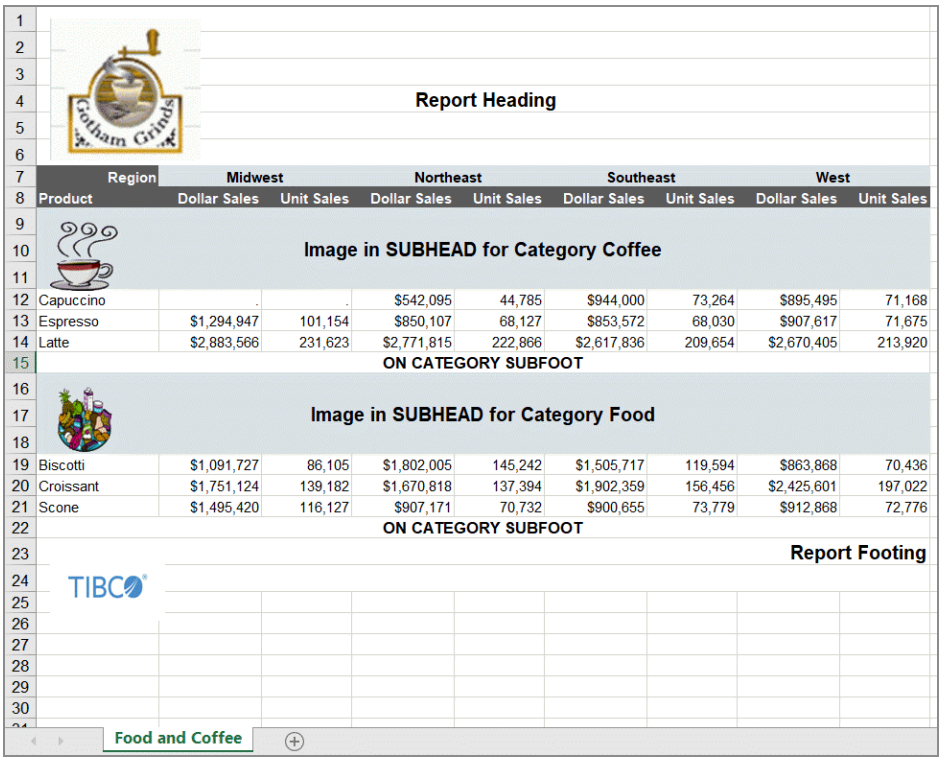
```
DEFINE FILE GGSALES
SHOWCAT/A100=CATEGORY || '.GIF';
END
TABLE FILE GGSALES
SUM DOLLARS/D12CM UNITS/D12C
BY LOWEST CATEGORY NOPRINT
BY SHOWCAT NOPRINT
```

```
BY PRODUCT
ACROSS REGION
WHERE CATEGORY NE 'Gifts'
```

```
ON CATEGORY SUBHEAD
" "
“Image in SUBHEAD for Category <CATEGORY "
" "
ON TABLE SUBHEAD
" "
" "
" "
" Report Heading "
" "
" "
ON CATEGORY SUBFOOT
"ON CATEGORY SUBFOOT"
```

```
ON TABLE SUBFOOT
"Report Footing"
" "
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE NOTOTAL
ON TABLE SET ACROSSTITLE SIDE
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT='ARIAL', SIZE=9, TITLETEXT='Food and Coffee',$
TYPE=REPORT, COLUMN=PRODUCT, SQUEEZE=1,$
TYPE=TITLE, BACKCOLOR=RGB(90 90 90), COLOR=RGB(255 255 255), STYLE=BOLD,$
TYPE=ACROSSTITLE, STYLE=BOLD, BACKCOLOR=RGB(90 90 90),
COLOR=RGB(255 255 255),$
TYPE=ACROSSVALUE, BACKCOLOR=RGB(218 225 232), STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=HEADING, STYLE=BOLD, COLOR=RGB(0 35 95), SIZE=12, JUSTIFY=Center,$
TYPE=FOOTING, BACKCOLOR=RGB(90 90 90), SIZE=12, COLOR=RGB(255 255 255),
STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=SUBHEAD, SIZE=12, STYLE=BOLD, BACKCOLOR=RGB(218 225 232),
JUSTIFY=CENTER,$
TYPE=SUBHEAD, IMAGE=(SHOWCAT), SIZE=(.6 .6),$
TYPE=SUBFOOT, SIZE=10, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=TABHEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=TABHEADING, IMAGE=gglogo.gif,$
TYPE=TABFOOTING, SIZE=12, STYLE=BOLD, JUSTIFY=RIGHT,$
TYPE=TABFOOTING, IMAGE=tibco.gif, SIZE=(1.0 .6),$
END
```

出力結果は次のとおりです。



	Region	Midwest		Northeast		Southeast		West	
	Product	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales
	Capuccino	.	.	$542,095	44,785	$944,000	73,264	$895,495	71,168
	Espresso	$1,294,947	101,154	$850,107	68,127	$853,572	68,030	$907,617	71,675
	Latte	$2,883,566	231,623	$2,771,815	222,866	$2,617,836	209,654	$2,670,405	213,920
	Biscotti	$1,091,727	86,105	$1,802,005	145,242	$1,505,717	119,594	$863,868	70,436
	Croissant	$1,751,124	139,182	$1,670,818	137,394	$1,902,359	156,456	$2,425,601	197,022
	Scone	$1,495,420	116,127	$907,171	70,732	$900,655	73,779	$912,868	72,776

複合リクエストへの GIF イメージの追加

注意：複合レイアウト構文には、非表示のキャリッジリターンおよびラインフィード文字を含めることはできません。この例では、サンプルコードにラインフィード文字が追加され、印刷ページ幅にコードが収まるようコード行が折り返されています。使用する環境でこのサンプルコードを実行するには、コードをテキストエディタにコピーし、各行の末尾に移動して Delete キーを押すことにより、複合レイアウト構文内のラインフィード文字を削除します。場合によっては、文字列の構造を保持するために、ブランクを追加する必要があります。

```
APP PATH IBISAMP
SET HTMLARCHIVE=ON
*-HOLD_SOURCE
COMPOUND LAYOUT PCHOLD FORMAT XLSX
UNITS=IN,$
SECTION=section1, LAYOUT=ON, METADATA='prop_with_names, Margins_Left=0.5,
Margins_Top=0.5, Margins_Right=0.5, Margins_Bottom=0.5,
thumbnailscale=4', MERGE=OFF, ORIENTATION=LANDSCAPE, PAGESIZE=Legal,
SHOW_GLOBALFILTER=OFF,$
PAGELAYOUT=1, NAME='Page layout 1', text='Page layout 1', TOC-LEVEL=1,
BOTTOMMARGIN=0.5, TOPMARGIN=0.5, METADATA='BOTTOMMARGIN=0.5, TOPMARGIN=0.5,
```

```
LEFTMARGIN=0,RIGHTMARGIN=0,',$
COMPONENT='report1', TEXT='report1', TOC-LEVEL=2, POSITION=(0.650 0.917),
DIMENSION=(7.250 3.000), BYTOC=0,  ARREPORTSIZE=DIMENSION,
METADATA='left:0.65in; top:0.917in; width:7.25in; height:3in;
position: absolute; z-index:1;',$
COMPONENT='chart1', TEXT='chart1', TOC-LEVEL=2, POSITION=(0.735 4.332),
DIMENSION=(7.167 2.917), COMPONENT-TYPE=GRAPH,  ARREPORTSIZE=DIMENSION,
METADATA='left:0.735in; top:4.332in; width:7.167in; height:2.917in;
position: absolute; z-index:2;',$
END
```

```
SET COMPONENT='report1'
-*component_type report
DEFINE FILE GGSALES
SHOWCAT/A100=CATEGORY || '.GIF';
SHOWDATEQ/Q=DATE;
SHOWDATEY/YY=DATE;
SHOWDATEQY/YYQ=DATE;
END
```

```
TABLE FILE GGSALES
SUM DOLLARS/D12CM AS 'Dollars'
BY REGION AS ''
BY LOWEST CATEGORY
BY SHOWCAT AS 'Data Image'
ACROSS SHOWDATEY AS ''
ACROSS SHOWDATEQ AS ''
WHERE REGION NE 'Midwest' OR 'West'
ON TABLE SET HIDENULLACRS ON
HEADING
" "
"Image in Page Heading "
ON REGION SUBHEAD
" <+0> SUBHEAD: <REGION"
FOOTING
" "
"Image in Page Footing"
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=10, BACKCOLOR=NONE, GRID=OFF, FONT='ARIAL',
XLSXPAGESETS=ON, TOPMARGIN=1, BOTTOMMARGIN=1, ORIENTATION=LANDSCAPE,
PAGESIZE=LEGAL, TITLETEXT='With Images',$
TYPE=REPORT, COLUMN=REGION, SQUEEZE=1.5, JUSTIFY=CENTER,$
```

```
TYPE=DATA, BACKCOLOR=NONE,$
TYPE=DATA, COLUMN=SHOWCAT, IMAGE=(SHOWCAT), SIZE=(.5 .5),$
TYPE=TITLE, BACKCOLOR=RGB(218 225 232), BORDER=LIGHT,
STYLE=-UNDERLINE+BOLD,$
TYPE=HEADING, IMAGE=GGLOGO.GIF, SIZE=(.65 .65),$
TYPE=HEADING, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=SUBHEAD, SIZE=10, STYLE=BOLD, BORDER-TOP=LIGHT,$
TYPE=SUBHEAD, BY=1, JUSTIFY=CENTER, BORDER-TOP=LIGHT,$
TYPE=SUBFOOT, STYLE=BOLD,$
TYPE=FOOTING, SIZE=12, STYLE=+BOLD, JUSTIFY=CENTER,$
TYPE=FOOTING, IMAGE=tibco.gif, SIZE=(1.0 .6),$
TYPE=ACROSS, JUSTIFY=CENTER, BORDER=LIGHT,$
TYPE=ACROSSTITLE, STYLE=-UNDERLINE+BOLD,$
TYPE=ACROSSVALUE, BACKCOLOR=RGB(218 225 232), STYLE=-UNDERLINE+BOLD,$
END
```

```
SET COMPONENT='chart1'
ENGINE INT CACHE SET ON
-DEFAULTH &WF_STYLE_UNITS='PIXELS';
-DEFAULTH &WF_STYLE_HEIGHT='1005.0';
-DEFAULTH &WF_STYLE_WIDTH='1070.0';
-DEFAULTH &WF_TITLE='WebFOCUS Report';
GRAPH FILE GGSALES
HEADING
"Sales Graph"
SUM
GGSALES.SALES01.DOLLARS
BY SHOWDATEY AS Year
BY GGSALES.SALES01.REGION
ON GRAPH PCHOLD FORMAT XLSX
ON GRAPH SET VZERO OFF
ON GRAPH SET HTMLENCODE ON
ON GRAPH SET GRAPHDEFAULT OFF
ON GRAPH SET GRWIDTH 1
ON GRAPH SET UNITS &WF_STYLE_UNITS
ON GRAPH SET HAXIS 1000
ON GRAPH SET VAXIS 1000
ON GRAPH SET GRMERGE ADVANCED
ON GRAPH SET GRLEGEND 0
ON GRAPH SET GRXAXIS 2
ON GRAPH SET LOOKGRAPH HBAR
ON GRAPH SET STYLE *
```

```
*GRAPH_SCRIPT
setPieDepth(0);
```

```
setPieTilt(0);
setDepthRadius(0);
setCurveFitEquationDisplay(false);
setPlace(true);
setPieFeelerTextDisplay(1);
setUseSeriesShapes(true);
setMarkerSizeDefault(50);
setScaleMustIncludeZero(getX1Axis(), false);
setScaleMustIncludeZero(getY1Axis(), false);
setScaleMustIncludeZero(getY2Axis(), false);
setMarkerSizeDefault(60);
*END
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/javaassist/intl/EN/ENIADefault_combine.sty,$
TYPE=REPORT, TITLETEXT='Graph Over Time',$
*GRAPH_SCRIPT
setReportParsingErrors(false);
setSelectionEnableMove(false);
*END
ENDSTYLE
END
COMPOUND END
```

下図は、出力結果を示しています。

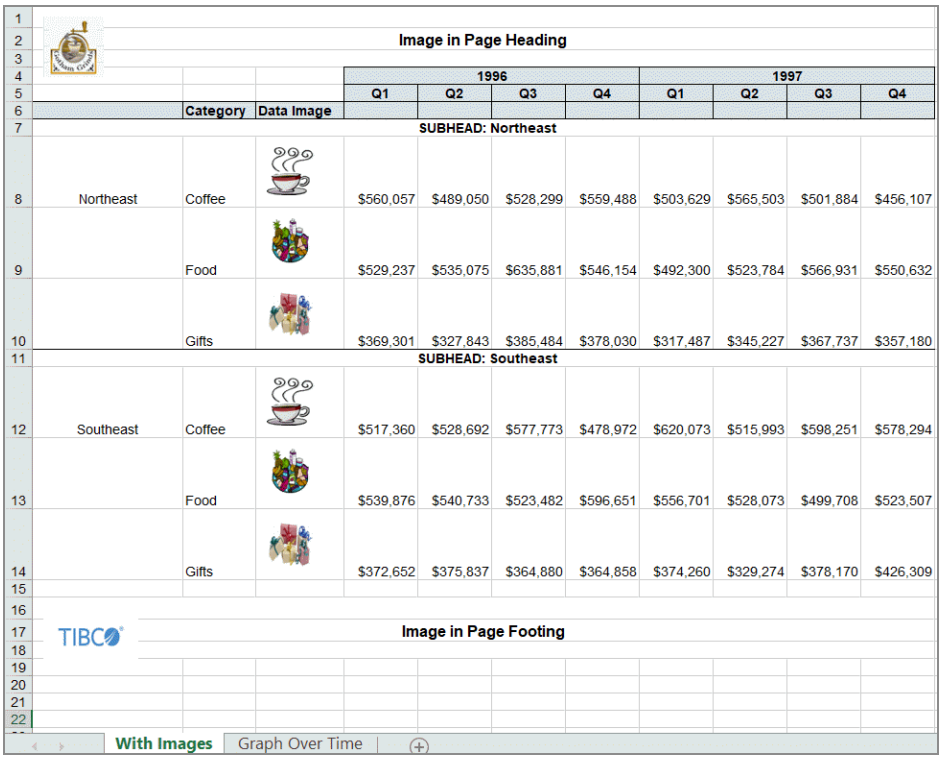
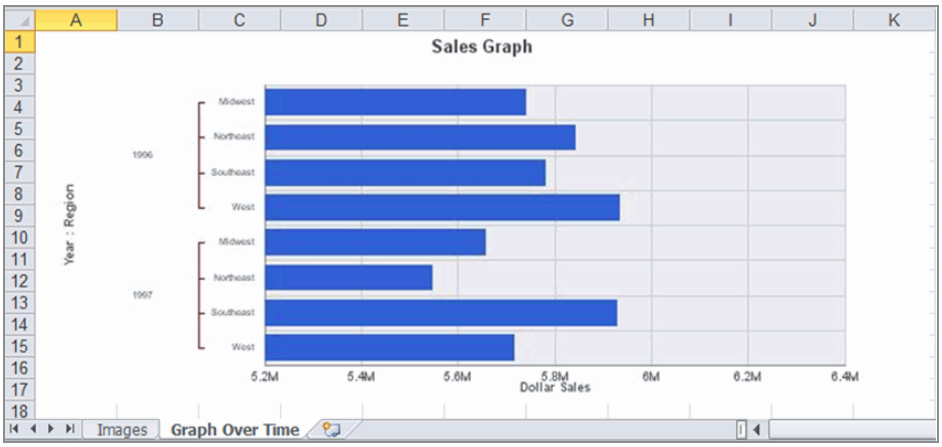


Image in Page Heading

	Category	Data Image	1996 Q1	1996 Q2	1996 Q3	1996 Q4	1997 Q1	1997 Q2	1997 Q3	1997 Q4
SUBHEAD: Northeast										
Northeast	Coffee		$560,057	$489,050	$528,299	$559,488	$503,629	$565,503	$501,884	$456,107
	Food		$529,237	$535,075	$635,881	$546,154	$492,300	$523,784	$566,931	$550,632
	Gifts		$369,301	$327,843	$385,484	$378,030	$317,487	$345,227	$367,737	$357,180
SUBHEAD: Southeast										
Southeast	Coffee		$517,360	$528,692	$577,773	$478,972	$620,073	$515,993	$598,251	$578,294
	Food		$539,876	$540,733	$523,482	$596,651	$556,701	$528,073	$499,708	$523,507
	Gifts		$372,652	$375,837	$364,880	$364,858	$374,260	$329,274	$378,170	$426,309

Image in Page Footing

With Images | Graph Over Time



BYTOC 複合リクエストへの GIF イメージの追加

次のコードは、前述の例で使用されたコードの一部を示しています。この例では、COMPOUND BYTOC 構文を使用します。前述の複合レポートに ON TABLE SET COMPOUND BYTOC コマンドを追加することで、複合レポートを複合目次レポートに変換することができます。BYTOC 構文は、スタンドアロンリクエストに追加することも、複合ドキュメントのコンポーネントに追加することもできます。

```
TABLE FILE GGSALES
SUM DOLLARS/D12CM AS 'Dollars'
SHOWREG
NOPRINT
BY REGION AS ''
BY LOWEST CATEGORY
BY SHOWCAT AS 'Data Image'
WHERE REGION NE 'Midwest' OR 'West'
ACROSS SHOWDATEY AS '' ACROSS SHOWDATEQ AS ''
ON TABLE SET HIDENULLACRS ON
HEADING
"Image in Page Heading"
ON REGION SUBHEAD
"<+0> Image in SUBHEAD:<REGION"
FOOTING
" "
"Image in Page Footing"
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT XLSX
```

```
ON TABLE SET COMPOUND BYTOC
ON TABLE SET STYLE *
```

出力結果は次のとおりです。

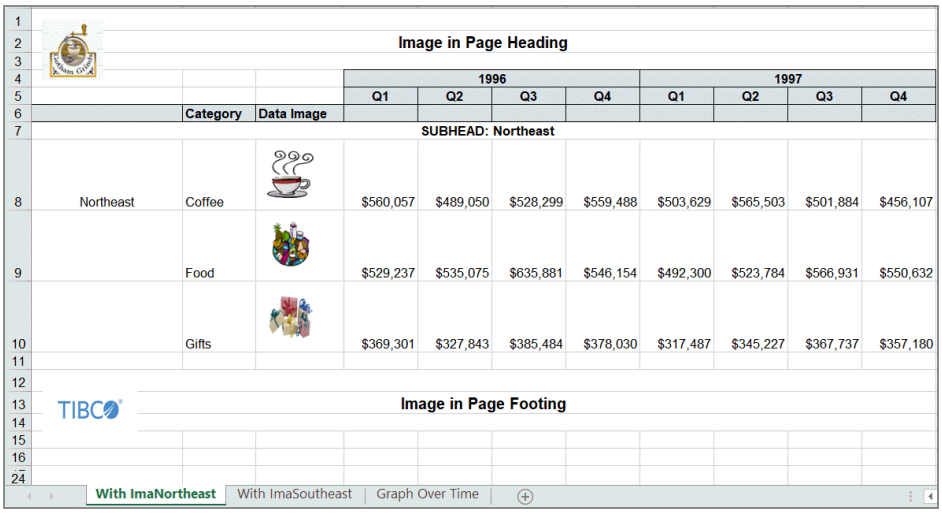


Image in Page Heading

			1996				1997			
	Category	Data Image	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
SUBHEAD: Northeast										
Northeast	Coffee		$560,057	$489,050	$528,299	$559,488	$503,629	$565,503	$501,884	$456,107
	Food		$529,237	$535,075	$635,881	$546,154	$492,300	$523,784	$566,931	$550,632
	Gifts		$369,301	$327,843	$385,484	$378,030	$317,487	$345,227	$367,737	$357,180

Image in Page Footing

With ImaNortheast | With ImaSoutheast | Graph Over Time

XLSX ブックの見出しおよび脚注へのテキストおよびイメージの挿入

WebFOCUS では、Excel の見出しおよび脚注へのテキストおよびイメージの挿入がサポートされます。また、主要なページ設定を定義することで、挿入したテキストおよびイメージをワークシート全体と Excel が生成する改ページに関連させて配置することができます。この方法で Excel ページ機能を利用することにより、ワークシートにレポートを表示する際の全体的なユーザ操作性が向上します。ページの向き、サイズ、マージンなどのページ設定は、プロシジャ内で定義された値に基づいて各 Excel ページのレイアウトに直接反映されます。イメージおよびテキストは、表示する各ページ、レポートの先頭ページのみ、または後続のすべてのページのみの見出しおよび脚注に挿入することができます。WebFOCUS の見出しおよび脚注は、通常どおりワークシート内に表示されます。この機能により、レポートの最上部に一度だけ表示するロゴを挿入したり、すべてのページに表示する透かしイメージを挿入したりすることが可能になります。

XLSX ブックの見出しおよび脚注へのテキストおよびイメージの挿入

XLSX ブックの見出しおよび脚注にイメージを配置するには、次の構文を使用します。

```
TYPE={PAGEHEADER|PAGEFOOTER},OBJECT=IMAGE,
 IMAGE=imagename, JUSTIFY={LEFT|CENTER|RIGHT}
 [,DISPLAYON={FIRST|NOT-FIRST}] [,SIZE=(wh)],$
```

XLSX ブックの見出しおよび脚注にテキストを配置するには、次の構文を使用します。

```
TYPE={PAGEHEADER|PAGEFOOTER},OBJECT=STRING,
 TEXT=text, JUSTIFY={LEFT|CENTER|RIGHT}
 [,DISPLAYON={FIRST|NOT-FIRST}] ,$
```

説明

PAGEHEADER

ワークシートの見出しにテキストまたはイメージを配置します。

PAGEFOOTER

ワークシートの脚注にテキストまたはイメージを配置します。

imagename

見出しまたは脚注に配置する有効なイメージファイルの名前です。イメージは、Reporting Server で定義されたアプリケーションパス上に存在する必要があります。サポートされるイメージタイプは、GIF および JPEG です。

text

見出しまたは脚注に配置するテキストです。

JUSTIFY={LEFT|CENTER|RIGHT}

見出しまたは脚注内でテキストまたはイメージを配置する領域、およびその領域内での位置揃えを識別します。

DISPLAYON

テキストまたはイメージを先頭ページのみに配置するか、先頭ページ以外のすべてのページに配置するかを定義します。すべてのページにテキストまたはイメージを配置する場合

は、この属性を省略します。

有効な値には、次のものがあります。

FIRST - 先頭ページのみにテキストまたはイメージを配置します。

NOT-FIRST - 先頭ページ以外のすべてのページにテキストまたはイメージを配置します。

SIZE=(wh)

イメージのサイズです。デフォルト設定では、イメージは元のサイズで追加されます。

w - UNITS パラメータで指定された単位でイメージの幅を指定します。

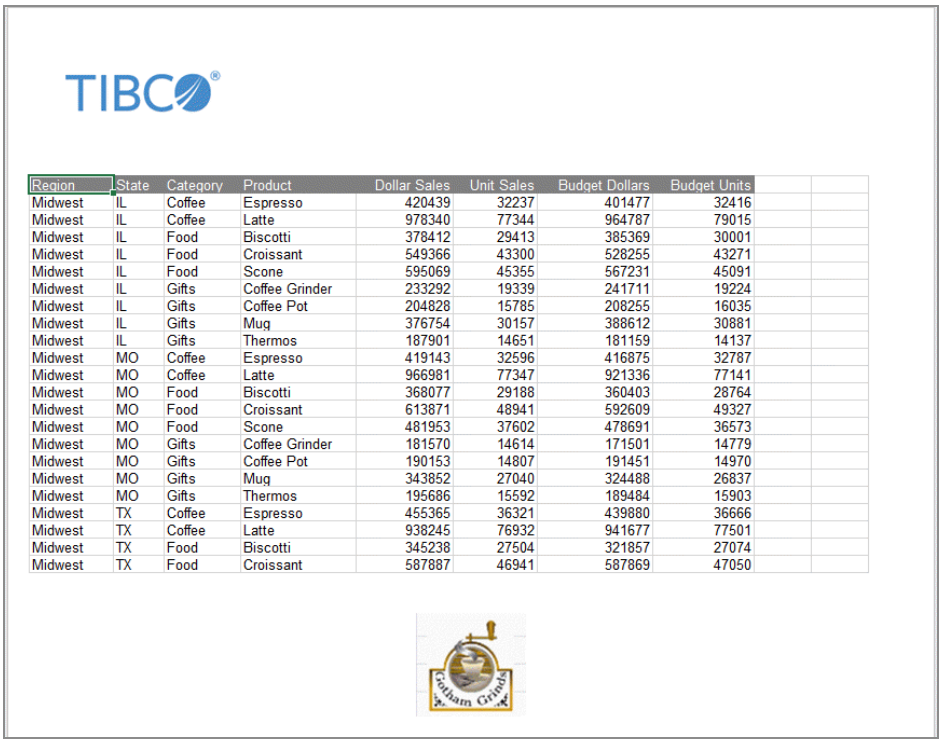
h - UNITS パラメータで指定された単位でイメージの高さを指定します。

Excel の見出しおよび脚注へのイメージ挿入とページ設定の定義

次のリクエストは、GGSALES データソースに対して実行され、出力されるワークシートの先頭ページの見出し左領域と、残りの各ページの見出し右領域に tibco.gif イメージを配置します。また、各ページの脚注の中央領域に gglogo1.gif イメージを配置します。

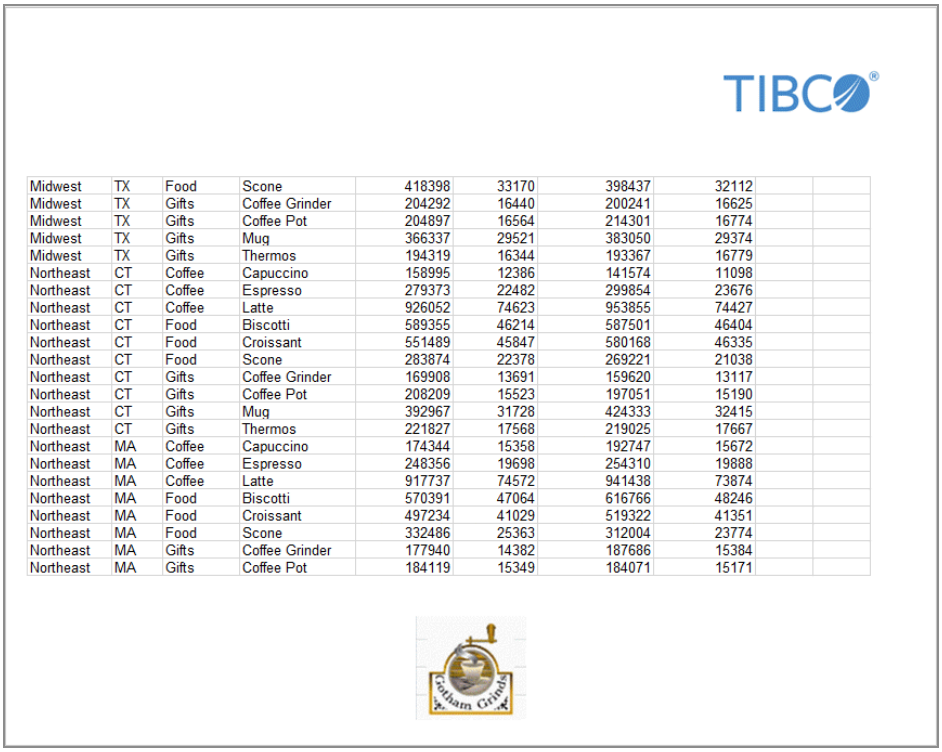
```
TABLE FILE GGSALES
SUM DOLLARS UNITS BUDDOLLARS BUDUNITS
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
ON TABLE SET BYDISPLAY ON
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
FONT=ARIAL,SIZE=12,
XLSXPAGESETS=ON,TOPMARGIN=2,BOTTOMMARGIN=2,ORIENTATION=LANDSCAPE,
PAGESIZE=LETTER,$
TYPE=TITLE, COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=PAGEHEADER, OBJECT=IMAGE, JUSTIFY=LEFT, IMAGE=tibco.GIF,
DISPLAYON=FIRST,$
TYPE=PAGEHEADER, OBJECT=IMAGE, JUSTIFY=RIGHT, IMAGE=tibco.GIF, DISPLAYON=NOT-
FIRST,$
TYPE=PAGEFOOTER, OBJECT=IMAGE, JUSTIFY=CENTER, IMAGE=gglogo.GIF,$
END
```

出力結果の先頭ページでは、見出しの左領域に tibco.gif イメージが配置され、脚注の中央領域に gglogo.gif イメージが配置されています。



Region	State	Category	Product	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
Midwest	IL	Coffee	Espresso	420439	32237	401477	32416
Midwest	IL	Coffee	Latte	978340	77344	964787	79015
Midwest	IL	Food	Biscotti	378412	29413	385369	30001
Midwest	IL	Food	Croissant	549366	43300	528255	43271
Midwest	IL	Food	Scone	595069	45355	567231	45091
Midwest	IL	Gifts	Coffee Grinder	233292	19339	241711	19224
Midwest	IL	Gifts	Coffee Pot	204828	15785	208255	16035
Midwest	IL	Gifts	Mug	376754	30157	388612	30881
Midwest	IL	Gifts	Thermos	187901	14651	181159	14137
Midwest	MO	Coffee	Espresso	419143	32596	416875	32787
Midwest	MO	Coffee	Latte	966981	77347	921336	77141
Midwest	MO	Food	Biscotti	368077	29188	360403	28764
Midwest	MO	Food	Croissant	613871	48941	592609	49327
Midwest	MO	Food	Scone	481953	37602	478691	36573
Midwest	MO	Gifts	Coffee Grinder	181570	14614	171501	14779
Midwest	MO	Gifts	Coffee Pot	190153	14807	191451	14970
Midwest	MO	Gifts	Mug	343852	27040	324488	26837
Midwest	MO	Gifts	Thermos	195686	15592	189484	15903
Midwest	TX	Coffee	Espresso	455365	36321	439880	36666
Midwest	TX	Coffee	Latte	938245	76932	941677	77501
Midwest	TX	Food	Biscotti	345238	27504	321857	27074
Midwest	TX	Food	Croissant	587887	46941	587869	47050

出力結果の 2 ページ目では、見出しの右領域に tibco.gif イメージが配置され、脚注の中央領域に gglogo.gif イメージが配置されています。



Midwest	TX	Food	Scone	418398	33170	398437	32112
Midwest	TX	Gifts	Coffee Grinder	204292	16440	200241	16625
Midwest	TX	Gifts	Coffee Pot	204897	16564	214301	16774
Midwest	TX	Gifts	Mug	366337	29521	383050	29374
Midwest	TX	Gifts	Thermos	194319	16344	193367	16779
Northeast	CT	Coffee	Capuccino	158995	12386	141574	11098
Northeast	CT	Coffee	Espresso	279373	22482	299854	23676
Northeast	CT	Coffee	Latte	926052	74623	953855	74427
Northeast	CT	Food	Biscotti	589355	46214	587501	46404
Northeast	CT	Food	Croissant	551489	45847	580168	46335
Northeast	CT	Food	Scone	283874	22378	269221	21038
Northeast	CT	Gifts	Coffee Grinder	169908	13691	159620	13117
Northeast	CT	Gifts	Coffee Pot	208209	15523	197051	15190
Northeast	CT	Gifts	Mug	392967	31728	424333	32415
Northeast	CT	Gifts	Thermos	221827	17568	219025	17667
Northeast	MA	Coffee	Capuccino	174344	15358	192747	15672
Northeast	MA	Coffee	Espresso	248356	19698	254310	19888
Northeast	MA	Coffee	Latte	917737	74572	941438	73874
Northeast	MA	Food	Biscotti	570391	47064	616766	48246
Northeast	MA	Food	Croissant	497234	41029	519322	41351
Northeast	MA	Food	Scone	332486	25363	312004	23774
Northeast	MA	Gifts	Coffee Grinder	177940	14382	187686	15384
Northeast	MA	Gifts	Coffee Pot	184119	15349	184071	15171

XLSX ワークシートの見出しおよび脚注にテキストおよびイメージを挿入する際の注意

- Microsoft Excel では、ページ見出しまたはページ脚注に含める最大文字数は 255 バイトです。これは、ページ全体の見出しまたは脚注 (左、中央、右すべて) の上限であり、テキスト文字列以外に Microsoft Excel で使用する必要のある記号およびその他の文字も含まれます。この上限を超えるテキスト文字列は切り捨てられます。
- Excel の見出しおよび脚注のサイズは、領域のコンテンツに合わせて自動的に変更されません。ページ設定 (XLSPAGESETS) でページマージンを定義する際に、レポートの各ページにイメージを表示するためのスペースを確保する必要があります。
- イメージの縦と横の長さを指定してサイズを変更する場合、縦横比は一定になりません。サイズの変更により、イメージが歪む場合があります。
- このバージョンでは、BLOB イメージフィールドはサポートされません。

XLSX レポート出力での透かしの表示

Excel の見出しに透かしイメージを配置することで、生成されるワークシートの出力ページごとに透かしを表示することができます。

Excel では、ページ上のイメージは、見出し内の左から右へ、次に脚注内の左から右へという順序で配置されます。大きいイメージを見出しに配置すると、そのイメージより前に配置されたイメージと重なり合う場合があります。ページの左領域にロゴを配置し、中央に透かしを配置したページレイアウトでは、ロゴのイメージが隠れないよう背後の透かしイメージを透明にする必要があります。

Excel では、最初にイメージがページ上に配置されます。次にワークシートのその他のコンテンツがイメージの上に重ねて配置されます。セル内のテキストおよびスタイル (例、背景色、描画オブジェクト) はイメージの上に重ねて配置されます。Excel では、描画オブジェクトおよびイメージで透明色がサポートされますが、セルの背景色ではサポートされません。BACKCOLOR を使用すると、その背景色により、ページ上に配置されたイメージが隠れます。

XLSX 見出しへの透かしの配置

次のリクエストは、GGSALES データソースに対して実行され、ワークシートの各ページの背後に表示する透かしとして internaluseonly.gif イメージを使用します。このイメージは見出しの中央に配置されていますが、サイズが大きいためワークシートのページ全体に表示されます。このイメージは透明な背景のため、見出しの左側および脚注の中央に配置されたロゴイメージは隠れません。

```
TABLE FILE GGSALES
SUM DOLLARS UNITS BUDDOLLARS BUDUNITS
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
ON TABLE SET BYDISPLAY ON
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
XLSXPAGESETS=ON,
TOPMARGIN=1,BOTTOMMARGIN=1,LEFTMARGIN=1, RIGHTMARGIN=1,
ORIENTATION=LANDSCAPE,PAGESIZE=LETTER,$
TYPE=PAGEHEADER, OBJECT=IMAGE, JUSTIFY=LEFT, IMAGE=TIBCO.GIF,
DISPLAYON=FIRST,$
TYPE=PAGEHEADER, OBJECT=IMAGE, JUSTIFY=CENTER, IMAGE=WFINTERNALUSEONLY.GIF,$
TYPE=PAGEFOOTER, OBJECT=IMAGE, JUSTIFY=RIGHT, IMAGE=GGLOGO.GIF,$
END
```

生成されたワークシートの先頭ページで、データの背後に透かしが表示されています。このイメージは、ワークシートの各ページに表示されます。

Region	State	Category	Product	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
Midwest	IL	Coffee	Espresso	420439	32237	401477	32416
Midwest	IL	Coffee	Latte	978340	77344	964787	79015
Midwest	IL	Food	Biscotti	378412	29413	385369	30001
Midwest	IL	Food	Croissant	549366	43300	528255	43271
Midwest	IL	Food	Scone	595069	46[illegible]	567231	45091
Midwest	IL	Gifts	Coffee Grinder	233292	18335	241711	19224
Midwest	IL	Gifts	Coffee Pot	204828	15785	208255	16035
Midwest	IL	Gifts	Mug	376[illegible]	30157	388612	30881
Midwest	IL	Gifts	Thermos	1[illegible]	14651	181159	14137
Midwest	MO	Coffee	Espresso	419143	32596	416875	32787
Midwest	MO	Coffee	Latte	966981	77347	921336	77141
Midwest	MO	Food	Biscotti	368077	29188	360403	28764
Midwest	MO	Food	Croissant	613871	48941	592609	49327
Midwest	MO	Food	Scone	481953	37602	478691	36573
Midwest	MO	Gifts	Coffee Grinder	181570	14614	171501	14779
Midwest	MO	Gifts	Coffee Pot	190153	14807	191451	14970
Midwest	MO	Gifts	Mug	343852	27040	324488	26837
Midwest	MO	Gifts	Thermos	195686	15592	189484	15903
Midwest	TX	Coffee	Espresso	455365	36321	439880	36666
Midwest	TX	Coffee	Latte	938245	76932	941677	77501
Midwest	TX	Food	Biscotti	345238	27504	321857	27074
Midwest	TX	Food	Croissant	587887	46941	587869	47050
Midwest	TX	Food	Scone	418398	33170	398437	32112
Midwest	TX	Gifts	Coffee Grinder	204292	16440	200241	16625
Midwest	TX	Gifts	Coffee Pot	204897	16564	214301	16774
Midwest	TX	Gifts	Mug	366337	29521	383050	29374
Midwest	TX	Gifts	Thermos	194319	16344	193367	16779
Northeast	CT	Coffee	Capuccino	158995	12386	141574	11098
Northeast	CT	Coffee	Espresso	279373	22482	299854	23676
Northeast	CT	Coffee	Latte	926052	74623	953855	74427
Northeast	CT	Food	Biscotti	589355	46214	587501	46404
Northeast	CT	Food	Croissant	551489	45847	580168	46335
Northeast	CT	Food	Scone	283874	22378	269221	21038
Northeast	CT	Gifts	Coffee Grinder	169308	13691	159620	13117
Northeast	CT	Gifts	Coffee Pot	208209	15523	197051	15190
Northeast	CT	Gifts	Mug	392967	31728	424333	32415

For Internal Use Only

テーブル範囲構文を使用した Excel テーブルの作成

スタイルシート内で TYPE=DATA 宣言の IN-XLSXTABLE 属性を使用して、Excel ワークシート内に Excel テーブルを作成できます。選択したスタイルシートに IN-XLSXTABLE 属性を含め、プロシジャに適用することができます。

IN-XLSXTABLE 属性は、FORMAT XLSX で生成された XLSX 表形式レポートを Excel テーブルに自動的に変換します。

Excel テーブルを作成すると、出力には Microsoft Excel テーブルを使用した場合の機能および利点のすべてが含められます。結果の Excel テーブルには、フィルタ、ソート、スタイルを簡単に適用することができます。

作成される Excel テーブルには、データ範囲全体が含められます。デフォルト設定では、列のタイトルは Excel テーブルの列見出しです。タイトルが見つからない場合、テーブル列見出しとしてデフォルト設定の列名が生成されます。

注意：複数シートのブックを生成するリクエストは、現在サポートされていません。これらのリクエストでは、IN-XLSXTABLE の範囲はオフに設定され、標準の XLSX 出力に戻されます。

テーブル範囲構文を使用した Excel テーブルの作成

テーブル範囲構文を使用して Excel テーブルを作成するには、テキストエディタを使用して、プロシジャに次のスタイルシート構文を含めます。この構文は、ブック内の 1 つのワークシートに 1 つのテーブル範囲を設定します。

```
TYPE=DATA, IN-XLSXTABLE='name',$
```

説明

name

目的のテーブル名です。

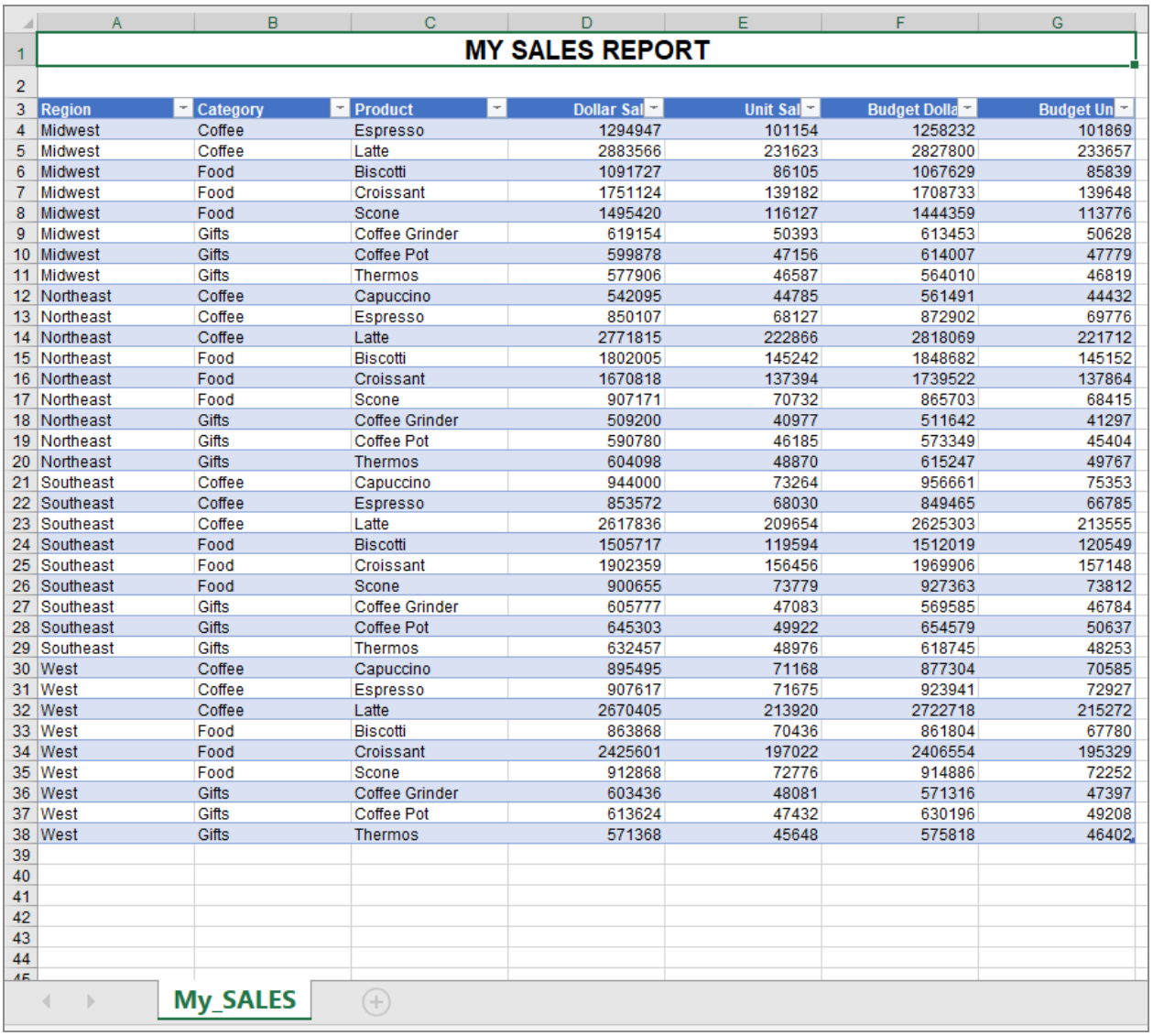
テーブル範囲構文を使用した Excel テーブルの作成

次の手順では、IN-XLSXTABLE 属性を使用して、XLSX テーブルのデータとして含めるテーブル範囲を作成します。テーブル範囲は DATA ノードに割り当てられ、すべてのデータ列およびデータ行が含まれます。デフォルト設定では、列のタイトルは Excel のテーブル列見出しとして含められます。

```
TABLE FILE GGSALES
SUM DOLLARS UNITS BUDDOLLARS BUDUNITS
BY REGION
BY CATEGORY
BY PRODUCT
WHERE PRODUCT NE 'Mug'
ON TABLE SUBHEAD
"MY SALES REPORT"
" "
ON TABLE SET BYDISPLAY ON
ON TABLE HOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, SQUEEZE=1.25, TITLETEXT='My_SALES',$
TYPE=TABHEADING, SIZE=16, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=DATA, IN-XLSXTABLE='Sales_
```

```
Table',$
END
```

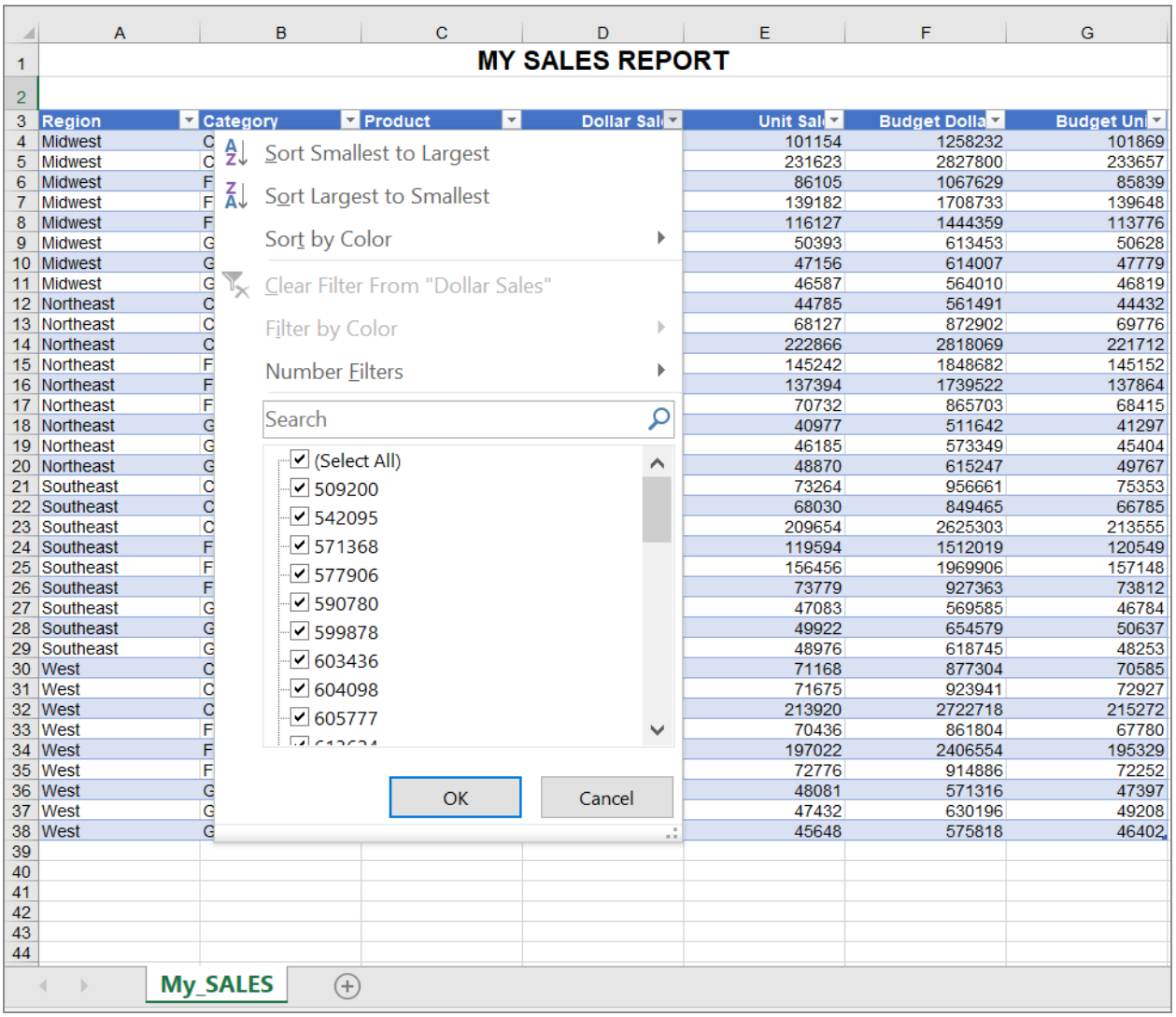
出力結果は次のとおりです。



MY SALES REPORT

Region	Category	Product	Dollar Sal	Unit Sal	Budget Dolla	Budget Un
Midwest	Coffee	Espresso	1294947	101154	1258232	101869
Midwest	Coffee	Latte	2883566	231623	2827800	233657
Midwest	Food	Biscotti	1091727	86105	1067629	85839
Midwest	Food	Croissant	1751124	139182	1708733	139648
Midwest	Food	Scone	1495420	116127	1444359	113776
Midwest	Gifts	Coffee Grinder	619154	50393	613453	50628
Midwest	Gifts	Coffee Pot	599878	47156	614007	47779
Midwest	Gifts	Thermos	577906	46587	564010	46819
Northeast	Coffee	Capuccino	542095	44785	561491	44432
Northeast	Coffee	Espresso	850107	68127	872902	69776
Northeast	Coffee	Latte	2771815	222866	2818069	221712
Northeast	Food	Biscotti	1802005	145242	1848682	145152
Northeast	Food	Croissant	1670818	137394	1739522	137864
Northeast	Food	Scone	907171	70732	865703	68415
Northeast	Gifts	Coffee Grinder	509200	40977	511642	41297
Northeast	Gifts	Coffee Pot	590780	46185	573349	45404
Northeast	Gifts	Thermos	604098	48870	615247	49767
Southeast	Coffee	Capuccino	944000	73264	956661	75353
Southeast	Coffee	Espresso	853572	68030	849465	66785
Southeast	Coffee	Latte	2617836	209654	2625303	213555
Southeast	Food	Biscotti	1505717	119594	1512019	120549
Southeast	Food	Croissant	1902359	156456	1969906	157148
Southeast	Food	Scone	900655	73779	927363	73812
Southeast	Gifts	Coffee Grinder	605777	47083	569585	46784
Southeast	Gifts	Coffee Pot	645303	49922	654579	50637
Southeast	Gifts	Thermos	632457	48976	618745	48253
West	Coffee	Capuccino	895495	71168	877304	70585
West	Coffee	Espresso	907617	71675	923941	72927
West	Coffee	Latte	2670405	213920	2722718	215272
West	Food	Biscotti	863868	70436	861804	67780
West	Food	Croissant	2425601	197022	2406554	195329
West	Food	Scone	912868	72776	914886	72252
West	Gifts	Coffee Grinder	603436	48081	571316	47397
West	Gifts	Coffee Pot	613624	47432	630196	49208
West	Gifts	Thermos	571368	45648	575818	46402

下図のように、出力内のすべての行と列に対して、簡単にフィルタおよびソートを適用することができます。



テーブルレベル構文およびテーブル範囲構文を使用した Excel テーブルの作成

次の手順では、IN-XLSXTABLE 属性を使用して、XLSX テーブルのデータとして含めるテーブル範囲を作成します。テーブル範囲は DATA ノードに割り当てられ、すべてのデータ列と行が含められます。

このリクエストには、中央揃えされたテーブル見出し、赤色のレポート脚注、太字の総合計も含まれています。

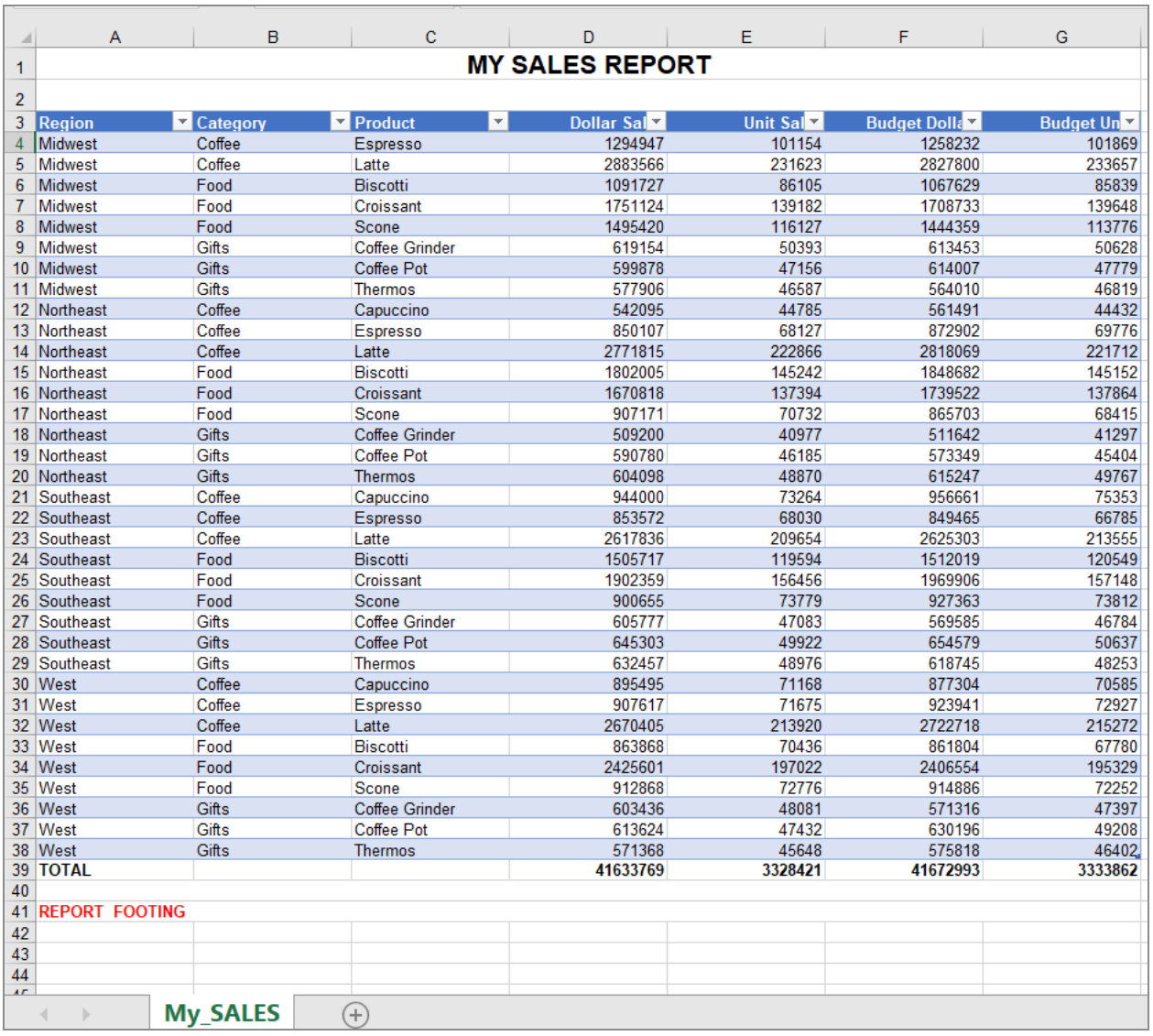
```
TABLE FILE GGSALES
SUM DOLLARS UNITS BUDDOLLARS BUDUNITS
BY REGION
BY CATEGORY
BY PRODUCT
WHERE PRODUCT NE 'Mug'
ON TABLE SUBHEAD
"MY SALES REPORT"
" "
FOOTING
" "
"REPORT  FOOTING"
ON TABLE SUBTOTAL
ON TABLE SET BYDISPLAY ON
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, SQUEEZE=1.25, TITLETEXT='My_SALES',$
TYPE=TABHEADING, SIZE=16, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=HEADING, STYLE=BOLD, SIZE=12,$
TYPE=FOOTING, STYLE=BOLD, COLOR=RED,$
TYPE=GRANDTOTAL, STYLE=BOLD,$
TYPE=DATA, IN-XLSXTABLE='Sales_Table',$
END
```

注意：テーブルレベルの中間見出し、中間脚注、見出し、脚注、総合計は、Excel テーブルの範囲外 (必要に応じて前または後) に表示されます。このリクエストでは、レポートの見出しが除外されることと同様に、テーブルのデータ範囲から脚注および総合計が除外されます。

テーブル範囲 (A3:G38) はデータ部分で、列タイトルはテーブルの列見出しです。

行 1、39、および 41 は、見出し、総合計、脚注です。これらの行はテーブルの外側にあります。

出力結果は次のとおりです。



	A	B	C	D	E	F	G
1	MY SALES REPORT						
2							
3	Region	Category	Product	Dollar Sal	Unit Sal	Budget Dolla	Budget Un
4	Midwest	Coffee	Espresso	1294947	101154	1258232	101869
5	Midwest	Coffee	Latte	2883566	231623	2827800	233657
6	Midwest	Food	Biscotti	1091727	86105	1067629	85839
7	Midwest	Food	Croissant	1751124	139182	1708733	139648
8	Midwest	Food	Scone	1495420	116127	1444359	113776
9	Midwest	Gifts	Coffee Grinder	619154	50393	613453	50628
10	Midwest	Gifts	Coffee Pot	599878	47156	614007	47779
11	Midwest	Gifts	Thermos	577906	46587	564010	46819
12	Northeast	Coffee	Capuccino	542095	44785	561491	44432
13	Northeast	Coffee	Espresso	850107	68127	872902	69776
14	Northeast	Coffee	Latte	2771815	222866	2818069	221712
15	Northeast	Food	Biscotti	1802005	145242	1848682	145152
16	Northeast	Food	Croissant	1670818	137394	1739522	137864
17	Northeast	Food	Scone	907171	70732	865703	68415
18	Northeast	Gifts	Coffee Grinder	509200	40977	511642	41297
19	Northeast	Gifts	Coffee Pot	590780	46185	573349	45404
20	Northeast	Gifts	Thermos	604098	48870	615247	49767
21	Southeast	Coffee	Capuccino	944000	73264	956661	75353
22	Southeast	Coffee	Espresso	853572	68030	849465	66785
23	Southeast	Coffee	Latte	2617836	209654	2625303	213555
24	Southeast	Food	Biscotti	1505717	119594	1512019	120549
25	Southeast	Food	Croissant	1902359	156456	1969906	157148
26	Southeast	Food	Scone	900655	73779	927363	73812
27	Southeast	Gifts	Coffee Grinder	605777	47083	569585	46784
28	Southeast	Gifts	Coffee Pot	645303	49922	654579	50637
29	Southeast	Gifts	Thermos	632457	48976	618745	48253
30	West	Coffee	Capuccino	895495	71168	877304	70585
31	West	Coffee	Espresso	907617	71675	923941	72927
32	West	Coffee	Latte	2670405	213920	2722718	215272
33	West	Food	Biscotti	863868	70436	861804	67780
34	West	Food	Croissant	2425601	197022	2406554	195329
35	West	Food	Scone	912868	72776	914886	72252
36	West	Gifts	Coffee Grinder	603436	48081	571316	47397
37	West	Gifts	Coffee Pot	613624	47432	630196	49208
38	West	Gifts	Thermos	571368	45648	575818	46402
39	TOTAL			41633769	3328421	41672993	3333862
40							
41	REPORT FOOTING						
42							
43							
44							

IN-XSLXTABLE 属性使用上の注意

- 複数シートのブックを生成するリクエストは、現在サポートされていません。これらのリクエストでは、IN-XLSXTABLE の範囲はオフに設定され、標準の XLSX 出力に戻されます。
- 選択したスタイルシートに IN-XLSXTABLE 属性を含め、プロシジャに適用することができます。
- ソートフィールドで中間見出し、中間脚注、中間合計を使用すると、Excel テーブルに表示されます。

- 中間見出し、中間脚注、中間合計の行では、標準 XLSX 出力の場合とは異なり、セルが結合されることはありません。
- 中間見出し、中間脚注、中間合計を使用する Excel テーブル内で値をソートする場合、これらの行は他のデータ行と同様に扱われるため、対応するデータ行とともには保持されません。
- テーブルの途中でデータの整合性を損なうため、Excel テーブルを使用する場合は、中間見出し、中間脚注、中間合計を使用しないことをお勧めします。

Excel の名前付け規則に関する注意事項

次のトピックでは、Excel とテーブルの一般的な名前付け規則について説明します。

Excel の一般的な名前付け規則

- **最大 255 バイト**　Excel の名前として、最大 255 バイトまで使用できます。
- **有効な文字を使用する**　Excel の名前は、文字、アンダースコア (_)、または円記号 (¥) で始まる必要があります。名前の残りの部分には、文字、数字、ピリオド (.)、およびアンダースコア (_) を使用します。

  注意：Excel の名前として、C、c、R、r は使用しないでください。 これらを [名前] または [移動] ボックスに入力すると、アクティブセルの列または行を選択するためのショートカットとして機能します。
- **スペースを使用しない**　Excel の名前には、スペースおよびほとんどの句読点文字を含めることはできません。DeptSales、Sales_Tax、First.Quarter など、スペースを使用しない名前を指定する方法、または単語の区切りとしてアンダースコア (_)、ピリオド (.) を使用する方法を検討してください。
- **大文字と小文字の区別をしない**　Excel の名前では大文字と小文字が区別されないため、大文字と小文字が異なるだけの 2 つの値が存在する場合、これらの 2 つの値は同一値と見なされます。たとえば、Excel では「SALES」と「Sales」は同一値と見なされます。
- **セル参照を使用しない**　名前は、Z$100 や R1C1 などのセル参照と同一にすることはできません。

Excel テーブル名の追加規則

- **一意のテーブル名を使用する**　各テーブルには、ブック内で一意の名前を付ける必要があります。重複するテーブル名は使用できません。

- **大文字と小文字の区別をしない**　Excel の名前では、大文字と小文字が区別されないため、Sales という名前を入力し、同じブックに SALES という名前がすでにに存在する場合は、一意の名前を選択するよう要求されます。
- **有効な文字を使用する**　テーブル名には、文字、数字、アンダースコア (_) のみを使用できます。テーブル名は、文字またはアンダースコア (_) で始める必要があります。

テンプレートによる Excel XLSX ワークシートの作成

Excel テンプレートに基づいて XLSX レポート出力を生成することができます。この機能を使用して、複数のワークシートで構成されたブックに WebFOCUS レポートを統合することができます。任意の Excel テンプレートを使用して、WebFOCUS レポートが挿入された新しいブックを生成することができます。

次の Excel ファイルタイプは、XLSX ブックを生成するためのテンプレートファイルとして使用することができます。

テンプレートファイルタイプ	生成されるブック
テンプレート (.xltx)	ブック (.xlsx)
マクロを有効にしたテンプレート (.xltm)	マクロを有効にしたブック (.xlsm)
ブック (.xlsx)	ブック (.xlsx)
マクロを有効にしたブック (.xlsm)	マクロを有効にしたブック (.xlsm)

WebFOCUS XLSX TEMPLATE フォーマットでは、Excel 基本テンプレート(.xltx) ファイルがサポートされます。これらのテンプレートには、マクロなどの Microsoft がアクティブコンテンツとして見なすコンテンツを含めることはできません。また、アクティブコンテンツが含まれたテンプレート (.xltm/.xlsm) もサポートされます。マクロを有効にしたテンプレート (.xltm) を使用すると、アクティブコンテンツ (マクロ、VB スクリプト) をテンプレートに含めることができます。

WebFOCUS EXL07 TEMPLATE プロシジャは、定義されたテンプレートファイルに基づいて、標準 Excel 拡張子の Excel ブックを生成します。WebFOCUS リクエストにより、テンプ

レートブック内の既存のワークシートが置き換えられます。また、他のワークシートで定義されている数式または参照が置き換え後のワークシートのセルに関連付けられている場合、これらの数式または参照は、ブックを開いたときに自動的に更新されます。

テンプレート機能により既存のワークシートが置き換えられるため、指定したワークシートがテンプレートブックに存在する必要があります。テンプレート内の置き換えられたワークシートにコンテンツを追加した場合、そのコンテンツは保持されません。他のワークシートに格納されているコンテンツは保持され、更新されます。

プロシジャに INRANGES 属性を使用して名前付きセル範囲を定義することで、他のワークシートで参照できるセルグループを指定することができます。

Excel 2007/2010 テンプレートを生成するには、任意のブックを .xltx 拡張子で保存します。テンプレートファイルは、デフォルトの Excel テンプレートディレクトリではなく、アプリケーションパス (EDAPATH または APPPATH) に格納しておく必要があります。これにより、プロシジャの実行時に Reporting Server からテンプレートファイルへのアクセスが可能になります。

EXL07 TEMPLATE 機能は、基本的な EXL07 フォーマットレポートでサポートされます。このバージョンの EXL07 TEMPLATE では、FORMULA、EXL97、EXCEL、複合 Excel レポートの機能はサポートされません。

ほとんどの場合、.mht ファイルとして作成された既存の Excel 2003/2000 テンプレートは、簡単に Excel 2013/2010/2007 テンプレートに変換することができます。これを行うには、Excel 2013/2010/2007 で .mht ファイルを開き、そのファイルを Excel テンプレートファイル (.xltx)、またはマクロを有効にしたファイル (.xltm) として保存します。Excel の数式および機能は、これらのテンプレートに保持されます。マクロなどのアクティブコンテンツを保持するには、.xltm を使用します。この新しい XLTX テンプレートは、XLSX プロシジャで使用することができます。

サポート対象のテンプレートファイルタイプを使用した XLSX レポートの作成

テンプレートファイル名の属性が強化され、ファイルの拡張子を含めることが可能になりました。これにより、広範囲のテンプレートファイルタイプがサポートされます。テンプレート名に拡張子を指定しない場合、デフォルト設定でファイルの拡張子は .XLTX になります。

```
ON TABLE PCHOLD FORMAT XLSX TEMPLATE template_name SHEETNUMBER n
```

説明

template_name

WebFOCUS Reporting Server アプリケーションディレクトリの検索パスに存在する Excel テンプレートファイル (ブック) の名前を、最大 64 バイトのファイル名と拡張子で指定します。たとえば、IPOLICY.XLTX、PRINTSHEETS.XLTM、DASHBOARD.XLSM のように指定します。拡張子を指定しない場合、デフォルト値の .XLTX になります。

n

テンプレートファイル (ブック) 内で置き換える既存の Excel ワークシート番号です。

XLSX TEMPLATE 使用上の注意

WebFOCUS プロシジャで使用するブックテンプレートには、有効なワークシートを含めておく必要があります。

- 更新するワークシートがブック内に存在する必要があります。これは、WebFOCUS がワークシートを挿入する代わりに、既存のワークシートを置き換えるためです。指定したワークシートが存在しない場合、プロシジャがエラーを返します。
- テンプレートファイルでは、ブック内のワークシートのうち、少なくとも 1 つのワークシートのセルに有効な値 (ブランク、その他の値) が存在する必要があります。空のワークシートのみで構成されたテンプレートのワークシートを置き換えるには、いずれかのワークシート内でセルの 1 つをブランクに置き換えた上でテンプレートを保存します。これにより、ワークシートがインスタンス化されるため、WebFOCUS がこのワークシートにアクセスして更新することが可能になります。
- サポートされるファイル名の長さが、64 バイトにまで拡張されました。これより長いファイル名のテンプレートがプロシジャで参照されている場合、メッセージが生成されます。

Excel 目次レポートの作成

Excel 目次機能 (BYTOC) を使用すると、WebFOCUS レポートの先頭 BY フィールドの値ごとにレポートのインスタンスが作成され、それぞれが個別のワークシートとして生成されます。

Excel 目次機能の使用

BYTOC は、3 通りの方法で呼び出すことができます。

```
ON TABLE {HOLD|PCHOLD} FORMAT XLSX BYTOC
```

```
SET COMPOUND=BYTOC
```

```
ON TABLE SET COMPOUND BYTOC
```

BYTOC レポートでは、レポートの先頭 BY フィールドの値ごとに個別のワークシートが生成されるため、レポートには少なくとも 1 つの BY フィールドを含める必要があります。先頭 BY フィールドが、NOPRINT フィールドになる場合があります。

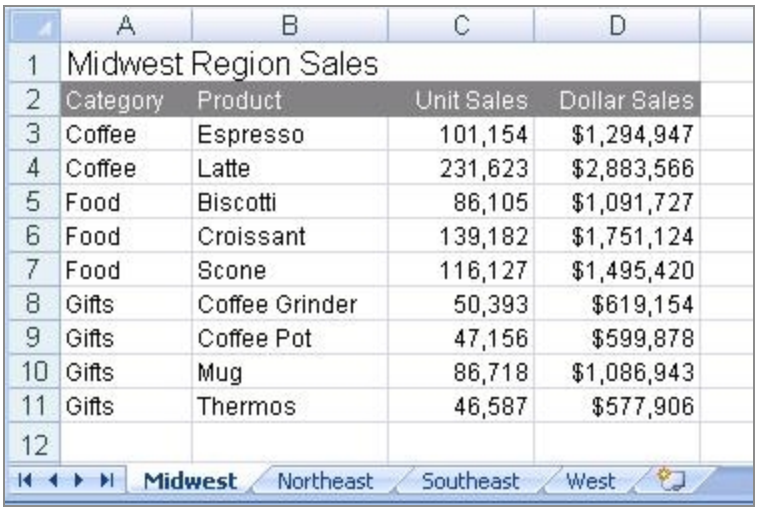
XLSX TEMPLATE フォーマットでは BYTOC 機能はサポートされません。

単純な BYTOC レポートの作成

次のリクエストは、GGSALES データソースに対して実行され、REGION ソートフィールドの値ごとに個別のタブを作成します。

```
TABLE FILE GGSALES
SUM UNITS/D12C DOLLARS/D12CM
BY REGION NOPRINT
BY CATEGORY
BY PRODUCT
HEADING
"<REGION Region Sales"
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET BYDISPLAY ON
ON TABLE SET COMPOUND BYTOC
ON TABLE SET STYLE *
TYPE=REPORT, FONT=ARIAL, SIZE=9,$
TYPE=HEADING, SIZE=12,$
TYPE=TITLE, BACKCOLOR=GREY, COLOR=WHITE,$
ENDSTYLE
END
```

出力結果は次のとおりです。



	A	B	C	D
1	Midwest Region Sales			
2	Category	Product	Unit Sales	Dollar Sales
3	Coffee	Espresso	101,154	$1,294,947
4	Coffee	Latte	231,623	$2,883,566
5	Food	Biscotti	86,105	$1,091,727
6	Food	Croissant	139,182	$1,751,124
7	Food	Scone	116,127	$1,495,420
8	Gifts	Coffee Grinder	50,393	$619,154
9	Gifts	Coffee Pot	47,156	$599,878
10	Gifts	Mug	86,718	$1,086,943
11	Gifts	Thermos	46,587	$577,906
12				

Midwest | Northeast | Southeast | West

ワークシートの名前指定

- ワークシートのタブ名は、現在のワークシート上のデータに対応する BY フィールドの値になります。ユーザがスタイルシートに TITLETEXT キーワードを指定した場合は、その名前は無視されます。
- Excel では、ワークシートのタイトルの最大長は 31 バイトに限定されています。特殊文字の「:」、「?」、「*」、「/」を使用することはできません。
- データフィールドを分割用 BY フィールドとして使用する場合は、「/」の代わりに「-」を使用することができます。「-」は、Excel のタブ名で有効になります。ただし、「/」を使用すると、WebFOCUS により「-」に変更されます。

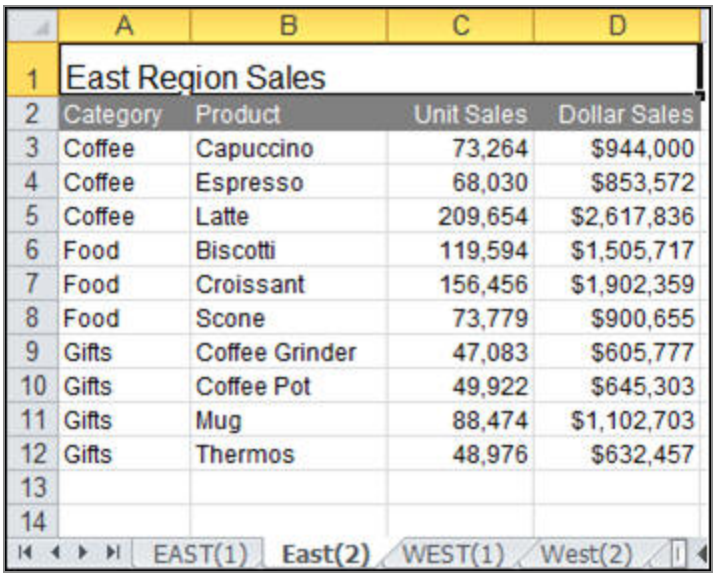
大文字と小文字が区別されたデータ使用時の XLSX ワークシート名

Excel では、各ワークシートに一意の名前を付ける必要があります。Excel では大文字と小文字が区別されないため、大文字と小文字が異なるだけの 2 つの値が存在する場合、これらの 2 つの値は同一値と見なされます。たとえば、Excel では「WEST」と「West」は同一値と見なされます。WebFOCUS XLSX フォーマットでは、重複名が識別され、ワークシート名に一意の番号が追加されます。これにより、Excel で両方のワークシートが保持されます。

デフォルト設定では、WebFOCUS のソート処理で大文字と小文字が区別されるため、大文字と小文字が異なるだけの同一フィールド値は、ソート (BY) フィールドとして使用された場合

に 2 つの異なる値と見なされます。Excel BYTOC レポートでは、WebFOCUS が生成する各ワークシートの名前には、大文字と小文字が区別されてソートされた、先頭ソート (BY) フィールドの値が使用されます。この動作に対応するために XLSX が強化され、データ内で重複するワークシート名が検出された場合、カウンタが追加され、各ワークシートに一意の名前が割り当てられます。

たとえば、レポートに Region 値として「EAST」および「East」が含まれている場合、下図のように、各ワークシートの名前は「EAST(1)」および「East(2)」と表示されます。



	A	B	C	D
1	East Region Sales			
2	Category	Product	Unit Sales	Dollar Sales
3	Coffee	Capuccino	73,264	$944,000
4	Coffee	Espresso	68,030	$853,572
5	Coffee	Latte	209,654	$2,617,836
6	Food	Biscotti	119,594	$1,505,717
7	Food	Croissant	156,456	$1,902,359
8	Food	Scone	73,779	$900,655
9	Gifts	Coffee Grinder	47,083	$605,777
10	Gifts	Coffee Pot	49,922	$645,303
11	Gifts	Mug	88,474	$1,102,703
12	Gifts	Thermos	48,976	$632,457
13				
14				

EAST(1) East(2) WEST(1) West(2)

オーバーフローワークシート機能による Excel 2007/2010 での行数制限への対処

Excel 2007/2010 ワークシートでサポートされる最大行数は 1,048,576 行 (1 メガバイト) です。WebFOCUS レポートから XLSX 出力ファイルを作成する際は、生成される行数が、この最大数を超える場合があります。

制限を超える行を「オーバーフローワークシート」と呼ばれる新しいワークシートに移動することができます。オーバーフローワークシート名は、それぞれ元のワークシート名に番号を追加したものになります。

また、オーバーフローワークシート機能を有効にした場合、ワークシートに格納する行の最大数を設定することができます。デフォルト設定の最大行数値は、LINES パラメータ のデフォルト値 (57) に設定されます。

注意：XLSX 出力を生成する際に WebFOCUS のページ見出しコマンドおよびページ脚注コマンドを使用すると、デフォルト設定でワークシート見出しおよびワークシート脚注のみが生成されます。

オーバーフローワークシート機能の有効化

WebFOCUS スタイルシートに ROWOVERFLOW 属性を追加します。

```
TYPE=REPORT, ROWOVERFLOW={ON|OFF|PBON}, [ROWLIMIT={n|MAX},]$
```

説明

ON

オーバーフローワークシート機能を有効にします。

OFF

オーバーフローワークシート機能を無効にします。デフォルト値は OFF です。

PBON

WebFOCUS 改ページを挿入します。これにより、ワークシートの行の正しい位置にページ見出し、ページ脚注、列タイトルが表示されます。このオプションでは、WebFOCUS 改ページが挿入された場合でも新しいワークシートは開始されません。

ROWLIMIT=n

ワークシートに含める行数のターゲット値を n 行数に設定します。デフォルト値は LINES の値 (デフォルト設定では 57) です。

ROWLIMIT=MAX

ワークシートに含める行数のターゲット値を設定します。XLSX 出力では 1,048,000 行を指定します。

XLSX オーバーフローワークシート使用上の注意

- レポート見出しは、最初のワークシートの先頭に、1 つだけ表示されます。レポート脚注は、最後のオーバーフローワークシートの末尾に、1 つだけ表示されます。
- PBON 設定を使用しない場合、ワークシートの見出しおよび列タイトルは、元のワークシートおよび後続の各オーバーフローシートの上部に繰り返し表示されます。ワークシートの脚注は、元のワークシートおよび後続の各オーバーフローシートの下部に表示されます。データ値は、標準的な新しいページ同様、各オーバーフローワークシートの先頭のデータ行に表示されます。
- レポートの合計行は、最後のオーバーフローワークシートの下部で、最後のページとテーブル脚注のすぐ上に表示されます。
- 中間見出し、中間脚注、中間合計行は、通常どおり、データフロー内に表示されます。シート内のグループを保持するために、特別な処理は適用されません。
- ROWOVERFLOW=PBON の場合、WebFOCUS コマンドで改ページが挿入されるたびに、ページ見出し、ページ脚注、列タイトルがワークシートに表示されます。
- XLSX 出力では、スタイルシートで ROWOVERFLOW 属性が指定され、ROWLIMIT が 1 メガバイトを超える場合、次のメッセージが表示され、出力ファイルは生成されません。

  ```
  (FOC3338) EXCEL XLSX ワークシートの最大行数は 1048576 です。
  ```

- 数式参照を含む出力タイプ (EXL2K PIVOT および EXL2K FORMULA) はサポートされません。これは、新しいオーバーフローワークシートでの位置を反映するよう数式参照が自動的に更新されないためです。
- オーバーフローワークシート機能は、行のみに適用され、列には適用されません。Excel 2007/2010 の制限値である 16,384 列を超える列が生成された場合、新しいワークシートは自動的に生成されません。
- ROWOVERFLOW は、XLSX の BYTOC レポートでサポートされます。
- Excel では名前付きセル範囲を複数のワークシートにわたって指定できないため、出力先のブックに名前付きセル範囲を定義する IN-RANGES 句は、ROWOVERFLOW 機能ではサポートされません。これらの両方が同一のリクエストに存在する場合、ROWOVERFLOW が優先され、IN-RANGES 句は無視されます。

オーバーフローワークシートの作成

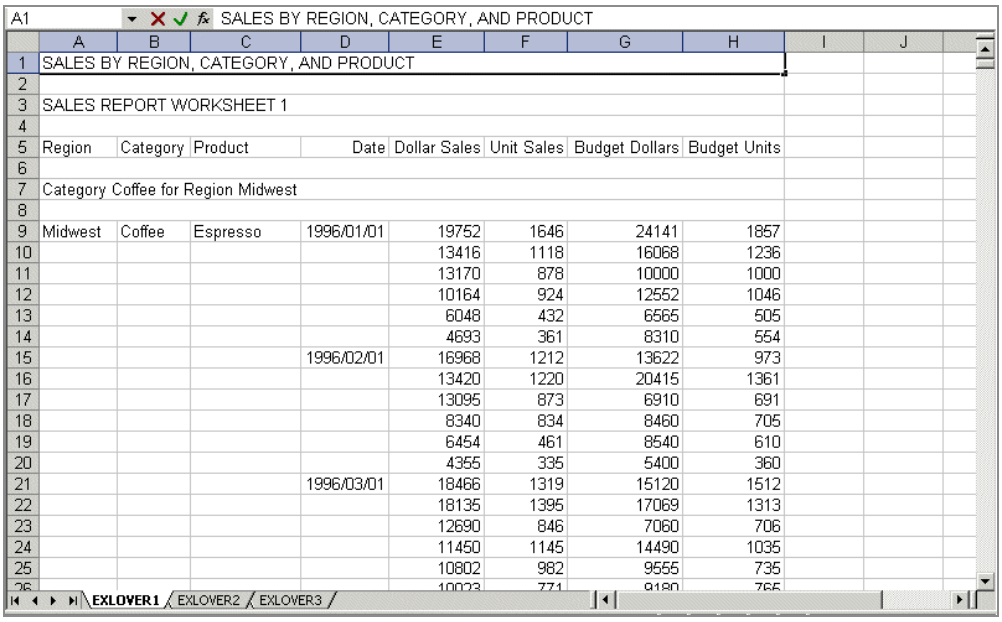
次のリクエストは、オーバーフローワークシートが含まれた XLSX レポート出力を作成します。スタイルシートの ROWOVERFLOW=ON に属性を設定することにより、オーバーフローワークシート機能が有効になります。この属性が存在しない場合は、3 つではなく 1 つのワークシートが生成されます。

```
TABLE FILE GGSALES
-* ****Report Heading****
ON TABLE SUBHEAD
"SALES BY REGION, CATEGORY, AND PRODUCT"
" "
-* ****Worksheet Heading****
HEADING
"SALES REPORT WORKSHEET <TABPAGENO"
" "
-* ****Worksheet Footing****
FOOTING
" "
"END OF WORKSHEET <TABPAGENO"
PRINT DOLLARS UNITS BUDDOLLARS BUDUNITS
BY REGION
BY CATEGORY
BY PRODUCT
BY DATE
```

```
-* ****Subfoot****
ON REGION SUBFOOT
" "
" End of Region <REGION"
" "
-* ****Subhead****
ON REGION SUBHESD
" "
"Category <CATEGORY for Region <REGION"
" "
-* ****Report Footing****
ON TABLE SUBFOOT
" "
"END OF REPORT"
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=EXLOVER, ROWOVERFLOW=ON, ROWLIMIT=2000,$
```

```
ENDSTYLE
END
```

レポート見出しは最初のワークシートのみに表示され、ページ見出しおよび列タイトルは各ワークシートに表示されます。また、中間見出しおよび中間脚注は、関連するソートフィールドの値が変更されるたびに表示されます。下図は、最初のワークシートの上部を示しています。レポート見出し、ページ見出し、列タイトル、最初の中間見出しが表示されています。



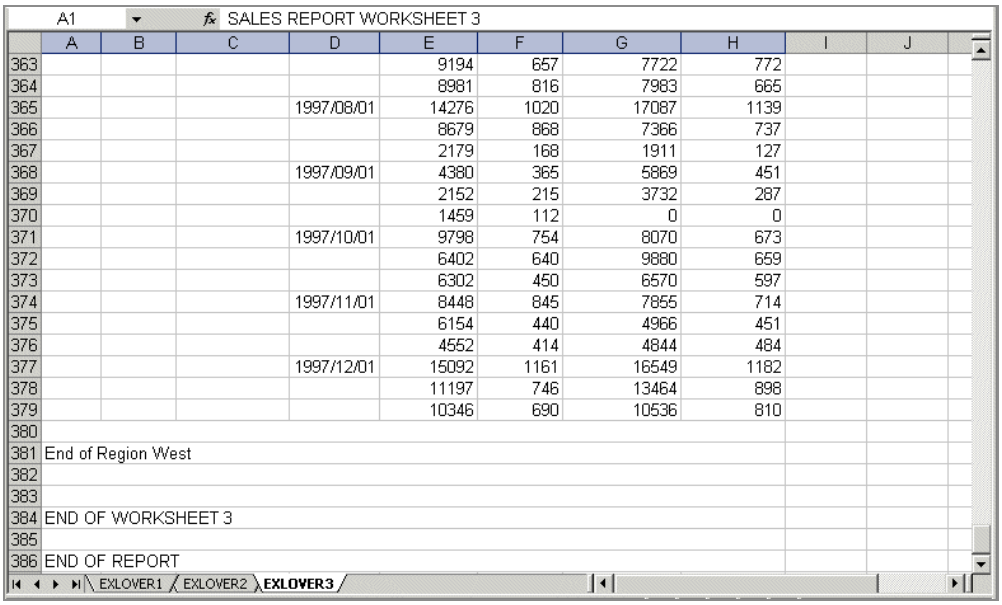
A1 SALES BY REGION, CATEGORY, AND PRODUCT

	A	B	C	D	E	F	G	H
1	SALES BY REGION, CATEGORY, AND PRODUCT							
2								
3	SALES REPORT WORKSHEET 1							
4								
5	Region	Category	Product	Date	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
6								
7	Category Coffee for Region Midwest							
8								
9	Midwest	Coffee	Espresso	1996/01/01	19752	1646	24141	1857
10					13416	1118	16068	1236
11					13170	878	10000	1000
12					10164	924	12552	1046
13					6048	432	6565	505
14					4693	361	8310	554
15				1996/02/01	16968	1212	13622	973
16					13420	1220	20415	1361
17					13095	873	6910	691
18					8340	834	8460	705
19					6454	461	8540	610
20					4355	335	5400	360
21				1996/03/01	18466	1319	15120	1512
22					18135	1395	17069	1313
23					12690	846	7060	706
24					11450	1145	14490	1035
25					10802	982	9555	735

EXLOVER1 EXLOVER2 EXLOVER3

スタイルシートの TITLETEXT 属性で「EXLOVER」という名前を指定したため、3 つのワークシートは「EXLOVER1」「EXLOVER2」「EXLOVER3」という名前で生成されます。
TITLETEXT 属性を指定しない場合、ワークシート名は、「sheet1」「sheet2」「sheet3」になります。

ワークシートの脚注は各ワークシートの下部に表示され、レポート脚注は最後のワークシートの下部に表示されます。下図は、最後のワークシートの下部を示しています。最後の中間脚注、ページ脚注、レポート脚注が表示されています。



	A	B	C	D	E	F	G	H	I	J
363					9194	657	7722	772		
364					8981	816	7983	665		
365				1997/08/01	14276	1020	17087	1139		
366					8679	868	7366	737		
367					2179	168	1911	127		
368				1997/09/01	4380	365	5869	451		
369					2152	215	3732	287		
370					1459	112	0	0		
371				1997/10/01	9798	754	8070	673		
372					6402	640	9880	659		
373					6302	450	6570	597		
374				1997/11/01	8448	845	7855	714		
375					6154	440	4966	451		
376					4552	414	4844	484		
377				1997/12/01	15092	1161	16549	1182		
378					11197	746	13464	898		
379					10346	690	10536	810		
380										
381	End of Region West									
382										
383										
384	END OF WORKSHEET 3									
385										
386	END OF REPORT									

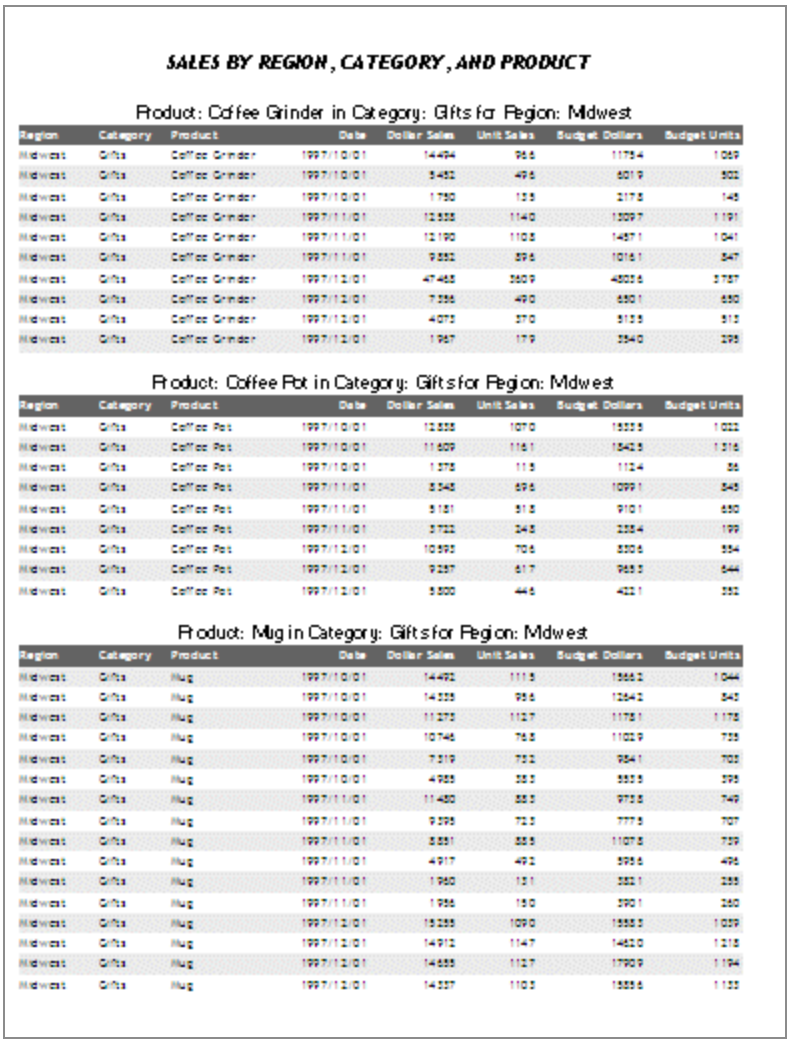
WebFOCUS 改ページを含むオーバーフローワークシートの作成

次のリクエストは、オーバーフローワークシートが含まれた XLSX レポート出力を作成します。スタイルシートの ROWOVERFLOW=PBON 属性によりオーバーフロー機能が有効になり、ROWLIMIT=250 により各ワークシートの最大行数がおよそ 250 に設定されます。この属性を指定しない場合、1 つのワークシートのみが生成されます。PRODUCT ソート句により、改ページが指定されます。

```
TABLE FILE GGSALES
-* ****Report Heading****
ON TABLE SUBHEAD
"SALES BY REGION, CATEGORY, AND PRODUCT"
" "
PRINT DOLLARS UNITS BUDDOLLARS BUDUNITS
BY REGION
BY HIGHEST CATEGORY
BY PRODUCT PAGE-BREAK
BY DATE
WHERE DATE GE '19971001'
-* ****Page Heading****
HEADING
" Product: <PRODUCT in Category: <CATEGORY for Region: <REGION"
-* ****Page Footing****
```

```
FOOTING
" "
-* ****Report Footing****
ON TABLE SUBFOOT
" "
"END OF REPORT"
ON TABLE SET BYDISPLAY ON
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/javaassist/intl/EN/ENIADefault_combine.sty,$
TITLETEXT=EXLOVER, ROWOVERFLOW=PBON, ROWLIMIT=250,
$
ENDSTYLE
END
```

レポート見出しは最初のワークシートのみに表示され、ページ見出し、ページ脚注、列タイトルは各ワークシートの WebFOCUS 改ページごとに (製品が変わるたびに) 表示されます。また、中間見出しおよび中間脚注は、関連するソートフィールドの値が変わるたびに表示されます。下図は、最初のワークシートの上部を示しています。



SALES BY REGION, CATEGORY, AND PRODUCT

Product: Coffee Grinder in Category: Gifts for Region: Midwest

Region	Category	Product	Date	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
Midwest	Gifts	Coffee Grinder	1997/10/01	14494	966	11754	1069
Midwest	Gifts	Coffee Grinder	1997/10/01	5452	496	6019	603
Midwest	Gifts	Coffee Grinder	1997/10/01	1750	135	2175	145
Midwest	Gifts	Coffee Grinder	1997/11/01	12555	1140	13097	1191
Midwest	Gifts	Coffee Grinder	1997/11/01	12190	1105	14671	1041
Midwest	Gifts	Coffee Grinder	1997/11/01	9852	896	10161	847
Midwest	Gifts	Coffee Grinder	1997/12/01	47468	3609	48036	3787
Midwest	Gifts	Coffee Grinder	1997/12/01	7356	490	6801	630
Midwest	Gifts	Coffee Grinder	1997/12/01	4073	370	5133	513
Midwest	Gifts	Coffee Grinder	1997/12/01	1967	179	3540	295

Product: Coffee Pot in Category: Gifts for Region: Midwest

Region	Category	Product	Date	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
Midwest	Gifts	Coffee Pot	1997/10/01	12555	1070	15555	1022
Midwest	Gifts	Coffee Pot	1997/10/01	11609	1161	13425	1376
Midwest	Gifts	Coffee Pot	1997/10/01	1375	115	1124	86
Midwest	Gifts	Coffee Pot	1997/11/01	8548	696	10991	845
Midwest	Gifts	Coffee Pot	1997/11/01	5151	515	9101	650
Midwest	Gifts	Coffee Pot	1997/11/01	3722	248	2554	199
Midwest	Gifts	Coffee Pot	1997/12/01	10593	706	8506	554
Midwest	Gifts	Coffee Pot	1997/12/01	9257	617	9653	644
Midwest	Gifts	Coffee Pot	1997/12/01	5500	446	4021	352

Product: Mug in Category: Gifts for Region: Midwest

Region	Category	Product	Date	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
Midwest	Gifts	Mug	1997/10/01	14492	1115	15663	1044
Midwest	Gifts	Mug	1997/10/01	14335	956	12642	843
Midwest	Gifts	Mug	1997/10/01	11275	1127	11761	1176
Midwest	Gifts	Mug	1997/10/01	10745	765	11029	735
Midwest	Gifts	Mug	1997/10/01	7319	732	9541	702
Midwest	Gifts	Mug	1997/10/01	4955	353	5535	395
Midwest	Gifts	Mug	1997/11/01	11480	853	9755	749
Midwest	Gifts	Mug	1997/11/01	9395	723	7775	707
Midwest	Gifts	Mug	1997/11/01	8851	885	11075	739
Midwest	Gifts	Mug	1997/11/01	4917	492	5956	496
Midwest	Gifts	Mug	1997/11/01	1960	131	3821	285
Midwest	Gifts	Mug	1997/11/01	1956	150	3901	260
Midwest	Gifts	Mug	1997/12/01	15285	1090	15553	1039
Midwest	Gifts	Mug	1997/12/01	14912	1147	14610	1218
Midwest	Gifts	Mug	1997/12/01	14655	1127	17909	1194
Midwest	Gifts	Mug	1997/12/01	14227	1103	15556	1133

XLSX による Excel 複合レポート

Excel 複合レポート機能を使用すると、XLSX 出力フォーマットで、複数のワークシートレポートで構成された複合ブックが生成されます。

標準の複合レイアウト構文を使用して、XLSX 複合ブックを生成することができます。デフォルト設定では、ドキュメントの各ページレイアウトは、別のワークシートとして作成されます。ワークシート内の項目は、セル固有の配置プロパティを使用して配置することができます。

Excel 複合レポートの各コンポーネントには、標準のテーブル、目次 (BYTOC) レポート、ROWOVERFLOW を有効にしたレポートを含めることができます。

コンポーネントグラフは、イメージとしてワークシートに追加されます。

XLSX による Excel 複合レポートの使用上の注意

- イメージおよびグラフはコンポーネント内に埋め込むことができますが、ページレイアウトの複合レイアウト構文で定義されたイメージおよび描画オブジェクト (線、ボックス、テキスト) は、生成されたブックに挿入されません。
- XLSX 複合ブック内の Excel 見出しおよび脚注では、グラフおよびイメージはサポートされません。

注意：複数のテーブルが作成されるため、WebFOCUS は各タブ名が一意であることを確認します。

コンポーネントを含む複合 Excel レポート

```
ENGINE INT CACHE SET ON
SET PAGE-NUM=NOLEAD
SET SQUEEZE=ON
-DEFAULTH &WF_HTMLENCODE=ON;
SET HTMLENCODE=&WF_HTMLENCODE

SET HTMLCSS=ON
-DEFAULTH &WF_EMPTYREPORT=ON;
SET EMPTYREPORT=&WF_EMPTYREPORT

-DEFAULTH &WF_EMBEDHEADING=ON;
SET EMBEDHEADING=&WF_EMBEDHEADING

SET GRAPHDEFAULT=OFF
-DEFAULTH &WF_UNITS=INCHES;
SET UNITS=&WF_UNITS


COMPOUND LAYOUT PCHOLD FORMAT XLSX
SECTION=Section_1, LAYOUT=ON, PAGESIZE=LETTER, ORIENTATION=PORTRAIT, SHOW_
GLOBALFILTER=OFF, $
PAGELAYOUT=1, NAME='Page 1', TEXT='Page 1', LEFTMARGIN=0.25,
RIGHTMARGIN=0.25, TOPMARGIN=0.25, BOTTOMMARGIN=0.25, $
COMPONENT=TableChart_1, COMPONENT-TYPE=REPORT, POSITION=(0.260416 0.260416),
DIMENSION=(3.885416 3.802083), ARREPORTSIZE=DIMENSION, CELL-POSITION=(1 1),
CELL-DIMENSION=(5 15), METADATA='VERSION:1.0', $
```

```
COMPONENT=TableChart_2, COMPONENT-TYPE=REPORT, POSITION=(4.229166 0.260416),
DIMENSION=(8.020833 4.21875), ARREPORTSIZE=DIMENSION, CELL-POSITION=(+2 +0),
CELL-RELATIVE-TO=TableChart_1, CELL-POSITION-POINT=TOP-LEFT, CELL-RELATIVE-
POINT=TOP-RIGHT, METADATA='VERSION:1.0', $
COMPONENT=TableChart_3, COMPONENT-TYPE=GRAPH, POSITION=(1.09375 4.364583),
DIMENSION=(4.96875 3.46875), ARREPORTSIZE=DIMENSION, CELL-POSITION=(+0 +1),
CELL-DIMENSION=(5 15), CELL-RELATIVE-TO=TableChart_2, CELL-POSITION-
POINT=TOP-LEFT, CELL-RELATIVE-POINT=BOTTOM-LEFT, METADATA='VERSION:1.0', $
PAGELAYOUT=2, NAME='Page 2', TEXT='Page 2', LEFTMARGIN=0.25,
RIGHTMARGIN=0.25, TOPMARGIN=0.25, BOTTOMMARGIN=0.25, $
COMPONENT=TableChart_4, COMPONENT-TYPE=REPORT, POSITION=(0.260416 0.260416),
DIMENSION=(5.0 3.5), ARREPORTSIZE=DIMENSION, CELL-POSITION=(1 1), CELL-
DIMENSION=(5 15), METADATA='VERSION:1.0', $
COMPONENT=TableChart_5, COMPONENT-TYPE=REPORT, POSITION=(0.260416 3.989583),
DIMENSION=(4.96875 3.46875), ARREPORTSIZE=DIMENSION, CELL-POSITION=(+0 +4),
CELL-RELATIVE-TO=TableChart_4, CELL-POSITION-POINT=TOP-LEFT, CELL-RELATIVE-
POINT=BOTTOM-LEFT, METADATA='VERSION:1.0', $
PAGELAYOUT=3, NAME='Page 3', TEXT='Page 3', LEFTMARGIN=0.25,
RIGHTMARGIN=0.25, TOPMARGIN=0.25, BOTTOMMARGIN=0.25, $
COMPONENT=TableChart_6, COMPONENT-TYPE=GRAPH, POSITION=(0.260416 0.260416),
DIMENSION=(5.0 3.5), ARREPORTSIZE=DIMENSION, CELL-POSITION=(1 1), CELL-
DIMENSION=(12 27), METADATA='VERSION:1.0', $
END

SET COMPONENT=TableChart_1
SET ARVERSION=2
-DEFAULTH &WF_SUMMARY='Summary';
-DEFAULTH &WF_TITLE='WebFOCUS Report';
TABLE FILE wfretail93/wf_retail_lite
SUM wf_retail_lite.WF_RETAIL_SALES.QUANTITY_SOLD
wf_retail_lite.WF_RETAIL_SALES.REVENUE_US
BY wf_retail_lite.WF_RETAIL_PRODUCT.PRODUCT_CATEGORY
BY wf_retail_lite.WF_RETAIL_TIME_SALES.TIME_YEAR
ON TABLE PCHOLD FORMAT PDF
ON TABLE NOTOTAL
ON TABLE SET VZERO OFF
ON TABLE SET STYLE *
INCLUDE=IBFS:/WFC/Global/Themes/Standard/Default/theme.sty,$
TYPE=REPORT, TITLETEXT='Report1', SUMMARY=&WF_SUMMARY.QUOTEDSTRING,
ARREPORTSIZE=DIMENSION, ARFILTER_TARGET='*', CHART-COLOR-MEASURES=ON,
ARGRAPHENGINE=JSCHART, $
ENDSTYLE
END

SET COMPONENT=TableChart_2
SET ARVERSION=2
-DEFAULTH &WF_SUMMARY='Summary';
```

```
-DEFAULTH &WF_TITLE='WebFOCUS Report';
TABLE FILE wfretail93/wf_retail_lite
SUM wf_retail_lite.WF_RETAIL_SALES.QUANTITY_SOLD
wf_retail_lite.WF_RETAIL_SALES.REVENUE_US
BY wf_retail_lite.WF_RETAIL_GEOGRAPHY_CUSTOMER.BUSINESS_REGION
BY wf_retail_lite.WF_RETAIL_TIME_SALES.TIME_YEAR
ON TABLE PCHOLD FORMAT PDF
ON TABLE NOTOTAL
ON TABLE SET VZERO OFF
ON TABLE SET STYLE *
INCLUDE=IBFS:/WFC/Global/Themes/Standard/Default/theme.sty,$
TYPE=REPORT, TITLETEXT='Report2', SUMMARY=&WF_SUMMARY.QUOTEDSTRING,
ARREPORTSIZE=DIMENSION, ARFILTER_TARGET='*', CHART-COLOR-MEASURES=ON,
ARGRAPHENGINE=JSCHART, $
ENDSTYLE
END

SET COMPONENT=TableChart_3
SET ARVERSION=2
-DEFAULTH &WF_TITLE='WebFOCUS Report';
GRAPH FILE wfretai93/wf_retail_lite
-* Created by Designer for Graph
SUM wf_retail_lite.WF_RETAIL_SALES.QUANTITY_SOLD
wf_retail_lite.WF_RETAIL_SALES.REVENUE_US
BY wf_retail_lite.WF_RETAIL_PRODUCT.PRODUCT_CATEGORY
ON GRAPH PCHOLD FORMAT JSCHART
ON GRAPH SET VZERO OFF
ON GRAPH SET HAXIS 4.96875
ON GRAPH SET VAXIS 3.46875
ON GRAPH SET GRAPHCVTUNITS ON
ON GRAPH SET UNITS INCHES
ON GRAPH SET LOOKGRAPH PIE
ON GRAPH SET AUTOFIT ON
ON GRAPH SET STYLE *
INCLUDE=IBFS:/WFC/Global/Themes/Standard/Default/theme.sty,$
TYPE=REPORT, TITLETEXT='Chart1', ARREPORTSIZE=DIMENSION, ARFILTER_TARGET='*',
ARGRAPHENGINE=JSCHART, $
TYPE=DATA, COLUMN=N1, BUCKET=color, $
TYPE=DATA, COLUMN=N2, BUCKET=measure, $
TYPE=DATA, COLUMN=N3, BUCKET=measure, $
*GRAPH_SCRIPT

*GRAPH_JS_FINAL
"extensions": {
    "com.ibi.galigeo_point": {
        "uniqueID": "uniqueId0x2827595240636638"
    },
```

```
    "com.ibi.galigeo_choropleth": {
        "uniqueID": "uniqueId0x5754462388070566"
    }
},
"pieProperties": {
    "holeSize":"65%"
}

*END
ENDSTYLE
END

SET COMPONENT=TableChart_4
SET ARVERSION=2
-DEFAULTH &WF_SUMMARY='Summary';
-DEFAULTH &WF_TITLE='WebFOCUS Report';
TABLE FILE wfretail93/wf_retail_lite
SUM wf_retail_lite.WF_RETAIL_SALES.QUANTITY_SOLD
wf_retail_lite.WF_RETAIL_SALES.REVENUE_US
wf_retail_lite.WF_RETAIL_SALES.GROSS_PROFIT_US
BY wf_retail_lite.WF_RETAIL_TIME_SALES.TIME_YEAR
ACROSS wf_retail_lite.WF_RETAIL_TIME_SALES.TIME_QTR
ON TABLE PCHOLD FORMAT PDF
ON TABLE NOTOTAL
ON TABLE SET VZERO OFF
ON TABLE SET STYLE *
INCLUDE=IBFS:/WFC/Global/Themes/Standard/Default/theme.sty,$
TYPE=REPORT, TITLETEXT='Report3', SUMMARY=&WF_SUMMARY.QUOTEDSTRING,
ARREPORTSIZE=DIMENSION, ARFILTER_TARGET='*', CHART-COLOR-MEASURES=ON,
ARGRAPHENGINE=JSCHART, $
ENDSTYLE
END

SET COMPONENT=TableChart_5
SET ARVERSION=2
-DEFAULTH &WF_SUMMARY='Summary';
-DEFAULTH &WF_TITLE='WebFOCUS Report';
TABLE FILE wfretail93/wf_retail_lite
SUM wf_retail_lite.WF_RETAIL_SALES.REVENUE_US
BY wf_retail_lite.WF_RETAIL_PRODUCT.BRAND
ACROSS wf_retail_lite.WF_RETAIL_PRODUCT.PRODUCT_CATEGORY
ON TABLE PCHOLD FORMAT PDF
ON TABLE NOTOTAL
ON TABLE SET VZERO OFF
ON TABLE SET STYLE *
INCLUDE=IBFS:/WFC/Global/Themes/Standard/Default/theme.sty,$
TYPE=REPORT, TITLETEXT='Report4', SUMMARY=&WF_SUMMARY.QUOTEDSTRING,
```

```
ARREPORTSIZE=DIMENSION, ARFILTER_TARGET='*', CHART-COLOR-MEASURES=ON,
ARGRAPHENGINE=JSCHART, $
ENDSTYLE
END

SET COMPONENT=TableChart_6
SET ARVERSION=2
-DEFAULTH &WF_TITLE='WebFOCUS Report';
GRAPH FILE wfretai93/wf_retail_lite
-* Created by Designer for Graph
SUM wf_retail_lite.WF_RETAIL_SALES.REVENUE_US
BY wf_retail_lite.WF_RETAIL_PRODUCT.PRODUCT_CATEGORY
BY wf_retail_lite.WF_RETAIL_TIME_SALES.TIME_DATE_QTR_COMPONENT
ON GRAPH PCHOLD FORMAT JSCHART
ON GRAPH SET VZERO OFF
ON GRAPH SET HAXIS 5.0
ON GRAPH SET VAXIS 3.5
ON GRAPH SET GRAPHCVTUNITS ON
ON GRAPH SET UNITS INCHES
ON GRAPH SET LOOKGRAPH LINE
ON GRAPH SET AUTOFIT ON
ON GRAPH SET STYLE *
INCLUDE=IBFS:/WFC/Global/Themes/Standard/Default/theme.sty,$
TYPE=REPORT, TITLETEXT='Chart2', ARREPORTSIZE=DIMENSION, ARFILTER_TARGET='*',
ARGRAPHENGINE=JSCHART, $
TYPE=DATA, COLUMN=N1, BUCKET=color, $
TYPE=DATA, COLUMN=N2, BUCKET=x-axis, AXIS-TYPE=time_auto, $
TYPE=DATA, COLUMN=N3, BUCKET=y-axis(1), $
*GRAPH_SCRIPT

*GRAPH_JS_FINAL
"extensions": {
    "com.ibi.galigeo_point": {
        "uniqueID": "uniqueId0x11339770590690268"
    },
    "com.ibi.galigeo_choropleth": {
        "uniqueID": "uniqueId0x7404939218587612"
    }
},
"blaProperties": {
    "seriesLayout": "absolute"
}

*END
ENDSTYLE
END
```

```
COMPOUND END
```

目次を含む Excel 複合レポート (BYTOC)

```
SET PAGE-NUM=OFF
COMPOUND LAYOUT PCHOLD FORMAT XLSX
SECTION=Example, LAYOUT=ON, MERGE=OFF,$
PAGELAYOUT=1,$
COMPONENT=R1, TYPE=REPORT,TEXT='report1', POSITION=(0.833 0.729), DIMENSION=
(6.250 1.771),$
COMPONENT=R2, TYPE=REPORT,TEXT='report2', POSITION=(0.833 2.917), DIMENSION=
(6.250 1.875),$
COMPONENT=R3, TYPE=REPORT,TEXT='report3', POSITION=(0.938 5.313), DIMENSION=
(6.250 1.354),$
COMPONENT=R4, TYPE=REPORT,TEXT='report4', POSITION=(0.938 7.083), DIMENSION=
(6.042 1.146),$
END
SET COMPONENT=R1
TABLE FILE GGSALES
HEADING CENTER
"Gotham Grinds Sales to Information Builders"
" "
"Report 1"
"Sales Summary by Region"
" "
SUM UNITS/D12C BUDUNITS/D12C DOLLARS/D12CM BUDDOLLARS/D12CM
BY REGION
ON TABLE HOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Region Summary,$
TYPE=REPORT, TOPMARGIN=1.5, BOTTOMMARGIN=1, PAGESIZE=LETTER,$
TYPE=TITLE, COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=HEADING, LINE=1, OBJECT=TEXT, COLOR=PURPLE, JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=HEADING, LINE=3, OBJECT=TEXT, COLOR=BLUE, JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=HEADING, LINE=4, OBJECT=TEXT, COLOR=PURPLE, JUSTIFY=CENTER, STYLE=BOLD,$
ENDSTYLE
END
```

```
SET COMPONENT=R2
TABLE FILE GGSALES
```

```
SUM UNITS/D12C DOLLARS/D12CM
BY REGION BY CATEGORY BY PRODUCT
HEADING CENTER
"Gotham Grinds Sales to Information Builders"
" "
"Report 2"
"Sales Detail By Region"
ON REGION SUBHEAD
"<REGION Region Sales"
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET BYDISPLAY ON
ON TABLE SET COMPOUND BYTOC
ON TABLE SET STYLE *
TYPE=REPORT, TOPMARGIN=1.5, BOTTOMMARGIN=1, PAGESIZE=LETTER,$
TYPE=REPORT, TITLETEXT='Region-',$
TYPE=TITLE, COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=HEADING, LINE=1, COLOR=PURPLE, JUSTIFY=CENTER,STYLE=BOLD,$
TYPE=HEADING, LINE=3, COLOR=BLUE, JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=HEADING, LINE=4, COLOR=PURPLE, JUSTIFY=CENTER, STYLE=BOLD,$
ENDSTYLE
END
```

```
SET COMPONENT=R3
TABLE FILE GGSALES
HEADING CENTER
"Gotham Grinds Sales to Information Builders"
" "
"Report 3"
"Sales Summary by Category"
" "
SUM UNITS/D12C BUDUNITS/D12C DOLLARS/D12CM BUDDOLLARS/D12CM
BY CATEGORY
ON TABLE HOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Sales Summary,$
TYPE=REPORT, TOPMARGIN=1.5, BOTTOMMARGIN=1, PAGESIZE=LETTER,$
TYPE=TITLE, COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=HEADING,LINE=1,COLOR=PURPLE, JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=HEADING,LINE=3,OBJECT=TEXT,COLOR=BLUE, JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=HEADING,LINE=4,OBJECT=TEXT,COLOR=PURPLE, JUSTIFY=CENTER, STYLE=BOLD,$
ENDSTYLE
END
```

```
SET COMPONENT=R4
TABLE FILE GGSALES
```

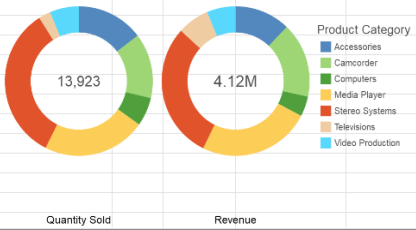
```
HEADING CENTER
"Gotham Grinds Sales to Information Builders"
" "
"Report 4"
"Sales Detail Report By Category"
" "
SUM UNITS/D12C BUDUNITS/D12C DOLLARS/D12CM BUDDOLLARS/D12CM
BY CATEGORY BY PRODUCT BY REGION
ON TABLE SET BYDISPLAY ON
ON TABLE HOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Sales Detail,$
TYPE=REPORT, TOPMARGIN=1.5, BOTTOMMARGIN=1, PAGESIZE=LETTER,$
TYPE=TITLE, COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=HEADING,LINE=1,OBJECT=TEXT,COLOR=PURPLE, JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=HEADING,LINE=3,OBJECT=TEXT,COLOR=BLUE, JUSTIFY=CENTER, STYLE=BOLD,$
TYPE=HEADING,LINE=4,OBJECT=TEXT,COLOR=PURPLE, JUSTIFY=CENTER, STYLE=BOLD,$
ENDSTYLE
END
COMPOUND END
```

出力結果は次のとおりです。

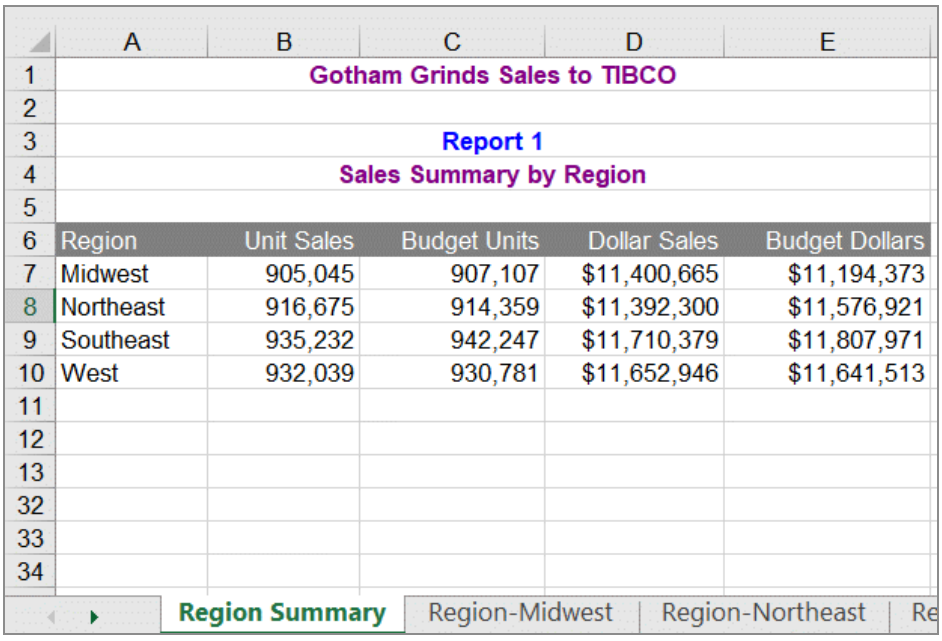
レポート 1 - コンポーネントを含む複合 Excel レポート

Product Category	Sale Year	Quantity Sold	Revenue
Accessories	2020	2,034	$499,551.40
Camcorder	2020	1,956	$667,254.48
Computers	2020	865	$179,761.46
Media Player	2020	3,140	$1,009,790.56
Stereo Systems	2020	4,679	$1,216,393.89
Televisions	2020	362	$286,160.68
Video Production	2020	887	$259,179.84

Customer Business Region	Sale Year	Quantity Sold	Revenue
EMEA	2020	5,947	$1,745,614.57
North America	2020	6,848	$2,032,343.92
Oceania	2020	52	$13,657.33
South America	2020	1,076	$326,476.49



レポート 2 - 地域別の概要レポート

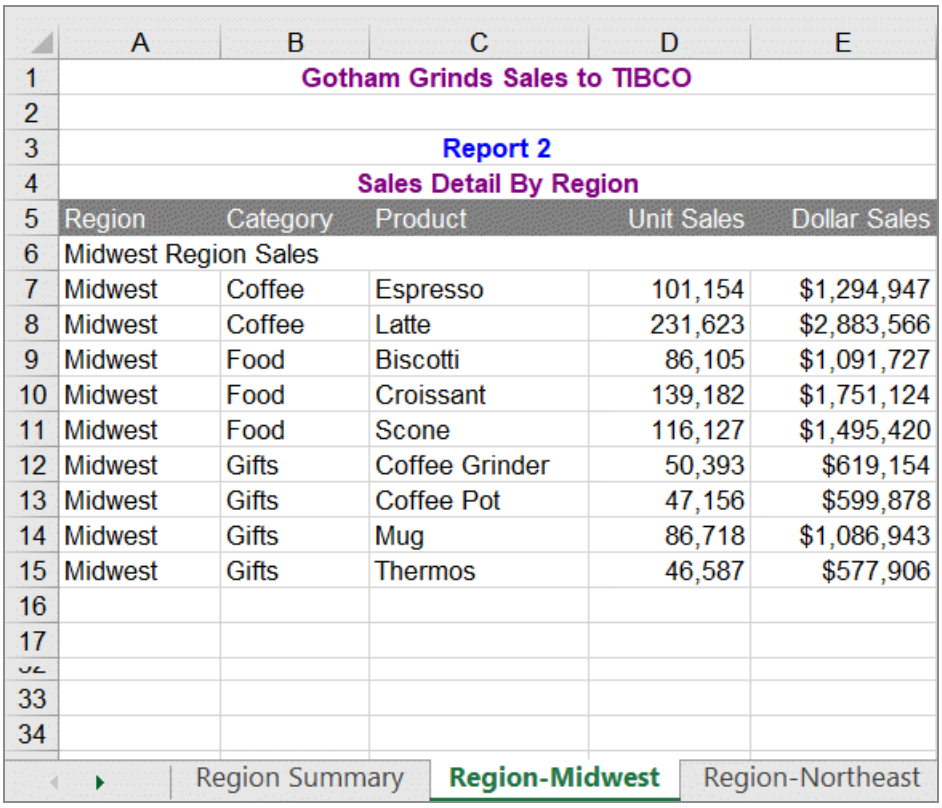


	A	B	C	D	E
1	Gotham Grinds Sales to TIBCO				
2					
3	Report 1				
4	Sales Summary by Region				
5					
6	Region	Unit Sales	Budget Units	Dollar Sales	Budget Dollars
7	Midwest	905,045	907,107	$11,400,665	$11,194,373
8	Northeast	916,675	914,359	$11,392,300	$11,576,921
9	Southeast	935,232	942,247	$11,710,379	$11,807,971
10	West	932,039	930,781	$11,652,946	$11,641,513
11					
12					
13					
32					
33					
34					

Region Summary | Region-Midwest | Region-Northeast | Re

レポート 3 - 地域別の BYTOC レポート

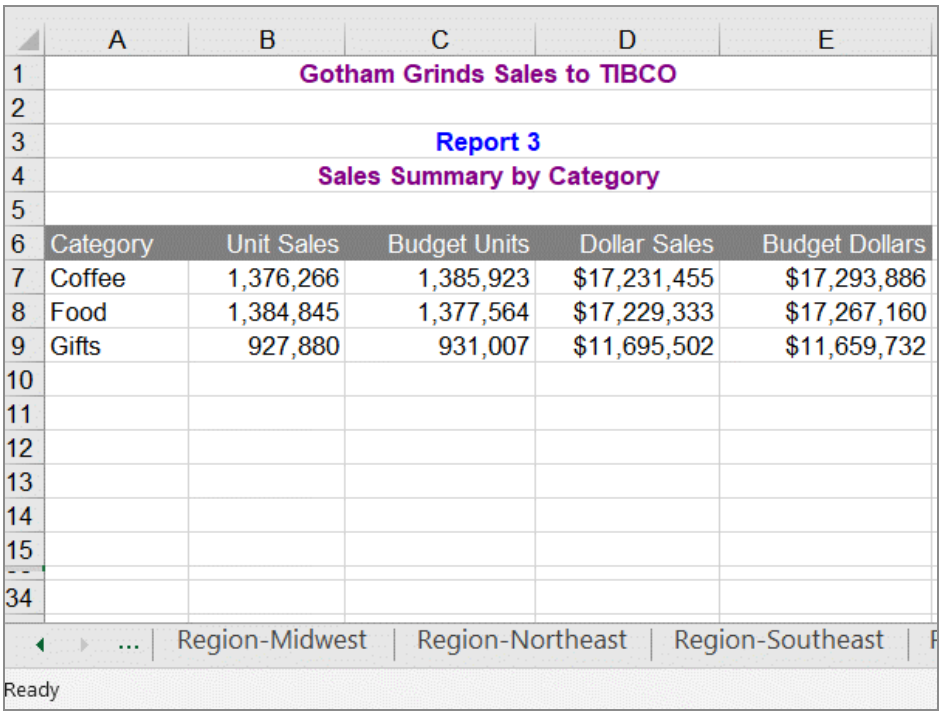
下図は、最初の地域のワークシートを示しています。



	A	B	C	D	E
1	Gotham Grinds Sales to TIBCO				
2					
3	Report 2				
4	Sales Detail By Region				
5	Region	Category	Product	Unit Sales	Dollar Sales
6	Midwest Region Sales				
7	Midwest	Coffee	Espresso	101,154	$1,294,947
8	Midwest	Coffee	Latte	231,623	$2,883,566
9	Midwest	Food	Biscotti	86,105	$1,091,727
10	Midwest	Food	Croissant	139,182	$1,751,124
11	Midwest	Food	Scone	116,127	$1,495,420
12	Midwest	Gifts	Coffee Grinder	50,393	$619,154
13	Midwest	Gifts	Coffee Pot	47,156	$599,878
14	Midwest	Gifts	Mug	86,718	$1,086,943
15	Midwest	Gifts	Thermos	46,587	$577,906
16					
17					
33					
34					

Region Summary | Region-Midwest | Region-Northeast

レポート 4 - カテゴリ別の売上概要レポート

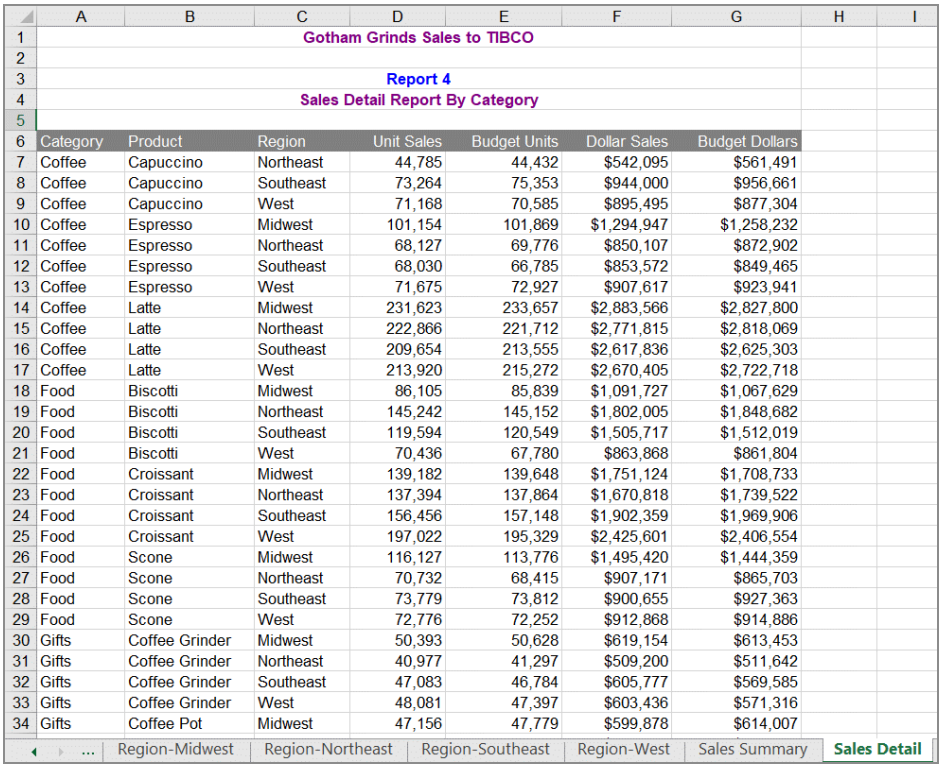


	A	B	C	D	E
1	Gotham Grinds Sales to TIBCO				
2					
3	Report 3				
4	Sales Summary by Category				
5					
6	Category	Unit Sales	Budget Units	Dollar Sales	Budget Dollars
7	Coffee	1,376,266	1,385,923	$17,231,455	$17,293,886
8	Food	1,384,845	1,377,564	$17,229,333	$17,267,160
9	Gifts	927,880	931,007	$11,695,502	$11,659,732
10					
11					
12					
13					
14					
15					
34					

... | Region-Midwest | Region-Northeast | Region-Southeast

Ready

レポート 5 - カテゴリ別の売上詳細レポート



	A	B	C	D	E	F	G	H	I
1	Gotham Grinds Sales to TIBCO								
2									
3	Report 4								
4	Sales Detail Report By Category								
5									
6	Category	Product	Region	Unit Sales	Budget Units	Dollar Sales	Budget Dollars		
7	Coffee	Capuccino	Northeast	44,785	44,432	$542,095	$561,491		
8	Coffee	Capuccino	Southeast	73,264	75,353	$944,000	$956,661		
9	Coffee	Capuccino	West	71,168	70,585	$895,495	$877,304		
10	Coffee	Espresso	Midwest	101,154	101,869	$1,294,947	$1,258,232		
11	Coffee	Espresso	Northeast	68,127	69,776	$850,107	$872,902		
12	Coffee	Espresso	Southeast	68,030	66,785	$853,572	$849,465		
13	Coffee	Espresso	West	71,675	72,927	$907,617	$923,941		
14	Coffee	Latte	Midwest	231,623	233,657	$2,883,566	$2,827,800		
15	Coffee	Latte	Northeast	222,866	221,712	$2,771,815	$2,818,069		
16	Coffee	Latte	Southeast	209,654	213,555	$2,617,836	$2,625,303		
17	Coffee	Latte	West	213,920	215,272	$2,670,405	$2,722,718		
18	Food	Biscotti	Midwest	86,105	85,839	$1,091,727	$1,067,629		
19	Food	Biscotti	Northeast	145,242	145,152	$1,802,005	$1,848,682		
20	Food	Biscotti	Southeast	119,594	120,549	$1,505,717	$1,512,019		
21	Food	Biscotti	West	70,436	67,780	$863,868	$861,804		
22	Food	Croissant	Midwest	139,182	139,648	$1,751,124	$1,708,733		
23	Food	Croissant	Northeast	137,394	137,864	$1,670,818	$1,739,522		
24	Food	Croissant	Southeast	156,456	157,148	$1,902,359	$1,969,906		
25	Food	Croissant	West	197,022	195,329	$2,425,601	$2,406,554		
26	Food	Scone	Midwest	116,127	113,776	$1,495,420	$1,444,359		
27	Food	Scone	Northeast	70,732	68,415	$907,171	$865,703		
28	Food	Scone	Southeast	73,779	73,812	$900,655	$927,363		
29	Food	Scone	West	72,776	72,252	$912,868	$914,886		
30	Gifts	Coffee Grinder	Midwest	50,393	50,628	$619,154	$613,453		
31	Gifts	Coffee Grinder	Northeast	40,977	41,297	$509,200	$511,642		
32	Gifts	Coffee Grinder	Southeast	47,083	46,784	$605,777	$569,585		
33	Gifts	Coffee Grinder	West	48,081	47,397	$603,436	$571,316		
34	Gifts	Coffee Pot	Midwest	47,156	47,779	$599,878	$614,007		

... | Region-Midwest | Region-Northeast | Region-Southeast | Region-West | Sales Summary | Sales Detail

Excel 複合ドキュメントへのコンポーネントの配置

バージョン 9.3.0 以降、XLSX では、PDF、AHTML、DHTML、その他のフォーマットで使用する構文に類似の複合ドキュメント構文を使用することができます。PAGELAYOUT を使用すると、これらは個別のワークシートとして処理されます。セル固有の配置オプションを使用することで、動的な出力フォーマット適用し、XLSX ドキュメントをセルを基準に配置することができます。XLSX 以外の出力では、測定単位 (インチやセンチメートルなど) を使用します。

次の機能がサポートされます。

- 統合されたドキュメント
- ページマスター
- Excel テーブル
- ReportCaster のバースト機能

例 - 複数ページレイアウトを含む複合レイアウト

```
  SECTION=Section_1, LAYOUT=ON, PAGESIZE=LETTER, ORIENTATION=PORTRAIT, SHOW_
GLOBALFILTER=OFF, $
  PAGELAYOUT=1, NAME='Page 1', TEXT='Page 1', LEFTMARGIN=0.25,
RIGHTMARGIN=0.25,
    TOPMARGIN=0.25, BOTTOMMARGIN=0.25, $
  COMPONENT='report1', COMPONENT-TYPE=REPORT,
    CELL-POSITION=(3 5), $
  COMPONENT='Chart',   COMPONENT-TYPE=GRAPH,
    CELL-RELATIVE-TO=report1, CELL-RELATIVE-POINT=BOTTOM-LEFT, CELL-POSITION=
(+3 +1),
    CELL-DIMENSION=(6 15), $
  COMPONENT='report2', COMPONENT-TYPE=REPORT,
    CELL-RELATIVE-TO=Chart,   CELL-RELATIVE-POINT=BOTTOM-LEFT, CELL-POSITION=
(-1 +1), $
  COMPONENT='report3', COMPONENT-TYPE=REPORT,
    CELL-RELATIVE-TO=report1, CELL-RELATIVE-POINT=TOP-RIGHT,   CELL-POSITION=
(+1 -2),$
```

説明

PAGELAYOUT

ページレイアウト内でコンポーネントを配置するページを指定します。

TEXT

各ページレイアウトのワークシート名を指定します。

COMPONENT-TYPE

コンポーネントのタイプに応じて、Report または Graph を指定します。イメージの場合、OBJECT=IMAGE を使用することもできます。

CELL-POSITION

各項目の左上隅を配置するセルを、xy 座標を使用して設定します。たとえば、(1 1) はセル A1 を表します。座標にプラス (+) またはマイナス (-) 符号を追加すると、相対的な配置が指定できます。

CELL-RELATIVE-TO

コンポーネントを配置する相対的な項目を指定します。CELL- で始まるプロパティは、PDF、DHTML、AHTML、その他の複合ドキュメント出力フォーマットなどのフォーマットに対する出力では無視されます。

CELL-RELATIVE-POINT

コンポーネントの左上隅を配置する CELL-RELATIVE-TO 項目の隅を指定します。

レガシー OPEN、CLOSE、NOBREAK キーワードおよび SET COMPOUND の使用に関するガイドライン

Excel 複合レポートを制御するには、OPEN、CLOSE、NOBREAK キーワードを使用します。これらのキーワードは、HOLD または PCHOLD コマンド、あるいは別の SET COMPOUND コマンドと併用することができます。

- OPEN キーワードは、一連のコンポーネントレポートの先頭レポートで使用し、複合レポートが開始されることを示します。
- CLOSE キーワードを使用して、複合レポートの最終レポートを指定します。
- NOBREAK キーワードを使用して、次のレポートが現在のレポートと同一のワークシートに配置されることを指定します。このキーワードによる指定がない場合、デフォルト設定では、次のレポートは新しいワークシートに配置されます。

- HOLD または PCHOLD 構文と併用する場合は、複合レイアウトキーワードの OPEN、CLOSE、NOBREAK を FORMAT XLSX の直後に記述する必要があります。たとえば、次のように指定します。
  - `ON TABLE PCHOLD FORMAT XLSX OPEN`
  - `ON TABLE HOLD AS MYHOLD FORMAT XLSX OPEN NOBREAK`
- PDF 複合レポートの場合と同様に、SET COMPOUND を使用して、これらの複合レポートキーワードを入れ替えて指定することができます。
  - `SET COMPOUND = OPEN`
  - `SET COMPOUND = 'OPEN NOBREAK'`
  - `SET COMPOUND = NOBREAK`
  - `SET COMPOUND = CLOSE`

XLSX による Excel 複合レポートの作成に関するガイドライン

- **ワークシートの名前指定**　ワークシートのタブ名は、デフォルトで Sheet1、Sheet2 のように設定されています。オプションとして、スタイルシートに TITLETEXT キーワードを使用して、異なる名前でワークシートのタブ名を指定することができます。以下はその例です。

  ```
  TYPE=REPORT, TITLETEXT='Summary Report',$
  ```

  Excel では、ワークシートのタイトルの最大長は 31 バイトに限定されています。特殊文字の「:」、「?」、「*」、「/」を使用することはできません。
- **ファイル名とフォーマット**　出力ファイル名 (AS 名。デフォルト値は HOLD) は、複合レポートの先頭レポートから取得することができます (OPEN キーワードが指定されたレポート)。先頭レポート以降の出力ファイル名は無視されます。

  複合レポートの先頭コンポーネントレポートで使用する HOLD FORMAT 構文は、フォーマットの種類に関係なく、先頭レポート以降のすべてのレポートに適用されます。
- **NOBREAK キーワードの動作**　NOBREAK キーワードを指定した場合、NOBREAK キーワードを含むレポートの最終行のすぐ後の行から次のレポートが表示されます。レ

ポート間で余分なスペースが必要な場合は、FOOTING または ON TABLE SUBFOOT を前のレポートに配置したり、HEADING または ON TABLE SUBHEAD を次のレポートに配置したりすることで対応します。デフォルト設定でブランク行が追加された場合は、それを削除することができないため、この方法で調整します。

XLSX による単純な複合レポートの作成

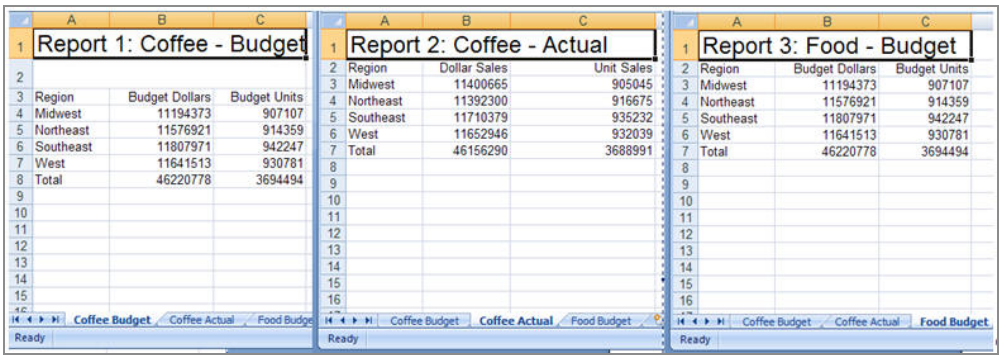
```
SET PAGE-NUM=OFF
TABLE FILE GGSALES
HEADING
"Report 1:Coffee - Budget"
" "
SUM BUDDOLLARS BUDUNITS COLUMN-TOTAL AS 'Total'
BY REGION
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Coffee Budget,$
TYPE=HEADING, SIZE=14,$
ENDSTYLE
ON TABLE PCHOLD AS EX1 FORMAT XLSX OPEN
END
```

```
TABLE FILE GGSALES
HEADING
"Report 2:Coffee - Actual "
SUM DOLLARS UNITS COLUMN-TOTAL AS 'Total'
BY REGION
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Coffee Actual,$
TYPE=HEADING, SIZE=14,$
ENDSTYLE
END
```

```
TABLE FILE GGSALES
HEADING
"Report 3:Food - Budget"
SUM BUDDOLLARS BUDUNITS COLUMN-TOTAL AS 'Total'
BY REGION
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Food Budget,$
TYPE=HEADING, SIZE=14,$
```

```
ENDSTYLE
ON TABLE PCHOLD FORMAT XLSX CLOSE
END
```

出力結果は次のとおりです。



Report 1: Coffee - Budget

Region	Budget Dollars	Budget Units
Midwest	11194373	907107
Northeast	11576921	914359
Southeast	11807971	942247
West	11641513	930781
Total	46220778	3694494

Report 2: Coffee - Actual

Region	Dollar Sales	Unit Sales
Midwest	11400665	905045
Northeast	11392300	916675
Southeast	11710379	935232
West	11652946	932039
Total	46156290	3688991

Report 3: Food - Budget

Region	Budget Dollars	Budget Units
Midwest	11194373	907107
Northeast	11576921	914359
Southeast	11807971	942247
West	11641513	930781
Total	46220778	3694494

NOBREAK キーワードによる複合レポートの作成

次の例では、NOBREAK が先頭レポートと 3 つ目のレポートに設定されているため、最初の 2 つのレポートが 1 つ目のワークシートに、最後の 2 つのレポートが 2 つ目のワークシートにそれぞれ表示されています。

```
TABLE FILE GGSALES
HEADING
"Report 1:Coffee - Budget"
SUM BUDDOLLARS BUDUNITS COLUMN-TOTAL AS 'Total'
BY REGION
IF CATEGORY EQ Coffee
ON TABLE PCHOLD FORMAT XLSX OPEN NOBREAK
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Coffee, FONT=ARIAL, SIZE=10, STYLE=NORMAL,$
TYPE=TITLE, STYLE=BOLD,$
TYPE=HEADING, SIZE=12, STYLE=BOLD, COLOR=BLUE,$
TYPE=GRANDTOTAL, STYLE=BOLD,$
END
```

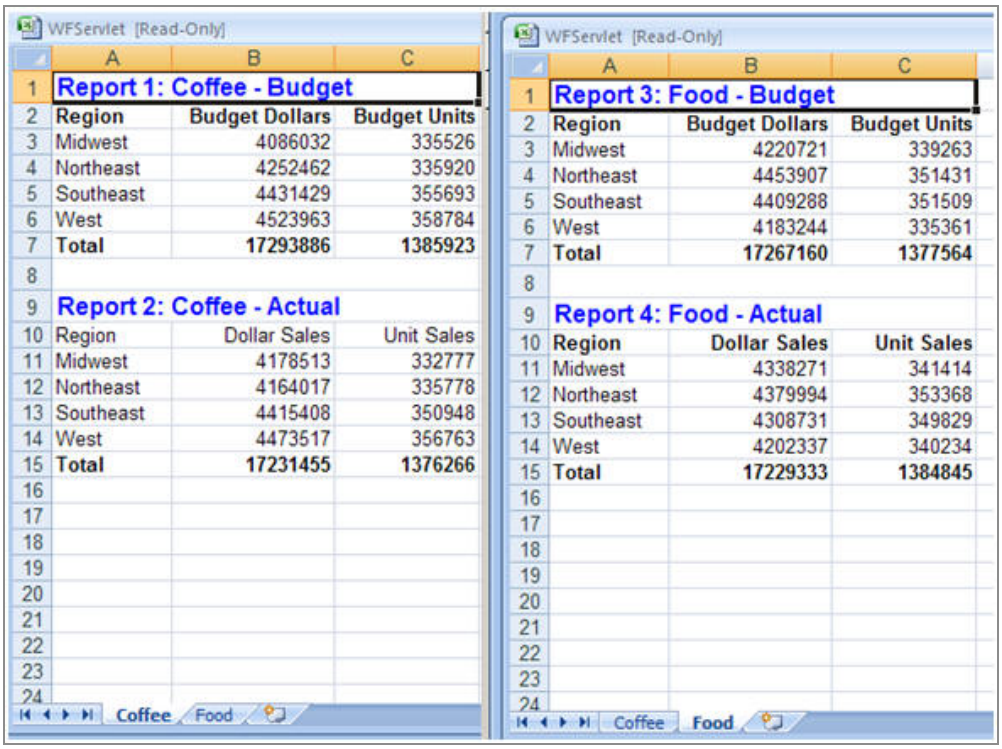
```
TABLE FILE GGSALES
HEADING
" "
```

```
"Report 2:Coffee - Actual "
SUM DOLLARS UNITS COLUMN-TOTAL AS 'Total'
BY REGION
IF CATEGORY EQ Coffee
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=REPORT, FONT=ARIAL, SIZE=10, STYLE=NORMAL,$
TYPE=GRANDTOTAL, STYLE=BOLD,$
TYPE=HEADING, SIZE=12, STYLE=BOLD, COLOR=BLUE,$
END
```

```
TABLE FILE GGSALES
HEADING
"Report 3:Food - Budget"
SUM BUDDOLLARS BUDUNITS COLUMN-TOTAL AS 'Total'
BY REGION
IF CATEGORY EQ Food
ON TABLE PCHOLD FORMAT XLSX NOBREAK
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT=Food, FONT=ARIAL, SIZE=10, STYLE=NORMAL,$
TYPE=HEADING, STYLE=BOLD, SIZE=12, COLOR=BLUE,$
TYPE=TITLE, STYLE=BOLD,$
TYPE=GRANDTOTAL, STYLE=BOLD,$
END
```

```
TABLE FILE GGSALES
HEADING
" "
"Report 4:Food - Actual"
SUM DOLLARS UNITS COLUMN-TOTAL AS 'Total'
BY REGION
IF CATEGORY EQ Food
ON TABLE PCHOLD FORMAT XLSX CLOSE
ON TABLE SET STYLE *
TYPE=REPORT, FONT=ARIAL, SIZE=10,  $
TYPE=TITLE, STYLE=BOLD,$
TYPE=HEADING, SIZE=12, STYLE=BOLD, COLOR=BLUE,$
TYPE=GRANDTOTAL, STYLE=BOLD,$
END
```

レポート出力は、2 つのタブに表示されます。



複合レポートでの XLSX FORMULA の使用

新しい複合レイアウト構文では、次のいずれかの方法で、XLSX FORMULA を使用した複合ブックの実装を有効にすることができます。どちらの方法でも、ブック内のすべてのレポートコンポーネントが FORMULA モードで生成されます。

1. 次のように、複合レイアウト構文ヘッダに FORMAT XLSX FORMULA を追加します。

```
COMPOUND LAYOUT PCHOLD FORMAT XLSX FORMULA
UNITS=IN, $
SECTION=section1, LAYOUT=ON, METADATA='prop_with_names, Margins_
Left=0.5,
Margins_Top=0.5, Margins_Right=0.5, Margins_Bottom=0.5,
thumbnailscale=4,
MERGE=OFF, ORIENTATION=PORTRAIT, PAGESIZE=Letter, SHOW_GLOBALFILTER=OFF,
$
PAGELAYOUT=1, NAME='Page layout 1', text='Page layout 1', TOC-LEVEL=1,
BOTTOMMARGIN=0.5, TOPMARGIN=0.5, METADATA='BOTTOMMARGIN=0.5,
TOPMARGIN=0.5,LEFTMARGIN=0,RIGHTMARGIN=0,', $
COMPONENT='report1', TEXT='report1', TOC-LEVEL=2, POSITION=(0.567
```

```
0.667),
DIMENSION=(6.883 2.314), BYTOC=0, ARREPORTSIZE=DIMENSION,
METADATA='left:0.567in; top:0.667in; width:6.883in; height:2.314in;
position: absolute; z-index:1;', $
COMPONENT='report2', TEXT='report2', TOC-LEVEL=2, POSITION=(0.567
3.250),
DIMENSION=(7.000 2.833), BYTOC=0, ARREPORTSIZE=DIMENSION,
METADATA='left:0.567in; top:3.25in; width:7in; height:2.833in;
position: absolute; z-index:2;', $
END
```

2. 次のように、先頭コンポーネントの出力設定として XLSX FORMULA を定義します。

```
SET COMPONENT='report1'
-*component_type report
-*File:IBFS:/localhost/EDA/9999/APPPATH/xlsx2015/Report1.fex
-*Created by WebFOCUS AppStudio
DEFINE FILE GGSALES
D_UOVERBUD/D12C=GGSALES.SALES01.UNITS - GGSALES.SALES01.BUDUNITS;
END
TABLE FILE GGSALES
SUM
     GGSALES.SALES01.UNITS
     GGSALES.SALES01.BUDUNITS
     GGSALES.SALES01.DOLLARS
     GGSALES.SALES01.BUDDOLLARS
     GGSALES.SALES01.D_UOVERBUD
     COMPUTE C_DOVERBUD/D12.2CM = GGSALES.SALES01.DOLLARS -
       GGSALES.SALES01.BUDDOLLARS;
BY GGSALES.SALES01.REGION
BY GGSALES.SALES01.CATEGORY
HEADING
"XLSX FORMULA - the difference betweem DEFINE & COMPUTES"
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE SET ASNAMES ON
ON TABLE COLUMN-TOTAL AS 'TOTAL'
ON TABLE PCHOLD FORMAT XLSX FORMULA
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
     INCLUDE = endeflt,
$
ENDSTYLE
```

注意

- 上記のいずれの方法も機能しますが、ツールの観点からすると、1 つ目の方法 (複合レ

イアウト構文ヘッダに FORMAT XLSX FORMULA を追加) に利点があります。それは、ページマスターを使用するページレイアウトなど、さまざまなページレイアウトのシナリオでシステムテーブルが生成されると、ユーザが先頭テーブルを制御できない場合があるためです。1 つ目の方法では、XLSX FORMULA 指定が常に先頭テーブルに移動されます。

- 複合ブックのフォーマットを変更するために、リクエストの複合レイアウト構文ヘッダで XLSX FORMULA の代わりに XLSX を使用した場合でも、FORMULA は自動的にオフになりません。先頭コンポーネントで FORMULA が設定されていると、その設定が常に複合レイアウト構文ヘッダ設定より優先されます。そのため、ドキュメントレベルで FORMULA をオフにしたにも関わらず、実際には先頭コンポーネントレベルで FORMULA のオンが保持されるため、混乱を招く可能性があります。

XLSX での WebFOCUS ピボットのサポート

WebFOCUS XLSX フォーマットを使用する際に、ピボットテーブルが事前に定義されたテンプレートに基づいてブックを生成することができます。これらのピボットテーブルは、レポートからフィードされたデータに基づいて作成することも、別のワークシート上の WebFOCUS データとは無関係に配置することもできます。

WebFOCUS レポートからピボットテーブルおよびピボットグラフへのデータフィード

ここでは、次のサンプルプロシジャを使用して、「wf2pivot.xltx」という既存の Excel テンプレートのピボットテーブルおよびピボットグラフに WebFOCUS レポートのデータをフィードする方法について説明します。

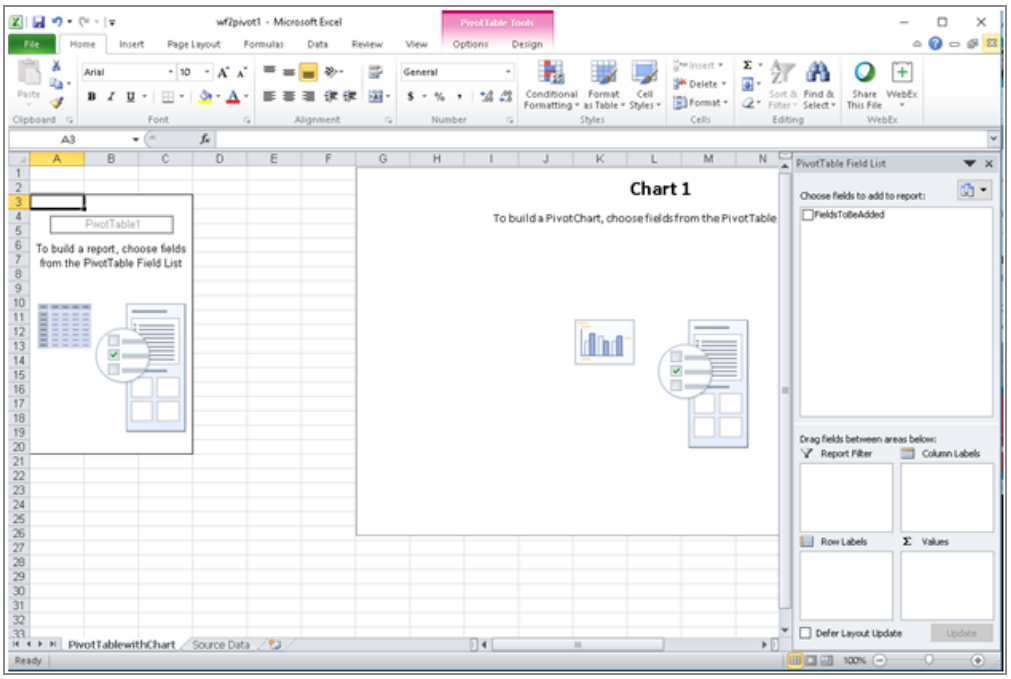
注意：テンプレート名にブランクが含まれている場合は、そのテンプレート名を一重引用符 (') で囲む必要があります。

```
TABLE FILE GGSALES
PRINT
UNITS/D12C DOLLARS/D12CM
BUDUNITS/D12C BUDDOLLARS/D12CM
BY LOWEST REGION
```

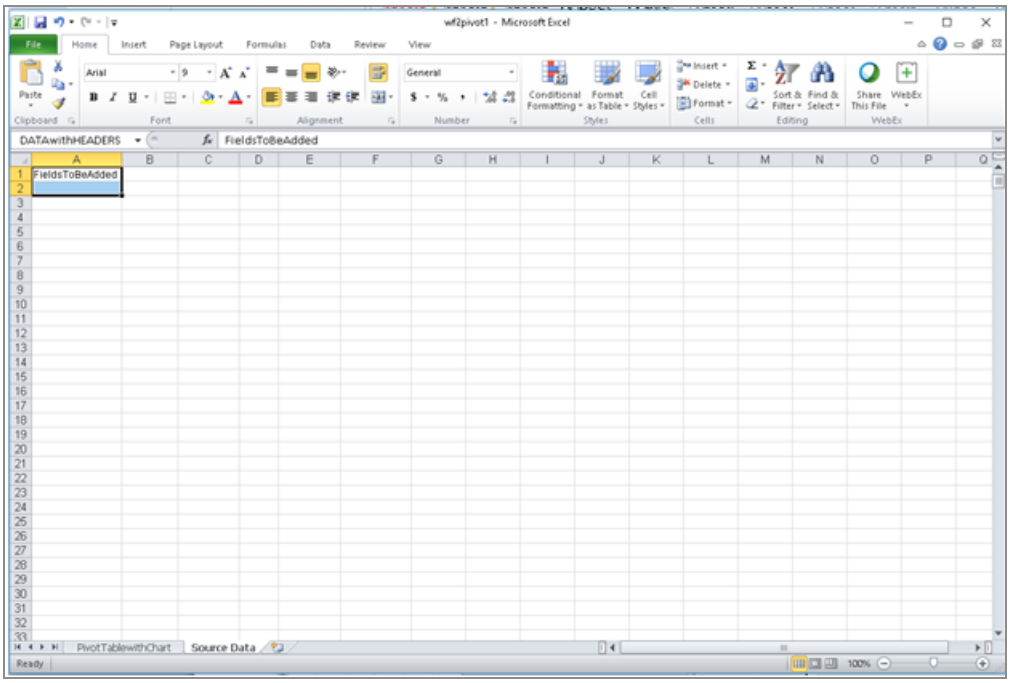
```
BY LOWEST ST
BY HIGHEST CATEGORY
BY LOWEST PRODUCT
ON TABLE SET BYDISPLAY ON
ON TABLE PCHOLD AS PIVOTWITHCHART FORMAT XLSX TEMPLATE wf2pivot.xltx
SHEETNUMBER 2
ON TABLE SET STYLE *
TYPE=DATA,IN-RANGES='DATAwithHEADERS',$
TYPE=TITLE,IN-RANGES='DATAwithHEADERS',$
ENDSTYLE
END
```

wf2pivot.xltx テンプレートファイルは、Reporting Server パス上に存在する必要があります。以下の 2 つの図は、上記のサンプルプロシジャを実行する前の 1 つ目のワークシートと 2 つ目のワークシートの初期状態を示しています。

1 つ目のワークシート (PivotTablewithChart) には、ブランクのピボットテーブルとピボットグラフが含まれています。また、このワークシートには、ブランクの PivotTable フィールドリストも含まれています。下図は、1 つ目のワークシートを示しています。



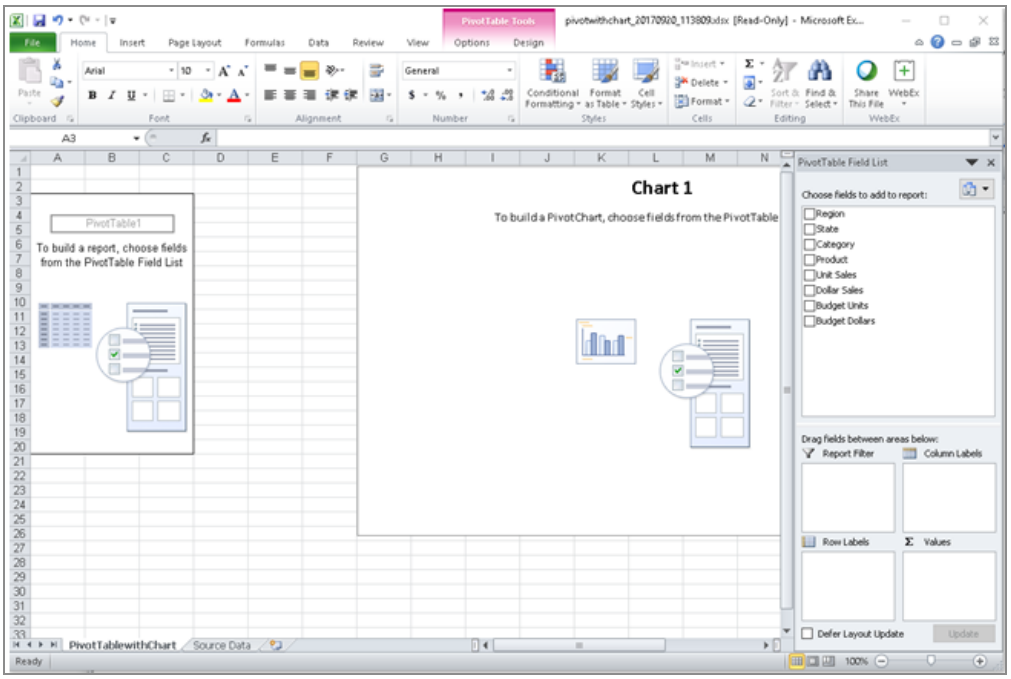
2 つ目のワークシート (Source Data) には、「FieldsToBeAdded」という名前の列が 1 つ含まれています。初期状態では、この列にデータはありません。下図は、2 つ目のワークシートを示しています。



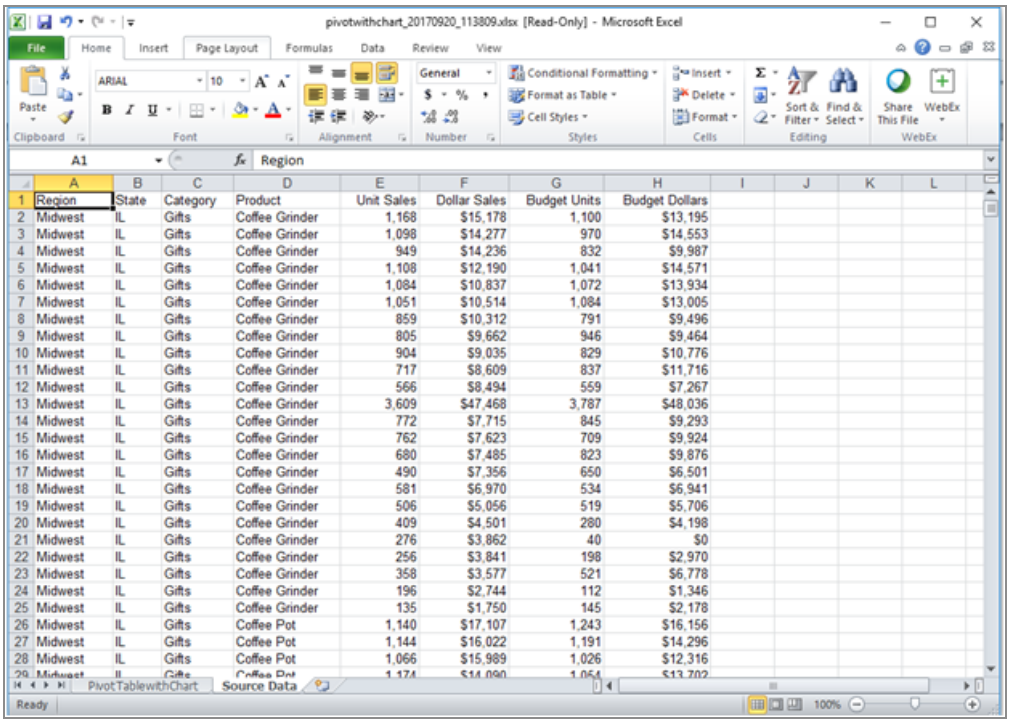
サンプルプロシジャを実行すると、pivotwithchart.xlsx ブックが生成され、WebFOCUS レポートデータが 2 つ目のワークシートに格納されます。

下図は、サンプルプロシジャを実行した後の pivotwithchart.xlsx ブックの 1 つ目のワークシートと 2 つ目のワークシートを示しています。

1 つ目のワークシート



2 つ目のワークシート

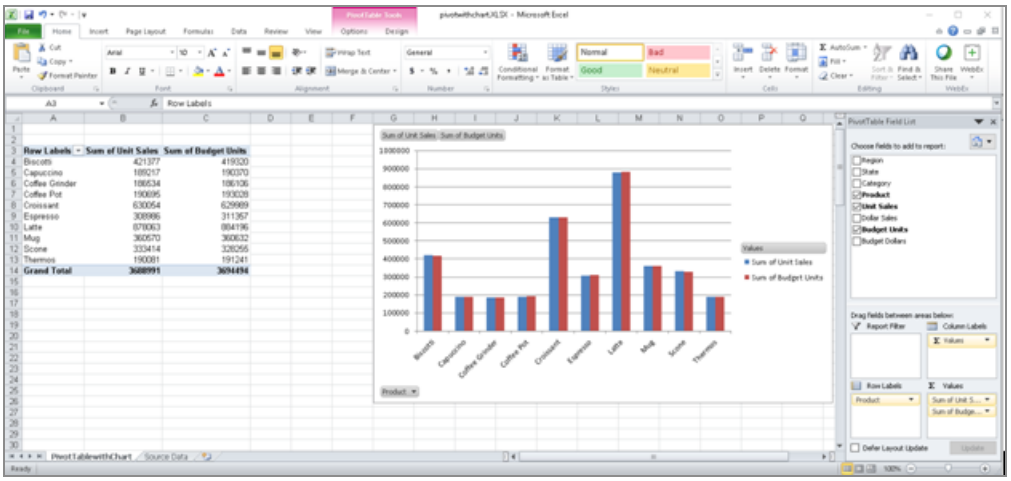


Region	State	Category	Product	Unit Sales	Dollar Sales	Budget Units	Budget Dollars
Midwest	IL	Gifts	Coffee Grinder	1,168	$15,178	1,100	$13,195
Midwest	IL	Gifts	Coffee Grinder	1,098	$14,277	970	$14,553
Midwest	IL	Gifts	Coffee Grinder	949	$14,236	832	$9,987
Midwest	IL	Gifts	Coffee Grinder	1,108	$12,190	1,041	$14,571
Midwest	IL	Gifts	Coffee Grinder	1,084	$10,837	1,072	$13,934
Midwest	IL	Gifts	Coffee Grinder	1,051	$10,514	1,084	$13,005
Midwest	IL	Gifts	Coffee Grinder	859	$10,312	791	$9,496
Midwest	IL	Gifts	Coffee Grinder	805	$9,662	946	$9,464
Midwest	IL	Gifts	Coffee Grinder	904	$9,035	829	$10,776
Midwest	IL	Gifts	Coffee Grinder	717	$8,609	837	$11,716
Midwest	IL	Gifts	Coffee Grinder	566	$8,494	559	$7,267
Midwest	IL	Gifts	Coffee Grinder	3,609	$47,468	3,787	$48,036
Midwest	IL	Gifts	Coffee Grinder	772	$7,715	845	$9,293
Midwest	IL	Gifts	Coffee Grinder	762	$7,623	709	$9,924
Midwest	IL	Gifts	Coffee Grinder	680	$7,485	823	$9,876
Midwest	IL	Gifts	Coffee Grinder	490	$7,356	650	$6,501
Midwest	IL	Gifts	Coffee Grinder	581	$6,970	534	$6,941
Midwest	IL	Gifts	Coffee Grinder	506	$5,056	519	$5,706
Midwest	IL	Gifts	Coffee Grinder	409	$4,501	280	$4,198
Midwest	IL	Gifts	Coffee Grinder	276	$3,862	40	$0
Midwest	IL	Gifts	Coffee Grinder	256	$3,841	198	$2,970
Midwest	IL	Gifts	Coffee Grinder	358	$3,577	521	$6,778
Midwest	IL	Gifts	Coffee Grinder	196	$2,744	112	$1,346
Midwest	IL	Gifts	Coffee Grinder	135	$1,750	145	$2,178
Midwest	IL	Gifts	Coffee Pot	1,140	$17,107	1,243	$16,156
Midwest	IL	Gifts	Coffee Pot	1,144	$16,022	1,191	$14,296
Midwest	IL	Gifts	Coffee Pot	1,066	$15,989	1,026	$12,316

PivotTablewithChart ワークシートでは、PivotTable フィールドリストに WebFOCUS レポートのフィールド群が挿入されています。レポートプロシジャで指定されたフィールドごとにチェックボックスが 1 つ表示されます。デフォルト設定では、これらのチェックボックスのいずれも選択されていません。

これらのチェックボックスの挿入に使用されたデータは、サンプルプロシジャの実行時に WebFOCUS レポートのデータが保存された [Source Data] ワークシートから取得されています。

ピボットテーブルおよびピボットグラフの作成を開始するには、PivotTable フィールドリストで、使用する 1 つまたは複数のフィールドのチェックボックスを選択します。たとえば、[Product]、[Unit Sales]、[Budget Units] フィールドのチェックボックスを選択すると、[Source Data] ワークシートのデータが、PivotTablewithChart ワークシートのピボットテーブルおよびピボットグラフに自動的にフィードされます。下図は、生成されたピボットテーブルおよびピボットグラフを示しています。



wf2pivot.xltx テンプレートは、デフォルト設定で、インストールした WebFOCUS Reporting Server のレガシーサンプルの一部として提供されます。Reporting Server レガシーサンプルをダウンロードする方法についての詳細は、『ibi™ WebFOCS® サーバ管理者ガイド』を参照してください。

この WebFOCUS XLSX テンプレートをそのまま使用することも、ユーザのビジネス要件に合わせてカスタマイズすることもできます。

注意：wf2pivot.xltx テンプレートには、同一のワークシートにピボットテーブルとピボットグラフが含まれていますが、マクロを有効にしたテンプレートを使用して、ピボットテーブルとピボットグラフをそれぞれ別のワークシート (共通のデータソースにリンクされる) に生成することもできます。

FORMAT XLSX の制限事項

EXL2K フォーマットで現在サポートされている機能のうち、XLSX フォーマットでサポートされない機能は次のとおりです。

- セルの保護

PowerPoint PPT 表示フォーマットの使用

出力フォーマットとして PowerPoint (PPT) を指定すると、レポートが単一スライドに配置された PowerPoint ドキュメントが作成されます。

PowerPoint プレゼンテーションには、複数のグラフやイメージを追加することができます。PowerPoint 出力フォーマットでは、スライドの任意の位置にさまざまなグラフを配置して、視覚的効果のあるレイアウトを作成することができます。

レポート出力は、PowerPoint テンプレートの特定のスライドに配置することもできます。

PowerPoint PPT テンプレートの使用

WebFOCUS レポートは、既存の PowerPoint プレゼンテーション内に配置することができます。これにより、既存のプレゼンテーションに定義済みのスライドマスター、スタイル設定、他のビジネスコンテンツを挿入することができます。PowerPoint PPT テンプレートは、.mht 拡張子でサーバに格納され、ReportCaster を使用して自動配信することができます。

PowerPoint PPT レポート出力の作成

```
ON TABLE {PCHOLD|HOLD|SAVE} [AS name] FORMAT PPT
      [TEMPLATE 'template' SLIDENUMBER n]
```

説明

name

PowerPoint 出力ファイルの名前です。

template

PowerPoint テンプレートファイルの名前です。テンプレートファイルには、少なくとも 1 つの空のスライドが含まれている必要があります。また、テンプレートファイルは、WebFOCUS Reporting Server アプリケーションディレクトリに Web アーカイブ (拡張子 .MHT) として保存しておく必要があります。

ヒント：Reporting Server は、Excel と PowerPoint のテンプレートファイル (.MHT) を区別できないため、テンプレートファイル名には特定の規則を適用することをお勧めします。これにより、Excel と PowerPoint のテンプレートファイルを整理し、レポート作成時にそれぞれのファイルを区別することができます。たとえば、Excel テンプレートには「Excel_template.MHT」、PowerPoint テンプレートには「PPT_template.MHT」のような名前を付けます。

n

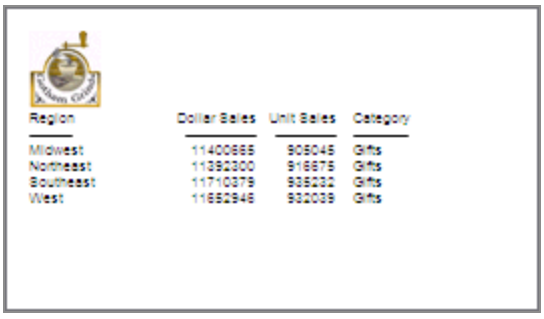
レポート出力を配置するスライドの番号です。テンプレートに含まれているスライドが 1 つの場合、この番号はオプションです。

PowerPoint PPT テンプレートの使用

次のリクエストは、GGSALES データソースに対して実行され、アプリケーションディレクトリに格納されている「mytemplate.mht」という PowerPoint テンプレートに WebFOCUS レポートを挿入します。

```
TABLE FILE GGSALES
HEADING
" "
" "
" "
" "
" "
SUM DOLLARS UNITS CATEGORY
BY REGION
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PPT TEMPLATE 'mytemplate' SLIDENUMBER 1
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
TYPE=REPORT, FONT=ARIAL, SIZE=10,$
TYPE=HEADING, image=gglogo.gif, POSITION=(0.000000 0.000000),$
ENDSTYLE
END
```

出力結果は次のとおりです。



Region	Dollar Sales	Unit Sales	Category
Midwest	11400666	906046	Gifts
Northeast	11382300	916875	Gifts
Southeast	11710379	935232	Gifts
West	11652946	932039	Gifts

PPTX フォーマットでのレポート出力の保存

WebFOCUS がサポートする任意のデータソースからデータを取得し、PPTX フォーマット (PowerPoint 2007、PowerPoint 2010、PowerPoint 2013) のプレゼンテーションを作成することができます。ここでは、重要な PPTX の機能について説明しますが、これらの説明は、特に指定のない限り、PowerPoint 2007、PowerPoint 2010、PowerPoint 2013 に適用されます。

WebFOCUS の PPTX フォーマットは、次の Microsoft Office ソフトウェア製品をサポートします。

- Microsoft Office 2013、2010、2007
- Microsoft Office 2000、2003 (Microsoft Office 互換性パックを使用)

  注意：Microsoft の「Support has ended for Office 2003」ステートメントによると、Office 2003 のサポートおよびアップデートは終了しています。Microsoft Office 2003 アプリケーションは今後も使用できますが、Microsoft では、サポートおよびアップデートを継続するため、新しいバージョンの Office へのアップグレードが推奨されています。Office 2003 の使用についての詳細は「end of support page for Windows XP SP3 and Office 2003」を参照してください。
- Open Office サポート (FORMAT PPTX)　PPTX フォーマットで生成される主要な PowerPoint 機能が Open Office でサポートされます。Open Office についての詳細は、「http://www.openoffice.org/」を参照してください。
- MAC Office 2008 および 2011。FORMAT PPTX がサポートされます。

WebFOCUS では、Microsoft PPTX の標準に基づいて PPTX プレゼンテーションが生成されます。これらのプレゼンテーションは、すべてのブラウザおよび Microsoft PowerPoint のモバイルアプリケーションからアクセスできます。

- Microsoft Office 365　Microsoft Office 365 は、PowerPoint 2013 のローカルインストール機能を提供します。Office 2010 に対応し、Office 2007 の一部機能にも対応しています。Microsoft Office 365 では、Office Online を使用してほとんどのデバイスからアクセス可能なクラウドに Microsoft PowerPoint のファイルをアップロードすることもできます。Microsoft Office 365 についての詳細は、「Office 365 for business FAQ」を参照してください。

  Microsoft Office 365 を使用して、WebFOCUS PPTX レポートにアクセスすることができます。最初に PCHOLD コマンドを使用して画面に PPTX レポートを表示し、次にこのレポートを OneDrive for Business に保存します。ファイルがクラウドに保存された

後は、Office Online を使用してこのファイルにアクセスできます。

PowerPoint Online および Microsoft Office 2013 で使用可能な機能の違いについての詳細は、「Office Online Service Description」を参照してください。

Office Online および OneDrive for Business の使用についての詳細は、「Using Office Online in OneDrive」を参照してください。

プレゼンテーションファイル (.pptx) の作成

Microsoft は、PowerPoint 2007 で、PowerPoint プレゼンテーションファイルのフォーマットおよび構造を変更しました。新しい .pptx ファイルは、複数の xml ファイルを含むバイナリ圧縮ファイルです。WebFOCUS でこの新しいファイルフォーマットを生成する処理は、レポート出力を含む xml ファイルを生成する処理と、xml ドキュメントをバイナリ .pptx フォーマットに圧縮する処理の 2 つで構成されます。xml 生成処理は、Reporting Server が実行します。圧縮処理は、クライアント (WebFOCUS Servlet) またはサーバ (JSCOM3) のいずれかが実行することができます。

- **WebFOCUS Servlet**　Application Server 内の WebFOCUS Client が圧縮処理を実行します。この処理は、ローカルクライアントで実行することも、リモートアクセスするクライアントで実行することもできます。Servlet を使用するオプションは、各 WebFOCUS Client で定義されているデフォルトの処理方法です。デフォルト設定では、クライアントの指定先がそのクライアント自体になっています。
- **JSCOM3**　Reporting Server の Java 層が圧縮処理を実行します。このオプションは、セキュリティの設定された Web サーバまたは Application Server で WebFOCUS Servlet が構成されている場合に使用する必要があります。これは、JSCOM3 がリモート WebFOCUS Client への URL アクセスを必要としないためです。

.pptx ファイルの圧縮方法の選択

圧縮を実行する方法と場所を指定するには、EXCELSERVURL を URL (WebFOCUS Servlet の場合) に設定するか、ブランク (JSCOM3 の場合) に設定します。この設定は、名前の先頭に EXCEL が指定されていますが、PowerPoint でも同じ結果が得られます。この値を特定のプロシジャに対して設定することも、環境全体に対して設定することもできます。

- **プロシジャの場合**　プロシジャで SET EXCELSERVURL コマンドを発行します。

- **環境全体の場合**　WebFOCUS 管理コンソールの [構成] タブで、[Client 設定]を開き、[Excel Server URL] (IBIF_excelservurl) を選択することで、IBIF_excelservurl 変数を編集できます。

WebFOCUS 管理コンソールへのアクセスおよび IBIF_excelservurl 変数の設定についての詳細は、『ibi™ WebFOCUS® セキュリティ管理ガイド』を参照してください。

EXCELSERVURL に割り当てた値に基づいて、WebFOCUS Servlet または JSCOM3 のどちらが圧縮処理を実行するかが決定されます。

- **Servlet の指定**　WebFOCUS Servlet を使用するよう指定するには、EXCELSERVURL パラメータまたは IBIF_excelservurl 変数を、WebFOCUS バージョン 8.0 SP 09 以降の Client 構成の URL に設定します。

  プロシジャで指定する場合

  ```
  SET EXCELSERVURL = http://servername:8080/ibi_apps
  ```

  WebFOCUS 管理コンソールで指定する場合

  ```
  IBIF_excelservurl = http://servername:8080/ibi_apps
  ```

- **JSCOM3 の指定**　現在の Reporting Server で JSCOM3 を使用するよう指定するには、EXCELSERVURL をブランクに設定します。

  プロシジャで指定する場合

  ```
  SET EXCELSERVURL = ' '
  ```

  WebFOCUS 管理コンソールで指定する場合

  ```
  IBIF_excelservurl = ' '
  ```

  デフォルト設定で、各 WebFOCUS Client には、その Client 自体を指定する次の URL が定義されています。

  ```
  &URL_PROTOCOL://&servername:&server_port&IBIF_webapp
  ```

PPTX プレゼンテーションの生成

レポートを PPTX プレゼンテーションに保存するか、ブラウザまたは PowerPoint アプリケーションのいずれかで表示するかを指定することができます。

```
ON TABLE {PCHOLD|HOLD} AS name FORMAT PPTX
```

説明

PCHOLD

デスクトップ設定に基づいて、生成されたプレゼンテーションをブラウザまたは PowerPoint アプリケーションのいずれかに表示します。

HOLD

拡張子に .pptx た使用されたプレゼンテーションを、指定した場所に保存します。

name

生成されるプレゼンテーションの名前を指定します。

注意：生成されるプレゼンテーションにファイル名を割り当てるには、WebFOCUS Client 出力先変更設定で、.pptx ファイル拡張子の [保存レポート] オプションを [はい] に設定します。PowerPoint アプリケーションで開いた場合、生成されたプレゼンテーションには、指定した AS 名が保持されます。出力先変更設定についての詳細は、『ibi™ WebFOCUS® セキュリティ管理ガイド』を参照してください。

PPTX レポート出力を開く

PPTX プレゼンテーションを開くには、Microsoft Office365 のアカウントを使用するか、デスクトップに Microsoft PowerPoint 2013、2010、2007 のいずれかをインストールしておく必要があります。

FORMAT PPTX でレポートを実行すると、ユーザは、PPTX ファイルを開くか保存するかの選択が要求されます。拡張子 .pptx の前に表示されるファイル名は、内部で生成されます。

WebFOCUS プロシジャで、出力の表示に必要な数のスライドを含むプレゼンテーションが生成されます。レポートには、見出し、中間合計、タイトルなど定義済みの要素、および条件付きスタイル、ドリルダウンなどのスタイルシート構文が含まれます。

Microsoft PowerPoint 2000/2003 で PPTX レポート出力を開く

PowerPoint 2000 および PowerPoint 2003 は、Microsoft Office 互換機能パックを使用することで PowerPoint PPTX プレゼンテーションを読み取れるよう更新することができます。この互換機能パックは、Microsoft ダウンロードサイト (http://www.microsoft.com/downloads/en/default.aspx) からダウンロードすることができます。拡張子が .pptx (PPTX プレゼンテーション) のファイルを開く際に、Microsoft Office 互換機能パックが、このファイルを PowerPoint 2000/2003 で読み取り、開くために必要な変換を実行します。

Microsoft Office 互換機能パックを使用すること以外に、WebFOCUS Client 出力先変更設定で [保存レポート] オプションを有効にしておくことが重要です。これにより、ユーザがレポート出力を .pptx ファイル拡張子でマシンに保存しなくても、PowerPoint 2000/2003 で PPTX レポート出力を開けるようになります。WebFOCUS Client 出力先変更設定の [保存レポート] オプションを使用して、ファイルタイプごとに WebFOCUS Client がレポート出力をどのようにユーザのマシンに送信するかを構成します。

このオプションは次のように設定することができます。

- **[保存レポート] オプションを無効 (いいえ) に設定する**　WebFOCUS Client 出力先変更設定の [保存レポート] オプションは、デフォルト設定で無効 (いいえ) になっています。[保存レポート] オプションが [いいえ] に設定されている場合、WebFOCUS Client がユーザのマシンに送信するレポート出力には、WebFOCUS Client 出力先変更設定の構成ファイル (mime.wfs) でレポートフォーマットに対して指定されたアプリケーション関連付け情報が付加されます。

  ユーザのマシンに PowerPoint 2007 以降がインストールされていない場合、PowerPoint のアプリケーション関連付け情報は認識されず、PowerPoint のメッセージが表示されます。PowerPoint 2000/2003 ユーザは、[保存] を選択し、拡張子が .pptx のファイル名を入力して、レポート出力をユーザのマシンに保存することができます。これで、ユーザは PowerPoint 2000/2003 から .pptx ファイルを直接開くことができます。

- **[保存レポート] オプションを有効 (はい) に設定する**　[保存レポート] オプションが [はい] に設定されている場合、WebFOCUS Client がユーザに送信するレポート出力は、WebFOCUS Client 出力先変更設定の構成ファイル (mime.wfs) で指定された拡張子のファイルになります。ファイルを受信した時点で [ファイルのダウンロード] ダイアログボックスが表示され、指定されたアプリケーションタイプでファイルを開くか、保存するかの選択がユーザに要求されます。[ファイルのダウンロード] ダイアログボックスには、PowerPoint PPTX ファイルタイプとして認識されたレポート出力の名前が .pptx 拡張子とともに表示されます。

注意：[ファイルのダウンロード] ダイアログボックスは、PowerPoint 2007 以降がマシンにインストールされているユーザも含めて、すべてのユーザに表示されます。

PowerPoint 2000/2003 ユーザがファイルを開くよう選択した場合、Microsoft Office 互換機能パックが .pptx ファイル拡張子を認識し、必要な変換を実行します。これにより、PowerPoint 2000/2003 で PowerPoint PPTX プレゼンテーションの読み取りが可能になります。

PowerPoint 2007 以降でファイルを開く選択をした場合、PowerPoint が .pptx ファイル拡張子を認識し、PowerPoint PPTX プレゼンテーションを読み取ります。

WebFOCUS Client 出力先変更設定についての詳細は、『ibi™ WebFOCUS® セキュリティ管理ガイド』を参照してください。

ブラウザおよび PowerPoint アプリケーションでの PowerPoint プレゼンテーションの表示

クライアントに送信された PowerPoint 出力がインターネットブラウザウィンドウと PowerPoint アプリケーションのどちらに表示されるかは、オペレーティングシステムおよびデスクトップの設定に基づいて決定されます。Windows 環境で PowerPoint 出力が [同じウィンドウで開く] に設定されている場合、WebFOCUS リクエストで生成されたプレゼンテーションは Internet Explorer ブラウザウィンドウに表示されます。[同じウィンドウで開く] オプションのチェックをオフにした場合、レポート出力はスタンドアロン PowerPoint アプリケーションウィンドウに表示されるため、WebFOCUS によって作成されたブラウザウィンドウはブランクになります。

- Windows XP 以前のバージョンでは、ファイルタイプの設定はデスクトップの Windows エクスプローラで管理します。この設定を行うには、エクスプローラの [ツール] メニューから [フォルダオプション] を選択し、[ファイルタイプ] タブをクリックした後、拡張子 (.ppt または .pptx) を選択し、[詳細設定] ボタンをクリックして [同じウィンドウで開く] のチェックをオンにします。

  Microsoft の「Support has ended for Office 2003」ステートメントによると、Office 2003 のサポートおよびアップデートは終了しています。Microsoft Office 2003 アプリケーションは今後も使用できますが、Microsoft では、サポートおよびアップデートを継続するため、新しいバージョンの Office へのアップグレードが推奨されています。組織で Office 2003 が使用されている場合は、「end of support page for Windows XP

SP3 and Office 2003」で詳細を確認してください。

- Windows 7 では、ワークシートをブラウザ内で開くようにするデスクトップ設定が削除されました。そのため、[フォルダオプション] ダイアログボックスを表示してこの動作を変更することができなくなり、代わりにレジストリ設定を変更する必要があります。

  注意：この変更方法は、PPT および PPTX フォーマットで共通です。異なる点は、生成する PowerPoint 出力のバージョンに基づいて選択するファイルタイプのみです。

PowerPoint でのテーブルとコンポーネントのグループ化

PowerPoint スライドにテーブルの各要素が配置される際は、それぞれの要素が個別のテキストボックスに挿入され、各要素の明示的な配置が可能になります。

PPTXGROUP パラメータを使用することで、PPTX レポートでこれらの要素をグループ化することができます。デザイン時に各要素はそれぞれ単一オブジェクトとして配置されていますが、実行時にはグループ内のすべてのオブジェクトを同時に回転、反転、移動、サイズ変更することができます。また、グループ内のすべてのオブジェクトの属性 (例、フォント、色、サイズ) を同時に変更することも、必要に応じてオブジェクトのグループを解除したり、後から再びグループ化したりすることも可能です。

WebFOCUS では、これらのオブジェクトはレポートコンポーネントごとにグループ化されます。データ、見出しと脚注、イメージなど、レポートコンポーネント (スタンドアロンレポート) 内のオブジェクトがグループ化されます。複合レポートでは、各コンポーネントレポートはそれぞれ独立してグループ化されます。ただし、コンポーネント以外の要素 (例、描画オブジェクト、線、イメージ) は、どのグループにも含まれません。

PowerPoint でのテーブルとコンポーネントのグループ化

```
SET PPTXGROUP = {ON|OFF}
```

このコマンドは、次のように記述してレポート内から発行することもできます。

```
ON TABLE SET PPTXGROUP = {ON|OFF}
```

スタイルシートでは、次のように記述します。

```
TYPE=REPORT, PPTXGROUP = {ON|OFF}
```

説明

ON

PPTX レポートでの複数の要素のグループ化を有効にします。

OFF

複数の要素をグループ化しません (従来の動作)。デフォルト値は OFF です。

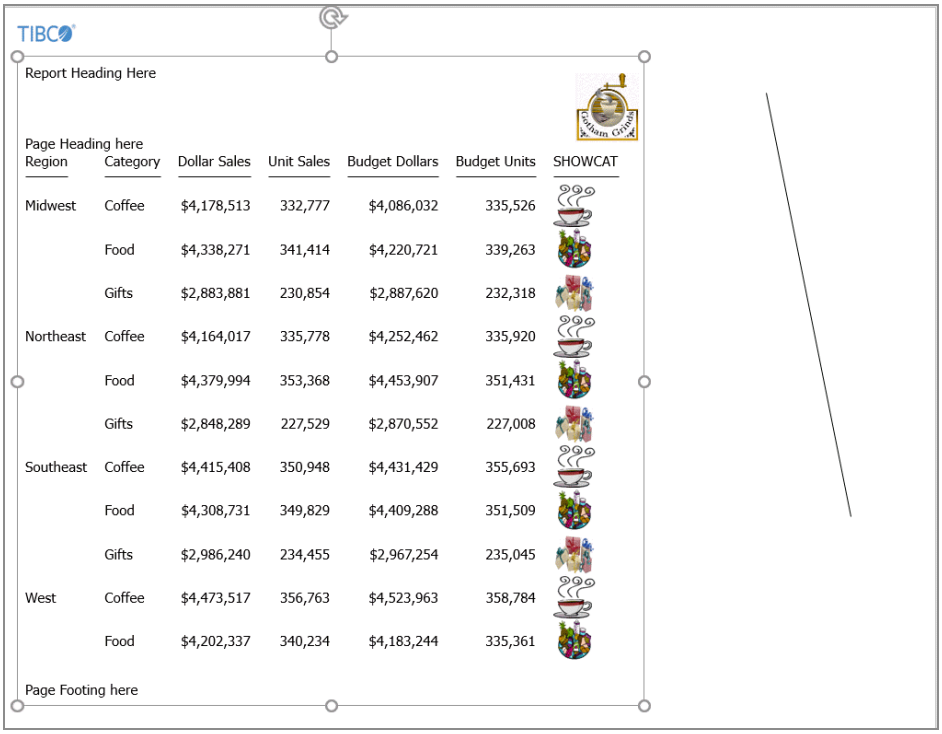
標準レポートでのテーブルとコンポーネントのグループ表示

次の標準レポートでは、グループ化は、TYPE=REPORT レベルで指定された主要レポート要素 (イメージおよび描画オブジェクトを除く) に対して定義されます。

```
TABLE FILE GGSALES
SUM DOLLARS/D12CM UNITS/D12C BUDDOLLARS/D12CM BUDUNITS/D12C
COMPUTE SHOWCAT/A100=CATEGORY||'.GIF';
BY REGION
BY CATEGORY
ON TABLE SUBHEAD
"Report Heading Here"
" "
" "
" "
HEADING
"Page Heading here"
FOOTING
"Page Footing here"
ON TABLE SUBFOOT
"Report Footing Here"
ON TABLE PCHOLD AS GROUPTEST FORMAT PPTX
ON TABLE SET STYLE *
TYPE=REPORT, PPTXGROUP = ON, SQUEEZE=ON, FONT=TAHOMA,SIZE=12,
ORIENTATION=LANDSCAPE, $
```

```
TYPE=REPORT, OBJECT=LINE, POSITION=(9 1), DIMENSION=(1 5), COLOR=BLUE, $
TYPE=REPORT, IMAGE=TIBCO, POSITION=(0 0),SIZE=(1 .6), $
TYPE=TABHEADING, IMAGE=GGLOGO.GIF, POSITION=(6.5 .10), $
TYPE=DATA,COLUMN=SHOWCAT, IMAGE=(SHOWCAT), PRESERVERATIO=ON,
SIZE=(.5 .5), $
END
```

出力結果は次のとおりです。



Region	Category	Dollar Sales	Unit Sales	Budget Dollars	Budget Units	SHOWCAT
Midwest	Coffee	$4,178,513	332,777	$4,086,032	335,526	
	Food	$4,338,271	341,414	$4,220,721	339,263	
	Gifts	$2,883,881	230,854	$2,887,620	232,318	
Northeast	Coffee	$4,164,017	335,778	$4,252,462	335,920	
	Food	$4,379,994	353,368	$4,453,907	351,431	
	Gifts	$2,848,289	227,529	$2,870,552	227,008	
Southeast	Coffee	$4,415,408	350,948	$4,431,429	355,693	
	Food	$4,308,731	349,829	$4,409,288	351,509	
	Gifts	$2,986,240	234,455	$2,967,254	235,045	
West	Coffee	$4,473,517	356,763	$4,523,963	358,784	
	Food	$4,202,337	340,234	$4,183,244	335,361	

複合レポートでのテーブルとコンポーネントのグループ表示

次の複合レポートでは、グループ化は、レポートコンポーネントに対して定義されます。ページ上のその他のオブジェクト (例、グラフイメージ、ロゴイメージ、線、テキストボックス) はグループ化に含まれません。

```
SET HTMLARCHIVE=ON
SET PPTXGRAPHTYPE=PNG
SET PPTXGROUP=ON
COMPOUND LAYOUT PCHOLD FORMAT PPTX
```

```
UNITS=IN, $
SECTION=section1, LAYOUT=ON, MERGE=OFF, ORIENTATION=LANDSCAPE, PAGESIZE=PPT
Slide, $
PAGELAYOUT=1, NAME='Page layout 1', TEXT='Page layout 1', BOTTOMMARGIN=0.5,
TOPMARGIN=0.5, $
OBJECT=STRING, NAME='text1', TEXT='<left>Grouping is supported within the
component reports of a compound report:
<ul type=disc>
<LI>Each table / report is grouped together. </LI>
<LI>Drawing objects such as images and lines are not included in any group.
</LI>
<LI>Charts are inserted as images.</LI></ul><div><br></div><br></left>',
POSITION=(2.764 4.958), MARKUP=ON, WRAP=ON, DIMENSION=(4.000 1.992),
FONT='TREBUCHET MS', COLOR=RGB(0 0 0), SIZE=10, $
COMPONENT='report1', TEXT='report1', POSITION=(1.028 2.083), DIMENSION=(4.431
1.667), $
COMPONENT='chart2', TEXT='chart2', POSITION=(4.972 1.319), DIMENSION=(4.456
3.085), COMPONENT-TYPE=GRAPH, $
OBJECT=BOX, NAME='line1', POSITION=(0.498 6.500),
DIMENSION=(9.167 0.022), BACKCOLOR=BLACK, BORDER-COLOR=BLACK, $
OBJECT=BOX, NAME='line2', POSITION=(0.502 1.097),
DIMENSION=(9.167 0.022), BACKCOLOR=BLACK, BORDER-COLOR=BLACK, $
OBJECT=IMAGE, NAME='image4', IMAGE=tibco.gif, ALT='',
POSITION=(0.502 0.499), DIMENSION=(1.861 0.7), $
END
```

```
SET COMPONENT='report1'
TABLE FILE GGSALES
SUM
   GGSALES.SALES01.DOLLARS
BY GGSALES.SALES01.REGION
ACROSS LOWEST GGSALES.SALES01.CATEGORY
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
     INCLUDE=IBFS:/FILE/IBI_HTML_DIR/ibi_themes/Warm.sty,$
$
ENDSTYLE
END
```

```
SET COMPONENT='chart2'
SET PAGE-NUM=NOLEAD
SET ARGRAPHENGINE=JSCHART
SET EMBEDHEADING=ON
```

```
SET GRAPHDEFAULT=OFF
GRAPH FILE GGSALES
SUM GGSALES.SALES01.DOLLARS
BY GGSALES.SALES01.CATEGORY
ACROSS GGSALES.SALES01.REGION
ON GRAPH PCHOLD FORMAT HTML
ON GRAPH SET VZERO OFF
ON GRAPH SET GRWIDTH 1
ON GRAPH SET UNITS 'PIXELS'
ON GRAPH SET HAXIS 770.0
ON GRAPH SET VAXIS 405.0
ON GRAPH SET GRMERGE ADVANCED
ON GRAPH SET GRMULTIGRAPH 0
ON GRAPH SET GRLEGEND 1
ON GRAPH SET GRXAXIS 1
ON GRAPH SET LOOKGRAPH HBAR
ON GRAPH SET STYLE *
```

```
*GRAPH_SCRIPT
setPieDepth(0);
setPieTilt(0);
setDepthRadius(0);
setCurveFitEquationDisplay(false);
setPlace(true);
*END
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/ibi_themes/Warm.sty,$
TYPE=REPORT, TITLETEXT='WebFOCUS Report', $
*GRAPH_SCRIPT
setLegendPosition(4);
*GRAPH_JS_FINAL
"blaProperties": {
    "orientation": "horizontal"
},
"agnosticSettings": {
    "chartTypeFullName":"Bar_Clustered_Horizontal"
}
*END
ENDSTYLE
END
-RUN

COMPOUND END
```

出力結果は次のとおりです。

Category Region	Coffee	Food	Gifts
Midwest	4178513	4338271	2883881
Northeast	4164017	4379994	2848289
Southeast	4415408	4308731	2986240
West	4473517	4202337	2977092

日付およびページ/スライド番号

日付およびページ番号は、単一レポートにも複合レポートにも追加することができます。

- 単一レポートの場合、TABPAGENO 機能および関連する属性を使用します。
- 複合レポートの場合、データおよびページ番号をテキストオブジェクトに追加します。

テキストオブジェクトのテキストフォーマットマークアップタグ

注意：テキストに開始キャレット文字 (<) が含まれている場合、テキストの一部であるキャレット文字それぞれの後に、ブランクを 1 つ入力する必要があります (例、“< 250”)。そうしない場合、開始キャレット文字に続くすべての文字は、マークアップタグの開始と解釈されて、テキストとして表示されません。

フォントのプロパティ

font タグは、face、size、color という 3 つの属性をサポートします。

```
<font face="font" size=[+|-]n color=color_code>text</font>
```

以下はその例です。

```
<font face="New Century Schoolbook">Test1</font>
<font face="Times" size=12>test2</font>
<font face="Times New Roman" color=#0000FF size=+4>Test3</font>
<font size=-2 face="Times New Roman" color=#0000FF >Test4</font>
```

テキストスタイル

サポートされるテキストスタイルは、太字、斜体、下線、上付き文字です。

太字 - `<b>text</b>`

斜体 - `<i>text</i>`

下線 - `<u>text</u>`

上付き文字 - `<sup>text</sup>`

改行

テキストの一部分の後の改行タグは、次のテキスト部分を新しい行で開始します。改行タグに終了タグはありません。

```
<br>
```

テキストの位置揃え

位置揃えオプションは、指定した改行同様、折り返しテキストにも適用されます。横方向の位置揃えと縦方向の位置揃えの両方がサポートされます。

- **横方向の位置揃え**

  左揃え

  ```
  <left>text</left>
  ```

  右揃え

  ```
  <right>text</right>
  ```

  中央揃え

```
<center>text</center>
```

両端揃え

```
<full>text</full>
```

- **縦方向の位置揃え**

  上揃え

  ```
  <top>text</top>
  ```

  中央揃え

  ```
  <mid>text</mid>
  ```

  下揃え

  ```
  <bottom>text</bottom>
  ```

箇条書きリスト (番号なし)

項目リストを囲む番号なしの (ul) リストタグです。各項目は、リスト項目タグ (li) で囲まれます。リストの開始タグと終了タグは、それぞれ個別の行として記述する必要があります。各リスト項目は、新しい行に入力する必要があります。

```
<ul>
<li>list item1</li>
<li>list item2</li>
  .
  .
  .
</ul>
```

デフォルト設定の行頭文字は、黒丸 (disc) です。白丸 (circle) または黒い四角 (square) を指定することもできます。

```
<ul type=disc>
```

```
<ul type=circle>
```

```
<ul type=square>
```

箇条書きリスト (番号付き)

各項目に連続番号または連続文字の付いたリストを囲む、番号付き (ol) リストタグです。各項目は、リスト項目タグ (li) で囲まれます。リストの開始タグと終了タグは、それぞれ個別の行として記述する必要があります。各リスト項目は、新しい行に入力する必要があります。

```
<ol>
<li>list item1</li>
<li>list item2</li>
  .
  .
  .
</ol>
```

デフォルト設定では、リストの番号としてアラビア数字 (type=1) が使用されます。指定できる順序のタイプは次のとおりです。

アラビア数字 (デフォルト設定) - `<ol type=1>`

小文字のアルファベット - `<ol type=a>`

大文字のアルファベット - `<ol type=A>`

小文字のローマ数字 - `<ol type=I>`

大文字のローマ数字 - `<ol type=I>`

ハイパーリンク

ハイパーリンクは、PPTX 出力ファイルのテキストマークアップに含めることができます。

アンカーマークアップタグの構文は、HTML アンカー構文のサブセットです。

```
<a href="hyperlink">Text to display</a>
```

説明

hyperlink

テキストをクリックしたときの移動先ハイパーリンクです。

Text to display

ハイパーリンクとして表示するテキストです。

以下はその例です。

```
<a href="http://www.example.com/help.htm">Click here for help</a>
```

アンカーマークアップタグでは、その他の属性はサポートされません。

ページ番号

次の 2 つの擬似 HTML タグを使用して、統合された複合レイアウトレポートのページマスターのテキストに、ページ番号を埋め込むことができます。

現在のページ番号 - `<ibi-page-number/>`

総ページ数 - `<ibi-total-pages/>`

MARKUP=ON に設定した場合、最大ページ数を表示するためのスペースが割り当てられるため、ページ番号と、それに続くテキストとの間隔が広くなる場合があります。ページ番号タグが含まれたテキストオブジェクト内で余分なスペースを削除するには、次のように指定します。

- テキストオブジェクトに特定のスタイルを指定する必要がない場合、マークアップタグを挿入せず、MARKUP=OFF に設定します。

  ```
  MARKUP=OFF, TEXT='Page <ibi-page-number/> of <ibi-total-pages/> of Sales
  Report', $
  ```

  次のように表示されます。

  ```
  Page 1 of 100 of Sales Report
  ```

- テキストオブジェクトに特定のスタイルを指定する必要がある場合、MARKUP=ON に設定する必要があります。MARKUP=ON に続けて、WRAP=OFF に設定します。文字列内では、2 つのページ番号変数間にスタイルタグを配置しないでください。「Page n of m」という文字列の周囲にはタグを使用することができます。次のようにコーディングすると、ページ番号の文字列に余分なスペースが表示されなくなります。

  ```
  MARKUP=ON, WRAP=OFF, TEXT='<font face="ARIAL" size=10><i>Page <ibi-page-
  number/> of <ibi-total-pages/> of Sales Report </i>', $
  ```

  次のように表示されます。

  Page 1 of 100 of Sales Report

日付

レポート出力に日付を表示するには、ページマスターのテキストオブジェクトに、&DATEtrMDYY など、WebFOCUS の日付変数を挿入します。

マークアップタグによる複合レイアウトテキストオブジェクトのフォーマット設定

次のリクエストは、マークアップタグ付きテキストオブジェクトを PPTX 出力ファイルに表示します。

重要：テキストマークアップ構文には、非表示のキャリッジリターンやラインフィード文字を含めることはできません。このマニュアルの例では、サンプルコードにラインフィード文字が追加され、印刷ページ幅にコードが収まるようコード行が折り返されています。使用する環境でこのサンプルコードを実行するには、コードをテキストエディタにコピーし、各行の末尾に移動して Delete キーを押すことにより、テキストマークアップオブジェクト内のラインフィード文字を削除します。場合によっては、文字列の構造を保持するために、ブランクを追加する必要があります。

```
SET PAGE-NUM=OFF
SET LAYOUTGRID=ON
TABLE FILE GGSALES
BY REGION NOPRINT
ON TABLE PCHOLD AS LINESP1 FORMAT PPTX
ON TABLE SET STYLE *
type=report, size=8, $
object=string, position=(1 1), dimension=(7 3), wrap=on, markup=on,
 linespacing=multiple(3),
 text='<b><font face="Arial" size=12>This paragraph is triple-spaced
 (LINESPACING=MULTIPLE(3)):</font></b>
 <full>Our <i>primary</I> goal for fiscal 2006 was to accelerate our
 transformation to customer centricity.In this letter, I’d like to
 give you an update on this work, which contributed to the 22-percent
 increase in earnings from continuing operations we garnered for fiscal
 2006.Since the past is often prologue to the future, I’d like to
 describe how customer centricity is influencing not only our goals for
 fiscal 2007, but also our long-term plans.At Gotham Grinds, customer
 centricity means treating each customer as a unique individual, meeting
 their needs with end-to-end solutions, and engaging and energizing our
 employees to serve them.</full>', $
ENDSTYLE
END
```

以下は、このリクエストの説明です。

- データソースのフィールドは、どれも表示されません。出力には、テキストオブジェクトのみが表示されます。
- SET LAYOUTGRID コマンドは、罫線を表示して、テキストオブジェクトの座標およびサイズを示します。
- OBJECT=STRING 宣言は、LINESPACING=MULTIPLE(3) によって、3 行の行間を指定します。
- 次のテキストは、太字、Arial、12 ポイント (デフォルト値は TYPE=REPORT で指定される 8 ポイント) で表示されます。

  'This paragraph is triple-spaced (LINESPACING=MULTIPLE(3)):'

  このフォーマットのマークアップは、次のとおりです。

  ```
  <b><font face="Arial" size=12>This paragraph is triple-spaced
  (LINESPACING=MULTIPLE(3)):</font></b>
  ```

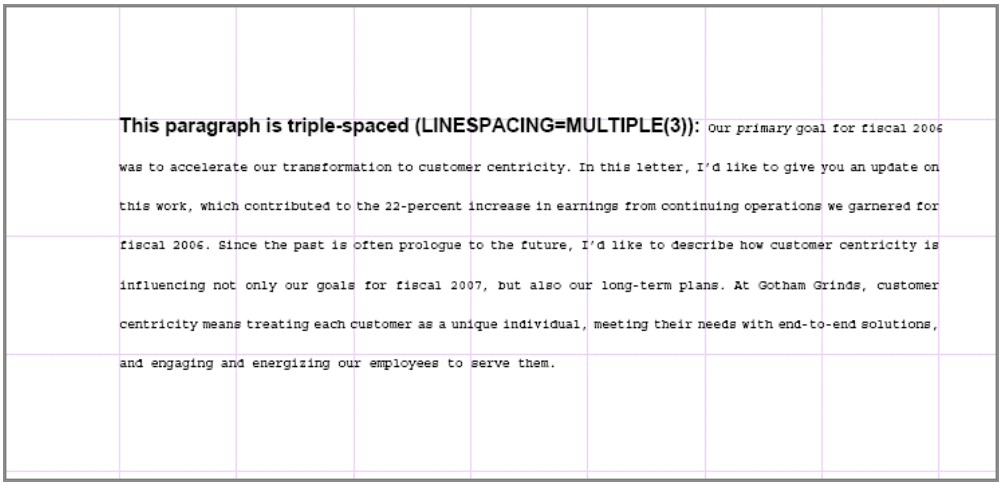
  ただし、イメージのサイズは、ページに収まるように変更されているため、小さいフォントで表示されます。
- 残りのテキストは、両端揃え (左右の端で揃える) で表示されます。

  ```
  <full>Our ... </full>
  ```

- 次のマークアップは、「primary」というテキストを斜体で表示します。

  ```
  <i>primary</I>
  ```

出力結果は次のとおりです。



ページマスターへのテキストおよび線オブジェクトの配置

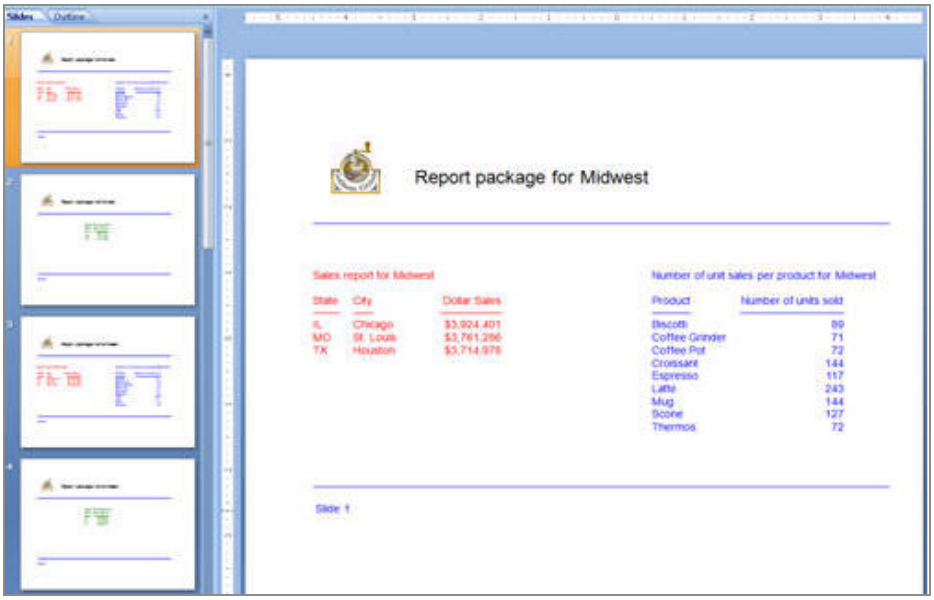
次のリクエストは、ページマスター上のヘッダレポートとコンポーネントレポートとの間に線を配置し、各ページの最下部に線とテキストを配置します。

```
SET PAGE-NUM=OFF
SET SQUEEZE=ON
COMPOUND LAYOUT PCHOLD FORMAT PPTX
SECTION=S1, LAYOUT=ON, MERGE=ON, ORIENTATION=LANDSCAPE, $
PAGELAYOUT=ALL, $
COMPONENT=HEADER, TYPE=REPORT, POSITION=(1 1), DIMENSION=(4 4), $
OBJECT=STRING, POSITION=(1 6.6), MARKUP=ON,
TEXT='<font face="Arial" color=#0000FF size=12> Slide <ibi-page-number/>
        </font> ', WRAP=ON, DIMENSION=(4 4),$
OBJECT=LINE, POSITION=(1 2.5), ENDPOINT=(9.5 2.5),
        BORDER-COLOR=BLUE,$
OBJECT=LINE, POSITION=(1 6.5), ENDPOINT=(9.5 6.5),
        BORDER-COLOR=BLUE,$
PAGELAYOUT=1, $
COMPONENT=R1, TYPE=REPORT, POSITION=(1 3), DIMENSION=(4 4), $
COMPONENT=R2, TYPE=REPORT, POSITION=(6 3), DIMENSION=(4 4), $
PAGELAYOUT=2, $
COMPONENT=R3, TYPE=REPORT, POSITION=(4 3), DIMENSION=(4 4), $
END
```

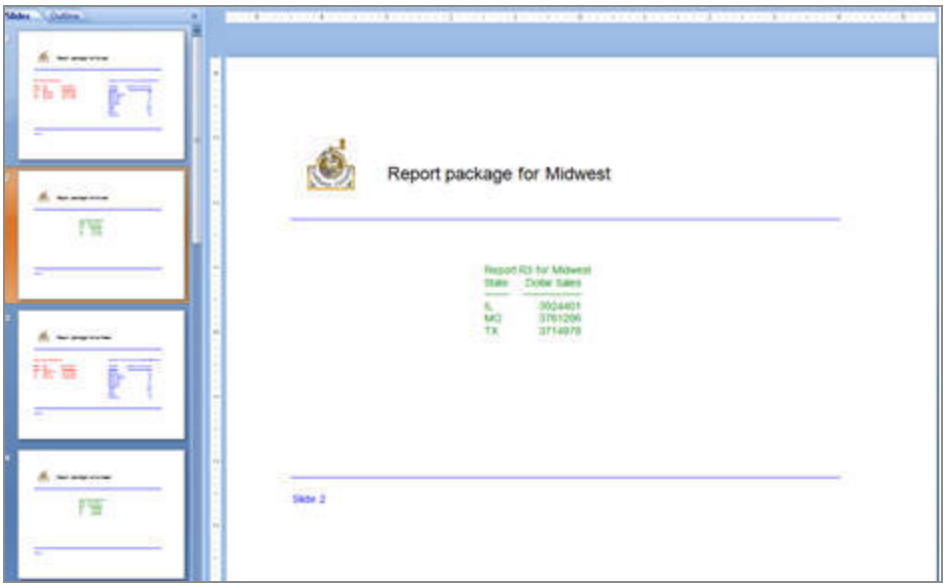
```
SET COMPONENT=HEADER
TABLE FILE GGSALES
" "
"Report package for <REGION"
BY REGION NOPRINT
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, SIZE=20, $
TYPE=REPORT, IMAGE=gglogo.gif, POSITION=(+.25 +.25), $
TYPE=HEADING, LINE=2, ITEM=1, POSITION=1.5, $
END
SET COMPONENT=R1
TABLE FILE GGSALES
"Sales report for <REGION"
" "
SUM DOLLARS/F8M
```

```
BY REGION NOPRINT
BY ST
BY CITY
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, COLOR=RED, SQUEEZE=ON, $
END
SET COMPONENT=R2
TABLE FILE GGSALES
"Number of unit sales per product for <REGION"
" "
SUM CNT.UNITS AS 'Number of units sold'
BY REGION NOPRINT
BY PRODUCT
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, COLOR=BLUE, SQUEEZE=ON, $
END
SET COMPONENT=R3
TABLE FILE GGSALES
"Report R3 for <REGION"
BY REGION NOPRINT
SUM DOLLARS BY ST
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, COLOR=GREEN, $
END
COMPOUND END
```

次は出力結果の 1 ページ目です。



出力結果の 2 ページ目には、同一の描画オブジェクトが配置されています。



PPTX レポート出力でのテキストの縦方向配置マークアップ

次のリクエストは、3 つのボックスを作成し、各ボックス内にテキストオブジェクトを配置します。

- 左側のボックスでは、縦方向の上揃えでテキストが上段に配置されています。
- 中央のボックスでは、縦方向の中央揃えでテキストが中段に配置されています。
- 右側のボックスでは、縦方向の下揃えでテキストが下段に配置されています。

重要：テキストマークアップ構文には、非表示のキャリッジリターンやラインフィード文字を含めることはできません。このマニュアルの例では、サンプルコードにラインフィード文字が追加され、印刷ページ幅にコードが収まるようコード行が折り返されています。使用する環境でこのサンプルコードを実行するには、コードをテキストエディタにコピーし、各行の末尾に移動して Delete キーを押すことにより、テキストマークアップオブジェクト内のラインフィード文字を削除します。場合によっては、文字列の構造を保持するために、ブランクを追加する必要があります。

```
SET PAGE-NUM=OFF
TABLE FILE GGSALES
BY REGION NOPRINT
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
```

```
type=report, font=arial, size=10, $
object=box, position=(1 1), dimension=(6 1), $
object=line, position=(3 1), endpoint=(3 2), $
object=line, position=(5 1), endpoint=(5 2), $
object=string, text='<top>Vertically aligned text within a text object using
top alignment.</top>', position=(1.05 1), dimension=(2 1), linespacing=exact
(.15), markup=on, wrap=on, $
object=string, text='<mid>Vertically aligned text within a text object using
middle alignment.</mid>', position=(3.05 1), dimension=(2 1),
linespacing=exact(.15), markup=on, wrap=on, $
object=string, text='<bottom>Vertically aligned text within a text object
using bottom alignment.</bottom>', position=(5.05 .9), dimension=(2
1),linespacing=exact(.15), markup=on, wrap=on,
$
END
```

出力結果は次のとおりです。

Vertically aligned text within a text object using top alignment.	Vertically aligned text within a text object using middle alignment.	Vertically aligned text within a text object using bottom alignment.

黒丸、白丸、黒い四角を使用した箇条書きリスト (番号なし) の表示

番号なし (ul) リストタグは、箇条書きリストを囲みます。リスト内の各項目は、特定の形状またはデザインを使用する箇条書きで表示されます。各項目または点は、リスト項目タグ (li) で囲まれます。各点の開始タグと終了タグは、連続した 1 行に配置することも、それぞれ別の行に配置することもできます。各リストが新しい行に配置される場合は、各終了タグの後に円記号 (¥) を挿入します。

黒丸、白丸、黒い四角を使用した箇条書きリスト (番号なし) の表示

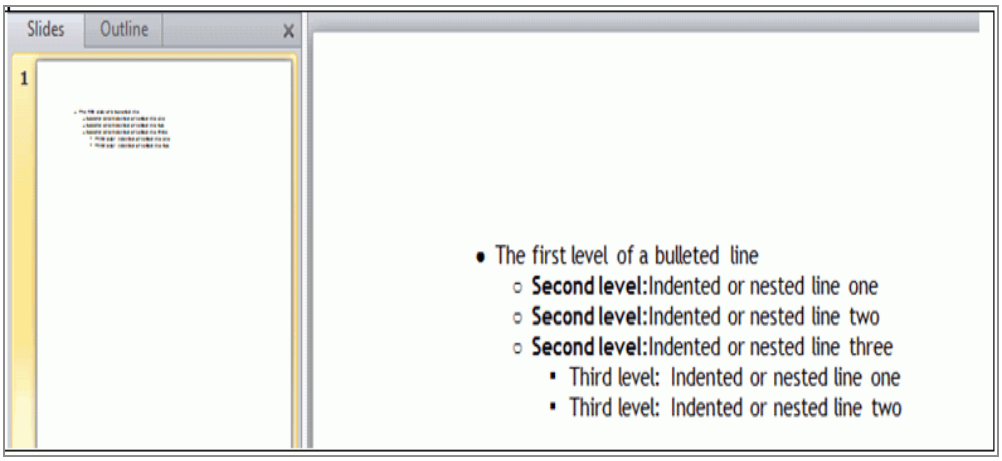
次のリクエストは、特定の形状およびデザインを使用して箇条書きリストを表示します。

```
<ul>
<li>list item1</li>¥
<li>list item2</li>¥
</ul>
```

```
SET HTMLARCHIVE=ON
COMPOUND LAYOUT PCHOLD FORMAT PPTX
UNITS=IN, $
SECTION=section1, LAYOUT=ON, MERGE=OFF, ORIENTATION=PORTRAIT,
PAGESIZE=Letter, SHOW_GLOBALFILTER=OFF, $
PAGELAYOUT=1, NAME='Page layout 1', text='Page layout 1',
BOTTOMMARGIN=0.5, TOPMARGIN=0.5, $
OBJECT=STRING, NAME='text1', TEXT='<font face="TREBUCHET MS" size=10>¥
<UL>
<LI>The first level of a bulleted line </LI>¥
<UL>
<LI><b>Second level:</b> Indented or nested line one</LI>¥
<LI><b>Second level:</b> Indented or nested line two</LI>¥
<LI><b>Second level:</b> Indented or nested line three</LI>¥
<UL>¥
<LI>Third level:Indented or nested line one</LI>¥
<LI>Third level:Indented or nested line two</LI>¥
</UL>¥
```

```
<BR><BR><BR><U><DIV><BR></DIV></U></font>', POSITION=(0.938 0.938),
MARKUP=ON, WRAP=ON, DIMENSION=(6.563 4.167), $
COMPONENT='DfltCmpt1', POSITION=(0 0), DIMENSION=(0 0), $
END
SET COMPONENT='DfltCmpt1'
TABLE FILE SYSCOLUM
" "
SUM TBNAME NOPRINT
IF READLIMIT EQ 1
ON TABLE SET PREVIEW ON
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
END
COMPOUND END
```

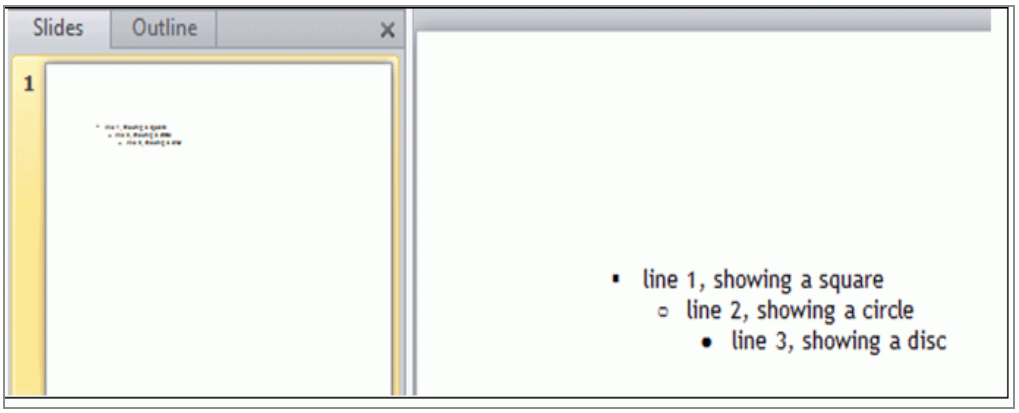
出力結果は次のとおりです。



デフォルトの箇条書きタイプでは、黒丸が使用されます。次のリクエストのように、白丸または黒い四角を指定することもできます。

```
SET HTMLARCHIVE=ON
COMPOUND LAYOUT PCHOLD FORMAT PPTX
UNITS=IN, $
SECTION=section1, LAYOUT=ON, MERGE=OFF, ORIENTATION=PORTRAIT,
PAGESIZE=Letter, SHOW_GLOBALFILTER=OFF, $
PAGELAYOUT=1, NAME='Page layout 1', text='Page layout 1',
BOTTOMMARGIN=0.5, TOPMARGIN=0.5, $
OBJECT=STRING, NAME='text1', TEXT='<font face="TREBUCHET MS" size=10>¥
<UL type=square>¥
<LI> line 1, showing a square </LI>¥
<UL type=circle>¥
<LI> line 2, showing a circle </LI>¥
<UL type=disc>¥
<LI> line 3, showing a disc </LI>¥
</UL>¥
<BR><BR><BR><U><DIV><BR></DIV></U></font>', POSITION=(0.938 0.938),
MARKUP=ON, WRAP=ON, DIMENSION=(6.563 4.167), $
COMPONENT='DfltCmpt1', POSITION=(0 0), DIMENSION=(0 0), $
END
SET COMPONENT='DfltCmpt1'
TABLE FILE SYSCOLUM
" "
SUM TBNAME NOPRINT
IF READLIMIT EQ 1
ON TABLE SET PREVIEW ON
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
END
COMPOUND END
```

出力結果は次のとおりです。



PowerPoint PPTX レポート要素へのイメージの挿入

WebFOCUS では、レポートの各要素または領域にイメージを配置することができます。ロゴなどのイメージは、レポートに企業イメージの統一感や視覚的訴求力を与えます。データ固有のイメージを見出しや脚注に配置することで、データの意味を明確にしたり、レポートの外観を整えたりできます。イメージは、WebFOCUS Reporting Server 上の EDAPATH または APPPATH ディレクトリに存在する必要があります。ファイルが検索パスにない場合、完全なパス名を指定します。

イメージはすべて、スタイルシートの POSITION 属性で明示的に定義された位置に基づいて、指定の領域に配置されます。

イメージは、WebFOCUS レポートの任意の領域または要素に配置することができます。サポートされるイメージフォーマットには、.gif、.jpg、.png などがあります。イメージは、POSITION 属性を使用して xy 座標を設定したり、SIZE 属性を使用して高さと幅を設定したりして、位置やサイズの調整を行うことができます。イメージの位置揃えはサポートされません。

注意：グラフに使用する最高品質のイメージフォーマットは PNG です。PNG フォーマットでは、透明度が設定できるほか、スライド背景内のスタイルとの統合に優れています。

WebFOCUS PPTX レポートへのイメージの挿入

```
TYPE={REPORT|HEADING|data}, IMAGE={file|(column)}
[,BY=byfield] [,SIZE=(wh)] ,$
```

説明

REPORT

レポート本文にイメージを埋め込みます。イメージはレポートの背景に表示されます。REPORT がデフォルト値です。

HEADING

見出しまたは脚注にイメージを埋め込みます。有効な値は、TABHEADING、TABFOOTING、FOOTING、HEADING、SUBHEAD、および SUBFOOT です。見出しまたは脚注内に十分な余白をとり、見出しや脚注のテキストと重ならないようにします。また、スポットマーカーを使用して、イメージの右側に見出しまたは脚注のテキストを配置することもできます。

data

イメージを配置するデータフィールドを定義します。COLUMNS=*column title* とともに使用して、イメージを固定する特定のレポート列を識別する必要があります。

file

イメージファイル名です。このファイルは、EDAPATH または APPPATH で指定された、WebFOCUS Reporting Server 上のディレクトリに存在する必要があります。ファイルが検索パスにない場合、完全なパス名を指定します。GIF を指定する場合は、ファイルの拡張子を省略できます。

レポート見出しおよび脚注へのイメージの挿入

次のリクエストは、レポートの見出しおよび脚注にイメージを挿入します。

```
TABLE FILE EMPDATA
SUM
EMPDATA.EMPDATA.SALARY
```

```
BY LOWEST EMPDATA.EMPDATA.DEPT
BY EMPDATA.EMPDATA.LASTNAME
ON EMPDATA.EMPDATA.DEPT SUBFOOT "Subfoot"
" "
ON EMPDATA.EMPDATA.DEPT PAGE-BREAK
ON TABLE SUBHEAD
"Report Heading"
" "
HEADING
"Page Heading"
" "
FOOTING
"Page Footing"
" "
ON TABLE SUBFOOT "Report Footing"
" "
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
PAGESIZE='PPT Slide',
  ORIENTATION=LANDSCAPE,
$
TYPE=REPORT,
  OBJECT=STATUS-AREA,
  JUSTIFY=LEFT,
  PAGE-LOCATION=BOTTOM,
$
```

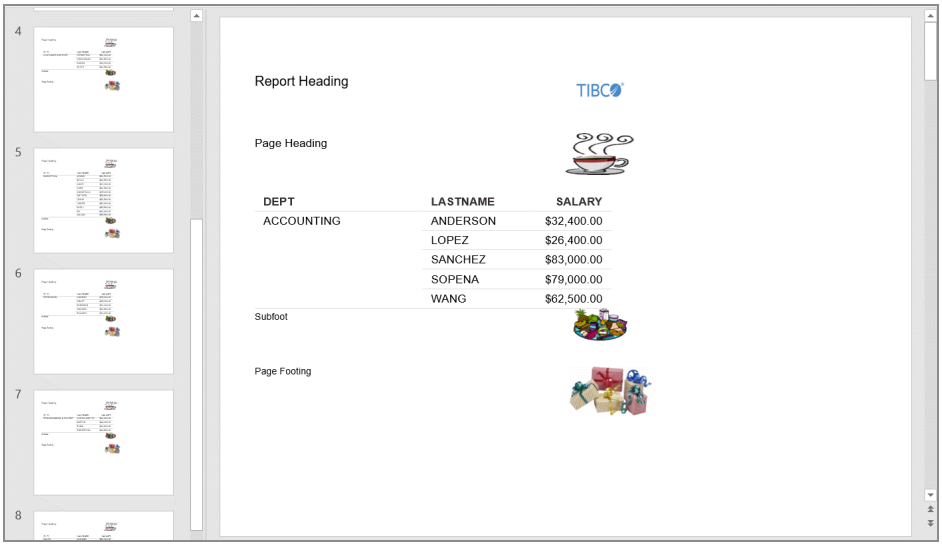
```
TYPE=REPORT,
  GRID=OFF,
  FONT='ARIAL',
  SIZE=12,
  STYLE=NORMAL,
  SQUEEZE=ON,
  TOPGAP=0.05,
  BOTTOMGAP=0.05,
  BORDER-COLOR=RGB(219 219 219),
  TITLELINE=SKIP,
  TOPMARGIN=.75,
  LEFTMARGIN=.5,
$
TYPE=TITLE,
  COLOR=RGB(51 51 51),
  STYLE=-UNDERLINE +BOLD,
$
```

```
TYPE=DATA,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=SUBTOTAL,
  STYLE=BOLD,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=TABHEADING,
  SIZE=14,
  JUSTIFY=LEFT,
$
TYPE=TABHEADING,
  IMAGE=tibco.gif,
  POSITION=(+4.500000 +0.000000),SIZE=(1 .6),
$
TYPE=TABFOOTING,
  SIZE=10,
$
```

```
TYPE=SUBHEAD,
  BACKCOLOR=RGB(246 246 246),
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=SUBHEAD,
  BY=1,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(102 102 102),
$
TYPE=SUBHEAD,
  OBJECT=FIELD,
  STYLE=BOLD,
$
TYPE=SUBFOOT,
  SIZE=9,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=TABFOOTING,
  IMAGE=gglogo.gif,
  POSITION=(+4.500000 +0.000000),
$
```

```
TYPE=HEADING,
  SIZE=12,
$
TYPE=HEADING,
  IMAGE=coffee.gif,
  POSITION=(+4.500000 +0.000000),SIZE=(1 .6),
$
TYPE=FOOTING,
  SIZE=10,
$
TYPE=FOOTING,
  STYLE=BOLD,
  JUSTIFY=LEFT,
  IMAGE=gifts.gif,
  POSITION=(+4.500000 +0.000000),
$
TYPE=SUBFOOT,
  SIZE=10,
$
TYPE=SUBFOOT,
  IMAGE=food.gif,
  POSITION=(+4.500000 +0.000000),
  SIZE=(1.000000 0.500000),
$
ENDSTYLE
END
```

出力結果は次のとおりです。



レポートのデータセルへのイメージの挿入

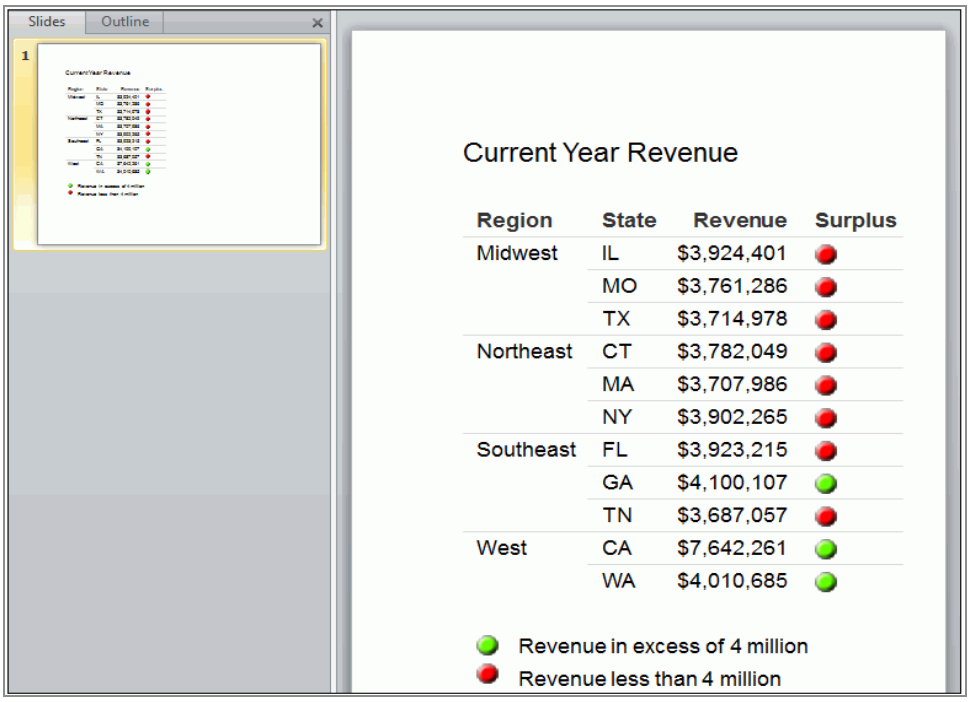
次のリクエストは、レポートのデータセルにイメージを挿入します。

```
APP PATH IBISAMP IBIDEMO
TABLE FILE GGSALES
SUM DOLLARS/D17M AS 'Revenue'
COMPUTE Surplus/A15 = IF DOLLARS GE 4000000 THEN 'g1.gif' ELSE 'r1.gif';
BY REGION
BY ST
ON TABLE SUBHEAD
"Current Year Revenue"
FOOTING
"Revenue in excess of 4 million"
"Revenue less than 4 million"
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET SQUEEZE ON
ON TABLE SET STYLE *
TYPE=REPORT,
     GRID=OFF,
     FONT='ARIAL',
     SIZE=14,
     STYLE=NORMAL,
     SQUEEZE=ON,
     TOPGAP=0.05,
     BOTTOMGAP=0.05,
     BORDER-COLOR=RGB(219 219 219),
     TITLELINE=SKIP,
     ORIENTATION = LANDSCAPE,TOPMARGIN=1,LEFTMARGIN=1,
$
TYPE=DATA, COLUMN=Surplus, IMAGE=(Surplus), SIZE=(.2 .2),$
TYPE=FOOTING,IMAGE=g1.gif, position=(.122 .055), SIZE=(.2 .2 ),$
TYPE=FOOTING,line=1,item=1,position=.5,$
TYPE=FOOTING,IMAGE='r1.gif', position=(.122 .33), SIZE=(.2 .2),$
TYPE=FOOTING,line=2,item=1, position=.5,$
TYPE=DATA,
     BORDER-TOP=LIGHT,
     BORDER-TOP-COLOR=RGB(219 219 219),
$
```

```
TYPE=TITLE,
     COLOR=RGB(51 51 51),
```

```
     STYLE=-UNDERLINE +BOLD,
$
TYPE=TABHEADING,
     SIZE=18,
     JUSTIFY=LEFT,
$
TYPE=TABFOOTING,
     SIZE=10,
$
TYPE=HEADING,
     JUSTIFY=LEFT,
     SIZE=12,
$
TYPE=SUBHEAD,
     BACKCOLOR=RGB(246 246 246),
     BORDER-TOP=LIGHT,
     BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=SUBHEAD,
      BY=1,
      BORDER-TOP=LIGHT,
      BORDER-TOP-COLOR=RGB(102 102 102),
$
TYPE=SUBHEAD,
     OBJECT=FIELD,
     STYLE=BOLD,
$
TYPE=SUBFOOT,
     SIZE=9,
     BORDER-TOP=LIGHT,
     BORDER-TOP-COLOR=RGB(219 219 219),
$
END
```

レポート出力は次のとおりです。



PNG イメージフォーマットでの PPTX グラフの表示

PPTXGRAPHTYPE 属性を使用すると、PowerPoint (PPTX) のスライドに埋め込まれたグラフの画質が向上します。バージョン 8.2 SP01M では、PNG 出力フォーマットを使用して、イメージおよびテキストの画質を向上させるとともに、透明度をサポートすることができます。

この出力フォーマットは、背景色が定義されたテンプレートを使用する場合や、グラフに他のコンポーネントや描画オブジェクトを重ね合わせる場合など、多くの重要なシナリオで役立ちます。

PNG イメージフォーマットでの PPTX グラフの表示

```
SET PPTXGRAPHTYPE={PNG|PNG_NOSCALE|JPEG}
```

説明

PNG

画質を大幅に改善するために、PNG イメージを元のサイズの 2 倍に拡大します。絶対ポイントサイズのテキストなど (埋め込みスケール見出しを含む)、拡大縮小できない項目がグラフに含まれている場合、このオプションを選択すると問題が発生することがあります。また、ビットマップよりサイズが大きくなるため、出力サイズも増加します。グラフ内のテキストは、従来の JPEG フォーマットより、かなり鮮明になります。

PNG では、解像度の増加に伴いフォントサイズが内部的に拡大された場合でも、グラフ内のフォントサイズは保持されます。スタイルシート (*GRAPH_SCRIPT) で設定された絶対フォントサイズは、仮想座標で表現されるサイズに変換されて (グラフのサイズに相対的なサイズ)、仮想座標で埋め込み見出しおよび脚注のフォントサイズが生成されます。

PNG_NOSCALE

PNG で描画されますが、拡大はされません。この設定では、JPEG より画質が若干向上します。JPEG を PNG_NOSCALE に変更すると、グラフがより鮮明になりますが、テキストには若干の効果があるのみです。

JPEG

従来のフォーマットです。これがデフォルト値です。

透明度を使用した PNG グラフの表示

透明度を使用することにより、さまざまなコンポーネントや描画オブジェクトをスライド上に同時に配置する際に、これらの配置を制御しやすくなります。次のレポートには、テキストボックス、レポート、グラフが含まれています。グラフの背景により他のオブジェクトのコンテンツが隠れることなく、各コンポーネントが同一ページ上に表示されています。

```
SET PPTXGRAPHTYPE=PNG
```

```
COMPOUND LAYOUT PCHOLD FORMAT PPTX
UNITS=IN, $
SECTION=section1, LAYOUT=ON, MERGE=OFF, ORIENTATION=LANDSCAPE,
PAGESIZE=PPT Slide, $
PAGELAYOUT=1, NAME='Page layout 1', text='Page layout 1',
BOTTOMMARGIN=0.5, TOPMARGIN=0.5, $
COMPONENT='report1', TEXT='report1', POSITION=(5.088 1.375),
DIMENSION=(2.260 2.500), $
COMPONENT='chart2', TEXT='chart2', POSITION=(0.815 1.351),
DIMENSION=(5.104 2.917), COMPONENT-TYPE=GRAPH,  $
OBJECT=STRING, NAME='text1', TEXT='<left>PNG charts can be defined with
transparency to allow the background to show through and allow for
overlapping components to optimize the use of space on the slide.
</left>',
POSITION=(0.500 0.979), MARKUP=ON, WRAP=ON, DIMENSION=(4.635 0.729),
font='TREBUCHET MS', color=RGB(0 0 0), size=10, $
END
```

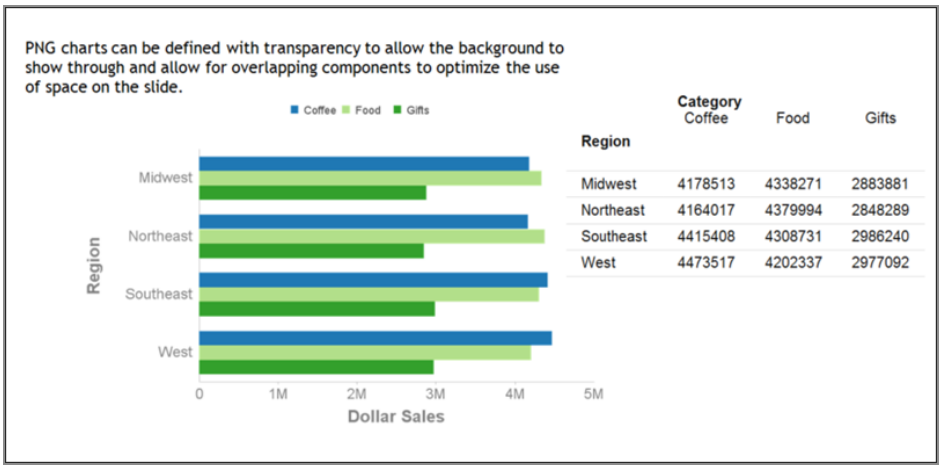
```
SET COMPONENT='report1'
TABLE FILE GGSALES
SUM
   GGSALES.SALES01.DOLLARS
BY GGSALES.SALES01.REGION
ACROSS LOWEST GGSALES.SALES01.CATEGORY
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/ibi_themes/Warm.sty,$
$
ENDSTYLE
END
```

```
SET COMPONENT='chart2'
GRAPH FILE GGSALES
SUM GGSALES.SALES01.DOLLARS
BY GGSALES.SALES01.CATEGORY
ACROSS GGSALES.SALES01.REGION
ON GRAPH PCHOLD FORMAT HTML
ON GRAPH SET VZERO OFF
ON GRAPH SET HTMLENCODE ON
ON GRAPH SET GRAPHDEFAULT OFF
ON GRAPH SET EMBEDHEADING ON
ON GRAPH SET GRWIDTH 1
ON GRAPH SET UNITS 'PIXELS'
```

```
ON GRAPH SET HAXIS 770.0
ON GRAPH SET VAXIS 405.0
ON GRAPH SET GRMERGE ADVANCED
ON GRAPH SET GRMULTIGRAPH 0
ON GRAPH SET GRLEGEND 1
ON GRAPH SET GRXAXIS 1
ON GRAPH SET LOOKGRAPH HBAR
ON GRAPH SET STYLE *
```

```
*GRAPH_SCRIPT
setPieDepth(0);
setPieTilt(0);
setDepthRadius(0);
setPlace(true);
setCurveFitEquationDisplay(false);
*END
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/ibi_themes/Warm.sty,$
TYPE=REPORT, TITLETEXT='WebFOCUS Report', $
*GRAPH_SCRIPT
setFillColor(getChartBackground(),new Color(255,255,255,0));
setLegendPosition(4);
*GRAPH_JS_FINAL
"blaProperties": {
    "orientation": "horizontal"
},
"agnosticSettings": {
    "chartTypeFullName":"Bar_Clustered_Horizontal"
}
*END
ENDSTYLE
END
-RUN

COMPOUND END
```

出力結果は次のとおりです。



Region	Category Coffee	Food	Gifts
Midwest	4178513	4338271	2883881
Northeast	4164017	4379994	2848289
Southeast	4415408	4308731	2986240
West	4473517	4202337	2977092

透明度を使用した PNG イメージの指定テンプレートでの表示

次の複合レポートでは、背景色とパターンを含む指定テンプレートのスライド上に、透明度を定義したグラフが配置されています。

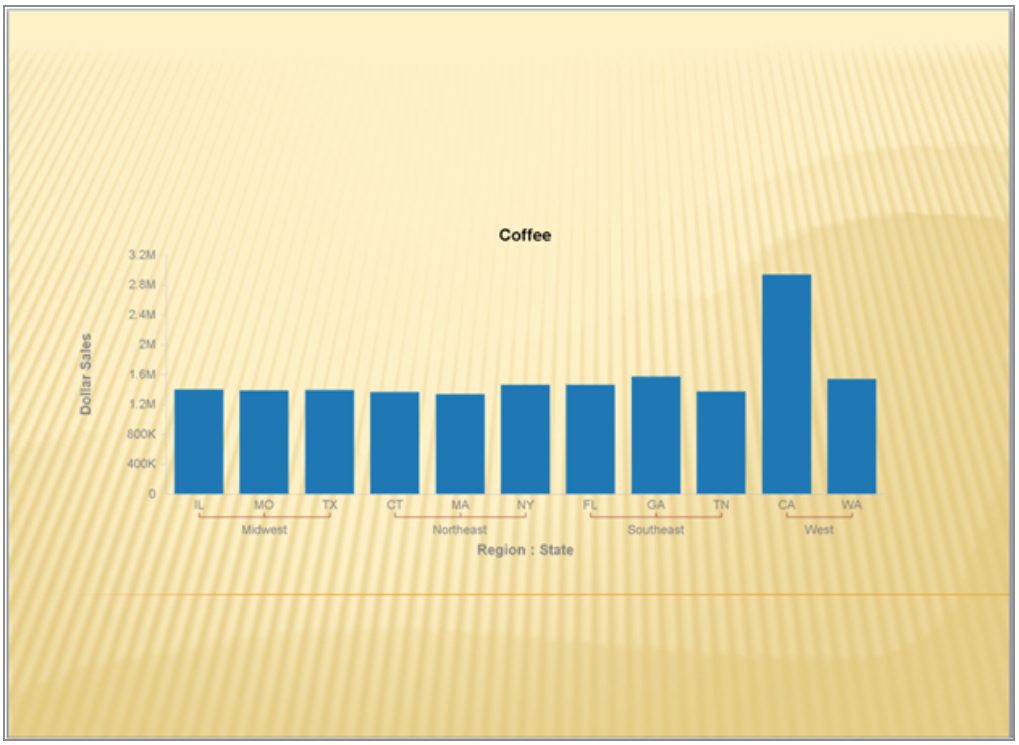
```
SET PPTXGRAPHTYPE=PNG

COMPOUND LAYOUT PCHOLD FORMAT PPTX
UNITS=IN, $
SECTION=section1, LAYOUT=ON, MERGE=OFF, ORIENTATION=LANDSCAPE,
PAGESIZE=Letter, $
PAGELAYOUT=1, NAME='Page layout 1', text='Page layout 1', $
COMPONENT='ppt_template', $
COMPONENT='chart1', TEXT='chart1', POSITION=(0.500 2.10),
DIMENSION=(9.336 3.437),
COMPONENT-TYPE=GRAPH, ARREPORTSIZE=DIMENSION, $
END

SET COMPONENT='ppt_template'
TABLE FILE SYSCOLUM
SUM TBNAME NOPRINT
IF READLIMIT EQ 1
ON TABLE PCHOLD FORMAT PPTX TEMPLATE 'golden.potx' SLIDENUMBER 1
END
```

```
SET COMPONENT='chart1'
GRAPH FILE GGSALES
```

```
SUM GGSALES.SALES01.DOLLARS
BY GGSALES.SALES01.CATEGORY
BY GGSALES.SALES01.REGION
BY GGSALES.SALES01.ST
ON GRAPH PCHOLD FORMAT HTML
ON GRAPH SET EMBEDHEADING ON
ON GRAPH SET GRWIDTH 1
ON GRAPH SET UNITS PIXELS
ON GRAPH SET HAXIS 770.0
ON GRAPH SET VAXIS 405.0
ON GRAPH SET GRMERGE ADVANCED
ON GRAPH SET GRMULTIGRAPH 1
ON GRAPH SET GRLEGEND 0
ON GRAPH SET GRXAXIS 2
ON GRAPH SET LOOKGRAPH VBAR
ON GRAPH SET STYLE *
```

```
*GRAPH_SCRIPT
setPieDepth(0);
setPieTilt(0);
setDepthRadius(0);
setCurveFitEquationDisplay(false);
setPlace(true);
*END
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/ibi_themes/Warm.sty,$
TYPE=REPORT, TITLETEXT='WebFOCUS Report', $
*GRAPH_SCRIPT
setFillType(getChartBackground(),2);
setGradientNumPins(getChartBackground(),2);
setFillColor(getChartBackground(),new Color(255,255,255,0));
setFillType(getChartBackground(),1);
*END
ENDSTYLE
END
-RUN
COMPOUND END
```

出力結果は次のとおりです。



Microsoft PowerPoint からのドリルダウン

次の 2 つのタイプのドリルダウンがサポートされます。

- WebFOCUS コンテンツ
- 外部 URL

WebFOCUS リポジトリまたはコンテンツ環境では、Microsoft PowerPoint がブラウザではなく PowerPoint アプリケーションウィンドウで開いている場合、PPTX レポート内のドリルダウンハイパーリンクが機能しません。この場合、現在のセキュリティコンテキストや保持されているセッション Cookie が失われ、その結果、ユーザ認可が変更されます。Microsoft PowerPoint アプリケーションで表示されたレポートのドリルダウンが正常に機能するためには、次の 3 つのセキュリティモデルからいずれかを構成することをお勧めします。

Remember Me セキュリティモデル

Remember Me (ユーザを記憶する) セキュリティモデルは、WebFOCUS が、デフォルト期間で 14 日間、信頼済みのログイン Cookie をローカルのワークステーションに保存することが

できるユーザ認証方法です。ただし、このログイン Cookie には、ユーザパスワードは格納されません。ログインページで [ユーザを記憶する] 機能を有効にします。エンドユーザが [ユーザを記憶する] 機能を使用した場合、永続 Cookie が使用されます。

パブリックアクセス

パブリックアクセスは、組織内のすべての人または一般向けに使用可能で、認証を必要としないアクセスとして便利です。WebFOCUS のセキュリティ設定で、レポートへのドリルダウンに必要なアクセス権限を Public ユーザに許可します。

Windows 統合認証

Windows 統合認証 (IWA) は、ブラウザの構成により使用できます。SSO を IIS/Tomcat Windows 統合認証と併用します。再ネゴシエーションが自動的に発生し、PowerPoint フォーマットレポートが正しく表示されます。

これらの認証モデルについての詳細は、『ibi™ WebFOCUS® セキュリティ管理ガイド』を参照してください。

外部 URL へのドリルダウン

次のリクエストは、総合計行タグに外部 URL への参照を配置します。

```
TABLE FILE GGSALES
SUM
  GGSALES.SALES01.BUDDOLLARS/D12CM
  GGSALES.SALES01.DOLLARS/D12CM
BY GGSALES.SALES01.REGION
BY GGSALES.SALES01.CATEGORY
BY GGSALES.SALES01.PRODUCT
HEADING
"REVENUE BY REGION "
ON GGSALES.SALES01.REGION SUBTOTAL AS '*TOTAL'
WHERE GGSALES.SALES01.CATEGORY NE 'Gifts';
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE COLUMN-TOTAL AS 'TOTAL'
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
PAGESIZE='PPT Slide',
  ORIENTATION=LANDSCAPE,
$
```

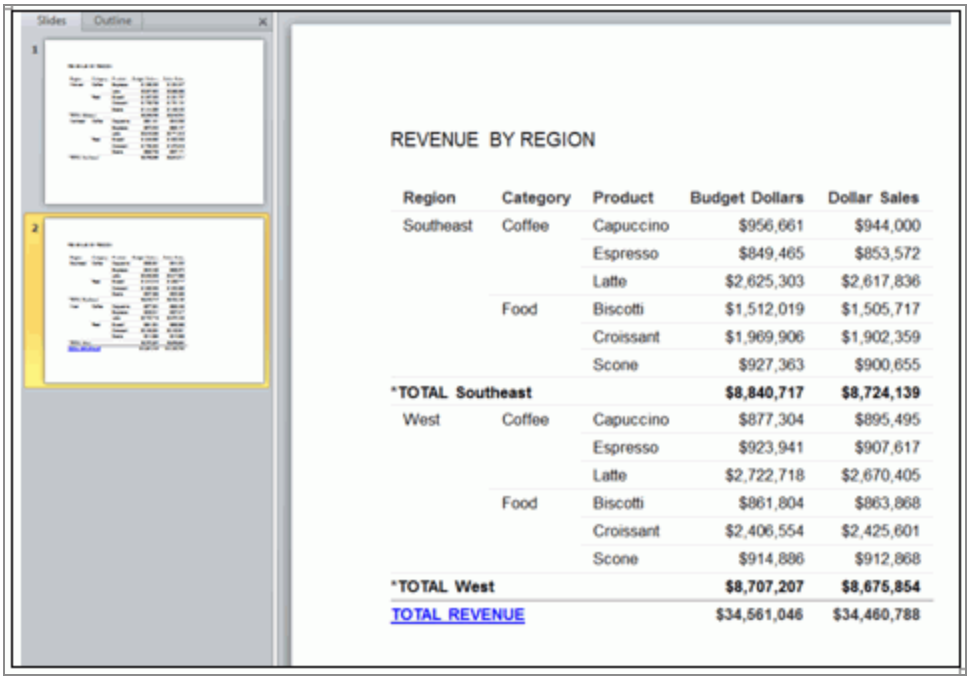
```
TYPE=REPORT,
  OBJECT=STATUS-AREA,
```

```
  JUSTIFY=LEFT,
  PAGE-LOCATION=BOTTOM,
$
TYPE=REPORT,
  GRID=OFF,
  FONT='ARIAL',
  SIZE=12,
  STYLE=NORMAL,
  SQUEEZE=ON,
  TOPGAP=0.05,
  BOTTOMGAP=0.05,
  BORDER-COLOR=RGB(219 219 219),
  TITLELINE=SKIP,
  TOPMARGIN=1,
  LEFTMARGIN=1,
$
TYPE=TITLE,
  COLOR=RGB(51 51 51),
  STYLE=-UNDERLINE +BOLD,
$
TYPE=DATA,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=HEADING,
  JUSTIFY=LEFT,
  SIZE=14,
$
TYPE=SUBTOTAL,
  STYLE=BOLD,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=GRANDTOTAL,
  OBJECT=TAG,
  URL=http://www.tibco.com,
$
```

```
TYPE=REPORT,
  OBJECT=STATUS-AREA,
  JUSTIFY=LEFT,
  PAGE-LOCATION=BOTTOM,
$
TYPE=GRANDTOTAL,
  COLOR=RGB(51 51 51),
  STYLE=BOLD,
```

```
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(102 102 102),
$
ENDSTYLE
END
```

出力結果は次のとおりです。



テンプレートを使用した PowerPoint PPTX プレゼンテーション

PowerPoint テンプレートに基づいて PPTX レポート出力を生成することができます。この機能を使用して、複数のスライドで構成されたプレゼンテーションに WebFOCUS レポートを統合することができます。任意の PowerPoint テンプレートを使用して、WebFOCUS レポートが挿入された新しいプレゼンテーションを生成することができます。

次の PowerPoint ファイルタイプは、PPTX プレゼンテーションを生成するためのテンプレートファイルとして使用することができます。

テンプレートファイルタイプ	生成されるプレゼンテーション出力
テンプレート (.potx)	プレゼンテーション (.pptx)
マクロ有効テンプレート (.potm)	マクロ有効プレゼンテーション (.pptm)
プレゼンテーション (.pptx)	プレゼンテーション (.pptx)
マクロ有効プレゼンテーション (.pptm)	マクロ有効プレゼンテーション (.pptm)

注意：マクロ有効テンプレートでのアクティブコンテンツの使用についての詳細は、Microsoft の Web ページ (https://support.office.com/en-nz/article/Enable-or-disable-macros-in-Office-documents-7b4fdd2e-174f-47e2-9611-9efe4f860b12) を参照してください。

標準 PowerPoint テンプレート (POTX) の使用

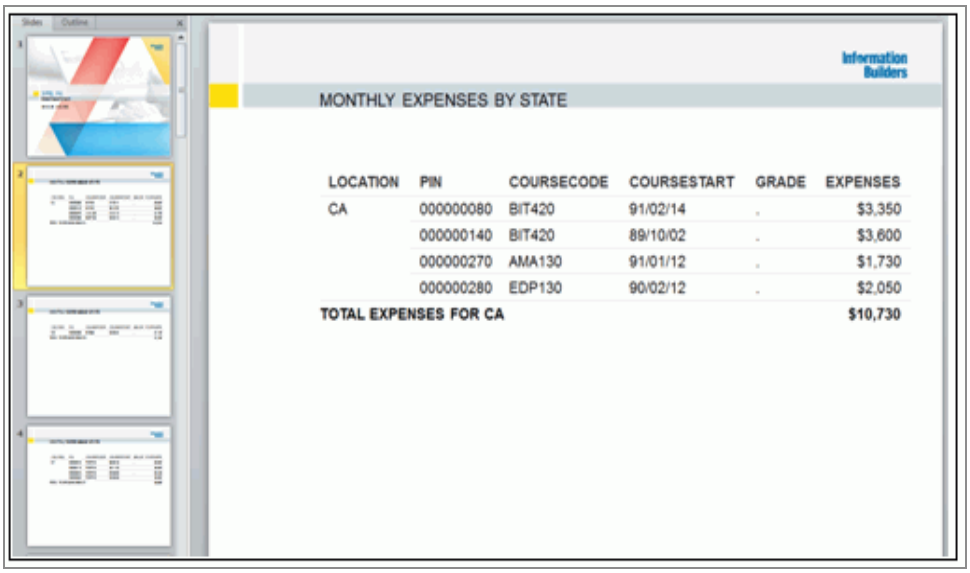
次のリクエストでは、複数のスライドを使用するレポートを生成します。指定したスライドは、必要な数のスライドに置き換えられて、レポート出力に表示されます。

```
TABLE FILE TRAINING
SUM
TRAINING.TRAINING.EXPENSES/D12CM
BY TRAINING.TRAINING.LOCATION
BY TRAINING.TRAINING.PIN
BY LOWEST TRAINING.TRAINING.COURSECODE
BY TRAINING.TRAINING.COURSESTART
BY TRAINING.TRAINING.GRADE
ON TRAINING.TRAINING.LOCATION SUBTOTAL AS 'TOTAL EXPENSES FOR'
ON TRAINING.TRAINING.LOCATION PAGE-BREAK
HEADING
"MONTHLY EXPENSES BY STATE"
" "
ON TABLE SET ASNAMES ON
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PPTX TEMPLATE 'ibi_template.potx' SLIDENUMBER 2
ON TABLE SET STYLE *
PAGESIZE='PPT Slide',
  ORIENTATION=LANDSCAPE,
```

```
$
TYPE=REPORT,
  OBJECT=STATUS-AREA,
  JUSTIFY=LEFT,
  PAGE-LOCATION=BOTTOM,
$
TYPE=REPORT,
  GRID=OFF,
  FONT='ARIAL',
  SIZE=14,
  STYLE=NORMAL,
  SQUEEZE=ON,
  TOPGAP=0.05,
  BOTTOMGAP=0.05,
  BORDER-COLOR=RGB(219 219 219),
  TITLELINE=SKIP,
  TOPMARGIN=.1,
  LEFTMARGIN=1.5,
$
```

```
TYPE=TITLE,
  COLOR=RGB(51 51 51),
  STYLE=-UNDERLINE +BOLD,
$
TYPE=DATA,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=HEADING,
  JUSTIFY=LEFT,
  SIZE=16,
$
TYPE=SUBTOTAL,
  STYLE=BOLD,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
ENDSTYLE
END
```

出力結果は次のとおりです。



複数リクエストによる指定スライドのテンプレートへの挿入

次のリクエストは、複数のスライドに異なるコンポーネントを挿入する方法を示しています。

```
-* Replace Slide #2
TABLE FILE GGSALES
HEADING
"FIRST SLIDE"
SUM
DOLLARS/D12CM UNITS
BY REGION AS 'My Field'
BY CATEGORY
ON TABLE COLUMN-TOTAL
ON TABLE HOLD AS SLIDE_A FORMAT PPTX TEMPLATE 'ibi_template.potx' SLIDENUMBER
2
ON TABLE SET STYLE *
TYPE=REPORT,
  OBJECT=STATUS-AREA,
  JUSTIFY=LEFT,
  PAGE-LOCATION=BOTTOM,
$
TYPE=REPORT,
  GRID=OFF,
  FONT='ARIAL',
```

```
  SIZE=14,
  STYLE=NORMAL,
  SQUEEZE=ON,
  TOPGAP=0.05,
  BOTTOMGAP=0.05,
  BORDER-COLOR=RGB(219 219 219),
  TITLELINE=SKIP,
  ORIENTATION = LANDSCAPE,
  TOPMARGIN=.1,
  LEFTMARGIN=1.5,
$
TYPE=TITLE,
  COLOR=RGB(51 51 51),
  STYLE=-UNDERLINE +BOLD,
$
```

```
TYPE=DATA,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=HEADING,
  JUSTIFY=LEFT,
  SIZE=16,
$
TYPE=DATA,
  COLUMN=DOLLARS,
  COLOR=BLUE,
$
TYPE=REPORT,
  COLUMN=REGION,
  COLOR=RED,
$
TYPE=REPORT,
  COLUMN=CATEGORY,
  COLOR=GREEN,
$
TYPE=GRANDTOTAL,
  COLOR=RGB(51 51 51),
  STYLE=BOLD,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(102 102 102),
$
END
-* Replace Slide #3
TABLE FILE GGSALES HEADING
"SECOND SLIDE"
```

```
SUM DOLLARS/D12CM UNITS
BY REGION AS 'My Field'
BY CATEGORY
ON TABLE COLUMN-TOTAL
ON TABLE HOLD AS SLIDE_1 FORMAT PPTX TEMPLATE 'slide_a.pptx' SLIDENUMBER 3
ON TABLE COLUMN-TOTAL
ON TABLE SET STYLE *
$
TYPE=REPORT,
  OBJECT=STATUS-AREA,
  JUSTIFY=LEFT,
  PAGE-LOCATION=BOTTOM,
$
```

```
TYPE=REPORT,
  GRID=OFF,
  FONT='ARIAL',
  SIZE=14,
  STYLE=NORMAL,
  SQUEEZE=ON,
  TOPGAP=0.05,
  BOTTOMGAP=0.05,
  BORDER-COLOR=RGB(219 219 219),
  TITLELINE=SKIP,
  ORIENTATION = LANDSCAPE,
  TOPMARGIN=.1,
  LEFTMARGIN=1.5,
$
TYPE=TITLE,
  COLOR=RGB(51 51 51),
  STYLE=-UNDERLINE +BOLD,
$
TYPE=DATA,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=HEADING,
  JUSTIFY=LEFT,
  SIZE=16,
$
TYPE=DATA,
  COLUMN=DOLLARS,
  COLOR=BLUE,
$
TYPE=REPORT,
  COLUMN=REGION,
```

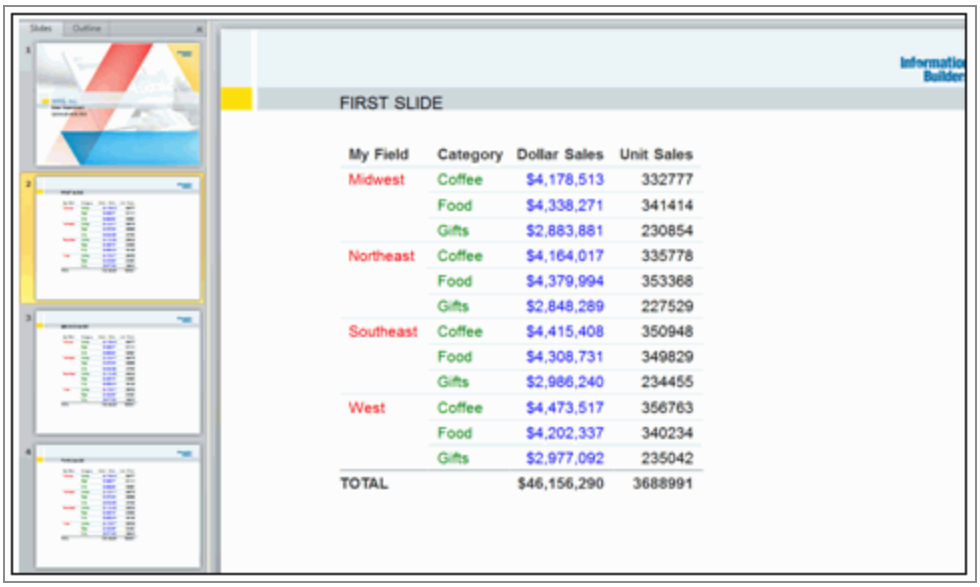
```
  COLOR=RED,
$
TYPE=REPORT,
  COLUMN=CATEGORY,
  COLOR=GREEN,
$
TYPE=GRANDTOTAL,
  COLOR=RGB(51 51 51),
  STYLE=BOLD,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(102 102 102),
$
END
```

```
-* Replace Slide #4
TABLE FILE GGSALES
HEADING
"THIRD SLIDE"
SUM
DOLLARS/D12CM
UNITS
BY REGION AS 'My Field'
BY CATEGORY
ON TABLE COLUMN-TOTAL
ON TABLE PCHOLD AS THRID_SLIDE FORMAT PPTX TEMPLATE 'SLIDE_1.pptx'
SLIDENUMBER 4
ON TABLE SET STYLE *
TYPE=REPORT,
  OBJECT=STATUS-AREA,
  JUSTIFY=LEFT,
  PAGE-LOCATION=BOTTOM,
$
TYPE=REPORT,
  GRID=OFF,
  FONT='ARIAL',
  SIZE=14,
  STYLE=NORMAL,
  SQUEEZE=ON,
  TOPGAP=0.05,
  BOTTOMGAP=0.05,
  BORDER-COLOR=RGB(219 219 219),
  TITLELINE=SKIP,
  ORIENTATION = LANDSCAPE,
  TOPMARGIN=.1,
  LEFTMARGIN=1.5,
$
```

```
TYPE=TITLE,
  COLOR=RGB(51 51 51),
  STYLE=-UNDERLINE +BOLD,
$
TYPE=DATA,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=HEADING,
  JUSTIFY=LEFT,
  SIZE=16,
```

```
$
TYPE=DATA,
  COLUMN=DOLLARS,
  COLOR=BLUE,
$
TYPE=REPORT,
  COLUMN=REGION,
  COLOR=RED,
$
TYPE=REPORT,
  COLUMN=CATEGORY,
  COLOR=GREEN,
$
TYPE=GRANDTOTAL,
  COLOR=RGB(51 51 51),
  STYLE=BOLD,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(102 102 102),
$
END
```

出力結果は次のとおりです。



PowerPoint PPTX 複合ドキュメント構文

PowerPoint 複合ドキュメントは、複数のスライドを含むプレゼンテーションを生成します。PowerPoint 複合ドキュメントのコンポーネントには、標準のテーブルおよびグラフを含めることができます。

複合ドキュメントの作成

次のリクエストは、スライドデッキを作成し、選択した情報を標準のテーブルおよびグラフで表します。

```
SET HTMLARCHIVE=ON
COMPOUND LAYOUT PCHOLD FORMAT PPTX
UNITS=IN, $
SECTION=section1, LAYOUT=ON, MERGE=OFF,
ORIENTATION=LANDSCAPE, PAGESIZE=PPT Slide, SHOW_GLOBALFILTER=OFF, $
PAGELAYOUT=1, NAME='Page layout 1', text='Page layout 1',
BOTTOMMARGIN=0.2, TOPMARGIN=0.5, LEFTMARGIN=2.0, $
COMPONENT='chart1', TEXT='chart1', POSITION=(0.707 0.520),
DIMENSION=(8.750 2.917), COMPONENT-TYPE=GRAPH, $
COMPONENT='report1', TEXT='report1', POSITION=(0.500 3.542),
DIMENSION=(9.271 3.646), $
END
```

```
SET COMPONENT='chart1'
ENGINE INT CACHE SET ON
-DEFAULTH &WF_STYLE_UNITS='PIXELS';
-DEFAULTH &WF_STYLE_HEIGHT='405.0';
-DEFAULTH &WF_STYLE_WIDTH='770.0';
-DEFAULTH &WF_TITLE='WebFOCUS Report';
GRAPH FILE ibisamp/ggsales
SUM GGSALES.SALES01.DOLLARS
BY GGSALES.SALES01.REGION
BY TOTAL HIGHEST GGSALES.SALES01.DOLLARS NOPRINT
BY GGSALES.SALES01.ST
ON GRAPH PCHOLD FORMAT PPTX
ON GRAPH SET HTMLENCODE ON
ON GRAPH SET GRAPHDEFAULT OFF
ON GRAPH SET ARGRAPHENGIN JSCHART
ON GRAPH SET EMBEDHEADING ON
ON GRAPH SET VZERO OFF
ON GRAPH SET GRWIDTH 1
```

```
ON GRAPH SET UNITS &WF_STYLE_UNITS
ON GRAPH SET HAXIS &WF_STYLE_WIDTH
ON GRAPH SET VAXIS &WF_STYLE_HEIGHT
ON GRAPH SET GRMERGE ADVANCED
ON GRAPH SET GRMULTIGRAPH 1
ON GRAPH SET GRLEGEND 0
ON GRAPH SET GRXAXIS 2
ON GRAPH SET LOOKGRAPH VBAR
ON GRAPH SET STYLE *
*GRAPH_SCRIPT
setPieDepth(0);
setPieTilt(0);
setDepthRadius(0);
setCurveFitEquationDisplay(false);
setPlace(true);
setPieFeelerTextDisplay(1);
setUseSeriesShapes(true);
setMarkerSizeDefault(50);
*END
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/javaassist/intl/EN/combine_
templates/warm.sty,$
TYPE=REPORT, TITLETEXT=&WF_TITLE.QUOTEDSTRING, $
*GRAPH_SCRIPT
setReportParsingErrors(false);
setSelectionEnableMove(false);
setFillType(getSeries(0),2);
setGradientPinLeftColor0(getSeries(0),new Color(0,127,192));
```

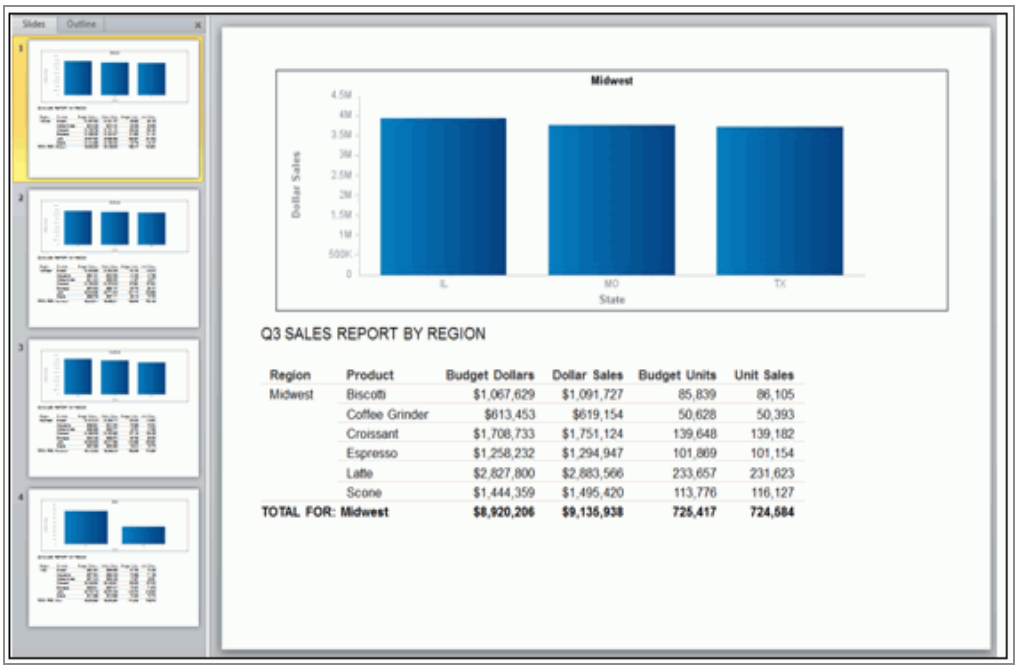
```
setGradientPinRightColor0(getSeries(0),new Color(0,127,192));
setGradientPinLeftColor2(getSeries(0),new Color(0,127,192));
setGradientPinRightColor2(getSeries(0),new Color(0,127,192));
setGradientPinLeftColor1(getSeries(0),new Color(0,64,128));
setGradientPinRightColor1(getSeries(0),new Color(0,64,128));
setGradientPinPosition0(getSeries(0),0.0);
setGradientPinPosition1(getSeries(0),1.0);
setPieTilt(0);
setPieDepth(0);
setDepthRadius(0);
setDepthAngle(0);
setFillType(getSeries(7),2);
setGradientPinPosition0(getSeries(7),0.0);
setGradientPinPosition1(getSeries(7),1.0);
setFillType(getSeries(9),2);
setGradientDirection(getSeries(9),16);
setGradientPinPosition0(getSeries(9),0.0);
setGradientPinPosition1(getSeries(9),1.0);
*END
ENDSTYLE
END
-RUN
```

```
SET COMPONENT='report1'
TABLE FILE IBISAMP/GGSALES
SUM
  GGSALES.SALES01.BUDDOLLARS/D12CM
  GGSALES.SALES01.DOLLARS/D12CM
  GGSALES.SALES01.BUDUNITS/D12C
  GGSALES.SALES01.UNITS/D12C
BY GGSALES.SALES01.REGION
BY GGSALES.SALES01.PRODUCT
ON GGSALES.SALES01.REGION SUBTOTAL AS 'TOTAL FOR:'
ON GGSALES.SALES01.REGION PAGE-BREAK
HEADING
"Q3 SALES REPORT BY REGION"
WHERE GGSALES.SALES01.PRODUCT NE 'Coffee Pot' OR 'Mug' OR 'Thermos';
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE SET ASNAMES ON
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
  PAGESIZE='PPT Slide',
  LEFTMARGIN=1.000000,
  TOPMARGIN=0.500000,
  BOTTOMMARGIN=0.000000,
```

```
  SQUEEZE=ON,
  ORIENTATION=LANDSCAPE,
$
TYPE=REPORT,
  BORDER-TOP-COLOR=RGB(219 219 219),
  BORDER-BOTTOM-COLOR=RGB(219 219 219),
  BORDER-LEFT-COLOR=RGB(219 219 219),
  BORDER-RIGHT-COLOR=RGB(219 219 219),
  FONT='ARIAL',
  SIZE=12,
    TITLELINE=SKIP,
  STYLE=NORMAL,
  TOPGAP=0.041667,
$
TYPE=DATA,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
```

```
TYPE=TITLE,
  COLOR=RGB(51 51 51),
  STYLE=-UNDERLINE+BOLD,
$
TYPE=HEADING,
  SIZE=14,
  JUSTIFY=LEFT,
$
TYPE=SUBTOTAL,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
  STYLE=BOLD,
$
TYPE=REPORT,
  OBJECT=STATUS-AREA,
  JUSTIFY=LEFT,
  PAGE-LOCATION=BOTTOM,
$
ENDSTYLE
END
COMPOUND END
```

生成された複合ドキュメントの出力は次のとおりです。



統合された複合レイアウトレポート

統合された複合レイアウトレポートでは、共通のソートフィールドを含むすべてのレポートおよびグラフが、個々のページレイアウトに分割 (バースト) されます。共通のソートフィールドの値ごとにページが生成され、ソート値に対して取得されたデータが各ページのすべてのコンポーネントに表示されます。統合された複合レイアウトレポートを作成するには、複合レイアウトレポートの SECTION 宣言で MERGE=ON を指定します。

統合された複合レイアウトレポートでは、特定のソートフィールド値のデータがコンポーネントのいずれかに含まれている場合、一部のコンポーネントのデータが欠落している場合でもそのソートフィールド値のページが生成されます。

レポートの縦方向にはクエリで生成されたデータ行のすべてが常に表示されますが、レポートの幅は、定義されたコンポーネントコンテナのサイズにより制限されます。つまり、複合レポートではマルチパネル表示はサポートされません。ただし、複合レポート以外の PPTX ドキュメントではサポートされます。

複合レポート以外の PPTX ドキュメントでは、レポートデータの幅が定義済みページサイズより長い場合、パネル (横方向のオーバーフローページ) が自動的に生成されます。

従来の複合レポートの構文では、コンポーネントレポートのいずれかが定義済みページ幅に収まらない場合、レポートの実行が中断され、マルチパネル表示がサポートされないことを示すエラーメッセージが表示されます。

複合レイアウトレポートの構文では、コンポーネントが定義済みコンテナサイズ幅に収まらない場合、コンテナ内でレポートのコンテンツが折り返されます。コンテナのサイズは、複合レポートの構文で、コンポーネントの POSITION と DIMENSIONS パラメータの両方を組み合わせて定義します。

統合された複合レイアウトレポートの生成

次のリクエストは、統合された複合レイアウトレポートを生成します。複合レポートは、地域の値で統合されます。つまり、主キーである地域 (Region) の値ごとに個別のスライドが生成されます。

```
SET HTMLARCHIVE=ON
*-HOLD_SOURCE
COMPOUND LAYOUT PCHOLD FORMAT PPTX
UNITS=IN, $
SECTION=section1, LAYOUT=ON, MERGE=ON, ORIENTATION=LANDSCAPE,
PAGESIZE=PPT Slide, SHOW_GLOBALFILTER=OFF,$
PAGELAYOUT=1, NAME='Page layout 1', text='Page layout 1', BOTTOMMARGIN=0.5,
TOPMARGIN=0.5, $
COMPONENT='report1', TEXT='report1', POSITION=(0.500 0.625),
DIMENSION=(* *), $
COMPONENT='report2', TEXT='report2', POSITION=(0.712 0.771),
DIMENSION=(* *), $
COMPONENT='report3', TEXT='report3', POSITION=(5.702 0.759),
DIMENSION=(* *), $
END
SET COMPONENT='report1'
TABLE FILE WF_RETAIL
BY WF_RETAIL.WF_RETAIL_GEOGRAPHY_CUSTOMER.BUSINESS_REGION NOPRINT
HEADING
"PROFIT REPORTS FOR <WF_RETAIL.WF_RETAIL_GEOGRAPHY_CUSTOMER.BUSINESS_REGION
<+15>
DISCOUNTS APPLIED: <WF_RETAIL.WF_RETAIL_GEOGRAPHY_CUSTOMER.BUSINESS_REGION"
" "
ON TABLE SET ASNAMES ON
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
```

```
TYPE=REPORT,
  GRID=OFF,
  FONT='ARIAL',
  SIZE=9,
```

```
  STYLE=NORMAL,
  SQUEEZE=ON,
  TOPGAP=0.05,
  BOTTOMGAP=0.05,
  PAGECOLOR='WHITE',
  BORDER-COLOR=RGB(219 219 219),
  TITLELINE=SKIP,
    TOPMARGIN=.5,
    LEFTMARGIN=.15,
$
TYPE=HEADING,
  JUSTIFY=LEFT,
  SIZE=14,
$
ENDSTYLE
END
SET COMPONENT='report2'
TABLE FILE WF_RETAIL
SUM
WF_RETAIL.WF_RETAIL_SALES.COGS_US
WF_RETAIL.WF_RETAIL_SALES.GROSS_PROFIT_US
BY WF_RETAIL.WF_RETAIL_GEOGRAPHY_CUSTOMER.BUSINESS_REGION NOPRINT
BY WF_RETAIL.WF_RETAIL_PRODUCT.PRODUCT_CATEGORY
ON WF_RETAIL.WF_RETAIL_GEOGRAPHY_CUSTOMER.BUSINESS_REGION SUBTOTAL AS 'TOTAL
FOR'
ON TABLE SET ASNAMES ON
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
PAGESIZE='PPT Slide',
  ORIENTATION=LANDSCAPE,
$
TYPE=REPORT,
  OBJECT=STATUS-AREA,
  JUSTIFY=LEFT,
  PAGE-LOCATION=BOTTOM,
$
```

```
TYPE=REPORT,
  GRID=OFF,
  FONT='ARIAL',
  SIZE=14,
  STYLE=NORMAL,
  SQUEEZE=ON,
  TOPGAP=0.05,
  BOTTOMGAP=0.05,
```

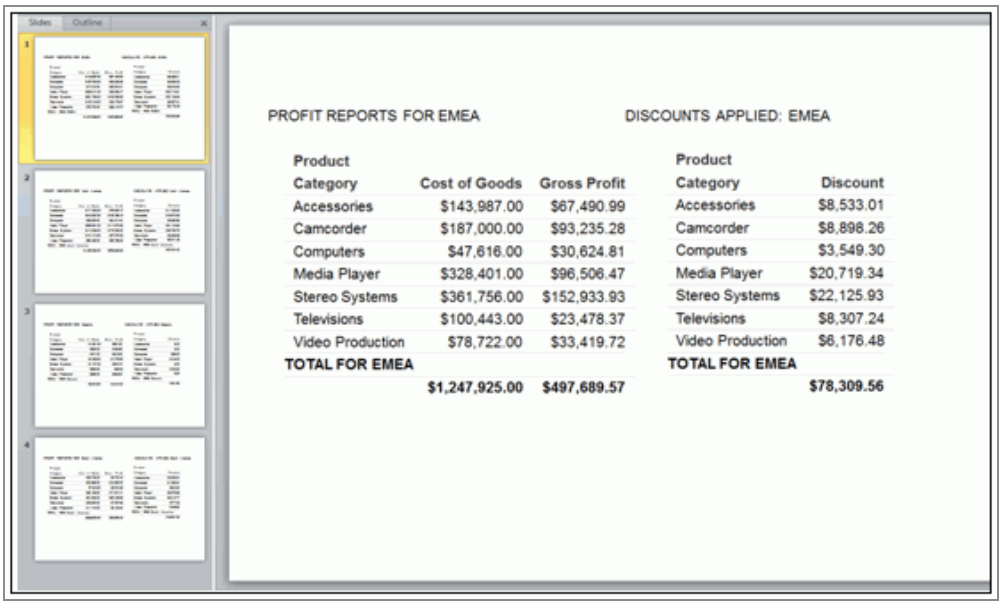
```
  PAGECOLOR='WHITE',
  BORDER-COLOR=RGB(219 219 219),
  TITLELINE=SKIP,
    TOPMARGIN=.5,
    LEFTMARGIN=.15,
$
TYPE=DATA,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=TITLE,
  COLOR=RGB(51 51 51),
  STYLE=-UNDERLINE +BOLD,
$
TYPE=DATA,
  COLUMN=ROWTOTAL(*),
  STYLE=BOLD,
$
TYPE=TITLE,
  COLUMN=ROWTOTAL(*),
  COLOR=RGB(51 51 51),
  STYLE=-UNDERLINE +BOLD,
$
TYPE=SUBTOTAL,
  STYLE=BOLD,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
ENDSTYLE
END
SET COMPONENT='report3'
TABLE FILE WF_RETAIL
SUM
WF_RETAIL.WF_RETAIL_SALES.DISCOUNT_US
BY WF_RETAIL.WF_RETAIL_GEOGRAPHY_CUSTOMER.BUSINESS_REGION NOPRINT
BY WF_RETAIL.WF_RETAIL_PRODUCT.PRODUCT_CATEGORY
ON WF_RETAIL.WF_RETAIL_GEOGRAPHY_CUSTOMER.BUSINESS_REGION SUBTOTAL
AS 'TOTAL FOR'
```

```
ON TABLE SET ASNAMES ON
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
PAGESIZE='PPT Slide',
  ORIENTATION=LANDSCAPE,
$
```

```
TYPE=REPORT,
  OBJECT=STATUS-AREA,
  JUSTIFY=LEFT,
  PAGE-LOCATION=BOTTOM,
$
TYPE=REPORT,
  GRID=OFF,
  FONT='ARIAL',
  SIZE=14,
  STYLE=NORMAL,
  SQUEEZE=ON,
  TOPGAP=0.05,
  BOTTOMGAP=0.05,
  PAGECOLOR='WHITE',
  BORDER-COLOR=RGB(219 219 219),
  TITLELINE=SKIP,
$
TYPE=DATA,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=TITLE,
  COLOR=RGB(51 51 51),
  STYLE=-UNDERLINE +BOLD,
$
TYPE=DATA,
  COLUMN=ROWTOTAL(*),
  STYLE=BOLD,
$
```

```
TYPE=TITLE,
  COLUMN=ROWTOTAL(*),
  COLOR=RGB(51 51 51),
  STYLE=-UNDERLINE +BOLD,
$
TYPE=SUBTOTAL,
  STYLE=BOLD,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
ENDSTYLE
END
COMPOUND END
```

出力結果は次のとおりです。



複合レポートのテンプレート

複合レポートでは、テンプレートは、1 つ目のコンポーネントレポートで定義されます。ページマスターを含む複合レポートの場合、このテーブルをデフォルトテーブルとして、実際のページレイアウトを作成します。統合された標準複合レポートの場合、このデフォルトコンポーネントは、システムテーブルを使用して作成されます。以下は、この場合のリクエストを示しています。

```
SET COMPONENT='ppt_template'
TABLE FILE SYSCOLUM
SUM TBNAME NOPRINT
IF READLIMIT EQ 1
ON TABLE PCHOLD FORMAT PPTX TEMPLATE 'template_plus.potx' SLIDENUMBER 1
END
```

統合された複合レポートでは、このテーブルには、レポートの他のコンポーネントと同一の主キーが必要です。これには、JOIN および DEFINE など、使用可能な主キーの領域を定義するためのデータの実行前処理が含まれます。

```
SET COMPONENT='ppt_template'
TABLE FILE GGSALES
BY REGION NOPRINT
IF READLIMIT EQ 1
```

```
ON TABLE PCHOLD FORMAT PPTX TEMPLATE 'template_plus.potx' SLIDENUMBER 1
END
```

複合レポートリクエストへのイメージの追加

ページマスター、ページレイアウト、PowerPoint テンプレート (PPTX) にイメージを挿入して、複合ドキュメントを強化することができます。ページマスターに挿入するイメージは、複合ドキュメント内のすべてのページレイアウトに表示されます。ページレイアウトのイメージは、このページにのみ表示されます。PowerPoint テンプレートでドキュメントを表示する場合、イメージはテンプレートに保存され、各スライド上の配置に従って表示されます。

重要：複合レイアウト構文には、非表示のキャリッジリターンおよびラインフィード文字を含めることはできません。この例では、サンプルコードにラインフィード文字が追加され、印刷ページ幅にコードが収まるようコード行が折り返されています。使用する環境でこのサンプルコードを実行するには、コードをテキストエディタにコピーし、各行の末尾に移動して Delete キーを押すことにより、複合レイアウト構文内のラインフィード文字を削除します。

複合レポートリクエストへのイメージの追加

次の複合レポート構文は、ページマスターおよびページレイアウトにイメージを含むドキュメントを作成します。ドキュメントは、PowerPoint テンプレートに表示され、イメージが挿入されます。

```
SET HTMLARCHIVE=ON
*-HOLD_SOURCE
COMPOUND LAYOUT PCHOLD FORMAT PPTX
UNITS=IN, $
SECTION=section1, LAYOUT=ON, MERGE=OFF,
ORIENTATION=LANDSCAPE, PAGESIZE=PPT Slide, SHOW_GLOBALFILTER=OFF, $
PAGELAYOUT=ALL, NAME='Page Master', $
COMPONENT='ppt_template', $
OBJECT=BOX, NAME='line1', POSITION=(0.052 0.500), DIMENSION=(10.000 0.031),
BACKCOLOR=RGB(176 196 222), BORDER-COLOR=RGB(176 196 222), $
OBJECT=BOX, NAME='line2', POSITION=(-12.000 -10.000), DIMENSION=(0.000
```

```
0.000), BACKCOLOR=BLACK, BORDER-COLOR=BLACK, $
OBJECT=BOX, NAME='line3', POSITION=(0.479 0.000), DIMENSION=(0.025 7.600),
BACKCOLOR=RGB(176 196 222), BORDER-COLOR=RGB(176 196 222), $
OBJECT=IMAGE, NAME='image1', IMAGE=webfocus1.gif, ALT='',
POSITION=(0.601 6.700), DIMENSION=(1.248 0.436), $
PAGELAYOUT=1, NAME='Page layout 1', text='Page layout 1',
BOTTOMMARGIN=0.15, TOPMARGIN=0.5, $
COMPONENT='report1', TEXT='report1', POSITION=(1.853 1.289),
DIMENSION=(6.458 5.417), $
OBJECT=IMAGE, NAME='image2', IMAGE=analyst_logo.gif, ALT='',
POSITION=(0.499 0.509), DIMENSION=(2.081 0.477), $
END
```

```
SET COMPONENT='ppt_template'
TABLE FILE SYSCOLUM
SUM TBNAME NOPRINT
IF READLIMIT EQ 1
ON TABLE PCHOLD FORMAT PPTX TEMPLATE '_ibi_template.potx' SLIDENUMBER 2
END
SET COMPONENT='report1'
TABLE FILE IBISAMP/EMPDATA
SUM
  EMPDATA.EMPDATA.SALARY
BY LOWEST EMPDATA.EMPDATA.DEPT
BY EMPDATA.EMPDATA.LASTNAME
ON EMPDATA.EMPDATA.DEPT SUBFOOT
"Subfoot"
" "
ON EMPDATA.EMPDATA.DEPT PAGE-BREAK
ON TABLE SUBHEAD
" Report Heading "
" "
" "
HEADING
"Page Heading "
" "
FOOTING
"Page Footing"
" "
ON TABLE SUBFOOT
"Report Footing"
WHERE EMPDATA.EMPDATA.SALARY GE 40900;
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE SET ASNAMES ON
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PPTX
```

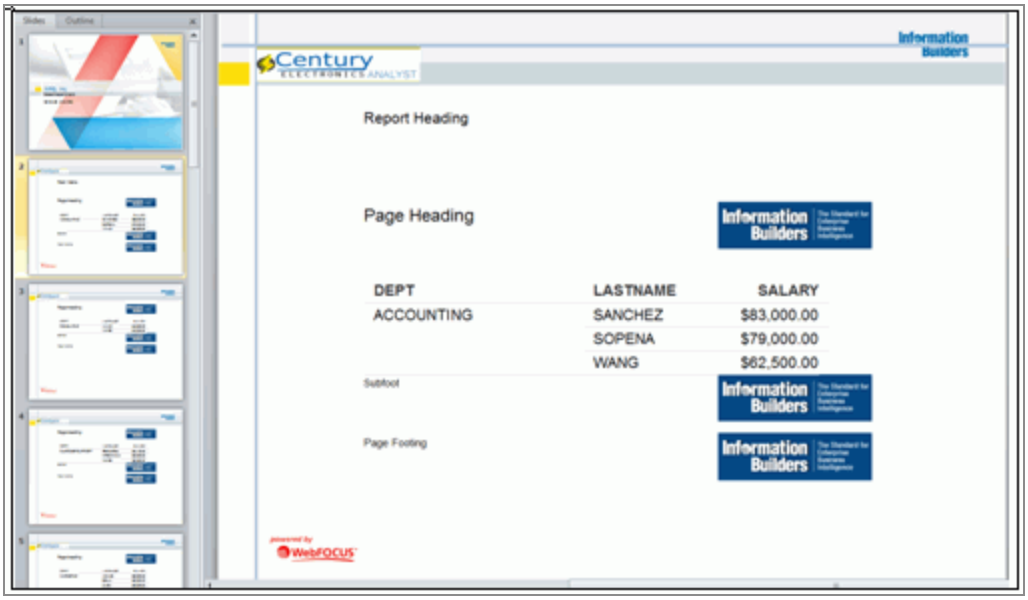
```
ON TABLE SET STYLE *
  PAGESIZE='PPT Slide',
    ORIENTATION=LANDSCAPE,
$
TYPE=REPORT,
  OBJECT=STATUS-AREA,
  JUSTIFY=LEFT,
  PAGE-LOCATION=BOTTOM,
$
```

```
TYPE=REPORT,
  GRID=OFF,
  FONT='ARIAL',
  SIZE=14,
  STYLE=NORMAL,
  SQUEEZE=ON,
  TOPGAP=0.05,
  BOTTOMGAP=0.05,
  BORDER-COLOR=RGB(219 219 219),
  TITLELINE=SKIP,
  TOPMARGIN=.25,
  LEFTMARGIN=2,
  BOTTOMMARGIN=0.05,
$
TYPE=TITLE,
  COLOR=RGB(51 51 51),
  STYLE=-UNDERLINE +BOLD,
$
TYPE=DATA,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=HEADING,
  JUSTIFY=LEFT,
  SIZE=16,
$
TYPE=HEADING,
  IMAGE=smplogo1.gif,
  POSITION=(+4.500000 +0.000000),
$
TYPE=TABFOOTING,
  IMAGE=smplogo1.gif,
  POSITION=(+4.500000 +0.000000),
$
TYPE=TABFOOTING,
  SIZE=10,
```

```
$
TYPE=FOOTING,
  SIZE=10,
$
TYPE=FOOTING,
  IMAGE=smplogo1.gif,
  POSITION=(+4.500000 +0.000000),
$
```

```
TYPE=SUBFOOT,
  SIZE=9,
  BORDER-TOP=LIGHT,
  BORDER-TOP-COLOR=RGB(219 219 219),
$
TYPE=SUBFOOT,
  BY=1,
  IMAGE=smplogo1.gif,
  POSITION=(+4.500000 +0.000000),
$
ENDSTYLE
END
COMPOUND END
```

出力結果は次のとおりです。



テンプレートマスターおよびスライドレイアウト

Microsoft PPTX 2007 以降では、テンプレートに 1 つまたは複数のスライドマスターを追加できるため、さまざまなスライドレイアウトを定義することができます。

スライドマスターは、プレゼンテーションのテーマやスライドレイアウトに関する情報が格納される、スライド階層の最上位スライドです。これらの情報には、背景、色、フォント、効果、プレースホルダサイズ、位置などがあります。

テーマごとに異なるスライドマスターをテンプレートに挿入することで、複数の異なるスタイルまたはテーマ (例、背景、配色、フォント、効果) を組み合わせることができます。

注意：Microsoft PowerPoint スライドレイアウトの追加情報については、Microsoft サポートサイトの「スライドレイアウトとは」を参照してください。

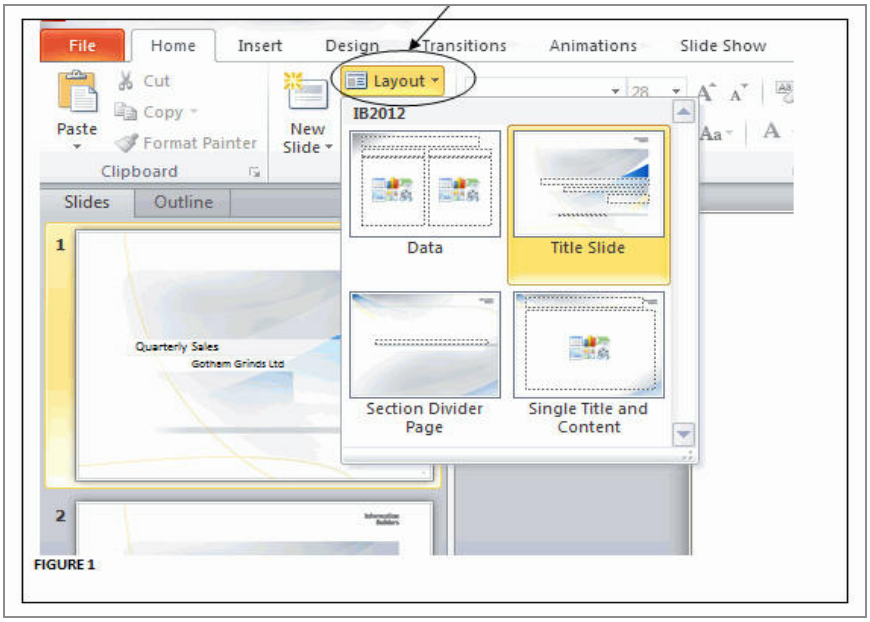
デフォルト設定では、WebFOCUS データが表示されるスライドには、1 つ目のスライドマスターの最初のスライドレイアウトが適用されます。

開発者は、この新機能を使用して、PowerPoint テンプレート (POTX/POTM) またはプレゼンテーションファイル (PPTX/PPTM) に挿入されている任意のスライドマスターの任意のスライドレイアウトを選択することができます。たとえば、特定のスライドレイアウトをスライドに適用して標準レポートの出力を表示する一方で、別のスライドレイアウトを PPTX フォーマットの複合ドキュメントの各ページレイアウトに適用することができます。WebFOCUS から生成される出力は、選択したスライドレイアウトで定義されているスタイル設定の最上面に配置されます。

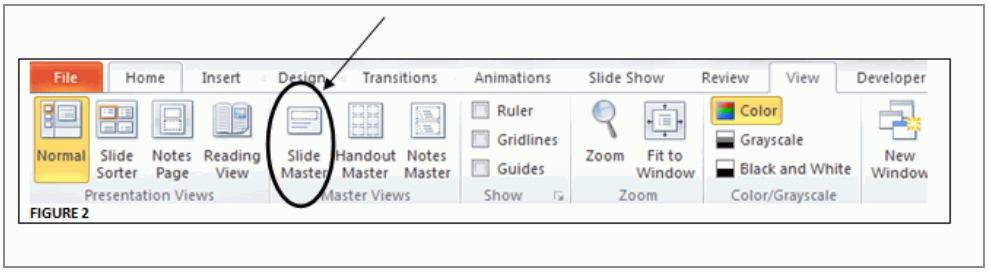
PowerPoint でのスライドマスター属性の指定

テンプレートでスライドマスターおよびスライドレイアウトを指定するには、PowerPoint アプリケーションでファイルを開き、[ホーム] タブを選択して、リボンの [レイアウト] ボタンをクリックします。

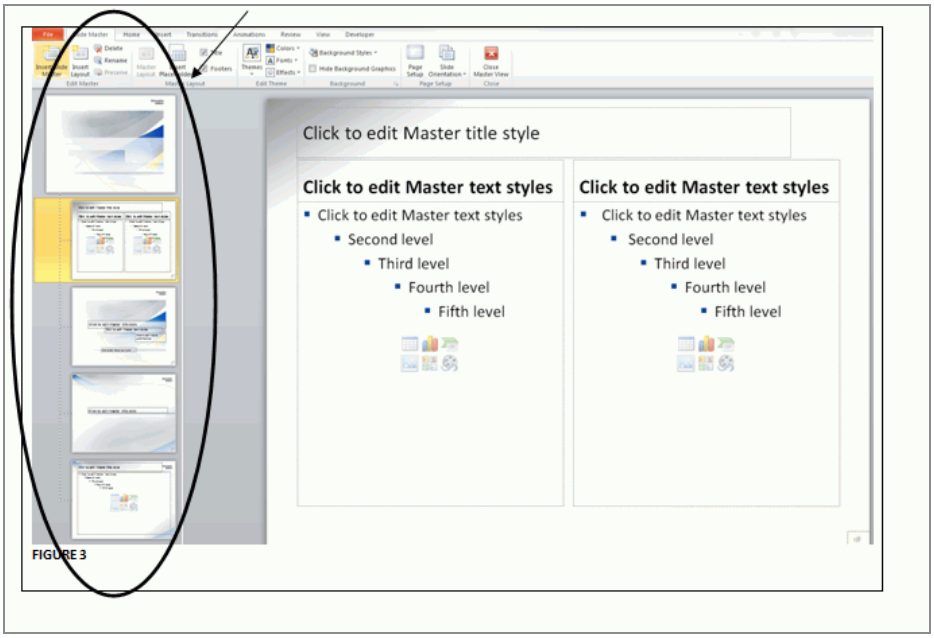
下図のように、すべてのスライドレイアウトおよびラベルがコンテキストメニューに表示されます。



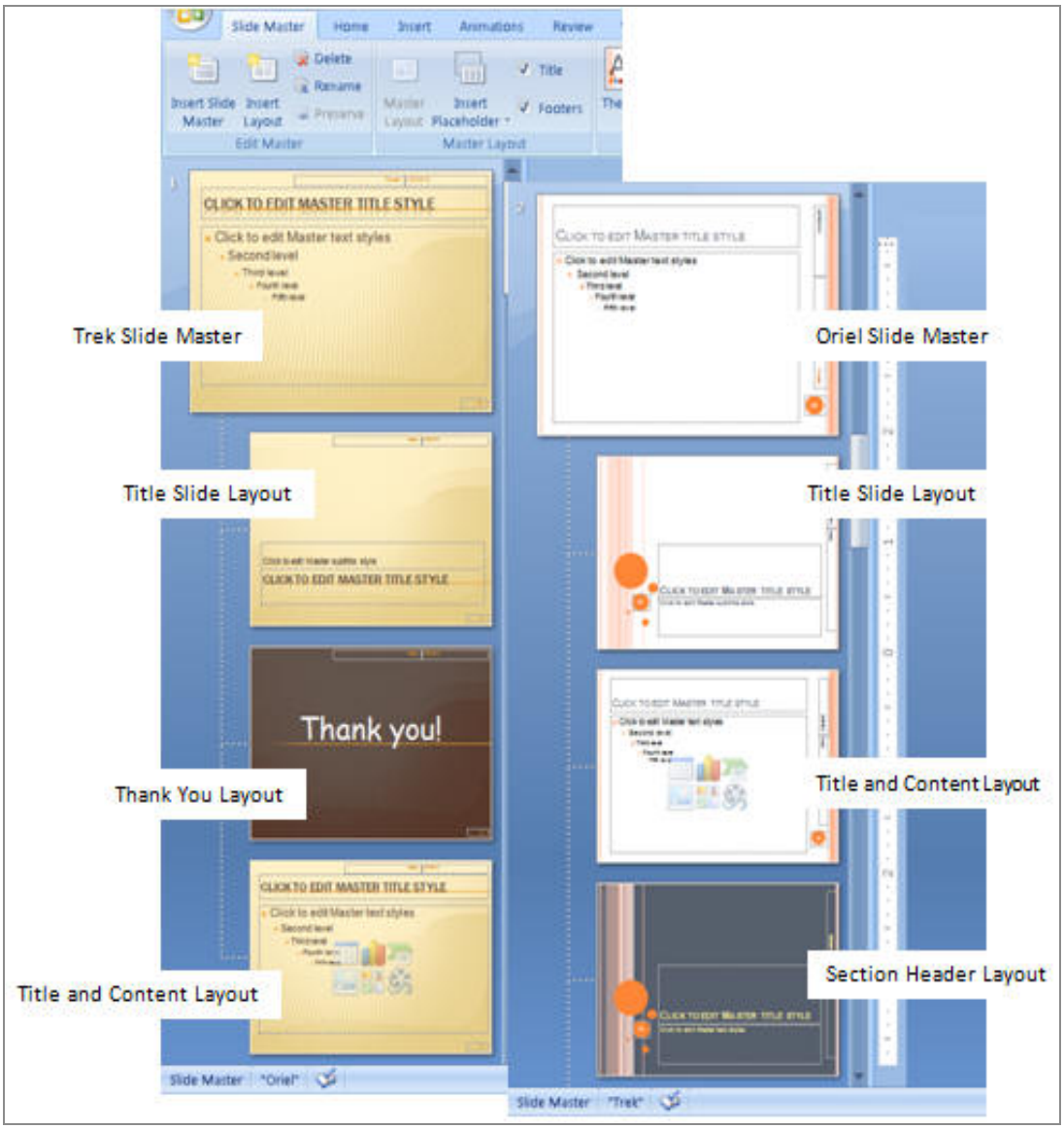
スライドマスターを表示するには、下図のように、[表示] タブを選択し、リボンの [マスター表示] グループで [スライドマスター] ボタンをクリックします。



下図のように、マスター表示が開き、スライドマスターおよび関連するスライドレイアウトが表示されます。



下図は、この例で使用したテンプレートのスライドマスター表示を示しています。この場合、2 つのスライドマスターが含まれます (Trek および Oriel)。各マスター内のイメージには、レポート生成用に選択可能なレイアウトが表示されます。スライドマスターおよびレイアウトの名前は、各スライドのイメージ上にマウスポインタを置くと識別できます。末尾の「スライドマスター」または「スライドレイアウト」を除外した名前を使用します。



PowerPoint でのスライドマスター属性の指定

単一レポートの場合は、次の構文を使用します。

```
TYPE=REPORT,SLIDE-MASTER='slidemaster_name',SLIDE-LAYOUT='layout_name', $
```

複合レポートの場合は、次の構文を使用します。

```
PAGELAYOUT=n, NAME='Page layout (n)', SLIDE-MASTER='slidemaster_name', SLIDE-
LAYOUT='layout_name', $
```

注意：スライドマスターおよびスライドレイアウトは、SECTION 構文内のページマスターまたは任意のページレイアウトで定義できます。

複合レポートによる複数マスターへのアクセス

次の構文は、2 つのスライドマスターおよび 4 つの異なるスライドレイアウトを使用するテンプレートに基づいて、完全な PPTX プレゼンテーションを作成します。この例に示す 5 つのプロシジャ (.fex) は、それぞれ個別のファイルにコピーする必要があります。

```
COMPOUND LAYOUT PCHOLD FORMAT PPTX
UNITS=IN, $
SECTION=section1, LAYOUT=ON, SLIDE-MASTER='TREK',
SLIDE-LAYOUT='Title and Content',
MERGE=OFF, ORIENTATION=LANDSCAPE,
PAGESIZE=PPT Slide, SHOW_GLOBALFILTER=OFF, $
PAGELAYOUT=1, NAME='Page layout 1', SLIDE-MASTER='TREK',
SLIDE-LAYOUT='Title Slide', text='Page layout 1',
BOTTOMMARGIN=0.5, TOPMARGIN=0.5, $
COMPONENT='ppt_template', $
OBJECT=STRING, NAME='text1', TEXT='Gotham Grinds Sales Summary',
POSITION=(1.083 3.117), MARKUP=OFF, WRAP=ON, DIMENSION=(6.799 0.620),
font='TREBUCHET MS', color=RGB(0 0 0), size=36, $
OBJECT=STRING, NAME='text2', TEXT='Profit By Category',
POSITION=(1.100 3.733), MARKUP=OFF, WRAP=ON, DIMENSION=(6.833 0.500),
font='TREBUCHET MS', color=RGB(0 0 0), size=18, $
OBJECT=STRING, NAME='text3', TEXT='Prepared By:<br><br>Anne T Jones,
EVP of Sales<br>Joe F Smith, VP of Sales<br><br>&DATEtrMDYY',
POSITION=(5.683 5.433), MARKUP=ON, WRAP=ON, DIMENSION=(4.049 1.721),
font='TREBUCHET MS', color=RGB(0 0 0), size=18, $
PAGELAYOUT=2, NAME='Page layout 2', text='Page layout 2',
BOTTOMMARGIN=0.025, TOPMARGIN=4.8, $
```

```
OBJECT=STRING, NAME='pl2_text2', TEXT='Sales By Region',
POSITION=(0.5 0.325), MARKUP=ON, WRAP=ON, DIMENSION=(4.146 0.609),
font='TREBUCHET MS', color=RGB(0 0 0), style=bold, size=24, $
OBJECT=STRING, NAME='pl2_text3', TEXT=' ', POSITION=(0.5 .725), MARKUP=ON,
WRAP=ON, DIMENSION=(4.146 0.609), font='TREBUCHET MS', color=RGB(0 0 0),
style=bold, size=20, $
COMPONENT='report1', TEXT='report1', POSITION=(2.2 4.8),
```

```
DIMENSION=(* *), $
COMPONENT='chart1', TEXT='chart1', POSITION=(0.733 1.25),
DIMENSION=(8.680 3.10), COMPONENT-TYPE=GRAPH, $
PAGELAYOUT=5, NAME='Page layout 5', SLIDE-MASTER='ORIEL',
SLIDE-LAYOUT='Section Header', text='Page layout 3',
BOTTOMMARGIN=0.5, TOPMARGIN=0.5, $
OBJECT=STRING, NAME='pl3text1', TEXT='Sales Performance Regional Breakdowns',
POSITION=(1.0 3.35), MARKUP=OFF, WRAP=ON,
DIMENSION=(6.799 0.620), font='TREBUCHET MS', color=RGB(0 0 0), size=24,$
COMPONENT='DfltCmpt2_3', POSITION=(0 0), DIMENSION=(0 0), $
PAGELAYOUT=6, NAME='Page layout 2', text='Page layout 4',
SLIDE-MASTER='ORIEL', SLIDE-LAYOUT='Title and Content',
BOTTOMMARGIN=0.025, TOPMARGIN=4.8, $
OBJECT=STRING, NAME='pl4_text2', TEXT='Sales By Category',
POSITION=(0.5 0.325), MARKUP=ON, WRAP=ON, DIMENSION=(4.146 0.609),
font='TREBUCHET MS', color=RGB(0 0 0), style=bold, size=24, $
OBJECT=STRING, NAME='pl4_text3', TEXT=' ', POSITION=(0.5 .725), MARKUP=ON,
WRAP=ON, DIMENSION=(4.146 0.609), font='TREBUCHET MS', color=RGB(0 0 0),
style=bold, size=20, $
COMPONENT='report2_2', TEXT='report2', POSITION=(2.2 4.8),
DIMENSION=(* *), $
COMPONENT='chart2_2', TEXT='chart2', POSITION=(0.733 1.25),
DIMENSION=(8.680 3.10), COMPONENT-TYPE=GRAPH, $
COMPONENT='report2_3', TEXT='report3', POSITION=(0.5 .725),
DIMENSION=(3 3), $
PAGELAYOUT=9, NAME='Page layout 9', SLIDE-MASTER='TREK',
SLIDE-LAYOUT='Thank You', text='Page layout 3',
BOTTOMMARGIN=0.5, TOPMARGIN=0.5, $
COMPONENT='DfltCmpt9', POSITION=(0 0), DIMENSION=(0 0), $
END

SET COMPONENT='ppt_template'
TABLE FILE SYSCOLUM
SUM TBNAME NOPRINT
IF READLIMIT EQ 1
ON TABLE PCHOLD FORMAT PPTX TEMPLATE 'template_plus.potx' SLIDENUMBER 1
END
```

```
SET COMPONENT='report1'
-INCLUDE GG_RPT1

SET COMPONENT='chart1'
-INCLUDE GG_CHART1

SET COMPONENT='DfltCmpt2_3'
TABLE FILE SYSCOLUM
```

```
" "
SUM TBNAME NOPRINT
IF READLIMIT EQ 1
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
END

SET COMPONENT='report2_2'
-INCLUDE GG_RPT2
SET COMPONENT='chart2_2'
-INCLUDE GG_CHART2
SET COMPONENT='report2_3'
-INCLUDE GG_RPT3

SET COMPONENT='DfltCmpt9'
TABLE FILE SYSCOLUM
" "
SUM TBNAME NOPRINT
IF READLIMIT EQ 1
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
END

COMPOUND END
```

```
-*gg_rpt1.fex
TABLE FILE GGSALES
SUM DOLLARS/D12CM BUDDOLLARS/D12CM UNITS/D12 BUDUNITS/D12
BY REGION
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
     INCLUDE = warm,
$
ENDSTYLE
END
```

```
-*gg_rpt2.fex
TABLE FILE GGSALES
SUM DOLLARS/D12CM BUDDOLLARS/D12CM UNITS/D12 BUDUNITS/D12
BY REGION NOPRINT
BY CATEGORY
ON REGION PAGE-BREAK
```

```
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
     INCLUDE = warm,
$
ENDSTYLE
END
```

```
-*gg_chart1.fex
ENGINE INT CACHE SET ON
-DEFAULTH &WF_STYLE_UNITS='INCHES';
-DEFAULTH &WF_STYLE_HEIGHT='4.21875';
-DEFAULTH &WF_STYLE_WIDTH='8.020833';
-DEFAULTH &WF_TITLE='WebFOCUS Report';
GRAPH FILE GGSALES
SUM DOLLARS UNITS
BY REGION NOPRINT
ON GRAPH PCHOLD FORMAT PPTX
ON GRAPH SET HTMLENCODE ON
ON GRAPH SET GRAPHDEFAULT OFF
ON GRAPH SET ARGRAPHENGIN JSCHART
ON GRAPH SET VZERO OFF
ON GRAPH SET GRWIDTH 1
ON GRAPH SET UNITS &WF_STYLE_UNITS
ON GRAPH SET HAXIS &WF_STYLE_WIDTH
ON GRAPH SET VAXIS &WF_STYLE_HEIGHT
ON GRAPH SET GRMERGE ADVANCED
ON GRAPH SET GRMULTIGRAPH 0
ON GRAPH SET GRLEGEND 1
ON GRAPH SET GRXAXIS 0
ON GRAPH SET LOOKGRAPH VBAR
ON GRAPH SET STYLE *
*GRAPH_SCRIPT
setPieDepth(0);
setPieTilt(0);
setDepthRadius(0);
setCurveFitEquationDisplay(false);
setPlace(true);
*END
```

```
INCLUDE = warm,$
TYPE=REPORT, TITLETEXT=&WF_TITLE.QUOTEDSTRING, $
*GRAPH_SCRIPT
setReportParsingErrors(false);
setSelectionEnableMove(false);
```

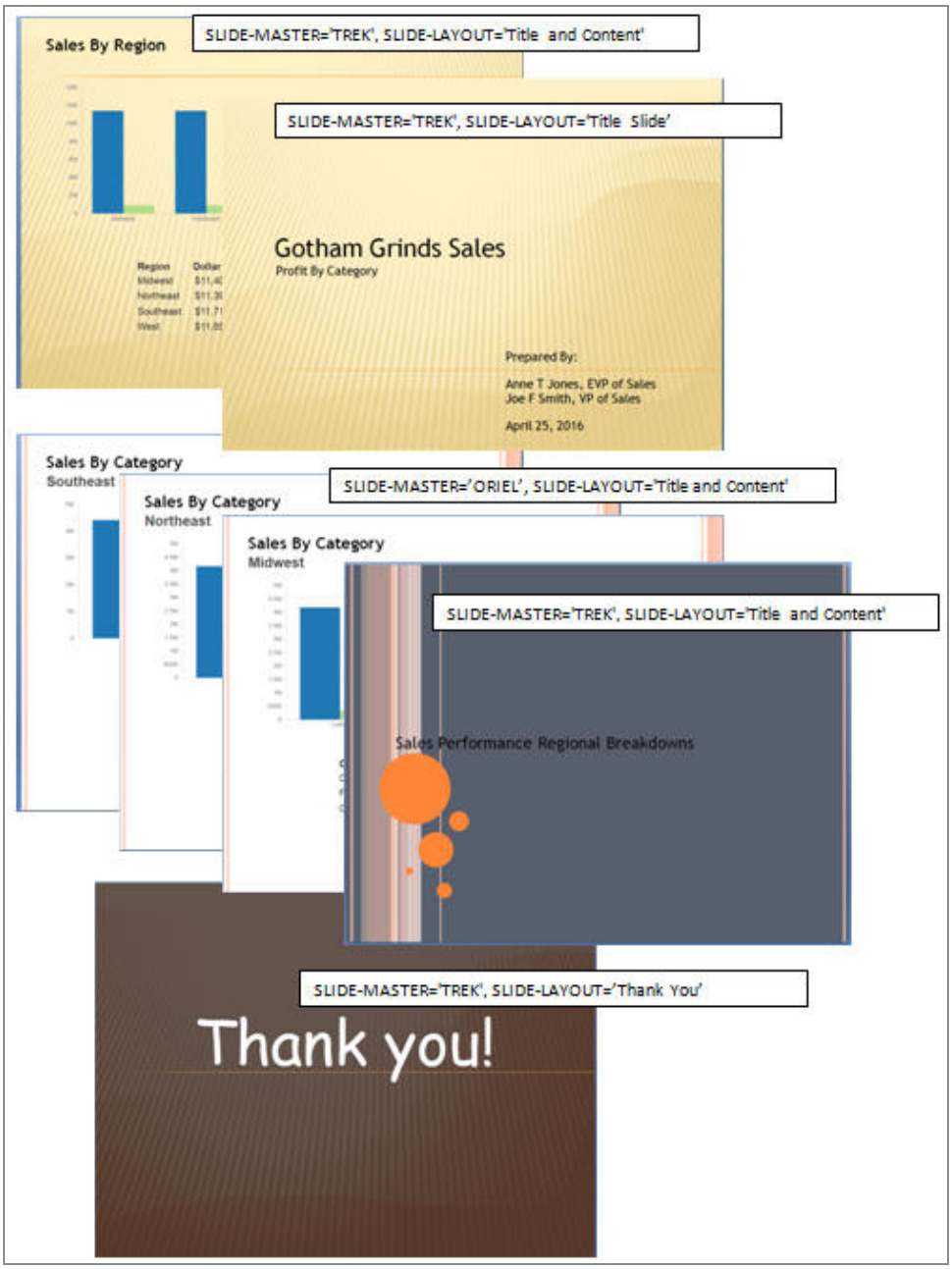
```
setTransparentBorderColor(getChartBackground(),true);
setTransparentFillColor(getFrameSide(),true);
setTransparentBorderColor(getFrameSide(),true);
setTransparentFillColor(getFrameBottom(),true);
setTransparentBorderColor(getFrameBottom(),true);
*GRAPH_SCRIPT
-* Make the chart background, borders, etc. transparent:
setTransparentFillColor(getChartBackground(),true);
setFillColor(getChartBackground(),new Color(255,255,255,0));
setTransparentBorderColor(getChartBackground(),true);
setTransparentFillColor(getFrameSide(),true);
setTransparentBorderColor(getFrameSide(),true);
setTransparentFillColor(getFrameBottom(),true);
setTransparentBorderColor(getFrameBottom(),true);
setPlace(true);
*END
ENDSTYLE
END
-RUN
```

```
-*gg_chart2.fex
-DEFAULTH &WF_STYLE_UNITS='INCHES';
-DEFAULTH &WF_STYLE_HEIGHT='4.21875';
-DEFAULTH &WF_STYLE_WIDTH='8.020833';
-DEFAULTH &WF_TITLE='WebFOCUS Report';
GRAPH FILE GGSALES
SUM DOLLARS UNITS
BY REGION NOPRINT
BY CATEGORY
ON GRAPH PCHOLD FORMAT PPTX
ON GRAPH SET HTMLENCODE ON
ON GRAPH SET GRAPHDEFAULT OFF
ON GRAPH SET ARGRAPHENGIN JSCHART
ON GRAPH SET VZERO OFF
ON GRAPH SET GRWIDTH 1
ON GRAPH SET UNITS &WF_STYLE_UNITS
ON GRAPH SET HAXIS &WF_STYLE_WIDTH
ON GRAPH SET VAXIS &WF_STYLE_HEIGHT
ON GRAPH SET GRMERGE ADVANCED
ON GRAPH SET GRMULTIGRAPH 1
ON GRAPH SET GRLEGEND 1
ON GRAPH SET GRXAXIS 0
ON GRAPH SET LOOKGRAPH VBAR
ON GRAPH SET STYLE *
```

```
*GRAPH_SCRIPT
setPieDepth(0);
setPieTilt(0);
setDepthRadius(0);
setCurveFitEquationDisplay(false);
setPlace(true);
*END
INCLUDE = warm,$
TYPE=REPORT, TITLETEXT=&WF_TITLE.QUOTEDSTRING, $
*GRAPH_SCRIPT
setReportParsingErrors(false);
setSelectionEnableMove(false);
setTransparentBorderColor(getChartBackground(),true);
setTransparentFillColor(getFrameSide(),true);
setTransparentBorderColor(getFrameSide(),true);
setTransparentFillColor(getFrameBottom(),true);
setTransparentBorderColor(getFrameBottom(),true);
*GRAPH_SCRIPT
-* Make the chart background, borders, etc. transparent:
setTransparentFillColor(getChartBackground(),true);
setFillColor(getChartBackground(),new Color(255,255,255,0));
setTransparentBorderColor(getChartBackground(),true);
setTransparentFillColor(getFrameSide(),true);
setTransparentBorderColor(getFrameSide(),true);
setTransparentFillColor(getFrameBottom(),true);
setTransparentBorderColor(getFrameBottom(),true);
setPlace(true);
*END
ENDSTYLE
END
-RUN
```

```
-*gg_rpt3.fex
TABLE FILE GGSALES
BY REGION PAGE-BREAK NOPRINT
HEADING
"<REGION"
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
     INCLUDE = warm,
$
TYPE=HEADING, SIZE=20,
$
ENDSTYLE
END
```

出力結果は次のとおりです。



WebFOCUS コンテンツと PowerPoint テンプレートコンテンツとの結合

TEMPLATE-ACTION スタイルシート属性を使用して、WebFOCUS コンテンツと PowerPoint テンプレートコンテンツを結合することができます。MERGE アクションを使用することで、ネイティブ PowerPoint のテキストボックス内のコメントなどのコンポーネントを、結果の PowerPoint PPTX 出力で編集することができます。

WebFOCUS コンテンツと PowerPoint テンプレートコンテンツとの結合

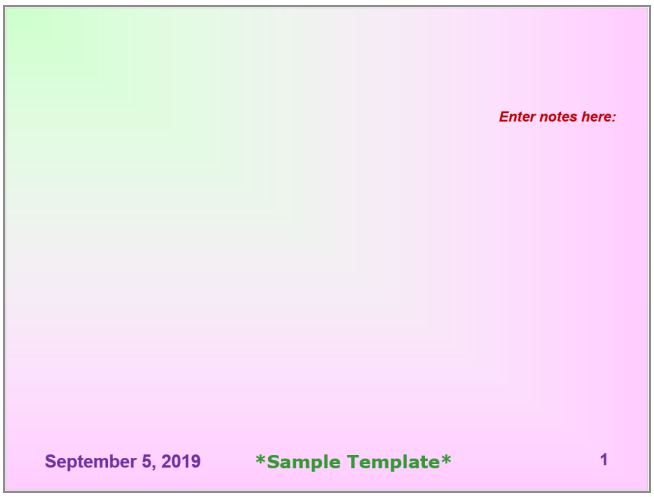
WebFOCUS コンテンツと PowerPoint テンプレートコンテンツを結合するには、プロシジャに次のスタイルシート構文を追加します。

```
TYPE=REPORT, TEMPLATE-ACTION=MERGE ,$
```

注意：TEMPLATE-ACTION=REPLACE スタイルシート属性を追加して、デフォルト動作を保持することができます。この場合、WebFOCUS 出力によって、すべてのターゲットスライドのコンテンツが上書きされます。

WebFOCUS コンテンツと PowerPoint テンプレートコンテンツとの結合

次の例は、下図のように、WebFOCUS コンテンツと PowerPoint テンプレートコンテンツを結合する方法を示しています。PowerPoint テンプレートの名前は、my_template.potx です。

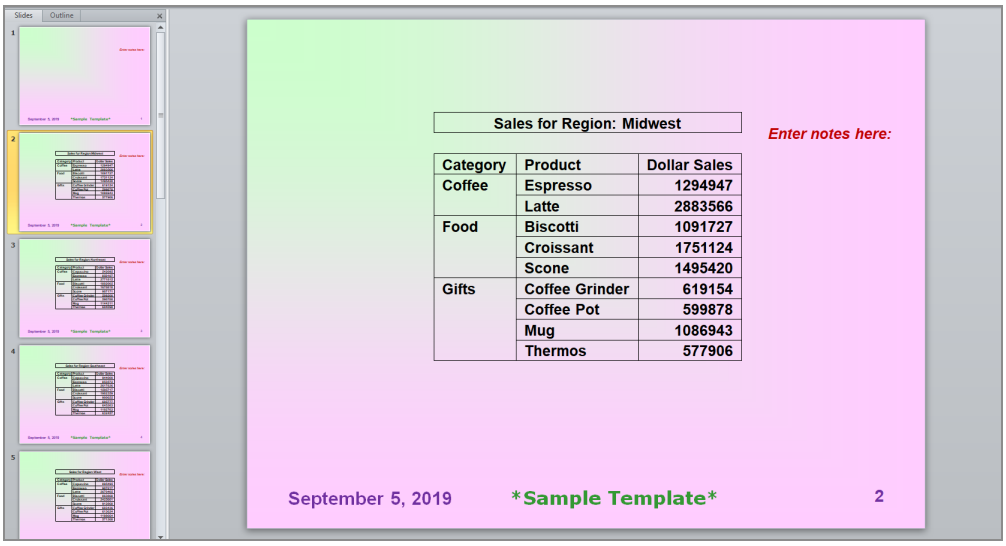


結合プロシジャ

次のプロシジャには、WebFOCUS コンテンツと PowerPoint テンプレートコンテンツ (my_template.potx) を結合するための TEMPLATE-ACTION=MERGE スタイルシート属性が含まれます。

```
TABLE FILE GGSALES
HEADING CENTER
"Sales for Region: <REGION"
SUM DOLLARS
BY REGION NOPRINT PAGE-BREAK
BY CATEGORY
BY PRODUCT

ON TABLE HOLD AS MERGEPPTX FORMAT PPTX TEMPLATE 'my_template.potx'
SLIDENUMBER 2

ON TABLE SET STYLE *
TYPE=REPORT, PAGESIZE=PPT-SLIDE, ORIENTATION=LANDSCAPE,
  TEMPLATE-ACTION=MERGE,
  LEFTMARGIN=2.75, TOPMARGIN=.75, BOTTOMMARGIN=1.5,
  FONT=ARIAL, SQUEEZE=ON, BORDER=LIGHT, STYPE=BOLD, SIZE=16, $
ENDSTYLE

END
```

出力結果は次のとおりです。

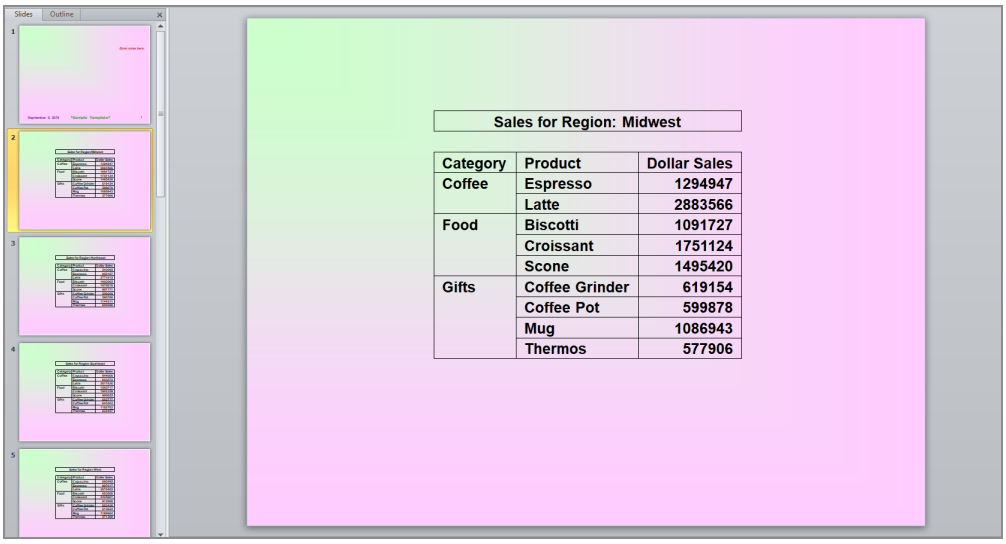


ターゲットスライドコンテンツの上書き

次のプロシジャには、デフォルト動作を保持し、すべてのターゲットスライドコンテンツを上書きするための TEMPLATE-ACTION=REPLACE スタイルシート属性が含まれています。

```
TABLE FILE GGSALES
HEADING CENTER
"Sales for Region: <REGION"
SUM DOLLARS
BY REGION NOPRINT PAGE-BREAK
BY CATEGORY
BY PRODUCT

ON TABLE HOLD AS REPLACEPPTX FORMAT PPTX TEMPLATE 'my_template.potx'
SLIDENUMBER 2
ON TABLE SET STYLE *
TYPE=REPORT, PAGESIZE=PPT-SLIDE, ORIENTATION=LANDSCAPE,
  TEMPLATE-ACTION=REPLACE,
  LEFTMARGIN=2.75, TOPMARGIN=.75, BOTTOMMARGIN=1.5,
  FONT=ARIAL, SQUEEZE=ON, BORDER=LIGHT, STYPE=BOLD, SIZE=16, $
ENDSTYLE

END
```

出力結果は次のとおりです。



ReportCaster による配信およびバースト

ReportCaster による配信は、単純なレポート、イメージを含むレポート、PPTX 連プレート、統合された複合レポート、およびグラフでサポートされます。バースト機能は、単純なレポート、イメージを含むレポート、PPTX テンプレート、統合された複合レポートでサポートされます。詳細は、『ibi™ WebFOCUS® ReportCaster 利用ガイド』を参照してください。

PPTX の制限事項

PPTX 出力フォーマットの使用に関する制限事項は次のとおりです。

- レポート要素内のイメージ位置揃え

複合レポート構文では、PPTX 出力フォーマットに使用に関して次の制限事項があります。

- パネル化されたレポート
- ネストされたレポート
- マルチパネルレポートまたは、マルチカラムを含むレポート
- OPEN、CLOSE、または NOBREAK コマンド

関連情報

これらのトピックの関連情報は、次の WebFOCUS マニュアルを参照してください。

- 『ibi™ WebFOCUS® アプリケーション作成ガイド』
- 『ibi™ WebFOCUS® 関数リファレンス』
- 『ibi™ WebFOCUS® ReportCaster 利用ガイド』

レポートの別リソースへのリンク

スタイルシート属性を使用して、任意のレポートコンポーネントからのリンクを定義することができます。リンクには、次の用途があります。

- レポートを生成するプロシジャをリンクすることにより、一連のドリルダウンレポートを作成する。
- URL へのリンク。これらの URL には、Web ページ、Web サイト、Servlet プログラムのほか、Email アプリケーションなどの World Wide Web 以外のリソースを指定することもできます。
- JavaScript 関数を実行してレポートデータの追加分析を行う。

レポート内のグラフィックイメージやレポートデータから、リンクを作成することができます。また、グラフからリンクを作成することもできます。グラフからのリンクについての詳細は、グラフの作成を参照してください。

スタイルシートを使用したリンク

スタイルシートを使用して、任意のレポートコンポーネントからのリンクを定義することができます。見出しと脚注を含むレポートデータ、およびグラフィックイメージ (会社ロゴや商品イメージなど) から、別のレポート、プロシジャ、URL、または JavaScript 関数へのリンクを作成することができます。

動的なリンクを作成することもできます。動的リンクを使用すると、選択したレポートコンポーネントの値がリンクしたレポート (プロシジャ、URL、または JavaScript 関数) に渡されます。このリソースは、渡された値を使用して、返される結果を動的に決定します。1 つまたは複数のパラメータを渡すことができます。詳細は、パラメータの作成を参照してください。

スタイルシートを使用したリンクを作成するには

次の手順は、スタイルシートを使用したリンク作成の基本的な概要を示しています。

手順

1. リンクを実行するためにユーザが Web ブラウザで選択する、レポートコンポーネントを特定します。
2. 実行する埋め込みプロシジャ、URL、または JavaScript 関数の名前を指定します。
3. 必要に応じて、リンクの詳細を定義するパラメータを指定します。

別レポートへのリンク

リンクを使用すると、詳細レポートにドリルダウンしたり、レポート内で割り当てられたホットスポット (リンク) を選択してプロシジャを実行したりすることが可能になります。複数のレポートをリンクすることで、基本レポートの情報を補足する詳細データに簡単にアクセスすることができます。ドリルダウンレポートには、基本レポートのデータとは無関係な情報を含めることも、基本レポートの特定のデータ値をさらに拡張した情報を含めることもできます。

リンクを作成する場合、基本レポートからのレポート、およびドリルダウンレポートにリンクするためのレポートが必要です。ドリルダウンレポートが基本レポートの特定のデータ値に基づいている場合、パラメータを作成し、その値もドリルダウンレポートに渡す必要があります。詳細は、パラメータの作成を参照してください。

レポートおよびプロシジャへのリンク

```
TYPE=type, [subtype], FOCEXEC=fex[(parameters ...)], [TARGET=frame,]
[ALT = 'description',] $
```

説明

type

リンクを実行するために、ユーザが Web ブラウザで選択するレポートコンポーネントを特定します。TYPE 属性とその値は、構文の先頭部分に配置する必要があります。

subtype

追加の属性です。COLUMN、LINE、ITEM などがあり、フォーマット設定対象のレポートコンポーネントを特定するために必要です。レポートコンポーネントの特定についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

fex

レポートコンポーネントを選択するときに、実行されるリンク先プロシジャのファイル名を指定します。

注意：プロシジャに NONE (すべて大文字) という名前を付けることはできません。プロシジャ名として NONE を使用すると、構文エラーが発生します。大文字と小文字の混在および小文字のみの場合は使用可能です。

WebFOCUS のファイル名の決定についての詳細は、別レポートへのリンクを参照してください。

関連パラメータを含む FOCEXEC=*fex* 引数は最大 2400 バイトの長さにすることができます。プロシジャ引数は 2 行以上に拡張することができます。詳細は、WebFOCUS スタイルシートの作成と管理を参照してください。

parameters

レポート、URL、または JavaScript 関数へ渡される値です。詳細は、パラメータの作成を参照してください。

frame

ドリルダウンリンクからの出力が表示される Web ページのターゲットフレームを指定します。詳細は、ターゲットフレームの指定を参照してください。

description

HTML レポートでサポートされる、セクション 508 (米国リハビリテーション法 508 条) アクセシビリティに準拠したリンクの説明テキストを指定します。説明は、一重引用符 (') で囲みます。

この説明は、レポート出力のリンク上にマウスポインタ (カーソル) を置いたときのポップアップ説明としても表示されます。

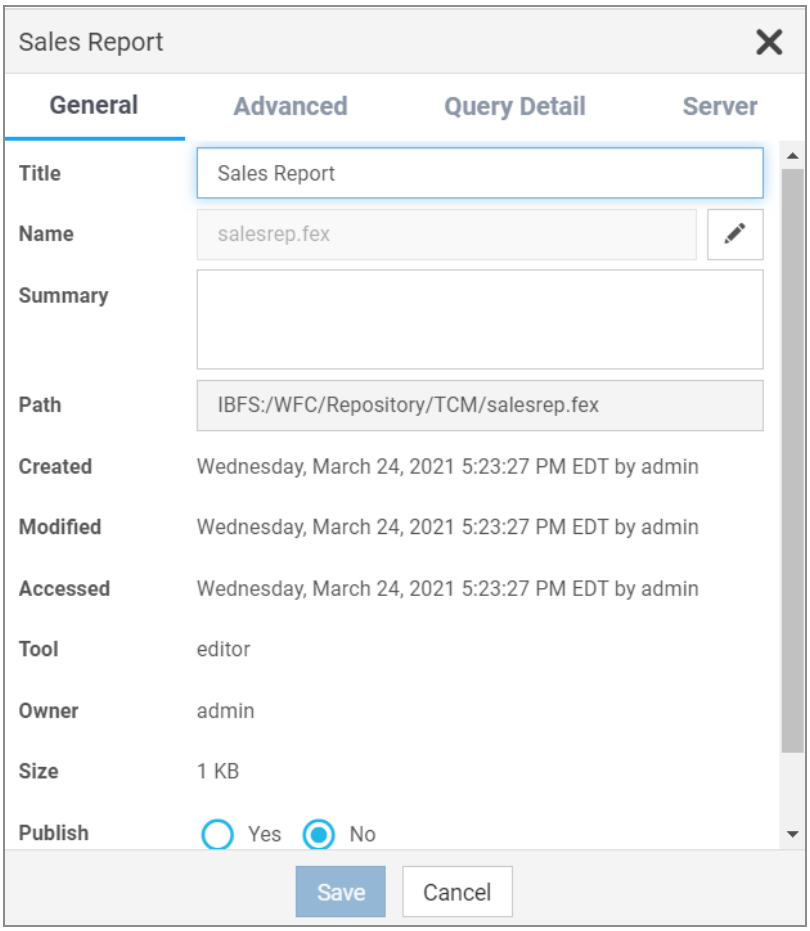
PDF フォーマットでのドリルダウンレポート使用上の注意

PDF フォーマットのドリルダウンレポートから元のレポートに戻る場合、[戻る] ボタンをすばやく 2 回クリックする必要があります。別の方法として、[戻る] ボタンの右側にあるドロップダウンリストからブラウザ履歴を表示し、2 つ前のリンクを選択することもできます。1 つ前の履歴はリダイレクトページを指定する項目で、そのタイトルは WFServlet へのアクセスに使用された方法に基づいて表示されます。さらに 1 つ前の項目には「WebFOCUS レポート」というタイトルが表示され、元の PDF レポートに戻ります。

WebFOCUS ファイル名を決定するには

手順

1. レポート名を右クリックし、[プロパティ] を選択します。[レポートプロパティ] ダイアログボックスが開きます。
2. [名前] の横にファイル名が表示されます。次の例では、ファイル名は「Sales Report」です。ファイル拡張子 (.fex) やディレクトリの場所とスラッシュ (/) は挿入しないでください。



脚注からレポートへのリンク

次のリクエストは商品の売上を集計し、データを地域別、州別、および店舗コード別にソートします。店舗コードには、店舗の売上 (商品別または日付別) の詳細レポートへのリンクが設定され、中間脚注にも表示されます。中間脚注の各行には、2 つのテキストオブジェクト、および 1 つの埋め込みフィールドが含まれます。関連するスタイルシートは、リクエスト内で太字で表示されています。

メインレポートは次のとおりです。

```
TABLE FILE GGSALES
HEADING
"Sales Report"
SUM DOLLARS/I08M
BY REGION BY ST BY STCD
ON STCD SUBFOOT
```

```
"View Store <STCD Sales By Product"
" "
"View Store <STCD Sales By Date"
ON REGION PAGE-BREAK
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=HEADING, SIZE=12, STYLE=BOLD, $
TYPE=SUBFOOT, LINE=1, OBJECT=TEXT, ITEM=2, COLOR=GREEN,
     FOCEXEC=PRDSALES(STOREID=STCD), $
TYPE=SUBFOOT, LINE=3, OBJECT=TEXT, ITEM=2, COLOR=BLUE,
     FOCEXEC=HSTSALES(STOREID=STCD), $
ENDSTYLE
END
```

スタイルシートを使用して、脚注句の Sales By Product は「PRDSALES」という名前の 2 つ目のプロシジャにリンクされ、それが脚注句に表示される STCD の値に渡されます。脚注句の Sales By Date は「HSTSALES」という名のプロシジャにリンクされ、それが脚注句に表示される STCD の値に渡されます。

リンクレポート HSTSALES のリクエストは次のとおりです。

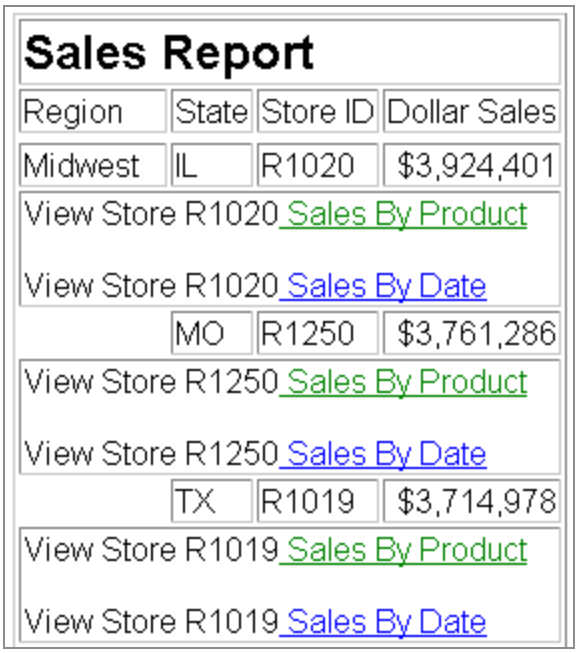
```
TABLE FILE GGSALES
SUM UNITS
BY STCD
BY DATE
WHERE STCD = '&STOREID'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

リンクレポート PRDSALES のリクエストは次のとおりです。

```
TABLE FILE GGSALES
SUM UNITS
BY STCD
BY PRODUCT
WHERE STCD = '&STOREID'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

下図は、メインレポートの出力です。R1020 店に対して「Sales By Product」を選択すると、R1020 の値が PRDSALES プロシジャに渡されます。R1019 店に対して「Sales By Date」を選択すると、R1019 の値が HSTSALES プロシジャに渡されます。

出力結果は次のとおりです。



Sales Report

Region	State	Store ID	Dollar Sales
Midwest	IL	R1020	$3,924,401
View Store R1020 Sales By Product View Store R1020 Sales By Date			
	MO	R1250	$3,761,286
View Store R1250 Sales By Product View Store R1250 Sales By Date			
	TX	R1019	$3,714,978
View Store R1019 Sales By Product View Store R1019 Sales By Date			

R1020 店に対して「Sales By Product」リンクをクリックした場合は、出力結果は次のようになります。

Store ID	Product	Unit Sales
R1020	Biscotti	29413
	Coffee Grinder	19339
	Coffee Pot	15785

	Croissant	43300
	Espresso	32237
	Latte	77344
	Mug	30157
	Scone	45355
	Thermos	14651

URL へのリンク

任意のレポートコンポーネントから任意の URL へのリンクを定義することができます。この URL には、Web ページ、Web サイト、Servlet プログラム、および Email アプリケーションなどの World Wide Web 以外のリソースを含めることが可能です。リンクの定義後、レポートコンポーネントを選択して URL にアクセスすることができます。

動的なリンクを作成することもできます。動的リンクを使用すると、選択したレポートコンポーネントの値が URL に渡されます。このリソースは、渡された値を使用して、返される結果を動的に決定します。1 つまたは複数のパラメータを渡すことができます。詳細は、パラメータの作成を参照してください。

URL へのリンク

```
TYPE=type, [subtype], URL=url[(parameters ...)], [TARGET=frame,] [ALT =
'description',] $
```

説明

type

リンクを実行するために、ユーザが Web ブラウザで選択するレポートコンポーネントを特定します。TYPE 属性とその値は、構文の先頭部分に配置する必要があります。

subtype

追加の属性です。COLUMN、LINE、ITEM などがあり、フォーマット設定対象のレポートコンポーネントを特定するために必要です。レポートコンポーネントの特定についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

url

任意の有効な URL を特定します。これには、WebFOCUS Servlet プログラムを指定する URL や、括弧で囲まれたレポートフィールド名も含まれます。この値はリンクのジャンプ先として有効な URL である必要があります。

注意

- URL=url 引数の最大長は、関連する variable=object パラメータを含めて、ブラウザで許容される最大バイト数が上限になります。使用するブラウザでのこの制限についての詳細は、ブラウザベンダーのサポートサイトを参照してください。 URL 引数は 2 行以上に拡張することができます。詳細は、WebFOCUS スタイルシートの作成と管理を参照してください。

  URL の長さは、ブラウザで許容される最大バイト数が上限になります。使用するブラウザでのこの制限についての詳細は、ブラウザベンダーのサポートサイトを参照してください。

- URL がパラメータを持つ WebFOCUS Servlet プログラムを参照する場合、URL を疑問符 (?) で終了する必要があります

parameters

URL に渡される値です。詳細は、パラメータの作成を参照してください。

frame

ドリルダウンリンクからの出力が表示される Web ページのターゲットフレームを指定します。詳細は、ターゲットフレームの指定を参照してください。

description

HTML レポートでサポートされる、セクション 508 (米国リハビリテーション法 508 条) アクセシビリティに準拠したリンクの説明テキストを指定します。説明は、一重引用符 (')

で囲みます。

この説明は、レポート出力のリンク上にマウスポインタ (カーソル) を置いたときのポップアップ説明としても表示されます。

URL へのリンク

下図は、レポートから URL にリンクする方法を示しています。「Click here to access the IB homepage」という見出しは、URL :www.ibi.com にリンクされています。関連するスタイルシートは、リクエスト内で太字で表示されています。

```
TABLE FILE GGSALES
ON TABLE SET PAGE-NUM OFF
SUM UNITS AND DOLLARS
BY CATEGORY BY REGION
HEADING
"Regional Sales Report"
"Click here to access the IB homepage."
" "
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=HEADING, LINE=2, OBJECT=TEXT, ITEM=1,
   URL=http://www.ibi.com, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Regional Sales Report
Click here to access the IB homepage.

Category	Region	Unit Sales	Dollar Sales
Coffee	Midwest	332777	4178513
	Northeast	335778	4164017
	Southeast	350948	4415408
	West	356763	4473517
Food	Midwest	341414	4338271
	Northeast	353368	4379994
	Southeast	349829	4308731
	West	340234	4202337
Gifts	Midwest	230854	2883881
	Northeast	227529	2848289
	Southeast	234455	2986240
	West	235042	2977092

リンクをクリックすると、指定されたサイトがブラウザに表示されます。

WebFOCUS Server ドリルダウンプロシジャを実行する URL へのリンク

次のリクエストは、ブラウザセッションから開始され、WebFOCUS Reporting Server に格納されているドリルダウンレポートを実行します。

このプロシジャはブラウザから実行されます。この場合、現在のブラウザセッションのプロトコル、ホスト、ポートを使用してプロシジャが実行されるため、この例のドリルダウンは相対 URL (プロトコル、ホスト、ポートを含まない) として指定されています。

注意：この方法は、BI Portal プロシジャで WebFOCUS Server プロシジャへのドリルダウンを作成する場合に便利です。BI Portal は常にリポジトリ内のプロシジャを検索するため、FOCEXEC= を使用してドリルダウンプロシジャを実行する方法は機能しません。

メインプロシジャは次のとおりです。

```
TABLE FILE GGSALES
ON TABLE SET PAGE-NUM OFF
SUM UNITS AND DOLLARS
```

```
BY CATEGORY BY REGION
HEADING
"Regional Sales Report"
" "
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=DATA, COLUMN=REGION,
URL=/ibi_apps/WFServlet?(IBIF_ex='ggdrill' AREA=REGION IBIC_server='EDASERVE'
IBI_APPS='IBISAMP'),$
ENDSTYLE
END
```

ドリルダウンレポートは次のとおりです。このレポートは、アプリケーションの ibisamp に格納されている必要があります。

```
-DEFAULTS &REGION='$*';
TABLE FILE GGSALES
ON TABLE SET PAGE-NUM OFF
SUM UNITS AND DOLLARS
BY PRODUCT
WHERE REGION = '&AREA'
HEADING
"Sales Report for Region &AREA"
" "
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

メインレポートの出力結果は次のとおりです。

Regional Sales Report
Click a region to see a report for that region.

Category	Region	Unit Sales	Dollar Sales
Coffee	Midwest	332777	4178513
	Northeast	335778	4164017
	Southeast	350948	4415408
	West	356763	4473517
Food	Midwest	341414	4338271
	Northeast	353368	4379994
	Southeast	349829	4308731
	West	340234	4202337
Gifts	Midwest	230854	2883881
	Northeast	227529	2848289
	Southeast	234455	2986240
	West	235042	2977092

Northeast 地域をクリックすると、出力結果は次のようににになります。

Sales Report for Region Northeast

Product	Unit Sales	Dollar Sales
Biscotti	145242	1802005
Capuccino	44785	542095
Coffee Grinder	40977	509200
Coffee Pot	46185	590780
Croissant	137394	1670818
Espresso	68127	850107
Latte	222866	2771815
Mug	91497	1144211
Scone	70732	907171
Thermos	48870	604098

URL へのリンク使用上の注意

- URL 内でテキストとして解釈する特殊文字は、エンコードする必要があります。たとえば、URL 文字列にテキストとしてスラッシュ (/) を含める必要がある場合は、エンコードされた値「'%2F'」を使用します。たとえば、レポートのタイトルから URL にドリルダウンする場合 (例、'A' を値 '2009/03' で test.asp に渡す場合)、スタイルシートコマンドは、次のようになります。

```
TYPE=TITLE, URL=/test.asp?(A='2009%2F03'), $
```

ハイパーリンクの色の定義

HYPERLINK-COLOR 属性を使用して、レポート内のハイパーリンクの色を設定することができます。この設定は、レポート内で生成されたハイパーリンクのすべてに適用されます。レポート全体に単一色を定義することも、各要素に異なる色を定義することもできます。

ハイパーリンクの色の定義

```
TYPE = type, HYPERLINK-COLOR = color
```

説明

type

マクロを適用するレポートコンポーネントです。このキーワードをレポート全体に適用するには、TYPE=REPORT を使用します。この属性は、レポートの他の要素に個別に設定することもできます。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

color

スタイルシートでサポートされる任意の色指定値です。この構文で使用可能な色指定値についての詳細は、レポートの色値を参照してください。

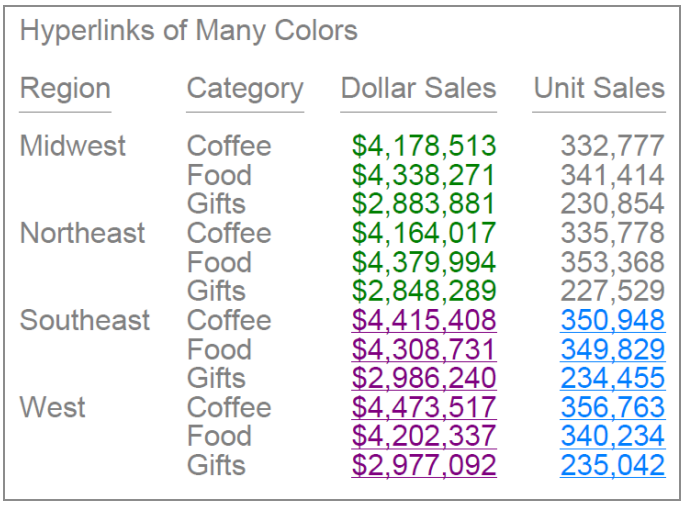
ハイパーリンクの色の定義

次の PDF リクエストは、レポート全体および個別要素のハイパーリンク色を定義する方法を示しています。

- レポート全体のデフォルトフォント色は灰色で、レポート全体のデフォルトハイパーリンク色は青色です。
- Dollar Sales 列 (DOLLARS) については、フォント色は緑色で、ハイパーリンク色は紫色です。
- Dollar Sales (DOLLARS) および Unit Sales (UNITS) の両方の列については、同一条件 (REGION GE 'O') を使用して、条件付きスタイルが適用されています。
- Unit Sales 列 (UNITS) については、条件付きスタイルと一致する場合、ハイパーリンク色はレポートのデフォルトハイパーリンク色 (青色) から継承されます。
- Dollar Sales 列 (DOLLARS) については、条件付きスタイルと一致する場合、ハイパーリンク色は紫色です。

```
TABLE FILE GGSALES
SUM DOLLARS/D12CM UNITS/D12C
BY REGION
BY CATEGORY
HEADING
"Hyperlinks of Many Colors"
""
ON TABLE SET PAGE-NUM OFF
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, SQUEEZE=ON, FONT=ARIAL, GRID=OFF, COLOR=GREY,
  HYPERLINK-COLOR='SLATE BLUE',$
TYPE=DATA, COLUMN=UNITS, WHEN=REGION GE 'O', URL='http://www.ibi.com',$
TYPE=DATA, COLUMN=DOLLARS, COLOR=GREEN, HYPERLINK-COLOR='PURPLE',$
TYPE=DATA, COLUMN=DOLLARS, WHEN=REGION GE 'O', URL='http://www.ibi.com',$
ENDSTYLE
END
```

出力結果は次のとおりです。



Hyperlinks of Many Colors

Region	Category	Dollar Sales	Unit Sales
Midwest	Coffee	$4,178,513	332,777
	Food	$4,338,271	341,414
	Gifts	$2,883,881	230,854
Northeast	Coffee	$4,164,017	335,778
	Food	$4,379,994	353,368
	Gifts	$2,848,289	227,529
Southeast	Coffee	$4,415,408	350,948
	Food	$4,308,731	349,829
	Gifts	$2,986,240	234,455
West	Coffee	$4,473,517	356,763
	Food	$4,202,337	340,234
	Gifts	$2,977,092	235,042

HYPERLINK-COLOR 使用上の注意

- デフォルト設定では、ドリルダウンリンクは、ハイパーリンク色の青色で下線付きで表示されます。
- HTML および DHTML レポートでは、レポートフォント色を指定した場合は、この色でドリルダウンのデフォルトフォント色が上書きされます。
- 標準レポートでは、スタイルシートの TYPE=REPORT 宣言を使用して HYPERLINK-COLOR 属性を設定します。
- 複合レポートでは、1 つ目のコンポーネントレポート (ページマスターの要素は除く) のスタイルシートの TYPE=REPORT 宣言を使用して、HYPERLINK-COLOR 属性を設定します。
- PPTX では、ハイパーリンク色は、PPTX スライドマスターのテーマの一部として保存されます。各リクエスト (レポートおよび複合レポート) で定義可能な HYPERLINK-COLOR 属性は 1 つのみです。

JavaScript 関数へのリンク

スタイルシートを使用して、任意のレポートコンポーネントから JavaScript 関数へのリンクを定義することができます。リンクの定義後、レポートコンポーネントを選択して JavaScript 関数を実行することができます。

プロシジャおよび URL へのドリルダウンリンクと同様に、レポートコンポーネントの値を JavaScript 関数へ渡すオプションパラメータを指定することができます。関数は、渡された値を使用して結果を動的に決定し、結果をブラウザに送信します。詳細は、パラメータの作成を参照してください。

注意：

- ここで、JavaScript 関数は別の JavaScript 関数を呼び出すことができます。
- JavaScript 関数を実行する場合、ターゲットフレームを指定することはできません。ただし、JavaScript 関数自体の結果としてターゲットフレームを指定することは可能です。

JavaScript 関数へのリンク

```
TYPE=type, [subtype], JAVASCRIPT=function[(parameters ...)], $
```

説明

type

リンクを実行するために、ユーザが Web ブラウザで選択するレポートコンポーネントを特定します。TYPE 属性とその値は、構文の先頭部分に配置する必要があります。

subtype

追加の属性です。COLUMN、LINE、ITEM などがあり、フォーマット設定対象のレポートコンポーネントを特定するために必要です。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

function

レポートコンポーネントの選択時に実行する JavaScript 関数を指定します。

すべての関連パラメータを含む JAVASCRIPT=*function* 引数は最大 2400 バイトの長さにすることができ、2 行以上に拡張することができます。単一引数を複数行に分割するには、1 つ目の行末に継続を示す円記号 (¥) を使用する必要があります。ブランクを区切り文字として使用している位置で引数を分割する場合、ブランクは、円記号 (¥) の前、または次の行の先頭に配置する必要があります。円記号 (¥) は区切り文字としては動作しません。

引数が適切に 2 行に分割されます。

```
JAVASCRIPT=myfunc(COUNTRY ¥
CAR MODEL 'ABC'),$
```

注意

- ダイアログマネージャの -HTMLFORM コマンドを使用して、レポートを関数定義した HTML ドキュメントに埋め込むことができます。
- -HTMLFORM で呼び出した HTML ドキュメントがある場合、拡張子は (.HTML ではなく) .HTM であることを確認します。

HTMLFORM コマンドについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

parameters

JavaScript 関数に渡される値です。詳細は、パラメータの作成を参照してください。

JavaScript 関数へのリンク

JavaScript 関数へのリンクに使用するレポート、およびスタイルシート構文は次のとおりです。実行された JavaScript 関数、およびブラウザに表示される結果も表示されています。

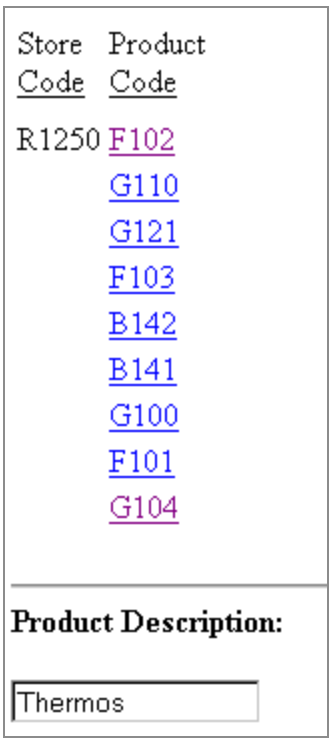
次のリクエストには、インラインスタイルシートが含まれています。

```
TABLE FILE GGORDER
SUM PRODUCT_ID
BY STORE_CODE
BY PRODUCT_DESCRIPTION NOPRINT
IF STORE_CODE EQ 'R1250'
ON TABLE HOLD AS JAVATEMP FORMAT HTMTABLE
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=DATA, COLUMN=PRODUCT_ID, JAVASCRIPT=showitem(PRODUCT),$
ENDSTYLE
END
-RUN
-HTMLFORM JAVAFORM
```

JAVAFORM.HTM ファイルには、JavaScript 関数が含まれています。

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
// This function will display the value in the text box
function showitem(string) {
document.form1.text1.value = string;
}
// End the hiding here
</SCRIPT>
</HEAD>
<BODY>
!IBI.FIL.JAVATEMP;
<HR>
<B>Product Description:</B>
<FORM NAME="form1">
<INPUT TYPE="text" NAME="text1" SIZE="16"> </FORM>
</BODY>
</HTML>
```

レポートプロシジャを実行すると、Web ブラウザに次のレポートが表示されます。Product Code リンクを選択すると、JavaScript 関数 ShowItem が実行され、レポート下部にあるフォーム内のテキストボックスに PRODUCT_DESCRIPTION フィールド (NOPRINT フィールド) の値が表示されます。たとえば、Product Code *G104* を選択すると、Product Description フィールドに「Thermos」が表示されます。



カスケードメニューおよびユーザ定義のスタイルを使用したマルチドリルダウン機能

マルチドリルダウン機能は、複数のメニュー項目および複数のカスケードレベルをサポートします。これらは、JavaScript を使用したすべての WebFOCUS レポート (例、HTML および DHTML) に組み込むことができます。メニューのスタイルは、WebFOCUS スタイルシート構文を使用してカスタマイズできます。

HTML および DHTML のマルチドリルダウンは、次の機能を提供します。

- 柔軟なカスケードメニュー
- メニュー内の各要素およびレベルで完全カスタマイズ可能なスタイル設定
- 選択したデータ要素に対して高度に配置されたメニュー

PDF、PPT、EXL2K、XLSX フォーマットでは、1 つ目のアクティブリンクを使用して、指定した場所にハイパーリンクを作成します。

アクセシビリティサポート

マルチドリルダウン機能は、HTML フォーマットで使用可能なカスケードマルチドリルダウンメニューで、セクション 508 アクセシビリティをサポートします。

レポートでマルチドリルダウンのカスケードメニューオプション、[オートドリルダウン] および [オートリンク] を使用する場合、アクセシビリティユーザは次のことを行う必要があります。

- 仮想 PC カーソルの設定をオフにします。
- キーボードを使用して、レポート内のリンクをナビゲートします。
- カスケードメニュー項目を開いた後、Enter キーを押して項目を表示します。
- 仮想 PC カーソルの設定をオンにして、レポート内をナビゲートします。

複数ドリルダウンリンクの作成

ドリルダウンメニューは、2 つのレベルでカスタマイズすることができます。

- 現在のプロシジャ (fex) 内のすべてのメニューに対するグローバルスタイル設定
- メニューの各エントリに対する項目レベルのスタイル設定

グローバルメニューのスタイル設定

現在のプロシジャ (fex) 内のすべてのメニューに対してスタイル属性を定義する場合、次の構文を使用します。

```
TYPE=REPORT, OBJECT=MENU, [FONT=font], [SIZE=size], [COLOR=color],
[HOVER-COLOR=hover_color], [BACKCOLOR=backcolor],
[HOVER-BACKCOLOR=hover_backcolor], [BORDER={ON|OFF|n}],
[BORDER-COLOR=border_color], [BORDER-STYLE=border_style]
$
```

説明

font

メニュー項目のフォントタイプフェースを定義します。デフォルト値は、レポートから継承されます。

size

メニュー項目のフォントサイズを定義します。デフォルトフォントサイズは 9 です。

color

メニュー項目のテキスト色を色名または RGB/HEX 値で定義します。メニュー項目のデフォルトテキスト色は、RGB(#6B6B6B) です。分割線 (SEPARATOR) およびコントロールで使用されるキャレット (<) の色の定義にも使用されます。

hover_color

メニュー項目にマウスポインタを置いた際のテキスト色、またはメニュー項目を選択した際のテキスト色を色名または RGB/HEX 値で定義します。これらのデフォルトテキスト色は、RGB(#495263) です。

backcolor

メニュー項目の背景色を色名または RGB/HEX 値で定義します。デフォルト背景色は、RGB(#F8F8F8) です。

hover_backcolor

メニュー項目にマウスポインタを置いた際の背景色、またはメニュー項目を選択した際の背景色を色名または RGB/HEX 値で定義します。これらのデフォルト背景色は、RGB (#DFDFDF) です。

BORDER={ON|OFF|n}

説明

- ON にすると、ユーザ定義のスタイル設定またはシステムデフォルト値に基づいて、メニューオブジェクト周囲の境界線および分割線が表示されます。
- OFF にすると、メニューオブジェクト周囲に境界線は表示されません。
- *n* は、境界線の太さを示すピクセル数です。有効な値は 1、2、3 で、それぞれ細線、標準、太線を示します。

デフォルトでは、細線に設定されています。

border_color

メニュー周囲の境界線に使用する境界線色を定義します。これは、メニュー内の分割線の色にも使用されます。定義済みの色です。デフォルト境界線色は、RGB(#D6D6D6) です。

border_style

メニュー周囲の境界線および分割線に使用する線スタイルを定義します。下表は、使用可能な値を示しています。これは、メニュー内の分割線のスタイルにも使用されます。デフォルト境界線スタイルは実線です。境界線スタイルの区別を明確にするため、太線 (例、border=heavy または border=3) の使用が必要な場合があります。

スタイル	説明
NONE	境界線/分割線なし
SOLID	実線
DOTTED	点線
DASHED	破線
DOUBLE	二重線

スタイル	説明
GROOVE	3D 溝線
RIDGE	3D 稜線
INSET	3D 凹線
OUTSET	3D 凸線

注意

- 複数ドリルダウンメニューに DRILL-SOURCE 属性が付加されている場合、この属性は、そのドリルダウンメニューが WebFOCUS によって生成され、ユーザが特定のレポート要素に追加した既存ドリルダウンに統合する必要があることを示しています。この属性値は、どの WebFOCUS 機能によってメニューが生成されたかを示すものです。この属性は、内部使用に限定された予約語です。
- 複数ドリルダウンリンクを作成する際は、最初の DRILLMENUITEM の前に単一ドリルダウン動作 (例、プロシジャ、URL) を指定することはできません。

メニュー項目のスタイル設定

カスケードメニューの構文は、既存のマルチドリルダウン (DRILLMENUITEM) 構文の拡張構文です。現在有効な構文はすべて、拡張構文の実装後に同様に動作します。

レポート領域またはデータ要素に付属の個別メニューおよび項目を定義する場合は、次の構文を使用します。

```
TYPE=type, [subtype], [DRILLMENUITEM='description', action|'keyword'],
  [NAME=name], [PARENT=parentname],
```

説明

type

リンクを実行するために、ユーザが Web ブラウザで選択するレポートコンポーネントを特定します。TYPE 属性とその値は、構文の先頭部分に配置する必要があります。

subtype

追加の属性です。COLUMN、LINE、ITEM などがあり、フォーマット設定対象のレポートコンポーネントを特定するために必要です。

DRILLMENUITEM の各項目には、description または keyword の組み合わせが必要です。action を伴わない description は、デフォルト設定で自動的に無効になります。

ただし、NAME/PARENT の組み合わせでリンクされた子エントリを含む親項目については例外です。この場合、action でカスケードメニューに子が含まれます。

description

レポート出力で、ドリルダウンオプションのポップアップメニューに表示されるテキストです。デフォルト値は ドリルダウン *n* で、*n* はドリルダウン 1、ドリルダウン 2 などの連続する整数です。

注意

- DRILLMENUITEM が、特別な値である 'SEPARATOR' に設定されている場合は、次のようになります。
  - 横方向の分割線が、メニュー内の指定した位置に、メニュー境界線で定義されたスタイル属性と色属性を使用して描画されます。
  - 分割線は、action と関連付けることはできません。
- DRILLMENUITEM の値はブランクにすることはできません。

action

リンクのタイプです。たとえば、詳細レポートまたは URL へのリンクです。

次の属性はオプションです。これらは、階層の定義が必要なカスケードメニューでのみ必要です。

name

親子項目間のリンクとして使用する現在の項目の一意の識別子です (オプション)。このノードが、リンクを特定する必要のある子メニュー項目の親として機能する場合のみ必要です。

parentname

現在の子項目の親メニュー項目の一意の識別子または名前です (オプション)。このノードが、階層内の別の項目の親として機能する場合のみ必要です。

ドリルダウンアクションのオプション

各ドリルダウンメニュー項目は、次のアクションの単一インスタンスにリンクすることができます。

FOCEXEC=report.fex

別のレポート。このスタイルシート属性は FOCEXEC です。

```
TYPE=type, [subtype], FOCEXEC=fex[(parameters...)], [TARGET=frame,]
[ALT='description',] $
```

URL=url string

URL。このスタイルシート属性は URL です。有効な URL を渡します。URL の長さは、ブラウザで許容される最大バイト数が上限になります。使用するブラウザでのこの制限についての詳細は、ブラウザのサポートサイトを参照してください。

```
TYPE=type, [subtype], URL=url[(parameters...)], [TARGET=frame,]
[ALT='description',] $
```

URL=(field)

フィールドの URL。このスタイルシート属性は URL です。有効な値の URL を持つレポートフィールド名を、リンクのジャンプ先に渡します。

```
TYPE=type, [subtype], URL=url[(parameters ...)], [TARGET=frame,]
[ALT='description',] $
```

JAVASCRIPT=function

JavaScript 関数。このスタイルシート属性は JAVASCRIPT です。

```
TYPE=type, [subtype], JAVASCRIPT=function[(parameters ...)], $
```

注意：ドリルダウンリンクが、ポップアップメッセージを表示する JavaScript 関数を呼び出す場合、ドリルダウンメニューは、ポップアップメッセージが閉じるまで開いた状態で保持されます。

ドリルダウンリンクの概要

マルチドリルダウンメニューのリンク先には次のものがあります。

- 別のレポート。このスタイルシート属性は FOCEXEC です。構文についての詳細は、別レポートへのリンクを参照してください。
- URL。このスタイルシート属性は URL です。有効な URL を渡します。構文についての詳細は、URL へのリンクを参照してください。

  URL の長さは、ブラウザで許容される最大バイト数が上限になります。使用するブラウザでのこの制限についての詳細は、ブラウザベンダーのサポートサイトを参照してください。
- フィールドの URL。このスタイルシート属性は URL です。有効な値の URL を持つレポートフィールド名を、リンクのジャンプ先に渡します。構文についての詳細は、URL へのリンクを参照してください。
- JavaScript 関数。このスタイルシート属性は JAVASCRIPT です。構文についての詳細は、JavaScript 関数へのリンクを参照してください。

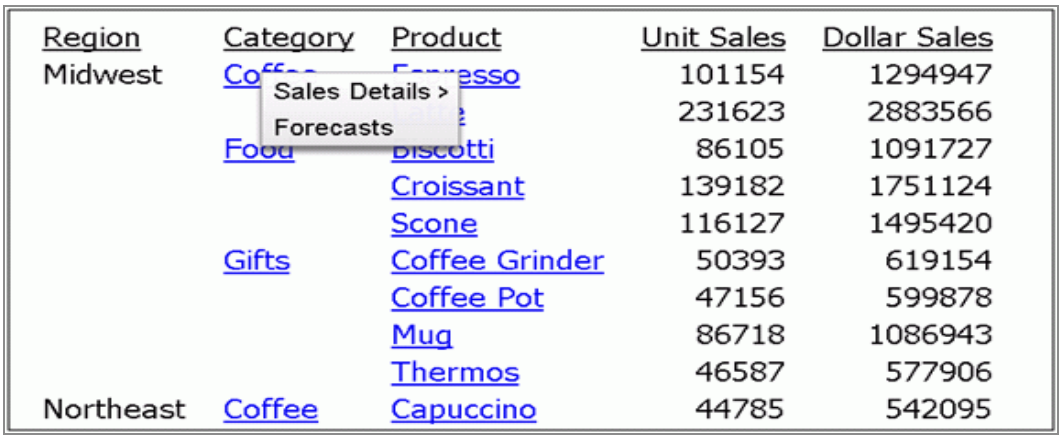
ドリルダウンメニューのスタイルシート構文の例

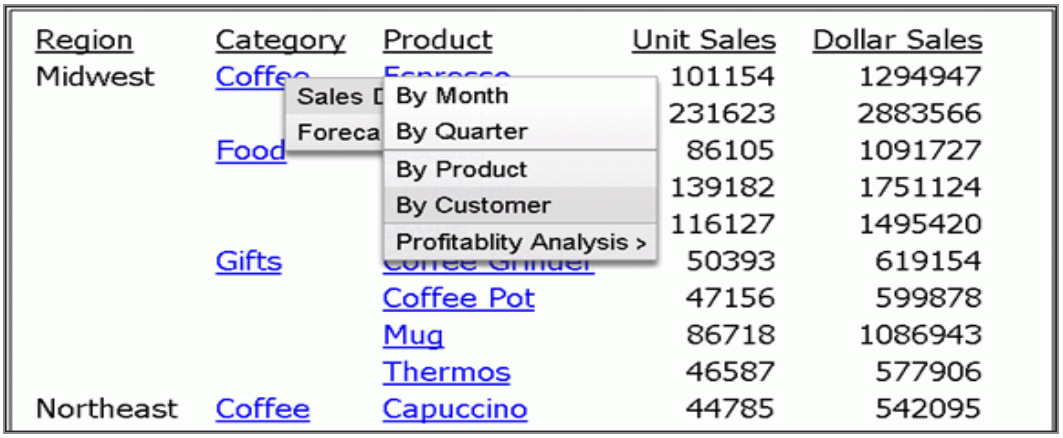
```
TABLE FILE GGSALES
SUM
   GGSALES.SALES01.UNITS
   GGSALES.SALES01.DOLLARS
BY GGSALES.SALES01.REGION
BY GGSALES.SALES01.CATEGORY
BY GGSALES.SALES01.PRODUCT
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE SET ASNAMES ON
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLEMBEDIMG ON
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
FONT=TAHOMA, GRID=OFF,$
```

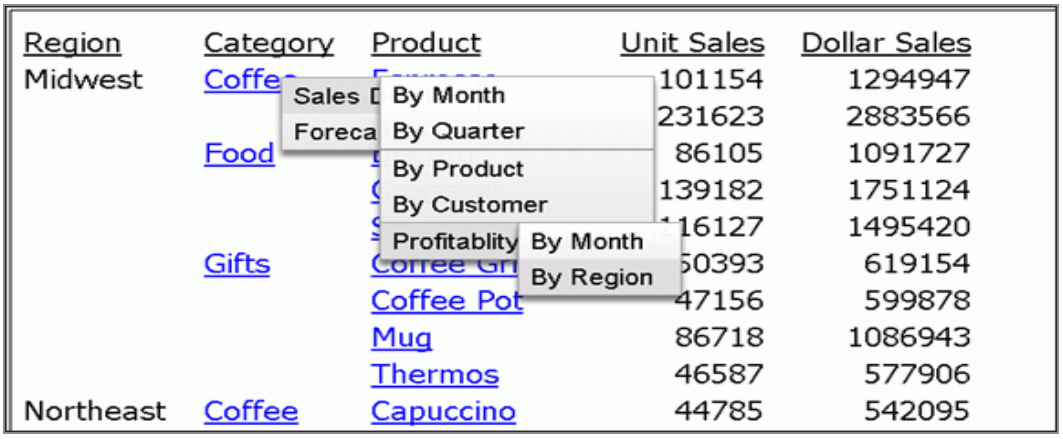
```
TYPE=DATA,COLUMN=B2,
  DRILLMENUITEM='Sales Details', NAME=menu2,
  DRILLMENUITEM='By Month',
    PARENT=menu2, NAME=menu21,
    FOCEXEC=detailreport.fex(PARAMETER=CATEGORY),TARGET=_blank,
  DRILLMENUITEM='By Quarter',
    PARENT=menu2, NAME=menu23,
    FOCEXEC=detailreport.fex(PARAMETER=CATEGORY),TARGET=_blank,
  DRILLMENUITEM=SEPARATOR, PARENT=menu2,
  DRILLMENUITEM='By Product',
    PARENT=menu2, NAME=menu24,
    FOCEXEC=detailreport.fex(PARAMETER=CATEGORY),TARGET=_blank,
  DRILLMENUITEM='By Customer',
    PARENT=menu2,NAME=menu25,
    FOCEXEC=detailreport.fex(PARAMETER=CATEGORY),TARGET=_blank,
  DRILLMENUITEM=SEPARATOR, PARENT=menu2,
  DRILLMENUITEM='Profitablity Analysis',
    PARENT=menu2,NAME=menu3,
  DRILLMENUITEM='By Month',
    PARENT=menu3, NAME=menu31,
    FOCEXEC=detailreport.fex(PARAMETER=CATEGORY),TARGET=_blank,
  DRILLMENUITEM='By Region',
    PARENT=menu3, NAME=menu32,
    FOCEXEC=detailreport.fex(PARAMETER=CATEGORY),TARGET=_blank,
  DRILLMENUITEM='Forecasts',
    FOCEXEC=detailreport.fex(PARAMETER=CATEGORY),TARGET=_blank,
$
TYPE=DATA,COLUMN=B3,
  DRILLMENUITEM='IBI Links',NAME=menu4a,
  DRILLMENUITEM='Information Builders',
    PARENT=menu4a, NAME=menu41,
    URL=http://www.ibi.com,TARGET=_blank,
  DRILLMENUITEM='Summit 2015',
    PARENT=menu4a, NAME=menu42,
    URL=http://www.ibi.com,TARGET=_blank,
  DRILLMENUITEM='Competative Analysis',
    PARENT=menu4a, NAME=menu43,
    URL=http://www.ibi.com,TARGET=_blank,
  DRILLMENUITEM='External Links', NAME=menu4b,
  DRILLMENUITEM='Google',
    PARENT=menu4b, NAME=menu45, URL=http://www.google.com,TARGET=_blank,
  DRILLMENUITEM='Weather',
    PARENT=menu4b, NAME=menu46, URL=http://www.weather.com,TARGET=_blank,
  DRILLMENUITEM='CNN',
    PARENT=menu4b,NAME=menu47, URL=http://www.cnn.com,TARGET=_blank,
$
ENDSTYLE
```

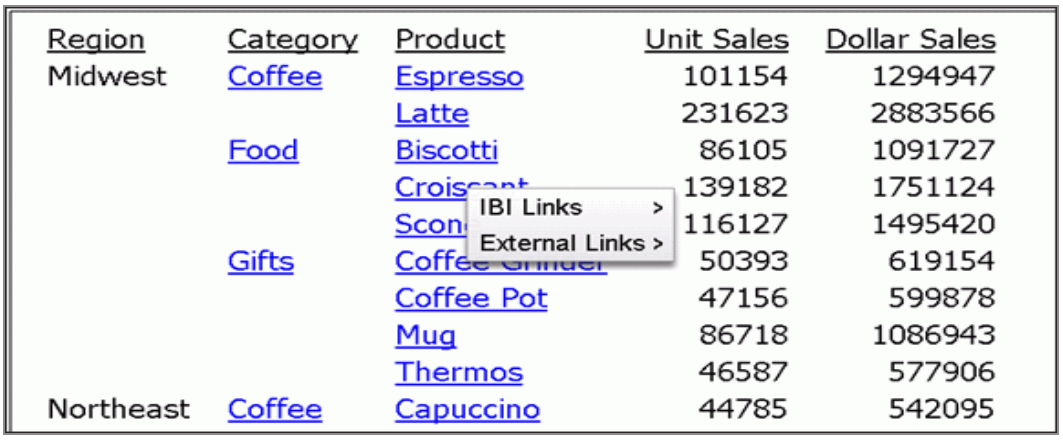
```
END
```

上記の構文は、下図のようなメニュー構造を生成します。





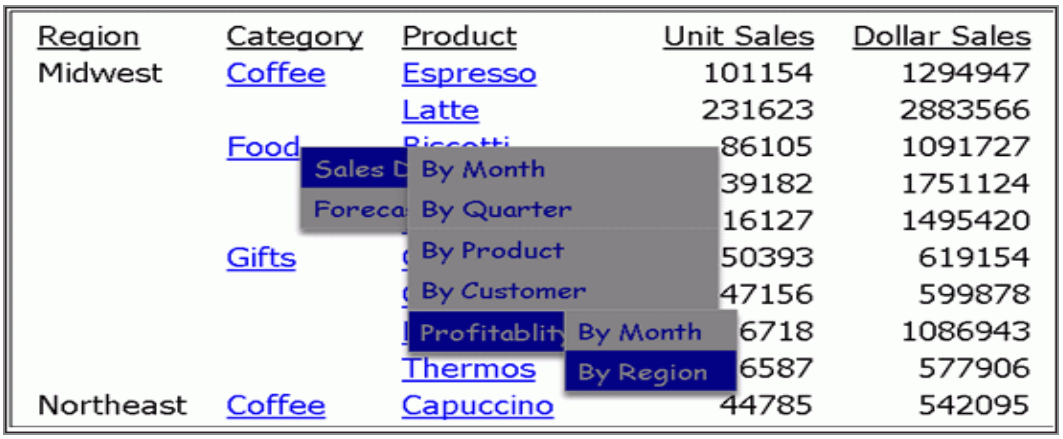


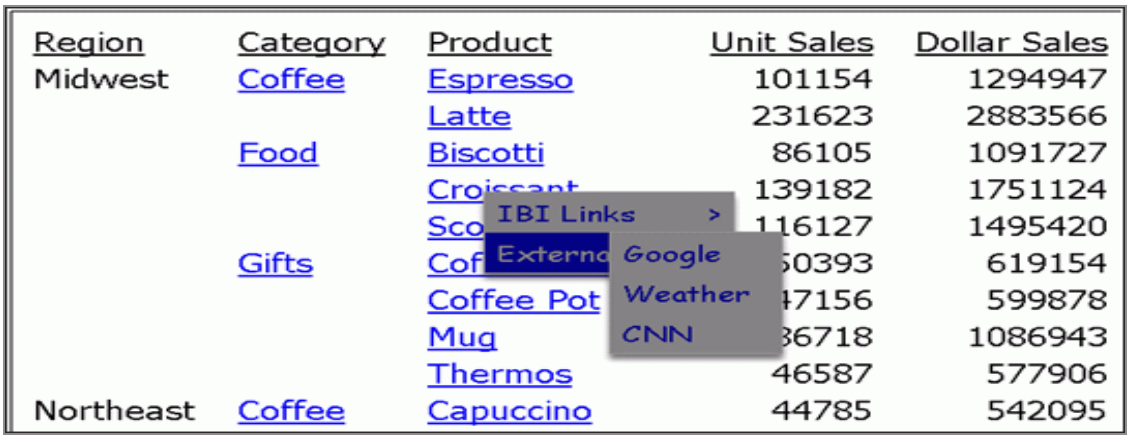


メニューにカスタムスタイルを適用するには、スタイルシートに次の構文を追加します。

```
TYPE=REPORT, OBJECT=MENU, FONT="COMIC SANS MS", COLOR=NAVY,
BACKCOLOR=GREY, HOVER-COLOR=GREY, HOVER-BACKCOLOR=NAVY, $
```

メニュー構造は、下図のよう表示されます。





マルチドリルダウンメニュー使用上の注意

ユーザ定義のスタイルを設定したマルチドリルダウンカスケードメニューでは、次のインタラクティブレポートがサポートされます。

- HFREEZE
- 行単位のアコーディオンレポート (EXPANDBYROW、EXPANDBYROWTREE)
- 列単位のアコーディオンレポート (EXPANDABLE)
- OLAP
- HTML TOC

注意：バージョン 8.2 SP01 では、レポートでマルチドリルダウン機能および Web ビューア機能が有効な場合、カスケードマルチドリルダウンメニューは表示されません。その代わり、バージョン 8.2 SP01 より前のバージョンで生成されるレガシーマルチドリルダウンメニューが生成されます。カスケードメニューは使用できず、すべてのハイパーリンクは、メニューの同一レベルに表示されます。

条件付きスタイル設定の適用

WHEN などの句を使用して条件付きスタイル設定をレポートコンポーネントに適用し、それを使用してレポートのフィールド値に基づくさまざまな異なる動作のうちの 1 つを選択することができます。

WHEN 条件は DRILLMENUITEM 構文の前に記述する必要があります。

条件の作成についての詳細は、条件付きリンクを参照してください。

条件付きスタイル設定の複数ドリルダウンレポートへの適用

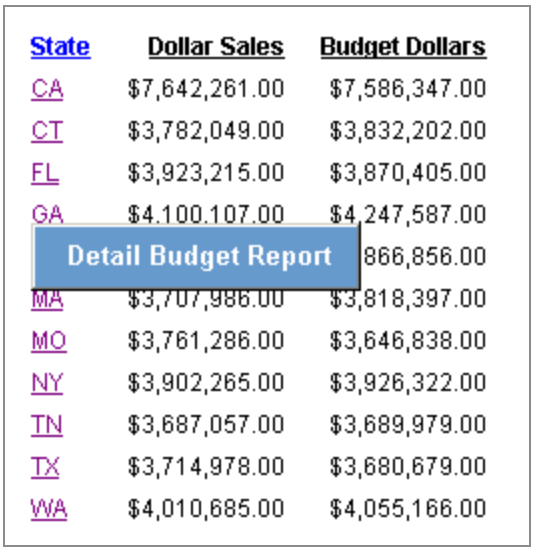
「複数ドリルダウンリンクの作成」概要レコードのサンプルに、次のコードの太字部分を追加します。要件どおりに、WHEN 条件が DRILLMENUITEM コードの前に記述されています。

概要レポートを実行すると、予算が売上を超える State フィールドは青ではなく赤になり、ドリルダウンオプションのポップアップメニューに、ドリルダウン 1 とドリルダウン 2 の代わりに「Detail Budget Report」と表示されます。

```
.
.
.
TYPE=DATA,
     COLUMN=N1,
     COLOR='BLUE',
     STYLE=UNDERLINE,
     DRILLMENUITEM='DrillDown 1',
          URL=http://www.informationbuilders.com?,
     DRILLMENUITEM='DrillDown 2',
          FOCEXEC=DETAILREPORT(PARAMETER=N1),$
```

```
TYPE=DATA,
     COLUMN=N1,
     COLOR='RED',
     STYLE=UNDERLINE,
     WHEN=BUDDOLLARS GT DOLLARS,
     DRILLMENUITEM='Detail Budget Report',
          FOCEXEC=DETAILREPORT(PARAMETER=N1),$.
.
.
```

出力結果のサンプルは次のとおりです。



パラメータの作成

ドリルダウンレポートが基本レポートの特定のデータ値に基づいている場合、ドリルダウン先のレポートに値を渡す 1 つまたは複数のパラメータを作成する必要があります。

パラメータは、動的リンクを作成する場合に役立ちます。たとえば、1 つ目のレポートが、特定日に会社により注文された商品の総数のリストを含む概要レポートであることを想定します。レポートの特定商品から、商品ベンダー名および注文番号により注文を受けた商品の個数を示す詳細レポートにドリルダウンすることができます。動的リンクを使用すると、複数の静的レポートの代わりに、1 つ目のレポートから渡された値を使用してどの情報を表示するかを決定するドリルダウンレポートが 1 つのみ作成されます。

複数のパラメータを作成することができます。パラメータの文字列をすべて括弧で囲み、それぞれをブランクで区別する必要があります。パラメータは、2400 バイト以内で指定する必要があります。

パラメータを作成するためには、スタイルシートで以下を組み合わせて使用することができます。次の方法があります。

- 定数値。
- フィールドの名前または位置。
- 値を渡す変数の名前。変数はインラインスタイルシートにのみ使用することができます。インラインスタイルシートについての詳細は、WebFOCUS スタイルシートの作成と管理を参照してください。

パラメータの作成

```
parameter=value
```

説明

parameter

リンクされたプロシジャの変数名です。

注意：競合を回避するために、Date、IBI、または WF で始まる変数は指定できません。これらの値で始まる変数名は、内部使用のための予約語です。

value

渡す値を指定します。使用できる値は次のとおりです。

'constant_value' - 渡す実際の値を指定します。この値は一重引用符 (') で囲む必要があります。

field - レポート内でプロシジャに渡す値を持つフィールドを特定します。フィールド名またはフィールドの位置を使用してフィールドを特定することができます。フィールドの位置についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

'&variable' - プロシジャに渡す値を持つ変数を指定します。変数名は一重引用符 (') で囲む必要があります。変数は、インラインスタイルシート内のみで使用することができます。

注意：変数は通常、定数値を渡すために使用します。変数が文字フィールドに相当する場合は、一重引用符 (') で囲む必要があります。以下はその例です。

```
'&ABC'.
```

パラメータ名および値全体を括弧で囲む必要があります。各 *parameter=value* の組はブランクで区切る必要があります。複数のパラメータをリクエストに含めることができますが、合計は 2400 バイトを超えることはできません。

注意：ドリルダウンレポートに、デフォルト値をメインレポートから渡されたものと同一の変数に設定する -DEFAULTS の記述が含まれる場合、渡された変数値により、ターゲットプロシジャの -DEFAULTS の記述が上書きされます。

定数値指定によるパラメータの作成

次の例は、定数値の指定によりパラメータを作成します。関連するスタイルシートは、リクエスト内で太字で表示されています。

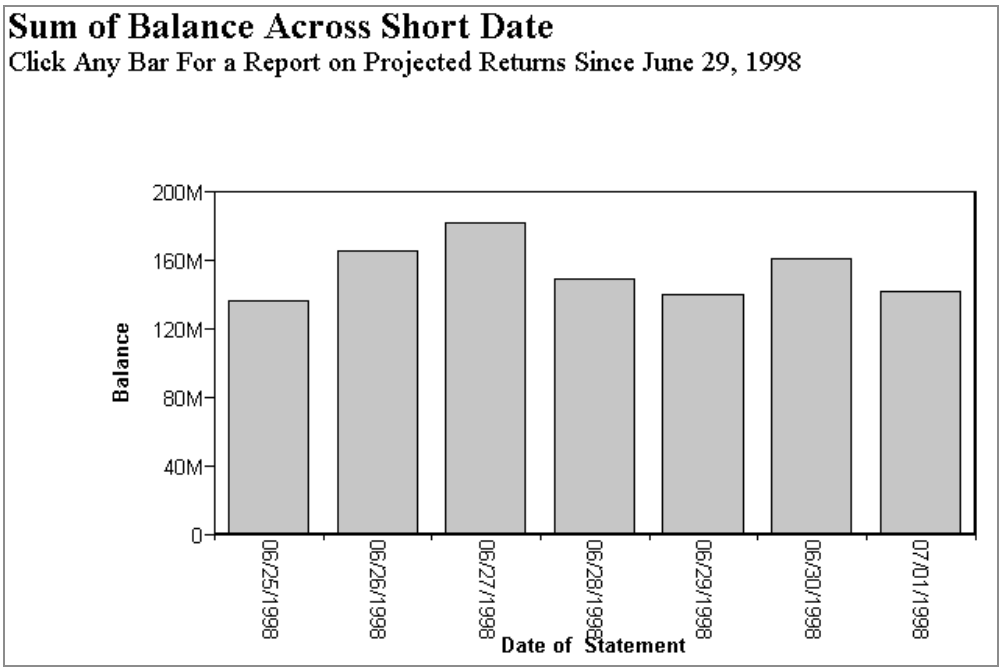
メインレポートは次のとおりです。

```
SET LOOKGRAPH BAR
SET 3D=OFF
GRAPH FILE SHORT
HEADING
"Sum of Balance Across Short Date"
"Click Any Bar For a Report on Projected Returns Since June 29, 1998 "
SUM BALANCE
ACROSS SHORT_DATE
ON GRAPH SET STYLE *
TYPE=DATA, ACROSSCOLUMN=N1,FOCEXEC=PROJRET(Short_Date='06291998'),$
ENDSTYLE
END
```

ドリルダウンレポート (PROJRET) は次のとおりです。

```
TABLE FILE SHORT
HEADING
"Projected Returns Since June 29, 1998 "
SUM PROJECTED_RETURN
BY SHORT_DATE
BY REGION
WHERE SHORT_DATE GE '&Short_Date';
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

メインレポートの出力結果は次のとおりです。



グラフの棒をクリックすると、次の出力結果が得られます。

```
Projected Returns Since June 29, 1998
```

Date of		Projected
Statement	Region	Annualized
		Return
06/29/1998	CENTRAL AMERICA	1.360

	EASTERN EUROPE	2.300
	FAR EAST	1.300
	MIDDLE EAST	1.140
	NORTH AMERICA	1.780
	SOUTH AMERICA	1.200
	WESTERN EUROPE	1.140
06/30/1998	CENTRAL AMERICA	1.360
	EASTERN EUROPE	2.350
	FAR EAST	1.300
	MIDDLE EAST	1.140
	NORTH AMERICA	1.780

	SOUTH AMERICA	1.200
	WESTERN EUROPE	1.140
07/01/1998	CENTRAL AMERICA	1.360
	EASTERN EUROPE	2.300
	FAR EAST	1.300
	MIDDLE EAST	1.140
	NORTH AMERICA	1.780
	SOUTH AMERICA	1.200
	WESTERN EUROPE	1.140

フィールドの指定によるパラメータの作成

次の例は、フィールド (CATEGORY) の指定によりパラメータを作成します。SALES ドリルダウンレポート (メインレポートにリンクされたレポート) により CATEGORY フィールドを &TYPE に設定します。基本レポートでは TYPE に CATEGORY フィールドが設定されます。

レポートを実行すると、CATEGORY (Coffee、Food、Gifts) フィールドの値はそれぞれの値に対して商品および地域の内訳を含むレポートにリンクされます。

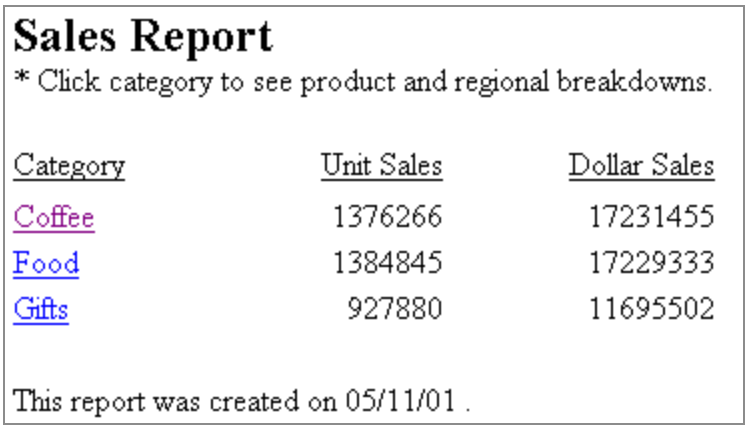
メインレポートは次のとおりです。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY CATEGORY
HEADING
"* Click category to see product and regional breakdowns."
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=DATA, COLUMN=CATEGORY, FOCEXEC=SALES (TYPE=CATEGORY), $
ENDSTYLE
FOOTING
"This report was created on &DATE ."
END
```

ドリルダウンレポート (SALES) は次のとおりです。

```
TABLE FILE GGSALES
ON TABLE SET PAGE-NUM OFF
SUM UNITS DOLLARS
BY CATEGORY BY PRODUCT
ACROSS REGION
WHERE CATEGORY = '&TYPE';
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

メインレポートの出力結果は次のとおりです。



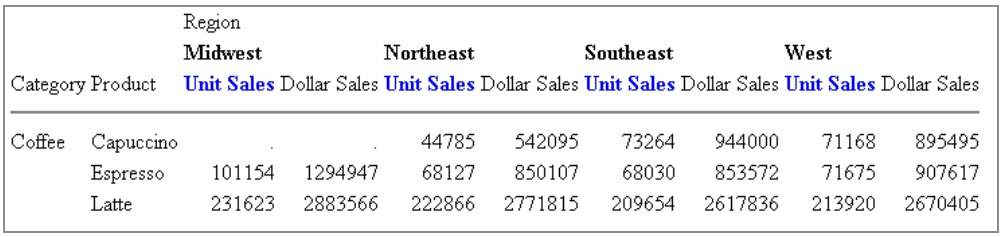
Sales Report

* Click category to see product and regional breakdowns.

Category	Unit Sales	Dollar Sales
Coffee	1376266	17231455
Food	1384845	17229333
Gifts	927880	11695502

This report was created on 05/11/01 .

「Coffee」をクリックすると、Coffee の商品および地域別の内訳が表示されます。



		Region							
		Midwest		**Northeast**		**Southeast**		**West**	
Category	Product	**Unit Sales**	Dollar Sales	**Unit Sales**	Dollar Sales	**Unit Sales**	Dollar Sales	**Unit Sales**	Dollar Sales
Coffee	Capuccino	.	.	44785	542095	73264	944000	71168	895495
	Espresso	101154	1294947	68127	850107	68030	853572	71675	907617
	Latte	231623	2883566	222866	2771815	209654	2617836	213920	2670405

変数の指定によるパラメータの作成

次のリクエストは、変数の指定によりパラメータを作成します。関連するスタイルシートは、リクエスト内で太字で表示されています。

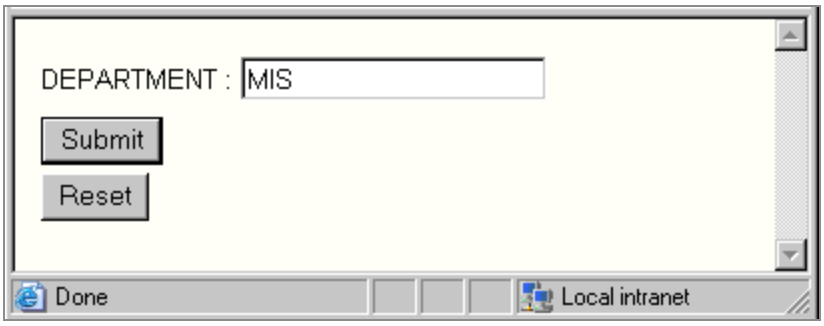
メインレポートは次のとおりです。

```
SET3D=OFF
GRAPH FILE EMPLOYEE
HEADING
"Salary Report Per Employee ID"
"Click A Bar For The List of Employees in the '&DEPARTMENT' Department"
SUM SALARY
ACROSS EMP_ID AS 'EMPLOYEE ID'
ON GRAPH SET STYLE *
TYPE=DATA, ACROSSCOLUMN=SALARY,
FOCEXEC=EMPBYDEP(DEPARTMENT='&DEPARTMENT'), $
ENDSTYLE
END
```

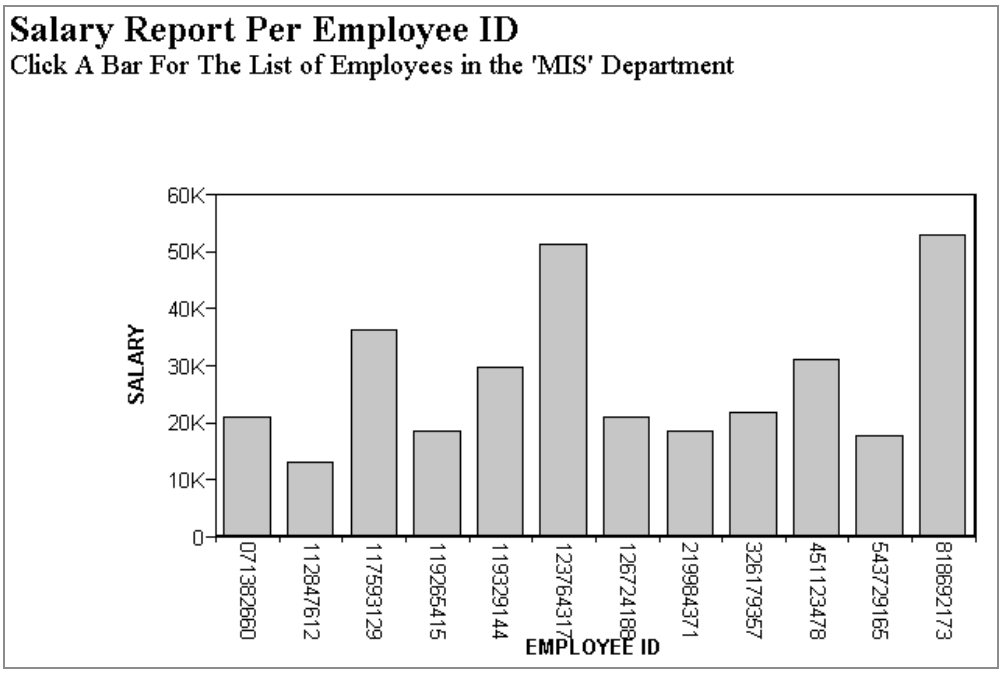
リンクされたレポート (EMPBYDEP) は次のとおりです。

```
TABLE FILE EMPLOYEE
HEADING
"List Of Employees in the '&DEPARTMENT' Department "
PRINT FIRST_NAME LAST_NAME
BY DEPARTMENT
WHERE DEPARTMENT EQ '&DEPARTMENT';
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

メインリクエストを実行すると、次のプロンプトが表示されます。



「MIS」と入力し、[実行] をクリックします。出力結果は次のとおりです。



グラフの棒をクリックすると、次の出力結果が得られます。

```
List Of Employees in the 'MIS' Department
```

DEPARTMENT	FIRST_NAME	LAST_NAME
MIS	MARY	SMITH
	DIANE	JONES
	JOHN	MCCOY
	ROSEMARIE	BLACKWOOD
	MARY	GREENSPAN
	BARBARA	CROSS

ドリルダウンリクエストでの DRILLMETHOD の使用

次の例は、ドリルダウンリクエストで DRILLMETHOD パラメータを使用して、ドリルダウンオプションを制御する方法を示しています。

- DRILLMETHOD を POST に設定すると、パラメータおよび値は URL に含められず、ログにも格納されません。この方法は、より安全性が高いメソッドです。
- DRILLMETHOD を GET に設定すると、パラメータと値は URL に含められ、ログに格納されます。

次のドリルダウンリクエストは、GGSALES データソースを使用し、SET DRILLMETHOD=POST コマンドおよび「CATEGORY」、「PRODUCT」という 2 つのパラメータが含まれています。

```
SET DRILLMETHOD=POST
TABLE FILE GGSALES
SUM DOLLARS BUDDOLLARS
BY CATEGORY
BY PRODUCT
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/ibi_themes/Warm.sty,$
TYPE=DATA, COLUMN=N1, TARGET='_blank', FOCEXEC=IBFS:/WFC/Repository/My_
Workspace/~admin/child_report.fex(PARA1=CATEGORY PARA2=PRODUCT), $
ENDSTYLE
END
-RUN
```

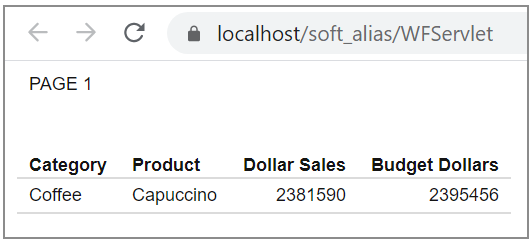
注意：DRILLMETHOD をスタイルシートコマンドの各ドリルダウンとして指定することもできます。以下はその例です。

```
TYPE=DATA, COLUMN=N1, DRILLMETHOD='POST', FOCEXEC=drilldown.fex, $
```

次のリクエストは、child_report の内容です。

```
TABLE FILE GGSALES
SUM DOLLARS BUDDOLLARS
BY CATEGORY
BY PRODUCT
WHERE CATEGORY EQ &PARA1.QUOTEDSTRING AND PRODUCT EQ &PARA2.QUOTEDSTRING;
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/ibi_themes/Warm.sty,$
ENDSTYLE
END
```

DRILLMETHOD=POST のドリルダウンレポートを実行し、カテゴリ (例、Coffee) を選択すると、下図のように、パラメータおよび値は URL には含められません。



DRILLMETHOD=GET のドリルダウンレポートを実行し、カテゴリ (例、Coffee) を選択すると、下図のように、パラメータおよび値が URL に含めらます。

注意：SET DRILLMETHOD コマンドについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』の「環境のカスタマイズ」を参照してください。

複数パラメータの使用

複数のパラメータを使用する場合は、文字列全体を括弧で囲み、それぞれをブランクで区切る必要があります。関連するスタイルシートは、リクエスト内で太字で表示されています。

メインレポートは次のとおりです。

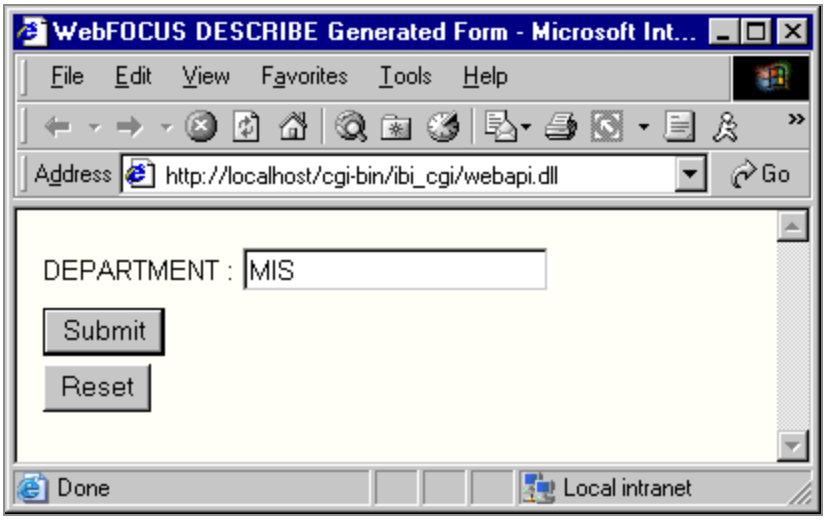
```
SET 3D=OFF
GRAPH FILE EMPLOYEE
SUM CURR_SAL
ACROSS DEPARTMENT
ON GRAPH SET STYLE *
TYPE=DATA, ACROSSCOLUMN=CURR_SAL,
FOCEXEC=REPORT2 (DEPARTMENT='&DEPARTMENT' LAST_NAME='SMITH'), $
ENDSTYLE
END
```

ドリルダウンレポート (REPORT2) は次のとおりです。

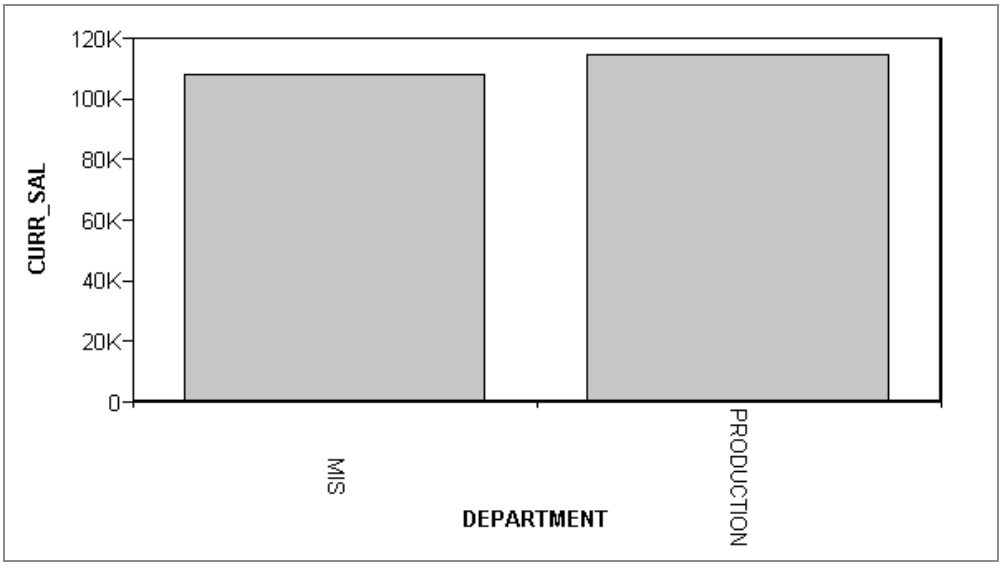
```
TABLE FILE EMPLOYEE
PRINT SALARY
BY DEPARTMENT
BY FIRST_NAME
BY LAST_NAME
WHERE DEPARTMENT EQ '&DEPARTMENT'
```

```
WHERE LAST_NAME EQ '&LAST_NAME'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

メインリクエストを実行すると、次のプロンプトが表示されます。



「MIS」と入力し、[実行] をクリックします。出力結果は次のとおりです。



MIS の棒をクリックすると、次の出力結果が得られます。

DEPARTMENT	FIRST NAME	LAST NAME	SALARY
MIS	MARY	SMITH	$13,200.00

条件付きリンク

レポートまたはグラフからレポート、URL、JavaScript 関数へリンクするときは、条件を作成することができます。たとえば特定部門の現在給与のみを表示する場合について考察します。これを行うには、WHEN 条件を作成します。

WHEN についての詳細は、レポートフォーマット設定の管理を参照してください。

注意：条件付きリンクは、GRAPH リクエストではサポートされません。

条件付きリンク

レポートへの条件付きリンクを指定するには、次の構文を使用します。

```
TYPE=type, [subtype], FOCEXEC=fex[(parameters...)],
   WHEN=expression,[TARGET=frame,] $
```

URL への条件付きリンクを指定するには、次の構文を使用します。

```
TYPE=type, [subtype], URL=url[(parameters...)],
   WHEN=expression,[TARGET=frame,] $
```

JavaScript 関数への条件付きリンクを指定するには、次の構文を使用します。

```
TYPE=type, [subtype], JAVASCRIPT=function[(parameters...)],
   WHEN=expression,[TARGET=frame,] $
```

説明

type

リンクを実行するために、ユーザが Web ブラウザで選択するレポートコンポーネントを特定します。TYPE 属性とその値は、構文の先頭部分に配置する必要があります。

subtype

追加の属性です。COLUMN、LINE、ITEM などがあり、フォーマット設定対象のレポートコンポーネントを特定するために必要です。レポートコンポーネントの特定についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

fex

レポートコンポーネントを選択するときに、実行されるリンク先プロシジャのファイル名を指定します。別のプロシジャへのリンクについての詳細は、別レポートへのリンクを参照してください。

url

任意の有効な URL、または URL を持つレポートフィールド名を特定します。フィールド名は括弧で囲みます。URL へのリンクについての詳細は、URL へのリンクを参照してください。

function

レポートコンポーネントの選択時に実行する JavaScript 関数を指定します。JavaScript 関数の呼び出しについての詳細は、JavaScript 関数へのリンクを参照してください。

parameters

レポート、URL、または JavaScript 関数へ渡される値です。詳細は、パラメータの作成を参照してください。

expression

COMPUTE 式の右辺で有効にする任意のブール式です。

注意：IF... THEN... ELSE は、WHEN 句では必要なく、サポートされていません.WHEN 式内の数値以外のリテラルは、すべて一重引用符 (') 内で指定される必要があります。

frame

ドリルダウンリンクからの出力が表示される Web ページのターゲットフレームを指定します。詳細は、ターゲットフレームの指定を参照してください。

条件付きリンク

次の例では、DEPARTMENT フィールドの MIS 値のみを REPORT3 にリンクさせます。そのためには、スタイルシートに WHEN=DEPARTMENT EQ 'MIS' 句を含めます。関連するスタイルシートは、リクエスト内で太字で表示されています。

メインレポートは次のとおりです。

```
TABLE FILE EMPLOYEE
SUM CURR_SAL AS 'Total,Current,Salaries'
BY DEPARTMENT AS 'Department'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=DATA, COLUMN=N1, FOCEXEC=REPORT3(DEPARTMENT=N1),
     WHEN=DEPARTMENT EQ 'MIS', $
ENDSTYLE
END
```

ドリルダウンレポート (REPORT3) は次のとおりです。

```
TABLE FILE EMPLOYEE
PRINT SALARY
BY DEPARTMENT
BY LAST_NAME
WHERE DEPARTMENT EQ '&DEPARTMENT'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

次の出力結果では、MIS 部門のみにリンクが設定されます。

Department	Total Current Salaries
MIS	$108,002.00
PRODUCTION	$114,282.00

[MIS] をクリックすると、次の出力結果が得られます。

DEPARTMENT	LAST NAME	SALARY
MIS	BLACKWOOD	$21,780.00
	CROSS	$27,062.00
		$25,755.00
	GREENSPAN	$9,000.00
		$8,650.00
	JONES	$18,480.00
		$17,750.00
	MCCOY	$18,480.00
	SMITH	$13,200.00

グラフィックイメージからのリンク

HTML レポートでは、イメージからレポートまたはプロシジャへリンクすることができます。このイメージをレポート全体、レポートの見出しまたは脚注 (ページ見出し、ページ脚注、中間見出し、中間脚注を含む) に添付することができます。

グラフィックイメージからのリンク構文は、レポートコンポーネントからのリンク構文と同様に指定します。唯一の相違点は、IMAGE=*image* のスタイルシートへの追加です。

注意：HTML フォーマットの使用時には、イメージをレポートまたはプロシジャにのみリンク可能です。

グラフィックイメージからのリンクの指定

イメージからのリンクをレポートまたはプロシジャで指定するには、次の構文を使用します。

```
TYPE=type, [subtype], IMAGE=image, FOCEXEC=fex
   [(parameters ...)],[TARGET=frame,] $
```

URL のイメージからのリンクを指定するには、次の構文を使用します。

```
TYPE=type, [subtype], IMAGE=image, URL=url
   [(parameters ...)],[TARGET=frame,] $
```

JavaScript 関数のイメージからのリンクを指定するには、次の構文を使用します。

```
TYPE=type, [subtype], IMAGE=image, JAVASCRIPT=function
   [(parameters ...)],$
```

説明

type

リンクを実行するために、ユーザが選択するレポートコンポーネントを指定します。TYPE 属性とその値は、構文の先頭部分に配置する必要があります。指定できるコンポーネントのタイプは次のとおりです。

REPORT - レポート全体に添付するグラフィックイメージからのドリルダウンが可能になります。

TABHEADING または TABFOOTING - レポートの見出しまたは脚注に添付するグラフィックイメージからのドリルダウンが可能になります。

HEADING または FOOTING - ページの見出しまたは脚注に添付するグラフィックイメージからのドリルダウンが可能になります。

SUBHEAD または SUBFOOT - ページの中間見出しまたは中間脚注に添付するグラフィックイメージからのドリルダウンが可能になります。

レポートコンポーネントについての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

subtype

追加の属性です。COLUMN、LINE、ITEM などがあり、フォーマット設定対象のレポートコンポーネントを特定するために必要です。レポートコンポーネントの特定についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

image

グラフィックイメージのファイル名を指定します。イメージは、ブラウザがサポートするフォーマットで別のグラフィックファイルとして存在する必要があります。ほとんどのブラウザでは、GIF および JPEG ファイルタイプがサポートされます。

ローカルのイメージファイルを指定することも、URL を使用してネットワーク上の任意の場所にあるイメージファイルを指定することもできます。URL は、絶対 URL で指定するか (例、http://www.ibi.com/graphic.gif)、Application Server や Web サーバへのパスを識別できる相対エイリアスで指定します(例、/ibi_apps/ibi_html/ibi_logo.gif)。

代わりに、値がイメージファイル名と一致する文字フィールド (BY フィールドまたは表示フィールドのいずれか) をレポートに指定することもできます。スタイルシートを使用して、レポートでグラフィックイメージを統合、配置する方法についての詳細は、レポートページのレイアウトを参照してください。

fex

ユーザがレポートコンポーネントを選択するときに実行される、リンクされたプロシジャのファイル名を指定します。別のプロシジャへのリンクについての詳細は、別レポートへのリンクを参照してください。

url

任意の有効な URL、または URL を持つレポートフィールド名を特定します。フィールド名は括弧で囲みます。URL へのリンクについての詳細は、URL へのリンクを参照してください。

function

ユーザがレポートコンポーネントを選択したときに実行する JavaScript 関数を指定します。JavaScript 関数の呼び出しについての詳細は、JavaScript 関数へのリンクを参照してください。

parameters

レポート、URL、または JavaScript 関数へ渡される値です。1 つまたは複数のパラメータを渡すことができます。パラメータの文字列をすべて括弧で囲み、それぞれをブランクで区別する必要があります。詳細は、パラメータの作成を参照してください。

frame

ドリルダウンリンクからの出力が表示される Web ページのターゲットフレームを指定します。詳細は、ターゲットフレームの指定を参照してください。

注意：JavaScript 関数を実行する場合、ターゲットフレームを指定することはできません。ただし、JavaScript 関数自体の結果としてターゲットフレームを指定することは可能です。

イメージからのリンクの指定

次の例は、イメージからレポートにリンクを設定する方法を示しています。関連するスタイルシートは、リクエスト内で太字で表示されています。

メインレポートは次のとおりです。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME BY EMP_ID
HEADING
"List Of Employees By Employee ID"
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=HEADING, STYLE=BOLD, $
TYPE=REPORT, GRID=OFF, $
TYPE=REPORT,
IMAGE=E:¥IBI¥WEBFOCUS81¥APPS¥IBINCCEN¥IMAGES¥LEFTLOGO.GIF,
     FOCEXEC=IMAGE-D, $
ENDSTYLE
END
```

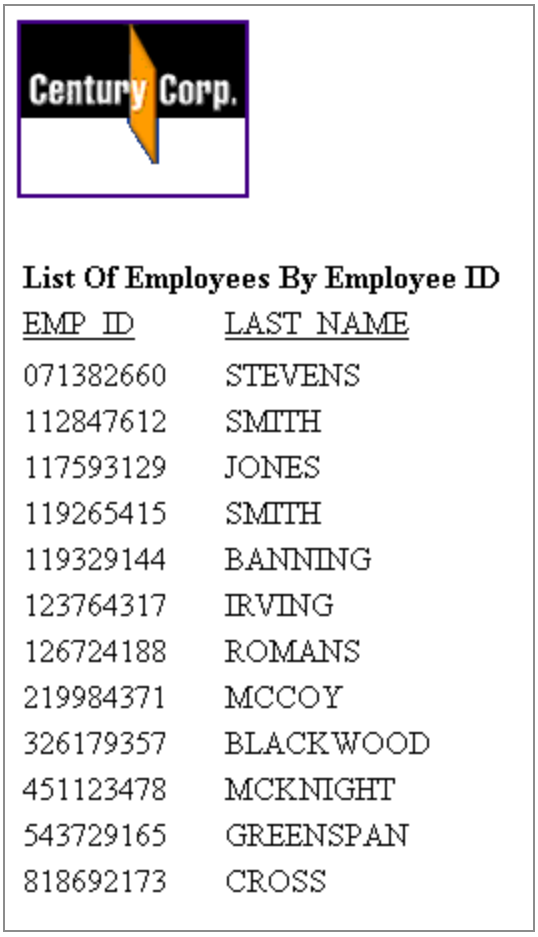
注意：IBINCCEN ディレクトリには、英語版のサンプルが格納されています。

ドリルダウンレポート (IMAGE-D) は次のとおりです。

```
TABLE FILE EMPDATA
PRINT SALARY
BY DIV
WHERE DIV LE 'CORP';
ON TABLE SET PAGE-NUM OFF
```

```
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

メインレポートの出力結果は次のとおりです。



このイメージをクリックすると次の出力結果が表示されます。

DIV	SALARY
CE	$62,500.00

	$54,100.00
	$25,400.00
	$115,000.00
	$33,300.00
	$25,000.00
	$49,000.00
	$40,900.00
	$43,000.00
	$45,000.00
CORP	$55,500.00
	$83,000.00
	$32,000.00

	$62,500.00
	$79,000.00
	$35,200.00
	$62,500.00
	$26,400.00

ベース URL の指定

ファイル、イメージ、および Java ファイルへリンクするための完全な物理 URL が分からない場合は、ブラウザが相対 URL を検索するためのデフォルトパスを指定することができます。

デフォルトの URL を指定するには、SET BASEURL コマンドを使用します。SET BASEURL を使用すると、WebFOCUS が生成する HTML ファイルに <BASE HREF="*url*"> が追加されます。レポートが実行される際は、生成された Web ページが呼び出す HTML ファイル、グラフィックファイル、および Java アプレット CLASS ファイルの検索に、指定したディレクトリが使用されます。

ベース URL の指定

```
SET BASEURL=url
```

説明

url

アプリケーションが HTML ドキュメントを作成した際に指定した相対 URL で、ブラウザはこの URL をデフォルトパスとして検索を実行します。

URL は http:// で開始し、末尾の区切り文字 (/) で終了する必要があります。

ベース URL の指定

次の例は、ベース URL の指定方法を示しています。

```
SET BASEURL=http://host[:port]/ibi_apps/ibi_html/
```

説明

host

WebFOCUS Web アプリケーションの展開先のホスト名です。

port

WebFOCUS Web アプリケーションの展開先のポート番号です。ポート番号はデフォルト値以外を使用している場合に入力します。

指定のベース URL に格納されているグラフィックイメージをレポートに含める場合、URL すべてを入力する代わりに、スタイルシートに次の構文を追加します。

```
TYPE=HEADING, IMAGE=ib_logo.gif, ..., $
```

注意：URL がリモート Web サイトにある場合、検索に長時間かかることがあります。可能な限り、WebFOCUS システムにグラフィックイメージファイルを保存します。

ターゲットフレームの指定

フレームを使用することにより、アプリケーションの HTML ページをスクロール可能なセクションに分割することができます。リンクを使用して別のページに移動する方法と異なり、フレームを使用すると、セクション内をスクロールすることでページ上のさまざまな情報項目の閲覧が可能になります。レポートコンポーネントからレポートプロシジャまたは URL へのリンクを定義するときは、ドリルダウンリンクの結果を Web ページのターゲットフレームに表示するよう指定することができます。

ターゲットフレームの指定には 2 つの方法があります。次の方法があります。

- TARGET 属性を使用したスタイルシートのターゲットフレーム。スタイルシートを使用して、レポートまたはグラフを表示する Web ページのターゲットフレームに、そのドリルダウンリンクを表示するよう指定することができます。ただし、スタイルシートでターゲットフレームを指定すると、生成される各 HREF に HTML 構文が追加されます。

  注意：レポートキャンバスでターゲットフレームを指定する場合、スタイルシートに直接追加したコマンドは認識されません。レポートキャンバスは、このツール以外で生成されたコマンドを削除します。

- SET コマンドによるデフォルトターゲットフレーム。SET TARGETFRAME により、WebFOCUS が表示する HTML ファイルの見出しに「<BASE TARGET="*framename*">」という HTML コードが追加されます。ベースレポートまたはグラフからのドリルダウンリンクは、スタイルシートの TARGET 属性により上書きされない限り、すべて指定のフレームへ設定されます。

TARGET 属性または SET TARGETFRAME コマンドを使用するには、Web ページで複数フレームを作成する必要があります。

注意：JavaScript 関数を実行する場合、ターゲットフレームを指定することはできません。ただし、JavaScript 関数自体の結果としてターゲットフレームを指定することは可能です。

ターゲットフレームの指定

レポートまたはプロシジャでターゲットフレームを指定するには、次の構文を使用します。

```
TYPE=type, [subtype], FOCEXEC=fex[(parameters ...)], [TARGET=frame,] $
```

URL のターゲットフレームを指定するには、次の構文を使用します。

```
TYPE=type, [subtype], URL=url[(parameters ...)], [TARGET=frame,] $
```

説明

type

リンクを実行するために、ユーザが Web ブラウザで選択するレポートコンポーネントを指定します。TYPE 属性とその値は、構文の先頭部分に配置する必要があります。

subtype

追加の属性です。COLUMN、LINE、ITEM などがあり、フォーマット設定対象のレポートコンポーネントを特定するために必要です。レポートコンポーネントの特定についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

fex

ユーザがレポートコンポーネントを選択するときに実行される、リンクされたプロシジャのファイル名を指定します。別のプロシジャへのリンクについての詳細は、別レポートへのリンクを参照してください。

url

任意の有効な URL、または URL を持つレポートフィールド名を特定します。フィールド名は括弧で囲み、値はリンクのジャンプ先として有効な URL です。URL へのリンクについての詳細は、URL へのリンクを参照してください。

parameters

プロシジャまたは URL に渡される値です。1 つまたは複数のパラメータを渡すことができます。値の文字列全体を括弧で囲み、それぞれをブランクで区切る必要があります。詳細は、パラメータの作成を参照してください。

frame

ドリルダウンリンク (プロシジャまたは URL) からの出力が表示される Web ページのターゲットフレームを指定します。

ターゲットフレーム名にブランクが埋め込まれている場合、名前が引用符で囲まれていなくても、正確に解析されます。以下はその例です。

```
TYPE=DATA, COLUMN=N1,
FOCEXEC=MYREPORT, TARGET=MY FRAME, $
```

ターゲットフレームの名前「MY FRAME」は正確に解析されます。

使用可能な標準 HTML フレーム名は _blank、_self、_parent、_top です。

デフォルトターゲットフレームの指定

```
SET TARGETFRAME=frame
```

説明

frame

ドリルダウンリンク (プロシジャまたは URL) からの出力が表示される Web ページのターゲットフレームを指定します。

ターゲットフレームの指定

デフォルトのターゲットフレームは次のように指定します。

```
SET TARGETFRAME=_SELF
```

次の例は、リクエストでのターゲットフレームの指定方法を示しています。リクエストの太字の部分が、この例に関連するスタイルシートです。

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL
BY DEPARTMENT
ON TABLE SET STYLE *
TYPE=DATA, COLUMN=N1, URL=http:¥¥www.informationbuilders.com,
     TARGET=_SELF, $
ENDSTYLE
END
```

複合レポートの作成

複合レポートを作成すると、複数のレポートが 1 つのファイルに統合されます。これにより、スタイルフォーマット (例、PDF、DHTML、EXL2K、XLSX) を使用して作成された複数レポートの連結が可能になります。複合レポート内に、イメージとして保存したグラフを含むイメージファイルも埋め込むことができます。

複合レポートには、次の 3 種類があります。

- **従来の複合レポート (単純に「複合レポート」)** この複合レポートでは、個々のレポートやグラフが連続的に配置されて単一出力ファイルに統合されます。このタイプの複合レポートの機能は安定しています。このドキュメントには、上位互換性を保持するための構文が含まれています。この手法を使用するアプリケーションは、引き続き動作します。

- **複合レイアウトレポート**　複合レイアウトレポートは、埋め込みまたは外部からの、個々のレポートコンポーネントやグラフコンポーネントで構成されます。レポートおよびグラフは、ページ上の任意の位置に配置できます。複数のレポートを組み合わせて特定のページに表示したり、新しいページにオーバーフローするレポートの処理方法を指定したりできます。
- **統合された複合レイアウトレポート**　統合された複合レイアウトレポートでは、共通のソートフィールドを含むすべてのレポートおよびグラフが、個々のページレイアウトに分割 (バースト) されます。共通のソートフィールドの値ごとにページが生成され、ソート値に対して取得されたデータが各ページのすべてのコンポーネントに表示されます。統合された複合レイアウトレポートを作成するには、複合レイアウトレポートの SECTION 宣言で MERGE=ON を指定します。

  統合された複合レイアウトレポートでは、特定のソートフィールド値のデータがコンポーネントのいずれかに含まれている場合、一部のコンポーネントのデータが欠落している場合でもそのソートフィールド値のページが生成されます。

レポートの縦方向にはクエリで生成されたデータ行のすべてが常に表示されますが、レポートの幅は、定義されたコンポーネントコンテナのサイズにより制限されます。つまり、複合レポートではマルチパネル表示はサポートされません。ただし、複合レポート以外の PDF ドキュメントではサポートされます。複合レポート以外の PDF ドキュメントでは、レポートデータの幅が定義済みページサイズより長い場合、パネル (横方向のオーバーフローページ) が自動的に生成されます。このパネル機能は、複合 PDF ドキュメントではサポートされません。そのため、レポートを正しく生成するには、すべての複合コンポーネントを定義済みコンテナ幅に収める必要があります。コンテナのサイズは、レポートのタイプごとに定義します。

- 従来の複合レポートの構文では、コンポーネントレポートのいずれかが定義済みページ幅に収まらない場合、レポートの実行が中断され、マルチパネル表示がサポートされないことを示すエラーメッセージが表示されます。
- 複合レイアウトレポートの構文では、コンポーネントが定義済みコンテナサイズ幅に収まらない場合、コンテナ内でレポートのコンテンツが折り返されます。コンテナのサイズは、複合レポートの構文で、コンポーネントの POSITION と DIMENSIONS パラメータの両方を組み合わせて定義します。

ドキュメント構文による複合レイアウトレポートの作成

通常、複合レイアウトレポートは、ドキュメントキャンバスのオプションを使用して作成します。別の方法として、ドキュメント構文を任意のテキストエディタで編集することで、複

合レイアウトレポートを作成することもできます。

複合レイアウトレポートの構文は、複合レイアウトブロックという構造で定義され、レポートに先行して単一ブロック内にレイアウト情報のすべてが配置されます。このブロックは、COMPOUND LAYOUT 宣言で開始し、END で終了します。このブロック内の言語は、スタイルシート構文に基づいており、スタイルシートパーサにより解析されます。

この解析は、スタイルフォーマット (例、PDF、DHTML、EXL2K、XLSX) でサポートされます。

ヒント：スタイルシート構文についての詳細は、WebFOCUS スタイルシートの作成と管理を参照してください。

複合レイアウトブロックは、SECTION、PAGELAYOUT、COMPONENT 宣言で構成されます。複合レイアウトブロック構文の一般的な構造は次のとおりです。

```
COMPOUND LAYOUT PCHOLD AS filename FORMAT format
```

```
SECTION
    PAGELAYOUT
      COMPONENT
      COMPONENT
    ...
    PAGELAYOUT
      COMPONENT
      COMPONENT
    ...
  ...
END
...
COMPOUND END
```

COMPOUND LAYOUT 宣言の注意事項　使用可能な複合レイアウト出力フォーマットは、PDF、DHTML、PowerPoint、AHTML、Excel、FLEX、APDF です。選択した複合レイアウトフォーマットは、各コンポーネントで指定されたレポート出力フォーマットより優先されます。出力ファイル名は、COMPOUND ブロックで AS filename 句を使用して定義することができます。出力ファイル名を定義しない場合、最初のコンポーネントレポートの ON TABLE HOLD 句からファイル名が取得されます。

END は COMPOUND LAYOUT ブロックの終了を表し、COMPOUND END は複合レポートの終了を表します。

さらに、各コンポーネントの後に SET COMPONENT=report(n) 構文を追加し、その後にレポートを生成する実際の WebFOCUS コードを続けます。

SECTION 宣言および構文

複合レポートセクション (SECTION 宣言) では、複合レポート内の各コンポーネントレポートをグループ化します。現在の機能では単一セクションで指定されたレポートのみがサポートされますが、この構造を使用することで、より複雑なレポートがサポートされます。複合レイアウトレポートを作成する場合、SECTION 宣言は必須です。

SECTION 構文は次のとおりです。

```
SECTION=section-name, LAYOUT=ON, [MERGE=ON|OFF,]
 [UNITS=IN|CM|PTS,] [PAGESIZE=size,] [ORIENTATION=PORTRAIT|LANDSCAPE,]
 [LEFTMARGIN=m,] [RIGHTMARGIN=m,] [TOPMARGIN=m,] [BOTTOMMARGIN=m,] $
```

説明

section-name

セクションの一意の識別子です。最大長は 16 バイトです。

LAYOUT=ON

このセクションで複合レイアウトを使用することを指定します。

注意：現時点で適用可能なオプションは LAYOUT=ON のみです。

MERGE={ON|OFF}

このセクションで複数のレポートを最初の BY フィールドの値に基づいて統合するかどうかを指定します。

注意：デフォルト値は OFF です。

m

マージン (LEFT、RIGHT、TOP、BOTTOM) をインチ、センチメートル、またはポイントで指定します。

SECTION 宣言にオプション項目として UNITS、PAGESIZE、ORIENTATION、または MARGIN を指定した場合、これらの項目が、コンポーネントレポート、グローバル SET コマンド、ON TABLE SET コマンド、スタイルシートキーワードで設定されたパラメータより優先されます。

PAGELAYOUT 宣言

SECTION 宣言は 1 つまたは複数の PAGELAYOUT 宣言で構成されます。各 PAGELAYOUT 宣言で複数の COMPONENT 宣言がグループ化され、これらのコンポーネントがセクションの特定のページに配置されます。

PAGELAYOUT キーワードにより、このキーワードから次の PAGELAYOUT キーワードまたはセクション末尾までの一連の COMPONENT 宣言がグループ化されます。

PAGELAYOUT 構文は次のとおりです。

```
PAGELAYOUT={n|ALL},
 [TOPMARGIN=m,] [BOTTOMMARGIN=m,]$
```

説明

{n|ALL}

複数ページレイアウトのどのページにコンポーネントを表示するかを指定します。値の (n) には、現在のセクション内のページ番号を指定するか、ALL を使用してセクションの各ページにコンポーネントを表示するよう指定します。

- PAGELAYOUT の値には、1 から始まる番号が付けられます。たとえば、複合レポートをページの両面に印刷する場合、表側のコンポーネントレポートには PAGELAYOUT 1、裏側には PAGELAYOUT 2 と指定します。

  注意：レポートに単一ページのみを含める場合でも構文は必要です。

  ```
  PAGELAYOUT=1, $
  ```

- PAGELAYOUT=ALL 構文を指定すると、各ページにコンポーネントが表示されます。この構文は、ページ見出しやページ脚注を生成するコンポーネントに使用すると便利です。

  ```
  PAGELAYOUT=ALL, $
  ```

m

フローレポートの境界 (TOP、BOTTOM) を単位で定義します。

ページマスター

PAGELAYOUT=ALL の宣言に含まれているコンポーネントは、レポート出力の各ページに表示されます。この構文は、複合レポート出力のデザインテーマの作成に使用すると便利です。

- ページマスターにはデフォルトのレポートコンポーネントを含める必要がありますが (COMPONENT が必須の構文要素のため)、そのレポートコンポーネントにデータを表示することはできません (見出し内を除く)。レポートコンポーネントの見出しにデータまたはグラフィックスを表示する場合、各ページに見出しを配置するためのブランク領域を確保し、他のコンポーネントをページ上の領域に配置する際に見出しと重なり合わないよう注意する必要があります。

  たとえば、次のレポートではデータは表示されません。

  ```
  SET COMPONENT='DfltCmpt1'
  TABLE FILE SYSCOLUM
  SUM TBNAME NOPRINT
  IF READLIMIT EQ 1
  ON TABLE SET PREVIEW ON
  ON TABLE SET PAGE-NUM NOLEAD
  END
  ```

- 統合された複合レポートのページマスターには、複合レイアウトレポート内の他のコンポーネントで使用されている最初の BY フィールドを含める必要があります。統合された複合レポートでページマスターのデフォルトコンポーネントを作成する場合、最適な方法は複合レイアウトレポートのいずれかで使用されているデータソースの 1 つを使用することです。たとえば、PAGELAYOUT1 のコンポーネントレポートが GGSALES データソースに対して実行され、最初の BY フィールドが PCD の場合、ページマスターのデフォルトコンポーネントを次のように指定することができます。

  ```
  SET COMPONENT='DfltCmpt1'
  TABLE FILE GGSALES
  SUM UNITS NOPRINT
  BY PCD NOPRINT
  IF READLIMIT EQ 1
  ON TABLE SET PREVIEW ON
  ON TABLE SET PAGE-NUM NOLEAD
  END
  ```

  ページマスターでテキストやイメージを繰り返し表示するデザインテーマを作成する場合は、ページマスター上に描画オブジェクトを配置する方法をお勧めします。

- ページマスターのサイズおよびページ方向のスタイル設定は、デフォルトのドキュメ

ントレベルにのみ適用されます。ドキュメント内でページ方向の縦と横が混在している場合、ページマスター要素によるサイズおよび位置の自動変更は実行されません。ドキュメント内でページ方向の縦と横を混在させる必要がある場合は、各ページレイアウトでそれぞれのページ方向を定義することができます。この場合、ページ方向の縦と横が切り替わるごとにスタイル設定が正しく適用されるように、スタイル設定の要素を各ページレイアウトに個別に適用する必要があります。

COMPONENT 宣言

COMPOUND LAYOUT ブロック内の COMPONENT 宣言の順序は、コンポーネントレポートの実行順序に一致させる必要があります。また、COMPONENT 宣言はコンポーネントレポートごとに指定する必要があります。

コンポーネントには、固定とフローの 2 種類があります。固定コンポーネントはページの DIMENSION パラメータで定義されたコンテナ内に配置され、追加データが存在する場合は、次ページにオーバーフローして前ページと同一の位置に同一サイズで配置されます。オーバーフローページの固定コンポーネントのサイズと位置はカスタマイズすることができます (OVERFLOW-POSITION および OVERFLOW-DIMENSION パラメータの使用)。フローコンポーネントでは、データが定義済みコンポーネントの最上部から下ページマージンまで連続して表示され、続いて次のデータがオーバーフローページの上ページマージンからデータの末尾まで連続して表示されます。各タイプのコンポーネントの開始位置は、POSITION パラメータで定義します。

また、RELATIVE-TO、RELATIVE-POINT、POSITION-POINT キーワードを使用して、開始位置を別のコンポーネントとの相対的な位置として指定することもできます。この場合、コンポーネントの位置 (POSITION) の相対座標は、コンポーネントが存在する位置、相対関係の対象、コンポーネントの終端に基づいて決定されます。相対コンポーネントがフローコンポーネントの場合は、RELATIVE-TO キーワードとともに REQUIRED-SPACE キーワードを使用することで、現在のコンポーネントの開始位置を指定することもできます。

固定またはフローのどちらのコンポーネントを使用するかの選択は、DIMENSION パラメータで指定します。固定コンポーネントの場合、DIMENSION パラメータで境界ボックスのサイズを指定します。フローコンポーネントの場合、DIMENSION パラメータで指定するサイズにアスタリスク (* *) を使用します。

COMPONENT 構文は次のとおりです。

```
COMPONENT=component-name, TYPE=component-type,
    POSITION=(x y), DIMENSION=(xsize ysize),
    [OVERFLOW-POSITION=(x y),] [OVERFLOW-DIMENSION=(xsize ysize),]
    [RELATIVE-TO=relative_component_name,]
```

```
    [RELATIVE-POINT=relative-value, POSITION-POINT=position-value,
      REQUIRED-SPACE=required-space-value,]
    [DRILLMAP=((L1 targetreport)),]  $
```

説明

component-name

コンポーネントを識別する一意の名前を最大 16 バイトで指定する必要があります。この名前は、後からリクエストでコンポーネントレポートを指定する際に使用します (同一プロシジャまたは呼び出し先プロシジャで使用)。レポートコンポーネントは、SET COMPONENT=component-name 構文にこの名前を使用することで識別されます。

注意：この SET 構文は複合レポート内のスタイル設定されたレポートのみに影響するため、ここで指定するレポートに先行するスタイル未設定レポートの前にこの構文を配置することができます。たとえば、抽出ファイルを生成するレポートがこれに該当します。

component-type

宣言するコンポーネントのタイプを指定します。現在、REPORT のみがサポートされています。

POSITION=(x y)

コンポーネントの左上端を配置するページ上 x y 座標を指定します。すべての座標は、UNIT で指定されている単位 (デフォルト値はインチ) で指定します。(0 0) は、物理ページの左上端を表します。

注意：デフォルト設定の座標は、物理ページの絶対位置を表します。x または y の前にプラス符号 (+) またはマイナス符号 (-) を追加すると (例、 (+ 0.25 +0))、座標はページマージンの上左端との相対位置になります。RELATIVE-TO キーワードが存在する場合、相対座標は、このキーワードで宣言された RELATIVE-TO で指定されるオブジェクトを基準に設定されます。

DIMENSION=(xsize ysize)

コンポーネントの境界ボックスのサイズを、UNIT で指定された単位で指定します。

固定コンポーネントの場合、xsize および ysize にはサイズを数値で指定する必要があります。

フローコンポーネントの場合、DIMENSION = (* *) のように、xsize と ysize の両方にアスタリスク (*) を指定する必要があります。

OVERFLOW-POSITION=(x y) and OVERFLOW-DIMENSION=(xsize ysize)

これらのオプション項目では、最初の境界ボックスからオーバフローした場合に、後続ページに表示する位置とサイズを指定します。

OVERFLOW-POSITION および OVERFLOW-DIMENSION は、フローコンポーネントでもサポートされます。以下はその例です。

```
COMPONENT='report2', TEXT='report2', TOC-LEVEL=2,
POSITION=(0.5 1.7), DIMENSION=(* *), OVERFLOW-POSITION=(0.5 1.7),
OVERFLOW-DIMENSION=(* *),
```

RELATIVE-TO=relative_component_name

現在のコンポーネントの POSITION との相対座標の基準とする別のコンポーネントを指定します。

この相対コンポーネントは、位置が固定されているか、現在のコンポーネントより前に生成済みの必要があります。

RELATIVE-POINT=relative-value and POSITION-POINT=position-value

オプションのパラメータです。現在のコンポーネント上のどの点が、相対コンポーネント上のどの点に対して相対的に配置されているかを指定します (例、現在のコンポーネントの TOP-LEFT が相対コンポーネントの BOTTOM-LEFT に対して相対的に配置)。

次の値の任意の組み合わせに設定することができます。

- TOP-LEFT
- TOP-CENTER
- TOP-RIGHT
- CENTER-LEFT
- CENTER
- CENTER-RIGHT
- BOTTOM-LEFT
- BOTTOM-CENTER
- BOTTOM-RIGHT

REQUIRED-SPACE=required-space-value

現在のコンポーネントを開始するために、相対コンポーネントの終了点からページ上に残す必要最小スペースを指定します。相対コンポーネントの終了点からの残りのスペースが十分でない場合、現在のコンポーネントは次のページから開始されます。

次のいずれかの値です。

- (*x y*) の組み合わせ。現在のコンポーネントを開始する前、相対コンポーネントの後にページに残す水平、垂直方向の最小スペースを、現在の測定単位で指定します。
- AUTO。相対コンポーネントの後にページに残すスペースを自動的に計算し、現在のコンポーネント全体が、このスペースに納まるかどうかを特定します。このスペースに収まらない場合は、現在のコンポーネントは、次のページから開始されます。

DRILLMAP=((L1 targetreport))

リンク ID およびこのレポートのハイパーリンクからドリルスルーするターゲットレポートを指定します。現時点では、リンク ID は L1 で十分です。

注意：二重括弧で囲む必要があります。

ドキュメント構文による複合レイアウトレポートの作成

この単純な例では、GGSALES マスターファイルを使用し、MERGE キーワードを指定して、統合された複合レイアウトレポートを生成します。各コンポーネントの最初の BY フィールドが REGION のため、REGION の値が変わるたびに新しいページが生成され、最初のレポート (Sales) が (1 1) の位置に、2 つ目のレポート (Units) が (6.25 1) の位置に配置されます。

テキストエディタで、次の構文を入力します。

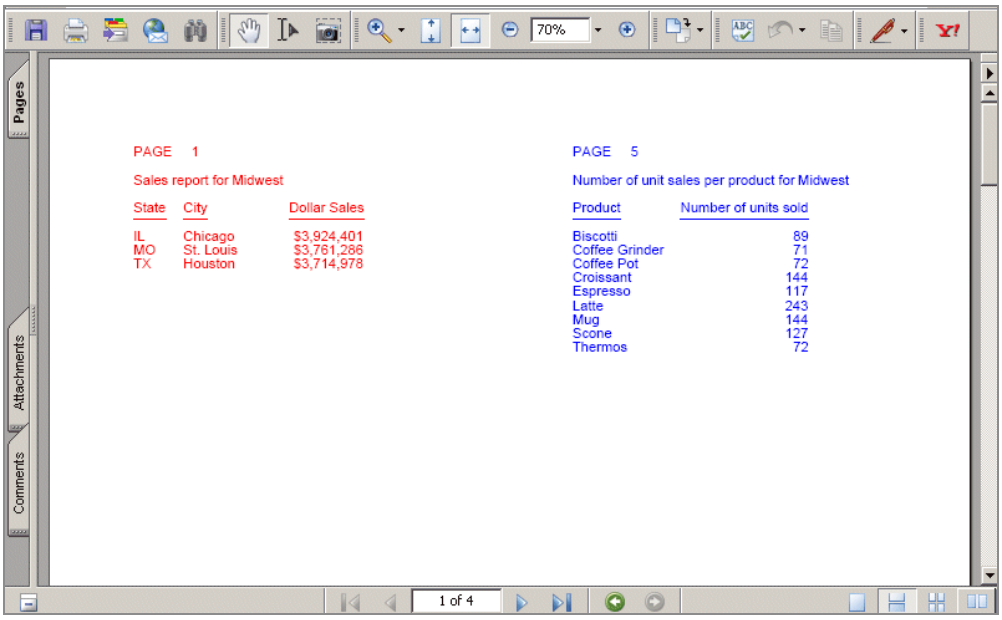
```
SET PAGE-NUM=OFF
COMPOUND LAYOUT PCHOLD FORMAT PDF
SECTION=S1, LAYOUT=ON, MERGE=ON, ORIENTATION=LANDSCAPE, $
PAGELAYOUT=1, $
COMPONENT=Sales, TYPE=REPORT, POSITION=(1 1), DIMENSION=(4 4), $
COMPONENT=Units, TYPE=REPORT, POSITION=(6.25 1), DIMENSION=(4 4), $
END
SET COMPONENT=Sales
TABLE FILE GGSALES
"Sales report for <REGION"
```

```
" "
SUM DOLLARS/F8M
BY REGION NOPRINT
BY ST
BY CITY
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, COLOR=RED, SQUEEZE=ON, $
END
SET COMPONENT=Units
TABLE FILE GGSALES
"Number of unit sales per product for <REGION"
" "
SUM CNT.UNITS AS 'Number of units sold'
BY REGION NOPRINT
BY PRODUCT
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, COLOR=BLUE, SQUEEZE=ON, $
ENDSTYLE
END
COMPOUND END
```

次の構文例は、R1 および R2 で指定されたコンポーネントレポートを、既存のプロシジャで作成する方法を示します。前例の Sales のコンポーネントレポートを REPORT1.FEX、Units のコンポーネントレポートを REPORT2.FEX として保存しておきます。次の構文例を実行すると、前例と同一の複合レイアウトレポートが表示されます。

```
SET PAGE-NUM=OFF
SET SQUEEZE=ON
COMPOUND LAYOUT PCHOLD FORMAT PDF
SECTION=S1, LAYOUT=ON, MERGE=ON, ORIENTATION=LANDSCAPE, $
PAGELAYOUT=1, $
COMPONENT=R1, TYPE=REPORT, POSITION=(1 1), DIMENSION=(4 4), $
COMPONENT=R2, TYPE=REPORT, POSITION=(6 1), DIMENSION=(4 4), $
END
SET COMPONENT=R1
EX REPORT1
SET COMPONENT=R2
EX REPORT2
COMPOUND END
```

次は出力結果の 1 ページ目です。



ドキュメント構文による統合されたグラフの作成

この例では、GGSALES マスターファイルを使用し、グラフとレポートを含む、統合された複合レイアウトレポートを生成します。ここでは、前述の例の最初のレポートをグラフに置き換えています。GRAPH リクエストには 2 つの BY フィールドが使用されています。最初の BY フィールド (REGION) の値が変わるたびに新しいグラフが作成されます。これらのファイルには、HOLD ファイル名に連続番号を追加した名前が付けられます。たとえば、HOLD0.SVG、HOLD1.SVG のような名前になります。

これらのグラフを、統合された複合レイアウトレポートのコンポーネントとしてレポートに配置するには、各グラフファイルの名前 (例、HOLD0.SVG、HOLD1.SVG) を作成するために複数の COMPUTE コマンドが必要になります。また、COMPUTE コマンドにより、イメージファイルが TABLE リクエストの HEADING に追加されます。これにより、各イメージファイルがそれぞれの生成元である REGION の同一値に関連付けられます。

テキストエディタで、次の構文を入力します。

```
SET PAGE-NUM=OFF
COMPOUND LAYOUT PCHOLD FORMAT PDF
SECTION=S1, LAYOUT=ON, MERGE=ON, ORIENTATION=LANDSCAPE, $
PAGELAYOUT=1, $
COMPONENT=Sales, TYPE=REPORT, POSITION=(0.25 1), DIMENSION=(4 4), $
```

```
COMPONENT=Fuel, TYPE=REPORT, POSITION=(7.25 1), DIMENSION=(4 4), $
END
```

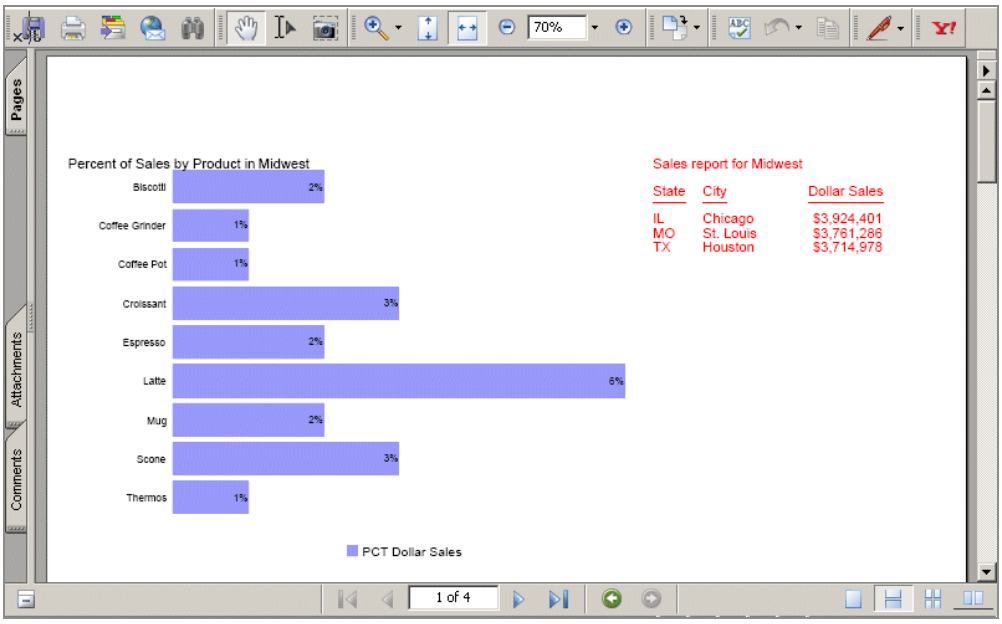
```
 SET COMPONENT=Sales
 GRAPH FILE GGSALES
 SUM PCT.DOLLARS
 BY REGION NOPRINT
 BY PRODUCT
 ON GRAPH SET LOOKGRAPH HBAR
 ON GRAPH HOLD AS HOLD FORMAT SVG
 ON GRAPH SET GRAPHSTYLE *
 setPlace(true);
 setColorMode(1);
 setDepthRadius(0);
 setDepthAngle(0);
 setDisplay(getO1MajorGrid(),false);
 setTransparentBorderColor(getFrame(),true);
 setDisplay(getDataText(),true);
 setTextFormatPreset(getDataText(),28);
 setFontSizeAbsolute(getDataText(),true);
 setFontSizeInPoints(getDataText(),9);
 setPlaceResize(getDataText(),0);
 setFontStyle(getDataText(),0);
 setTransparentBorderColor(getSeries(0),true);
 setDisplay(getO1AxisLine(),false);
 setFontSizeAbsolute(getO1Label(),true);
 setFontSizeInPoints(getO1Label(),9);
 setPlaceResize(getO1Label(),0);
 setFontSizeAbsolute(getY1Label(),true);
 setFontSizeInPoints(getY1Label(),9);
 setPlaceResize(getY1Label(),0);
 setTextFormatPreset(getY1Label(),28);
 setGridStyle(getY1MajorGrid(),3);
 setDisplay(getY1AxisLine(),false);
 setDisplay(getY1MajorGrid(),false);
 setDisplay(getY1Label(),false);
 setDataTextPosition(3);
 setTextString(getO1Title(),"");
 setTextString(getY1Title(),"");
 setFontStyle(getTitle(),0);
 ENDSTYLE
 END
```

```
 TABLE FILE GGSALES
 "Percent of Sales by Product in <REGION"
```

```
" "
SUM
COMPUTE CNTR/I4 = CNTR + 1; NOPRINT
COMPUTE CNTR2/A4 = IF &FOCGRAPHCNT EQ 1 THEN ' ' ELSE FTOA(CNTR-1,'
(F4)','A4'); NOPRINT
COMPUTE IMG/A16 = 'HOLD'||LJUST(4,CNTR2,'A4')|| '.svg'; NOPRINT
BY REGION NOPRINT
ON REGION PAGE-BREAK
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
type=HEADING, IMAGE=(IMG), position=(0 0), $
TYPE=REPORT,PAGE-LOCATION=OFF,$
TYPE=REPORT, FONT=HELVETICA, COLOR=BLACK, SQUEEZE=ON, $
ENDSTYLE
END
```

```
SET COMPONENT=Fuel
TABLE FILE GGSALES
"Sales report for <REGION"
" "
SUM DOLLARS/F8M
BY REGION NOPRINT
BY ST
BY CITY
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, COLOR=RED, SQUEEZE=ON, $
ENDSTYLE
END
COMPOUND END
```

次は出力結果の 1 ページ目です。



ドキュメント構文による複数ページレイアウトの作成

この例では GGSALES マスターファイルを使用し、複数ページレイアウトを使用することで、各コンポーネントを複数ページ上の固定された位置に配置します。たとえば、統合された複合レイアウトレポートには、最初のソートフィールドの各値に対応するコンポーネントレポート R1 と R2 を奇数番号ページ (表側) に配置し、コンポーネントレポート R3 を偶数番号ページ (裏側) に配置することができます。また、ロゴおよびテキスト (最初のソートフィールドの埋め込み値を含む) を共通の見出しとして、この見出しを表裏両側の上部に配置することもできます。

見出しレポートのプロシジャ (HEADER.FEX) を作成し、次の構文を入力します。

```
TABLE FILE GGSALES
" "
"Report package for <REGION"
BY REGION NOPRINT
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, SIZE=20, $
TYPE=REPORT, IMAGE=poweredbyibi.gif, POSITION=(+.25 +.25), $
TYPE=HEADING, LINE=2, ITEM=1, POSITION=4, $
ENDSTYLE
END
```

ここでは、前述の例のコンポーネント R1 および R2 を使用します。R3 レポートコンポーネントのプロシジャ (REPORT3.FEX) を作成し、次の構文を入力します。

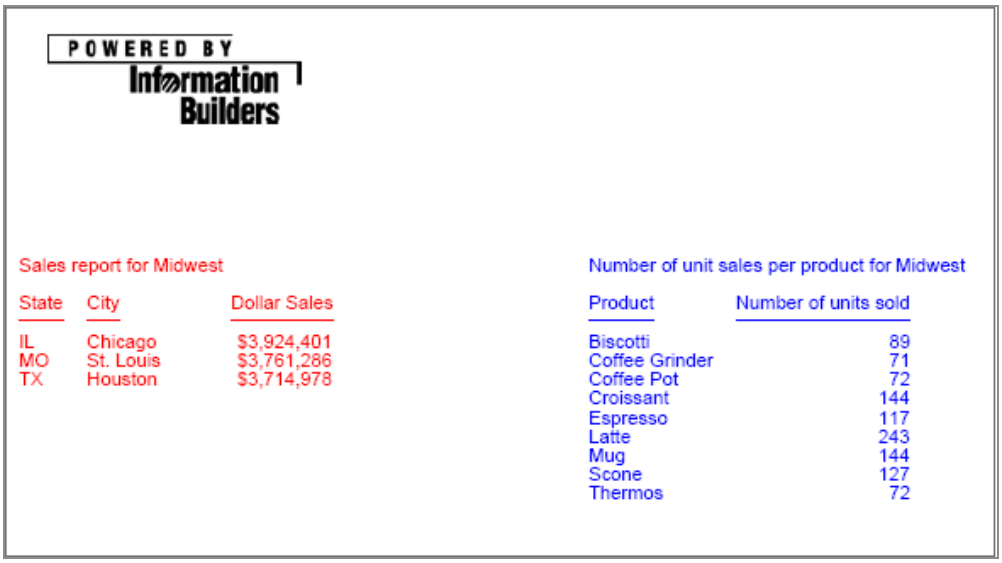
```
TABLE FILE GGSALES
"Report R3 for <REGION"
BY REGION NOPRINT
SUM DOLLARS BY ST
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, COLOR=GREEN, $
ENDSTYLE
END
```

テキストエディタで次の構文を入力し、1 ページ目に R 1 および R2 レポートコンポーネントを表示するよう指定します。R3 レポートコンポーネントを 2 ページ目に表示し、見出しレポートをドキュメントのすべてのページに表示します。

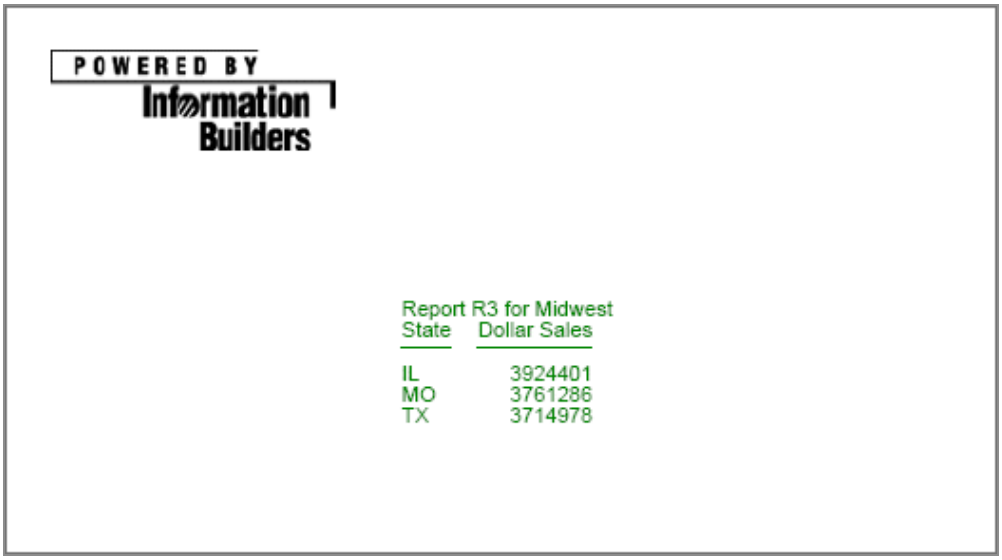
```
SET PAGE-NUM=OFF
SET SQUEEZE=ON
COMPOUND LAYOUT PCHOLD FORMAT PDF
SECTION=S1, LAYOUT=ON, MERGE=ON, ORIENTATION=LANDSCAPE, $
PAGELAYOUT=ALL, $
COMPONENT=HEADER, TYPE=REPORT, POSITION=(1 1), DIMENSION=(4 4), $
PAGELAYOUT=1, $
COMPONENT=R1, TYPE=REPORT, POSITION=(1 3), DIMENSION=(4 4), $
COMPONENT=R2, TYPE=REPORT, POSITION=(6 3), DIMENSION=(4 4), $
PAGELAYOUT=2, $
COMPONENT=R3, TYPE=REPORT, POSITION=(4 3), DIMENSION=(4 4), $
END
```

```
SET COMPONENT=HEADER
EX HEADER
SET COMPONENT=R1
EX REPORT1
SET COMPONENT=R2
EX REPORT2
SET COMPONENT=R3
EX REPORT3
COMPOUND END
```

出力結果の 1 ページ目は次のとおりです。



出力結果の 2 ページ目は次のとおりです。



ドキュメント構文によるページオーバーフローの作成

一般的なレポートでは、ページ上部に固定レイアウトが配置され、長さが固定されていない、詳細レコードを表示するためのレポートがその下に配置されます。たとえば、証券会社の明細書では、ページ上部に顧客の名前、住所、資産配分グラフ、市場指数との資産比較が配置され、その下に口座内の証券一覧が配置されます。先頭ページで証券一覧がオーバーフローする場合、その部分を 2 ページ目に続けて、すべてのページに表示される共通の見出し

(例、ロゴ、口座番号、ページ番号) の下に配置します。OVERFLOW-POSITION および OVERFLOW-DIMENSION 構文を使用すると、オーバーフローページのどの位置にレポートを配置するかを指定したり、各オーバーフローページのレポートの最大長を指定したりすることが可能になります。ただし、レポートの幅はページ間で統一する必要があります。

次の例は、GGSALES マスターファイルを使用し、OVERFLOW-POSITION および OVERFLOW-DIMENSION を指定することで 2 つ目のレポートコンポーネント (R2) の位置を変更する方法を示しています。この位置変更により、R2 レポートコンポーネントの開始位置が先頭ページの最初のコンポーネントの下になり、後続のページではページ上部から 2 インチ下になります。この設定では、各ページの最上部に見出しレポートコンポーネント (HEADER) 用の領域が十分に確保されます。

また、PAGELAYOUT=ALL により、HEADER コンポーネントが各オーバーフローページの最上部に表示されます。

テキストエディタで、次の構文を入力します。

```
SET PAGE-NUM=OFF
COMPOUND LAYOUT PCHOLD FORMAT PDF
SECTION=Example, LAYOUT=ON, MERGE=OFF, $
 PAGELAYOUT=1, $
    COMPONENT=R1, TYPE=REPORT, POSITION=(1.5 2), DIMENSION=(8 3), $
    COMPONENT=R2, TYPE=REPORT, POSITION=(.5 5), DIMENSION=(8 5),
        OVERFLOW-POSITION=(.5 2), OVERFLOW-DIMENSION=(8 8.5), $
  PAGELAYOUT=ALL, $
    COMPONENT=HEADER, TYPE=REPORT, POSITION=(1.25 1), DIMENSION=(6 1), $
END
```

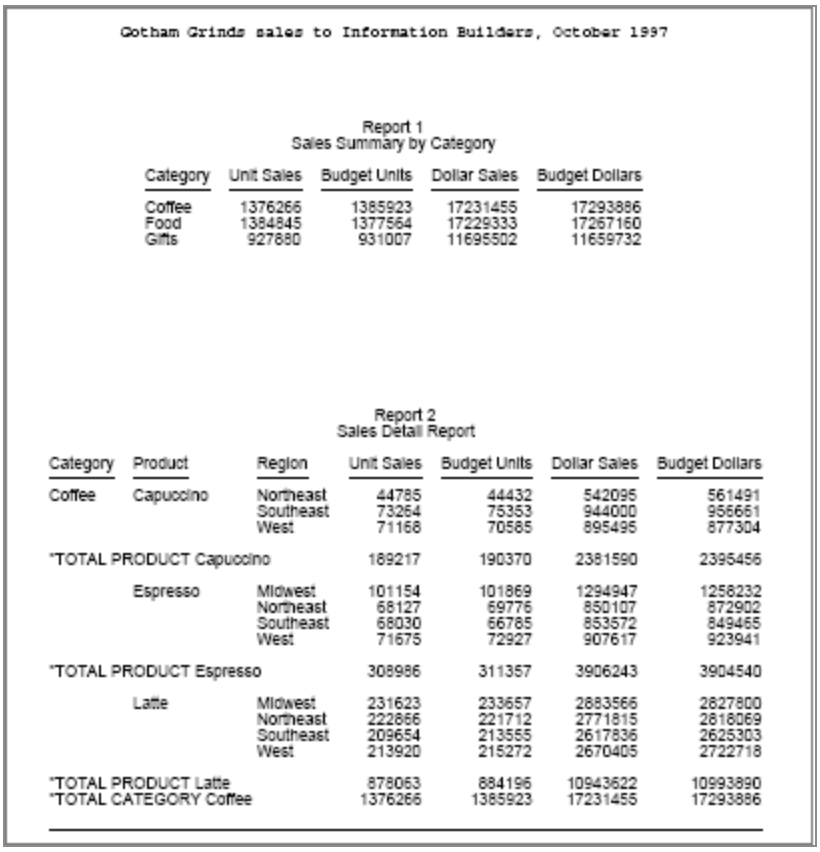
```
SET COMPONENT=R1
TABLE FILE GGSALES
HEADING CENTER
"Report 1"
"Sales Summary by Category"
" "
SUM UNITS BUDUNITS DOLLARS BUDDOLLARS BY CATEGORY
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, SQUEEZE=ON, $
ENDSTYLE
END
```

```
SET COMPONENT=R2
TABLE FILE GGSALES
HEADING CENTER
```

```
"Report 2"
"Sales Detail Report"
" "
SUM UNITS BUDUNITS DOLLARS BUDDOLLARS
BY CATEGORY BY PRODUCT BY REGION
ON CATEGORY UNDER-LINE
ON PRODUCT SUB-TOTAL
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, SQUEEZE=ON, $
ENDSTYLE
END
```

```
SET COMPONENT=HEADER
TABLE FILE GGSALES
HEADING
"Gotham Grinds sales to Information Builders, October 1997"
BY CATEGORY NOPRINT
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, IMAGE=gotham.gif, POSITION=(3.25 .25), DIMENSION=(2 .75), $
ENDSTYLE
END
```

次は出力結果の 1 ページ目です。



Gotham Grinds sales to Information Builders, October 1997

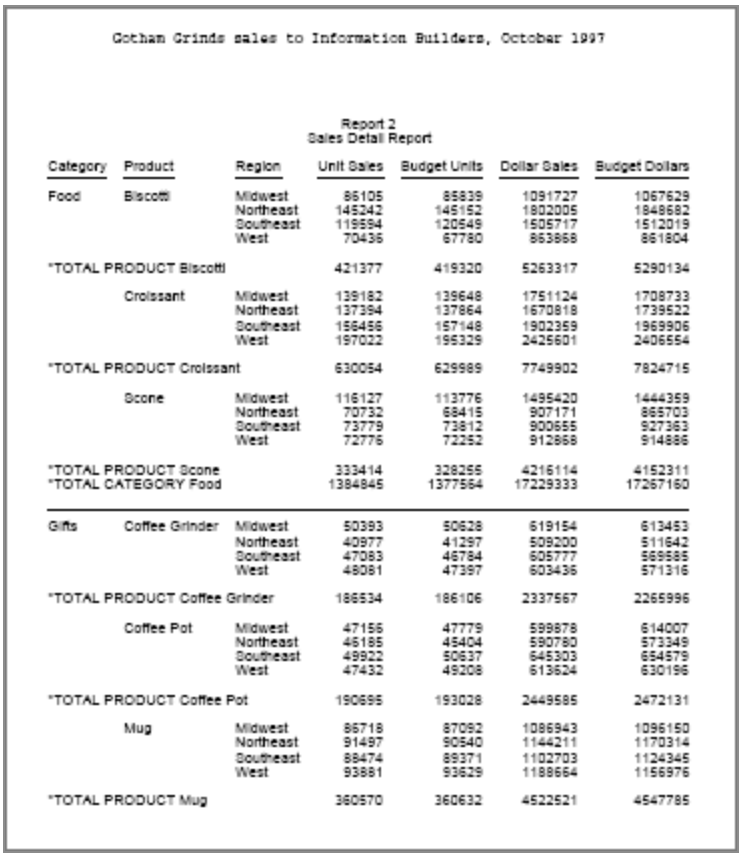
Report 1
Sales Summary by Category

Category	Unit Sales	Budget Units	Dollar Sales	Budget Dollars
Coffee	1376266	1385923	17231455	17293886
Food	1384845	1377564	17229333	17267160
Gifts	927680	931007	11695502	11659732

Report 2
Sales Detail Report

Category	Product	Region	Unit Sales	Budget Units	Dollar Sales	Budget Dollars
Coffee	Capuccino	Northeast	44785	44432	542095	561491
		Southeast	73264	75353	944000	956661
		West	71168	70585	895495	877304
*TOTAL PRODUCT Capuccino			189217	190370	2381590	2395456
	Espresso	Midwest	101154	101869	1294947	1258232
		Northeast	68127	69776	850107	872902
		Southeast	68030	66785	853572	849465
		West	71675	72927	907617	923941
*TOTAL PRODUCT Espresso			308986	311357	3906243	3904540
	Latte	Midwest	231623	233657	2883566	2827800
		Northeast	222866	221712	2771815	2818069
		Southeast	209654	213555	2617836	2625303
		West	213920	215272	2670405	2722718
*TOTAL PRODUCT Latte			878063	884196	10943622	10993890
*TOTAL CATEGORY Coffee			1376266	1385923	17231455	17293886

次は出力結果の 2 ページ目です。



Gotham Grinds sales to Information Builders, October 1997

Report 2
Sales Detail Report

Category	Product	Region	Unit Sales	Budget Units	Dollar Sales	Budget Dollars
Food	Biscotti	Midwest	86105	85839	1091727	1067629
		Northeast	145242	146152	1802005	1848682
		Southeast	119594	120549	1505717	1512019
		West	70436	67780	863868	861804
*TOTAL PRODUCT Biscotti			421377	419320	5263317	5290134
	Croissant	Midwest	139182	139648	1751124	1708733
		Northeast	137394	137864	1670818	1739522
		Southeast	156456	157148	1902359	1969906
		West	197022	195329	2425601	2406554
*TOTAL PRODUCT Croissant			630054	629989	7749902	7824715
	Scone	Midwest	116127	113776	1495420	1444359
		Northeast	70732	68415	907171	865703
		Southeast	73779	73812	900655	927363
		West	72776	72252	912868	914886
*TOTAL PRODUCT Scone			333414	328256	4216114	4152311
*TOTAL CATEGORY Food			1384845	1377564	17229333	17267160
Gifts	Coffee Grinder	Midwest	50393	50628	619154	613453
		Northeast	40977	41297	509200	511642
		Southeast	47083	46784	605777	569585
		West	48081	47397	603436	571316
*TOTAL PRODUCT Coffee Grinder			186534	186106	2337567	2265996
	Coffee Pot	Midwest	47156	47779	599878	614007
		Northeast	46185	45404	590780	573349
		Southeast	49922	50637	645303	654579
		West	47432	49208	613624	630196
*TOTAL PRODUCT Coffee Pot			190695	193028	2449585	2472131
	Mug	Midwest	86718	87092	1086943	1096150
		Northeast	91497	90540	1144211	1170314
		Southeast	88474	89371	1102703	1124345
		West	93881	93629	1188664	1156976
*TOTAL PRODUCT Mug			360570	360632	4522521	4547785

ドキュメント構文によるオブジェクトの描画

ページ上にさまざまなオブジェクトを描画してレポートを強調することができます。現在サポートされているオブジェクトには、線、ボックス、静的テキスト、イメージがあります。

これらのオブジェクトを描画する構文はレポートのスタイルシートに含めることができますが、COMPOUND LAYOUT 宣言の PAGELAYOUT グループ内に含めることもできます。各描画オブジェクトの構文は次のとおりです。

- **線**　ポイント (x1 y1) からポイント (x2 y2) に線を引くには、次の構文を入力します。

```
OBJECT=LINE, POSITION=(x1 y1), ENDPOINT=(x2 y2),
        [BORDER=b,] [BORDER-COLOR=c,] [BORDER-STYLE=s,]$
```

  必要に応じて、以下のように既存の BORDER 構文の後に BORDER、BORDER-

COLOR、BORDER-STYLE 属性を続けます。

```
OBJECT=LINE, POSITION=(1 1), ENDPOINT=(8 1),
        BORDER=HEAVY, BORDER-COLOR=RED, BORDER-STYLE=DASHED, $
```

- **ボックス**　左上の位置が (x y) で、サイズが横 xdim、縦 ydim のボックスを描画するには、次の構文を入力します。

```
OBJECT=BOX, POSITION=(x y), DIMENSION=(xdim ydim),
        BACKCOLOR=c,
        [BORDER=b,] [BORDER-COLOR=bc,] [BORDER-STYLE=bs,] $
```

ヒント：背景色の **c** は、ボックスを塗りつぶす色を指定します。有効な任意の色を使用することができます。たとえば、「YELLOW」または「RGB(200 200 200)」と指定します。

BORDER 構文では、ボックスの各辺のスタイルを個別に設定することができます。以下はその例です。

```
OBJECT=BOX, POSITION=(1 1), DIMENSION=(2 3),
        BACKCOLOR=YELLOW,
        BORDER=HEAVY, BORDER-TOP-COLOR=RED, BORDER-BOTTOM-COLOR=BLUE, $
```

BORDER 構文では、線またはボックスの属性を明示的に指定しない限り、次のデフォルト設定が適用されます。

  - 色 - 黒
  - スタイル - 実線
  - 境界線の幅 - ふつう

- **静的テキスト**　静的テキストを使用して、オブジェクト説明の一部として入力したテキストを表示します。

テキストオブジェクトのテキスト部分にマークアップタグを追加することで、テキストオブジェクト内に表示するテキストのフォーマットを設定することができます。テキストオブジェクトにマークアップタグを追加したレポートは、「マークアップレポート」と呼ばれます。マークアップレポートは、PDF、DHTML、PPT、PPTX 出力ファイルとして生成することができます。WebFOCUS は、HTML タグのサブセット、および独自のページ番号タグをサポートします。これらのマークアップタグを有効にするには、テキストオブジェクトに MARKUP=ON 属性を追加します。これにより、マークアップタグがテキストとして表示される代わりに、フォーマット要素として扱

われます。

位置 (x y) に静的テキストを配置するには、次の構文を入力します。

```
OBJECT=STRING, POSITION=(X Y), TEXT='any text you like', [MARKUP=
{ON|OFF},] [FONT=f,] [SIZE=sz,] [STYLE=st,] [COLOR=c,]
        [WRAP=ON, DIMENSION=(xdim ydim),] [LINESPACING=linesoption ,]
 $
```

説明

POSITION=(xy)

コンポーネントの左上端を配置するページ上 x y 座標を指定します。すべての座標は、UNIT で指定されている単位 (デフォルト値はインチ) で指定します。(0 0) は、物理ページの左上端を表します。

注意：デフォルト設定の座標は、物理ページの絶対位置を表します。x または y の前にプラス符号 (+) またはマイナス符号 (-) を追加すると (例 (+ 0.25 +0))、座標はページマージンの上端または左端との相対位置になります。

TEXT='任意のテキスト'

テキストオブジェクトに表示するテキストです。

注意：テキストに開始キャレット文字 (<) が含まれている場合、テキストの一部であるキャレット文字それぞれの後に、ブランクを 1 つ入力する必要があります。そうしない場合、開始キャレット文字に続くすべての文字は、マークアップタグの開始と解釈されて、テキストとして表示されません。

MARKUP={ON|OFF}

ON に設定すると、マークアップタグはフォーマットオプションとして解釈されます。OFF に設定すると、タグはテキストとして表示されます。デフォルト値は OFF です。

FONT=f

テキストに使用するデフォルトフォントです。

SIZE=sz

テキストに使用するデフォルトフォントサイズです。

STYLE=st

テキストに使用するデフォルトフォントスタイルです。

COLOR=c

テキストに使用するデフォルトフォント色です。

WRAP=ON

テキストオブジェクトの境界ボックスの終端に達した場合に、テキストの折り返しを指定します。

DIMENSION=(xdimydim)

テキストオブジェクト境界ボックスのサイズを UNIT で指定される単位で指定します。

LINESPACING=linesoption

段落内テキスト行の縦方向スペースの間隔を決定します。次の 2 種類の LINESPACING 属性がサポートされます。

```
LINESPACING={SINGLE|1.5LINES|DOUBLE}
```

または

```
LINESPACING=type(value)
```

説明

SINGLE

行で最大のフォントの高さにわずかに余分のスペースを加えたサイズになります。余分のスペース量は、使用されるフォントによって異なります。デフォルト値は SINGLE です。

1.5LINES

SINGLE の 1.5 倍のサイズです。

DOUBLE

SINGLE の 2 倍のサイズです。

type(value)

次のいずれかです。value は正の数値です。

タイプ	値	例
`MULTIPLE`	行間を拡張または減少させるパーセント値です。	LINESPACING=MULTIPLE (1.2) は、行間を 20 パーセント拡張させます。
`MIN`	行で最大のフォントに合わせるために必要な、最小の行間 (UNITS パラメータで指定した単位) です。	UNITS=IN の場合、LINESPACING=MIN(0.5) は、最小行間を 0.5 インチに設定します。
`EXACT`	WebFOCUS による補正は行われない、固定の行間 (UNITS パラメータで指定した単位) です。	UNITS=IN の場合、LINESPACING=EXACT(0.3) は、固定の行間を 0.3 インチに設定します。

必要に応じて、レポート内の任意のテキストオブジェクトに FONT、SIZE、STYLE、COLOR 属性を指定することができます。以下はその例です。

```
OBJECT=STRING, POSITION=(1 1), TEXT='Hello world!',
        FONT=TIMES, SIZE=12, STYLE=BOLD, COLOR=RED, $
```

注意：テキストの位置はその左下を基準に測定されるため、高さが異なる複数のテキストを共通の基準線に合わせて位置揃えすることができます。ただし、WRAP=ON が指定されている場合、テキストは左上の位置が (x y) で、サイズが (xdim ydim) の境界ボックスで折り返されます。この場合、テキストの左上がポイント (x y) に配置されます。

- **イメージ**　イメージを描画オブジェクトとして配置するには、次の構文を入力します。

```
OBJECT=IMAGE, IMAGE=file, POSITION=(x y), DIMENSION=(xdim ydim), $
```

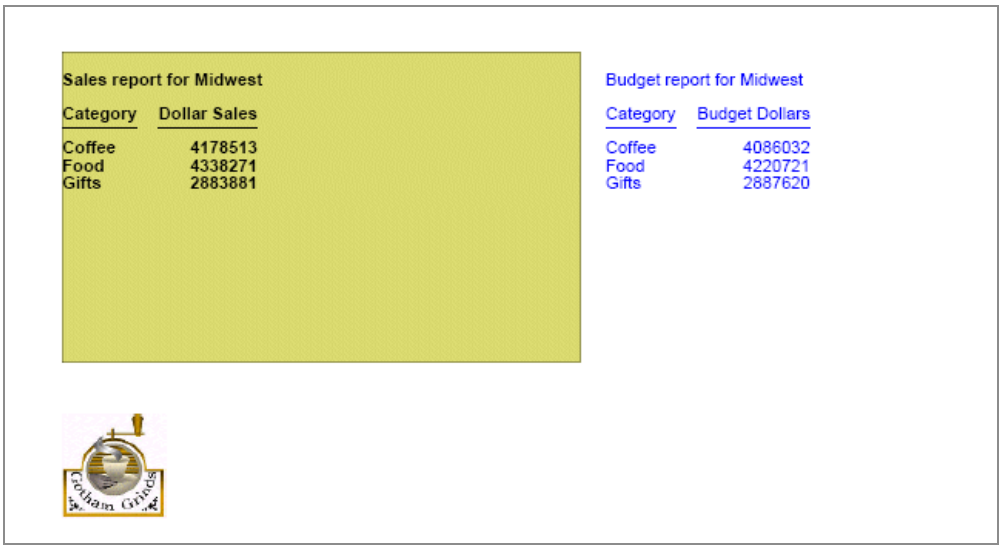
注意：イメージファイル名 (IMAGE=file) には、PDF レポートで有効な任意のイメージファイルを指定することができます。POSITION は従来のイメージと同様の方法で使用し、DIMENSION は従来のイメージの SIZE 属性の代わりに使用します。

ドキュメント構文によるオブジェクトの描画

次の例は、GGSALES マスターファイルを使用して COMPOUND LAYOUT 構文内に描画オブジェクトを配置する方法を示しています。描画オブジェクトは、COMPONENT のように、そのオブジェクトの前に指定された PAGELAYOUT 宣言のページに表示されます。

```
SET PAGE-NUM=OFF
SET SQUEEZE=ON
COMPOUND LAYOUT PCHOLD FORMAT PDF
SECTION=S1, LAYOUT=ON, MERGE=ON, ORIENTATION=LANDSCAPE, $
PAGELAYOUT=1, $
COMPONENT=Sales, TYPE=REPORT, POSITION=(1 1), DIMENSION=(4 4), $
COMPONENT=Budget, TYPE=REPORT, POSITION=(6.25 1), DIMENSION=(4 4), $
OBJECT=IMAGE, IMAGE=gglogo.gif, POSITION=(1 4.5), DIMENSION=(1 1), $
OBJECT=BOX, POSITION=(1 1), DIMENSION=(5 3), BACKCOLOR=GOLDENROD, $
END
SET COMPONENT=Sales
TABLE FILE GGSALES
"Sales report for <REGION"
" "
SUM DOLLARS
BY REGION NOPRINT
BY CATEGORY
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, STYLE=BOLD, $
ENDSTYLE
END
SET COMPONENT=Budget
TABLE FILE GGSALES
"Budget report for <REGION"
" "
SUM BUDDOLLARS
BY REGION NOPRINT
BY CATEGORY
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, COLOR=BLUE, $
ENDSTYLE
END
COMPOUND END
```

次は出力結果の 1 ページ目です。



注意：描画オブジェクトは、PAGELAYOUT 宣言内に少なくとも 1 つの COMPONENT が配置されていない限り、表示されません。

ドキュメント構文によるグリッドの表示

レポートレイアウトを手動で作成する際の補助機能として、SET LAYOUTGRID コマンドを使用することができます。グリッドを表示すると、縦横 1 インチ単位の薄い罫線がページ上に重なって表示されるため、さまざまなレポートコンポーネントの位置の確認が容易になります。現在、グリッドはインチ単位でのみ機能します。

グリッドを表示するには、次の構文を入力します。

```
SET LAYOUTGRID=ON
```

テキストオブジェクトのテキストフォーマットマークアップタグ

注意：テキストに開始キャレット文字 (<) が含まれている場合、テキストの一部であるキャレット文字それぞれの後に、ブランクを 1 つ入力する必要があります。そうしない場合、開始キャレット文字に続くすべての文字は、マークアップタグの開始と解釈されて、テキストとして表示されません。

フォントのプロパティ

font タグは、face、size、color という 3 つの属性をサポートします。

```
<font face="font" size=[+|-]n color=color_code>text</font>
```

以下はその例です。

```
<font face="New Century Schoolbook">Test1</font>
<font face="Times" size=12>test2</font>
<font face="Times New Roman" color=#0000FF size=+4>Test3</font>
<font size=-2 face="Times New Roman" color=#0000FF >Test4</font>
```

テキストスタイル

サポートされるテキストスタイルは、太字、斜体、下線、上付き文字です。

太字 - `<b>text</b>`

斜体 - `<i>text</i>`

下線 - `<u>text</u>`

上付き文字 - `<sup>text</sup>`

改行

テキストの一部分の後の改行タグは、次のテキスト部分を新しい行で開始します。改行タグに終了タグはありません。

```
<br>
```

テキストの位置揃え

位置揃えオプションは、指定した改行同様、折り返しテキストにも適用されます。横方向の位置揃えと縦方向の位置揃えの両方がサポートされます。

- **横方向の位置揃え**

  左揃え

  ```
  <left>text</left>
  ```

  右揃え

```
<right>text</right>
```

中央揃え

```
 <center>text</center>
```

両端揃え

```
 <full>text</full>
```

- **縦方向の位置揃え**

  上揃え

  ```
  <top>text</top>
  ```

  中央揃え

  ```
  <mid>text</mid>
  ```

  下揃え

  ```
  <bottom>text</bottom>
  ```

箇条書きリスト (番号なし)

項目リストを囲む番号なしの (ul) リストタグです。各項目は、リスト項目タグ (li) で囲まれます。リストの開始タグと終了タグは、それぞれ個別の行として記述する必要があります。各リスト項目は、新しい行に入力する必要があります。

```
<ul>
<li>list item1</li>
<li>list item2</li>
  .
  .
  .
</ul>
```

デフォルト設定の行頭文字は、黒丸 (disc) です。白丸 (circle) または黒い四角 (square) を指定することもできます。

```
<ul type=disc>
```

```
<ul type=circle>
```

```
<ul type=square>
```

箇条書きリスト (番号付き)

各項目に連続番号または連続文字の付いたリストを囲む、番号付き (ol) リストタグです。各項目は、リスト項目タグ (li) で囲まれます。リストの開始タグと終了タグは、それぞれ個別の行として記述する必要があります。各リスト項目は、新しい行に入力する必要があります。

```
<ol>
<li>list item1</li>
<li>list item2</li>
  .
  .
  .
</ol>
```

デフォルト設定では、リストの番号としてアラビア数字 (type=1) が使用されます。指定できる順序のタイプは次のとおりです。

アラビア数字 (デフォルト設定) - `<ol type=1>`

小文字のアルファベット - `<ol type=a>`

大文字のアルファベット - `<ol type=A>`

小文字のローマ数字 - `<ol type=I>`

大文字のローマ数字 - `<ol type=I>`

ハイパーリンク

ハイパーリンクは、PDF ドキュメントのテキストマークアップ内に含めることができます。

アンカーマークアップタグの構文は、HTML アンカー構文のサブセットです。

```
<a href="hyperlink">Text to display</a>
```

説明

hyperlink

テキストをクリックしたときの移動先ハイパーリンクです。

Text to display

ハイパーリンクとして表示するテキストです。

以下はその例です。

```
<a href="http://www.example.com/help.htm">Click here for help</a>
```

アンカーマークアップタグでは、その他の属性はサポートされません。

ページ番号と日付

次の 2 つの擬似 HTML タグを使用して、統合された複合レイアウトレポートのページマスターのテキストに、ページ番号を埋め込むことができます。

現在のページ番号 - `<ibi-page-number/>`

総ページ数 - `<ibi-total-pages/>`

MARKUP=ON に設定した場合、最大ページ数を表示するためのスペースが割り当てられるため、ページ番号と、それに続くテキストとの間隔が広くなる場合があります。ページ番号タグが含まれたテキストオブジェクト内で余分なスペースを削除するには、次のように指定します。

- テキストオブジェクトに特定のスタイルを指定する必要がない場合、マークアップタグを挿入せず、MARKUP=OFF に設定します。

  ```
  MARKUP=OFF, TEXT='Page <ibi-page-number/> of <ibi-total-pages/> of Sales
  Report', $
  ```

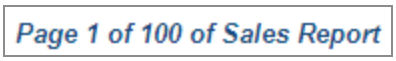
  次のように表示されます。

  ```
  Page 1 of 100 of Sales Report
  ```

- テキストオブジェクトに特定のスタイルを指定する必要がある場合、MARKUP=ON に設定する必要があります。MARKUP=ON に続けて、WRAP=OFF に設定します。文字列内では、2 つのページ番号変数間にスタイルタグを配置しないでください。「Page n of m」という文字列の周囲にはタグを使用することができます。次のようにコーディングすると、ページ番号の文字列に余分なスペースが表示されなくなります。

  ```
  MARKUP=ON, WRAP=OFF, TEXT='<font face="ARIAL" size=10><i>Page <ibi-page-
  number/> of <ibi-total-pages/> of Sales Report </i>', $
  ```

  次のように表示されます。



レポート出力に日付を表示するには、ページマスターのテキストオブジェクトに、&DATEtrMDYY など、WebFOCUS の日付変数を挿入します。

マークアップタグによる複合レイアウトテキストオブジェクトのフォーマット設定

次のリクエストは、マークアップタグ付きテキストオブジェクトを PDF 出力ファイルに表示します。

注意：テキストマークアップ構文には、非表示のキャリッジリターンやラインフィード文字を含めることはできません。このマニュアルの例では、サンプルコードにラインフィード文字が追加され、印刷ページ幅にコードが収まるようコード行が折り返されています。使用する環境でこのサンプルコードを実行するには、コードをテキストエディタにコピーし、各行の末尾に移動して Delete キーを押すことにより、テキストマークアップオブジェクト内のラインフィード文字を削除します。場合によっては、文字列の構造を保持するために、ブランクを追加する必要があります。

```
SET PAGE-NUM=OFF
SET LAYOUTGRID=ON
TABLE FILE GGSALES
BY REGION NOPRINT
ON TABLE PCHOLD AS LINESP1 FORMAT PDF
ON TABLE SET STYLE *
type=report, size=8, $
object=string, position=(1 1), dimension=(7 3), wrap=on, markup=on,
 linespacing=multiple(3),
 text='<b><font face="Arial" size=12>This paragraph is triple-spaced
 (LINESPACING=MULTIPLE(3)):</font></b>
 <full>Our <i>primary</I> goal for fiscal 2006 was to accelerate our
 transformation to customer centricity.In this letter, I’d like to
 give you an update on this work, which contributed to the 22-percent
 increase in earnings from continuing operations we garnered for fiscal
 2006.Since the past is often prologue to the future, I’d like to
 describe how customer centricity is influencing not only our goals for
 fiscal 2007, but also our long-term plans.At Gotham Grinds, customer
 centricity means treating each customer as a unique individual, meeting
 their needs with end-to-end solutions, and engaging and energizing our
 employees to serve them.</full>', $
ENDSTYLE
END
```

以下は、このリクエストの説明です。

- データソースのフィールドは、どれも表示されません。出力には、テキストオブジェクトのみが表示されます。
- SET LAYOUTGRID コマンドは、罫線を表示して、テキストオブジェクトの座標およびサイズを示します。
- OBJECT=STRING 宣言は、LINESPACING=MULTIPLE(3) によって、3 行の行間を指定します。
- 次のテキストは、太字、Arial、12 ポイント (デフォルト値は TYPE=REPORT で指定される 8 ポイント) で表示されます。

  'This paragraph is triple-spaced (LINESPACING=MULTIPLE(3)):'

  このフォーマットのマークアップは、次のとおりです。

  ```
  <b><font face="Arial" size=12>This paragraph is triple-spaced
  (LINESPACING=MULTIPLE(3)):</font></b>
  ```

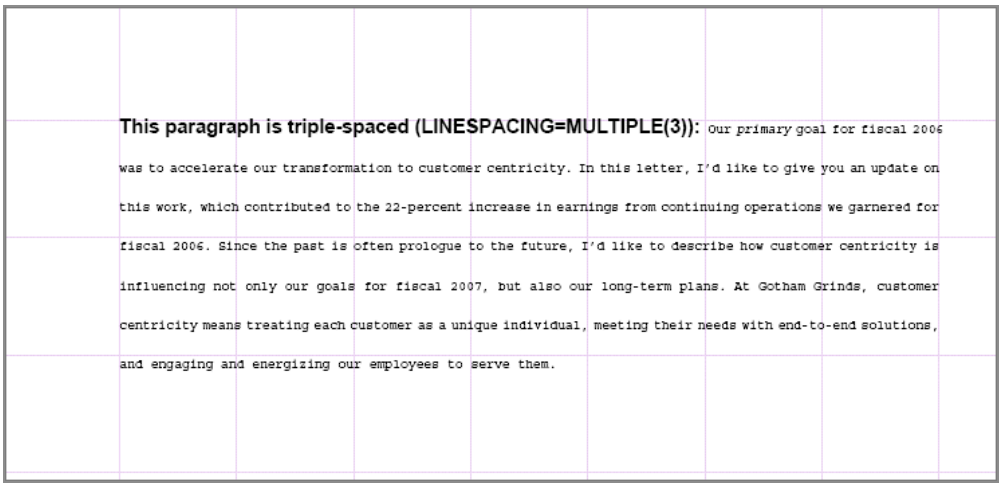
  ただし、イメージのサイズは、ページに収まるように変更されているため、小さいフォントで表示されます。
- 残りのテキストは、両端揃え (左右の端で揃える) で表示されます。

  ```
  <full>Our ... </full>
  ```

- 次のマークアップは、「primary」というテキストを斜体で表示します。

  ```
  <i>primary</I>
  ```

出力結果は次のとおりです。



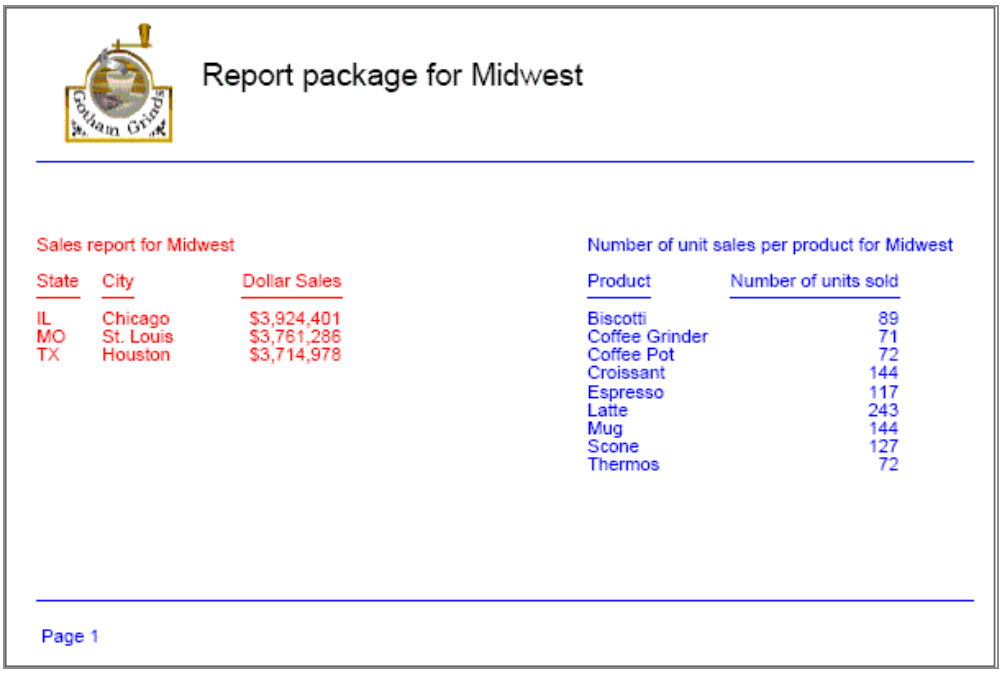
ページマスターへのテキストおよび線オブジェクトの配置

次のリクエストは、ページマスター上のヘッダレポートとコンポーネントレポートとの間に線を配置し、各ページの最下部に線とテキストを配置します。

```
SET PAGE-NUM=OFF
SET SQUEEZE=ON
COMPOUND LAYOUT PCHOLD FORMAT PDF
SECTION=S1, LAYOUT=ON, MERGE=ON, ORIENTATION=LANDSCAPE, $
PAGELAYOUT=ALL, $
COMPONENT=HEADER, TYPE=REPORT, POSITION=(1 1), DIMENSION=(4 4), $
OBJECT=STRING, POSITION=(1 6.6), MARKUP=ON,
TEXT='<font face="Arial" color=#0000FF size=12> Page <ibi-page-number/>
        </font> ', WRAP=ON, DIMENSION=(4 4),$
OBJECT=LINE, POSITION=(1 2.5), ENDPOINT=(9.5 2.5),
        BORDER-COLOR=BLUE,$
OBJECT=LINE, POSITION=(1 6.5), ENDPOINT=(9.5 6.5),
        BORDER-COLOR=BLUE,$
PAGELAYOUT=1, $
COMPONENT=R1, TYPE=REPORT, POSITION=(1 3), DIMENSION=(4 4), $
COMPONENT=R2, TYPE=REPORT, POSITION=(6 3), DIMENSION=(4 4), $
PAGELAYOUT=2, $
COMPONENT=R3, TYPE=REPORT, POSITION=(4 3), DIMENSION=(4 4), $
END
```

```
SET COMPONENT=HEADER
TABLE FILE GGSALES
" "
"Report package for <REGION"
BY REGION NOPRINT
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, SIZE=20, $
TYPE=REPORT, IMAGE=gglogo.gif, POSITION=(+.25 +.25), $
TYPE=HEADING, LINE=2, ITEM=1, POSITION=1.5, $
ENDSTYLE
END
SET COMPONENT=R1
TABLE FILE GGSALES
"Sales report for <REGION"
" "
SUM DOLLARS/F8M
BY REGION NOPRINT
BY ST
BY CITY
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, COLOR=RED, SQUEEZE=ON, $
ENDSTYLE
END
SET COMPONENT=R2
TABLE FILE GGSALES
"Number of unit sales per product for <REGION"
" "
SUM CNT.UNITS AS 'Number of units sold'
BY REGION NOPRINT
BY PRODUCT
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, COLOR=BLUE, SQUEEZE=ON, $
ENDSTYLE
END
SET COMPONENT=R3
TABLE FILE GGSALES
"Report R3 for <REGION"
BY REGION NOPRINT
SUM DOLLARS BY ST
ON TABLE SET STYLE *
TYPE=REPORT, FONT=HELVETICA, COLOR=GREEN, $
ENDSTYLE
END
COMPOUND END
```

次は出力結果の 1 ページ目です。



Report package for Midwest

Sales report for Midwest

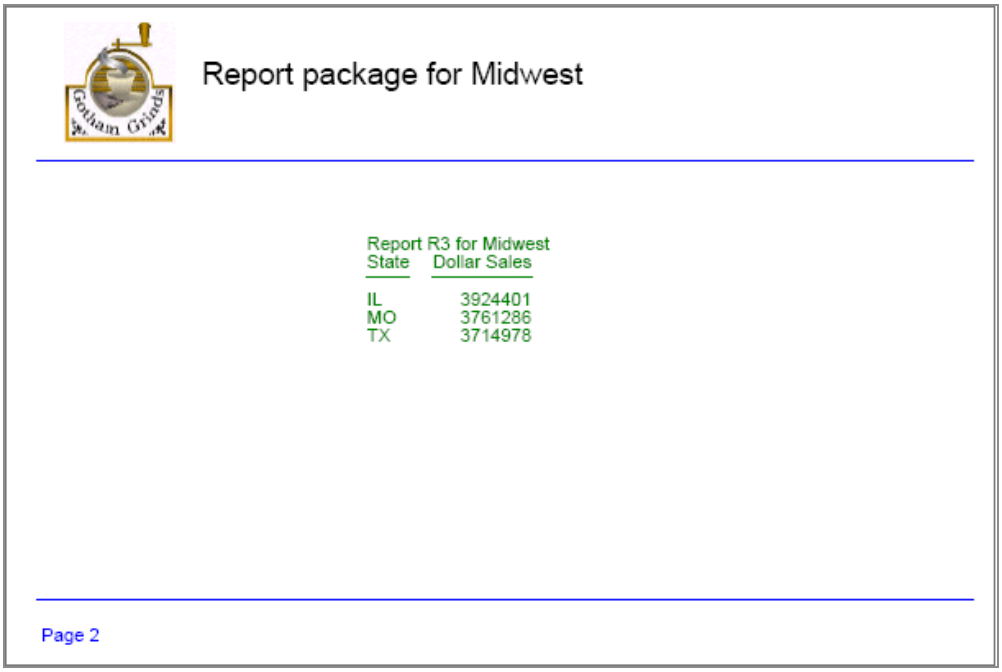
State	City	Dollar Sales
IL	Chicago	$3,924,401
MO	St. Louis	$3,761,286
TX	Houston	$3,714,978

Number of unit sales per product for Midwest

Product	Number of units sold
Biscotti	89
Coffee Grinder	71
Coffee Pot	72
Croissant	144
Espresso	117
Latte	243
Mug	144
Scone	127
Thermos	72

Page 1

出力結果の 2 ページ目には、同一の描画オブジェクトが配置されています。



Report package for Midwest

Report R3 for Midwest

State	Dollar Sales
IL	3924401
MO	3761286
TX	3714978

Page 2

PDF レポート出力でのテキストの縦方向配置マークアップ

次のリクエストは、3 つのボックスを作成し、各ボックス内にテキストオブジェクトを配置します。

- 左側のボックスでは、縦方向の上揃えでテキストが上段に配置されています。
- 中央のボックスでは、縦方向の中央揃えでテキストが中段に配置されています。
- 右側のボックスでは、縦方向の下揃えでテキストが下段に配置されています。

注意：テキストマークアップ構文には、非表示のキャリッジリターンやラインフィード文字を含めることはできません。このマニュアルの例では、サンプルコードにラインフィード文字が追加され、印刷ページ幅にコードが収まるようコード行が折り返されています。使用する環境でこのサンプルコードを実行するには、コードをテキストエディタにコピーし、各行の末尾に移動して Delete キーを押すことにより、テキストマークアップオブジェクト内のラインフィード文字を削除します。場合によっては、文字列の構造を保持するために、ブランクを追加する必要があります。

```
SET PAGE-NUM=OFF
TABLE FILE GGSALES
BY REGION NOPRINT
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
type=report, font=arial, size=10, $
object=box, position=(1 1), dimension=(6 1), $
object=line, position=(3 1), endpoint=(3 2), $
object=line, position=(5 1), endpoint=(5 2), $
object=string, text='<top>Vertically aligned text within a text object using
top alignment.</top>', position=(1.05 1), dimension=(2 1), linespacing=exact
(.15), markup=on, wrap=on, $
object=string, text='<mid>Vertically aligned text within a text object using
middle alignment.</mid>', position=(3.05 1), dimension=(2 1),
linespacing=exact(.15), markup=on, wrap=on, $
object=string, text='<bottom>Vertically aligned text within a text object
using bottom alignment.</bottom>', position=(5.05 .9), dimension=(2
1),linespacing=exact(.15), markup=on, wrap=on, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Vertically aligned text within a text object using top alignment.	Vertically aligned text within a text object using middle alignment.	Vertically aligned text within a text object using bottom alignment.

複合レイアウトレポートとミッシングデータ

複合レイアウトレポートは、第 1 ソートフィールドが共通の個々のコンポーネントレポートまたはコンポーネントグラフで構成されます。複合プロシジャは、ソートフィールドの個々の値をそれぞれ単一ページまたはページセットとして表示する単一の出力ドキュメントを生成します。埋め込まれたコンポーネントは分割されて、そのソートフィールド値に対応するデータを表示します。

複合レイアウトレポートでは、コンポーネントレポート群の少なくとも 1 つにソートフィールド値が存在する場合に、そのソートフィールド値ごとのページが生成され、指定されたページレイアウトでページが表示されます。ソートフィールド値のデータが存在するコンポーネントにはそのデータが表示され、データが存在しないコンポーネントにはブランクのレポートが表示されます。

ブランクコンポーネントのレポートページでの表示方法は、複合レイアウトレポート内でのコンポーネントの配置定義によって決定されます。コンポーネントの配置は、絶対位置として、またはレイアウト内のその他のコンポーネントとの相対位置として定義することができます。ブランクのコンポーネントとそれに続くページ上のコンポーネントを絶対位置として定義する場合、ブランクコンポーネントのレポートは、指定された位置にブランクスペースとして表示されます。ブランクコンポーネントとそれに続くコンポーネントとの間に相対位置を定義する場合は、ブランクコンポーネントの後のコンポーネントはページの上部に移動され、ブランクコンポーネントに定義されたエリアにブランクスペースは表示されません。

レポート間に相対位置が定義されている複合レポートでは、ブランクのレポートが存在する場合、それに続くレポートはデータが存在する最後のレポートの最下部を基準にして縦方向に相対配置され、横方向にはページマージンを基準にして相対配置されます。つまり、データが含まれていないレポートがある場合、それに続くレポートは縦方向に浮遊した状態 (移動可能) になり、前のレポートを基準にして相対配置されますが、ページの横方向には前のレポートを基準にした移動はありません。

レポートの位置は、POSITION = (X Y) を使用して、左座標 (X = 横方向) と上座標 (Y = 縦方向) で指定します。これらの座標は、それぞれ独立して定義することができ、前のレポートを基準にして相対位置を指定したり、絶対位置を指定できます。

位置を定義する際に、ページ上のレポートを縦方向に浮遊した状態にしてブランクレポートを置換する一方、横方向にはページの左側に固定させる場合は、Y 座標 (上) を相対座標にし、X 座標 (左) を固定座標にします。位置座標のいずれかを固定するには、相対位置の参照 (+/-) を絶対位置に変更します。以下はその例です。

両方を相対座標にする場合は、POSITION(+0.003 +0.621) のように定義します。

横方向を固定し、縦方向を浮動にする場合は、POSITION(0.520 +0.621) のように定義します。

この機能は、リクエストの WebFOCUS 構文の変更や追加を行わずに利用することができます。ただし、複合コンポーネント内でデータを選択して関連付ける方法に注意して、期待どおり出力されることを確認してください。

設定 - ミッシングデータを含む複合レイアウトレポートの作成

この例では、選択した店舗グループの未払在庫注文レポートを作成するステートメントセットを作成します。各店舗は、「Food」、「Coffee」、「Gifts」という 3 つの在庫カテゴリにおいて、注文残高を保有する可能性があります。

この機能について説明するため、最初に、選択した店舗情報が格納されたヘッダファイル、各在庫カテゴリのトランザクションファイル、というデータファイルセットを作成します。ブランクのレポートコンポーネントが異なる場合に、この機能がどのように動作するかについて説明するため、特定の店舗データを使用します。

データファイルを作成後、4 つのコンポーネントレポートを作成します。1 つ目はヘッダ情報を表示するレポートで、それ以外は各在庫カテゴリについてのレポートです。

最後に、この情報すべてを店舗ごとの単一ページに統合するために、これらのレポートを複合レイアウトレポートに集約します。

手順 1 - データファイルの作成

GGORDER データソースと GGSALES を結合し、次の 4 つのデータファイルを作成します。

作成されるデータファイル	格納される情報のタイプ	選択される店舗
GGHDR	店舗情報	R1019、R1020、R1040、R1041
GG1	Coffee の注文トランザクション	R1019、R1040、R1088
GG2	Food の注文トランザクション	R1019、R1020、R1041、R1088
GG3	Gifts の注文トランザクション	R1019、R1020、R1040、R1088

APP HOLD、JOIN、DEFINE の各コマンドを次のように使用します。

```
APP HOLD baseapp
JOIN
 GGORDER.ORDER01.STORE_CODE IN GGORDER TO
   UNIQUE GGSTORES.STORES01.STORE_CODE
   IN GGSTORES AS J1
END
DEFINE FILE GGORDER
PRODUCT_CATEGORY/A15=IF (PRODUCT_DESCRIPTION IN
                     ('Biscotti','Croissant','Scone')) THEN 'Food'
                     ELSE IF (PRODUCT_DESCRIPTION IN
                     ('French Roast','Hazelnut','Kona')) THEN 'Coffee' ELSE
'Gifts';
END
```

次のプロシジャは、データソース GGHDR を生成します。

```
TABLE FILE GGORDER
SUM
     FST.STORE_NAME
     FST.ADDRESS1
     FST.ADDRESS2
     FST.CITY
     FST.STATE
     FST.ZIP
BY STORE_CODE
```

```
WHERE STORE_CODE IN ('R1019','R1020','R1040','R1041');
ON TABLE NOTOTAL
ON TABLE HOLD AS GGHDR FORMAT FOCUS INDEX 'STORE_CODE'
END
```

次のプロシジャは、データソース GG1 を生成します。

```
TABLE FILE GGORDER
PRINT
     QUANTITY
     UNIT_PRICE
     PACKAGE_TYPE
     SIZE
     VENDOR_NAME
     PRODUCT_CATEGORY
BY STORE_CODE
BY ORDER_DATE
BY PRODUCT_DESCRIPTION
WHERE ( PRODUCT_CATEGORY EQ 'Coffee' ) AND ( ORDER_DATE GE '09/01/97' );
WHERE STORE_CODE IN ('R1019','R1040','R1088');
ON TABLE HOLD AS GG1 FORMAT FOCUS INDEX 'STORE_CODE'
END
```

次のプロシジャは、データソース GG2 を生成します。

```
TABLE FILE GGORDER
PRINT
     QUANTITY
     UNIT_PRICE
     PACKAGE_TYPE
     SIZE
     VENDOR_NAME
     PRODUCT_CATEGORY
BY STORE_CODE
BY ORDER_DATE
BY PRODUCT_DESCRIPTION
WHERE ( PRODUCT_CATEGORY EQ 'Food' ) AND ( ORDER_DATE GE '09/01/97' );
WHERE STORE_CODE IN ('R1019','R1020','R1041','R1088');
ON TABLE HOLD AS GG2 FORMAT FOCUS INDEX 'STORE_CODE'
END
```

次のプロシジャは、データソース GG3 を生成します。

```
TABLE FILE GGORDER
PRINT
     QUANTITY
     UNIT_PRICE
     PACKAGE_TYPE
     SIZE
     VENDOR_NAME
     PRODUCT_CATEGORY
BY STORE_CODE
BY ORDER_DATE
BY PRODUCT_DESCRIPTION
WHERE ( PRODUCT_CATEGORY EQ 'Gifts' ) AND ( ORDER_DATE GE '09/01/97' );
WHERE STORE_CODE IN ('R1019','R1020','R1040','R1088');
ON TABLE HOLD AS GG3 FORMAT FOCUS INDEX 'STORE_CODE'
END
```

手順 2 - コンポーネントレポートの作成

次の GGHDR.FEX プロシジャは、GGHDR データソースから複合レイアウトレポートの最初のレポートコンポーネントを作成します。これは、複合レイアウトレポートの各ページのヘッダとして機能します。レポートコンポーネントすべての共通ソートフィールドは、STORE_CODE です。このプロシジャによって、PDF 出力ファイルの各ページに 1 つの店舗名と住所が表示されます。

```
TABLE FILE GGHDR
BY STORE_CODE NOPRINT
HEADING
"<STORE_NAME "
"<ADDRESS1 "
"<ADDRESS2 "
"<CITY , <STATE   <ZIP "
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
     UNITS=IN,
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,$
TYPE=REPORT,
     GRID=OFF,
     FONT='ARIAL',
```

```
     SIZE=9,$
TYPE=HEADING,
     SIZE=10,
     STYLE=BOLD,
     COLOR=BLUE,$
ENDSTYLE
END
```

次の GGRPT1.FEX プロシジャは、複合レイアウトレポートの 2 つ目のレポートコンポーネントを作成します。ヘッダレポートと店舗コードの値が同一の場合、GG1 データソースから製品カテゴリ Coffee についてのデータが表示されます。

```
TABLE FILE GG1
SUM
     QUANTITY
     UNIT_PRICE
     FST.PACKAGE_TYPE AS ',Package'
     FST.SIZE AS ',Size'
     VENDOR_NAME
BY STORE_CODE AS 'Store'
BY PRODUCT_DESCRIPTION
HEADING
"<PRODUCT_CATEGORY"
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
     UNITS=IN,
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,$
TYPE=REPORT,
     GRID=OFF,
     FONT='ARIAL',
     SIZE=9,$
TYPE=DATA,
     SIZE=10,$
TYPE=TITLE,
     STYLE=BOLD,
     SIZE=10,$
TYPE=HEADING,
     SIZE=10,
     STYLE=BOLD,
    COLOR=RED,$
```

```
ENDSTYLE
END
```

次の GGRPT2.FEX プロシジャは、複合レイアウトレポートの 3 つ目のレポートコンポーネントを作成します。ヘッダレポートと店舗コードの値が同一の場合、GG2 データソースから製品カテゴリ Food についてのデータが表示されます。

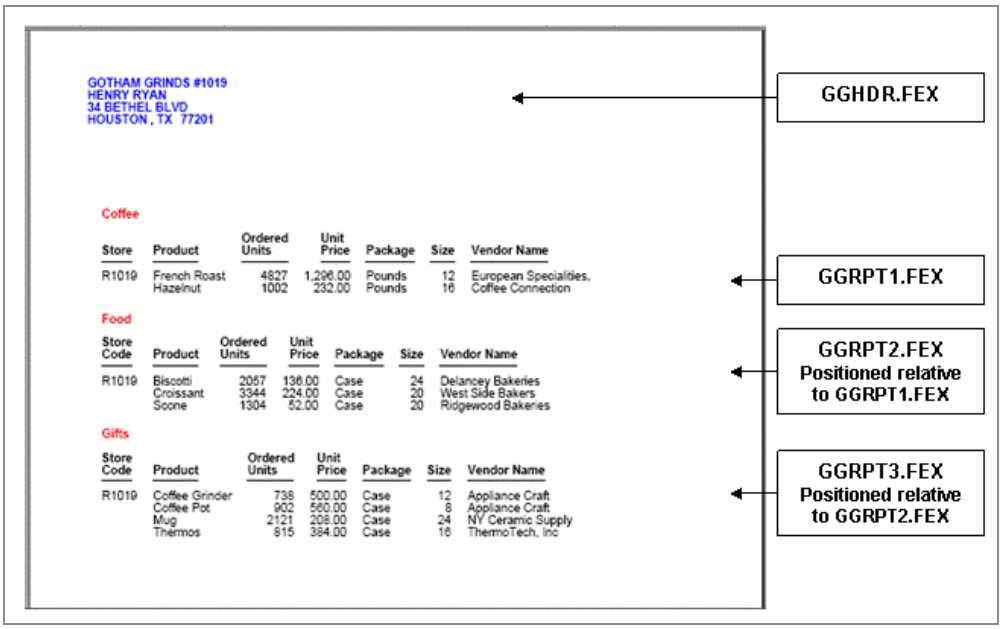
```
TABLE FILE GG2
SUM
     QUANTITY
     UNIT_PRICE
     FST.PACKAGE_TYPE AS ',Package'
     FST.SIZE AS ',Size'
     VENDOR_NAME
BY STORE_CODE
BY PRODUCT_DESCRIPTION
HEADING
"<PRODUCT_CATEGORY"
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
     UNITS=IN,
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,$
TYPE=REPORT,
     GRID=OFF,
     FONT='ARIAL',
     SIZE=9,$
TYPE=DATA,
    SIZE=10,$
TYPE=TITLE,
     STYLE=BOLD,
     SIZE=10,$
TYPE=HEADING,
     SIZE=10,
     STYLE=BOLD,
     COLOR=RED,$
ENDSTYLE
END
```

次の GGRPT3.FEX プロシジャは、複合レイアウトレポートの最後のレポートコンポーネントを作成します。ヘッダレポートと店舗コードの値が同一の場合、GG3 データソースから製品カテゴリ Gifts についてのデータが表示されます。

```
TABLE FILE GG3
SUM
     QUANTITY
     UNIT_PRICE
     FST.PACKAGE_TYPE AS ',Package'
     FST.SIZE AS ',Size'
     VENDOR_NAME
BY STORE_CODE
BY PRODUCT_DESCRIPTION
HEADING
"<PRODUCT_CATEGORY"
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
     UNITS=IN,
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,$
TYPE=REPORT,
     GRID=OFF,
     FONT='ARIAL',
     SIZE=9,$
TYPE=DATA,
     SIZE=10,$
TYPE=TITLE,
     STYLE=BOLD,
     SIZE=10,$
TYPE=HEADING,
     SIZE=10,
     STYLE=BOLD,
     COLOR=RED,$
ENDSTYLE
END
```

手順 3 - 複合レイアウトレポートの作成

次の GGCMPD.FEX プロシジャは、4 つのコンポーネントを統合して、複合レイアウトレポートを作成します。下図は、レポートと各コンポーネントの相対的な位置関係を示しています。

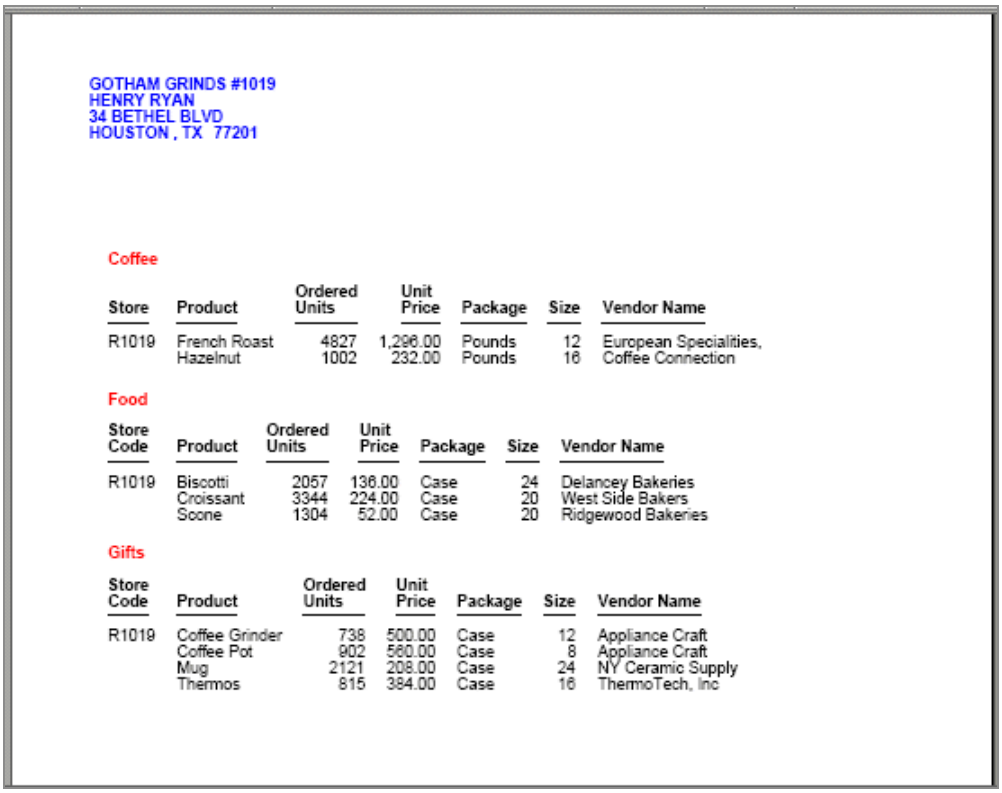


複合レイアウトレポートの構文は、次のとおりです。

```
SET HTMLARCHIVE=ON
COMPOUND LAYOUT PCHOLD FORMAT PDF
UNITS=IN, $
SECTION=section1, LAYOUT=ON, MERGE=ON, ORIENTATION=PORTRAIT,
   PAGESIZE=Letter,  $
PAGELAYOUT=1, NAME='Page layout 1', text='Page layout 1',
   TOC-LEVEL=1, BOTTOMMARGIN=0.5, TOPMARGIN=0.5,  $
COMPONENT='report1', TEXT='report1', TOC-LEVEL=2,
   POSITION=(0.667 1.083), DIMENSION=(3.417 1.412),  $
COMPONENT='report2', TEXT='report2', TOC-LEVEL=2,
   POSITION=(0.837 2.584), DIMENSION=(* *),  $
COMPONENT='report3', TEXT='report3', TOC-LEVEL=2,
   POSITION=(-0.006 +0.084), DIMENSION=(* *),
   RELATIVE-TO='report2', RELATIVE-POINT=BOTTOM-LEFT,
   POSITION-POINT=TOP-LEFT, $
COMPONENT='report4', TEXT='report4', TOC-LEVEL=2,
   POSITION=(+0.010 +0.080), DIMENSION=(* *),
   RELATIVE-TO='report3', RELATIVE-POINT=BOTTOM-LEFT,
   POSITION-POINT=TOP-LEFT, $
END
SET COMPONENT='report1'
-INCLUDE GGHDR.FEX
SET COMPONENT='report2'
-INCLUDE GGRPT1.FEX
SET COMPONENT='report3'
-INCLUDE GGRPT2.FEX
```

```
SET COMPONENT='report4'
-INCLUDE GGRPT3.FEX
COMPOUND END
```

すべてのコンポーネントのデータが存在することから、PDF 出力ファイルの最初のページには、これらがレポートに表示されます。



GOTHAM GRINDS #1019
HENRY RYAN
34 BETHEL BLVD
HOUSTON , TX 77201

Coffee

Store	Product	Ordered Units	Unit Price	Package	Size	Vendor Name
R1019	French Roast	4827	1,296.00	Pounds	12	European Specialities,
	Hazelnut	1002	232.00	Pounds	16	Coffee Connection

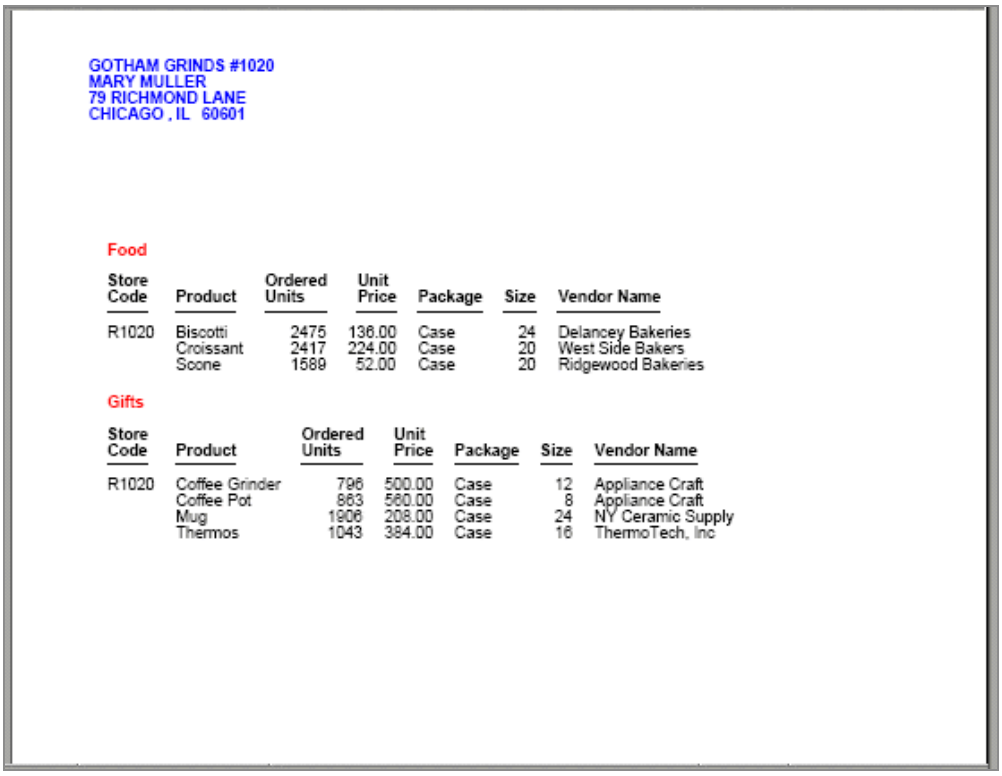
Food

Store Code	Product	Ordered Units	Unit Price	Package	Size	Vendor Name
R1019	Biscotti	2057	136.00	Case	24	Delancey Bakeries
	Croissant	3344	224.00	Case	20	West Side Bakers
	Scone	1304	52.00	Case	20	Ridgewood Bakeries

Gifts

Store Code	Product	Ordered Units	Unit Price	Package	Size	Vendor Name
R1019	Coffee Grinder	738	500.00	Case	12	Appliance Craft
	Coffee Pot	902	560.00	Case	8	Appliance Craft
	Mug	2121	208.00	Case	24	NY Ceramic Supply
	Thermos	815	384.00	Case	16	ThermoTech, Inc

GGRPT2.FEX レポートでは、このヘッダに記載されている店舗のデータが存在しませんでした。そのため、Coffee コンポーネントが欠落しています。コンポーネント 3 およびコンポーネント 4 は、複合レイアウトレポート構文で、上部に定義されているコンポーネントとの相対位置で指定されていることから、Coffee コンポーネントの配置されるべき位置にブランクのスペースが残されるのではなく、Food コンポーネントおよび Gifts コンポーネントは上部に移動され、詰めて表示されます。



GOTHAM GRINDS #1020
MARY MULLER
79 RICHMOND LANE
CHICAGO , IL 60601

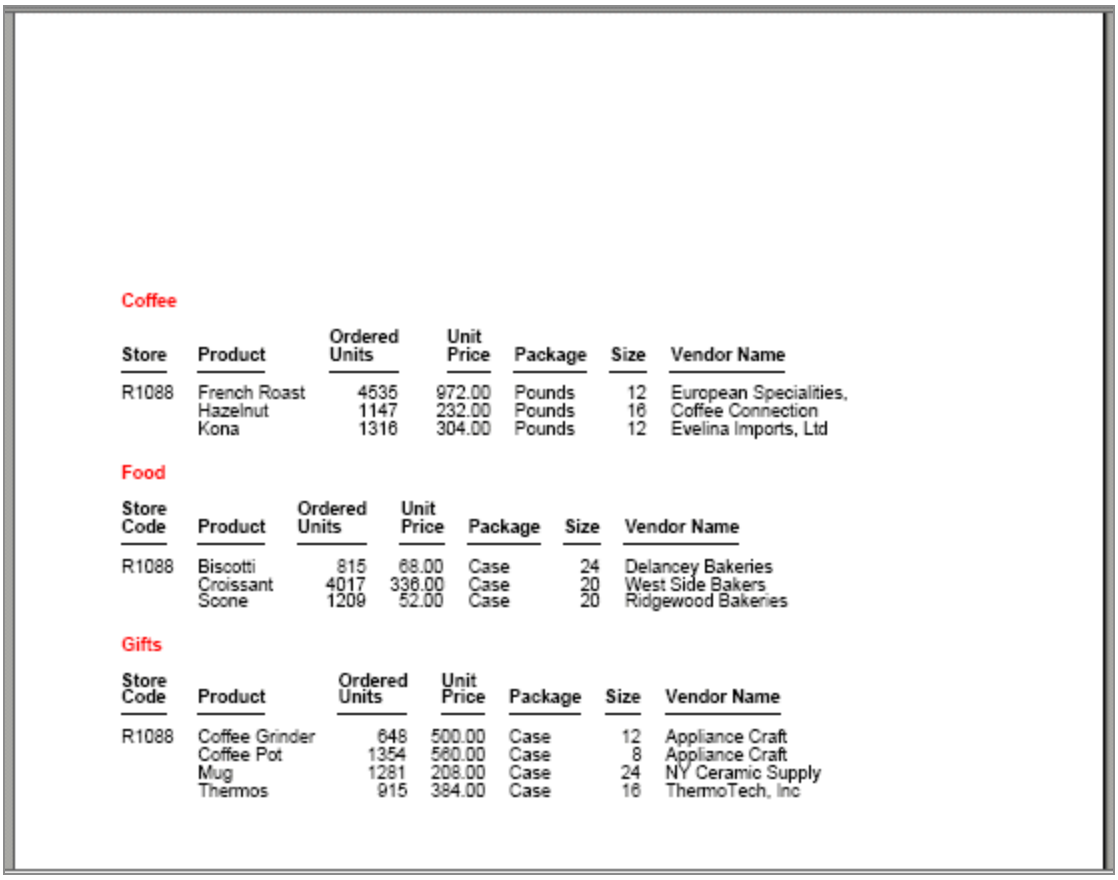
Food

Store Code	Product	Ordered Units	Unit Price	Package	Size	Vendor Name
R1020	Biscotti	2475	136.00	Case	24	Delancey Bakeries
	Croissant	2417	224.00	Case	20	West Side Bakers
	Scone	1589	52.00	Case	20	Ridgewood Bakeries

Gifts

Store Code	Product	Ordered Units	Unit Price	Package	Size	Vendor Name
R1020	Coffee Grinder	796	500.00	Case	12	Appliance Craft
	Coffee Pot	863	560.00	Case	8	Appliance Craft
	Mug	1906	208.00	Case	24	NY Ceramic Supply
	Thermos	1043	384.00	Case	16	ThermoTech, Inc

5 ページ目には、その他 3 つのコンポーネントの店舗コードの値について、ヘッダレポートのレコードが存在しません。このような場合でも、この店舗コードのページは生成されます。2 つ目のコンポーネント (Coffee レポート) は、複合レイアウトレポート構文によって、最初のコンポーネント (ヘッダレポート) との相対位置ではなく、絶対位置で配置されるため、ヘッダレポートが配置されるべきスペースは、ブランクのままになっています。



Coffee

Store	Product	Ordered Units	Unit Price	Package	Size	Vendor Name
R1088	French Roast	4535	972.00	Pounds	12	European Specialities,
	Hazelnut	1147	232.00	Pounds	16	Coffee Connection
	Kona	1316	304.00	Pounds	12	Evelina Imports, Ltd

Food

Store Code	Product	Ordered Units	Unit Price	Package	Size	Vendor Name
R1088	Biscotti	815	68.00	Case	24	Delancey Bakeries
	Croissant	4017	336.00	Case	20	West Side Bakers
	Scone	1209	52.00	Case	20	Ridgewood Bakeries

Gifts

Store Code	Product	Ordered Units	Unit Price	Package	Size	Vendor Name
R1088	Coffee Grinder	648	500.00	Case	12	Appliance Craft
	Coffee Pot	1354	560.00	Case	8	Appliance Craft
	Mug	1281	208.00	Case	24	NY Ceramic Supply
	Thermos	915	384.00	Case	16	ThermoTech, Inc

PPTX、PDF 複合レイアウトレポートの BY フィールドによる目次作成

複合レイアウトレポート構文を使用して、PPTX および PDF 複合レポートの目次を生成することができます。

PPTX の場合、目次をドキュメントの最初に目次ページとして表示することができます。PDF の場合、目次は、Adobe Reader の PDF のしおり (ブックマーク) として表示することや、目次ページとしてドキュメントの最初に表示することができます。また、これらを両方表示することもできます。目次の項目としおり (PDF) は、どちらも各コンポーネント、および追加済みソートフィールドへのリンクを提供します。これらのリンクを使用して、現在のページから、リンク先にジャンプすることができます。

複合レイアウトレポート構文で、TOC 説明および TOC レベルを定義することにより、TOC (目次) の特定レベルに任意のレポートコンポーネントおよびグラフコンポーネントを含めることができます。また、TOC ツリーの最上位にレポートコンポーネントエントリを表示し、その下位にレポートコンポーネントの BY フィールド値をインデントさせて表示することも

できます。BYTOC エントリは、レポートコンポーネントでのみサポートされ、グラフコンポーネントではサポートされません。

複合レポートで BY フィールド値に基づく目次またはしおり (PDF) を定義した場合、その複合レポートを、統合された複合レポート (MERGE=ON) として出力することも、統合されていない複合レポート (MERGE=OFF) として出力することもできます。統合された複合レポートの TOC ツリーでは、レポートの各インスタンスを表す最上位エントリとして主ソートキー値が表示され、その下位エントリとして後続のキー値が表示されます。また、統合された複合レポートを、主ソートキー値に基づいてそれぞれ個別のドキュメントにバーストし、ReportCaster を使用して分割配信することもできます。

目次の機能

目次 (TOC) ページには、ドキュメント内容の概要が、ページ番号とともに表示されます。目次は、ドキュメントとともに印刷することができます。目次を利用すると、ドキュメントをオンラインで表示したまま、特定のセクションに簡単に移動することができます。目次項目は、複合レポート出力で任意のコンポーネント (ページ、レポート、グラフ)、複合レポート内で任意のオブジェクト (イメージ、テキストボックス)、各コンポーネントレポート内の縦方向ソートフィールド値 (BY フィールド値) にリンクすることができます。

複合レイアウト構文では、実際の目次内容は、テキスト要素として記述されます。目次ページを使用する場合は、次の操作を行うことができます。

- 目次タイトルをカスタマイズし、テキストのフォーマットを一括設定する。
- 目次に含める各コンポーネントの目次レベル (TOC レベル) を指定する。
- 目次のページ番号を有効にし、目次の各要素に番号を付ける。また、各項目とページ番号の間にタブリーダー (点線) を挿入し、コンテンツを選択しやすくすることもできます。
- オブジェクトのプロパティをカスタマイズし、目次に表示するレポートとグラフを制御する。
- 目次にハイパーテキストリンクを使用することにより、項目をクリックして、ドキュメント内の特定のページに移動することができます。

注意：実行時に目次が複数ページにオーバーフローする場合は、目次全体の出力が完了するまで、先頭ページと同一のサイズが保持された状態ですべてのコンテンツが表示されます。

PPTX 複合レポートの目次生成

次の例は、PPTX 複合レポートの目次を作成します。各コンポーネントは目次の第 1 レベルに位置し、これらのコンポーネントには、第 1 レベルの項目下に 2 つのレベルの BY フィールドが存在します。

```
COMPOUND LAYOUT PCHOLD FORMAT PPTX
OBJECT=TOC, NAME='text1',
TEXT='<font face="ARIAL" size=10>Table of Contents</font>',
MARKUP=ON, TOC-NUMBERING=ON, POSITION=(0.854 0.854),
DIMENSION=(7.000 9.500), font='ARIAL', color=RGB(0 0 0),
size=10, METADATA=' TOCTITLE:Table of Contents', $
SECTION=S1, LAYOUT=ON, MERGE=OFF, ORIENTATION=PORTRAIT, $
PAGELAYOUT=1, $
COMPONENT=report1, TEXT='Sales By Product', TOC-LEVEL=1, BYTOC=2,
POSITION=(1 1), DIMENSION=(* *), $
COMPONENT=report2, TEXT='Sales By Region', TOC-LEVEL=1, BYTOC=2,
POSITION=(+0.00 +0.519), DIMENSION=(* *), RELATIVE-TO='report1',
RELATIVE-POINT=BOTTOM-LEFT, POSITION-POINT=TOP-LEFT, $
END
SET COMPONENT=report1
TABLE FILE GGSALES
SUM DOLLARS/F8M
BY CATEGORY
BY PRODUCT
BY REGION
BY ST
HEADING
"Sales by Category"
ON TABLE HOLD FORMAT PPTX
ON TABLE SET STYLE *
TYPE=REPORT, SQUEEZE=ON, $
ENDSTYLE
END
SET COMPONENT=report2
TABLE FILE GGSALES
SUM DOLLARS/F8M
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
HEADING
"Sales by Region"
ON TABLE HOLD FORMAT PPTX
ON TABLE SET STYLE *
TYPE=REPORT, SQUEEZE=ON, $
```

```
ENDSTYLE
END
COMPOUND END
```

出力では、各コンポーネントレポートは目次の第 1 レベルとして表示され、その下には、2 つのレベルのソートフィールドが表示されています。Sales By Product レポートで、BY フィールドは Category と Product です。Sales By Region レポートでは、BY フィールドは Region と State です。目次の各項目は、その値を含むページへのリンクになっています。

PowerPoint プレゼンテーションモードでは、目次ページの任意の項目をクリックすると、この項目を含むページが開きます。たとえば、Sales by Region レポートの 「Southeast」をクリックすると、次のページが表示されます。

```
Sales by Region
Region     State  Category  Product         Dollar Sales
─────────  ─────  ────────  ──────────────  ────────────
Northeast  CT     Coffee    Espresso            $279,373
                            Latte               $926,052
                  Food      Biscotti            $589,355
                            Croissant           $551,489
                            Scone               $283,874
                  Gifts     Coffee Grinder      $169,908
                            Coffee Pot          $208,209
                            Mug                 $392,967
                            Thermos             $221,827
           MA     Coffee    Capuccino           $174,344
                            Espresso            $248,356
                            Latte               $917,737
                  Food      Biscotti            $570,391
                            Croissant           $497,234
                            Scone               $332,486
                  Gifts     Coffee Grinder      $177,940
                            Coffee Pot          $184,119
                            Mug                 $401,944
                            Thermos             $203,435
           NY     Coffee    Capuccino           $208,756
                            Espresso            $322,378
                            Latte               $928,026
                  Food      Biscotti            $642,259
                            Croissant           $622,095
                            Scone               $290,811
                  Gifts     Coffee Grinder      $161,352
                            Coffee Pot          $198,452
                            Mug                 $349,300
                            Thermos             $178,836
Southeast  FL     Coffee    Capuccino           $317,027
                            Espresso            $256,539
                            Latte               $889,887
                  Food      Biscotti            $511,597
                            Croissant           $602,076
                            Scone               $311,836
                  Gifts     Coffee Grinder      $217,204
                            Coffee Pot          $212,057
                            Mug                 $409,466
                            Thermos             $195,526
           GA     Coffee    Capuccino           $352,161
                            Espresso            $317,389
                            Latte               $907,365
                  Food      Biscotti            $555,231
                            Croissant           $661,806
                            Scone               $273,420
                  Gifts     Coffee Grinder      $217,254
                            Coffee Pot          $232,552
                            Mug                 $355,447
                            Thermos             $227,482
```

PDF 複合レポートの目次生成

複合レイアウトレポート構文で、BOOKMARKS オブジェクトと TOC オブジェクトを指定することにより、ブックマーク (しおり)、目次ページ、またはその両方を生成することができます。さらに、コンポーネント定義で、目次レベル (TOC レベル) と説明を指定し、必要に応じて BYTOC も指定します。

TOC 属性

```
COMPOUND LAYOUT PCHOLD FORMAT PDF
[OBJECT=BOOKMARKS, $]
```

```
[OBJECT=TOC, NAME='text1',
   TEXT='<font face="font1" size=sz1>Table of Contents</font>',
   MARKUP=ON, TOC-NUMBERING={OFF|ON}, POSITION=(xy), [TOC-FILL=DOTS,]
   DIMENSION=(mn),
   font='font2', color={color|RGB(rgb)}, size=sz2,
   METADATA=' TOCTITLE:Table of Contents', $]
```

説明

OBJECT=BOOKMARKS

COMPONENT で指定した目次項目の PDF のしおりを生成します。

OBJECT=TOC

COMPONENT で指定した目次項目の目次ページを生成します。

NAME='text1'

目次ページ項目の名前を指定します。

TEXT='Table of Contents'

目次のタイトルとテキスト属性を指定します。この情報が正しく解釈されるには、MARKUP=ON 属性を指定する必要があります。テキストは折り返し表示されないため、テキスト要素の全体幅に適合する長さにする必要があります。

MARKUP=ON

目次タイトルで使用するマークアップタグを、テキストではなくフォーマットオプションとして解釈します。

TOC-NUMBERING={OFF|ON}

目次ページの項目に番号を付けるかどうかを指定します。デフォルト値は ON です。

POSITION=(xy)

ページのオブジェクトの X 座標および Y 座標を指定します。

DIMENSION=(ab)

オブジェクトの境界ボックスのサイズを指定します。

font='font2', color={color|RGB(rgb)}, size=sz2,

目次本文のテキスト属性を指定します。省略した場合、この属性は TEXT 属性から取得されます。

TOC-FILL=DOTS

目次項目とページ番号の間に、タブリーダーの点線を表示します。この属性を指定しない場合、目次項目とページ番号は、ブランクで区切られます。

コンポーネントエントリ

```
COMPONENT=component1, TITLE='title1', TOC-LEVEL=n, [BYTOC=m,]
   POSITION=(xy),DIMENSION=(ab), $
```

説明

component1

目次に含めるコンポーネントです。

title1

TOC 項目として使用するコンポーネントのタイトルです。

TOC-LEVEL=n

レポート、グラフ、またはページレイアウトオブジェクトの目次レベルを「n」として定義します。このオプションは、オブジェクトの目次内での階層順序を指定します。

0 = オブジェクトは、目次に表示されません。

1 = オブジェクトは、目次の第 1 レベルに表示されます。

2 = オブジェクトは、目次の第 2 レベルに表示されます。

BYTOC=m

現在のコンポーネントエントリに含める BY フィールド数を、「m」として指定します。

POSITION=(xy)

ページのオブジェクトの X 座標および Y 座標を指定します。

DIMENSION=(ab)

オブジェクトの境界ボックスのサイズを指定します。

目次使用上の注意

一般的な注意

- 目次項目は、選択したデータ要素のページ内での位置ではなく、ページ全体へのリンクとして設定されます。
- オーバーフローページ (DisplayOn=OVERFLOW-ONLY) の見出しまたは脚注で使用するレポートを、TOC や BYTOC の項目として含めることは推奨されません。目次にこれらのコンポーネントを含めた場合、これらのコンポーネントは、2 ページ目以降すべてのページ (ソートフィールド値ではなく、ページ数に基づくページレイアウトのページ) で繰り返されるため、重複する目次項目が生成されます。
- ドキュメントにレポートが統合されていない場合、目次として指定されたすべてのコンポーネントとページレイアウトが、目次に表示されます。
- 複合レポートの場合、目次には、統合されたソートフィールドへのリンクが、共通ソートフィールドに関連付けられた最初のページのリンクとして表示されます。

BYTOC の注意

- 指定された BYTOC 値が、複合レポートの BY フィールド数より大きい場合、この値は、デフォルト設定に変更され、有効な BY フィールド数の合計が使用されます。
- データソースから直接取得されたフィールドまたは DEFINE で作成されたフィールドを含めて、有効な BY フィールドは、BYTOC 項目として使用することができます。
- 複合レポート内で非表示が指定されている BY フィールド (NOPRINT フィールド) は、目次内では表示されます。
- TOC レベルの値は、第 1 レベル、第 2 レベル、第 3 レベルのように順に定義する必要があり、途中のレベルを省略することはできません。途中のレベルを省略すると、それより下位のしおりが表示されません。

しおりページおよび目次ページの作成

次の例には、2 つのコンポーネントレポートが含まれています。PDF のしおりと目次ページの両方が生成されます。どちらのコンポーネントも、目次の第 1 レベルで、いずれも第 1 レベルの項目下に、2 つのレベルの BY フィールドが存在します。

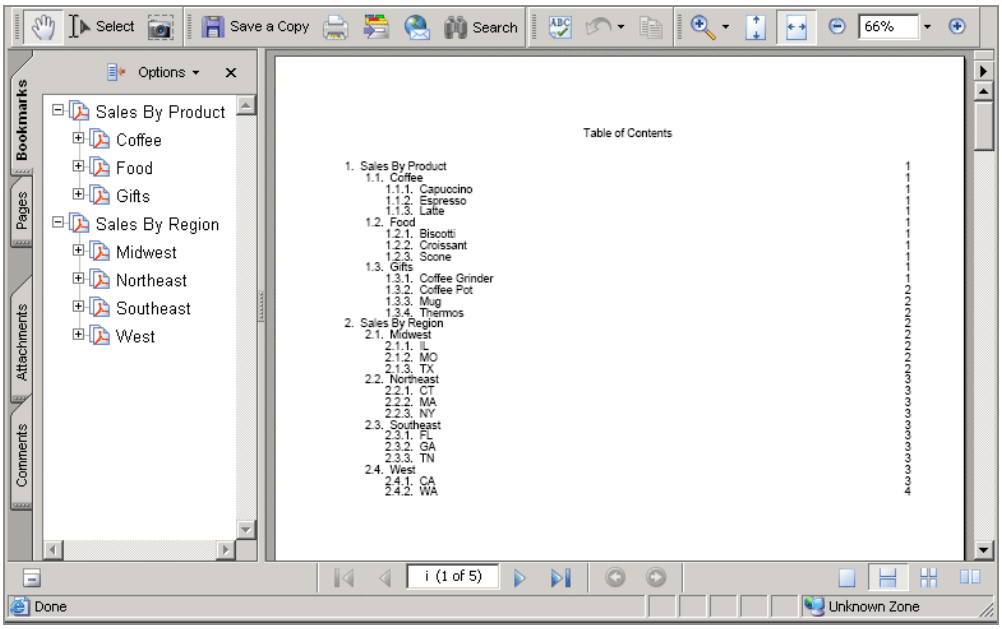
```
COMPOUND LAYOUT PCHOLD FORMAT PDF
OBJECT=BOOKMARKS, $
OBJECT=TOC, NAME='text1',
  TEXT='<font face="ARIAL" size=10>Table of Contents</font>',
  MARKUP=ON, TOC-NUMBERING=ON, POSITION=(0.854 0.854),
  DIMENSION=(7.000 9.500), font='ARIAL', color=RGB(0 0 0),
  size=10, METADATA=' TOCTITLE:Table of Contents', $
```

```
SECTION=S1, LAYOUT=ON, MERGE=OFF, ORIENTATION=PORTRAIT, $
PAGELAYOUT=1, $
COMPONENT=report1, TEXT='Sales By Product', TOC-LEVEL=1, BYTOC=2,
  POSITION=(1 1), DIMENSION=(* *), $
COMPONENT=report2, TEXT='Sales By Region', TOC-LEVEL=1, BYTOC=2,
  POSITION=(+0.00 +0.519), DIMENSION=(* *), RELATIVE-TO='report1',
  RELATIVE-POINT=BOTTOM-LEFT, POSITION-POINT=TOP-LEFT, $
END
```

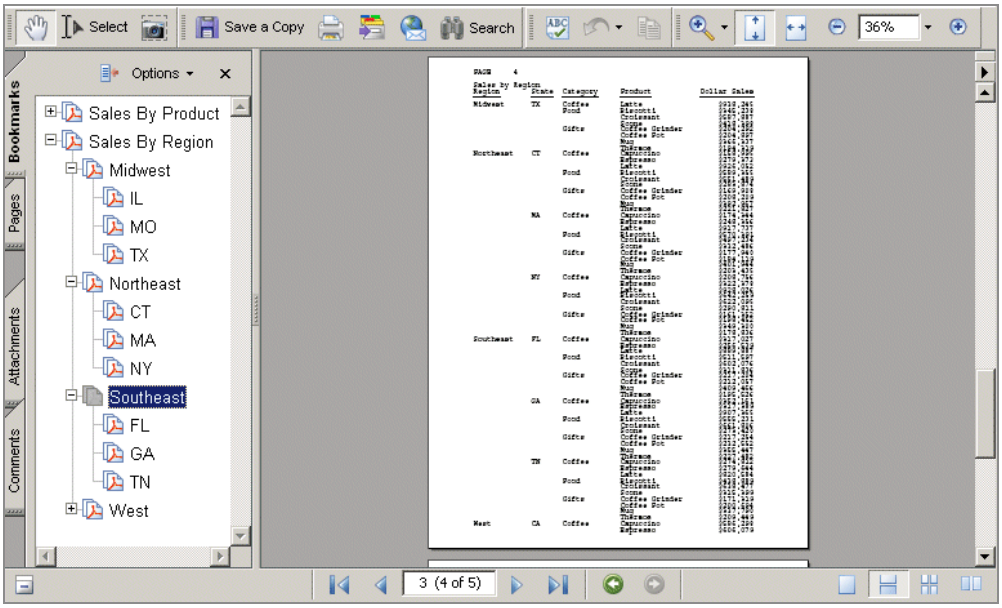
```
SET COMPONENT=report1
TABLE FILE GGSALES
SUM DOLLARS/F8M
BY CATEGORY
BY PRODUCT
BY REGION
BY ST
HEADING
"Sales by Category"
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, SQUEEZE=ON, $
ENDSTYLE
END
```

```
SET COMPONENT=report2
TABLE FILE GGSALES
SUM DOLLARS/F8M
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
HEADING
"Sales by Region"
ON TABLE HOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, SQUEEZE=ON, $
ENDSTYLE
END
COMPOUND END
```

出力で、各コンポーネントレポートは目次の第 1 レベルとして表示され、その下には、2 つのレベルのソートフィールドが表示されています。Sales By Product レポートで、BY フィールドは Category と Product です。Sales By Region レポートでは、BY フィールドは Region と State です。目次の各項目は、その値を含むページへのリンクとして機能します。



目次ページまたは [しおり] ウィンドウで任意の項目をクリックすると、その項目を含むページが開きます。たとえば、Sales by Region レポートの 「Southeast」をクリックすると、次のページが開きます。



複合 PDF レポートの作成

複合レポートは、単一 PDF ファイルに複数レポートを組み合わせます。最初の PDF レポートでフォーマットを指定することにより、異なるフォーマットの後続レポートでも統合する

ことができます。複合レポートを使用すると、異なるデータソースからデータを収集して 1 つのレポートに統合することができます。

作成したレポートは、Adobe Reader で複合 PDF レポートとして表示することができます。

この解析は、スタイルフォーマット (例、PDF、EXL2K、XLSX) でサポートされます。

ドリルスルー PDF 複合レポートの作成についての詳細は、ドリルスルー PDF 複合レポートの作成を参照してください。

複合レポートの表示

異なるレポートタイプで構成された複合レポートでは、次の構文を使用します。

```
SET COMPOUND= {OPEN|CLOSE} [NOBREAK]
```

または

```
ON TABLE SET COMPOUND {OPEN|CLOSE}
```

この構文を使用するとき、連続する各レポートの表示フォーマットを識別するために、次のコードを含む必要があることに注意します。

```
ON TABLE {PCHOLD|HOLD|SAVE} [AS name] FORMAT formatname
```

複合セット内のすべてのレポートが同じ (PDF) である場合、よりコンパクトな次の構文を使用できます。

```
ON TABLE {PCHOLD|HOLD|SAVE} [AS name] FORMAT {PDF|PS} {OPEN|CLOSE} [NOBREAK]
```

説明

name

ファイルの名前です。この名前は、複合レポートの最初のレポートから取得されます。最初のレポートに名前が指定されていない場合は、「HOLD」という名前が使用されます。

OPEN

最初のレポートで指定され、連結プロセスを開始します。OPEN 属性を含むレポートは、PDF フォーマットの必要があります。

CLOSE

最後のレポートで指定され、連結プロセスを終了します。

NOBREAK

改ページさせない場合に使用するオプションです。デフォルト設定では、各レポートは別ページに表示されます。

NOBREAK を使用すると、同じページにレポートを表示することができます。

注意

- 複合レポートをネストすることはできません。
- 複合レポートからの出力を保存することができます。詳細は、レポート出力の保存と再利用を参照してください。
- 複数枠レポートは、複合レポートでは使用できません。

複合 PDF レポートの作成

次の例は、複合レポートを作成し、3 つの個別 PDF レポート を 1 つに統合する方法を示しています。このフォーマット設定の要点は次のとおりです。

- レポート 1 では、ON TABLE PCHOLD FORMAT PDF OPEN を指定します。これにより、このレポートが先頭レポートとして定義され、複合レポート全体のフォーマットが PDF に設定されます。
- レポート 2 では、フォーマット設定の ON TABLE PCHOLD FORMAT PDF のみを指定します。
- レポート 3 では、ON TABLE PCHOLD FORMAT PDF CLOSE を指定します。これにより、このレポートが最終レポートとして定義されます。

この例では、すべてのレポートは、PDF フォーマットに設定されています。ただし、複合レポートを作成する際は、PDF フォーマットでなければならないのは最初のレポートのみであり、それに続くレポートは任意のスタイルフォーマットにすることができます。

レポート 1

```
SET PAGE-NUM=OFF
TABLE FILE CENTORD
HEADING
"Sales Report"
```

```
" "
SUM LINEPRICE
BY PRODCAT
ON TABLE SET STYLE *
TYPE=HEADING, SIZE=18, $
ENDSTYLE
ON TABLE PCHOLD FORMAT PDF OPEN NOBREAK
END
```

レポート 2

```
TABLE FILE CENTORD
HEADING
"Inventory Report"
" "
SUM QUANTITY
BY PRODCAT
ON TABLE SET STYLE *
TYPE=HEADING, SIZE=18, $
ENDSTYLE
ON TABLE PCHOLD FORMAT PDF NOBREAK
END
```

レポート 3

```
TABLE FILE CENTORD
HEADING
"Cost of Goods Sold Report"
" "
SUM LINE_COGS
BY PRODCAT
ON TABLE SET STYLE *
TYPE=HEADING, SIZE=18, $
ENDSTYLE
ON TABLE PCHOLD FORMAT PDF CLOSE
END
```

出力は PDF として表示されます。レポート 1 および 2 の構文には、NOBREAK コマンドが含まれることから、3 つのレポートが単一ページに表示されます。NOBREAK が指定されていない場合、各レポートはそれぞれ別のページに表示されます。

Sales Report

PRODCAT	Line Total
CD Players	$40,694,863.53
Camcorders	$333,884,927.52
Cameras	$9,718,752.55
DVD	$53,182,674.89
Digital Tape Recorders	$38,773,229.69
PDA Devices	$233,967,566.49
VCRs	$23,801,009.34

Inventory Report

PRODCAT	Quantity:
CD Players	308,274
Camcorders	950,700
Cameras	98,141
DVD	241,287
Digital Tape Recorders	524,027
PDA Devices	610,844
VCRs	170,977

Cost of Goods Sold Report

PRODCAT	Line Cost Of Goods Sold
CD Players	30,519,126.00
Camcorders	299,487,742.00
Cameras	8,944,891.00
DVD	47,275,593.00
Digital Tape Recorders	36,157,863.00
PDA Devices	202,123,656.00
VCRs	22,056,033.00

複合レポートへのグラフの埋め込み

ロゴや WebFOCUS グラフなど、GIF ファイルとして保存したグラフ を、複合レポートに埋め込むことができます。グラフファイルは、複合レポートの特定レポートに埋め込む必要があります。

グラフをグラフィックイメージとして保存するには、GRAPH リクエストに次の構文を含めます。

```
HOLD FORMAT GIF
```

グラフをイメージファイルとして保存する方法についての詳細は、グラフの作成を参照してください。

複合レポートにグラフを埋め込むには、サイズおよび位置とともに、イメージファイルをレポートのスタイルシートで指定する必要があります。レポート内のグラフの埋め込みおよび配置についての詳細は、レポートへのイメージの追加を参照してください。

複合レポート内のレポートフォーマットおよびグラフの組み合わせ

次のリクエストは、3 つのレポートタイプ (PDF、HTML、EXL2K) から複合レポートを生成し、各レポート内にグラフを埋め込みます。それぞれのグラフは、GRAPH リクエスト内で GIF ファイルとして保存されます。その後、グラフは埋め込まれているレポートのスタイルシート (TYPE=REPORT, IMAGE=*graphname*...) 内で識別、サイズ設定、配置されます。3 つの SET COMPOUND= 構文 (OPEN、NOBREAK、CLOSE) により、同一ページ上で 3 つのレポートが統合されます。次のリクエストでは、コードの主要行が太字で表示されています。

レポート 1

```
SET GRMERGE = ON
GRAPH FILE SHORT
SUM PROJECTED_RETURN AS 'Return on Investment'
BY HOLDER
ACROSS CONTINENT
ON GRAPH SET LOOKGRAPH 3D_BAR
ON GRAPH SET GRAPHEDIT SERVER
ON GRAPH HOLD AS SLSGRPH1 FORMAT GIF
END
```

```
SET COMPOUND='OPEN NOBREAK'
TABLE FILE SHORT
SUM PROJECTED_RETURN AS 'Return on Investment'
BY CONTINENT
BY HOLDER
HEADING
"Investment Report"
" "
```

```
ON TABLE SET STYLE *
TYPE=DATA, BACKCOLOR=( BY=B2 'SILVER' 'WHITE' ), $
TYPE=HEADING, SIZE=14, STYLE=BOLD, $
TYPE=REPORT, IMAGE=SLSGRPH1.gif, POSITION=(4.5 0.5), SIZE=(3.5 2.5), $
ENDSTYLE
```

```
ON TABLE PCHOLD FORMAT PDF

END
```

レポート 2

```
GRAPH FILE TRADES
SUM AMOUNT
BY CONTINENT
ON GRAPH SET LOOKGRAPH PIE
ON GRAPH SET GRAPHEDIT SERVER
ON GRAPH HOLD AS TRDSGR1 FORMAT GIF
END
```

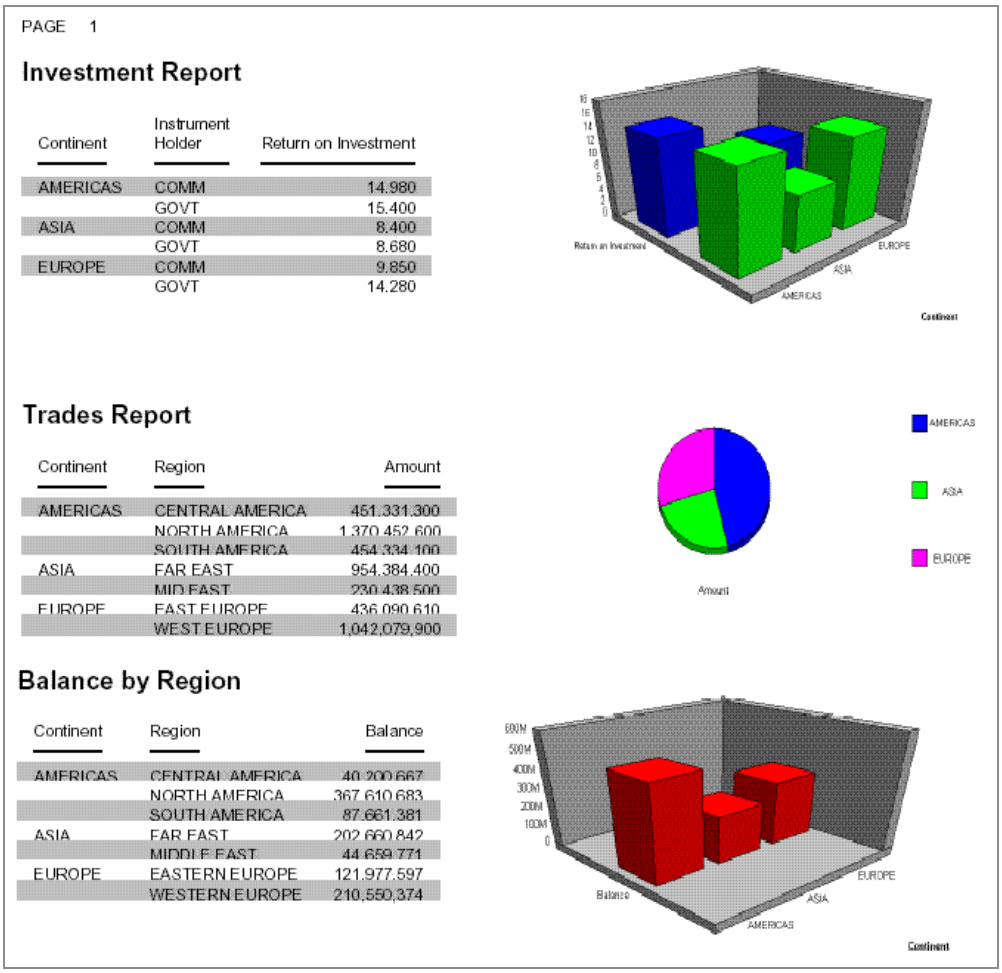
```
SET COMPOUND=NOBREAK
TABLE FILE TRADES
SUM AMOUNT AS 'Amount'
BY CONTINENT AS 'Continent'
BY REGION AS 'Region'
HEADING
"Trades Report"
" "
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
 TYPE=DATA, BACKCOLOR=( BY=B2 'SILVER' 'WHITE' ), $
TYPE=HEADING, SIZE=14, STYLE=BOLD, $
TYPE=REPORT, IMAGE=TRDSGR1.gif, POSITION=(4 3), SIZE=(4 2.5),
$
ENDSTYLE
ON TABLE PCHOLD FORMAT HTML
END
```

レポート 3

```
GRAPH FILE SHORT
SUM BALANCE
BY CONTINENT
ON GRAPH SET LOOKGRAPH 3D_BAR
ON GRAPH SET GRAPHEDIT SERVER
ON GRAPH SET STYLE *
TYPE=DATA, COLOR=RED,$
ENDSTYLE
ON GRAPH HOLD AS BALGR1 FORMAT GIF
END
```

```
SET COMPOUND=CLOSE
TABLE FILE SHORT
SUM BALANCE AS 'Balance'
BY CONTINENT AS 'Continent'
BY REGION AS 'Region'
HEADING
"Balance by Region"
" "
ON TABLE SET STYLE *
 TYPE=DATA, BACKCOLOR=( BY=B2 'SILVER' 'WHITE' ), $
TYPE=HEADING, SIZE=14, STYLE=BOLD, $
TYPE=REPORT, IMAGE=BALGR1.gif, POSITION=(4 6), SIZE=(4 2.5),
$
ENDSTYLE
ON TABLE PCHOLD FORMAT EXL2K
END
```

出力結果は次のとおりです。


PAGE 1

Investment Report

Continent	Instrument Holder	Return on Investment
AMERICAS	COMM	14,980
	GOVT	15,400
ASIA	COMM	8,400
	GOVT	8,680
EUROPE	COMM	9,850
	GOVT	14,280

Trades Report

Continent	Region	Amount
AMERICAS	CENTRAL AMERICA	451,331,300
	NORTH AMERICA	1,370,452,600
	SOUTH AMERICA	454,334,100
ASIA	FAR EAST	954,384,400
	MID EAST	230,438,500
EUROPE	EAST EUROPE	436,090,610
	WEST EUROPE	1,042,079,900

Balance by Region

Continent	Region	Balance
AMERICAS	CENTRAL AMERICA	40,200,667
	NORTH AMERICA	367,610,683
	SOUTH AMERICA	87,661,381
ASIA	FAR EAST	202,660,842
	MIDDLE EAST	44,659,771
EUROPE	EASTERN EUROPE	121,977,597
	WESTERN EUROPE	210,550,374

EXL2K による Excel 複合レポートの作成

Excel 複合レポートを使用すると、EXL2K 出力フォーマットで複数のワークシートレポートが生成されます。

Excel 複合レポートの構文は、PDF 複合レポートの構文と同一です。デフォルト設定では、複合レポートの各コンポーネントレポートは、新しい Excel ワークシートとして作成されます (PDF で新しいページが作成されることと同様)。NOBREAK キーワードを使用する場合は、次のレポートが同一ワークシート上の現在のレポートの下に作成されます (PDF の同一ページでのレポートの開始と同様)。

出力は、PCHOLD を使用してクライアントに送信した場合または HOLD を使用してファイルに保存した場合のいずれにおいても、Microsoft の Web アーカイブフォーマットで生成されます。このフォーマットは、Excel の [名前を付けて保存] ダイアログボックスの [単一ファイル Web ページ] に対応します。このフォーマットのファイル拡張子は、.mht または .mhtml

です。WebFOCUS では EXL2K レポートと同一の拡張子 .xht を使用します。レポートは常に 1 つのファイルで出力されるため、ReportCaster で容易に配信することができます。

Excel 複合レポートのコンポーネントは、FORMULA レポートまたは PIVOT レポート (制約あり) のいずれかです。コンポーネントに目次レポートを使用することはできません。

注意：Excel 2002 (Office XP) またはそれ以降をインストールしておく必要があります。Excel 複合レポートは Web アーカイブファイルフォーマットをサポートしないため、以前のバージョンの Excel では、Excel 複合レポートは動作しません。

OPEN、CLOSE、NOBREAK キーワードおよび SET COMPOUND の使用に関するガイドライン

PDF の場合と同様に、Excel 複合レポートを制御するには、OPEN、CLOSE、NOBREAK キーワードを使用します。これらのキーワードは、HOLD または PCHOLD コマンド、あるいは別の SET COMPOUND コマンドと併用することができます。

- OPEN キーワードは、一連のコンポーネントレポートの先頭レポートで使用し、出力するレポートが複合レポートであることを指定します。
- CLOSE キーワードを使用して、複合レポートの最終レポートを指定します。
- NOBREAK キーワードを使用して、次のレポートが現在のレポートと同一のワークシートに配置されることを指定します。このキーワードによる指定がない場合、デフォルト設定では、次のレポートは新しいワークシートに配置されます。

  NOBREAK キーワードは、先頭レポートで OPEN キーワードと併用したり、先頭レポートから最終レポートまでの個々のレポートに単独で使用したりすることができます。CLOSE キーワードを指定した最終レポートには後に続くレポートが存在しないため、NOBREAK キーワードとの併用は不適切です。
- 複合レポートのキーワードである OPEN、CLOSE、NOBREAK を HOLD/PCHOLD 構文と併用する場合は、FORMAT EXL2K のすぐ後に配置するか、FORMULA または PIVOT などの追加キーワードの前に配置する必要があります。たとえば、次のように指定します。

  - `ON TABLE PCHOLD FORMAT EXL2K OPEN`
  - `ON TABLE HOLD AS MYHOLD FORMAT EXL2K OPEN NOBREAK`

    - `ON TABLE PCHOLD FORMAT EXL2K NOBREAK FORMULA`
    - `ON TABLE HOLD FORMAT EXL2K CLOSE PIVOT PAGEFIELDS COUNTRY`
- PDF 複合レポートの場合と同様に、SET COMPOUND を使用して、これらの複合レポートキーワードを入れ替えて指定することができます。
    - `SET COMPOUND = OPEN`
    - `SET COMPOUND = 'OPEN NOBREAK'`
    - `SET COMPOUND = NOBREAK`
    - `SET COMPOUND = CLOSE`

EXL2K による Excel 複合レポートの作成に関するガイドライン

- **PivotTable と NOBREAK**　複合レポートに PivotTable レポートを表示させることはできますが、NOBREAK キーワードを使用した同一ワークシート上の別のレポートと結合させることはできません。
- **ワークシートの名前指定**　ワークシートのタブ名は、デフォルトで Sheet1、Sheet2 のように設定されています。オプションとして、スタイルシートに TITLETEXT キーワードを使用して、異なる名前でワークシートのタブ名を指定することができます。以下はその例です。

  ```
  TYPE=REPORT, TITLETEXT='Summary Report', $
  ```

  Excel では、ワークシートのタイトルの最大長は 31 バイトに限定されています。特殊文字の「:」、「?」、「*」、「/」を使用することはできません。
- **ファイル名とフォーマット**　出力ファイル名 (AS 名。デフォルト値は HOLD) は、複合レポートの先頭レポートから取得することができます (OPEN キーワードが指定されたレポート)。先頭レポート以降の出力ファイル名は無視されます。

  複合レポートの先頭コンポーネントレポートで使用する HOLD FORMAT 構文は、フォーマットの種類に関係なく、先頭レポート以降のすべてのレポートに適用されます。
- **NOBREAK キーワードの動作**　NOBREAK キーワードを指定した場合、NOBREAK キーワードを含むレポートの最終行のすぐ後の行から次のレポートが表示されます。レ

ポート間で余分なスペースが必要な場合は、FOOTING または ON TABLE SUBFOOT を前のレポートに配置したり、HEADING または ON TABLE SUBHEAD を次のレポートに配置したりすることで対応します。デフォルト設定でブランク行が追加された場合は、それを削除することができないため、この方法で調整します。

EXL2K による単純な複合レポートの作成

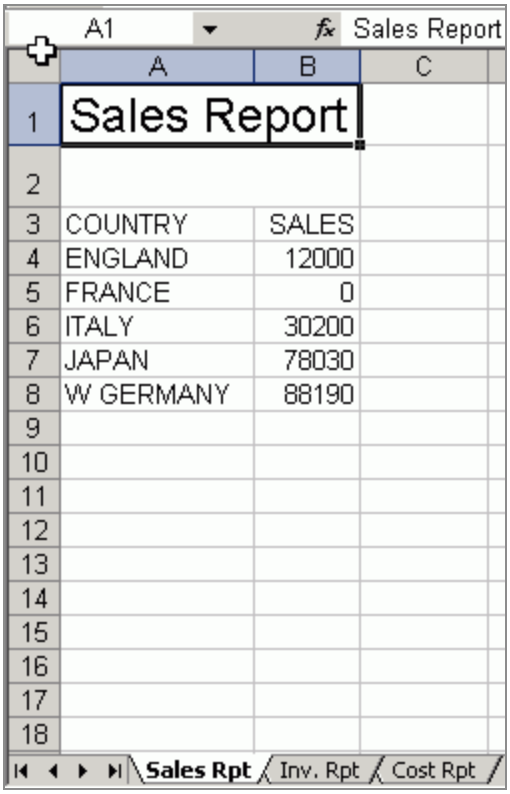
```
SET PAGE-NUM=OFF
TABLE FILE CAR
HEADING
"Sales Report"
" "
SUM SALES
BY COUNTRY
ON TABLE SET STYLE *
type=report, titletext='Sales Rpt', $
type=heading, size=18, $
ENDSTYLE
ON TABLE PCHOLD AS EX1 FORMAT EXL2K OPEN
END
```

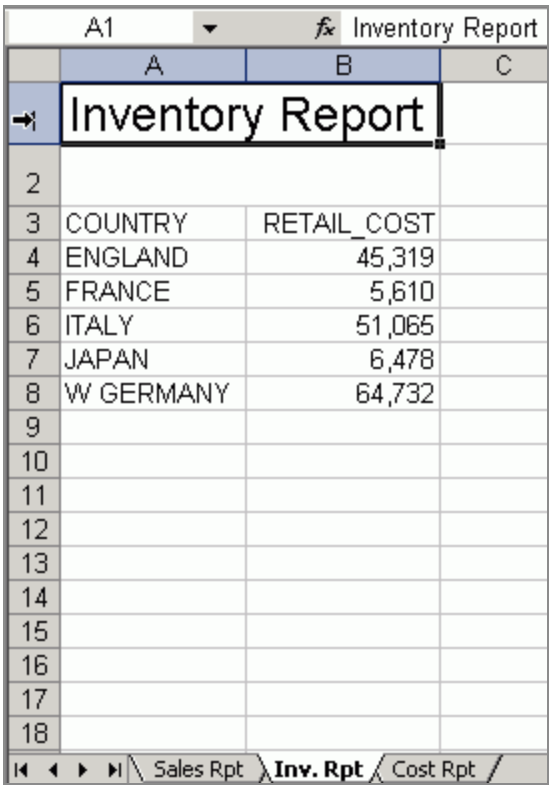
```
TABLE FILE CAR
HEADING
"Inventory Report"
" "
SUM RC
BY COUNTRY
ON TABLE SET STYLE *
type=report, titletext='Inv.Rpt', $
type=heading, size=18, $
ENDSTYLE
ON TABLE HOLD AS EX1 FORMAT EXL2K
END
```

```
TABLE FILE CAR
HEADING
"Cost of Goods Sold Report"
" "
SUM DC
BY COUNTRY
ON TABLE SET STYLE *
```

```
type=report, titletext='Cost Rpt', $
type=heading, size=18, $
ENDSTYLE
ON TABLE HOLD AS EX1 FORMAT EXL2K CLOSE
END
```

Excel ワークシートの各タブの出力は次のとおりです。

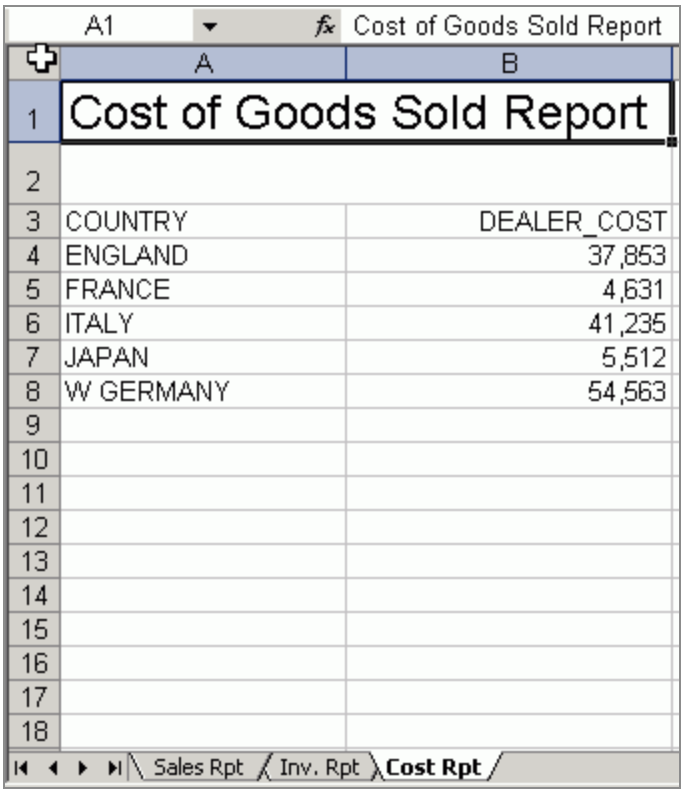




A1 fx Inventory Report

	A	B	C
1	Inventory Report		
2			
3	COUNTRY	RETAIL_COST	
4	ENGLAND	45,319	
5	FRANCE	5,610	
6	ITALY	51,065	
7	JAPAN	6,478	
8	W GERMANY	64,732	
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			

Sales Rpt | Inv. Rpt | Cost Rpt



	A	B
1	Cost of Goods Sold Report	
2		
3	COUNTRY	DEALER_COST
4	ENGLAND	37,853
5	FRANCE	4,631
6	ITALY	41,235
7	JAPAN	5,512
8	W GERMANY	54,563
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		

PivotTable および数式による複合レポートの作成

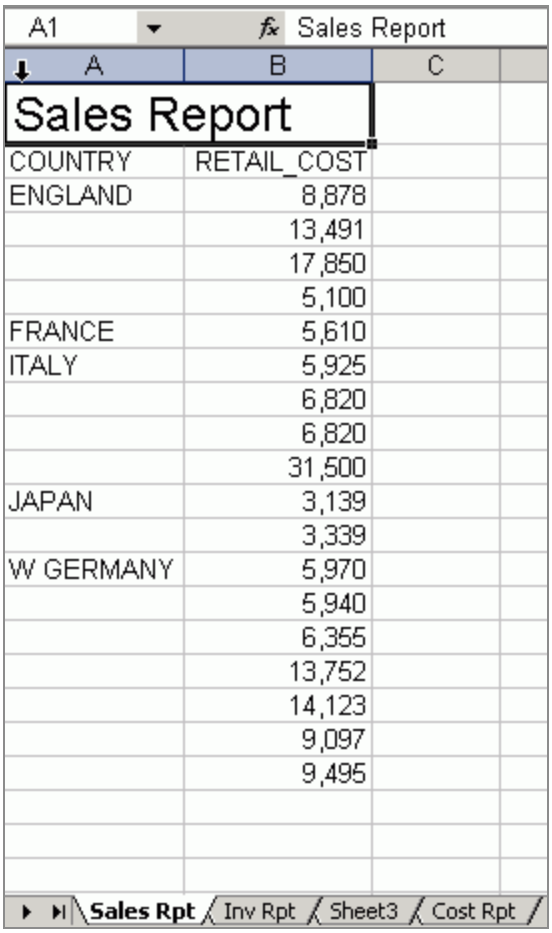
```
SET PAGE-NUM=OFF
TABLE FILE CAR
HEADING
"Sales Report"
" "
PRINT RCOST
BY COUNTRY
ON TABLE SET STYLE *
type=report, titletext='Sales Rpt', $
type=heading, size=18, $
ENDSTYLE
ON TABLE PCHOLD AS PIV1 FORMAT EXL2K OPEN
END
```

```
TABLE FILE CAR
HEADING
"Inventory Report"
" "
PRINT SALES
BY COUNTRY
ON TABLE SET STYLE *
type=report, titletext='Inv.Rpt', $
type=heading, size=18, $
ENDSTYLE
ON TABLE HOLD AS PPPP FORMAT EXL2K PIVOT
PAGEFIELDS TYPE SEATS
CACHEFIELDS MODEL MPG RPM
END
```

```
TABLE FILE CAR
SUM RCOST
BY COUNTRY BY CAR BY MODEL BY TYPE BY SEATS SUMMARIZE
ON MODEL SUB-TOTAL
ON TABLE HOLD AS XFOCB FORMAT EXL2K FORMULA
END
```

```
TABLE FILE CAR
HEADING
"Cost of Goods Sold Report"
" "
PRINT DCOST
BY COUNTRY
ON TABLE SET STYLE *
type=report, titletext='Cost Rpt', $
type=heading, size=18, $
ENDSTYLE
ON TABLE HOLD AS ONE FORMAT EXL2K CLOSE PIVOT
PAGEFIELDS RCOST
CACHEFIELDS MODEL TYPE SALES ACCEL SEATS
END
```

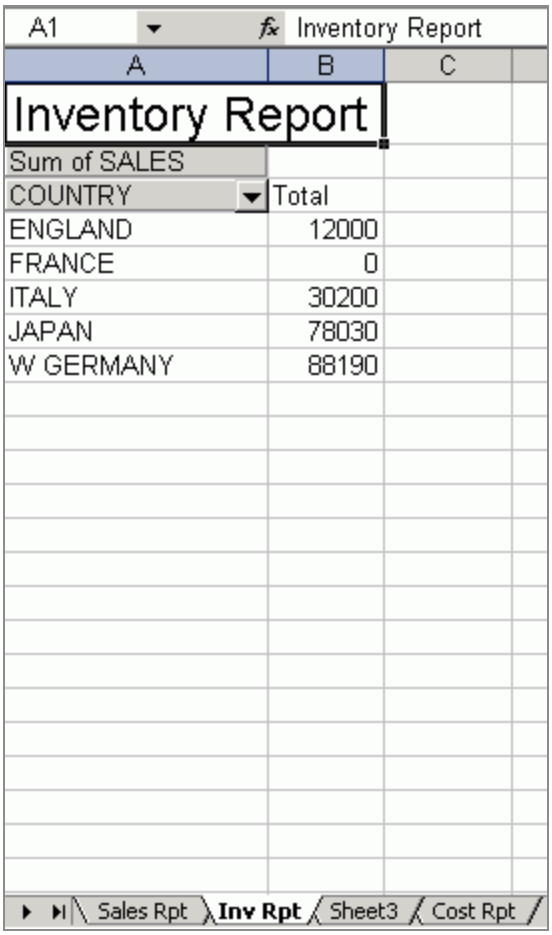
Excel ワークシートの各タブの出力は次のとおりです。



A1 fx Sales Report

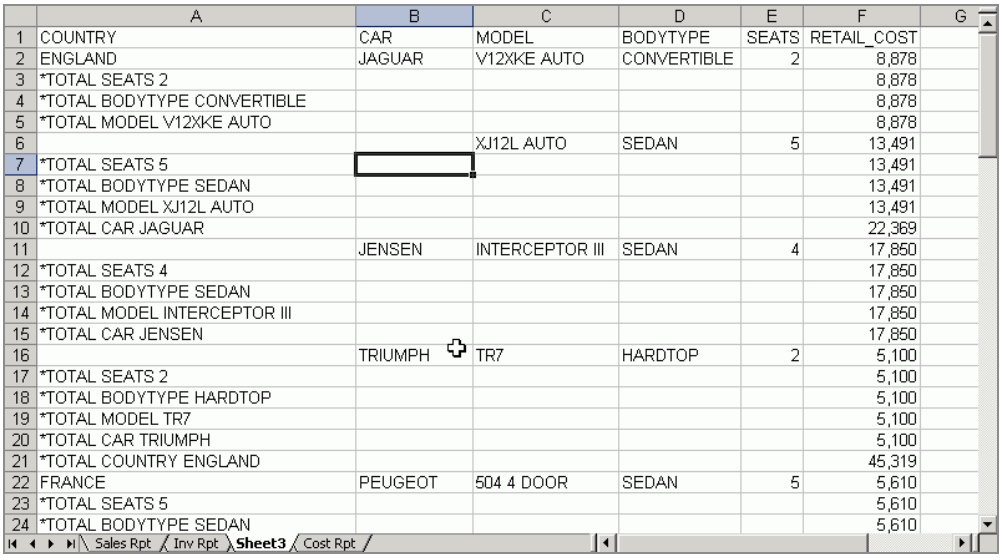
	A	B	C
1	Sales Report		
2	COUNTRY	RETAIL_COST	
3	ENGLAND	8,878	
4		13,491	
5		17,850	
6		5,100	
7	FRANCE	5,610	
8	ITALY	5,925	
9		6,820	
10		6,820	
11		31,500	
12	JAPAN	3,139	
13		3,339	
14	W GERMANY	5,970	
15		5,940	
16		6,355	
17		13,752	
18		14,123	
19		9,097	
20		9,495	
21			
22			
23			

Sales Rpt | Inv Rpt | Sheet3 | Cost Rpt



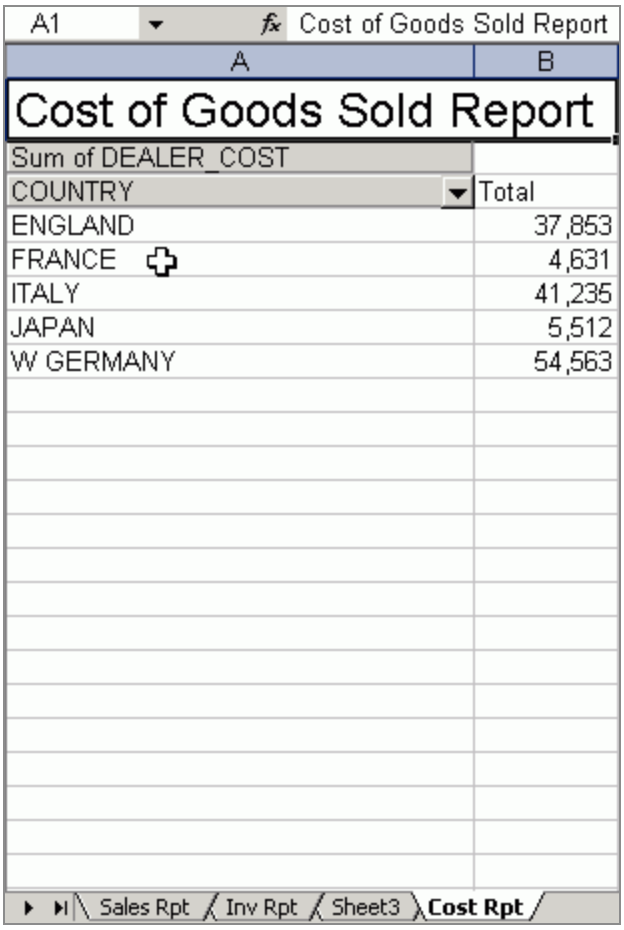
	A	B	C
	Inventory Report		
	Sum of SALES		
	COUNTRY	Total	
	ENGLAND	12000	
	FRANCE	0	
	ITALY	30200	
	JAPAN	78030	
	W GERMANY	88190	

Sales Rpt | Inv Rpt | Sheet3 | Cost Rpt



	A	B	C	D	E	F	G
1	COUNTRY	CAR	MODEL	BODYTYPE	SEATS	RETAIL_COST	
2	ENGLAND	JAGUAR	V12XKE AUTO	CONVERTIBLE	2	8,878	
3	*TOTAL SEATS 2					8,878	
4	*TOTAL BODYTYPE CONVERTIBLE					8,878	
5	*TOTAL MODEL V12XKE AUTO					8,878	
6			XJ12L AUTO	SEDAN	5	13,491	
7	*TOTAL SEATS 5					13,491	
8	*TOTAL BODYTYPE SEDAN					13,491	
9	*TOTAL MODEL XJ12L AUTO					13,491	
10	*TOTAL CAR JAGUAR					22,369	
11		JENSEN	INTERCEPTOR III	SEDAN	4	17,850	
12	*TOTAL SEATS 4					17,850	
13	*TOTAL BODYTYPE SEDAN					17,850	
14	*TOTAL MODEL INTERCEPTOR III					17,850	
15	*TOTAL CAR JENSEN					17,850	
16		TRIUMPH	TR7	HARDTOP	2	5,100	
17	*TOTAL SEATS 2					5,100	
18	*TOTAL BODYTYPE HARDTOP					5,100	
19	*TOTAL MODEL TR7					5,100	
20	*TOTAL CAR TRIUMPH					5,100	
21	*TOTAL COUNTRY ENGLAND					45,319	
22	FRANCE	PEUGEOT	504 4 DOOR	SEDAN	5	5,610	
23	*TOTAL SEATS 5					5,610	
24	*TOTAL BODYTYPE SEDAN					5,610	

Sales Rpt | Inv Rpt | Sheet3 | Cost Rpt



NOBREAK キーワードによる複合レポートの作成

次の例では、NOBREAK が先頭レポートと 3 つ目のレポートに設定されているため、最初の 2 つのレポートが 1 つ目のワークシートに、最後の 2 つのレポートが 2 つ目のワークシートにそれぞれ表示されています。

```
TABLE FILE GGSALES
HEADING
"Report 1:Coffee - Budget"
SUM BUDDOLLARS BUDUNITS COLUMN-TOTAL AS 'Total'
BY REGION
IF CATEGORY EQ Coffee
ON TABLE PCHOLD FORMAT EXL2K OPEN NOBREAK
ON TABLE SET STYLE *
```

```
type=report, font=Arial, size = 10, style=normal, $
type=title, style=bold, $
type=heading,   size=12, style=bold, color=blue, $
type=grandtotal,  style=bold,  $
ENDSTYLE
END
```

```
TABLE FILE GGSALES
HEADING
" "
"Report 2:Coffee - Actual "
SUM DOLLARS UNITS COLUMN-TOTAL AS 'Total'
BY REGION
IF CATEGORY EQ Coffee
ON TABLE PCHOLD FORMAT EXL2K
ON TABLE SET STYLE *
type=report, font=Arial, size=10, style=normal, $
type=grandtotal,  style=bold, $
type=heading,  size=12, style=bold, color=blue, $
ENDSTYLE
END
```

```
TABLE FILE GGSALES
HEADING
"Report 3:Food - Budget"
SUM BUDDOLLARS BUDUNITS COLUMN-TOTAL AS 'Total'
BY REGION
IF CATEGORY EQ Food
ON TABLE PCHOLD FORMAT EXL2K NOBREAK
ON TABLE SET STYLE *
type=REPORT, font=Arial, size=10, style=normal, $
type=HEADING,  style=bold, size=12, color=blue, $
type=title, style=bold, $
type=grandtotal,  style=bold,  $
ENDSTYLE
END
```

```
TABLE FILE GGSALES
HEADING
" "
"Report 4:Food - Actual"
SUM DOLLARS UNITS COLUMN-TOTAL AS 'Total'
BY REGION
IF CATEGORY EQ Food
```

```
ON TABLE PCHOLD FORMAT EXL2K CLOSE
ON TABLE SET STYLE *
type=report, font=Arial, size=10,  $
type=title, style=bold, $
type=heading,size=12, style=bold,  color=blue,$
type=grandtotal, style=bold,  $
ENDSTYLE
END
```

出力結果は次のとおりです。

	A	B	C
1	**Report 1: Coffee - Budget**		
2	**Region**	**Budget Dollars**	**Budget Units**
3	Midwest	4086032	335526
4	Northeast	4252462	335920
5	Southeast	4431429	355693
6	West	4523963	358784
7	**Total**	**17293886**	**1385923**
8			
9	**Report 2: Coffee - Actual**		
10	Region	Dollar Sales	Unit Sales
11	Midwest	4178513	332777
12	Northeast	4164017	335778
13	Southeast	4415408	350948
14	West	4473517	356763
15	**Total**	**17231455**	**1376266**

	A	B	C
1	**Report 3: Food - Budget**		
2	**Region**	**Budget Dollars**	**Budget Units**
3	Midwest	4220721	339263
4	Northeast	4453907	351431
5	Southeast	4409288	351509
6	West	4183244	335361
7	**Total**	**17267160**	**1377564**
8			
9	**Report 4: Food - Actual**		
10	**Region**	**Dollar Sales**	**Unit Sales**
11	Midwest	4338271	341414
12	Northeast	4379994	353368
13	Southeast	4308731	349829
14	West	4202337	340234
15	**Total**	**17229333**	**1384845**

ドリルスルーリンクを使用した PDF 複合レポートの作成

ビジネスレポートを作成する際に一般的に使用する手法は、相互に関連する 2 つのレポートを作成することです。

- **概要レポート**　特定のカテゴリに属する概要情報が表示されます。たとえば、企業のアカウント情報として総合収支と売上高を集計したデータがこれに該当します。
- **詳細レポート**　概要レポートで指定した特定のフィールドに関して、概要レポートのフィールド値の計算に使用したすべてのコンポーネント値が詳細レポートに表示されます。

ドリルスルーを使用して、これら 2 種類のレポートのデータを簡単に関連付けることができます。たとえば、概要レポートに表示されたアカウントの 1 つで異常な数字を発見した場合、その数字の背景にある特定のデータを詳細レポートで調査することができます。

複合ドリルスルーレポートには、次の 2 つの形式があります。

- **従来のドリルスルー複合レポート**　従来の複合レポート構文 (OPEN および CLOSE オプションを指定した PCHOLD) を使用して複合レポートを作成します。
- **ドリルスルー複合レイアウトレポート**　ドキュメント構文の宣言を使用して複合プロシジャを作成し、ハイパーリンクで関連付けるレポートを定義します。

この両方のタイプの複合ドリルスルーレポートでは、PDF ファイル内のレポート間のハイパーリンクを作成する構文は同一です。

ドリルスルーとドリルダウンの比較

ドリルダウンを使用して、概要レポートを作成する際にハイパーリンクを追加し、それをクリックすると詳細データが表示されるようなレポートを作成することができます。ドリルダウンは動的に実装されます。つまり、ハイパーリンクをクリックした段階で新しいレポートが実行されます。通常、詳細レポートには、概要レポートで指定したフィールドに対する詳細データのみが表示されます。

これに対して、ドリルスルーは静的なレポートです。ドリルスルーにより、概要レポートと詳細レポートの両方を含んだ PDF ドキュメントが作成されます。詳細レポートには、概要レポートの所定のフィールドの詳細データがすべて含まれています。ドリルスルーのハイパーリンクをクリックすると、PDF ファイル内部のリンク先に移動します。他のレポートは実行されません。PDF ファイルはディスクに保存したり、ReportCaster で配信したりすることが

できます。このファイルを Adobe Reader で開くと、すべてのドリルスルー機能が保持されています。

ドリルスルーでは、レポートの外観やハイパーリンクの位置を自由に設定することができます。

- ドリルスルーハイパーリンクは、データ行だけではなく、見出しや脚注などに表示することもできます。
- WebFOCUS スタイルシートを使用してレポートのフォーマットを設定できます。
- 色、フォント、下線などを使用してハイパーリンクを明確に表示することができます。
- 1 つのレポート内に、ドリルスルーと従来のドリルダウンのハイパーリンクを共存させることができます。
- ドリルスルーで作成した PDF ファイルに、複数のレポートを含めることができます。

他の機能との併用

ドリルスルーを WebFOCUS の他の機能と併用することができます。

- リンク付きのドリルスルーレポートを含む複合レポートには、ドリルスルーレポートの前後どちらにでも、関連性のないレポートを含めることもできます。たとえば、複合レポートパッケージに、グラフが埋め込まれた PDF レポートを追加することができます。
- ドリルスルーハイパーリンクを含む PDF レポートに、ドリルダウンおよび URL のハイパーリンクを追加することができます。
- ドリルスルーレポートは 1 つの PDF ファイルにまとめられた標準の PDF 複合レポートであるため、ReportCaster を使用して配信することができます。
- DRILLTHROUGH 構文を含むレポートは、HTML や EXL2K などのスタイルが設定されたすべての出力フォーマットで表示することができます。これらのフォーマットでは、DRILLTHROUGH 構文は無視されます。
- ドリルスルーにより、相互に関連する複数のレポートを 1 つの PDF 複合レポートにまとめて、概要情報から詳細情報にいたるすべての情報を簡単にすばやく検索するプロセスが自動化されます。ドリルスルー構文を使用して、概要レポートの項目から詳細レポートの関連項目へ移動するハイパーリンクを設定します。

PDF 複合レポートでドリルスルーを作成するには

PDF 複合レポートでドリルスルーを作成するには、次の手順を実行します。

手順

1. 概要レポートを作成し、DRILLTHROUGH ハイパーリンクを設定します。
2. 詳細レポートを作成し、ソート値をハイパーリンクフィールドの値と一致させます。
3. 複合レイアウトレポートを作成してコンポーネントレポートのそれぞれにページレイアウトを設定し、さらに、呼び出し元のレポートで DRILLMAP 属性を定義し、ドリルスルーハイパーリンクのターゲットを指定します。

PDF 複合レポートでドリルスルーを作成するには

PDF 複合レポートでドリルスルーを作成するには、次の手順を実行します。

手順

1. 概要レポートを作成します。
2. 詳細レポートを作成します。
3. これらのレポートをハイパーリンクで関連付けます。
4. 概要レポートと詳細レポートを結合して、PDF 複合レポートを作成します。

ドリルスルーハイパーリンクの指定

```
TYPE=type, [element,] [styling_attributes,]
     DRILLTHROUGH={DOWN|FIRST}(link_fields) , $
```

説明

type

次のスタイルシートタイプのいずれかです。

- DATA
- HEADING
- FOOTING
- SUBHEAD
- SUBFOOT
- SUBTOTAL
- RECAP

element

WebFOCUS スタイルシートとドリルスルーレポートで使用可能な 1 つまたは複数の識別要素です。1 つの要素で、特定のフィールド (例、COLUMN=PRODUCT) または見出し項目 (例、LINE=2、OBJECT=field、ITEM=3) を指定することができます。

styling_attributes

オプションとして、ハイパーリンクの外観を指定します (例、COLOR=RED, STYLE=BOLD)。

DOWN

次のレポート (下方向) にリンクさせます。

FIRST

先頭のドリルスルーレポートにリンクさせます。

link_fields

リンクフィールドの組み合わせを指定します。複数の組み合わせがある場合は、それぞれをブランクで区切ります。次のフォーマットで指定します。

T1=S1 T2=S2 T3=S3...

T1、T2、T3 はターゲットレポート (リンク先) の参照フィールドを表し、S1、S2、S3 はソースレポート (現行) の参照フィールドを表しています。組み合わせを 4 つ以上指定することもできます。

参照フィールドには、フィールド名または WebFOCUS スタイルシート構文で有効な記号であれば任意のものを使用することができます (例、Bn、Cn、Pn、Nn、An、下付きフィールド名)。

構文の順序は、ドリルダウン構文と同様に、パラメータの組で右辺に現行 (ソース) レポートの参照フィールドを指定し、左辺にドリルダウン (ターゲット) プロシジャのダイアログマネージャ変数名を指定します。

ターゲットレポートの参照フィールドが、ソースレポートの参照フィールドと一致する場合、使用する参照フィールドを 1 つにまとめることができます (例、COUNTRY=COUNTRY の代わりに COUNTRY)。

ドリルスルーパイパーリンクの指定

次のスタイルシート宣言は、PRODUCT フィールドの各 DATA 行にハイパーリンクを配置し、CATEGORY と PRODUCT が次のレポートへのリンクフィールドで、アクションが DOWN であることを指定します。これにより、ハイパーリンクの 1 つをクリックすることで、これらの 2 つのリンクフィールドに対応する値を持つ次のレポートに移動できるようになります。また、ハイパーリンクのデフォルトの外観である青色の下線付きのテキストを使用します。詳細レポートのターゲットフィールドの名前が概要レポートでの名前と同一であることから、CATEGORY=CATEGORY の代わりに CATEGORY という表記を使用します。

```
TYPE=DATA, COLUMN=PRODUCT, DRILLTHROUGH=DOWN(CATEGORY PRODUCT), $
```

複合レイアウトレポートへのハイパーリンク関係設定

ターゲットレポートは、呼び出し元レポートで、COMPONENT 宣言の DRILLMAP 属性で指定します。

```
DRILLMAP=((L1 targetreport))
```

説明

L1

リンク ID です。

targetreport

ハイパーリンクのリンク先コンポーネント名です。

注意：DRILLMAP の値を二重括弧で囲む必要があります。

DRILLMAP 属性を使用したサンプルコンポーネント宣言

次の REPORT1 の COMPONENT 宣言は、REPORT2 を指定する DRILLMAP を指定します。

```
COMPONENT='REPORT1', TEXT='REPORT1', TOC-LEVEL=2,
DRILLMAP=((L1 REPORT2)), POSITION=(0.750 1.083), DIMENSION=(7.000 3.167),
METADATA='Z-INDEX:100; LEFT:0.75in; OVERFLOW: auto; WIDTH:7in;
POSITION: absolute; TOP:1.083in; HEIGHT:3.167in', $
```

次の REPORT2 の COMPONENT 宣言は、REPORT1 を指定する DRILLMAP を指定します。

```
COMPONENT='REPORT2', TEXT='REPORT2', TOC-LEVEL=2,
DRILLMAP=((L1 REPORT1)), POSITION=(0.500 0.667), DIMENSION=(7.417 7.000),
METADATA='Z-INDEX:100; LEFT:0.5in; OVERFLOW: auto; WIDTH:7.417in;
POSITION: absolute; TOP:0.667in; HEIGHT:7in', $
```

ドリルスルー使用上の注意

- バージョン 8.2 SP01 以降では、ドリルスルーターゲットを含む個別のコンポーネントレポートを、スタンドアロンとして実行することができます。メッセージビューアに、複合レポート以外ではドリルスルーが無効になることを示す警告メッセージが表示されます。
- 1 つのレポートで指定可能なドリルスルー動作は 1 つのみです。
- ドリルスルー動作の格納を指定するフィールドは、ターゲットレポートにも存在する必要があります。

- ドリルスルーリンクが格納されている元のレポートは、ターゲットレポートより前に、(Reporting Server によって) 出力される必要があります。レポートの順序は、ユーザによって処理されます。
- ライブドリルスルーリンクは、PDF 出力でのみ生成されます。DRILLTHROUGH 構文を含むレポートは、HTML や EXL2K などのスタイルが設定されたすべての出力フォーマットで表示することができます。これらのフォーマットでは、DRILLTHROUGH 構文は無視されます。
- ドリルスルーは、(TABLE) コマンドでのみサポートされます。

サンプルのドリルスルー PDF 複合レポート

次の例は、ドリルスルー構文を使用して、概要レポートと詳細レポートで構成された複合レポートを作成し、これらのレポート間を移動できるようにする方法を示しています。

概要レポートの作成 (手順 1)

次の構文により、概要レポートのサンプルが作成されます。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS BY CATEGORY BY PRODUCT
ON TABLE PCHOLD FORMAT PDF
END
```

出力結果は次のとおりです。

```
PAGE     1

Category      Product           Unit Sales  Dollar Sales

Coffee        Capuccino             189217       2381590
              Espresso              308986       3906243
              Latte                 878063      10943622
Food          Biscotti              421377       5263317
              Croissant             630054       7749902
              Scone                 333414       4216114
Gifts         Coffee Grinder        186534       2337567
              Coffee Pot            190695       2449585
              Mug                   360570       4522521
              Thermos               190081       2385829
```

詳細レポートの作成 (手順 2)

次の構文により、詳細レポートのサンプルが作成されます。

これは、出力結果の 1 ページ目です。

```
SET SQUEEZE=ON
TABLE FILE GGSALES
SUM UNITS BUDUNITS DOLLARS
BY CATEGORY NOPRINT BY PRODUCT NOPRINT
ON CATEGORY PAGE-BREAK
HEADING CENTER
"Category: <CATEGORY"
" "
ON PRODUCT SUBHEAD
"**** Product: <PRODUCT"
ON PRODUCT SUBFOOT
" "
"<25 **** Return to Summary ****"
ON PRODUCT PAGE-BREAK
BY REGION BY CITY
ON TABLE PCHOLD FORMAT PDF
END
```

```
PAGE     1

                        Category: Coffee

Region     City           Unit Sales  Budget Units  Dollar Sales

**** Product: Capuccino
Northeast  Boston              15358         15672        174344
           New Haven           12386         11098        158995
           New York            17041         17662        208756
Southeast  Atlanta             27522         29171        352161
           Memphis             21599         23090        274812
           Orlando             24143         23092        317027
West       Los Angeles         23222         22909        306468
           San Francisco       23311         23555        279830
           Seattle             24635         24121        309197

                        **** Return to Summary ****
```

ハイパーリンクによるレポートの関連付け (手順 3)

この例では、次のことが示されています。

- ソート区切り要素にドリルスルーハイパーリンクを配置する場合、そのソート区切りがこのドリルスルーに関与するソートフィールドの中で最終のソートフィールドレベルまたはそれ以降になるようにします。たとえば、2 番目のレポートでは、ドリルスルーハイパーリンクは、CATEGORY に関連付けられた (ソート区切りのある) 見出しではなく、PRODUCT に関連付けられた中間脚注に配置されています。

  コーディングにより、CATEGORY の見出しで PRODUCT の値を読み出すことは可能ですが (これを確認するには、見出しに <PRODUCT> フィールドを埋め込む)、その値は常に CATEGORY の最初の PRODUCT 値になります。各 CATEGORY に属する各 PRODUCT の値にドリルスルーハイパーリンクを配置するのが一般的な方法です。

  埋め込み項目にハイパーリンクを配置する必要はありません。ハイパーリンクは、テキスト項目に直接配置することができます。この中間脚注の項目は、CATEGORY および PRODUCT の同一値に関連付けられています。同様に、DATA 行の任意のフィールドにハイパーリンクを配置することができ、これにより関連付けられたリンクフィールド間の値が一致します。従来のドリルダウンハイパーリンクも、これと同様に機能します。

概要レポートの内容は次のとおりです。

- 各 DATA 行の PRODUCT フィールドにハイパーリンクを設定します。
- 次のレポートへリンクを設定するフィールドとして、CATEGORY と PRODUCT を指定

します。詳細レポートのターゲットフィールドの名前が概要レポートでの名前と同一であることから、CATEGORY=CATEGORY の代わりに CATEGORY という表記を使用します。

- アクションとして DOWN を指定します。これにより、ハイパーリンクをクリックすると次のレポートにジャンプして、2 つのリンクフィールドの値が一致する位置へ移動します。
- ハイパーリンクには、青色の下線テキストであるデフォルト設定を使用します。

概要レポートは次のとおりです。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS BY CATEGORY BY PRODUCT
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=DATA, COLUMN=PRODUCT, DRILLTHROUGH=DOWN(CATEGORY PRODUCT), $
ENDSTYLE
END
```

詳細レポートの内容は次のとおりです。

- PRODUCT リンクフィールドに関連付けられた中間見出しに、ハイパーリンクを設定します。CATEGORY は PRODUCT より上位の BY フィールドであるため、PRODUCT のそれぞれの中間見出しは CATEGORY のユニーク値にも関連付けられます。
- 中間見出しの 2 行目の最初の項目にハイパーリンクを設定します。これは、「Return to Summary」というテキストです。
- アクションとして FIRST を指定します。これにより、ハイパーリンクをクリックすると先頭レポート (概要レポート) にジャンプして、CATEGORY と PRODUCT の 2 つのリンクフィールドの値が一致する位置に移動します。
- COLOR 属性を使用して、ハイパーリンクを赤色の下線テキストで表示します。

詳細レポートは次のとおりです。

```
SET SQUEEZE=ON
TABLE FILE GGSALES
SUM UNITS BUDUNITS DOLLARS
BY CATEGORY NOPRINT BY PRODUCT NOPRINT
ON CATEGORY PAGE-BREAK
HEADING CENTER
"Category: <CATEGORY"
" "
ON PRODUCT SUBHEAD
```

```
"**** Product: <PRODUCT"
ON PRODUCT SUBFOOT
" "
"<25 **** Return to Summary ****"
ON PRODUCT PAGE-BREAK
BY REGION BY CITY
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=SUBFOOT, LINE=2, ITEM=1, DRILLTHROUGH=FIRST(CATEGORY PRODUCT),
COLOR=RED, $
ENDSTYLE
END
```

次の手順に限り、複合レイアウトレポートを作成する場合と、従来の複合レポートを作成する場合で手順が異なります。

複合レイアウトレポートの作成 (手順 4)

複合レイアウトレポートを作成する場合は、次の手順 4 を実行します。

次の手順に従って、複合レポートを作成します。

- プロシジャの先頭に、COMPOUND LAYOUT および SECTION 宣言を追加します。
- 2 つのレポートに、PAGELAYOUT および COMPONENT 宣言を追加します。COMPONENT 宣言に、DRILLMAP 属性を追加します。
- SET COMPONENT コマンドと 2 つのレポートを追加します。
- 次のように、COMPOUND END コマンドで、プロシジャを終了します。

```
SET HTMLARCHIVE=ON
COMPOUND LAYOUT PCHOLD FORMAT PDF
UNITS=IN, $
SECTION=section1, LAYOUT=ON, METADATA='0.5^0.5^0.5^0.5', MERGE=OFF,
   ORIENTATION=PORTRAIT, PAGESIZE=Letter,  $
PAGELAYOUT=1, NAME='Page layout 1', text='Page layout 1', TOC-LEVEL=1,
   BOTTOMMARGIN=0.5, TOPMARGIN=0.5, METADATA='BOTTOMMARGIN=0.5,
   TOPMARGIN=0.5,LEFTMARGIN=0,RIGHTMARGIN=0, $
COMPONENT='REPORT1', TEXT='REPORT1', TOC-LEVEL=2,
DRILLMAP=((L1 REPORT2)), POSITION=(0.750 1.083), DIMENSION=(7.000 3.167),
   METADATA='Z-INDEX:100; LEFT:0.75in; OVERFLOW: auto; WIDTH:7in;
   POSITION: absolute; TOP:1.083in; HEIGHT:3.167in', $
PAGELAYOUT=2, NAME='Page layout 2', text='Page layout 2', TOC-LEVEL=1,
```

```
   BOTTOMMARGIN=0.5, TOPMARGIN=0.5, METADATA='BOTTOMMARGIN=0.5,
   TOPMARGIN=0.5,LEFTMARGIN=0,RIGHTMARGIN=0, $
COMPONENT='REPORT2', TEXT='REPORT2', TOC-LEVEL=2,
DRILLMAP=((L1 REPORT1)), POSITION=(0.500 0.667), DIMENSION=(7.417 7.000),
   METADATA='Z-INDEX:100; LEFT:0.5in; OVERFLOW: auto; WIDTH:7.417in;
   POSITION: absolute; TOP:0.667in; HEIGHT:7in', $
END
```

```
-* Add Report1 code and SET COMPONENT command
SET COMPONENT='REPORT1'
TABLE FILE GGSALES
SUM UNITS DOLLARS BY CATEGORY BY PRODUCT
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=DATA, COLUMN=PRODUCT, DRILLTHROUGH=DOWN(CATEGORY PRODUCT), $
ENDSTYLE
END
```

```
-* Add report2 code and SET COMPONENT command
SET COMPONENT='REPORT2'
SET SQUEEZE=ON
TABLE FILE GGSALES
SUM UNITS BUDUNITS DOLLARS
BY CATEGORY NOPRINT BY PRODUCT NOPRINT
ON CATEGORY PAGE-BREAK
HEADING CENTER
"Category: <CATEGORY"
" "
ON PRODUCT SUBHEAD
"**** Product: <PRODUCT"
ON PRODUCT SUBFOOT
" "
"<25 **** Return to Summary ****"
ON PRODUCT PAGE-BREAK
BY REGION BY CITY
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=SUBFOOT, LINE=2, ITEM=1, DRILLTHROUGH=FIRST(CATEGORY PRODUCT),
COLOR=RED, $
ENDSTYLE
END
COMPOUND END
```

概要レポートと詳細レポートを結合した PDF 複合レポートの作成 (手順 4)

従来の複合レポートを作成する場合は、次の手順 4 を実行します。

次の手順では、複数のレポートを単一の PDF 複合レポートに統合します。次の処理を実行することができます。

- PCHOLD FORMAT PDF コマンドで OPEN と CLOSE オプションをコーディングする。
- レポートコンポーネントを配置する前に、SET COMPOUND コマンドで OPEN と CLOSE オプションをコーディングする。

ドリルスルーでは、改ページを間に入れずに複合レポートを表示するための NOBREAK オプションはサポートされていません。

次の例では、PCHOLD FORMAT PDF コマンドで OPEN と CLOSE オプションを使用します。

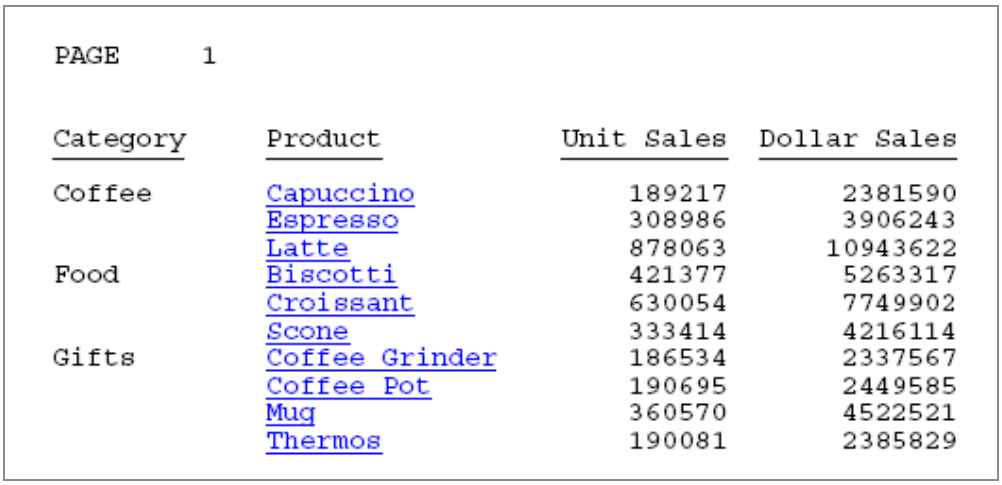
```
TABLE FILE GGSALES
SUM UNITS DOLLARS BY CATEGORY BY PRODUCT
ON TABLE PCHOLD FORMAT PDF OPEN
ON TABLE SET STYLE *
TYPE=DATA, COLUMN=PRODUCT, DRILLTHROUGH=DOWN(CATEGORY PRODUCT), $
ENDSTYLE
END
```

```
SET SQUEEZE=ON
TABLE FILE GGSALES
SUM UNITS BUDUNITS DOLLARS
BY CATEGORY NOPRINT BY PRODUCT NOPRINT
ON CATEGORY PAGE-BREAK
HEADING CENTER
"Category: <CATEGORY"
" "
ON PRODUCT SUBHEAD
"**** Product: <PRODUCT"
ON PRODUCT SUBFOOT
" "
"<25 **** Return to Summary ****"
ON PRODUCT PAGE-BREAK
BY REGION BY CITY
ON TABLE PCHOLD FORMAT PDF CLOSE
```

```
ON TABLE SET STYLE *
TYPE=SUBFOOT, LINE=2, ITEM=1, DRILLTHROUGH=FIRST(CATEGORY PRODUCT),
COLOR=RED, $
ENDSTYLE
END
```

ドリルスルーレポートの実行 (手順 5)

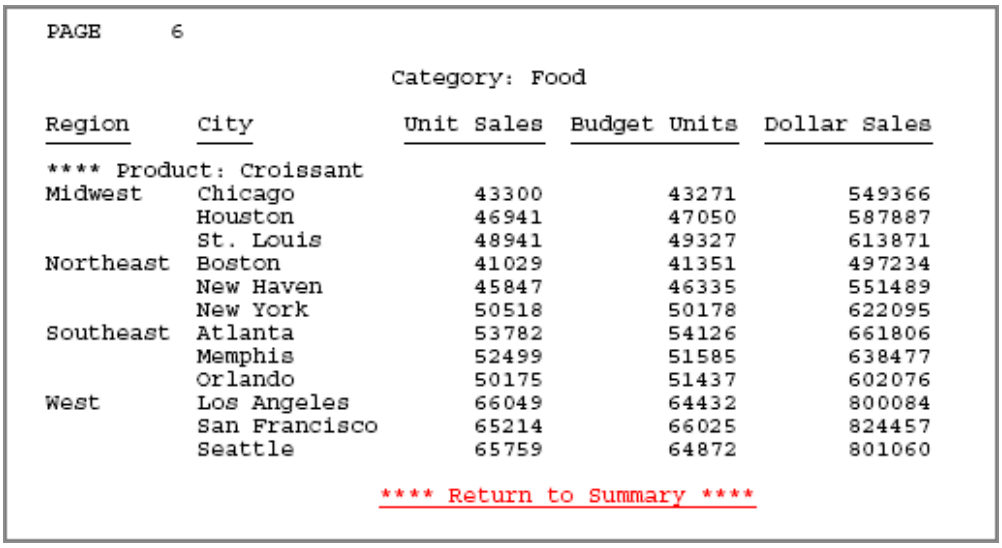
レポートを実行します。出力の最初のページには、概要レポートが表示されます。個々の製品へのハイパーリンクが、下線付きの青色で示されます。



PAGE 1

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

Food カテゴリの Croissant ハイパーリンクをクリックします。詳細情報にジャンプします。詳細レポートには、概要レポートに戻るためのハイパーリンクが下線付きの赤色で表示されます。



PAGE 6

Category: Food

Region	City	Unit Sales	Budget Units	Dollar Sales
**** Product: Croissant				
Midwest	Chicago	43300	43271	549366
	Houston	46941	47050	587887
	St. Louis	48941	49327	613871
Northeast	Boston	41029	41351	497234
	New Haven	45847	46335	551489
	New York	50518	50178	622095
Southeast	Atlanta	53782	54126	661806
	Memphis	52499	51585	638477
	Orlando	50175	51437	602076
West	Los Angeles	66049	64432	800084
	San Francisco	65214	66025	824457
	Seattle	65759	64872	801060

**** Return to Summary ****

「Return to Summary」ハイパーリンクをクリックして、最初のページ (概要レポート) に戻ります。

FIRST を使用したリンクに関するガイドライン

次のガイドラインが適用されます。

- FIRST を使用したすべてのリンクフィールドは、先頭レポートで DOWN を使用したすべてのリンクフィールドと対応していなければなりません。
- 最終レポートで FIRST ハイパーリンクをクリックすると、それに対応する先頭レポートの行に戻ります。選択したすべてのリンクは、先頭レポートの特定の行を正しく識別する必要があります。
- 先頭レポートの行にある DOWN ハイパーリンクは、先頭レポートの特定の行を識別し、その行と一致する行を詳細レポートで特定する必要があります。先頭レポートの DOWN に対応するすべてリンクと、最終レポートの FIRST リンクは、どちらも先頭レポートの特定の行を正しく識別しなければならないという点で共通しています。

ドリルスルーハイパーリンクの規則

ドリルスルーで複数のレポートをリンクさせる場合、レポート間のハイパーリンクを正しく機能させるために特定の規則に従う必要があります。主な要素は次のとおりです。

- **ソースレポート**　リンク元のレポートです。ソースレポートには、ターゲットレポートへのハイパーリンクを設定します。
- **ターゲットレポート**　リンク先のレポートです。
- **リンクフィールド**　ソースレポートとターゲットレポートの両方に存在するフィールドで、同一のデータタイプで相互に対応するフィールドの組み合わせです。リンクフィールドにより、ソースレポートのハイパーリンクからターゲットレポートへジャンプする際に、そのジャンプ先が特定されます。

「ソース」および「ターゲット」という用語は、リンクされた 2 つのレポートのいずれかを指しています。たとえば、ドリルスルーでリンクされたレポートが 3 つある場合、一般的に、2 番目のレポートは先頭レポートのターゲットレポートであると同時に、3 番目のレポートのソースレポートにもなります。

ドリルスルーとしてレポートを作成するには、ターゲットレポートに関連付けるリンクフィールドを特定し、ソースレポートでそのリンクフィールドを識別する必要があります。

- ソースレポートとターゲットレポートの間で一致する値が存在するフィールドをリンクフィールドとして選択します。たとえば、ソースレポートには機械の部品番号用のフィールドがあり、ターゲットレポートには従業員の社会保障番号用のフィールドがある場合、これらのフィールドには一致する値が存在しないため、ハイパーリンクを設定することはできません。
- 必要な数のリンクフィールドを指定して、ソースレポートのリンクフィールドに対応する場所をターゲットレポートで特定します。たとえば、ソースレポートが STATE と CITY でソートされた場合、リンクフィールドとして CITY だけを指定すると、他州 (STATE) に同じ名前の都市 (CITY) が存在する場合に混乱を生じます。
- ソースレポートとターゲットレポートのリンクフィールドには、同一の内部 (実) フォーマットが設定されていなければなりません。データタイプと内部データ長は一致させておく必要があります。表示桁数やカンマサプレスなどのフォーマットオプションに関しては、一致しなくてもよい場合があります。たとえば、A20 のフィールドは、他の A20 のフィールドとリンクさせる必要があります。逆に、I6C フィールドと I8C フィールドは、いずれも内部的に 4 バイトのフィールドのため、相互にリンクさせることができます。
- ソースレポートとターゲットレポートの両方において、リンクフィールドはソートフィールドまたは動詞オブジェクトでなければなりません。NOPRINT (非表示) フィー

ルドを含めることができます。この方法は、見出しテキストにフィールドが埋め込まれたレポートを作成する場合に役立ちます。

- ドリルスルーレポートの設計は、ドリルダウンレポートの設計と類似しています。ドリルスルーレポートでリンクフィールドを選択することは、ドリルダウンレポートでパラメータを選択することと類似しています。同様に、ドリルスルー用の構文は、ドリルダウン用の構文と類似しています。
- ドリルスルーレポートでは、リンクフィールドで相互に対応する値によりリンクが設定されるため、ハイパーリンクはリンクフィールドの特定の値に関連するレポート要素に表示されなければなりません。ただし、ハイパーリンクが、すべてのリンク自体に表示される必要はありません。
- すべての行タイプが、ドリルスルーハイパーリンクの使用に適しているとは限りません。たとえば、ドリルスルーリンクでもある BY フィールドで改ページが挿入される場合、各ページ見出しがその BY フィールドの値に関連付けられていることは明らかです。逆に、ページのオーバーフローにより改ページが挿入される場合は、見出しにドリルスルーリンクを配置しないようにします。同様に、中間見出し、中間脚注、中間合計、RECAP には、特定の BY フィールドの値のみが関連付けられます。

HTML レポートのナビゲーション

レポートの表示を制御するために、レポートに次のナビゲーション機能を含めることができます。

- **動的 TOC**　複数階層の目次 (TOC) をレポートに追加して、データの表示およびナビゲーションをしやすくすることができます。TOC はクリックできる値の展開可能リストとして、または一連のドロップダウンフォームコントロールとして表示することができます。TOC の値をクリックすると、データの表示を切り替え、レポートの値をすばやく見つけることができます。
- **Web ビューア**　HTML レポートを複数の Web ページに分割して、ブラウザへの情報の配信を高速化することができます。この機能は Web ビューア内に実装されており、出力の 1 ページ目の表示後に、前後のページや特定のページへ移動することができます。
- **ページ間のハイパーリンク**　複数のページを含むレポートに自動ハイパーリンクを定義し、連続するレポートページを相互にリンクして、前後のページへの移動を容易にすることができます。

目次からソートグループへのナビゲーション

動的 HTML を基準とするTOC (目次) を追加して、大規模な HTML レポートの内部ナビゲーション機能を拡張することができます。この機能を活用するには、少なくとも 1 つの BY フィールドを、レポートに含める必要があります。複数のソートフィールドが挿入されている場合、リクエストでソートフィールドが指定されている順序によって階層が決まります。

TOC によりデータグループの表示機能を強化することもできます。レポートの 1 セクション (または 1 ページ) ごとに、あるいはセクションのすべてを同時に表示することができます。これは改ページにより制御することができます。

TOC には、1 つ目の (最上位) 縦方向ソートフィールド値がすべて表示されます。また、指定した場合、任意の下位 BY フィールドを表示させることもできます。これらの値は、展開可能な一連のリンクまたは一連のリストコントロールとして表示されます。リクエストで特に

指定しない限り、最上位のソートフィールドが変更されたところで、新しいページが開始します。

最下位ソートフィールドのデータ表示は、それよりも上位のソートフィールド値の選択状態により制御されます。たとえば、まず国により、次に車種によりソートしたレポートで、TOC から国として Italy を選択すると、車の TOC ではイタリア製の車種のリストのみが表示されます。他国で生産された車は表示されません。

TOC を使用して、次のことが可能です。

- 関連リンクをクリックして、レポートの任意のセクションを表示する。
- [レポート全体表示] をクリックし、1 つのセクションとレポート内容全体で表示を切り替える。
- [目次を削除] をクリックし、TOC を削除する。これはブラウザからレポートを印刷するときに役立ちます。表示を元に戻してナビゲーションを続行するには、レポートの任意の場所をダブルクリックします。

TOC 自体は 1 つのオブジェクトであり、初期状態では、レポートの左上隅にアイコンとして表示されるか、ページ見出し、脚注、レポート見出し、または脚注に 1 つまたは複数のドロップダウンリストとして表示されます。

- **見出しオプション**　TOC を見出しまたは脚注に追加するには、スタイルシートを使用することができます。
- **レポートオプション**　HTML TOC オブジェクトをレポートの左上隅に追加するには、SET コマンドまたは PCHOLD コマンドを使用することができます。

HTML レポートでは、テーブルのサイズはブラウザによって制御されます。列幅のサイズは最大長のデータ値が収まるよう調整され、末尾のブランクは自動的に削除されます。標準の HTML レポートでは、データは単一テーブルに表示されるため、列幅はすべてのデータ行で固定されます。HTML BYTOC (レポート) および TOC (見出し) 機能を使用すると、各ソートキー値のデータがそれぞれ個別のテーブルまたはセクションに配置されます。これらのテーブルまたはセクションにアクセスするには、HTML コントロールを使用します。HTML BYTOC および TOC レポートでは、各テーブルの列幅は、レポート全体ではなく、各ソートキー値のデータに基づいて決定されます。[レポート全体表示] オプションを選択すると、各テーブルがそれぞれ異なるサイズで表示されていることが分かります。すべてのテーブルで列幅を固定するには、SQUEEZE = OFF を設定します。

HTML 目次で HTMLARCHIVE を使用する際の注意

WebFOCUS インタラクティブレポート機能が正常に動作するには、WebFOCUS Client に接続して、この機能に必要なコンポーネントにアクセスする必要があります。

WebFOCUS Client へのアクセスが利用できない場合は、HTMLARCHVE を使用して、ユーザ定義イメージが含まれた自己完結型 HTML ページを作成することができます。

ユーザ定義イメージが含まれた HTML ページを生成し、インタラクティブ動作を可能にするには、次のコマンドのいずれかを使用します。

```
SET HTMLEMBEDIMG=ON
SET HTMLARCHIVE=ON
```

次の構文を使用して、これらのファイルへのアクセスが可能なホストマシンを指定した BASEURL を定義します。

```
SET BASEURL=http://{hostname:portnumber}
```

SET BASEURL についての詳細は、ベース URL の指定を参照してください。

レポートへの HTML 目次ツリーコントロールの追加

HTML TOC オブジェクトをレポートへ追加するために、3 つの異なるタイプの構文を使用することができます。

SET コマンドによるレポートへの TOC ツリーコントロールの追加

SET コマンドを使用した構文は次のとおりです。

リクエストは、次の構文で開始します。

```
SET COMPOUND = 'BYTOC [n]'
```

リクエスト内では、次の構文を使用します。

```
ON TABLE SET COMPOUND 'BYTOC [n]'
```

説明

n

TOC に含める BY フィールドの数を表します。1 はリクエストで 1 つ目 (最上位) のソートフィールドから開始することを示します。ソートフィールドの階層はリクエストで指定した順序により決定されます。

デフォルト値は 1 で、最上位のソートフィールドおよびその値のみが TOC に表示されます。

デフォルト設定では、リクエストで指定しない限り、セクション区切りは 1 つ目 (最上位) のソートフィールドの後に配置されます。

注意：SET コマンドで BYTOC にフィールド数 (n) を指定する場合は、一重引用符 (') を使用する必要があります。

PCHOLD コマンドによるレポートへの TOC ツリーコントロールの追加

PCHOLD コマンドを使用した構文は次のとおりです。

```
ON TABLE PCHOLD FORMAT HTML BYTOC [n]
```

説明

n

TOC に含める BY フィールドの数を表します。1 はリクエストで 1 つ目 (最上位) のソートフィールドから開始することを示します。ソートフィールドの階層はリクエストで指定した順序により決定されます。

デフォルト値は 1 で、最上位のソートフィールドおよびその値のみが TOC に表示されます。

デフォルト設定では、リクエストで指定しない限り、セクション区切りは 1 つ目 (最上位) のソートフィールドの後に配置されます。

注意：PCHOLD コマンドで BYTOC にフィールド数 (n) を指定する場合は、一重引用符 (') を使用する必要があります。

スタイルシート宣言によるレポートへの TOC ツリーコントロールの追加

スタイルシートで TOC ツリーコントロールを追加するには、次の構文を使用します。

```
TYPE=REPORT, TOC='n',$
```

または

```
TYPE=REPORT, TOC='sortfieldname',$
```

説明

n

TOC に含める BY フィールドの数を表します。1 はリクエストで 1 つ目 (最上位) のソートフィールドから開始することを示します。TOC ツリーに表示されるソートフィールドの階層は、リクエストで指定した順序により決定されます。

sortfieldname

縦ソート (BY) フィールドの名前を指定します。

注意：スタイルシートで TOC を指定する場合は、一重引用符 (') を使用する必要があります。

レポートのオブジェクトとして HTML TOC を追加 (レポートオプション)

下図のように、リクエストの前に SET コマンドを追加することで、HTML TOC をレポートの左上隅にアイコンとして追加するこができます。TOC には、最初 (最上位) の BY ソートフィールドである PLANT の値が一覧表示されます。

```
SET COMPOUND='BYTOC 2'
```

```
TABLE FILE CENTORD
HEADING
"SALES REPORT"
SUM LINEPRICE BY PLANT BY PRODCAT
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

リクエストで SET コマンドを使用することにより、HTML TOC をアイコンとしてレポートの左上隅に追加することもできます。

```
TABLE FILE CENTORD
HEADING
"SALES REPORT"
SUM LINEPRICE BY PLANT BY PRODCAT
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET COMPOUND 'BYTOC 2'
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

次の例では、PCHOLD コマンドを使用してリクエストを実行します。

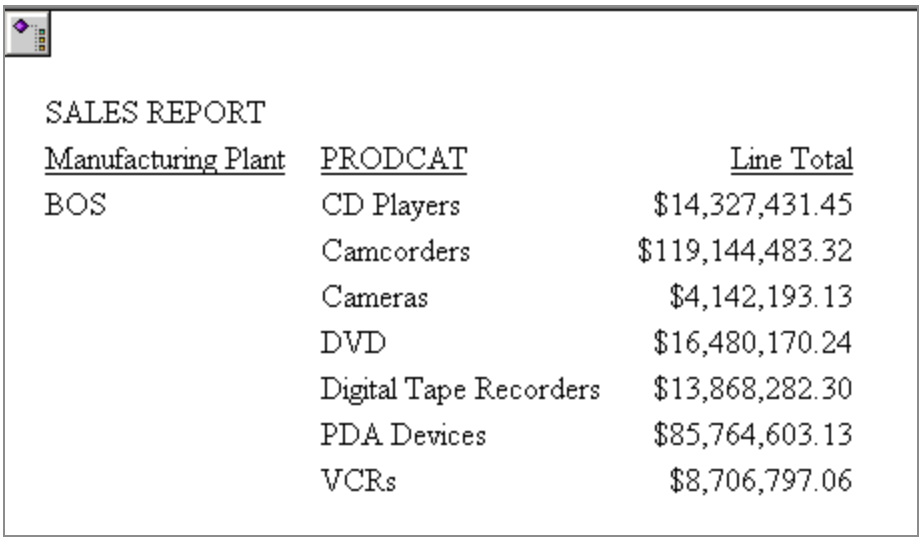
```
TABLE FILE CENTORD
HEADING
"SALES REPORT"
SUM LINEPRICE BY PLANT BY PRODCAT
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML BYTOC 2
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

次のリクエストでは、レポートのスタイルシートで TOC ツリーコントロールを有効にします。

```
TABLE FILE CENTORD
HEADING
"SALES REPORT"
SUM LINEPRICE
BY PLANT BY PRODCAT
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=REPORT, TOC='PRODCAT', $
ENDSTYLE
END
```

注意：スタイルシートで TOC を指定する場合は、一重引用符 (') を使用する必要があります。

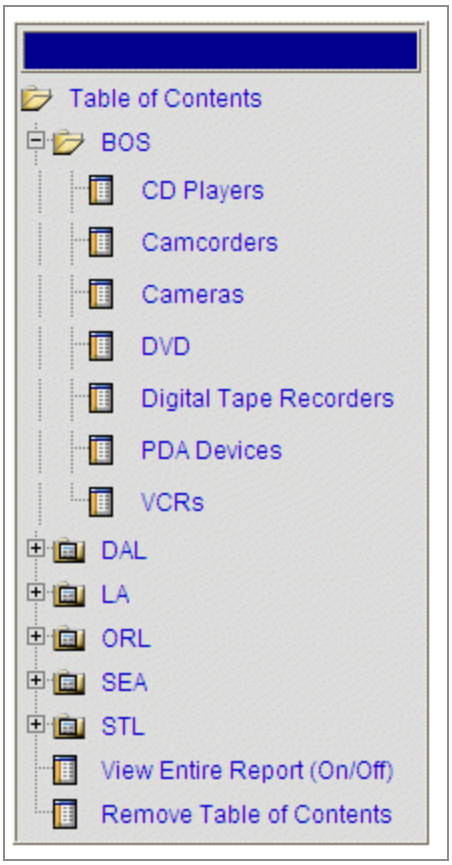
レポートを実行します。TOC アイコンが左上に表示されます。



SALES REPORT

Manufacturing Plant	PRODCAT	Line Total
BOS	CD Players	$14,327,431.45
	Camcorders	$119,144,483.32
	Cameras	$4,142,193.13
	DVD	$16,480,170.24
	Digital Tape Recorders	$13,868,282.30
	PDA Devices	$85,764,603.13
	VCRs	$8,706,797.06

TOC アイコンをダブルクリックすると、TOC ツリーコントロールが開きます。TOC ツリーコントロールでは、レポートのソートフィールドの値が、リクエストで指定された順序で表示されます。



注意：TOC 上部の青い部分をクリックし、レポートの別の領域にドラッグするか、任意の場所をダブルクリックすることにより、TOC を移動することができます。

使用可能なフィールドすべて (レポート全体) を表示するには、[レポート全体表示 (オン/オフ)] オプションをクリックします。

ヒント：.css ファイルを編集することにより、TOC オブジェクトのイメージをカスタマイズすることもできます。ファイルを編集する前に、ファイルのバックアップを作成しておくことをお勧めします。

- セルフサービスまたはブラウザから BI Portal 環境で作業している場合は、¥ibi¥WebFOCUS##¥ibi_html¥javaassist¥intl¥*xx* ディレクトリを開きます。*##* は WebFOCUS のバージョン番号で、*xx* は言語の省略形です。英語 (EN) の .css ファイル名は「toc.css」です。他の全言語の .css ファイル名は「*xx*toc.css」 (*xx* は言語の省略形) です。

注意：[目次を削除] をクリックした後で、再び TOC を表示するには、レポートの任意の位置をダブルクリックします。

表示用ソートフィールドのグループ化

TOC レポートのデータは、ソートフィールドに基づいて、セクションにグループ化されます。TOC レポートは、見やすくするために、セクションごとに表示されます。各セクションには、そのセクションに対応するソートフィールドの値が、すべて含まれます。改ページで、各セクションをカスタマイズすることができます。デフォルト設定では、改ページは 1 つ目 (最上位) のソートフィールドに追加されます。改ページを挿入して追加のセクションを作成し、下位ソートフィールドに基づいてデータをグループ化することができます。

TOC を見出しに追加するときは、下位の各ソートフィールドに改ページを追加します。これにより、ソートフィールドが適切にグループ化され、表示されます。

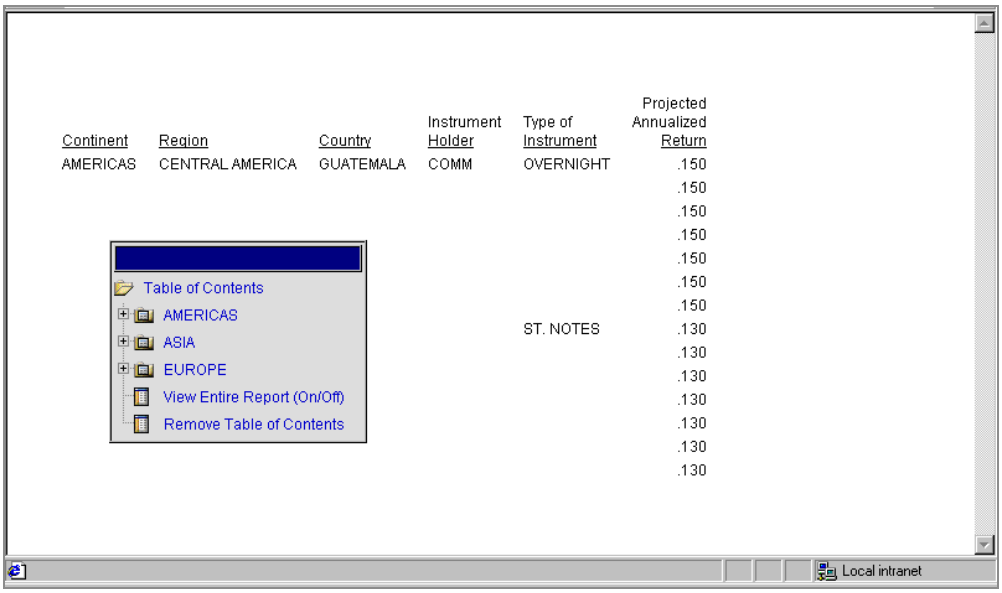
改ページによるレポートセクションのカスタマイズ

```
TABLE FILE SHORT
PRINT PROJECTED_RETURN
BY CONTINENT
BY REGION
BY COUNTRY
BY HOLDER
BY TYPE
```

```
ON HOLDER PAGE-BREAK
ON TABLE SET PAGE-NUM OFF
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET COMPOUND 'BYTOC 5'
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
TYPE=REPORT,
GRID=OFF,
FONT='ARIAL',
SIZE=9,
COLOR='BLACK',
BACKCOLOR='NONE',
STYLE=NORMAL,
$
ENDSTYLE
END
```

レポートのセクションは 1 つずつ表示されます。

レポートは、HOLDER の値に基づいてセクションに分割されます。1 つのセクションには、HOLDER 各値の詳細が表示されます。



複数階層 TOC でのナビゲーション動作

TOC で値を選択すると、その値はブラウザウィンドウで点滅し (灰色で強調表示され)、ユーザの注意を促します。点滅の表示位置および画面表示の切り替えは、次の要素により制御さ

れます。

- TOC で最上位ソートグループを変更すると (レポート上部の階層または見出し、脚注の 1 つ目のドロップダウンリストで変更)、選択値はブラウザウィンドウで 3 回点滅します。
- 使用中の最上位ソートグループで下位のソート値を変更すると、選択した値はウィンドウで 3 回点滅します。これは、まだ同一の主ソートグループ内に位置し、そのため、同じページ内に位置していることを意味します。ウィンドウの最上部の選択値から、関連詳細へすばやくスクロールすることができます。

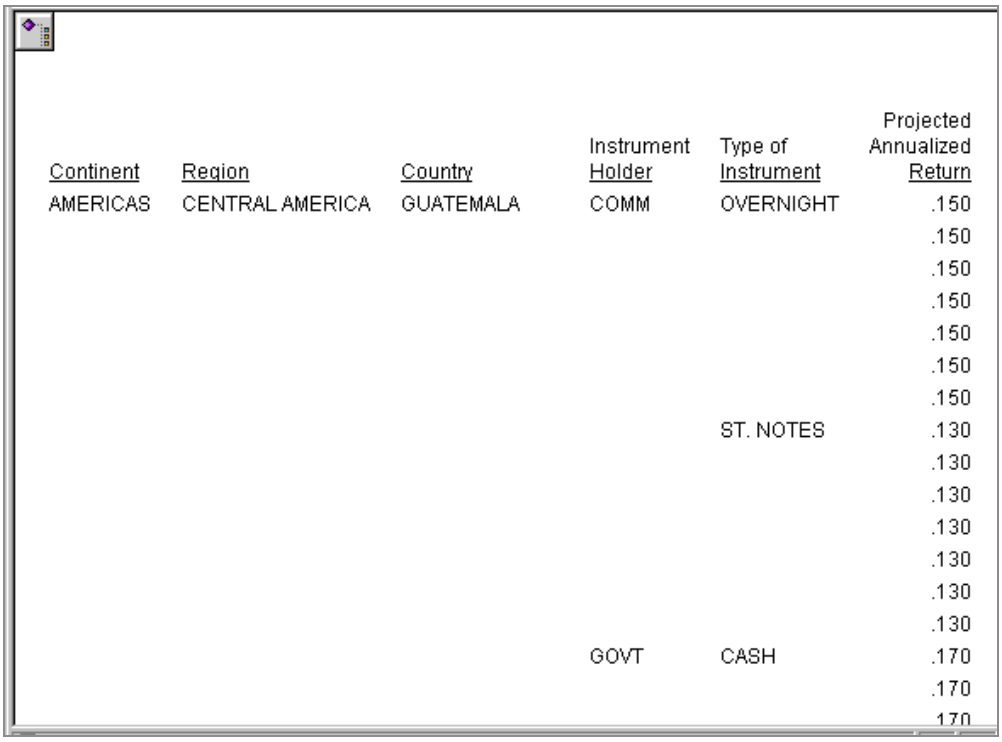
  下位の選択値がすでに画面に表示可能で、レポートの残りの部分が画面に収まる場合、その値は点滅し、レポートはスクロールしません。

複数階層 TOC からのソートデータのナビゲーション

次のリクエストは、SET コマンドを含めることにより、動的 HTML TOC をアイコンとしてレポートの左上隅に追加します。TOC には、第 1 (最上位) ソートフィールドから始まる 5 つのレベルで構成される階層が表示されます。ソートフィールドは CONTINENT、REGION、COUNTRY、HOLDER、TYPE です。

```
TABLE FILE SHORT
PRINT PROJECTED_RETURN
BY CONTINENT
BY REGION
BY COUNTRY
BY HOLDER
BY TYPE
ON TABLE SET PAGE-NUM OFF
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET COMPOUND 'BYTOC 5'
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
TYPE=REPORT,
GRID=OFF,
FONT='ARIAL',
SIZE=9,
STYLE=NORMAL,
$
ENDSTYLE
END
```

出力結果は、左上隅の TOC オブジェクトとともに表示されます。



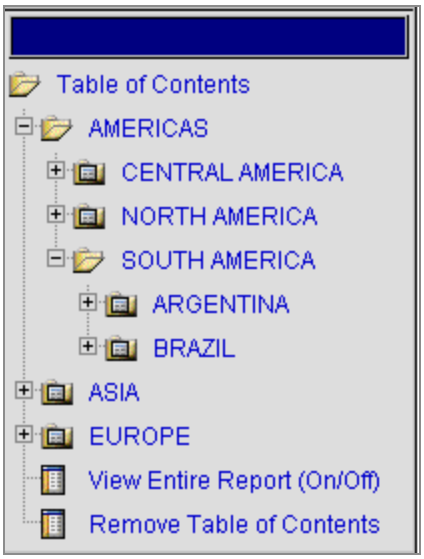
Continent	Region	Country	Instrument Holder	Type of Instrument	Projected Annualized Return
AMERICAS	CENTRAL AMERICA	GUATEMALA	COMM	OVERNIGHT	.150
					.150
					.150
					.150
					.150
					.150
					.150
				ST. NOTES	.130
					.130
					.130
					.130
					.130
					.130
					.130
			GOVT	CASH	.170
					.170
					.170

オブジェクトをダブルクリックして目次 (TOC) を展開します。

[レポート全体表示] を選択します。下方向へスクロールすると、レポートにすべての CONTINENT のデータが含まれていることが分かります。

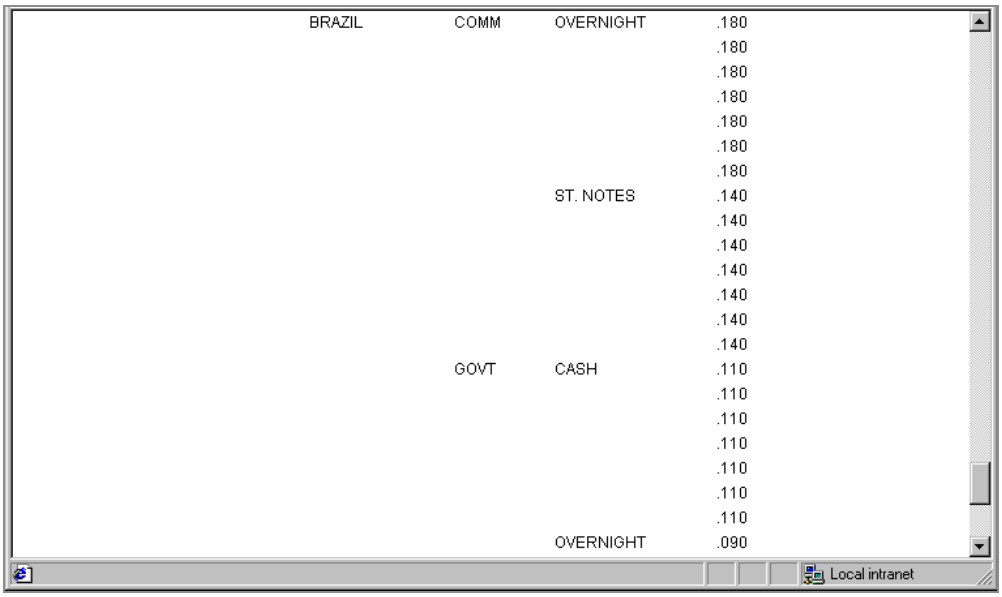
レポートウィンドウの最上部へスクロールして戻り、TOC を再び開きます。次に、「Americas」を選択します。選択した項目が画面内で点滅し、強調表示されます。レポート表示は変更されたように見えませんが、この時点で下方向へスクロールすると、レポートに「Americas」の値のみが含まれていることが分かります。

再び上方向へスクロールし、レポートの任意の場所をダブルクリックして TOC を開きます。次に「Americas」の横のプラス記号 (+) をクリックし、さらに「South America」の横のプラス記号 (+) をクリックします。



TOC リストにフィールド値 (Argentina および Brazil) が 含まれています。これらは COUNTRY フィールドの値です。TOC の値のフィールド名を表示するには、その値の上にマウスポインタを移動します。

「Brazil」を選択します。次に示すように、選択項目が点滅し、ウィンドウの最上部へ移動します。



下方向へスクロールして Brazil のデータを参照します。

TOC の任意のソートレベルで値を選択し、引き続き詳細を表示します。

- プラス記号 (+) をクリックするとフィールドが展開され、TOC にその値が表示されま

す。

- TOC で実際の値 (ハイパーリンク) をクリックすると、一瞬その値が強調表示され、必要に応じて、その値を表示するためにレポートの表示が調節されます。

  [目次] をクリックすると TOC が最小化されアイコンになりますが、引き続きスクロールバック、展開、選択を実行することができます。

見出しへの TOC ドロップダウンリストコントロールの追加

スタイルシートに次の属性を追加します。

```
TYPE=heading, [subtype,] TOC=sort_column, $
```

説明

heading

TOC を含める見出しまたは脚注のタイプです。

有効な値には、次のものがあります。

`TABHEADING`	レポート見出し
`TABFOOTING`	レポート脚注
`HEADING`	ページ見出し
`FOOTING`	ページ脚注

subtype

要求した各ドロップダウンリストを表示する見出し、または脚注の位置を指定する属性です。コンポーネントを識別する詳細度に応じて、これらのオプションを単一または組み合

わせて使用することができます。有効な値には、次のものがあります。

LINE_# - 見出しまたは脚注の行内の位置により行を指定します。

見出しまたは脚注に複数の行があり、LINE_# を指定しないスタイルシートを適用すると、その構文はすべての行に適用されます。ブランク行は、LINE の値を解析するときに計算されます。

LINE=n - 複数行の見出しまたは脚注で必要です。それ以外の場合は、省略することができます。

OBJECT - 見出しまたは脚注の TOC オブジェクトを、テキストまたはフィールド値として指定します。有効な値は TEXT または FIELD です。

フィールドまたはテキストを、TOC ドロップダウンリストのプレースホルダとして使用することができます。ただし、フィールドの方が適しています。TOC 機能が有効でない場合も、フィールド名はレポートに表示されます。

TEXT として、テキストまたはダイアログマネージャ (&) 変数のテキストを指定することができます。

同一の見出しまたは脚注内で、テキストおよび埋め込みフィールドの両方をスタイル設定しない場合は、OBJECT=TEXT を指定する必要はありません。

関連情報については、ITEM_# を参照してください。

ITEM_# - 行内の位置により、項目を指定します。

OBJECT の ITEM_# を決定するためのガイドラインは次のとおりです。

- OBJECT=TEXT を使用するときは、テキストのみが左から右に数えられます。
- OBJECT=FIELD を使用するときは、フィールドのみが左から右に数えられます。

スタイルシートで ITEM を指定する場合は、数は最初の行のはじめからではなく、見出しまたは脚注内の同じ行の開始から数えられます。

sort_column

TOC として含める BY フィールド を指定します。フィールドの識別には、次の規則が適用されます。

TOC=fieldname - フィールド名で BY フィールドを指定します。

TOC=Bn - リクエスト内での順序に基づいて BY フィールドを指定します。たとえば、B2 は 2 番目の BY フィールドを表します。この場合、個数には NOPRINT BY フィールドも含まれます。

TOC=n - TOC=Bn と同等です。

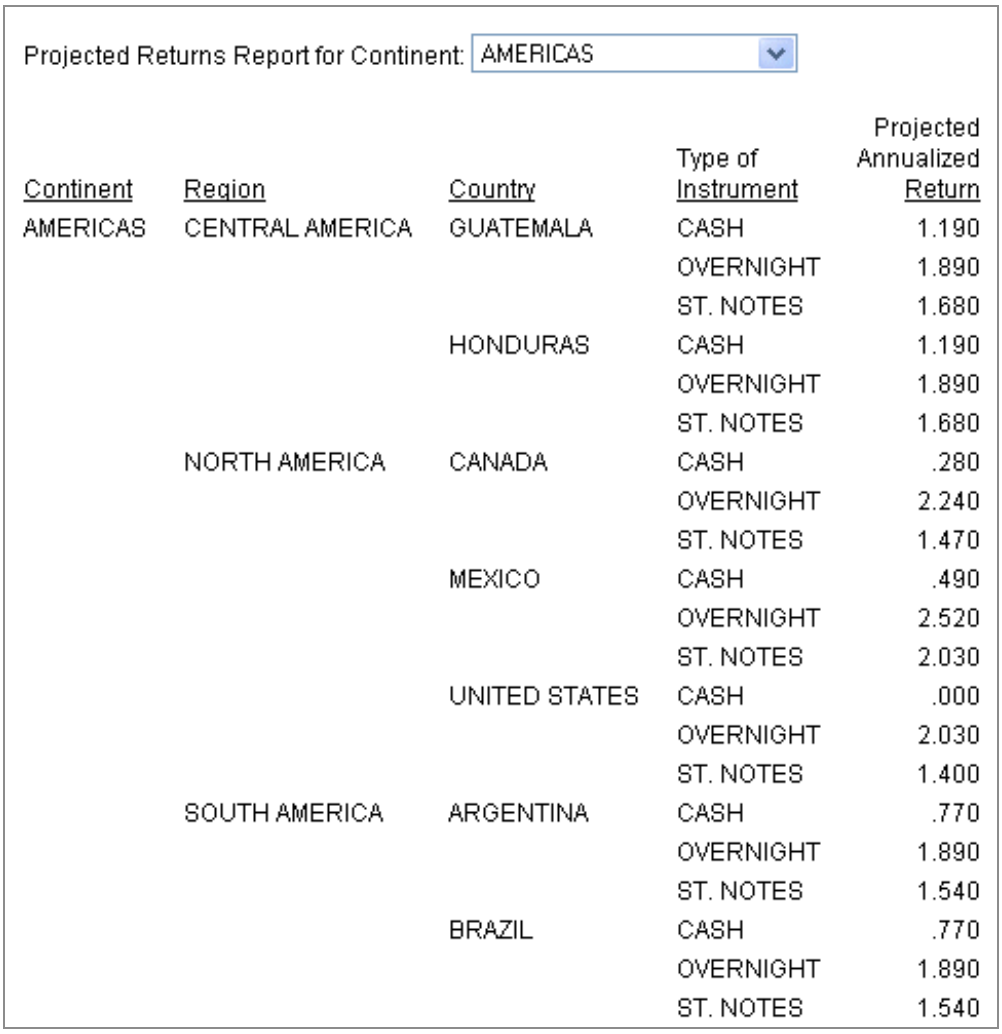
注意：見出しの TOC オブジェクト (ドロップダウンリスト) は、レポートの階層に依存し、これらは相互に依存するため、BY フィールドの階層を保持する必要があります。

ページ見出しへの HTML TOC ドロップダウンリストの追加

次のリクエストは、必要なスタイルシート属性を使用して、HTML レポートに TOC を追加します。TOC のドロップダウンリストには、CONTINENT フィールドの値がすべて表示されます。これは、スタイルシートの「OBJECT=FIELD, ITEM=1, TOC=CONTINENT」というコードで識別されます。

```
TABLE FILE SHORT
HEADING
"Projected Returns Report for Region:  <REGION in Continent:  <CONTINENT "
" "
SUM PROJECTED_RETURN
BY CONTINENT
BY REGION
BY COUNTRY
BY TYPE
ON TABLE SET PAGE-NUM OFF
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
TYPE=REPORT,
GRID=OFF,
FONT='ARIAL',
SIZE=9,
STYLE=NORMAL,
$
TYPE=HEADING, LINE=1, OBJECT=FIELD, ITEM=1, TOC=CONTINENT, $
ENDSTYLE
END
```

レポートを実行します。見出しの CONTINENT フィールドの位置に、TOC がドロップダウンリストとして表示されます。



Projected Returns Report for Continent: AMERICAS

Continent	Region	Country	Type of Instrument	Projected Annualized Return
AMERICAS	CENTRAL AMERICA	GUATEMALA	CASH	1.190
			OVERNIGHT	1.890
			ST. NOTES	1.680
		HONDURAS	CASH	1.190
			OVERNIGHT	1.890
			ST. NOTES	1.680
	NORTH AMERICA	CANADA	CASH	.280
			OVERNIGHT	2.240
			ST. NOTES	1.470
		MEXICO	CASH	.490
			OVERNIGHT	2.520
			ST. NOTES	2.030
		UNITED STATES	CASH	.000
			OVERNIGHT	2.030
			ST. NOTES	1.400
	SOUTH AMERICA	ARGENTINA	CASH	.770
			OVERNIGHT	1.890
			ST. NOTES	1.540
		BRAZIL	CASH	.770
			OVERNIGHT	1.890
			ST. NOTES	1.540

TOC をクリックすると、ソート値 (AMERICAS、ASIA、EUROPE) がリストに表示されます。

各 CONTINENT をクリックして関連情報を表示します。選択した値がウィンドウで灰色に点滅して強調表示されます。

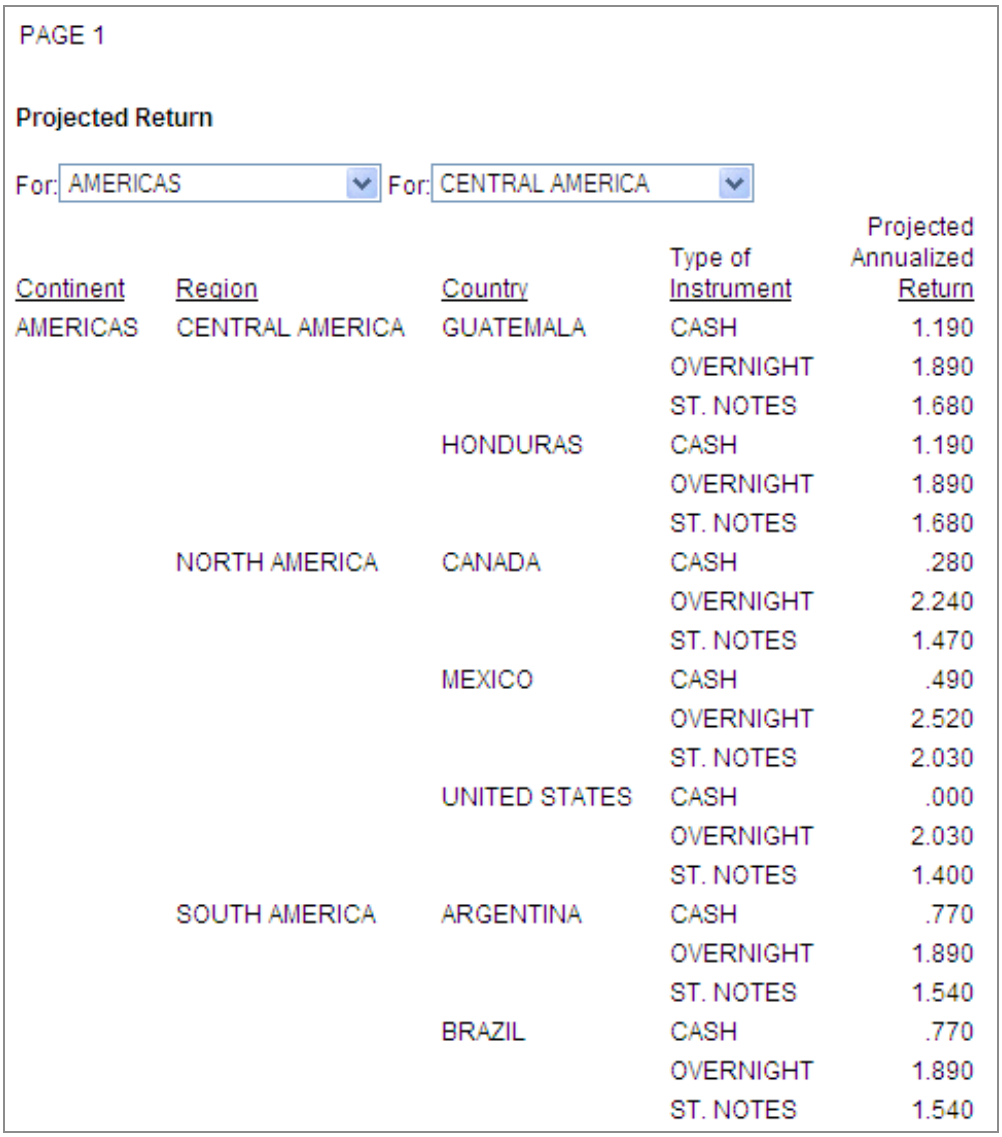
使用可能なすべてのフィールド (レポート全体) を表示するには、[レポート全体表示] オプションをクリックします。TOC を削除するには、[目次を削除] オプションをクリックします。TOC を復元するには、レポートの任意の場所をダブルクリックするか、ブラウザの [更新] ボタンをクリックします。

ページ見出しでの複数階層 HTML TOC のナビゲーション

次のリクエストでは、スタイルシートを使用して、このリクエストで指定された 2 つの BY フィールド (CONTINENT、REGION) の HTML TOC ドロップダウンリストを、ページ見出しの 3 行目に追加します。各フィールドがその TOC のプレースホルダになります。TOC 機能が有効でない場合、そのフィールドがレポートに表示されます。

```
TABLE FILE SHORT
"Projected Return"
" "
"For:<CONTINENT For:<REGION "
SUM PROJECTED_RETURN
BY CONTINENT BY REGION BY COUNTRY BY TYPE
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
ON CONTINENT PAGE-BREAK
ON TABLE SET STYLE *
TYPE=REPORT,
GRID=OFF,
FONT='ARIAL',
SIZE=9,
STYLE=NORMAL,
$
TYPE=HEADING, LINE=1, STYLE=BOLD, $
TYPE=HEADING, LINE=3, OBJECT=FIELD, ITEM=1, TOC=B1,$
TYPE=HEADING, LINE=3, OBJECT=FIELD, ITEM=2, TOC=REGION,$
ENDSTYLE
END
```

出力結果は次のとおりです。



PAGE 1

Projected Return

For: AMERICAS For: CENTRAL AMERICA

Continent	Region	Country	Type of Instrument	Projected Annualized Return
AMERICAS	CENTRAL AMERICA	GUATEMALA	CASH	1.190
			OVERNIGHT	1.890
			ST. NOTES	1.680
		HONDURAS	CASH	1.190
			OVERNIGHT	1.890
			ST. NOTES	1.680
	NORTH AMERICA	CANADA	CASH	.280
			OVERNIGHT	2.240
			ST. NOTES	1.470
		MEXICO	CASH	.490
			OVERNIGHT	2.520
			ST. NOTES	2.030
		UNITED STATES	CASH	.000
			OVERNIGHT	2.030
			ST. NOTES	1.400
	SOUTH AMERICA	ARGENTINA	CASH	.770
			OVERNIGHT	1.890
			ST. NOTES	1.540
		BRAZIL	CASH	.770
			OVERNIGHT	1.890
			ST. NOTES	1.540

2 つ目の TOC ドロップダウンリストから、[North America] を選択します。このドロップダウンリストには、上位 (1 つ目) のドロップダウンリストから選択された値に関連する値のみが表示されます。これらの値はすべて、同一の上位ソートグループ値に属するため、同一のセクション区切り内に表示されます。選択した値の North America が点滅します。そこから下方向にスクロールすると、下図のような関連データが表示されます。

		ST. NOTES	1.680
NORTH AMERICA	CANADA	CASH	.280
		OVERNIGHT	2.240
		ST. NOTES	1.470
	MEXICO	CASH	.490
		OVERNIGHT	2.520
		ST. NOTES	2.030
	UNITED STATES	CASH	.000
		OVERNIGHT	2.030
		ST. NOTES	1.400
SOUTH AMERICA	ARGENTINA	CASH	.770

次に上方向にスクロールし、1 つ目の TOC リストから [ASIA] を選択します。これを選択すると最上位のソートグループが変更され、その下のリストのすべてに影響します。ASIA が点滅してウィンドウの上部へ移動します。ここには Far East 地域の 1 つ目の国 (Hong Kong) の情報が表示されます。これは TOC の最上位から 2 つ目のソートグループであるため、ページ番号は 2 になります。

PAGE 2

Projected Return

For: ASIA For: FAR EAST

Continent	Region	Country	Type of Instrument	Projected Annualized Return
ASIA	FAR EAST	HONG KONG	CASH	.770
			OVERNIGHT	2.170
			ST. NOTES	1.610
		JAPAN	CASH	.770
			OVERNIGHT	2.170
			ST. NOTES	1.610
	MIDDLE EAST	ISRAEL	CASH	.700
			OVERNIGHT	1.680
			ST. NOTES	1.400
		SAUDI ARABIA	CASH	.770
			OVERNIGHT	1.890
			ST. NOTES	1.540

ドロップダウンリストから別の値を選択して、表示するデータを切り替えます。

レポート間のナビゲーション

WebFOCUS では、単一レポート内のナビゲーション方法およびツールに加え、複数のレポート間をナビゲートするためのいくつかの機能が提供されています。これらの機能により、ユーザはレポート表示からのナビゲーションを開始します。次の処理を実行することができます。

- 別のレポート、URL、JavaScript 関数へのドリルダウン。詳細は、レポートの別リソースへのリンクを参照してください。
- フレームの定義とレポートへの追加。詳細は、ターゲットフレームの指定を参照してください。

HTML 目次の制限事項

TOC 機能の使用上の制限事項は次のとおりです。

- HTML 出力に適用されます。
- Internet Explorer でサポートされます。その他のブラウザでこの機能を使用すると、予期しない動作が発生する場合があります。
- ダイアログマネージャコマンドの -HTMLFORM はサポートされません。
- アコーディオンレポートではサポートされません。
- BY フィールドを TOC として指定する場合は、同時にその親 BY フィールドを TOC 指定する必要があります。これは、TOC コントロールはそれぞれ独立しており、適切に動作するには各 TOC の親コントロールが物理的に存在する必要があるためです。たとえば、リクエストに BY COUNTRY BY CAR BY MODEL が含まれている場合、レポートに CAR の TOC コントロールを含めるには、同時に必ず COUNTRY の TOC コントロールを指定します。
- TOC が有効なレポートのサイズは、WebFOCUS Client のメモリサイズにより制限されます。
- リクエストに BYTOC 目次とスタイルシートの見出し目次の両方が含まれている場合、出力結果には見出し目次が表示されます。

ReportCaster がインストールされている場合は、次のコマンドをリクエストに含めることで、HTML TOC が追加されたレポートを配信することができます。

- SET BASEURL コマンドを使用して、WebFOCUS Client がインストールされている

Application Server に接続するための URL を設定します。詳細は、ベース URL の指定を参照してください。

アコーディオンレポート機能によるソートデータの表示制御

アコーディオンレポート機能を使用すると、HTML レポートページに表示するソートデータの数量を制御することができます。リクエストに複数の BY フィールドを指定し、BY ソートフィールドの値ごとに展開して表示可能なレポートを生成することができます。

次の 2 種類のアコーディオンレポートを作成することができます。

- **同一列内でのアコーディオンレポート**　同一列内でのアコーディオンレポートでは、ソートフィールド値およびその値に対応する各メジャーの集計値がロールアップされて表示されます。その結果、最上位のソートフィールド値および総合計がレポートの上部に表示されます。ツリーコントロールを使用して各ディメンションを開き、それぞれのディメンションに関連する集計値を表示することができます。ソートフィールド値の横にあるプラス記号 (+) をクリックすると、新しい行が開いて、次の下位レベルのソートフィールド値および中間合計が表示されます。最下位のソートフィールドを開くと、集計に使用されたデータ値が表示されます。このタイプのアコーディオンレポートを生成するには、SET EXPANDBYROW コマンドを使用します。

  同一列内アコーディオンレポートの新しいインターフェースでは、ポータルページで幅広い大規模レポートを操作する際のナビゲーションが容易になり、データがコンテナ内に収まるよう自動的にサイズ変更され、列幅はフィールド値の最大長または列タイトル幅のどちらか大きい方に自動調整されます。強化された同一列内アコーディオンレポート機能を有効にするには、プロシジャ内で SET EXPANDBYROWTREE=ON コマンドを使用します。詳細は、アコーディオンレポート機能によるソートデータの表示制御を参照してください。

  注意：EXPANDBYROW は、機能的に安定しています。EXPANDBYROWTREE の機能は、将来的に拡張される予定です。

- **列単位のアコーディオンレポート**　列単位のアコーディオンレポートでは、ソートフィールド値およびデータ値がロールアップされて表示されます。ただし、レポートの行全体は自動的には表示されません。最上位ソートフィールド列の各データ値の左側にプラス記号 (+) が表示されます。下位のソートフィールドに対応するデータ値の左側にプラス記号 (+) が表示され、実際のデータは、これらを手動で展開しない限り表示されません。最下位ソートフィールドのデータ値を展開することはできません。展開可能ソートフィールドのプラス記号 (+) をクリックするだけで、下位ソートフィールドのデータを表示することができます。最下位から 2 番目のソート項目を指定している

見出しのデータ値を展開すると、対応するデータがすべて表示されます。このタイプのアコーディオンレポートを生成するには、SET EXPANDABLE コマンドを使用します。

アコーディオンレポートを、ACROSS フィールドに指定することも可能です。ACROSS ソートフィールドの見出しは、標準の HTML レポートでは BY ソートフィールドの見出しの上部に表示されますが、アコーディオンレポートでは、各ソートフィールドの列で少なくとも 1 つのソートデータを手動で展開した段階で表示されます。

アコーディオンレポートを使用するには、BY 句に 2 つの BY フィールドが必要です。コマンド構文に 2 つ以上の BY 句が含まれていない場合、アコーディオンレポートの EXPANDABLE、EXPANDBYROW、EXPANDBYROWTREE コマンドは無視されます。この場合、メッセージは生成されず、標準の HTML レポートが作成されます。

注意：アコーディオンレポートは、HTML レポート出力でのみサポートされます。

アコーディオンレポートの要件

アコーディオンレポートを作成する際は、次の要件を考慮する必要があります。

- アコーディオンレポートにドリルダウンリンクを追加するには、TARGET パラメータに新しい HTML フレームを指定する必要があります。
- アコーディオンレポートの作成およびユーザへの配信完了後、ユーザがレポートで作業する際、WebFOCUS Reporting Server への呼び出しは行われません。ソートフィールドの折りたたみフォルダコントロールには、WebFOCUS Client の JavaScript が必要です。この機能を使用するには、ユーザが WebFOCUS Web 階層コンポーネントに接続している必要があります。ユーザが WebFOCUS へオンラインで接続している場合、レポートに変更を加える必要はありません。

  ただし、レポートを ReportCaster で配信する場合は、次の手順により、レポートが Email の添付ファイルまたは ReportLibrary のアーカイブレポートとして正しく配信されるようにします。

アコーディオンレポートでの HTMLARCHIVE 使用上の注意

WebFOCUS インタラクティブレポート機能が正常に動作するには、WebFOCUS Client に接続して、この機能に必要なコンポーネントにアクセスする必要があります。

WebFOCUS Client へのアクセスが利用できない場合は、HTMLARCHVE を使用して、ユーザ定義イメージが含まれた自己完結型 HTML ページを作成することができます。

ユーザ定義イメージが含まれた HTML ページを生成し、インタラクティブ動作を可能にするには、次のコマンドのいずれかを使用します。

```
SET HTMLEMBEDIMG=ON
SET HTMLARCHIVE=ON
```

次の構文を使用して、これらのファイルへのアクセスが可能なホストマシンを指定した BASEURL を定義します。

```
SET BASEURL=http://{hostname:portnumber}
```

SET BASEURL についての詳細は、ベース URL の指定を参照してください。

ReportCaster によるアコーディオンレポートの配信

アコーディオンレポートを ReportCaster で配信するには、WebFOCUS Client 上に存在する JavaScript コンポーネントおよびイメージを使用する必要があります。ReportCater で配信されたレポートからこれらのコンポーネントにアクセスするには、スケジュール済みプロシジャに SET FOCHTMLURL コマンドを記述する必要があります。このコマンドは、デフォルト値ではなく、絶対 URL に設定する必要があります。以下はその例です。

```
SET FOCHTMLURL = http://hostname[:port]/ibi_apps/ibi_html
```

説明

hostname[:port]

WebFOCUS Web アプリケーションの展開先のホスト名とオプションのポート番号です。ポート番号はデフォルト値以外を使用している場合に入力します。

ibi_apps/ibi_html

ibi_apps は、サイトごとに異なる WEBFOCUSxx/ibi_apps ディレクトリの Web サーバエイリアスです (ibi_apps はデフォルト値)。ibi_html は、アコーディオンレポートを表示する際にアクセスが必要となる JavaScript ファイルへのパス上のディレクトリです。

レポートを ReportCaster で使用するためのコーディングについての詳細は、『ibi™ WebFOCUS® ReportCaster 利用ガイド』の「ReportCaster レポートコーディング上のヒントとテクニック」を参照してください。

同一列内アコーディオンレポートの作成

同一列内でのアコーディオンレポートは、展開可能なツリー内にソートフィールドを表示することで、複数のレベルで集計したデータをインタラクティブなインターフェースで提供する HTML レポートです。デフォルト設定では、最上位のディメンション (BY ソートフィールド) と、BY ソートフィールド値ごとに集計された中間合計のみがレポートに表示されます。ツリーコントロールを使用して、各ディメンションを展開したり折りたたんだりして、それぞれのディメンションに関係する集計値を表示することができます。ソートフィールド値の横にあるプラス記号 (+) をクリックすると、新しい行が開いて、次の下位レベルのソートフィールド値および中間合計が表示されます。最下位のソートフィールドを開くと、集計に使用されたデータ値が表示されます。

SET EXPANDBYROW または SET EXPANDBYROWTREE コマンドを使用し、HTMLCSS を ON に設定することで、任意の HTML レポートを同一列内アコーディオンリクエストに変更することができます。EXPANDBYROW および EXPANDBYROWTREE は、自動的に SET SUBTOTALS=ABOVE コマンドを呼び出します。このコマンドにより、中間合計行が中間見出し行およびデータ行の上に移動されます。最後から 2 つ目の BY フィールドには、SUBTOTAL コマンドが追加されます。

同一列内でのアコーディオンレポートに PRINT コマンドを使用すると、ツリーの最下位の行にデータソースの明細レコードが表示されます。場合によっては、BY フィールドの組み合わせごとに明細レコードが多数表示され、レポート出力でこれらの明細レコードを識別することが困難になります。より役に立つレポートにするには、明細レベルの行を識別するためのフィールドを少なくとも 1 つレポートに含めます。

同一列内でのアコーディオンレポートに SUM コマンドを使用すると、ツリーの最下位の行を含む各行が中間合計行になります。これらの行は、リクエストの BY フィールドの組み合わせで記述されます。各レベルは集計レベルで表示され、データ値は最下位の BY の集計を表します。

同一列内でのアコーディオンレポートは、標準の HTML レポートの手法でスタイル設定することができますが、レポートの構造を念頭に置く必要があります。最下位以外のすべての行は、実際には SUBTOTAL 行で、最下位にはレポートのデータが表示されます。

同一列内でのアコーディオンレポートには、データの下に総合計がアンカー行として表示されます。このアンカー行は、レポートの左マージンを基準に位置揃えされたレポート脚注およびページ脚注の上に表示されます。総合計アンカー行を含めずに同一列内アコーディオンレポートを生成するには、リクエストに ON TABLE NOTOTAL を追加します。

同一列内アコーディオンレポートの新しいインターフェースでは、ポータルページで幅広い大規模レポートを操作する際のナビゲーションが容易になり、データがコンテナ内に収まるよう自動的にサイズ変更され、列幅はフィールド値の最大長または列タイトル幅のどちらか大きい方に自動調整されます。強化された同一列内アコーディオンレポート機能を有効にするには、プロシジャ内で SET EXPANDBYROWTREE=ON コマンドを使用します。プロシジャに SET AUTOFIT ON コマンドを指定すると、データのサイズがコンテナサイズに収まるように、自動的にサイズ変更できます。詳細は、アコーディオンレポート機能によるソートデータの表示制御を参照してください。

同一列内で展開するアコーディオンレポートの代わりに、列ごとに展開するアコーディオンレポートを作成することもできます。SET EXPANDABLE コマンドを使用してアコーディオンレポートを作成する方法についての詳細は、アコーディオンレポート機能によるソートデータの表示制御を参照してください。

同一列内で展開するアコーディオンレポートの作成

```
SET EXPANDBYROW = {OFF|ON|n}
```

```
ON TABLE SET EXPANDBYROW {OFF|ON|n}
```

説明

OFF

アコーディオンレポートを作成しません。デフォルト値は OFF です。

ON

アコーディオンレポートを作成し、初期状態で最上位のソートフィールドレベルのみを表示します。下位レベルの行を表示するには、表示されたいずれかのソートフィールド値の横にあるプラス記号 (+) をクリックします。

ALL

アコーディオンレポートを作成し、最初にすべてのソートフィールドレベルを展開した状態で表示します。ソートフィールドレベルを折りたたむには、そのレベルのいずれかのソートフィールド値の横にあるマイナス記号 (-) をクリックします。

n

当初展開されるソートフィールドのレベル数を *n* 個に設定してアコーディオンレポートを作成します。展開されたソートフィールドレベルを折りたたむには、そのレベルのいずれかのソートフィールド値の横にあるマイナス記号 (-) をクリックします。

注意

- 同一列内でのアコーディオンレポートを使用するには、HTMLCSS パラメータを ON に設定しておく必要があります。
- デフォルト設定では、ブランク行はレポート出力の中間合計の前に生成されます。これらの自動的なブランク行を削除するには、SET DROPBLNKLINE=ON コマンドを発行します。

SUM を使用した同一列内アコーディオンレポートの作成

次のリクエストは、GGSALES データソースに対して実行され、ソートフィールドとして REGION、ST、CATEGORY、PRODUCT が指定されています。

```
TABLE FILE GGSALES
SUM DOLLARS/D8MC
UNITS/D8C
BUDDOLLARS/D8MC BUDUNITS/D8C
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
ON TABLE SET HTMLCSS ON
ON TABLE SET EXPANDBYROW ON
ON TABLE SET DROPBLNKLINE ON
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET STYLE *
TYPE=REPORT,
    COLOR=RGB(66 70 73),
    FONT='TREBUCHET MS',
    SIZE=9,
    SQUEEZE=ON,
    GRID=OFF,
$
TYPE=REPORT,
    GRID=OFF,
```

```
    FONT='TREBUCHET MS',
    COLOR=RGB(52 85 64),
$
```

```
TYPE=TITLE,
    COLOR='WHITE',
    BACKCOLOR=RGB(52 85 64),
    STYLE=-UNDERLINE,
$
TYPE=HEADING,
    COLOR='WHITE',
    BACKCOLOR=RGB(52 85 64),
$
TYPE=FOOTING,
    COLOR='WHITE',
    BACKCOLOR=RGB(52 85 64),
$
TYPE=SUBTOTAL,
    BACKCOLOR=RGB(72 118 91),
$
TYPE=SUBTOTAL,
    BY=1,
    COLOR='WHITE',
$
TYPE=SUBTOTAL,
    BY=2,
    COLOR='WHITE',
    BACKCOLOR=RGB(132 159 126),
$
TYPE=SUBTOTAL,
    BY=3,
    COLOR='WHITE',
    BACKCOLOR=RGB(158 184 153),
$
TYPE=GRANDTOTAL,
    COLOR='WHITE',
    BACKCOLOR=RGB(52 85 64),
    STYLE=BOLD,
$
ENDSTYLE
END
```

下図のように、初期状態の出力では、最上位の BY フィールド (REGION) のみが表示されます。

	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
⊞ Midwest	$11,400,665	905,045	$11,194,373	907,107
⊞ Northeast	$11,392,300	916,675	$11,576,921	914,359
⊞ Southeast	$11,710,379	935,232	$11,807,971	942,247
⊞ West	$11,652,946	932,039	$11,641,513	930,781
TOTAL	$46,156,290	3,688,991	$46,220,778	3,694,494

下図のように、Midwest 地域の横にあるプラス記号 (+) をクリックすると、その地域に関連する州がそれぞれの行に表示されます。

	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
⊟ Midwest	$11,400,665	905,045	$11,194,373	907,107
⊞ IL	$3,924,401	307,581	$3,866,856	310,071
⊞ MO	$3,761,286	297,727	$3,646,838	297,081
⊞ TX	$3,714,978	299,737	$3,680,679	299,955
⊞ Northeast	$11,392,300	916,675	$11,576,921	914,359
⊞ Southeast	$11,710,379	935,232	$11,807,971	942,247
⊞ West	$11,652,946	932,039	$11,641,513	930,781
TOTAL	$46,156,290	3,688,991	$46,220,778	3,694,494

下図のように、IL 州の横にあるプラス記号 (+) をクリックすると、その州に関連するカテゴリがそれぞれの行に表示されます。

	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
⊟ Midwest	$11,400,665	905,045	$11,194,373	907,107
⊟ IL	$3,924,401	307,581	$3,866,856	310,071
⊞ Coffee	$1,398,779	109,581	$1,366,264	111,431
⊞ Food	$1,522,847	118,068	$1,480,855	118,363
⊞ Gifts	$1,002,775	79,932	$1,019,737	80,277
⊞ MO	$3,761,286	297,727	$3,646,838	297,081
⊞ TX	$3,714,978	299,737	$3,680,679	299,955
⊞ Northeast	$11,392,300	916,675	$11,576,921	914,359
⊞ Southeast	$11,710,379	935,232	$11,807,971	942,247
⊞ West	$11,652,946	932,039	$11,641,513	930,781
TOTAL	$46,156,290	3,688,991	$46,220,778	3,694,494

下図のように、Coffee カテゴリの横にあるプラス記号 (+) をクリックすると、そのカテゴリに関連する製品が表示されます。この行が、同一列内でのアコーディオンレポートの最下位です。

	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
⊟ Midwest	$11,400,665	905,045	$11,194,373	907,107
⊟ IL	$3,924,401	307,581	$3,866,856	310,071
⊟ Coffee	$1,398,779	109,581	$1,366,264	111,431
Espresso	$420,439	32,237	$401,477	32,416
Latte	$978,340	77,344	$964,787	79,015
⊞ Food	$1,522,847	118,068	$1,480,855	118,363
⊞ Gifts	$1,002,775	79,932	$1,019,737	80,277
⊞ MO	$3,761,286	297,727	$3,646,838	297,081
⊞ TX	$3,714,978	299,737	$3,680,679	299,955
⊞ Northeast	$11,392,300	916,675	$11,576,921	914,359
⊞ Southeast	$11,710,379	935,232	$11,807,971	942,247
⊞ West	$11,652,946	932,039	$11,641,513	930,781
TOTAL	$46,156,290	3,688,991	$46,220,778	3,694,494

PRINT を使用した同一列内アコーディオンレポートの作成

次のリクエストは、EMPLOYEE データソースに対して実行され、DEPARTMENT および YEAR の 2 つのソートフィールドが指定されています。このリクエストでは、PRINT コマンドを使用します。

```
SET EXPANDBYROW = ALL
DEFINE FILE EMPLOYEE
YEAR/YY = HIRE_DATE;
YEARMO/YYM = HIRE_DATE;
END
TABLE FILE EMPLOYEE
PRINT LAST_NAME AS 'Last,Name' FIRST_NAME AS 'First,Name'
CURR_SAL AS 'Current,Salary' ED_HRS AS 'Education,Hours'
BY DEPARTMENT BY YEAR
WHERE YEAR GT 1980
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
TYPE=REPORT,
    COLOR=RGB(66 70 73),
    FONT='TREBUCHET MS',
    SIZE=9,
    SQUEEZE=ON,
    GRID=OFF,
$
```

```
TYPE=TITLE,
    BACKCOLOR=RGB(102 102 102),
    COLOR=RGB(255 255 255),
    STYLE=-UNDERLINE+BOLD,
$
TYPE=DATA,
    BACKCOLOR=RGB(255 255 255),
$
TYPE=TITLE,
    COLOR='WHITE',
    BACKCOLOR=RGB(52 85 64),
    STYLE=-UNDERLINE,
$
TYPE=HEADING,
    COLOR='WHITE',
    BACKCOLOR=RGB(52 85 64),
$
TYPE=FOOTING,
    COLOR='WHITE',
    BACKCOLOR=RGB(52 85 64),
$
TYPE=SUBTOTAL,
    BACKCOLOR=RGB(72 118 91),
$
TYPE=SUBTOTAL,
    BY=1,
    COLOR='WHITE',
$
TYPE=SUBTOTAL,
    BY=2,
    COLOR='WHITE',
    BACKCOLOR=RGB(132 159 126),
$
TYPE=SUBTOTAL,
    BY=3,
    COLOR='WHITE',
    BACKCOLOR=RGB(158 184 153),
$
TYPE=GRANDTOTAL,
    COLOR='WHITE',
    BACKCOLOR=RGB(52 85 64),
    STYLE=BOLD,
$
ENDSTYLE
END
```

レポート出力に LAST_NAME および FIRST_NAME フィールドを含めることで、明細レコードの各行が識別されます。ただし、これらのフィールドは集計行に適用されないため、集計行ではブランクになります。

出力結果は次のとおりです。

	Last Name	First Name	Current Salary	Education Hours
⊟ MIS			$108,002.00	231.00
⊟ 1981			$58,742.00	81.00
	SMITH	MARY	$13,200.00	36.00
	MCCOY	JOHN	$18,480.00	.00
	CROSS	BARBARA	$27,062.00	45.00
⊟ 1982			$49,260.00	150.00
	JONES	DIANE	$18,480.00	50.00
	BLACKWOOD	ROSEMARIE	$21,780.00	75.00
	GREENSPAN	MARY	$9,000.00	25.00
⊟ PRODUCTION			$103,282.00	95.00
⊟ 1982			$103,282.00	95.00
	SMITH	RICHARD	$9,500.00	10.00
	BANNING	JOHN	$29,700.00	.00
	IRVING	JOAN	$26,862.00	30.00
	ROMANS	ANTHONY	$21,120.00	5.00
	MCKNIGHT	ROGER	$16,100.00	50.00
TOTAL			$211,284.00	326.00

強化されたインターフェースによるアコーディオンレポートの作成

```
SET EXPANDBYROWTREE = {OFF|ON|ALL|n}
ON TABLE SET EXPANDBYROWTREE {OFF|ON|ALL|n}
```

説明

OFF

強化されたインターフェースで、アコーディオンレポートを作成しません。デフォルト値は OFF です。

ON

強化されたインターフェースで、アコーディオンレポートを作成します。この設定では、初期状態で最上位のソートフィールドレベルのみを表示します。下位レベルの行を表示するには、表示されたいずれかのソートフィールド値の横にあるプラス記号 (+) をクリックします。

ALL

強化されたインターフェースで、アコーディオンレポートを作成します。この設定では、初期状態ですべてのソートフィールドレベルが展開されて表示されます。ソートフィールドレベルを折りたたむには、そのレベルのいずれかのソートフィールド値の横にあるマイナス記号 (-) をクリックします。

n

強化されたインターフェースで、アコーディオンレポートを作成します。この設定では、初期状態で n 個のソートフィールドレベルが展開されて表示されます。展開されたソートフィールドレベルを折りたたむには、そのレベルのいずれかのソートフィールド値の横にあるマイナス記号 (-) をクリックします。

強化されたインターフェースによるアコーディオンレポートの作成

次のリクエストは、GGSALES データソースに対して実行され、ソートフィールドとして REGION、ST、CATEGORY、PRODUCT が指定されています。このリクエストはデフォルト

設定のスタイルシートを使用し、デフォルト設定のプラス記号 (+) およびマイナス記号 (-) を使用して、行の展開および折りたたみを行います。強化されたインターフェースのアコーディオンレポートを作成するため、SET EXPANDBYROWTREE コマンドを ON に設定する必要があります。サイズを自動的に変更してデータがコンテナサイズに収まるようにするには、SET AUTOFIT コマンドを ON に設定する必要があります。

```
TABLE FILE GGSALES
SUM DOLLARS/D8MC
UNITS/D8C
BUDDOLLARS/D8MC BUDUNITS/D8C
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
ON TABLE SET EXPANDBYROWTREE ON
ON TABLE SET DROPBLNKLINE ON
ON TABLE SET AUTOFIT ON
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/ibi_themes/Warm.sty,$
ENDSTYLE
END
```

下図のように、初期状態の出力では、最上位の BY フィールド (REGION) のみが表示されます。

	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
+ Midwest	$11,400,665	905,045	$11,194,373	907,107
+ Northeast	$11,392,300	916,675	$11,576,921	914,359
+ Southeast	$11,710,379	935,232	$11,807,971	942,247
+ West	$11,652,946	932,039	$11,641,513	930,781
TOTAL	$46,156,290	3,688,991	$46,220,778	3,694,494

下図のように、Midwest 地域の横にあるプラス記号 (+) をクリックすると、その地域に関連する州がそれぞれの行に表示されます。

	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
− Midwest	$11,400,665	905,045	$11,194,373	907,107
+ IL	$3,924,401	307,581	$3,866,856	310,071
+ MO	$3,761,286	297,727	$3,646,838	297,081
+ TX	$3,714,978	299,737	$3,680,679	299,955
+ Northeast	$11,392,300	916,675	$11,576,921	914,359
+ Southeast	$11,710,379	935,232	$11,807,971	942,247
+ West	$11,652,946	932,039	$11,641,513	930,781
TOTAL	$46,156,290	3,688,991	$46,220,778	3,694,494

下図のように、IL 州の横にあるプラス記号 (+) をクリックすると、その州に関連するカテゴリがそれぞれの行に表示されます。

	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
− Midwest	$11,400,665	905,045	$11,194,373	907,107
− IL	$3,924,401	307,581	$3,866,856	310,071
+ Coffee	$1,398,779	109,581	$1,366,264	111,431
+ Food	$1,522,847	118,068	$1,480,855	118,363
+ Gifts	$1,002,775	79,932	$1,019,737	80,277
+ MO	$3,761,286	297,727	$3,646,838	297,081
+ TX	$3,714,978	299,737	$3,680,679	299,955
+ Northeast	$11,392,300	916,675	$11,576,921	914,359
+ Southeast	$11,710,379	935,232	$11,807,971	942,247
+ West	$11,652,946	932,039	$11,641,513	930,781
TOTAL	$46,156,290	3,688,991	$46,220,778	3,694,494

下図のように、Coffee カテゴリの横にあるプラス記号 (+) をクリックすると、そのカテゴリに関連する製品が表示されます。この行が、同一列内でのアコーディオンレポートの最下位です。

	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
− Midwest	$11,400,665	905,045	$11,194,373	907,107
− IL	$3,924,401	307,581	$3,866,856	310,071
− Coffee	$1,398,779	109,581	$1,366,264	111,431
Espresso	$420,439	32,237	$401,477	32,416
Latte	$978,340	77,344	$964,787	79,015
+ Food	$1,522,847	118,068	$1,480,855	118,363
+ Gifts	$1,002,775	79,932	$1,019,737	80,277
+ MO	$3,761,286	297,727	$3,646,838	297,081
+ TX	$3,714,978	299,737	$3,680,679	299,955
+ Northeast	$11,392,300	916,675	$11,576,921	914,359
+ Southeast	$11,710,379	935,232	$11,807,971	942,247
+ West	$11,652,946	932,039	$11,641,513	930,781
TOTAL	$46,156,290	3,688,991	$46,220,778	3,694,494

EBRT_ANCHOR スタイルシート属性を使用して、デフォルト設定のプラス記号 (+) とマイナス記号 (-) を矢印に変更することができます。EBRT_ANCHOR 属性で有効な設定は、「PLUSMINUS」および「ARROWS」です。次のリクエストは、デフォルト設定のプラス記号 (+) とマイナス記号 (-) を矢印に変更した上で、リクエストにスタイルシートフォーマットを適用してテキストの色を白、背景色を色合いの異なる紫色に設定します。

注意：矢印は、SUBTOTAL 行と一致する色に設定されます (ここでは白)。

```
TABLE FILE GGSALES
SUM DOLLARS/D8MC
UNITS/D8C
BUDDOLLARS/D8MC BUDUNITS/D8C
```

```
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
ON TABLE SET EXPANDBYROWTREE ON
ON TABLE SET DROPBLNKLINE ON
ON TABLE SET AUTOFIT ON
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET STYLE *
TYPE=REPORT, EBRT_ANCHOR=ARROWS,
COLOR=RGB(66 70 73), FONT='TREBUCHET MS', SIZE=9, SQUEEZE=ON,$
TYPE=REPORT, FONT='TREBUCHET MS', COLOR=RGB(151 43 153),$
TYPE=TITLE, COLOR='WHITE', BACKCOLOR=RGB(151 43 153), STYLE=-UNDERLINE,$
TYPE=HEADING, COLOR='WHITE', BACKCOLOR=RGB(151 43 153),$
TYPE=FOOTING, COLOR='WHITE', BACKCOLOR=RGB(151 43 153),$
TYPE=SUBTOTAL, COLOR=WHITE, BACKCOLOR=RGB(179 72 180),$
TYPE=SUBTOTAL, BY=2, BACKCOLOR=RGB(208 99 208),$
TYPE=SUBTOTAL, BY=3, BACKCOLOR=RGB(237 127 236),$
TYPE=GRANDTOTAL, COLOR='WHITE', BACKCOLOR=RGB(151 43 153), STYLE=BOLD,$
ENDSTYLE
END
```

下図のように、初期状態の出力では、最上位の BY フィールド (REGION) のみが表示されます。

	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
▸ Midwest	$11,400,665	905,045	$11,194,373	907,107
▸ Northeast	$11,392,300	916,675	$11,576,921	914,359
▸ Southeast	$11,710,379	935,232	$11,807,971	942,247
▸ West	$11,652,946	932,039	$11,641,513	930,781
TOTAL	$46,156,290	3,688,991	$46,220,778	3,694,494

下図のように、Midwest 地域の横にある矢印をクリックすると、その地域に関連する州がそれぞれの行に表示されます。

	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
▾ Midwest	$11,400,665	905,045	$11,194,373	907,107
▸ IL	$3,924,401	307,581	$3,866,856	310,071
▸ MO	$3,761,286	297,727	$3,646,838	297,081
▸ TX	$3,714,978	299,737	$3,680,679	299,955
▸ Northeast	$11,392,300	916,675	$11,576,921	914,359
▸ Southeast	$11,710,379	935,232	$11,807,971	942,247
▸ West	$11,652,946	932,039	$11,641,513	930,781
TOTAL	$46,156,290	3,688,991	$46,220,778	3,694,494

下図のように、IL 州の横にある右矢印をクリックすると、その州に関連するカテゴリがそれぞれの行に表示されます。

	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
▼ Midwest	$11,400,665	905,045	$11,194,373	907,107
▼ IL	$3,924,401	307,581	$3,866,856	310,071
► Coffee	$1,398,779	109,581	$1,366,264	111,431
► Food	$1,522,847	118,068	$1,480,855	118,363
► Gifts	$1,002,775	79,932	$1,019,737	80,277
► MO	$3,761,286	297,727	$3,646,838	297,081
► TX	$3,714,978	299,737	$3,680,679	299,955
► Northeast	$11,392,300	916,675	$11,576,921	914,359
► Southeast	$11,710,379	935,232	$11,807,971	942,247
► West	$11,652,946	932,039	$11,641,513	930,781
TOTAL	$46,156,290	3,688,991	$46,220,778	3,694,494

下図のように、Coffee カテゴリの横にある右矢印をクリックすると、そのカテゴリに関連する製品が表示されます。この行が、同一列内でのアコーディオンレポートの最下位です。

	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
▼ Midwest	$11,400,665	905,045	$11,194,373	907,107
▼ IL	$3,924,401	307,581	$3,866,856	310,071
▼ Coffee	$1,398,779	109,581	$1,366,264	111,431
Espresso	$420,439	32,237	$401,477	32,416
Latte	$978,340	77,344	$964,787	79,015
► Food	$1,522,847	118,068	$1,480,855	118,363
► Gifts	$1,002,775	79,932	$1,019,737	80,277
► MO	$3,761,286	297,727	$3,646,838	297,081
► TX	$3,714,978	299,737	$3,680,679	299,955
► Northeast	$11,392,300	916,675	$11,576,921	914,359
► Southeast	$11,710,379	935,232	$11,807,971	942,247
► West	$11,652,946	932,039	$11,641,513	930,781
TOTAL	$46,156,290	3,688,991	$46,220,778	3,694,494

矢印の色を指定するには、SUBTOTAL 行で CONTROLCOLOR スタイルシート属性を使用します。次の構文は、矢印の色を紫色に変更する方法を示しています。

```
TABLE FILE GGSALES
SUM DOLLARS/D8MC
UNITS/D8C
BUDDOLLARS/D8MC BUDUNITS/D8C
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
ON TABLE SET EXPANDBYROWTREE ON
ON TABLE SET DROPBLNKLINE ON
ON TABLE SET AUTOFIT ON
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET STYLE *
```

```
TYPE=REPORT, EBRT_ANCHOR=ARROWS,
COLOR=RGB(66 70 73), FONT='TREBUCHET MS', SIZE=9, SQUEEZE=ON,$
TYPE=REPORT, FONT='TREBUCHET MS', COLOR=RGB(151 43 153),$
TYPE=TITLE, COLOR='WHITE', BACKCOLOR=RGB(151 43 153), STYLE=-UNDERLINE,$
TYPE=HEADING, COLOR='WHITE', BACKCOLOR=RGB(151 43 153),$
TYPE=FOOTING, COLOR='WHITE', BACKCOLOR=RGB(151 43 153),$
TYPE=SUBTOTAL, COLOR=WHITE, BACKCOLOR=RGB(179 72 180), CONTROLCOLOR=PURPLE,$
TYPE=SUBTOTAL, BY=2, BACKCOLOR=RGB(208 99 208),$
TYPE=SUBTOTAL, BY=3, BACKCOLOR=RGB(237 127 236),$
TYPE=GRANDTOTAL, COLOR='WHITE', BACKCOLOR=RGB(151 43 153), STYLE=BOLD,$
ENDSTYLE
END
```

次の出力結果では、同一列内アコーディオンレポートの最下位行が、紫色で表示されています。

	Dollar Sales	Unit Sales	Budget Dollars	Budget Units
▼ Midwest	$11,400,665	905,045	$11,194,373	907,107
▼ IL	$3,924,401	307,581	$3,866,856	310,071
▼ Coffee	$1,398,779	109,581	$1,366,264	111,431
Espresso	$420,439	32,237	$401,477	32,416
Latte	$978,340	77,344	$964,787	79,015
► Food	$1,522,847	118,068	$1,480,855	118,363
► Gifts	$1,002,775	79,932	$1,019,737	80,277
► MO	$3,761,286	297,727	$3,646,838	297,081
► TX	$3,714,978	299,737	$3,680,679	299,955
► Northeast	$11,392,300	916,675	$11,576,921	914,359
► Southeast	$11,710,379	935,232	$11,807,971	942,247
► West	$11,652,946	932,039	$11,641,513	930,781
TOTAL	$46,156,290	3,688,991	$46,220,778	3,694,494

EXPANDBYROW および EXPANDBYROWTREE 使用上の注意

- バージョン 8.2 SP04 以降では、EXPANDBYROWTREE でグリッドがサポートされます。
- BY フィールドの最大長は 245 バイトです。
- EXPANDBYROWTREE を OLAP とともいに使用することはできません。OLAP と EXPANDBYROWTREE の両方を有効にした場合、EXPANDBYROWTREE は無視されま

す。この問題は回避するには、EXPANDBYROW を使用します。

- EXPANDBYROWTREE を AHTML 出力フォーマットで使用することはできません。EXPANDBYROWTREE を Analytic Document で使用すると、EdeXPANDBYROWTREE は無視されます。
- SQL Server Analysis Services (SSAS) キューブデータソースを使用してアコーディオン概要レポートを実行した場合、データの取得時に内部的に PRINT コマンドが使用されるため、表示されるレポート出力は、概要レポートではなく詳細レポートになります。これは、SSAS キューブデータソースには、事前集計済みデータが格納されることから、SUM コマンドは内部的に PRINT コマンドに変更されためです。

同一列内アコーディオンレポートのツールヒント

EXPANDBYROW および EXPANDBYROWTREE レポートのデフォルト設定では、ツリーの各レベルで値の上にマウスポインタを置くと、ツールヒントにフィールド情報が表示されます。標準の HTML レポートでは列の最上部にフィールドタイトルが表示されますが、同一列内アコーディオンレポートのツリーコントロール列の最上部にはフィールドタイトルが表示されません。代わりに、レポートの左上 (ツリーコントロール列の最上部) にマウスポインタを置くと、ツールヒントにツリーのフィールドリストが表示されます。

同一列内アコーディオンレポートでは、ポップアップフィールド説明を有効にすることで、マスターファイルに記述されているフィールド説明、または特定のフィールドに関連する DEFINE で定義されたフィールド説明を表示することができます。

タイトルは、AS 名を定義することでカスタマイズすることができます。展開可能ツリーにポップアップフィールド説明を表示しないよう設定するには、フィールドタイトルに対してブランクの AS 名を定義します。同一列内アコーディオンレポートでは、マウスポインタを置いたときに表示されるツールヒントには、レポート内のフィールドに関する追加情報が表示されます。標準のアコーディオンレポートでは、これらのツールヒントには、展開可能ツリー内のすべての BY 値のフィールドタイトルまたは AS 名が表示されます。

標準の HTML レポートと同様に、POPUPDESC パラメータを ON に設定することで、すべての動詞フィールドのツールヒントにフィールド説明を表示することができます。また、POPUPDESC パラメータを ON に設定すると、BY フィールドのツールヒントにもフィールド説明の値が表示されます (説明が存在する場合)。

下表は、EXPANDBYROW または EXPANDBYROWTREE 設定が ON の場合に、ツールヒントに表示される説明の優先順位を示しています。

既存のフィールド情報	ポップアップ説明オフ	ポップアップ説明オン
説明		1
AS 名	1	2
フィールドタイトル	2	3
フィールド名	3	4

ツールヒントおよびポップアップフィールド説明の色およびサイズは、すべてのレポートの外観を統一するために標準化されています。

同一列内アコーディオンレポートの作成 - ポップアップフィールド説明をオフ

次の例では、デフォルト表示で (ポップアップフィールド説明はオフ)、標準のアコーディオンレポートのツールヒントがどのように表示されるかについて説明します。

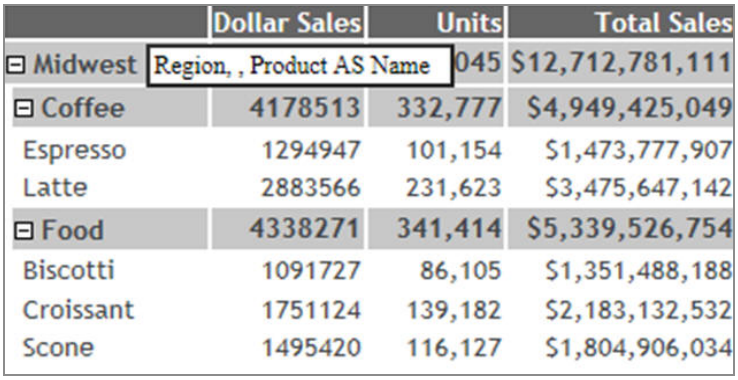
```
DEFINE FILE GGSALES.
UNITS/D12C DESCRIPTION ''=UNITS;
TOTSALES/D12CM DESCRIPTION 'DOLLARS*UNITS'=DOLLARS*UNITS;
END
TABLE FILE GGSALES
SUM DOLLARS UNITS AS 'Units'
TOTSALES AS 'Total Sales'
BY REGION
BY CATEGORY AS ''
BY PRODUCT AS 'Product AS Name'
ON TABLE SET EXPANDBYROW ALL
ON TABLE SET DROPBLNKLINE ALL
ON TABLE SET STYLE *
TYPE=REPORT,
    COLOR=RGB(66 70 73),
    FONT='TREBUCHET MS',
    SIZE=9,
    SQUEEZE=ON,
    GRID=OFF,
$
TYPE=TITLE,
```

```
    BACKCOLOR=RGB(102 102 102),
    COLOR=RGB(255 255 255),
    STYLE=-UNDERLINE+BOLD,
$
TYPE=DATA,
    BACKCOLOR=RGB(255 255 255),
$
TYPE=SUBTOTAL,
    BACKCOLOR=RGB(200 200 200),
    STYLE=BOLD,
 $
TYPE=GRANDTOTAL,
    BACKCOLOR=RGB(66 70 73),
    COLOR=RGB(255 255 255),
    STYLE=BOLD,
$
ENDSTYLE
END
```

マスターファイルでは、次のようにフィールドが定義されています。

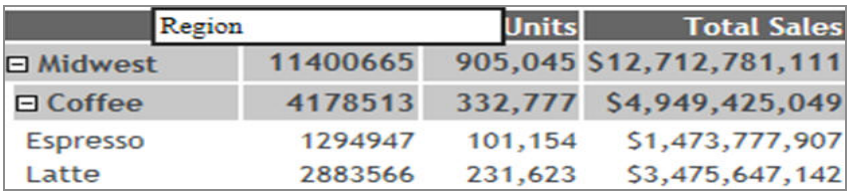
```
FIELD=CATEGORY, ALIAS=E02, FORMAT=A11, INDEX=I, TITLE='Category',
   DESC='Product category',$
FIELD=PRODUCT, ALIAS=E04, FORMAT=A16, TITLE='Product', DESC='Product name',$
 FIELD=REGION, ALIAS=E05, FORMAT=A11, INDEX=I, TITLE='Region',
   DESC='Region code',$
FIELD=UNITS, ALIAS=E10, FORMAT=I08, TITLE='Unit Sales',
   DESC='Number of units sold',$
```

下図は、テーブルの左上 (ツリーコントロール列の最上部) にマウスポインタを置いたときのツールヒントを示しています。このツールヒントには、ツリーコントロール内の BY フィールドのフィールドタイトルまたは AS 名のリストが表示されます。



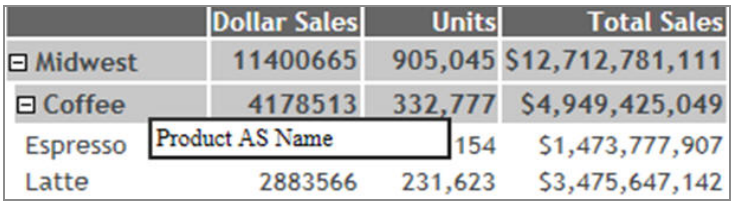
下図は、最上位 BY フィールド値のいずれかにマウスポインタを置いたときのツールヒントを示しています。この最上位 BY フィールドには AS 名は指定されていませんが、フィールド

説明およびタイトルが定義されています。この場合、フィールドのタイトルが表示されます。



Region		Units	Total Sales
⊟ Midwest	11400665	905,045	$12,712,781,111
⊟ Coffee	4178513	332,777	$4,949,425,049
Espresso	1294947	101,154	$1,473,777,907
Latte	2883566	231,623	$3,475,647,142

最上位 BY フィールドに AS 名が指定され、かつタイトルが定義されている場合、ツールヒントには AS 名が表示されます。



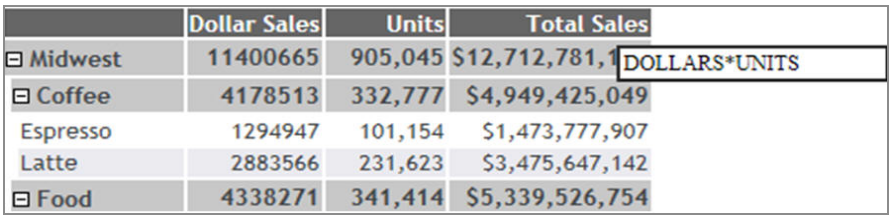
	Dollar Sales	Units	Total Sales
⊟ Midwest	11400665	905,045	$12,712,781,111
⊟ Coffee	4178513	332,777	$4,949,425,049
Espresso	Product AS Name	154	$1,473,777,907
Latte	2883566	231,623	$3,475,647,142

同一列内アコーディオンレポートの作成 - ポップアップフィールド説明をオン

次の例では、ポップアップフィールド説明をオンにした場合に、ツールヒントがどのように表示されるかについて説明します。

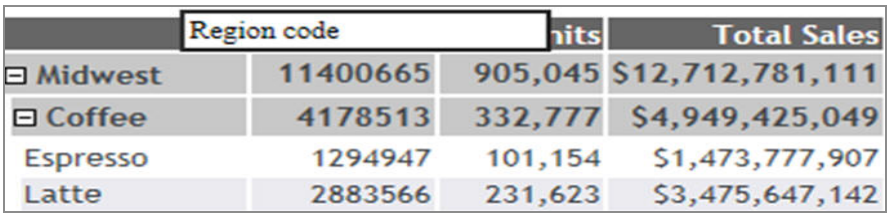
```
SET POPUPDESC = ON
DEFINE FILE GGSALES.
UNITS/D12C DESCRIPTION ''=UNITS;
TOTSALES/D12CM DESCRIPTION 'DOLLARS*UNITS'=DOLLARS*UNITS;
END
TABLE FILE GGSALES
SUM DOLLARS UNITS AS 'Units'
TOTSALES AS 'Total Sales'
BY REGION
BY CATEGORY AS ''
BY PRODUCT AS 'Product AS Name'
ON TABLE SET EXPANDBYROW ALL
ON TABLE SET DROPBLNKLINE ALL
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/javaassist/intl/EN/ENIADefault_combine.sty,$
ENDSTYLE
END
```

すべての HTML レポートと同様に、POPUPDESC=ON 設定を使用すると、各動詞フィールドタイトルのツールヒントにフィールド説明が表示されます。



また、POPUPDESC を ON に設定すると、ツリー内の BY フィールド値にもポップアップフィールド説明が表示されます。フィールド説明が定義されていない場合は、フィールドタイトルまたは AS 名が表示されます。

下図のように、POPUPDESC=ON 設定では、定義されているフィールド説明がツールヒントに表示されます。



上図のように、このフィールドに AS 名が指定されている場合でも、ツールヒントには BY フィールド値のフィールド説明が表示されます。

HTML レポートでのポップアップフィールド説明についての詳細は、レポートデータの表示を参照してください。

同一列内アコーディオンレポートでの NOPRINT の使用

同一列内アコーディオンレポートでは、非表示 (NOPRINT) の BY フィールドを使用することができます。非表示 BY フィールドを使用することで、レポートに表示されていないフィールドの値を計算したり、データをソートしたりできます。NOPRINT ソートフィールドは、内部マトリックスに格納され、このフィールドがレポートに表示されていない場合でも、アコーディオンレポートのデータのソートおよび集計に反映されます。NOPRINT ソートフィールドを定義するには、BY sortfield NOPRINT 句を使用します。

注意

- 非表示 (NOPRINT) フィールドは、ツールヒントおよびポップアップフィールド説明に

は表示されません。

- ブランクの AS 名を使用する際に、引用符の間にブランクを追加すると (例、「BY *fieldname* ' '」)、このブランクは削除され、作用は「BY *fieldname* ''」を指定した場合と同一になります。

同一列内アコーディオンレポートの作成 - 明示的な NOPRINT の使用

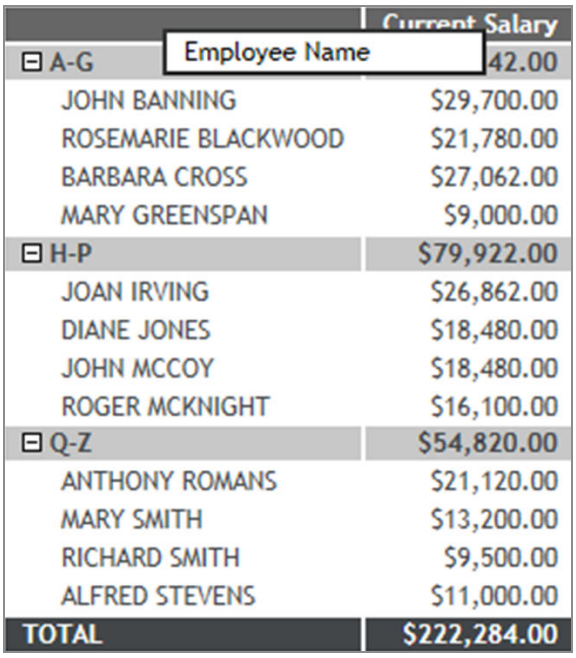
次のリクエストは、EMPLOYEE データソースを使用して、従業員の給与データをカテゴリ別に分類します。LAST_NAME フィールドと FIRST_NAME フィールドを連結して「NAME_SORT」という一時項目 (DEFINE) を作成し、このフィールドで出力結果をソートします。NAME_SORT フィールドは、ソート句に NOPRINT を指定して非表示にします。従業員名を表示するために、FIRST_NAME フィールドと LAST_NAME フィールドを連結して「NAME_DISPLAY」という一時項目 (DEFINE) を作成します。

```
DEFINE FILE EMPLOYEE
NAME_SORT/A50=EMPLOYEE.EMPINFO.LAST_NAME || ( ', ' | EMPLOYEE.EMPINFO.FIRST_
NAME );
NAME_DISPLAY/A57=EMPLOYEE.EMPINFO.FIRST_NAME | EMPLOYEE.EMPINFO.LAST_NAME;
NAME_CODE/A1=EDIT(LAST_NAME, '9');
NAME_GROUP/A10=IF NAME_CODE LE 'G' THEN 'A-G' ELSE IF NAME_CODE LE 'P'
 THEN 'H-P' ELSE 'Q-Z';
END
TABLE FILE EMPLOYEE
SUM
EMPLOYEE.EMPINFO.CURR_SAL AS 'Current Salary'
BY NAME_GROUP AS 'Alphabetical Group'
BY LOWEST NAME_SORT NOPRINT
BY LOWEST NAME_DISPLAY AS 'Employee Name'
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE SET EXPANDBYROW 2
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET DROPBLNKLINE ON
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
TYPE=REPORT,
    COLOR=RGB(66 70 73),
    FONT='TREBUCHET MS',
    SIZE=9,
    SQUEEZE=ON,
```

```
    GRID=OFF,
$
```

```
  TYPE=TITLE,
      BACKCOLOR=RGB(102 102 102),
      COLOR=RGB(255 255 255),
      STYLE=-UNDERLINE+BOLD,
  $
  TYPE=DATA,
      BACKCOLOR=RGB(255 255 255),
  $
  TYPE=SUBTOTAL
      BY=1,
      BACKCOLOR=RGB(200 200 200),
      STYLE=BOLD,
  $
  TYPE=SUBTOTAL,
      BY=2,
      BACKCOLOR=RGB(200 220 220),
      STYLE=BOLD,
  $
  TYPE=SUBTOTAL,
      BY=3,
      BACKCOLOR=RGB(220 220 200),
      STYLE=BOLD,
  $
  TYPE=GRANDTOTAL,
      BACKCOLOR=RGB(66 70 73),
      COLOR=RGB(255 255 255),
      STYLE=BOLD,
  $
  ENDSTYLE
  END
```

下図のように、NAME_DISPLAY 列の値の上にマウスポインタを置くと、ツールヒントに「Employee Name」という AS 名が表示されます。



BY フィールドのフォーマット再設定と再定義の相違点

ソートフィールドのフォーマットを動的に再設定すると、元のフィールドとフォーマットが再設定されたフィールドの両方が内部マトリックスに格納されます。元のフィールドは表示されませんが、値のソートや集計に使用されます。

再定義されたフィールドを使用する場合、値の表示、ソート、集計に、新しいフィールドが使用されます。

同一列内アコーディオンレポートの作成 - BY フィールドの動的なフォーマット再設定

次のリクエストは、EMPLOYEE データソースを使用して、雇用年別の従業員名および給与を表示します。表示フィールドの HIRE_DATE および CURR_SAL は、YY でフォーマットが再設定された HIRE_DATE、および一時項目 (DEFINE) の NAME_DISPLAY (従業員名) でソートされます。

```
DEFINE FILE EMPLOYEE
NAME_SORT/A50=EMPLOYEE.EMPINFO.LAST_NAME || ( ', ' |
 EMPLOYEE.EMPINFO.FIRST_NAME ); NAME_DISPLAY/A57=EMPLOYEE.EMPINFO.FIRST_NAME
| EMPLOYEE.EMPINFO.LAST_NAME;NAME_CODE/A1=EDIT(LAST_NAME, '9'); NAME_
GROUP/A10=IF NAME_CODE LE 'G'
 THEN 'A-G' ELSE IF NAME_CODE LE  'P' THEN 'H-P' ELSE 'Q-Z';
END
TABLE FILE EMPLOYEE
SUM
HIRE_DATE
EMPLOYEE.EMPINFO.CURR_SAL
BY HIRE_DATE/YY
BY LOWEST NAME_DISPLAY AS 'Employee Name'
ON TABLE SET PAGE-NUM NOLEAD
WHERE HIRE_DATE LT '820101';
ON TABLE SET EXPANDBYROW ALL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET DROPBLNKLINE ON
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/javaassist/intl/EN/ENIADefault_combine.sty,$
ENDSTYLE
END
```

生成されたレポートでは、同一年 (1981) のデータが 2 つの展開可能グループ (ノード) に分割されています。これは、値の表示にはフォーマットが再設定された値が使用され、値のソートおよび集計には元の値が使用されているためです。このレポートでは、2 つの異なる日付 (81/07/01、81/11/02) を表すために、1981 が共通値として使用されています。ソートは、年ではなく、日付に基づいて実行されています。

	HIRE_DATE	CURR_SAL
⊟ 1980		$11,000.00
ALFRED STEVENS	80/06/02	$11,000.00
⊟ 1981		$31,680.00
JOHN MCCOY	81/07/01	$18,480.00
MARY SMITH	81/07/01	$13,200.00
⊟ 1981		$27,062.00
BARBARA CROSS	81/11/02	$27,062.00
TOTAL		$69,742.00

同一列内アコーディオンレポートの作成 - BY フィールドの再定義

元のフィールド値の代わりに、フォーマットが再設定されたフィールド値でデータをソートするには、BY 値を格納する一時項目 (DEFINE) を作成し、新しいフォーマットを適用します。これにより、値の表示、ソート、集計が、再定義された新しい BY フィールド値に基づいて実行されます。

```
DEFINE FILE EMPLOYEE
NAME_SORT/A50=EMPLOYEE.EMPINFO.LAST_NAME || ( ', ' | EMPLOYEE.EMPINFO.FIRST_
NAME );
NAME_DISPLAY/A57=EMPLOYEE.EMPINFO.FIRST_NAME | EMPLOYEE.EMPINFO.LAST_NAME;
NAME_CODE/A1=EDIT(LAST_NAME, '9');
NAME_GROUP/A10=IF NAME_CODE LE 'G' THEN 'A-G' ELSE IF NAME_CODE LE 'P'
 THEN 'H-P' ELSE 'Q-Z';
DATE_HIRED/YY=HIRE_DATE;
END
TABLE FILE EMPLOYEE
SUM EMPLOYEE.EMPINFO.CURR_SAL
BY DATE_HIRED
BY LOWEST NAME_DISPLAY AS 'Employee Name'
ON TABLE SET PAGE-NUM NOLEAD
WHERE HIRE_DATE LT '820101';
ON TABLE SET EXPANDBYROW ALL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET DROPBLNKLINE ON
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/javaassist/intl/EN/ENIADefault_combine.sty,$
ENDSTYLE
END
```

上記のリクエストでは、動詞フィールドの HIRE_DATE および CURR_SAL が、一時項目 (DEFINE) の DATE_HIRED でソートされます。下図のように、再定義された日付値に基づいてレポートがソート、集計されます。

	HIRE_DATE	CURR_SAL
⊟ 1980		$11,000.00
ALFRED STEVENS	80/06/02	$11,000.00
⊟ 1981		$58,742.00
BARBARA CROSS	81/11/02	$27,062.00
JOHN MCCOY	81/07/01	$18,480.00
MARY SMITH	81/07/01	$13,200.00
TOTAL		$69,742.00

SET EXPANDBYROW および EXPANDBYROWTREE 使用上の注意

同一列内でのアコーディオンレポートでは、次の機能はサポートされません。

- HFREEZE
- HTML 目次 (TOC) レポート
- Web ビューア
- TABLEF
- OVER
- ROW-TOTAL
- ON field RECAP
- FOR
- IN
- SEQUENCE
- PAGENUM
- SUBHEAD
- BORDER
- MULTILINES
- BY HIERARCHY

- 複合レポート
- 単一 HTML レポートでの複数リクエスト

特定の状況では、レポート出力の中間合計の前にブランク行が自動的に生成されます。これらの自動的なブランク行を削除するには、SET DROPBLNKLINE=ON コマンドを発行します。

列単位のアコーディオンレポートの作成

列単位のアコーディオンレポートでは、レポートの行全体は自動的には表示されません。最上位ソートフィールド列の各データ値の左側にプラス記号 (+) が表示されます。下位のソートフィールドに対応するデータ値の左側にプラス記号 (+) が表示され、実際のデータは、これらを手動で展開しない限り表示されません。最下位ソートフィールドのデータ値を展開することはできません。展開可能ソートフィールドのプラス記号 (+) をクリックするだけで、下位ソートフィールドのデータを表示することができます。最下位から 2 番目のソート項目を指定している見出しのデータ値を展開すると、対応するデータがすべて表示されます。このタイプのアコーディオンレポートを生成するには、SET EXPANDABLE コマンドを使用します。

列単位のアコーディオンレポートのサポート

アコーディオンレポートを指定する場合、次のコマンドは使用できません。

```
BORDER, COLUMN, FOR, IN, OVER, PAGE-NUM, ROW-TOTAL, TOTAL
```

また、ピアグラフ、OLAP、HTML BYTOC、Web ビューア、列の固定、ReportCaster バースト機能も使用できません。

列単位のアコーディオンレポートの作成

列単位のアコーディオンレポートを有効にするには、次のように指定します。

```
ON TABLE SET EXPANDABLE = {ON|OFF}
```

説明

ON

列単位のアコーディオンレポートを有効にします。

OFF

列単位のアコーディオンレポートを無効にします。デフォルト値は OFF です。

列単位のアコーディオンレポートの作成

次の例は、EXPANDABLE コマンドを使用して、列単位のアコーディオンレポートを作成する方法を示しています。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY REGION BY ST BY CITY BY CATEGORY
ON TABLE SET EXPANDABLE ON
END
```

下図は、列単位のアコーディオンレポートを示しています。このレポートでは、デフォルト設定で、最上位のソートフィールドである Region に関連するデータがすべて表示されます。ここで表示されているデータ値は展開された状態ですが、これは、レポートの最上位、第 2 位、第 3 位のソートフィールドのプラス記号 (+) をユーザがクリックした結果です。

Region	State	City	Category	Unit Sales	Dollar Sales
⊞ Midwest					
⊟ Northeast	⊞ CT				
	⊞ MA				
	⊟ NY	⊟ New York	Coffee	116659	1459160
			Food	125473	1555165
			Gifts	70194	887940
⊞ Southeast					
⊟ West	⊟ CA	⊟ Los Angeles	Coffee	109082	1383924
			Food	110417	1381698
			Gifts	78571	1006381
		⊞ San Francisco			
	⊞ WA				

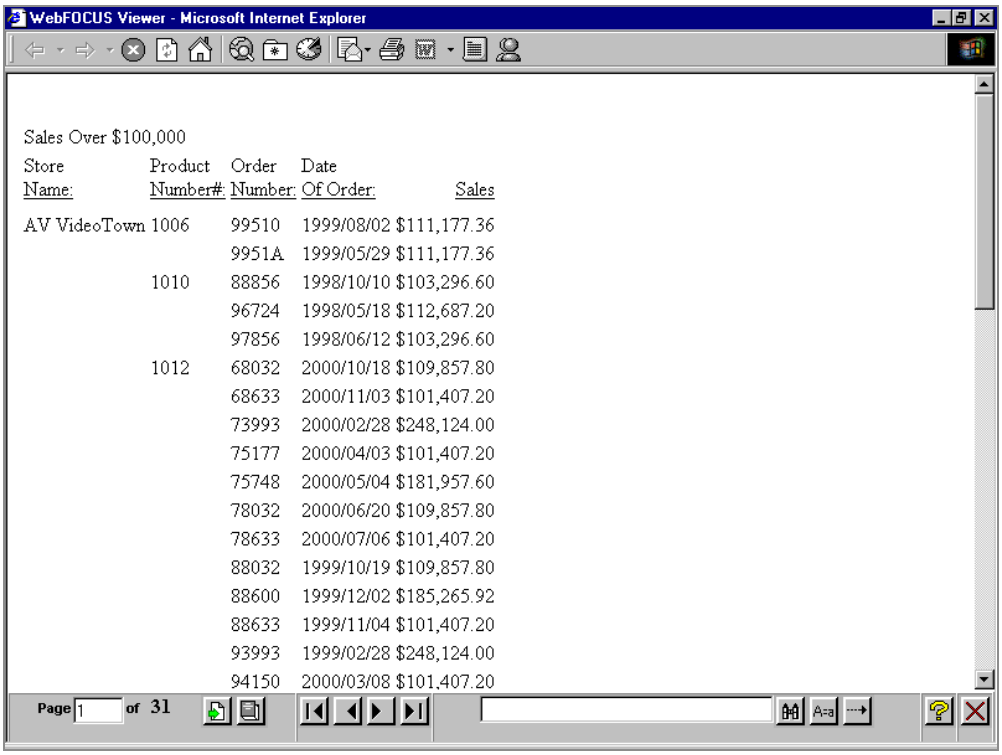
Web ビューアによる複数ページレポートのナビゲーション

通常、Web サーバは、HTML レポート全体をブラウザに返します。ブラウザは、すべてのページを受信してからレポートを表示します。Web ビューアで実行するオンデマンドページ設定により、レポートのすべての代わりに 1 ページをブラウザに返します。残りのページは、ユーザが要求するまで Web サーバに保持されます。この機能により 1 ページ目の表示までの待機時間が短縮され、長いレポートに特に役立ちます。これには、レポートページの転送を高速にするナビゲーション機能も含まれています。

Web ビューアでは、目次 (BYTOC) オプションはサポートされません。これは、Web ビューアは一度に表示するデータを絞り込んでページの一部のみを表示する機能ですが、目次オプションを使用するにはデータのすべてが同一の HTML ページ上に表示されている必要があるためです。Web ビューアは、出力結果を複数のページに分割し、一度に 1 ページのみをブラウザにダウンロードします。また、Web ビューアではアコーディオンレポートもサポートされません。

HFREEZE スタイルシートオプションを使用して、Web ビューアから返される出力結果の各ページにフィールドタイトルを表示することができます。

下図は、Web ビューアに 27 ページのレポートの1 ページ目が表示されています。



Web ビューアが 2 つのフレームに分割されます。

- レポートフレームは、レポートの 1 ページを含む上部の大きい方のフレームです。
- コントロールフレームには、レポートのナビゲーションおよびレポートの文字列の検索に使用するコントロールが含まれます。ナビゲーションコントロールを使用して、次ページ、前ページ、先頭ページ、最終ページ、または指定したページを表示することができます。

  注意：SET コマンドを使用することで、特定のボタンを表示するかどうかを制御することができます。詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

Web ビューアでの HTMLARCHIVE 使用上の注意

WebFOCUS インタラクティブレポート機能が正常に動作するには、WebFOCUS Client に接続して、この機能に必要なコンポーネントにアクセスする必要があります。

WebFOCUS Client へのアクセスが利用できない場合は、HTMLARCHVE を使用して、ユーザ定義イメージが含まれた自己完結型 HTML ページを作成することができます。

ユーザ定義イメージが含まれた HTML ページを生成し、インタラクティブ動作を可能にするには、次のコマンドのいずれかを使用します。

```
SET HTMLEMBEDIMG=ON
SET HTMLARCHIVE=ON
```

次の構文を使用して、これらのファイルへのアクセスが可能なホストマシンを指定した BASEURL を定義します。

```
SET BASEURL=http://{hostname:portnumber}
```

SET BASEURL についての詳細は、ベース URL の指定を参照してください。

Web ビューアでナビゲートするには

Web ビューアのコントロールパネルでは、さまざまな方法でレポートページを表示することができます。

- 前ページまたは次ページを続けて表示するには、[戻る] または [次へ] の矢印をクリックします。
- レポートの先頭ページまたは最終ページを表示するには、[先頭ページ] または [最終ページ] の矢印をクリックします。
- 特定のページを表示する方法は次のとおりです。
  1. [ページ] テキストボックスにページ番号を入力します。
  2. [ページへ移動] ボタンをクリックします。
- レポート全体を単一ドキュメントとしてダウンロードするには、[全ページ] ボタンをクリックします。ビューアコントロールパネルは表示されず、レポート全体が表示されます。

  [BACK SPACE] をクリックすると、レポートの単一ページ表示に戻ることができます。
- テキストを検索するには、テキストボックスにテキストを入力し、[検索] ボタンをクリックします。

  大文字、小文字を区別して検索するには、[A=a] ボタンを切り替えます。

  検索方向を制御するには、[-->/<--] ボタンをクリックします。

Web ビューア検索オプションの使用

ビューアコントロールパネルには、レポート内を検索するためのコントロールがいくつか用意されています。ビューアの検索コントロールを使用して、レポート内の語句や数字グループなどの文字列情報の選択や、一致した個々の文字列の検索が行えます。語句の大文字小文字の正確な一致 (大文字と小文字を区別した検索)、または検索の方向 (レポート上の検索開始地点から前方または後方のいずれか) を指定して、さらに検索方法をカスタマイズすることができます。

検索オプションは次のように動作します。

1. 最初の検索は、ドキュメントの最上部から開始されます。
2. 2 回目以降の検索は、前回の検索で見つかった文字列から開始され、ビューアコントロールパネルで選択した検索方向に進みます。方向コントロールは、検索文字列を指定するための [検索] 入力フィールドの右側にあります。

レポートページの接続

スタイルシートにより、複数ページ HTML レポートに 1 つまたは複数のナビゲーション用ハイパーリンクを挿入することができます。この機能を使用すると、各ページの個別のハイパーリンクを作成せずに、連続するレポートページを簡単にリンクすることができます。

任意のレポートコンポーネントを、ハイパーリンクとして定義することができます。

レポートページのリンク

HTML レポートで使用する構文は次のとおりです。

```
URL=#_destination, $
```

説明

destination

次のいずれかです。

`#_next` - 次のレポートページの先頭へ移動します。

`#_previous` - 前のレポートページの先頭へ移動します。

`#_top` - 使用中のレポートページの先頭へ移動します。

`#_start` - レポートの先頭ページへ移動します。

`#_end` - レポートの最終ページへ移動します。

注意：これらの配信先をセルフサービスアプリケーションで使用する場合は、FOCEXEC の先頭で次のコマンドを発行する必要があります。

```
SET BASEURL=''
```

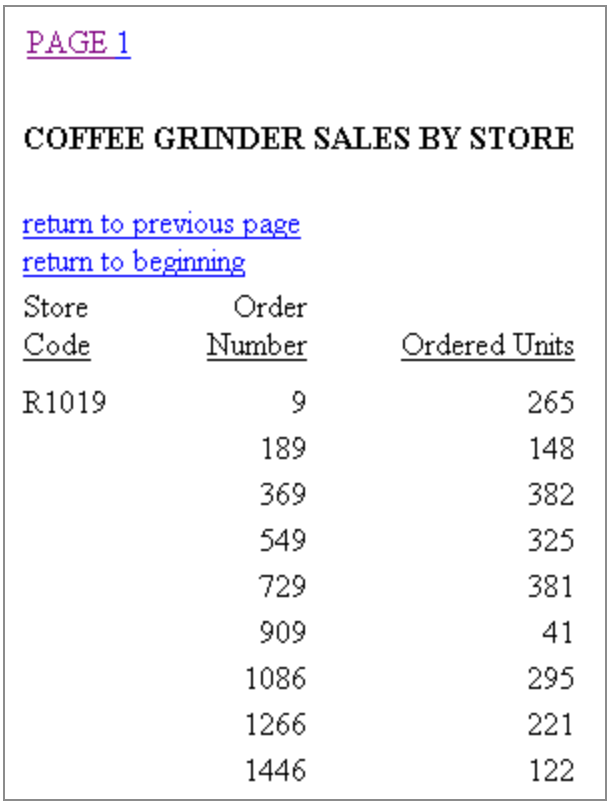
ページ番号および見出し要素によるページの接続

次のリクエストは、レポートのページ番号から次ページへのハイパーリンク、および各レポートページの最上部に表示されるページ見出しのテキストから前ページ、または 1 ページ目へ戻るハイパーリンクを作成します。

```
TABLE FILE GGORDER
ON TABLE SUBHEAD
"COFFEE GRINDER SALES BY STORE"
" "
HEADING
"return to previous page"
"return to beginning"
PRINT QUANTITY AS 'Ordered Units' BY STORE_CODE BY PRODUCT NOPRINT
BY ORDER_NUMBER
WHERE PRODUCT EQ 'Coffee Grinder'
ON STORE_CODE PAGE-BREAK
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=TABHEADING, STYLE=BOLD,$
TYPE=PAGENUM, URL=#_next, $
TYPE=HEADING, LINE=1, URL=#_previous, $
TYPE=HEADING, LINE=2, URL=#_start, $
```

```
ENDSTYLE
END
```

1 ページ目の外観は次のとおりです。



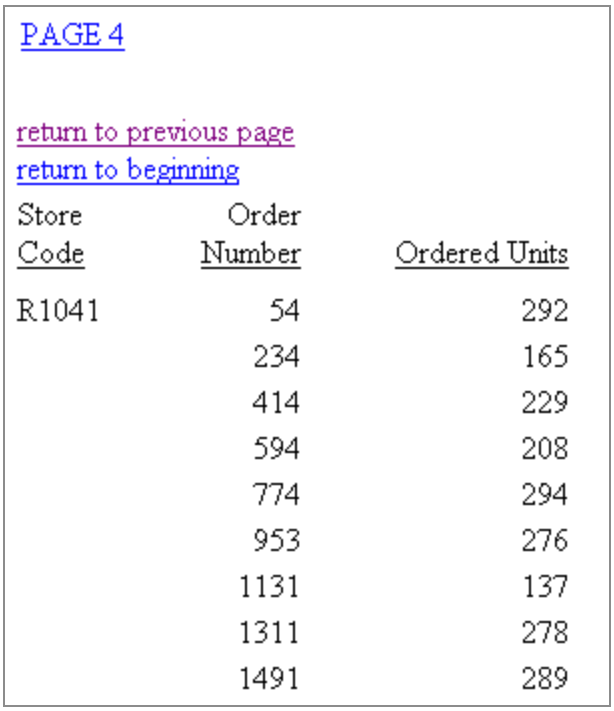
PAGE 1

COFFEE GRINDER SALES BY STORE

return to previous page
return to beginning

Store Code	Order Number	Ordered Units
R1019	9	265
	189	148
	369	382
	549	325
	729	381
	909	41
	1086	295
	1266	221
	1446	122

ページ番号を 3 回クリックして 4 ページ目へ移動します。



PAGE 4

return to previous page
return to beginning

Store Code	Order Number	Ordered Units
R1041	54	292
	234	165
	414	229
	594	208
	774	294
	953	276
	1131	137
	1311	278
	1491	289

3 ページ目に戻るには、「return to previous page」をクリックします。1 ページ目に直接戻るには、「return to beginning」をクリックします。

このプロシジャがセルフサービスアプリケーションに含まれる場合、プロシジャ開始時に次のコマンドを発行する必要があることに注意してください。

```
SET BASEURL = ''
```

複数 HTML ファイルへのレポート分割

バースト機能は、レポートの 1 つ目のソートフィールドの値に基づいて、1 つのレポートを複数の HTML ファイルに分割します。

バーストレポートの概要

バースト機能を使用して 1 つのレポートを複数の HTML ファイルに分割するときは、各ファイルにはソートフィールドの単一の特定値に対して要求したすべての情報が含まれます。レポートの異なるセクションへ直接アクセスすることができるため、次のことが可能になります。

- 大規模なレポートのナビゲーションを容易にする。バーストは、各結果ファイルへの直接リンクを含むインデックスファイルを自動的に作成します。
- 受信者が必要なセクションを容易にナビゲートできるよう、レポートの配信をカスタマイズする。
- 標準見出しおよび脚注コマンドを使用して、インデックスページまたは各レポートファイルにラベルを付ける。

複数 HTML ファイルへのレポートのバースト

```
ON sortfield PCSEND LOCATION dir [AS burstname] FORMAT HTML
```

説明

sortfield

発生するバーストの種類に基づいて、ソートフィールドを指定します。各バーストファイルには、1 つのソートグループのみを対象とするレポート出力が含められます。

レポートをバーストするときは、1 つ目のソートフィールド値のみを基準にすることができます。2 つ目以降のソートフィールドの値に基づいて、レポートをバーストすることはできません。

レポートを最大 10,000 までの個別ファイルにバーストすることができます。そのため、レポートをバーストできるのは、1 つ目のソートフィールドの各値の数が 10,000 を超えない場合に限られます。

PCSEND

バーストを開始します。リクエストで使用できる PCSEND コマンドは 1 つのみです。

dir

HTML インデックスファイルおよびレポートファイルが保存される Web サーバの場所を指定します。Web サーバが HTML ファイルを読み取るディレクトリを指定するための LOCATION パラメータが必要です。デフォルト値は設定されていません。

UNIX、Windows プラットフォームでは、ディレクトリ値で完全修飾されたディレクトリパスを指定する必要があります。

LOCATION パラメータで指定したディレクトリが無効な場合、または書き込み不可の場合、エラーメッセージが返されます。

burstname

バーストした HTML レポートファイルへのハイパーリンクのリストを含む、HTML インデックスファイル名を指定します。バーストした HTML レポートファイルへのハイパーリンクには、0 (ゼロ) から 9999 の番号が付きます。HTML インデックスファイルにファイル名を指定しない場合のデフォルト名は、HOLD です。

バーストした各 HTML レポートファイルは、インデックスファイル名の最初の 4 バイト文字を使用し、0000 から 9999 の間の数字が続きます。

注意：リクエストには、ミッシング値を含む表示フィールドを含めることができます。リクエストには、NOPRINT フィールドも含めることができます。詳細は、ミッシングフィールド値を含むレコード処理を参照してください。

インデックスページおよびバーストレポートでの見出しおよび脚注の規則

- HTML インデックスページに見出しまたは脚注を追加するには、次のコマンドを使用し

ます。

```
ON TABLE SUBHEAD
```

```
ON TABLE SUBFOOT
```

これらの見出しおよび脚注には、埋め込みフィールドを含めることができます。HTML レポート出力の各ページには、これらの見出しおよび脚注は表示されません。デフォルトのスタイル設定が、HTML インデックスページに適用されます。

- 標準の見出しまたは脚注を、すべての HTML レポート出力ページに含めるには、次のコマンドを使用します。

```
HEADING
```

```
FOOTING
```

これらの見出しおよび脚注には、埋め込みフィールドを含めることができます。デフォルトのスタイル設定が、HTML レポート出力ページに適用されます。

- ソートフィールドが変更されるたびに見出しまたは脚注を更新するには、次のコマンドを使用します。

```
ON sortfield SUBHEAD
```

```
ON sortfield SUBFOOT
```

これらの中間見出しまたは中間脚注には、PCSEND コマンドで使用するソートフィールドなどの埋め込みフィールドを含めることができます。デフォルトのスタイル設定が、HTML レポート出力ページに適用されます。

レポートのバースト

次のレポートプロシジャは、GGSALES データソースの異なる REGION 値に対して、それぞれ別の HTML レポート出力を作成します。レポート出力ファイルには test0000.htm、test0001.htm、test0002.htm というように名前が付けられます。HTML インデックスページには「test.html」という名前が付けられ、各 REGION データ値に対するハイパーリンクが含められます。WebFOCUS をインストールした場所により、選択するディレクトリが異なりま

す。次の例では、インデックスページが e:¥ibi¥WebFOCUS93¥temp のディレクトリに保存されます。

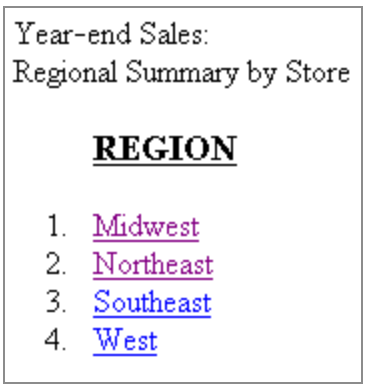
```
TABLE FILE GGSALES
HEADING
"Regional Report"
SUM UNITS AND DOLLARS
BY REGION BY STCD BY CATEGORY
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Year-end Sales:"
"Regional Summary by Store"
ON REGION PCSEND LOCATION E:¥IBI¥WebFOCUS93¥temp AS TEST FORMAT HTML
END
```

リクエストの実行後にはレポート出力は返されませんが、リクエストが正常に終了すると、次のメッセージが表示されます。

```
バーストファイルの作成に成功しました。
```

主ソートフィールドの REGION の各値に対して別の HTML ファイルが作成され、リクエストで指定した場所に保存されます。

このプロシジャにより、次のような HTML インデックスページが作成されます。



Midwest ハイパーリンクを選択すると、次のような HTML レポートが表示されます。

```
Regional Report
```

Region	Store ID	Category	Unit Sales	Dollar Sales
Midwest	R1019	Coffee	113253	1393610
		Food	107615	1351523
		Gifts	78869	969845
	R1020	Coffee	109581	1398779
		Food	118068	1522847
		Gifts	79932	1002775
	R1250	Coffee	109943	1386124
		Food	115731	1463901
		Gifts	72053	911261

ON TABLE SUBHEAD コマンドで指定した見出しが、HTML インデックスページに表示されます。詳細は、バーストレポートの概要を参照してください。

ミッシングフィールド値を含むレコード処理

ミッシング値は、関連しないまたはデータソースに存在しないためにレポートから欠落しているデータと定義されます。平均、集計計算を含むレポート出力、または親セグメントインスタンスの表示はミッシング値の影響を受けることがあります。データがレポートおよび計算から欠落する理由は次のとおりです。

- データがレポートの特定行および列に関連していない。詳細は、関連値を持たないレポートデータを参照してください。
- セグメントインスタンスのフィールドにデータ値がない。詳細は、ミッシングフィールド値を参照してください。
- 親セグメントインスタンスに子インスタンスがない (ミッシングセグメントインスタンス)。詳細は、ミッシングセグメントインスタンスの処理を参照してください。

関連値を持たないレポートデータ

レポートで関連値を持たない行または列は、欠落する場合があります。欠落している (該当しない) 値は、SET NODATA のデフォルトであるピリオド (.) で示されます。

ヒント：SET NODATA コマンドを発行して任意の NODATA 値を指定することができます。詳細は、NODATA 文字列の設定を参照してください。

関連値を持たないレポートデータ

次のリクエストは、レポートのデフォルトのミッシング値を表示します。

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL
BY LAST_NAME
BY FIRST_NAME
```

```
ACROSS DEPARTMENT
END
```

出力結果は次のとおりです。

LAST_NAME	FIRST_NAME	DEPARTMENT MIS	PRODUCTION
BANNING	JOHN	.	$29,700.00
BLACKWOOD	ROSEMARIE	$21,780.00	.
CROSS	BARBARA	$27,062.00	.
GREENSPAN	MARY	$9,000.00	.
IRVING	JOAN	.	$26,862.00
JONES	DIANE	$18,480.00	.
MCCOY	JOHN	$18,480.00	.
MCKNIGHT	ROGER	.	$16,100.00
ROMANS	ANTHONY	.	$21,120.00
SMITH	MARY	$13,200.00	.
		.	$9,050.00
STEVENS	ALFRED	.	$11,000.00

生産部門に就業する従業員給与は、PRODUCTION フィールドに、MIS 部門に就業する従業員給与は、MIS フィールドにそれぞれ表示されます。対応する PRODUCTION または MIS フィールドの値は、給与が個人の就業する部門の下にのみ表示されるため、それぞれ欠落しています。

ミッシングフィールド値

セグメントインスタンスのミッシング値は、インスタンスは存在してもそのフィールドのいくつかの値が欠落しているときに発生します。

フィールドの値が欠落している場合、数値フィールドには 0 (ゼロ) の値、文字フィールドにはブランクの値がそれぞれ割り当てられます。これらのデフォルト値はレポートに表示され、SUM および COUNT 表示コマンド、DEFINE コマンド、および MAX.、AVE. などの演算接頭語により実行されるすべての計算で使用されます。

(エラーを含む結果になることのある) これらのデフォルト値が計算で使用されることを防止するために、マスターファイルの実フィールドまたは一時項目 (DEFINE) に MISSING 属性を追加することができます。MISSING 属性を ON に設定すると、ブランクまたは 0 (ゼロ) と区別するためにミッシング値に特殊内部コードの印が付き、ミッシング値は計算で無視されます。この内部コードは、レポートには SET NODATA の値で表示されます。デフォルト値はピリオド (.) です。詳細は、NODATA 文字列の設定を参照してください。

次の例で示すように、セグメントインスタンスのフィールドのミッシング値は、データ値が不明な場合に発生します。EMPLOYEE データソースに記録される従業員の昇給日は決められているが、昇給額は決定されていない場合について考察します。会社は各従業員用の日付を

給与額なしでデータソースに入力します。給与額は後で入力します。各日付は給与履歴セグメントの個別インスタンスですが、各日付インスタンスの新しい給与が欠落しています。さらに、リクエストにより SALARY フィールド (SUM AVE.SALARY) の平均が計算されることを想定します。平均結果の精度は、SALARY フィールドのミッシング値が 0 (ゼロ) として処理 (MISSING=OFF) されるか、内部コードとして処理 (MISSING=ON) されるかにより異なります。

注意：数値集計値 (例、中間合計) の計算に使用されるフィールド値のすべてがミッシング値の場合、集計値には 0 (ゼロ) ではなく、ミッシングデータ値が割り当てられます。これには、中間脚注で ST. および CT. 演算子によって生成される集計値も含まれます。

ミッシング値の個数計算

CURR_SAL フィールドが 12 件のセグメントインスタンスに表示されることを想定します。このうち 3 件のインスタンスでは、このフィールドに値が提供されていません。表示コマンドは次のとおりです。

```
COUNT CURR_SAL
```

上記の表示コマンドで、CURR_SAL フィールドは 12 件として計算されます。これは、デフォルト設定で MISSING 属性が OFF に設定され、ミッシング値が合計数に含まれるためです。ミッシング値を合計数から除外するには、MISSING 属性を ON に設定します。

ミッシング値を使用した平均計算

次のフィールドデータレコードを想定します。

```
.
.
1
3
```

最初の 2 つのレコードの数値が欠落しています (ピリオドにより表示)。最後の 2 つのレコードに 1 および 3 の値があります。MISSING 属性なし (MISSING OFF) にこれらのフィールドの平均を計算すると、値の欠落している 2 つのレコードに 0 (ゼロ) の値が入力されます。そのため、レコードの平均は (0+0+1+3)/4 = 1 です。MISSING ON 属性を使用すると、平均は (1+3)/2 = 2 と計算され、2 つのミッシング値は無視されます。

マスターファイルの MISSING 属性

アプリケーションによっては、デフォルト値 (ブランクおよび 0 (ゼロ)) は欠落している情報ではなく、有効なデータを表す場合があります。ただし、それ以外の場合は、マスターファイルのフィールドフォーマットの後に、ミッシング値を含むフィールドの MISSING 属性を含めることができます。MISSING 属性はデータソースの実フィールド、またはマスターファイルで定義する一時項目 (DEFINE) に使用することができます。

たとえば、RETURNS フィールドに対して MISSING 属性を指定するフィールド宣言は次のとおりです。

```
FIELDNAME=RETURNS, ALIAS=RTN, FORMAT=I4, MISSING=ON,$
```

「PROFIT」と呼ばれる一時項目 (DEFINE) に対して MISSING 属性を指定する宣言は次のとおりです。

```
DEFINE PROFIT/D7 MISSING ON NEEDS SOME DATA = RETAIL_COST - DEALER_COST;$
```

一時項目 (DEFINE) に対するミッシング値の処理を確実にするために、DEFINE コマンドで MISSING 値を ON に設定し、計算の適用対象としてミッシング値のいくつかまたはすべてを指定することができます。SOME および ALL 句についての詳細は、ミッシングフィールド値を参照してください。

フィールド宣言で MISSING 属性を ON に設定すると、データを含まないフィールドにはブランクまたは 0 (ゼロ) の代わりに、特殊内部コードの印が付きます。レポートの生成時、SUM コマンドと COUNT コマンド、およびすべての演算接頭語 (たとえば AVE.、MAX.、MIN.) は、計算からミッシング値を除外します。

注意

- MISSING フィールド属性はマスターファイルにいつでも追加することができます。ただし、MISSING 属性の追加後にデータソースに入力されたデータのみが影響を受けます。
- レコードの識別にはキーフィールドが必須であるため、キーフィールドをミッシングとして指定することはできません。

MISSING 属性によるミッシング値の処理

次の例は、MISSING ON を含むフィールドと含まないフィールドの相違を示します。MISSING 属性を含まない一時項目 (DEFINE) の X_RETURNS が、実フィールドのマスター

ファイルで宣言した MISSING 属性を含む RETURNS と同一になるよう設定されています。MISSING 属性 (RETURNS) を含むフィールドに値が欠落している場合、値の欠落しているデータソースフィールドは計算目的のために 0 (ゼロ) またはブランクと評価されるため、X_RETURNS の関連値は 0 (ゼロ) になります。詳細は、ミッシングフィールド値を参照してください。

次のリクエストは一時項目 (DEFINE) を定義します。

```
DEFINE FILE SALES
X_RETURNS/I4 = RETURNS;
END
```

発行するリクエストは次のとおりです。

```
TABLE FILE SALES
SUM CNT.X_RETURNS CNT.RETURNS AVE.X_RETURNS AVE.RETURNS
END
```

RETURNS の値が欠落しており、X_RETURNS 値が 0 (ゼロ) になる場合を除いて、X_RETURNS フィールドの値は RETURNS と同一になります。

出力結果は次のとおりです。

X_RETURNS	RETURNS	AVE	AVE
COUNT	COUNT	X_RETURNS	RETURNS
22	20	2	3

RETURNS のミッシング値は計算の一部でないため、RETURNS フィールドの数は X_RETURNS の数より少なく、RETURNS の平均は X_RETURNS の平均より大きくなります。

DEFINE または COMPUTE コマンドの MISSING 属性

DEFINE または COMPUTE コマンドで MISSING 属性を ON に設定すると、ミッシング値を含む一時項目が、レポートで正確に解析、表示されるようになります。

一時項目の値の抽出に使用する式には、ミッシング値を含む実フィールドを含めることができます。ただし、一時項目の値の抽出のために使用する場合は、マスターファイルのフィールドの MISSING 属性が ON に設定されていても、値の欠落しているデータソースフィールドは計算の目的上、0 (ゼロ) またはブランクとして評価されます。

一時項目のミッシング値を適切に処理するため、DEFINE または COMPUTE コマンドで一時項目の MISSING 属性を ON に設定し、ミッシング値の適用対象として値のいくつかまたはすべてを指定することができます。詳細は、ミッシングフィールド値を参照してください。

DEFINE または COMPUTE コマンドによるミッシング値の指定

```
field[/format] MISSING {ON|OFF} [NEEDS] {SOME|ALL} [DATA] = expression;
```

説明

field

DEFINE コマンドで作成した一時項目名です。

/format

一時項目のフォーマットです。デフォルト値は D12.2 です。

MISSING

ON - 一時項目の値をミッシング (特殊内部コードにより意図的に入力した 0 (ゼロ) またはブランクと区別される) と見なし、レポートに NODATA 文字で表示することを可能にします。

OFF - 数値フィールドのミッシング値を 0 (ゼロ)、文字フィールドのミッシング値をブランクとして処理します。これがデフォルト値です。

NEEDS

オプションです。コマンドの意味を明確にします。

SOME

式のフィールドの少なくとも 1 つに値がある場合、一時項目には値が存在することを示します。計算でフィールドのミッシング値は 0 (ゼロ) またはブランクとして評価されます。式のすべてのフィールドの値が欠落している場合、一時項目の値は欠落します。デフォルト値は SOME です。

ALL

式のすべてのフィールドに値がある場合、一時項目に値が存在することを示します。式のフィールド少なくとも 1 つにミッシング値がある場合、一時項目にもミッシング値があります。

DATA

オプションです。コマンドの意味を明確にします。

expression

一時項目が値を抽出する有効な式です。

注意：一時項目 (DEFINE) および一時項目 (COMPUTE) で MISSING ON を設定するには、SET MISS_ON のデフォルト値を設定するには、SET MISS_ON コマンドを使用する ともできます。

MISSING OFF による一時項目 (DEFINE) のミッシング値の処理

次のリクエストは、NO_SALE フィールドを定義するための 2 つのフィールド RETURNS および DAMAGED の使用を示します。RETURNS と DAMAGED フィールドの両方に SALES マスターファイルで ON に設定した MISSING 属性がありますが、これらのフィールドの 1 つの値が欠落している場合、そのフィールドは 0 (ゼロ) と評価されます。

```
DEFINE FILE SALES
NO_SALE/I4 = RETURNS + DAMAGED;
END
TABLE FILE SALES
PRINT RETURNS AND DAMAGED AND NO_SALE
BY CITY BY DATE BY PROD_CODE
END
```

出力結果は次のとおりです。

```
CITY             DATE   PROD_CODE  RETURNS  DAMAGED  NO_SALE
----             ----   ---------  -------  -------  -------
NEW YORK         10/17  B10              2        3        5
                        B17              2        1        3
                        B20              0        1        1
                        C13              .        6        6
```

```
                        C14              4        .        4
                        C17              0        0        0
                        D12              3        2        5
                        E1               4        7       11
                        E2               .        .        0
                        E3               4        2        6
NEWARK          10/18   B10              1        1        2
                10/19   B12              1        0        1
STAMFORD        12/12   B10             10        6       16
                        B12              3        3        6
                        B17              2        1        3
                        C13              3        0        3
                        C7               5        4        9
                        D12              0        0        0
                        E2               9        4       13
                        E3               8        9       17
UNIONDALE       10/18   B20              1        1        2
                        C7               0        0        0
```

MISSING ON 属性がマスターファイルで設定されているため、New York 地区の商品 C13、C14、および E2 のすべてに RETURNS と DAMAGED のいずれかのミッシング値が表示されています。ただし、MISSING ON が一時項目 (DEFINE) に対して設定されていないため、NO_SALE の値を決定する計算によりこれらのミッシング値は 0 (ゼロ) と見なされます。

SOME および ALL による一時項目 (DEFINE) のミッシング値の処理

次のリクエストは、DEFINE コマンドで MISSING 属性を使用し、DEFINE コマンドで参照するフィールド値のいくつかまたはすべてが欠落している場合、一時項目 (DEFINE) の値も欠落していると見なすよう指定する方法を示します。

RETURNS と DAMAGED フィールドのいずれかに値が含まれると SOMEDATA フィールドに値が含まれます。それ以外は SOMEDATA の値は欠落します。RETURNS と DAMAGED フィールドの両方に値が含まれる場合にのみ、ALLDATA フィールドに値が含まれます。それ以外は ALLDATA の値は欠落します。

```
DEFINE FILE SALES
SOMEDATA/I5 MISSING ON NEEDS SOME=RETURNS + DAMAGED;
ALLDATA/I5 MISSING ON NEEDS ALL=RETURNS + DAMAGED;
END
```

```
TABLE FILE SALES
PRINT RETURNS AND DAMAGED SOMEDATA ALLDATA
BY CITY BY DATE BY PROD_CODE
END
```

出力結果は次のとおりです。

CITY	DATE	PROD_CODE	RETURNS	DAMAGED	SOMEDATA	ALLDATA
NEW YORK	10/17	B10	2	3	5	5
		B17	2	1	3	3
		B20	0	1	1	1
		C13	.	6	6	.
		C14	4	.	4	.
		C17	0	0	0	0
		D12	3	2	5	5
		E1	4	7	11	11
		E2	.	.	.	.
		E3	4	2	6	6
NEWARK	10/18	B10	1	1	2	2
	10/19	B12	1	0	1	1
STAMFORD	12/12	B10	10	6	16	16
		B12	3	3	6	6
		B17	2	1	3	3
		C13	3	0	3	3
		C7	5	4	9	9
		D12	0	0	0	0
		E2	9	4	13	13
		E3	8	9	17	17
UNIONDALE	10/18	B20	1	1	2	2
		C7	0	0	0	0

DEFINE および COMPUTE の MISSING ON 動作の設定

一時項目 (DEFINE) または一時項目 (COMPUTE) にミッシング値が含まれるよう設定されている場合、DEFINE または COMPUTE の作成式に使用されるフィールド値のすべて (ALL) がミッシング値の場合に結果フィールドをミッシングとして処理するか、フィールド値の一部 (SOME) がミッシング値の場合に結果フィールドをミッシングとして処理するかを制御することができます。MISSING ON が設定された DEFINE または COMPUTE で ALL または SOME のどちらも指定しない場合、デフォルト値は SOME になります。

SET パラメータの MISS_ON を使用して、MISSING ON が設定された DEFINE または COMPUTE で ALL または SOME のいずれも指定されていない場合に、どちらを使用するかを指定することができます。

```
SET MISS_ON = {SOME|ALL}
```

説明

SOME

式のフィールドの少なくとも 1 つに値がある場合、一時項目には値が存在することを示します。計算でフィールドのミッシング値は 0 (ゼロ) またはブランクとして評価されます。式のすべてのフィールドの値が欠落している場合、一時項目の値は欠落します。デフォルト値は SOME です。

ALL

式のすべてのフィールドに値がある場合、一時項目に値が存在することを示します。式で指定されたフィールドの少なくとも 1 つにミッシング値が存在する場合にも、一時項目にミッシング値があると解釈します。

一時項目 (DEFINE) および一時項目 (COMPUTE) での MISSING ON のデフォルト値の設定

次のリクエストは、MISSING ON が設定された 3 つの一時項目 (DEFINE) を作成します。フィールド AAA には、すべてミッシング値が格納されています。フィールド BBB は、カテゴリが Gift の場合にのみミッシング、それ以外では値 100 が格納されます。フィールド CCC は、AAA と BBB の合計です。

```
SET MISS_ON = SOME
DEFINE FILE GGSALES
AAA/D20 MISSING ON = MISSING;
BBB/D20 MISSING ON = IF CATEGORY EQ 'Gifts' THEN MISSING ELSE 100;
CCC/D20 MISSING ON = AAA + BBB;
END
TABLE FILE GGSALES
SUM
AAA
BBB
CCC
BY CATEGORY
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
```

```
GRID=OFF,$
END
```

このリクエストを SET MISS_ON=SOME (デフォルト設定) で実行すると、AAA と BBB の両方がミッシングでない場合に限り、CCC に値が表示されることが分かります。

Category	AAA	BBB	CCC
Coffee	.	144,000	144,000
Food	.	144,000	144,000
Gifts	.	.	.

SET MISS_ON を ALL に変更すると、次の出力が生成されます。計算に使用されるフィールドのいずれかが常にミッシングであるため、CCC にはミッシング値が割り当てられます。

Category	AAA	BBB	CCC
Coffee	.	144,000	.
Food	.	144,000	.
Gifts	.	.	.

IF-THEN-ELSE 式でのミッシング値のテスト

IF-THEN-ELSE 式の処理効率を最適化するため、式全体を解析せずに結果 (true または false) が特定できる場合は、式全体の解析は実行されません。たとえば、次の IF-THEN-ELSE 式について考察します。

```
COMPUTE CCC/D9.1 MISSING ON = IF AAA EQ 0 OR BBB EQ 0 THEN 0 ELSE AAA / BBB *
100;
```

注意

- AAA は MISSING に設定 (MISSING ON) され、ミッシングが許可されたフィールドです。
- BBB は MISSING ON に設定されていますが、ミッシング時の値が 100 にされているため、MISSING ではありません。

式を評価するために、MISSING フィールドを 0 (ゼロ) として処理します。

CCC は、IF-THEN-ELSE 式による MISSING ON の計算を含む数値フィールドです。IF 句には、OR 演算子で接続された 2 つのテストが含まれています。これら 2 つのテストのいずれかが TRUE の場合、THEN 句の値が MISSING の最終確認に渡されます。いずれも TRUE でない場合、ELSE 句の値が MISSING 確認に渡されます。

この場合、比較で使用される際に MISSING が 0 (ゼロ) として処理されるため、AAA EQ 0 が成り立ちます。従って、THEN の値 (0 (ゼロ)) が受容されます。CCC は MISSING ON (デフォルト値は「NEEDS SOME」) に設定されているため、IF 句で評価されるデータ値の最低 1 つはミッシング以外である必要があります。

TRUE 条件に該当するフィールドは、AAA のみです。BBB はテストの対象外となるため、ミッシング値確認の対象フィールドには含められません。AAA が MISSING であるため、CCC (NEEDS SOME だが該当する値が存在しない) を MISSING に設定する必要があります。この処理は、以前のバージョンから変更され、処理効率が向上する一方で、結果が異なる場合があります。

値のテストと値の有無の確認テスト (MISSING) は同一ではないため、常に適切なテストを使用することが重要です。MISSING のテストでは、EQ 0 と比較して、より直接的で、以前のバージョンと動作が異ならないため、IS MISSING の使用が推奨されます。

次のリクエストについて考察します。CCC は IF-THEN-ELSE で EQ 0、DDD は IS MISSING をそれぞれ使用します。

```
DEFINE FILE GGSALES
AAA/D20 MISSING ON = MISSING;
BBB/D20 MISSING ON = 100;
END
TABLE FILE GGSALES
SUM
  AAA
  BBB
  COMPUTE CCC/D9.1 MISSING ON = IF AAA EQ 0 OR BBB EQ 0 THEN 0 ELSE AAA / BBB
* 100;
  COMPUTE DDD/D9.1 MISSING ON = IF AAA IS MISSING OR BBB IS MISSING THEN 0
ELSE AAA / BBB * 100;
BY  CATEGORY
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
END
```

出力結果は次のとおりです。CCC はミッシング値に対する評価を行う一方、DDD は 0 (ゼロ) に対する評価を行います。これは、AAA と BBB は、いずれも TRUE 条件を満たすかどうかが評価され、BBB はミッシングではなく、DDD の NEEDS SOME VALUES 定義を満たすためためです。

Category	AAA	BBB	CCC	DDD
Coffee	.	144,000	.	.0
Food	.	144,000	.	.0
Gifts	.	143,700	.	.0

IF-THEN-ELSE 式での SET MISSINGTEST の使用

以前のバージョンのデフォルト設定では、IF-THEN-ELSE 式を使用して結果を計算する際に、その式の結果が 0 (ゼロ) (数値式の場合) またはブランク (文字式の場合) と評価された場合、その式の左側に MISSING ON が存在するかどうかが確認されていました。MISSING ON が存在し、一部の値がミッシング値であれば結果フィールドがミッシングとして解釈される場合 (NEEDS SOME)、IF 式の結果は true または false ではなく、MISSING でした。返される結果も MISSING で、THEN 式または ELSE 式を評価した結果ではありませんでした。SET MISSINGTEST = NEW コマンドを使用すると、IF 式のミッシングテストが除外されるため、THEN 式または ELSE 式のいずれかが評価され、式の結果として返されます。これがデフォルトの動作です。

構文は次のとおりです。

```
SET MISSINGTEST = {NEW|OLD|SPECIAL}
```

説明

NEW

IF 式をミッシング値の評価から除外します。これにより、IF 式の結果が MISSING ではなく、true または false になります。評価が true の場合、THEN 式を使用して結果が計算されます。結果が false の場合、ESLE 式を使用して結果が計算されます。これがデフォルト値です。

OLD

IF 式をミッシング値の評価に含めます。IF 式の評価が MISSING であり、ミッシングフィールドにいずれかのミッシング値が必要な場合に限り、結果も MISSING になります。

SPECIAL

パラメータを RStat に渡すために必要です。

IF-THEN-ELSE 式での SET MISSINGTEST の使用

次のリクエストは、国名「Austria」のミッシング値を格納する MISS_FIELD という名前のフィールドを定義します。TABLE リクエストには、IF-THEN-ELSE 式で、このフィールドをテストする CALC1、CALC2 という 2 つの一時項目 (COMPUTE) が含まれています。これらのフィールドには、いずれも MISSING ON が含まれており、ミッシングと評価されるには、いくつかのミッシング値のみが必要です。

```
SET MISSINGTEST = OLD
DEFINE FILE WF_RETAIL_LITE
MISS_FIELD/A10 MISSING ON = IF COUNTRY_NAME NE 'Austria' THEN 'DATAEXISTS'
ELSE MISSING;
END

TABLE FILE WF_RETAIL_LITE
SUM COGS_US MISS_FIELD
COMPUTE CALC1/A7 MISSING ON = IF ((MISS_FIELD EQ '') OR (MISS_FIELD EQ
MISSING)) THEN 'THEN' ELSE 'ELSE';
COMPUTE CALC2/A7 MISSING ON = IF ((MISS_FIELD EQ MISSING) OR (MISS_FIELD EQ
'')) THEN 'THEN' ELSE 'ELSE';
BY COUNTRY_NAME
WHERE BUSINESS_REGION EQ 'EMEA'
WHERE COUNTRY_NAME LT 'E'
ON TABLE SET NODATA 'MISSING'
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

このリクエストを MISSINGTEST=OLD で実行すると、下図のような出力が生成されます。

Customer Country	Cost of Goods	MISS_FIELD	CALC1	CALC2
Austria	$426,570.00	MISSING	MISSING	THEN
Belgium	$3,800.00	DATAEXISTS	ELSE	ELSE
China	$563.00	DATAEXISTS	ELSE	ELSE
Denmark	$1,346.00	DATAEXISTS	ELSE	ELSE

Austria の [MISS_FIELD] に [MISSING] と表示されています。

- CALC1 では、「MISS_FIELD EQ ' '」は true と評価されています。この式の結果が true と決定される場合、評価が終了するため、MISS_FIELD IS MISSING はまったく評価されていません。ただし、この式はフィールドをブランクと比較したため、結果フィールドがミッシング値をサポートするかどうかが確認されています。ミッシングがサポートされるため、最終結果は MISSING です。
- CALC2 では、MISS_FIELD EQ MISSING は true です。この式の結果が true と決定される場合、評価が終了するため、「MISS_FIELD EQ ' '」 はまったく評価されていません。ミッシングの確認が必要ないことから、IF 式の結果は TRUE となり、THEN 式が評価されて結果として返されます。

SET コマンドを SET MISSINGTEST=NEW に変更し、リクエストを再実行すると、このリクエストは下図のような出力を生成します。CALC1 および CALC2 の IF 式は、いずれも true と評価されます。これは、どちらの式も結果フィールドがミッシングをサポートするかどうかを確認せず、THEN 式が評価されて両者の結果として返されるためです。

Customer Country	Cost of Goods	MISS_FIELD	CALC1	CALC2
Austria	$426,570.00	MISSING	THEN	THEN
Belgium	$3,800.00	DATAEXISTS	ELSE	ELSE
China	$563.00	DATAEXISTS	ELSE	ELSE
Denmark	$1,346.00	DATAEXISTS	ELSE	ELSE

ミッシングフィールド値を含むセグメントのテスト

WHERE 条件を指定して、ミッシングフィールド値を含むセグメントインスタンスを指定することができます。

これらのテストを使用して、ミッシングインスタンスを指定することはできません。詳細は、ミッシングセグメントインスタンスの処理を参照してください。

ミッシングフィールド値を含むセグメントのテスト

ミッシングフィールド値を含むセグメントのテストを実行するには、次の構文を使用します。

```
WHERE field {IS|EQ} MISSING
```

フィールド値の存在をテストするには、次の構文を使用します。

```
WHERE field {NE|IS-NOT} MISSING
```

0 (ゼロ) の数値フィールド、またはブランクの文字フィールドをテストする WHERE 条件によりミッシング値のあるフィールドのインスタンスも検索されます。

ミッシングフィールド値のテスト

次のリクエストは、MISSING を使用して、顧客から返却されたパッケージ数が欠落している食料品項目をコード順に表示します。

```
TABLE FILE SALES
PRINT RETURNS
BY CITY BY DATE BY PROD_CODE
WHERE RETURNS IS MISSING
END
```

出力結果は次のとおりです。

```
CITY             DATE   PROD_CODE  RETURNS
----             ----   ---------  -------
NEW YORK         10/17  C13              .
                        E2               .
```

既存のフィールド値のテスト

次のリクエストは、MISSING を使用して、顧客から返却されたパッケージ数が欠落していない食料品項目をコード順に表示します。

```
TABLE FILE SALES
PRINT RETURNS
BY CITY BY DATE BY PROD_CODE
WHERE RETURNS IS-NOT MISSING
END
```

出力結果は次のとおりです。

CITY	DATE	PROD_CODE	RETURNS
NEW YORK	10/17	B10	2
		B17	2
		B20	0
		C14	4
		C17	0
		D12	3
		E1	4
		E3	4
NEWARK	10/18	B10	1
	10/19	B12	1
STAMFORD	12/12	B10	10
		B12	3
		B17	2
		C13	3
		C7	5
		D12	0
		E2	9
		E3	8
UNIONDALE	10/18	B20	1
		C7	0

ブランクまたは 0 (ゼロ) のテスト

次のリクエストは、返却パッケージ数が 0 (ゼロ) または返却パッケージ数が記録されていない食料品項目のいずれかを表示します。

```
TABLE FILE SALES
PRINT RETURNS
BY CITY BY DATE BY PROD_CODE
WHERE RETURNS EQ 0
END
```

出力結果は次のとおりです。

CITY	DATE	PROD_CODE	RETURNS
NEW YORK	10/17	B20	0
		C13	.
		C17	0
		E2	.
STAMFORD	12/12	D12	0
UNIONDALE	10/18	C7	0

ミッシング値のテストからの除外

顧客からの返却パッケージ数が 0 (ゼロ) の項目のみを表示するためには、2 つの WHERE 条件が必要です。次に示すように、1 つは返却数を 0 (ゼロ) に制限し、もう 1 つはミッシング値を除外します。

```
TABLE FILE SALES
PRINT RETURNS
BY CITY BY DATE BY PROD_CODE
WHERE RETURNS EQ 0
WHERE RETURNS IS-NOT MISSING
END
```

出力結果は次のとおりです。

CITY	DATE	PROD_CODE	RETURNS
NEW YORK	10/17	B20	0
		C17	0
STAMFORD	12/12	D12	0
UNIONDALE	10/18	C7	0

ミッシングデータ値の出力ファイルへの保存

フィールドでミッシング値およびデフォルト値 (ブランクおよび 0 (ゼロ)) を区別する機能は、出力ファイルに継承することができます。検索、処理された情報が NODATA 列としてレポートに表示される場合、デフォルト設定では、その NODATA 列を出力ファイルに保存することができます。SET HNODATA コマンドを使用すると、文字出力ファイルで使用される NODATA 値を変更することができます。SET HOLDMISS コマンドを使用して、出力ファイルの NODATA の代わりにミッシング値を保存することもできます。詳細は、レポート出力の保存と再利用を参照してください。

抽出ファイルでのミッシング値の区別

```
ON TABLE {HOLD|SAVE|SAVB} MISSING {ON|OFF}
```

説明

HOLD

後に続くレポートで使用する抽出ファイルを作成します。MISSING のデフォルト値は ON です。

SAVE

他のプログラムで使用するテキスト抽出ファイルを作成します。MISSING のデフォルト値は OFF です。

SAVB

他のプログラムで使用するバイナリ抽出ファイルを作成します。MISSING のデフォルト値は OFF です。

HOLD ファイルは、任意の順序で指定した MISSING と FORMAT ALPHA オプションの両方で作成することができます。以下はその例です。

```
ON TABLE HOLD FORMAT ALPHA MISSING OFF
ON TABLE HOLD MISSING OFF FORMAT ALPHA
```

MISSING 値の抽出ファイルへの統合

次のリクエストでは、HOLD 句で MISSING ON を指定しています。

```
TABLE FILE SALES
SUM RETURNS AND HOLD FORMAT ALPHA MISSING ON
BY CITY BY DATE BY PROD_CODE
END
```

RETURNS フィールドの MISSING=ON 属性は、HOLD マスターファイルに継承されます。また、ミッシングデータ記号は、ミッシングフィールド値の HOLD ファイルに継承されます。

```
FILENAME=HOLD    , SUFFIX=FIX     , $
  SEGMENT=HOLD, SEGTYPE=S3, $
    FIELDNAME=CITY, ALIAS=E01, USAGE=A15, ACTUAL=A15, $
    FIELDNAME=DATE, ALIAS=E02, USAGE=A4MD, ACTUAL=A04, $
    FIELDNAME=PROD_CODE, ALIAS=E03, USAGE=A3, ACTUAL=A03, $
    FIELDNAME=RETURNS, ALIAS=E04, USAGE=I3, ACTUAL=A03,
      MISSING=ON, $
```

HOLD 句で MISSING OFF を指定した場合、MISSING=ON 属性は継承されず、ミッシングデータ記号はデフォルト値で置換されます。

ミッシング値の HOLD ファイルでの保存

```
SET HOLDMISS={ON|OFF}
ON TABLE SET HOLDMISS {ON|OFF}
```

説明

ON

これによりミッシング値の HOLD ファイルでの保存が可能になります。TABLE は、データが見つからない場合にミッシングデータのデフォルト値を生成します。

OFF

ミッシングデータを HOLD ファイルに格納することはできません。デフォルト値は OFF です。

HOLD ファイルでのミッシング値使用上の注意

- HOLDMISS ON を設定すると、MISSING=ON 属性が抽出ファイルのすべてのフィールドに追加されます。
- 次の場合、データは検索されません。
    - SET ALL が ON に設定されている。

- マルチパスリクエスト。
- ACROSS ステートメントが発行されている。

HOLDMISS によるミッシング値の保持

```
SET HOLDMISS=ON
TABLE FILE MOVIES
 SUM WHOLESALEPR
BY CATEGORY ACROSS RATING
 ON TABLE HOLD AS HLDM
END
TABLE FILE HLDM
 PRINT *
 END
```

出力結果は次のとおりです。

```
CATEGORY  WHOLESALEPR  WHOLESALEPR  WHOLESALEPR  WHOLESALEPR  WHOLESALEPR
--------  -----------  -----------  -----------  -----------  -----------
ACTION              .            .        20.98            .        34.48
CHILDREN        54.49        51.38            .            .           .
CLASSIC         40.99       160.80            .            .           .
COMEDY              .            .        46.70        30.00        13.75
DRAMA               .            .            .            .        10.00
FOREIGN         13.25            .        62.00            .        70.99
MUSICALS        15.00            .        13.99         9.99        13.99
MYSTERY             .         9.00        18.00         9.00        80.97
SCI/FI              .            .            .        35.99        43.53
TRAIN/EX            .        60.98            .            .           .
```

ミッシング値をフォーマット再設定フィールドへ継承するリクエスト

リクエストでフィールドフォーマットを再設定 (例、SUM フィールド/フォーマット) するときは、内部 COMPUTE フィールドが作成されてフォーマットが再設定されたフィールド値が格納され、レポート出力に表示されます。元のフィールドにミッシング値が含まれている場合は、COMPMISS パラメータを ON に設定することで、そのミッシング値を内部フィールドに継承することができるようになります。ミッシング値が内部フィールドに継承されない場

合は、ゼロ (数字フィールド) またはブランク (文字フィールド) が表示されます。ミッシング値が内部フィールドに継承される場合は、レポート出力にミッシングデータシンボルが表示されます。

フォーマット再設定フィールドのミッシング値制御

```
SET COMPMISS = {ON|OFF}
```

説明

ON

ミッシング値をフォーマットが再設定されたフィールドに継承させます。デフォルト値は ON です。

OFF

フォーマットが再設定されたフィールドにブランクまたは 0 (ゼロ) を表示します。

注意：COMPMISS パラメータを ON TABLE コマンドで設定することはできません。

フォーマット再設定フィールドのミッシング値制御

次のプロシジャは、SALES データソースの店舗 14Z からの RETURNTS フィールドを表示します。COMPMISS を OFF に設定した場合、フォーマットが再設定されたフィールド値のミッシング値は 0 (ゼロ) として列に表示されます。

注意：この例を実行する前に、SALES データソースにミッシング値を追加する SALEMISS プロシジャを実行しておく必要があります。

```
SET COMPMISS = OFF
TABLE FILE SALES
PRINT RETURNS RETURNS/D12.2 AS 'REFORMATTED,RETURNS'
BY STORE_CODE
```

```
WHERE STORE_CODE EQ '14Z'
END
```

出力結果は次のとおりです。

```
                         REFORMATTED
STORE_CODE  RETURNS      RETURNS
----------  -------      -----------
14Z                2             2.00
                   2             2.00
                   0              .00
                   .              .00
                   4             4.00
                   0              .00
                   3             3.00
                   4             4.00
                   .              .00
                   4             4.00
```

COMPMISS を ON に設定した場合、フォーマットが再設定された RETURNS 列の値が欠落している箇所には、ミッシングデータ記号が表示されます。

```
SET COMPMISS = ON
TABLE FILE SALES
PRINT RETURNS RETURNS/D12.2 AS 'REFORMATTED,RETURNS'
BY STORE_CODE
WHERE STORE_CODE EQ '14Z'
END
```

出力結果は次のとおりです。

```
                         REFORMATTED
STORE_CODE  RETURNS      RETURNS
----------  -------      -----------
14Z                2             2.00
                   2             2.00
                   0              .00
                   .                .
                   4             4.00
                   0              .00
                   3             3.00
                   4             4.00
                   .                .
                   4             4.00
```

SET COMPMISS の使用上の注意

COMPMISS を ON に設定して HOLD ファイルを作成する場合、HOLD マスターファイルのフォーマットの再設定されたフィールドには、元のフィールドと同様に MISSING = ON が記述されます。COMPMISS を OFF に設定する場合、作成したマスターファイルのフォーマットを再設定したフィールドには、MISSING = ON は記述されません。

ミッシングセグメントインスタンスの処理

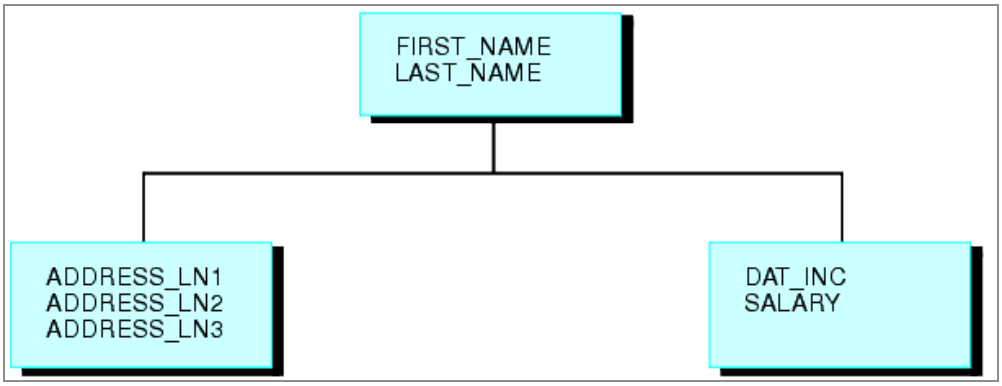
複数セグメントデータソースで親セグメントのインスタンスに子インスタンスが存在しない場合、存在しない子インスタンスは「ミッシングインスタンス」と呼ばれます。

ミッシングセグメントインスタンスのあるデータソースからリクエストを書き込むと、そのミッシングインスタンスによりレポートが影響されます。たとえば、リクエストがセグメントのフィールドおよびその子を指定すると、子を持たない親セグメントインスタンスはレポートから除外されます。フィールドが表示フィールドでもソートフィールドでも相違はありません。

子セグメント内のインスタンスが孫を持たないとき、そのインスタンス、その親、その親の親と続く、ルートセグメントまでのパスは「ショートパス」と呼ばれます。ユニークセグメントがミッシングセグメントとして見なされることはありません。

たとえば、次の EMPLOYEE データソースの一部について考察します。

- 最上位のセグメントに従業員名が含まれています。
- 左のセグメントには住所が含まれています。
- 右のセグメントには各従業員の給与履歴が含まれています。給与履歴は従業員が新しい給与を与えられた日付および給与額を含みます。



一部の従業員の給与が派遣会社から支払われている場合を想定します。これらの従業員には会社の給与履歴はありません。給与履歴セグメントのこれらの従業員を参照するインスタンスは欠落しています。

存在しない子インスタンスは、親セグメントインスタンスがレポート結果に含まれるかどうかに影響します。SET ALL パラメータおよび ALL. 接頭語により、親セグメントデータをレポートに含めることが可能になります。

子インスタンスを持たないセグメントに対するレポート

次のリクエストは各従業員の給与履歴を表示します。

```
TABLE FILE EMPLOYEE
PRINT SALARY
BY LAST_NAME BY FIRST_NAME
BY DAT_INC
END
```

ただし、LAST_NAME と FIRST_NAME フィールドがルートセグメントに属し、DAT_INC および SALARY フィールドが子給与履歴セグメントに属するため、従業員の 2 名、Davis と Gardner は次のレポートから除外されています。Davis と Gardner には給与履歴セグメントに子インスタンスがないため、レポートから除外されています。

出力結果は次のとおりです。

```
LAST_NAME       FIRST_NAME   DAT_INC          SALARY
---------       ----------   -------          ------
BANNING         JOHN        82/08/01      $29,700.00
BLACKWOOD       ROSEMARIE   82/04/01      $21,780.00
CROSS           BARBARA     81/11/02      $25,775.00
                            82/04/09      $27,062.00
GREENSPAN       MARY        82/04/01       $8,650.00
                            82/06/11       $9,000.00
IRVING          JOAN        82/01/04      $24,420.00
                            82/05/14      $26,862.00
JONES           DIANE       82/05/01      $17,750.00
                            82/06/01      $18,480.00
MCCOY           JOHN        82/01/01      $18,480.00
MCKNIGHT        ROGER       82/02/02      $15,000.00
                            82/05/14      $16,100.00
ROMANS          ANTHONY     82/07/01      $21,120.00
```

```
SMITH           MARY        82/01/01       $13,200.00
                RICHARD     82/01/04        $9,050.00
                            82/05/14        $9,500.00
STEVENS         ALFRED      81/01/01       $10,000.00
                            82/01/01       $11,000.00
```

子インスタンスを持つセグメントに対するレポート

次のリクエストは EMPDATA および TRAIN2 データソースの従業員のコースコードおよび支出を表示します。このレポート出力は、COURSECODE または EXPENSES フィールドにインスタンスを持つ従業員のすべてを表示します。これらのフィールドのいずれかのインスタンスが欠落している従業員はレポートから除外されます。これらのフィールドの 1 つのみのインスタンスを持つ従業員には、それぞれのフィールドに、ミッシング値の割り当て要素が表示されます。この例では、Henry Chisolm が 2 つのコースを受講し、1 つのみに対して経費が発生します。そのため、EXPENSES フィールドにはミッシングインスタンスの割り当て要素が表示されます。

```
JOIN EMPDATA.PIN IN EMPDATA TO ALL TRAINING.PIN IN TRAIN2 AS JOIN1
TABLE FILE EMPDATA
PRINT LASTNAME AND FIRSTNAME AND COURSECODE AND EXPENSES
BY PIN
END
```

出力結果は次のとおりです。

PIN	LASTNAME	FIRSTNAME	COURSECODE	EXPENSES
000000010	VALINO	DANIEL	PDR740	2300.00
000000030	CASSANOVA	LOIS	NAMA730	2600.00

	CASSANOVA	LOIS	EDP090	2300.00
. . .				
000000350	FERNSTEIN	ERWIN	SSI220	1850.00
	FERNSTEIN	ERWIN	MC90	1730.00
	FERNSTEIN	ERWIN	UMI720	3350.00
000000360	CHISOLM	HENRY	EDP090	0.00
	CHISOLM	HENRY	EDP690	3000.00
000000370	WANG	JOHN	UMI710	2050.00
000000380	ELLNER	DAVID	EDP090	.
	ELLNER	DAVID	UNI780	3350.00
000000410	CONTI	MARSHALL	EDP690	3100.00

注意：このレポート出力はサンプル用のため省略されています。

ALL. 接頭語によるミッシングインスタンスのレポートへの追加

リクエストにより子の欠落している親セグメントインスタンスが除外される場合、親セグメントのフィールドの 1 つに ALL. 接頭語を設定することにより、その親インスタンスを追加することができます。

ミッシングインスタンスを含むセグメントのフィールドを選別する WHERE または IF 条件をリクエストに含めると、ALL. 接頭語を使用しても親インスタンスはレポートから除外されます。これらのインスタンスを含めるには、 ALL=PASS コマンドを使用します。

ALL. 接頭語によるミッシングセグメントインスタンスの追加

次のリクエストは各従業員の給与履歴を表示します。従業員の Elizabeth Davis と David Gardner に給与履歴はありませんが、レポートに含まれます。

```
TABLE FILE EMPLOYEE
PRINT SALARY
BY ALL.LAST_NAME BY FIRST_NAME
BY DAT_INC
END
```

出力結果は次のとおりです。

```
LAST_NAME       FIRST_NAME   DAT_INC          SALARY
---------       ----------   -------          ------
BANNING         JOHN        82/08/01       $29,700.00
BLACKWOOD       ROSEMARIE   82/04/01       $21,780.00
CROSS           BARBARA     81/11/02       $25,775.00
                            82/04/09       $27,062.00
DAVIS           ELIZABETH          .                .
GARDNER         DAVID              .                .
GREENSPAN       MARY        82/04/01        $8,650.00
                            82/06/11        $9,000.00
IRVING          JOAN        82/01/04       $24,420.00
                            82/05/14       $26,862.00
JONES           DIANE       82/05/01       $17,750.00
                            82/06/01       $18,480.00
```

```
MCCOY           JOHN        82/01/01       $18,480.00
MCKNIGHT        ROGER       82/02/02       $15,000.00
                            82/05/14       $16,100.00
ROMANS          ANTHONY     82/07/01       $21,120.00
SMITH           MARY        82/01/01       $13,200.00
                RICHARD     82/01/04        $9,050.00
                            82/05/14        $9,500.00
STEVENS         ALFRED      81/01/01       $10,000.00
                            82/01/01       $11,000.00
```

SET ALL パラメータによるミッシングインスタンスのレポートへの追加

リクエストの実行前に SET ALL コマンドを発行することにより、子が欠落している親インスタンスの処理方法を制御することができます。JOIN では、SET ALL=ON コマンドを発行して、LEFT OUTER JOIN の処理を制御します。

注意：WHERE または IF 条件を含むリクエストは、ミッシングインスタンスを含むセグメントを選別します。この場合、SET ALL=ON コマンドを使用しても、親セグメントのインスタンスは無視されます。これらのインスタンスをレポートに含めるには、SET ALL=PASS コマンドを使用します。

WebFOCUS では、SET ALL = ON または JOIN LEFT_OUTER コマンドにより、LEFT OUTER JOIN を指定します。LEFT OUTER JOIN の場合、ホストファイルのレコードは、すべてレポート出力上に表示されます。クロスリファレンスセグメントインスタンスがホストセグメントインスタンスに存在しない場合、レポート出力にはクロスリファレンスセグメントのフィールドのミッシング値が表示されます。

従属セグメントに選別条件が設定されている場合、選別条件を満たさないセグメントインスタンスおよび対応するホストセグメントインスタンスはレポート出力から除外されます。セグメントインスタンスがミッシングの場合、セグメントのフィールドにミッシング値が割り当てられていないため、ミッシング値のテストは失敗します。

LEFT OUTER JOIN をリレーショナルエンジンで実行し、次の条件が両方とも適用される場合、ミッシングのクロスリファレンスセグメントインスタンスが存在するホストレコードは、WebFOCUS とは多少異なる方法で処理されます。

- クロスリファレンスセグメントに選別条件が設定されている。
- ホストのセグメントインスタンスに、対応するクロスリファレンスセグメントインスタンスが存在しない。これは「ショートパス」と呼ばれます。

これらの条件に該当する場合、WebFOCUS ではレポート出力からホストレコードが除外されますが、リレーショナルエンジンでは従属セグメントのフィールドにミッシング値を割り当て、選別条件を適用します。ミッシング値が選別条件をパスする場合は、レコード全体がレポート出力に保持されます。この種の処理は、クロスリファレンスファイルに一致レコードが存在しないホストレコードをすべて検索または集計する場合や、ミッシングインスタンスを含むクロスリファレンスセグメントから別の従属セグメントへの DEFINE ベースの JOIN を作成する場合に役立ちます。

LEFT OUTER JOIN が有効な場合に、WebFOCUS でミッシングセグメントインスタンスのフィールドにミッシング値を割り当てるには、SET SHORTPATH=SQL コマンドを発行します。

子が欠落している親インスタンスの追加

```
SET ALL= {OFF|ON|PASS}
```

説明

OFF

子が欠落している親インスタンスをレポートから除外します。デフォルト値は OFF です。

ON

子が欠落している親インスタンスをレポートに含めます。ただし、ミッシングセグメントのテストが失敗すると、その親が出力結果から除外されます。これは ALL. 演算接頭語と同様です。

PASS

(ミッシングセグメントテストをパスする) 子セグメントのフィールドを選別する WHERE または IF 条件が存在する場合でも、子の欠落している親インスタンスを含めます。

SET ALL によるミッシングセグメントインスタンスの追加

次のリクエストは、ALL=PASS コマンドが設定されているため、コースを受講したかどうかに関わらず、すべての従業員を表示します。

SET ALL=ON コマンドを使用した場合、コースを受講しなかった従業員は WHERE 条件のために除外されます。

```
JOIN EMPDATA.PIN IN EMPDATA TO ALL TRAINING.PIN IN TRAINING AS JOIN1
SET ALL = PASS
TABLE FILE EMPDATA
PRINT LASTNAME AND FIRSTNAME AND COURSECODE AND EXPENSES
BY PIN
WHERE EXPENSES GT 3000
END
```

出力結果は次のとおりです。

PIN	LASTNAME	FIRSTNAME	COURSECODE	EXPENSES
000000020	BELLA	MICHAEL	.	.
000000040	ADAMS	RUTH	EDP750	3,400.00
000000050	ADDAMS	PETER	UMI720	3,300.00
000000060	PATEL	DORINA	.	.
000000070	SANCHEZ	EVELYN	.	.
000000080	SO	PAMELA	EDP690	3,200.00
	SO	PAMELA	BIT420	3,350.00
000000090	PULASKI	MARIANNE	.	.
000000100	ANDERSON	TIM	NAMA930	3,100.00
000000130	CVEK	MARCUS	.	.
000000140	WHITE	VERONICA	BIT420	3,600.00
000000150	WHITE	KARL	UNI780	3,400.00
000000170	MORAN	WILLIAM	.	.
000000190	MEDINA	MARK	EDP690	3,150.00
000000220	LEWIS	CASSANDRA	.	.
000000230	NOZAWA	JIM	.	.
000000300	SOPENA	BEN	.	.
000000340	GOTLIEB	CHRIS	SSI670	3,300.00
	GOTLIEB	CHRIS	EDP750	3,450.00
000000350	FERNSTEIN	ERWIN	UMI720	3,350.00
000000380	ELLNER	DAVID	UNI780	3,350.00
000000390	GRAFF	ELAINE	.	.
000000400	LOPEZ	ANNE	.	.
000000410	CONTI	MARSHALL	EDP690	3,100.00

LEFT OUTER JOIN でのショートパス処理の制御

```
SET SHORTPATH = {FOCUS|SQL}
```

説明

FOCUS

対応するクロスリファレンスセグメントが存在せず、レポートにクロスリファレンスセグメントの選別条件が設定されている場合に、レポート出力のホストセグメントを除外します。デフォルト値は FOCUS です。

SQL

OUTER JOIN のミッシングクロスリファレンスセグメントフィールドに、ミッシング値を割り当てます。このレコードに対して選別条件を適用し、選別テストにパスした場合、レコードをレポート出力に保持します。

注意：SET ALL=ON コマンドまたは JOIN LEFT_OUTER コマンド (マスターファイル内の記述、またはそれ以外のいずれも) の結果として、有効な OUTER JOIN が存在する必要があります。

SET SHORTPATH = SQL 使用上の注意

FOCUS データソースは、SET SHORTPATH = SQL で使用される JOIN のホストファイルとしてサポートされます。

OUTER JOIN 処理の制御

次のプロシジャは、JOIN で使用する ORAEMP と ORAEDUC という 2 つの Oracle テーブルを作成します。

```
TABLE FILE EMPLOYEE
SUM LAST_NAME FIRST_NAME CURR_SAL CURR_JOBCODE DEPARTMENT
BY EMP_ID
ON TABLE HOLD AS ORAEMP FORMAT SQLORA
END
-RUN
TABLE FILE EDUCFILE
SUM COURSE_CODE COURSE_NAME
BY EMP_ID BY DATE_ATTEND
ON TABLE HOLD AS ORAEDUC FORMAT SQLORA
END
```

次のリクエストは、2 つの Oracle テーブルを結合し、LEFT OUTER JOIN を作成します (SET ALL = ON)。

```
JOIN EMP_ID IN ORAEMP TO ALL EMP_ID IN ORAEDUC AS J1
SET ALL = ON
TABLE FILE ORAEMP
PRINT COURSE_CODE COURSE_NAME
BY EMP_ID
END
```

この JOIN は OUTER JOIN であることから、ORAEMP 行はすべてレポート出力に表示されます。ORAEDUC テーブルのフィールドに対応する ORAEDUC 行が存在しない場合は、ミッシングデータ記号が表示されます。

```
EMP_ID     COURSE_CODE     COURSE_NAME
------     -----------     -----------
071382660  101             FILE DESCRPT & MAINT
112847612  101             FILE DESCRPT & MAINT
           103             BASIC REPORT PREP FOR PROG
117593129  101             FILE DESCRPT & MAINT
           103             BASIC REPORT PREP FOR PROG
           201             ADVANCED TECHNIQUES
           203             FOCUS INTERNALS
119265415  108             BASIC RPT NON-DP MGRS
119329144  .               .
123764317  .               .
126724188  .               .
219984371  .               .
326179357  104             FILE DESC & MAINT NON-PROG
           106             TIMESHARING WORKSHOP
           102             BASIC REPORT PREP NON-PROG
           301             DECISION SUPPORT WORKSHOP
           202             WHAT'S NEW IN FOCUS
451123478  101             FILE DESCRPT & MAINT
543729165  .               .
818692173  107             BASIC REPORT PREP DP MGRS
```

次のリクエストにより、ORAEDUC セグメントの選別条件が追加されます。選別条件を満たすには、コース名に「BASIC」という文字が含まれるか、コース名がミッシングの必要があります。

```
JOIN CLEAR
JOIN EMP_ID IN ORAEMP TO ALL EMP_ID IN ORAEDUC AS J1
SET ALL = ON
TABLE FILE ORAEMP
PRINT COURSE_CODE COURSE_NAME
BY EMP_ID
```

```
WHERE COURSE_NAME CONTAINS 'BASIC' OR COURSE_NAME IS MISSING
END
```

ただし、SET ALL = ON の場合は、ミッシング値を含む行はレポート出力に保持されません。

```
EMP_ID     COURSE_CODE     COURSE_NAME
------     -----------     -----------
112847612  103             BASIC REPORT PREP FOR PROG
117593129  103             BASIC REPORT PREP FOR PROG
119265415  108             BASIC RPT NON-DP MGRS
326179357  102             BASIC REPORT PREP NON-PROG
818692173  107             BASIC REPORT PREP DP MGRS
```

次のリクエストにより、SET SHORTPATH = SQL コマンドが追加されます。

```
JOIN CLEAR
JOIN EMP_ID IN ORAEMP TO ALL EMP_ID IN ORAEDUC AS J1
SET ALL = ON
SET SHORTPATH=SQL
TABLE FILE ORAEMP
PRINT COURSE_CODE COURSE_NAME
BY EMP_ID
WHERE COURSE_NAME CONTAINS 'BASIC' OR COURSE_NAME IS MISSING
END
```

これにより、「BASIC」という文字を含むレコードおよびミッシング値を含むレコードの両方が、レポート出力に表示されます。

```
EMP_ID     COURSE_CODE     COURSE_NAME
------     -----------     -----------
112847612  103             BASIC REPORT PREP FOR PROG
117593129  103             BASIC REPORT PREP FOR PROG
119265415  108             BASIC RPT NON-DP MGRS
119329144  .               .
123764317  .               .
126724188  .               .
219984371  .               .
326179357  102             BASIC REPORT PREP NON-PROG
543729165  .               .
818692173  107             BASIC REPORT PREP DP MGRS
```

一致クロスリファレンスレコードが存在しないホストレコードの検索

次のリクエストにより、コースを受講したことのない従業員が集計され、リスト表示されます。

```
JOIN LEFT_OUTER EMP_ID IN ORAEMP TO ALL EMP_ID IN ORAEDUC AS J1
SET ALL = ON
SET SHORTPATH=SQL
TABLE FILE ORAEMP
COUNT EMP_ID
LIST EMP_ID LAST_NAME FIRST_NAME
WHERE COURSE_NAME IS MISSING
END
```

出力結果は次のとおりです。

```
EMP_ID
COUNT    LIST  EMP_ID     LAST_NAME        FIRST_NAME
------   ----  ------     ---------        ----------
     5      1  119329144  BANNING          JOHN
            2  123764317  IRVING           JOAN
            3  126724188  ROMANS           ANTHONY
            4  219984371  MCCOY            JOHN
            5  543729165  GREENSPAN        MARY
```

FOCUS データソースでのミッシングインスタンスのテスト

ALL PASS パラメータを使用し、ミッシング子値を含む親インスタンスのみを含むレポートを作成することができます。これを実行するには、ミッシングインスタンスを含むセグメントのすべての既存インスタンスを選別するリクエストを記述します。ALL パラメータを PASS に設定後、ミッシングの子の親インスタンスのみがレポートに表示されます。

FOCUS データソースのミッシングインスタンスのテスト

次のリクエストは、COURSECODE フィールドのミッシングインスタンスのテストを実行します。COURSECODE で 'XXXX' と等しくなるものはないため、COURSECODE のミッシングインスタンスを含む従業員のみがレポート出力に表示されます。

```
JOIN EMPDATA.PIN IN EMPDATA TO ALL TRAINING.PIN IN TRAINING AS JOIN1
SET ALL = PASS
TABLE FILE EMPDATA
PRINT LASTNAME AND FIRSTNAME AND COURSECODE AND EXPENSES
BY PIN
WHERE COURSECODE EQ 'XXXX'
END
```

出力結果は次のとおりです。

PIN	LASTNAME	FIRSTNAME	COURSECODE	EXPENSES
000000020	BELLA	MICHAEL	.	.
000000060	PATEL	DORINA	.	.
000000070	SANCHEZ	EVELYN	.	.
000000090	PULASKI	MARIANNE	.	.
000000130	CVEK	MARCUS	.	.

000000170	MORAN	WILLIAM	.	.
000000220	LEWIS	CASSANDRA	.	.
000000230	NOZAWA	JIM	.	.
000000300	SOPENA	BEN	.	.
000000390	GRAFF	ELAINE	.	.
000000400	LOPEZ	ANNE	.	.

NODATA 文字列の設定

レポートでの NODATA はデータが存在しないことを示しています。デフォルトの NODATA はピリオド (.) です。ただし、この文字の指定は変更することができます。

NODATA 列の設定

```
SET NODATA = string
```

説明

string

レポートのミッシング値を示すために使用する文字列です。デフォルト文字 (列) は、ピリオド (.) です。この文字列は最大 11 バイトで指定します。NONE、N/A および MISSING

の選択が一般的です。

文字を非表示にするための NODATA の設定

文字を非表示にするには、次のコマンドを発行します。

```
SET NODATA = ' '
```

NODATA 文字列の設定

次のリクエストでは NODATA 文字列は MISSING に設定されています。レポートには、デフォルトのピリオド (.) の代わりに「MISSING」が表示されます。

```
SET NODATA=MISSING

TABLE FILE EMPLOYEE
PRINT CURR_SAL BY LAST_NAME BY FIRST_NAME
ACROSS DEPARTMENT
END
```

出力結果は次のとおりです。

LAST_NAME	FIRST_NAME	DEPARTMENT MIS	PRODUCTION
BANNING	JOHN	MISSING	$29,700.00
BLACKWOOD	ROSEMARIE	$21,780.00	MISSING
CROSS	BARBARA	$27,062.00	MISSING
GREENSPAN	MARY	$9,000.00	MISSING
IRVING	JOAN	MISSING	$26,862.00
JONES	DIANE	$18,480.00	MISSING
MCCOY	JOHN	$18,480.00	MISSING
MCKNIGHT	ROGER	MISSING	$16,100.00
ROMANS	ANTHONY	MISSING	$21,120.00
SMITH	MARY	$13,200.00	MISSING
	RICHARD	MISSING	$9,500.00
STEVENS	ALFRED	MISSING	$11,000.00

データソースの結合 (JOIN)

複数の関連するデータソースを結合して大規模な統合データ構造を作成し、そのデータ構造から単一リクエストでレポートを作成することができます。この結合 (JOIN) は仮想構造です。結合された複数のデータソースには、単一データソースのようにアクセスすることができます。合計 1023 個のセグメントに対して、一度に 1022 個までの JOIN を有効にすることができます。この値は使用中のセグメントの数、およびフィールドの数と長さにより異なります。すべてのフィールドの長さには 32 キロバイトの制限があります。

JOIN に使用可能なデータソースについての詳細は、JOIN のタイプを参照してください。

JOIN のタイプ

2 つのデータソースファイルを結合する場合、特定のレコードが一方のファイルには存在するが、他方のファイルに存在しないことがあります。「INNER JOIN」は、両方のファイルに存在する共通レコードのみをレポートに表示する結合です。この JOIN では、一方のファイルに存在し、他方のファイルに存在しないレコードはすべて除外されます。「LEFT OUTER JOIN」は、両方のファイルに存在する共通レコードのすべてと、ホストファイルにのみ存在するレコードのすべて (クロスリファレンスファイルに存在しないレコードを含む) をレポートに表示する結合です。「FULL OUTER JOIN」は、両方のファイルに存在する共通レコードのすべてと、両方のファイルに存在するレコードのすべて (一方のファイルに存在し、他方のファイルに存在しないレコードを含む) をレポートに表示する結合です。FULL OUTER JOIN は、リレーショナルデータソースでのみサポートされます。

SET ALL コマンドは、すべての結合の実装状態をグローバルに決定するコマンドです。SET ALL=ON を発行すると、すべての結合が OUTER JOIN として処理されます。SET ALL=OFF を発行すると、すべての結合が INNER JOIN として処理されます。

JOIN コマンドを個別に発行することで、グローバルに設定されている JOIN とは別に、実行する JOIN のタイプを指定することができます。この構文は、FOCUS、XFOCUS、リレーショナルデータベース、ADABAS でサポートされています。JOIN コマンドで JOIN タイプを指定しない場合は、ALL パラメータ設定で指定した JOIN タイプが実行されます。

また、複数のデータソースを結合する際に、次の用法のいずれかを使用して、各データソースのレコードを一致させる方法を指定することもできます。1 つ目の用法は「等結合 (EQUIJOIN)」、2 つ目の用法は「条件付き JOIN」と呼ばれます。2 つの JOIN の用法のどち

らを使用するかを決定する際は、2 つのデータソース間に等価条件が成立するときは、条件付き JOIN より EQUIJOIN を使用すると効率がよいことを覚えておくことが重要です。

各データソースに 1 つ、共通のフォーマット (文字、数値、または日付) と値を含んだ 2 つのフィールドがある複数データソースを結合するとき、EQUIJOIN 構造を使用することができます。売上データソース (ホストファイル) の製品コードフィールドからの製品データソース (クロスリファレンスファイル) の製品コードフィールドへの結合は、EQUIJOIN の一例です。EQUIJOIN の使用についての詳細は、EQUIJOIN の作成を参照してください。

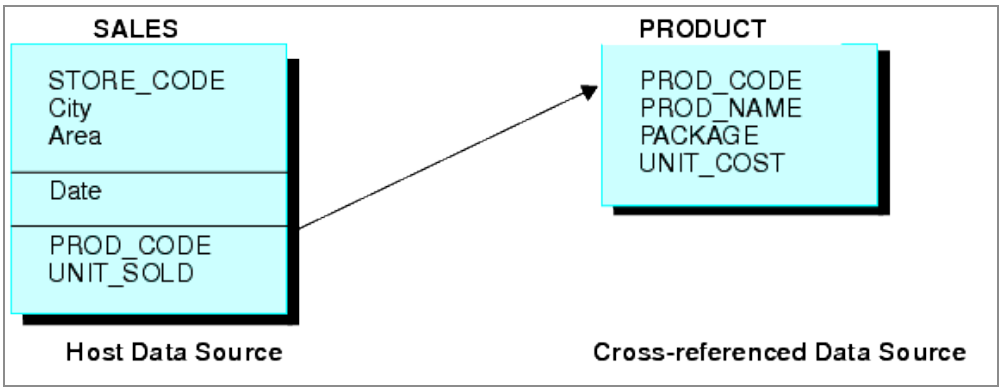
条件付き JOIN は、フィールド間の等価性のみではなく、WHERE 条件に基づく結合を指定します。また、ホストおよびクロスリファレンス JOIN フィールドに一致するフォーマットが含まれている必要はありません。たとえば、ID 番号が振られた従業員リストであるデータソース (ホストファイル) と、研修コースとこれらのコースを受講した従業員のリストであるデータソース (クロスリファレンスファイル) を想定します。条件付き JOIN によりホストファイルの従業員 ID をクロスリファレンスファイルの従業員 ID に結合すると、特定の日付範囲の間に研修コースを受講した従業員 (WHERE 条件) を特定できます。条件付き JOIN についての詳細は、条件付き JOIN の使用を参照してください。

JOIN にはユニーク JOIN と非ユニーク JOIN があります。ユニーク JOIN 構造 (1 対 1 の JOIN 構造) では、ホストデータソースの 1 つの値とクロスリファレンスデータソースの 1 つの値を一致させます。従業員データソースの従業員 ID を給与データソースの従業員 ID と結合することは、ユニーク EQUIJOIN 構造 の一例です。

非ユニーク JOIN 構造 (1 対 n JOIN 構造) では、ホストデータソースの 1 つの値とクロスリファレンスフィールドの複数の値を一致させます。企業の従業員データソースの従業員 ID を、研修クラスリストデータソースの従業員 ID と結合すると、各従業員の受講するすべてのクラスのリストが作成されます。この場合、ホストファイルの各 ID の単一インスタンスがクロスリファレンスファイルのその ID の複数インスタンスに結合されます。

JOIN データ構造

SALES と PRODUCT というデータソースについて考察します。SALES データソースの各店舗レコードには、PROD_CODE フィールドのインスタンスが多数含まれています。製品コードの各インスタンスに関連する製品情報をこのデータソースに格納すると、データの重複になります。代わりに、SALES データソースの PROD_CODE を PRODUCT データソースの PROD_CODE に結合します。PRODUCT データソースには各製品コードの単一インスタンスと各インスタンスに関連する製品情報のみが含まれているため、データ領域を節約でき、製品情報の管理をより容易に行えます。下図は、EQUIJOIN の結合構造の例です。



結合可能および不可能なデータソース

データソースを JOIN 構造のホストファイルとクロスリファレンスファイルに使用できるかどうかは、結合するデータソースのタイプにより異なります。

- 通常、JOIN は FOCUS で読み取り可能な任意のファイル間に設定することができます。
- DBA セキュリティで保護されるデータソースは、制約付きで結合することができます。詳細は、結合データ構造での DBA セキュリティの注意を参照してください。
- 条件付き JOIN は、FOCUS、ADABAS、およびすべてのリレーショナルデータソースでのみサポートされます。

結合データ構造での DBA セキュリティの注意

- 同一のパスワードを使用する限り、DBA で保護されたデータソースを DBA で保護されたもう 1 つのデータソースに結合することができます。
- さらに、DBAFILE 属性をセキュリティ定義に追加することにより、異なるパスワードを含む DBA で保護されたデータソースを結合することができます。DBAFILE 属性は、複数のマスターファイルの異なるパスワードと制限を含む集中マスターファイルを指定します。DBAFILE を次の構文で使用すると、各ファイルに別のパスワードを設定することができます。

```
SET PASS = pswd1 IN file1, pswd2 IN file2
```

JOIN の各ファイルに対する個別の DBA 情報は有効のままです。

- DBA で保護されたホストファイルを、DBA で保護されていないクロスリファレンスファイルに結合することもできます。DBA の情報はホストファイルから取得されます。

ユニークおよび非ユニーク JOIN 構造

ユニーク JOIN 構造ではホストフィールドの単一値をクロスリファレンスフィールドの単一値と一致させます。非ユニーク JOIN 構造ではホストフィールドの単一値をクロスリファレンスフィールドの複数値と一致させます。

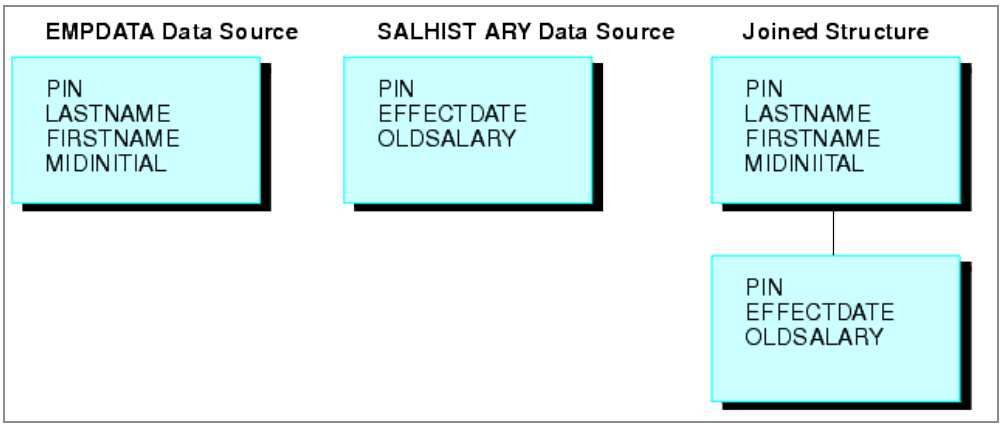
JOIN コマンドの ALL パラメータにより JOIN 構造が非ユニークであることが示されます。

- JOIN 構造がユニークであることが確実な場合のみ、ALL パラメータを省略します。ALL パラメータを省略するとオーバーヘッドを軽減できます。
- JOIN 構造がユニークである場合に ALL パラメータを使用しても、JOIN 構造は適切に作成されます。JOIN 構造がユニークかどうか確かでない場合、ALL パラメータを使用します。これにより、構造がユニークであるかどうかに関わらず、レポートにクロスリファレンスファイルからのすべての関連データが必ず含まれるようになります。

ユニーク EQUIJOIN 構造

次の例では、ユニーク EQUIJOIN 構造について説明します。2 つの FOCUS データソースが結合されます。EMPDATA データソースおよび SALHIST データソースです。両方のデータソースは PIN に基づいて構成されており、両方のファイルのルートセグメントの PIN フィールドで結合されます。各 PIN のセグメントインスタンスが EMPDATA データソースに 1 つ、SALHIST データソースに 1 つあります。これら 2 つのデータソースを結合するには、次の JOIN コマンドを発行します。

```
JOIN PIN IN EMPDATA TO PIN IN SALHIST
```



非ユニーク EQUIJOIN 構造

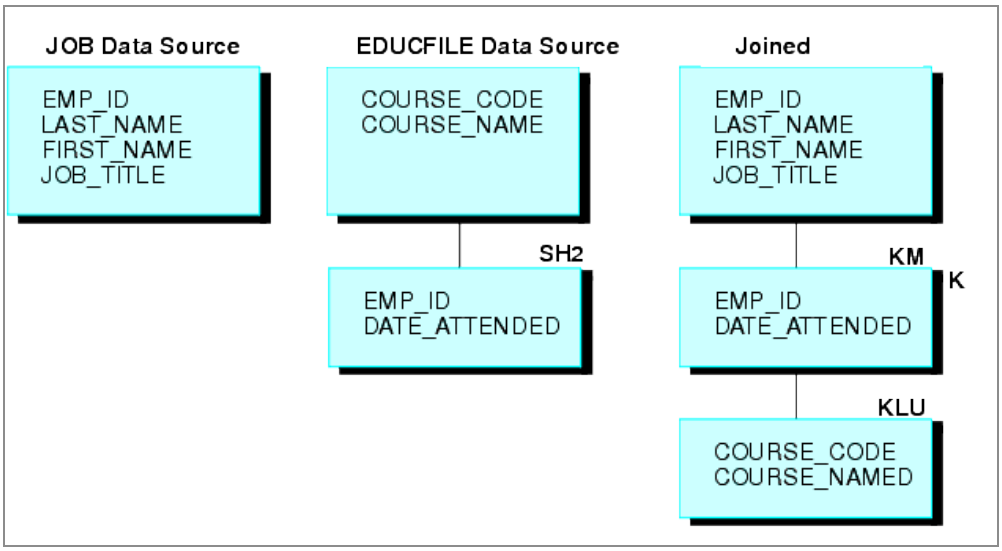
ホストファイルのフィールド値が、クロスリファレンスファイルの多くのセグメントインスタンスに存在する場合、JOIN 構文に ALL 句を含めます。この構造は、「非ユニーク JOIN 構造」と呼ばれます。

たとえば、JOB データソースに従業員の社内研修の出席を記録する EDUCFILE データソースを結合することを想定します。JOIN 構造は、次の図に示されています。

JOB データソースは従業員により構成されますが EDUCFILE データソースは研修コースにより構成されます。データソースは EMP_ID フィールドで結合されます。従業員 1 名に対応する役職は 1 つ、出席できる研修は複数である可能性があります。このため、JOB データソースの従業員のセグメントインスタンスは 1 つですが、EDUCFILE データソースには出席した研修コースとして多くのインスタンスを含めることができます。

これら 2 つのデータソースを結合するには、次の ALL 句付き JOIN コマンドを発行します。

```
JOIN EMP_ID IN JOB TO ALL EMP_ID IN EDUCFILE
```



ユニーク JOIN でのラギング値を修正するには

親セグメントに 2 つ以上のユニーク子セグメントが存在し、それぞれが複数の子を持つ場合、レポートでのミッシング値の表示が不正確になる可能性があります。その結果、それ以外の子の値が、レポートに正しく整列されなくなります。これらの不整列値は「ラギング値」と呼ばれます。JOINOPT パラメータは、ラギング値を補正することで、出力を正しく整列します。

```
SET JOINOPT={NEW|OLD|GNTINT}
```

説明

NEW

セグメントを左から右へ、上から下へ取得することを指定します。その結果、各レコードのすべてのデータが正しく整列されて表示されます。ミッシング値は、データに存在する場合にのみ表示されます。

OLD

セグメントをユニークセグメントとして取得することを指定します。その結果、値を持つすべてのレコードをレポートに表示する場合、ミッシング値が表示されます。これにより、ラギング値が表示される場合があります。デフォルト値は OLD です。

GNTINT

セグメントを左から右へ、上から下へ取得することを指定します。その結果、各レコードのすべてのデータが正しく整列されて表示されます。ミッシング値は、データに存在する場合にのみ表示されます。

注意：GNTINT 値は、ラギング値を補正するとともに、異なる数値データタイプの結合を可能にします。詳細は、異なる数値データタイプを含むフィールドの結合を参照してください。

プロシジャによるユニークセグメントと複数の子セグメントのラギング値の修正

この例では、JOINOPT パラメータを使用してラギング値を修正するシナリオを想定します。ラギング値はミッシング値として表示されるため、それぞれの値が 1 行ずれて表示されます。

単一セグメントのホストファイル (ROUTES) が、それぞれ 2 つのセグメントで構成された 2 つのファイル (ORIGIN および DEST) に結合されています。これらのファイルを結合して、列車番号および駅名に対応する都市名を表示するレポートを生成します。

次のリクエストは、出発地 (OR_CITY) および目的地 (DE_CITY) の都市名を表示します。ここで、ミッシング値が生成され、駅名および対応する都市名のデータがラギング値として 1 行ずれて表示されています。

```
TABLE FILE ROUTES
PRINT TRAIN_NUM
OR_STATION OR_CITY
DE_STATION DE_CITY
END
```

出力結果は次のとおりです。

```
TRAIN_NUM   OR_STATION   OR_CITY    DE_STATION   DE_CITY
---------   ----------   -------    ----------   -------
101         NYC          NEW YORK   ATL          .
202         BOS          BOSTON     BLT          ATLANTA
303         DET          DETROIT    BOS          BALTIMORE
404         CHI          CHICAGO    DET          BOSTON
505         BOS          BOSTON     STL          DETROIT
505         BOS          .          STL          ST.LOUIS
```

SET JOINOPT=NEW を発行すると、ミッシング値を含まずに、セグメントが正しい順序で取得されます (左から右へ、上から下へ)。

```
SET JOINOPT=NEW
TABLE FILE ROUTES
PRINT TRAIN_NUM
OR_STATION OR_CITY
DE_STATION DE_CITY
END
```

修正したレポートには 6 行ではなく 5 行のみが表示され、駅名および都市名のデータが正しく整列されています。出力結果は次のとおりです。

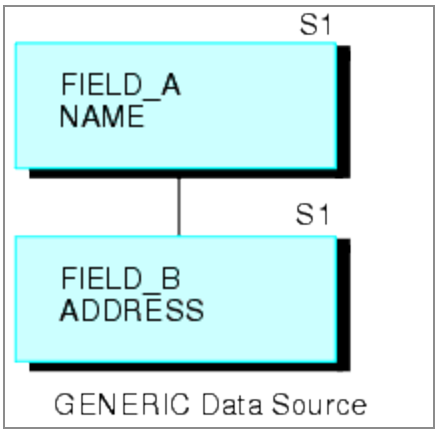
```
TRAIN_NUM   OR_STATION    OR_CITY    DE_STATION   DE_CITY
---------   ----------    -------    ----------   -------
101         NYC           NEW YORK   ATL          ATLANTA
202         BOS           BOSTON     BLT          BALTIMORE
303         DET           DETROIT    BOS          BOSTON
404         CHI           CHICAGO    DET          DETROIT
505         BOS           BOSTON     STL          ST.LOUIS
```

再帰的 JOIN 構造

FOCUS データソースをそれ自体に結合し、再帰的構造を作成することができます。再帰的構造の最も一般的なタイプは、親セグメントが子セグメントに結合され、親が子の子になるものです。この方法により、データソースに実際に存在するよりも多くのセグメントレベルを使用して、レポートを作成することが可能になります (パーツリストの保存などに役立ちます)。

再帰的 JOIN 構造の理解

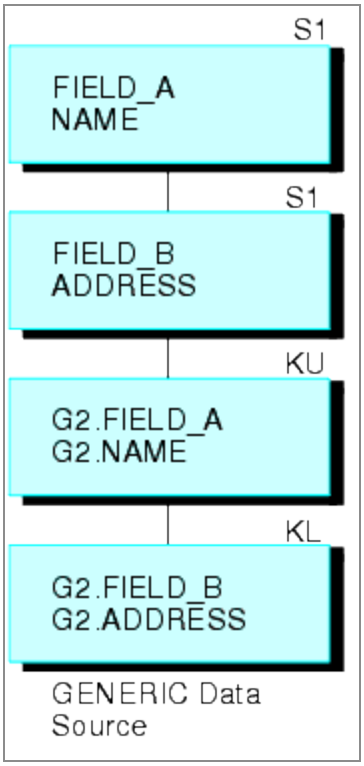
次の例では、GENERIC データソースはセグメント A および B で構成されています。



次のリクエストは再帰的構造を作成します。

```
JOIN FIELD_B IN GENERIC TAG G1 TO FIELD_A IN GENERIC TAG G2 AS RECURSIV
```

これは、次に示す JOIN 構造になります。



2 つのセグメントは、最下位で繰り返されています。(結合先のフィールドを除いた) 繰り返しセグメントのフィールドを参照するには、フィールド名およびエイリアスの前にタグ名を追加し、これらをピリオド (.) で区切る、または JOIN 名のはじめの 4 バイトをフィールド名およびエイリアスに追加します。上記の例では、JOIN 名は RECURSIV です。最下位セグメ

ントの FIELD_B を G2.FIELD_B (または RECUFIELD_B) として参照します。詳細は、再帰的 JOIN 構造使用上の注意を参照してください。

再帰的 JOIN 構造使用上の注意

- ユニーク JOIN 名を指定、または JOIN コマンドでタグ名を使用する必要があります。そうしないと、JOIN 構造の最下位で繰り返されるセグメントのフィールドを参照することはできません。
- 再帰的 JOIN 構造でタグ名を使用する場合、次のガイドラインに従います。
    - 再帰的 JOIN でタグ名が指定されている場合、重複フィールド名はそのタグ名で修飾する必要があります。
    - 再帰的 JOIN で JOIN 名は指定され、タグ名は指定されていない場合、重複フィールド名の前に JOIN 名のはじめの 4 バイトを追加する必要があります。
    - JOIN 名およびタグ名の両方が再帰的 JOIN で指定されている場合、タグ名を修飾子として使用する必要があります。
    - 非再帰的 JOIN の重複フィールド名を検索するためには、タグ名をフィールド名修飾子として使用する必要があります。フィールド名が修飾されていない場合、最初に一致するものが検索されます。
- DEFINE ベースの JOIN (一時項目 (DEFINE) から実フィールドへの結合を参照) を使用して子セグメントの一時項目を親セグメントのフィールドへ結合することができます。
- 構造内の最下位で繰り返されるセグメントから複数の JOIN コマンドを発行することにより、再帰的構造を親セグメントへ拡張し、16 レベルまでの構造を作成することができます。
- FOCUS データソースの場合、結合する親セグメントのフィールドにインデックスを付ける必要があります。

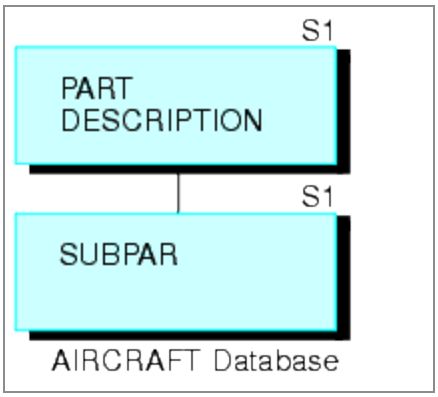
再帰的 JOIN 構造の使用

次の例では、再帰的 JOIN を使用して、パーツリストを保存し、レポートを実行する方法について説明します。航空機のパーツリストを含む「AIRCRAFT」と呼ばれるデータソースを作成することを想定します。データソースには、3 つのレベルの航空機部品が記録されています。

- コックピットや客室などの主要パーツ部門
- 操作パネルや座席などのパーツ部門
- ナットやボルトなどのサブパーツ

データソースは各パーツ、パーツの説明、およびパーツを構成する小パーツを記録する必要があります。ナットやボルトなど、航空機全体で共通のパーツもあります。パーツの各レベルに 1 つのセグメントを持つ 3 セグメント構造を作成する場合、共通使用されるパーツの説明は、使用されるすべての場所で繰り返されます。

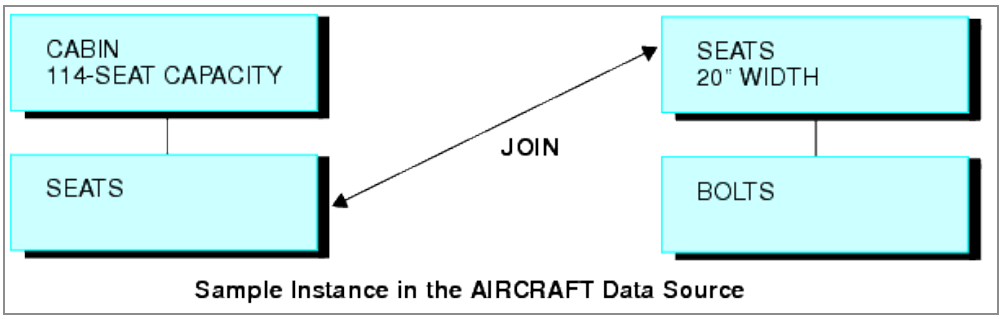
この繰り返しを減らすためには 2 セグメントのみのデータソースを作成します (下図参照)。最上位セグメントは、大小を含む航空機の各パーツの説明です。最下位セグメントは、説明のないコンポーネントパーツリストです。



(最大部門を除く) 各パーツは、説明のある最上位セグメントと、大きいパーツのコンポーネントリストの 1 つである最下位セグメントの両方に表示されます。ナットやボルトなどの最小パーツは、最上位セグメントに表示されますが、最下位セグメントの子インスタンスは持ちません。航空機での使用頻度に関わらず、各パーツは 1 回のみ記述されています。

最下位セグメントを最上位セグメントに結合すると、最下位セグメントのコンポーネントパーツの説明を検索することができます。第 1 レベルの主要パーツ部門は、第 1 から第 2、第 3 レベルへと移動し、第 3 レベルの小パーツに関連することもあります。そのため、この構造は実際にはより効率のよい 2 レベルデータソースでも、3 レベルデータソースのように動作します。

たとえば、CABIN は最上位セグメントに表示される第 1 レベルの部門です。ここで、SEATS は最下位セグメントのコンポーネントとして表示されます。SEATS は最上位セグメントにも存在し、BOLTS は最下位セグメントのコンポーネントとして表示されます。最下位セグメントを最上位セグメントに結合すると、CABIN から SEATS へ、および SEATS から BOLTS へと移動することができます。



Sample Instance in the AIRCRAFT Data Source

JOIN コマンドにより、最下位セグメントを最上位セグメントに結合します。

```
JOIN SUBPART IN AIRCRAFT TO PART IN AIRCRAFT AS SUB
```

これにより、次の再帰的構造が作成されます。

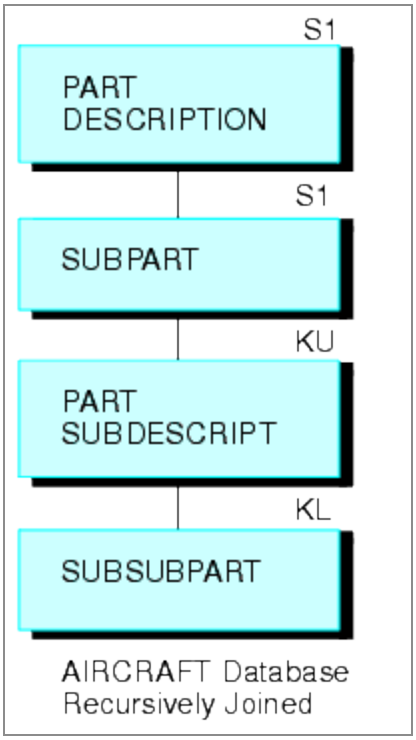


TABLE コマンド によりデータの 3 つのすべてのレベルでレポートを作成します。
SUBDESCRIPT フィールドは SUBPART フィールドの内容の説明です。

```
TABLE FILE AIRCRAFT
PRINT SUBPART BY PART BY SUBPART BY SUBDESCRIPT
END
```

JOIN コマンドの動作原理

JOIN コマンドにより、単一リクエストで複数の関連データソースからレポートを実行することが可能になります。結合されたデータソースは物理的には別ですが、単一データソースとして処理されます。同時に 1022 までの JOIN を有効にできます。

2 つのデータソースを結合するとき、そのうち 1 つは「ホストファイル」と呼ばれ、もう 1 つは「クロスリファレンスファイル」と呼ばれます。ホストファイルからレコードを検索するたびに、クロスリファレンスファイルの関連フィールドが、リクエストで参照された場合に、識別されます。次に、対応する値を含むクロスリファレンスファイルのレコードが検索されます。

ホストファイルのレコードをクロスリファレンスファイルに関連付ける式を定義できるときは、条件付き JOIN を使用して 2 つのデータソースを結合することができます。各データソースに共通のフォーマット (文字、数値、または日付)、および値のフィールドがある場合、EQUIJOIN 構造を使用して 2 つのデータソースを結合することができます。この共通フォーマットにより、値が正しく解釈されます。たとえば、2 つのデータソースから値を読み取る必要があることを想定します。1 つ目は仕事情報を含む「JOB」という名前で、2 つ目は給与情報を含む「SALARY」という名前です。それぞれに同一の従業員グループを同一の方法 (姓、シリアル番号、ソーシャルセキュリティ番号) で識別するフィールドがある場合、これらの 2 つのデータソースを結合することができます。JOIN フィールドに共通値 (たとえば共通のソーシャルセキュリティ番号) を検索するときに JOIN が効果的です。

JOIN コマンドの発行後、単一の TABLE、TABLEF、MATCH FILE、または GRAPH リクエストを発行して JOIN データソースを読み取ることができます。複数のデータソースからレポートを作成するためには、1 つ目のデータソース (ホストファイル) のみを指定する必要があります。たとえば、JOB および SALARY データソースからレポートを作成することを想定します。次の EQUIJOIN を実行します。

```
JOIN EMP_ID IN JOB TO ALL EMP_ID IN SALARY
```

このコマンドにより、JOB ファイルの EMP_ID フィールドを SALARY ファイルの EMP_ID フィールドに結合します。JOB がホストファイルで SALARY がクロスリファレンスファイルです。次に、このリクエストを実行します。

```
TABLE FILE JOB
PRINT SALARY AND JOB_TITLE BY EMP_ID
END
```

1 つ目の検索レコードは従業員 #071382660 の JOB ファイルレコードです。次に、従業員 #071382660 を含む SALARY データソースのすべてのレコードが検索されます。このプロセスはすべてのレコードが読み取られるまで続きます。

JOIN を構成できるものは次のとおりです。

- ホストおよびクロスリファレンスデータソースの、それぞれのマスターファイルで宣言した実フィールド。詳細は、実フィールドの結合を参照してください。
- マスターファイルと DEFINE コマンドのいずれかで定義したホストデータソースの一時項目、およびクロスリファレンスデータソースのマスターファイルで宣言した実フィールド。詳細は、一時項目 (DEFINE) から実フィールドへの結合を参照してください。
- JOIN コマンド自体で指定する条件。詳細は、条件付き JOIN の作成を参照してください。

JOIN データソースの検索の高速化

外部インデックスを使用することにより、JOIN 構造での検索を高速にすることができます。ただし、インデックスのターゲットセグメントをクロスリファレンスセグメントにすることはできません。詳細は、パフォーマンスの改善を参照してください。

EQUIJOIN の作成

最も一般的な JOIN 構造は、ホストおよびクロスリファレンスデータソースのそれぞれのマスターファイルで宣言した実フィールドに基づいています。

実フィールドの結合

次の JOIN 構文では、ファイルを結合するためのフィールドは、マスターファイルで宣言された実フィールドにする必要があります。この JOIN は結合する各ファイルの 1 つのフィールドに基づく単純 JOIN、またはこの動作タイプをサポートするデータソースの複数フィールドの JOIN にすることができます。次の構文は、単純かつ複数のフィールドバリエーションを記述します。

```
JOIN [LEFT_OUTER|RIGHT_OUTER|INNER] hfld1 [AND hfld2 ...] IN hostfile [TAG
tag1]
     TO [UNIQUE|MULTIPLE]
```

```
     crfield [AND crfld2 ...] IN crfile [TAG tag2] [AS joinname]
END
```

説明

JOIN hfld1

クロスリファレンスファイルのフィールドと共有する値を含むホストファイルのフィールド名です。このフィールドは「ホストフィールド」と呼ばれます。

AND hfld2...

ホストファイルの追加のフィールドです。注意点は次のとおりです。複数フィールドを指定する場合、AND で始まる句が必要です。

- 複数フィールドおよび連結 JOIN をサポートするアダプタ、および FOCUS または XFOCUS データソース (SET NFOC=ON (デフォルト設定) の場合) では、最大で 128 個までのフィールドを指定することができます。サポートされる JOIN 機能についての詳細は、使用するデータアダプタのマニュアルを参照してください。
- 2 つの FOCUS データソースの結合および SET NFOC=OFF を実行する場合、ホストファイルの文字フィールドを最大 4 つまで指定することができ、これらが連結されている場合、クロスリファレンスファイルと共有する値も含められます。ファイル接尾語 (SUFFIX) が FOC のときは、クロスリファレンスファイルの複数フィールドを指定することはできません。たとえば、クロスリファレンスファイルに市外局番接頭語交換フォーマット (area code-prefix-exchange format) が含まれており、ホストファイルには、市外局番フィールド、接頭語フィールド、および交換フィールドがあることを想定します。クロスリファレンスファイルにこれら 3 つのフィールドを指定することで、結合した電話番号フィールドを作成することができます。JOIN コマンドはこれら 3 つのフィールドを 1 つのフィールドとして扱います。クロスリファレンスファイルに別のデータソースを使用する場合、この制限は適用されません。

INNER

INNER JOIN を指定します。JOIN コマンドで JOIN タイプを指定しない場合は、ALL パラメータ設定で指定した JOIN タイプが実行されます。

LEFT_OUTER

LEFT OUTER JOIN を指定します。JOIN コマンドで JOIN タイプを指定しない場合は、ALL パラメータ設定で指定した JOIN タイプが実行されます。

LEFT OUTER JOIN では、ミッシングクロスリファレンスインスタンスが存在するホストレコードは、レポート出力に含まれます。ミッシングクロスリファレンスセグメントイン

スタンスに対するテストの処理方法を制御するには、SET SHORTPATH コマンドを使用します。詳細は、ミッシングフィールド値を含むレコード処理を参照してください。

RIGHT_OUTER

RIGHT OUTER JOIN を指定します。このオプションは、このタイプの JOIN をサポートするリレーショナルデータソースで使用可能です。このオプションを使用するには、SET SHORTPATH = SQL コマンドを発行する必要があります。

RIGHT OUTER JOIN では、ミッシングクロスリファレンスインスタンスが存在するホストレコードは、レポート出力に含められます。

IN hostfile

ホストファイル名です。

TAG tag1

ホストファイルのフィールドおよびエイリアスのユニーク修飾子として使用する、66 バイト以内のタグ名です。通常はマスターファイル名です。

ホストファイルのタグ名は、JOIN 構造のすべての JOIN コマンドで同一にする必要があります。

TO [UNIQUE|MULTIPLE] crfld1

hfld1 (または連結されたホストフィールド) に一致する値を含む、クロスリファレンスファイルのフィールド名です。このフィールドは「クロスリファレンスフィールド」と呼ばれます。

注意：UNIQUE は、単一インスタンスのみを返し、クロスリファレンスファイルに一致するインスタンスが存在しない場合、デフォルト値 (文字フィールドはブランク、数値フィールドは 0 (ゼロ)) が指定されます。

crfld1 に *hfld1* 内の 1 つの値と共通の複数インスタンスが存在する可能性がある場合、MULTIPLE パラメータを使用します。ALL は MULTIPLE の同義語で、このパラメータを完全に省略すると、UNIQUE と同義になります。

AND crfld2...

hfld2 と共通の値を持つクロスリファレンスファイルのフィールド名です。

注意：*crfld2* には修飾名を使用することができます。このフィールドは、複数フィールド JOIN をサポートするデータアダプタのみで利用可能です。

IN crfile

クロスリファレンスファイルの名前です。

TAG tag2

クロスリファレンスファイルのフィールドおよびエイリアスのユニーク修飾子として使用する、66 バイト以内のタグ名です。通常はマスターファイル名です。再帰的 JOIN 構造では、タグ名が指定されていないと、すべてのフィールド名およびエイリアスの前に JOIN 名の先頭 4 バイトが付加されます。詳細は、再帰的 JOIN 構造使用上の注意を参照してください。

ホストファイルのタグ名は、JOIN 構造のすべての JOIN コマンドで同一にする必要があります。

AS joinname

オプションで、JOIN 構造に指定する 8 バイト以内の名前です。次のような場合、JOIN 構造に一意の名前を付ける必要があります。

- 後から実行する JOIN コマンドによる上書きを回避する。
- 後で特定の JOIN をクリアする。
- 再帰的構造を持つ。詳細は、再帰的 JOIN 構造を参照してください。

注意：AS 句を使用して JOIN 構造に名前を指定しない場合、名前はブランクになります。JOIN に名前を付けないと、名前のない既存の JOIN が上書きされます。

END

JOIN コマンドの長さが 1 行を超える場合に必要です。END はコマンドを終了します。END 句は、独立した行に記述する必要があります。

単純ユニーク JOIN 構造の作成

単純ユニーク JOIN の例は次のとおりです。

```
JOIN JOBCODE IN EMPLOYEE TO JOBCODE IN JOBFILE AS JJOIN
```

INNER JOIN の作成

次のプロシジャは 3 つの FOCUS データソースを作成します。

- EMPINFO - EMPLOYEE データソースの EMPINFO セグメントには、フィールド EMP_

ID、LAST_NAME、FIRST_NAME、CURR_JOBCODE が含まれています。

- JOBINFO - JOBFILE データソースには、フィールドJOBCODE および JOB_DESC が含まれています。
- EDINFO - EDUCFILE データソースには、フィールド EMP_ID、COURSE_CODE、COURSE_NAME が含まれています。

次のプロシジャでは、JOBINFO および EDINFO データソースに一致するレコードが存在しない「Fred Newman」という従業員を EMPINFO に追加します。

```
TABLE FILE EMPLOYEE
SUM LAST_NAME FIRST_NAME CURR_JOBCODE
BY EMP_ID
ON TABLE HOLD AS EMPINFO FORMAT FOCUS INDEX EMP_ID CURR_JOBCODE
END
-RUN

TABLE FILE JOBFILE
SUM JOB_DESC
BY JOBCODE
ON TABLE HOLD AS JOBINFO FORMAT FOCUS INDEX JOBCODE
END
-RUN

TABLE FILE EDUCFILE
SUM COURSE_CODE COURSE_NAME
BY EMP_ID
ON TABLE HOLD AS EDINFO FORMAT FOCUS INDEX EMP_ID
END
-RUN

MODIFY FILE EMPINFO
FREEFORM EMP_ID LAST_NAME FIRST_NAME CURR_JOBCODE
MATCH EMP_ID
ON NOMATCH INCLUDE
ON MATCH REJECT
DATA
111111111, NEWMAN, FRED, C07,$
END
```

次のリクエストは EMPINFO の内容を出力します。データソースに Fred Newman が追加されます。

```
TABLE FILE EMPINFO
PRINT *
END
```

出力結果は次のとおりです。

```
EMP_ID     LAST_NAME        FIRST_NAME  CURR_JOBCODE
------     ---------        ----------  ------------
071382660  STEVENS          ALFRED      A07
112847612  SMITH            MARY        B14
117593129  JONES            DIANE       B03
119265415  SMITH            RICHARD     A01
119329144  BANNING          JOHN        A17
123764317  IRVING           JOAN        A15
126724188  ROMANS           ANTHONY     B04
219984371  MCCOY            JOHN        B02
326179357  BLACKWOOD        ROSEMARIE   B04
451123478  MCKNIGHT         ROGER       B02
543729165  GREENSPAN        MARY        A07
818692173  CROSS            BARBARA     A17
111111111  NEWMAN           FRED        C07
```

次の JOIN コマンドは、EMPINFO データソースと JOBINFO データソース間の INNER JOIN を実行します。

```
JOIN CLEAR *
JOIN INNER CURR_JOBCODE IN EMPINFO TO MULTIPLE JOBCODE IN JOBINFO AS J0
```

この JOIN コマンドは、複数の JOIN を指定しています。ユニーク JOIN では、クロスリファレンスセグメントはミッシングとして扱われることはありません。ホストファイルのすべてのレコードがリクエストの出力結果に表示されています。実データが存在しない場合、デフォルト値 (文字フィールドではブランク、数値フィールドでは 0 (ゼロ)) が表示されます。

次のリクエストは、JOIN 構造のフィールドを表示します。

```
TABLE FILE EMPINFO
PRINT LAST_NAME FIRST_NAME JOB_DESC
END
```

FRED NEWMAN はレポート出力から除外されています。これは、JOBINFO データソースに一致するジョブコード (JOB CODE) が存在しなかったためです。

```
LAST_NAME  FIRST_NAME  JOB_DESC
---------  ----------  --------
STEVENS    ALFRED      SECRETARY
SMITH      MARY        FILE QUALITY
JONES      DIANE       PROGRAMMER ANALYST
SMITH      RICHARD     PRODUCTION CLERK
BANNING    JOHN        DEPARTMENT MANAGER
IRVING     JOAN        ASSIST.MANAGER
ROMANS     ANTHONY     SYSTEMS ANALYST
MCCOY      JOHN        PROGRAMMER
BLACKWOOD  ROSEMARIE   SYSTEMS ANALYST
MCKNIGHT   ROGER       PROGRAMMER
GREENSPAN  MARY        SECRETARY
CROSS      BARBARA     DEPARTMENT MANAGER
```

LEFT OUTER JOIN の作成

次の JOIN コマンドは EMPINFO データソースと EDINFO データソース間に LEFT OUTER JOIN を作成します。

```
JOIN CLEAR *
JOIN LEFT_OUTER EMP_ID IN EMPINFO TO MULTIPLE EMP_ID IN EDINFO AS J1
```

次のリクエストは、JOIN 構造のフィールドを表示します。

```
TABLE FILE EMPINFO
PRINT LAST_NAME FIRST_NAME COURSE_NAME
END
```

レポート出力には、すべての従業員レコードが表示されます。EDINFO データソースに一致するレコードが存在しない従業員レコードの COURSE_NAME フィールドには、ミッシングデータ文字 (.) が表示されます。JOIN がユニークである場合、ミッシングデータ文字 (.) の代わりにブランクが表示されます。

```
LAST_NAME  FIRST_NAME  COURSE_NAME
---------  ----------  -----------
STEVENS    ALFRED      FILE DESCRPT & MAINT
SMITH      MARY        BASIC REPORT PREP FOR PROG
JONES      DIANE       FOCUS INTERNALS
SMITH      RICHARD     BASIC RPT NON-DP MGRS
BANNING    JOHN        .
```

```
IRVING     JOAN        .
ROMANS     ANTHONY     .
MCCOY      JOHN        .
BLACKWOOD  ROSEMARIE   DECISION SUPPORT WORKSHOP
MCKNIGHT   ROGER       FILE DESCRPT & MAINT
GREENSPAN  MARY        .
CROSS      BARBARA     HOST LANGUAGE INTERFACE
NEWMAN     FRED        .
```

RIGHT OUTER JOIN の作成

次のリクエストでは、結合する 2 つの Microsoft SQL Server テーブルを生成した後、その JOIN に対してリクエストを発行します。

次のリクエストは、WF_SALES テーブルを生成します。ID_PRODUCT フィールドは、RIGHT OUTER JOIN コマンドで使用されます。

```
TABLE FILE WF_RETAIL_LITE
SUM GROSS_PROFIT_US PRODUCT_CATEGORY PRODUCT_SUBCATEG
BY ID_PRODUCT
WHERE ID_PRODUCT FROM 2150 TO 4000
ON TABLE HOLD AS WF_SALES FORMAT SQLMSS
END
```

次のリクエストは、WF_PRODUCT テーブルを生成します。ID_PRODUCT フィールドは、RIGHT OUTER JOIN コマンドで使用されます。

```
TABLE FILE WF_RETAIL_LITE
SUM PRICE_DOLLARS PRODUCT_CATEGORY PRODUCT_SUBCATEG PRODUCT_NAME
BY ID_PRODUCT
WHERE ID_PRODUCT FROM 3000 TO 5000
ON TABLE HOLD AS WF_PRODUCT FORMAT SQLMSS
END
```

次のリクエストは、SET SHORTPATH=SQL コマンドおよび JOIN コマンドを発行し、結合済みテーブルの値を表示します。

```
SET SHORTPATH = SQL
JOIN RIGHT_OUTER ID_PRODUCT IN WF_PRODUCT TAG T1 TO ALL ID_PRODUCT IN
WF_SALES TAG T2
END
```

```
TABLE FILE WF_PRODUCT
PRINT T1.ID_PRODUCT AS 'Product ID'
PRICE_DOLLARS AS Price
T2.ID_PRODUCT AS 'Sales ID'
GROSS_PROFIT_US
BY T1.ID_PRODUCT NOPRINT
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

次のコマンドを追加することで、結果の SQL を表示するトレースを生成することができます。

```
SET TRACEUSER=ON
SET TRACESTAMP=OFF
SET TRACEOFF=ALL
SET TRACEON = STMTRACE//CLIENT
```

トレースを表示すると、リクエストが RDBMS への RIGHT OUTER JOIN として最適化されたことが分かります。

```
SELECT
T1."ID_PRODUCT",
T1."PRICE_DOLLARS",
T2."ID_PRODUCT",
T2."GROSS_PROFIT_US"
FROM
( WF_PRODUCT T1
RIGHT OUTER JOIN
WF_SALES T2
ON T2."ID_PRODUCT" = T1."ID_PRODUCT" )
ORDER BY
T1."ID_PRODUCT";
```

下図のように、出力では WF_PRODUCT テーブルの各 ID_PRODUCT 値行が表示されています。WF_SALES 行の列では、一致する ID_PRODUCT 値が存在しない場合に NODATA 記号が表示されています。

Product ID	Price	Sales ID	Gross Profit
.	.	2151	$7,172.24
.	.	2152	$6,766.45
.	.	2153	$7,291.10
.	.	2154	$19,913.65
.	.	2155	$12,472.93
.	.	2156	$14,484.75
.	.	2157	$6,223.34
.	.	2158	$11,267.75
.	.	2159	$9,015.75
.	.	2160	$7,194.15
.	.	2161	$8,683.00
.	.	2162	$9,197.81
.	.	2163	$21,347.98
.	.	2164	$5,615.33
.	.	2165	$8,540.53
.	.	2166	$12,469.45
.	.	2167	$6,771.26
3001	47,988.00	3001	$12,786.30
3002	34,990.00	3002	$5,586.10
3003	30,591.00	3003	$12,399.15
3004	44,187.00	3004	$8,434.95
3005	30,057.00	3005	$8,987.80
3006	25,348.05	3006	$6,422.59
3007	23,607.00	3007	$8,354.60
3008	42,898.57	3008	$18,469.01
3009	48,139.00	3009	$22,542.75
3010	44,850.00	3010	$16,722.15
3011	28,078.44	3011	$18,204.82
3012	31,146.00	3012	$19,739.45
3013	27,208.00	3013	$17,983.00
3014	23,968.59	3014	$19,996.09
3015	29,061.45	3015	$17,420.72

マルチパス構造の 2 つの INNER JOIN の作成

次の JOIN コマンドは、EMPINFO と JOBINFO のデータソース間に INNER JOIN を作成し、EMPINFO と EDINFO のデータソース間に INNER JOIN を作成します。

```
JOIN CLEAR *
JOIN INNER CURR_JOBCODE IN EMPINFO TO MULTIPLE JOBCODE IN JOBINFO AS J0
JOIN INNER EMP_ID IN EMPINFO TO MULTIPLE EMP_ID IN EDINFO AS J1
```

2 つの JOIN により作成された構造には、2 つの独立したパスが存在します。

```
          SEG01
 01       S1
**************
*EMP_ID      **I
*CURR_JOBCODE**I
*LAST_NAME   **
```

```
*FIRST_NAME  **
*            **
***************
 **************
       I
       +-----------------+
       I                 I
       I SEG01           I SEG01
 02    I KM        03    I KM
.............    .............
:EMP_ID      ::K  :JOBCODE     ::K
:COURSE_CODE ::   :JOB_DESC    ::
:COURSE_NAME ::   :            ::
:            ::   :            ::
:            ::   :            ::
:............::   :............::
 ............:    ............:
 JOINED  EDINFO    JOINED  JOBINFO
```

次のリクエストは、JOIN 構造のフィールドを表示します。

```
SET MULTIPATH=SIMPLE
TABLE FILE EMPINFO
PRINT LAST_NAME FIRST_NAME IN 12 COURSE_NAME JOB_DESC
END
```

MULTIPATH=SIMPLE により、独立したパスによる独立 JOIN が作成されます。いずれかの JOIN により受容されたすべての従業員レコードが、レポート出力に表示されます。Fred Newman には、クロスリファレンスファイルのどちらにも一致するレコードがないため、このレコードのみが除外されています。

```
LAST_NAME  FIRST_NAME  COURSE_NAME                    JOB_DESC
---------  ----------  -----------                    --------
STEVENS    ALFRED      FILE DESCRPT & MAINT           SECRETARY
SMITH      MARY        BASIC REPORT PREP FOR PROG     FILE QUALITY
JONES      DIANE       FOCUS INTERNALS                PROGRAMMER ANALYST
SMITH      RICHARD     BASIC RPT NON-DP MGRS          PRODUCTION CLERK
BANNING    JOHN        .                              DEPARTMENT MANAGER
IRVING     JOAN        .                              ASSIST.MANAGER
ROMANS     ANTHONY     .                              SYSTEMS ANALYST
MCCOY      JOHN        .                              PROGRAMMER
BLACKWOOD  ROSEMARIE   DECISION SUPPORT WORKSHOP      SYSTEMS ANALYST
MCKNIGHT   ROGER       FILE DESCRPT & MAINT           PROGRAMMER
GREENSPAN  MARY        .                              SECRETARY
CROSS      BARBARA     HOST LANGUAGE INTERFACE        DEPARTMENT MANAGER
```

MULTIPATH=COMPOUND により、クロスリファレンスファイルの両方に一致するレコードが存在する従業員のみを、レポート出力に表示することができます。

```
LAST_NAME  FIRST_NAME  COURSE_NAME                    JOB_DESC
---------  ----------  -----------                    --------
STEVENS    ALFRED      FILE DESCRPT & MAINT           SECRETARY
SMITH      MARY        BASIC REPORT PREP FOR PROG     FILE QUALITY
JONES      DIANE       FOCUS INTERNALS                PROGRAMMER ANALYST
SMITH      RICHARD     BASIC RPT NON-DP MGRS          PRODUCTION CLERK
BLACKWOOD  ROSEMARIE   DECISION SUPPORT WORKSHOP      SYSTEMS ANALYST
MCKNIGHT   ROGER       FILE DESCRPT & MAINT           PROGRAMMER
CROSS      BARBARA     HOST LANGUAGE INTERFACE        DEPARTMENT MANAGER
```

EQUIJOIN のクロスリファレンスフィールドの要件

JOIN に使用するクロスリファレンスファイルは、それぞれのデータソースで次の特性を持つ必要があります。

- リレーショナルデータソースおよび CA-DATACOM/DB データソースでは、クロスリファレンスフィールドは任意のフィールドにすることができます。
- FOCUS および XFOCUS データソースでは、クロスリファレンスフィールドにインデックスを含める必要があります。インデックスフィールドは、マスターファイルで FIELDTYPE=I、INDEX=I、または INDEX=ON 属性が指定されたフィールドです。クロスリファレンスフィールドにこの属性が指定されていない場合は、この属性をマスターファイルのフィールド宣言に追加し、INDEX オプションを含めた REBUILD ユーティリティを使用してファイルを再構築します。これにより、FOCUS または XFOCUS データソースにインデックスが追加されます。

  注意：インデックスフィールドは外部に存在させることができます。
- 固定フォーマットシーケンシャルファイルまたは区切り付きシーケンシャルファイルでは、任意のフィールドをクロスリファレンスフィールドにすることができます。ただし、ENGINE INT CACHE SET OFF が有効な場合、ホストファイルとクロスリファレンスファイルの両方を、指定した (キー) フィールドの昇順で取得する必要があります。この場合、データのソート順序が一致しないとエラーが表示されます。また、n 対 n の JOIN はサポートされません。一方、ENGINE INT CACHE SET ON が有効な場合、両方のファイルを昇順で取得する必要はありません。また、n 対 n の JOIN がサポートされます。デフォルト値は ON です。区切り付きファイルを JOIN のクロスリファレン

スファイルとして使用する場合、単一セグメントのみで構成されたファイルにする必要があります。複数フィールドに基づいた JOIN の場合、固定フォーマットのシーケンシャルファイルは単一セグメントで構成されたファイルにする必要があります。クロスリファレンスの固定フォーマットシーケンシャルファイルが単一セグメントで構成されている場合は、ホストファイルにセグメント宣言を追加する必要があります。

グループフィールドの制限事項

JOIN 構造でグループフィールドを使用する際は、ホストファイルのグループとクロスリファレンスファイルのグループの要素の数を同一にする必要があります。

- ISAM データソースでは、ユニーク JOIN を発行する場合はフィールドを完全な主キーに、非ユニーク JOIN を発行する場合はフィールドを主キーの初期部分にする必要があります。マスターファイルでは、主キーはキー GROUP により記述されます。その初期部分はそのグループの 1 つ目のフィールドです。
- Model 204 データソースでは、フィールドはキーフィールドにする必要があります。アクセスファイルでは、キーフィールドのタイプは文字 (KEY)、オーダー文字 (ORA)、オーダー数値 (ORN)、数値範囲 (RNG)、非表示 (IVK) および非表示範囲 (IVR) です。
- ADABAS データソースでは、フィールドはディスクリプタフィールド、.SPR または .NOP フィールド名の接尾語で定義したスーパーディスクリプタ、または .NOP フィールド名の接尾語で定義したサブディスクリプタにする必要があります。マスターファイルのフィールド記述には FIELDTYPE=I 属性を含める必要があります。

  アクセスファイルでは、クロスリファレンスセグメントにより、ACCESS=ADBS、および CALLTYPE=FIND または CALLTYPE=RL のいずれかを指定する必要があります。CALLTYPE=RL の場合、ディスクリプタ、スーパーディスクリプタまたはサブディスクリプタの上位の部分がホストフィールドより長い場合、その上位の部分にホストフィールドを結合することができます。
- CA-IDMS/DB データソースでは、フィールドはマスターファイルの FIELDTYPE=I 属性により識別されるネットワークレコードのインデックスフィールド、アクセスファイルの CLCFLD 句により識別されるネットワークレコードの CA-IDMS/DB CALC フィールド、または LRF または ASF レコードの任意のフィールドにする必要があります。
- 固定フォーマットまたは区切り付きシーケンシャルファイルを使用する部分キー結合では、ENGINE INT CACHE SET OFF 設定を有効にする必要があります。

INNER JOIN および OUTER JOIN コマンド構文使用上の注意

- SET ALL および SET CARTESIAN コマンドは、この構文により無視されます。
- ALL. パラメータはサポートされていません。ALL. パラメータが使用されている場合、次のメッセージが表示されます。

  ```
  (FOC32452) ALL. は LEFT_OUTER/INNER とともに使用できません。
  ```

- 複数の JOIN を定義する場合、結果は単一パス、またはマルチパスデータソース構造を取ることができます。
  - MULTIPATH=SIMPLE の設定が有効であり、レポートがマルチパスを基準とする場合、JOIN はそれぞれ別に構築されます。レポート出力には、異なるパスに存在する一致レコードが表示されます。これにより、1 つの JOIN のみが有効な場合には除外されるレコードを、結果に含めることができます。
  - マルチパスレポートで MULTIPATH=COMPOUND の設定が有効である場合、またはレポートに単一パス上のデータのみを表示する場合、レポート出力には、すべての JOIN 条件を満たすレコードのみが表示されます。

EQUIJOIN による一時項目 (DEFINE) から実フィールドへの結合

DEFINE に基づく JOIN 構文を使用して、実クロスリファレンスフィールドに結合できる仮想ホストフィールドを作成することができます。仮想ホストフィールドを作成する DEFINE 式には、ホストファイルのフィールドおよび定数のみ含めることができます。クロスリファレンスファイルのフィールドを含めることはできません。一時項目 (DEFINE) から複数の JOIN を実行することができます。

別の DEFINE コマンドまたはマスターファイルで、仮想ホストフィールドを作成することができます。

同一のリクエストに JOIN に基づく一時項目 (DEFINE)、および JOIN と無関係の一時項目 (DEFINE) を使用することができます。

DEFINE コマンドで一時項目を作成するときに、ホストファイルおよび JOIN 構造に対して DEFINE コマンドで作成したすべてのフィールドが JOIN コマンドによりクリアされるため、

JOIN コマンドの後、TABLE リクエストの前に DEFINE を発行する必要があります。マスターファイルで定義した一時項目 (DEFINE) はクリアされません。

ヒント：DEFINE コマンドが JOIN コマンドの前に来る場合、KEEPDEFINES ON または ALL を設定して続きの JOIN コマンドの解析中に一時項目 (DEFINE) を復帰させることができます。詳細は、KEEPDEFINES による一時項目 (DEFINE) の保持を参照してください。

一時項目 (DEFINE) から実フィールドへの結合

DEFINE に基づく JOIN コマンドにより、ホストファイルの一時項目 (DEFINE) をクロスリファレンスファイルの実フィールドへ結合することが可能です。構文は次のとおりです。

```
JOIN [LEFT_OUTER|RIGHT_OUTER|INNER] deffld WITH host_field ...
     IN hostfile [TAG tag1]
     TO [UNIQUE|MULTIPLE]
     cr_field IN crfile [TAG tag2] [AS joinname]
END
```

説明

JOIN deffld

ホストファイル (ホストフィールド) の一時項目 (DEFINE) 名です。一時項目 (DEFINE) は、マスターファイルで定義することも、DEFINE コマンドを使用して定義することもできます。詳細は、JOIN データソースによる一時項目 (DEFINE) 使用上の注意を参照してください。

WITH host_field

一時項目 (DEFINE) に関連付ける、ホストセグメントの任意の実フィールド名です。一時項目 (DEFINE) を検索するためには、関連付けを行います。

注意：JOIN コマンドで参照する WITH フィールドは、一時項目を作成する DEFINE コマンドで参照する WITH フィールドと同一セグメントに存在する必要があります。そうでない場合は、出力が生成されません。

KEEPDEFINES パラメーターが ON または ALL に設定され、JOIN コマンドを発行する前に *deffld* が定義されている場合を除いて、WITH 句が必要です。

一時項目 (DEFINE) を含むセグメントを識別するには、DEFINE コマンドを発行した後に ? DEFINE クエリを使用します。詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

INNER

INNER JOIN を指定します。JOIN コマンドで JOIN タイプを指定しない場合は、ALL パラメータ設定で指定した JOIN タイプが実行されます。

LEFT_OUTER

LEFT OUTER JOIN を指定します。JOIN コマンドで JOIN タイプを指定しない場合は、ALL パラメータ設定で指定した JOIN タイプが実行されます。

RIGHT_OUTER

RIGHT OUTER JOIN を指定します。このオプションは、このタイプの JOIN をサポートするリレーショナルデータソースで使用可能です。RIGHT OUTER JOIN を発行するには、SET SHORTPATH = SQL コマンドを有効にしておく必要があります。

IN hostfile

ホストファイル名です。

TAG tag1

ホストファイルのフィールドおよびエイリアスのユニーク修飾子として使用する、66 バイト以内のタグ名です。通常はマスターファイル名です。

ホストファイルのタグ名は、JOIN 構造のすべての JOIN コマンドで同一にします。

TO [UNIQUE|MULTIPLE] crfld1

一時項目 (DEFINE) と値が一致するクロスリファレンスデータソースの実フィールド名です。マスターファイルで宣言されている実フィールドです。

注意：UNIQUE は、単一インスタンスのみを返し、クロスリファレンスファイルに一致するインスタンスが存在しない場合、デフォルト値 (文字フィールドはブランク、数値フィールドは 0 (ゼロ)) が指定されます。

crfld1 に *hfld1* 内の 1 つの値と共通の複数インスタンスが存在する可能性がある場合、MULTIPLE パラメータを使用します。ALL は MULTIPLE の同義語で、このパラメータを完全に省略すると、UNIQUE と同義になります。

IN crfile

クロスリファレンスファイルの名前です。

TAG tag2

クロスリファレンスファイルのフィールドおよびエイリアスのユニーク修飾子として使用する、66 バイト以内のタグ名です。通常はマスターファイル名です。再帰的 JOIN 構造では、タグ名が指定されていない場合、すべてのフィールド名およびエイリアスの前に JOIN 名の先頭 4 バイトが追加されます。

ホストファイルのタグ名は、JOIN 構造のすべての JOIN コマンドで同一にします。

AS joinname

JOIN 構造に指定する 8 バイト以内の名前です (オプション)。次のような場合、JOIN 構造に一意の名前を付ける必要があります。

- 後から実行する JOIN コマンドによる上書きを回避する。
- 後で特定の JOIN をクリアする。
- 再帰的構造を持ち、タグ名を指定しない場合。詳細は、再帰的 JOIN 構造を参照してください。

JOIN 構造に AS 句により名前を指定しない場合、名前はブランクになります。JOIN に名前を付けないと、名前のない既存の JOIN が上書きされます。

END

JOIN コマンドの長さが 1 行を超える場合に必要です。END はコマンドを終了します。END 句は、独立した行に記述する必要があります。

JOIN データソースによる一時項目 (DEFINE) 使用上の注意

JOIN で結合したデータソースを読み取るリクエストには、次の方法で定義した一時項目 (DEFINE) を含めることができます。

- ホストデータソースのマスターファイル内。
- 構文により JOIN 構造のホストデータソースの定義される DEFINE コマンド内。

  ```
  DEFINE FILE hostfile
  ```

  DEFINE コマンド内。

注意：この JOIN のホストフィールドを定義する式には、ホストフィールドおよび定数のみを使用することができます。

その他すべての一時項目 (DEFINE) に、ホストファイルおよびクロスリファレンスファイルからの実フィールドを含めることができます。

ヒント：JOIN コマンドを発行するとホストファイルおよび JOIN 構造のすべての DEFINE コマンドがクリアされるため、JOIN の後に DEFINE コマンドを発行、または KEEPDEFINES ON または ALL を設定して一時項目 (DEFINE) を保存する必要があります。詳細は、JOIN 解析中の一時項目 (DEFINE) の保持を参照してください。

JOIN 構造の仮想ホストフィールドの作成

小売チェーンが 4 名のマネージャを研修に派遣することを想定します。ID 番号で識別される各個人が異なる都市で店を管理しています。その店およびそれらの位置する都市は SALES データソースに含まれます。マネージャの ID、研修、およびマネージャが出席した日付は EDUCFILE データソースに含まれています。

次のプロシジャは、マネージャの受講したコースを一覧表示し、勤務先都市名を識別します。プロシジャの 3 つの要素に注目します。

- JOIN コマンドにより、EDUCFILE の SALES および EMP_ID フィールドの (マネージャ ID を含む) ID_NUM フィールドに共通する値に基づいて、SALES データソースを EDUCFILE データソースへ結合します。この時時点では、ID_NUM フィールドは、まだ存在せず、DEFINE コマンドにより作成されます。
- DEFINE コマンドにより、ID_NUM フィールドが作成され、4 都市で就業するマネージャの ID がこのフィールドに割り当てられます。
- TABLE コマンドにより、レポートが生成されます。

プロシジャは次のとおりです。

```
JOIN ID_NUM WITH CITY IN SALES TO ALL EMP_ID IN EDUCFILE AS SALEDUC
```

```
DEFINE FILE SALES
ID_NUM/A9 = DECODE CITY ('NEW YORK' 451123478 'NEWARK' 119265415
                         'STAMFORD' 818692173 'UNIONDALE' 112847612);
END
```

```
TABLE FILE SALES
PRINT DATE_ATTEND BY CITY BY COURSE_NAME
END
```

出力結果は次のとおりです。

```
CITY                    COURSE_NAME                DATE_ATTEND
----                    -----------                -----------
NEW YORK                FILE DESCRPT & MAINT          81/11/15
NEWARK                  BASIC RPT NON-DP MGRS         82/08/24
STAMFORD                BASIC REPORT PREP DP MGRS     82/08/02
                        HOST LANGUAGE INTERFACE       82/10/21
UNIONDALE               BASIC REPORT PREP FOR PROG    81/11/16
                        FILE DESCRPT & MAINT          81/11/15
```

EQUIJOIN の JOIN モード

JOIN_LENGTH_MODE (JOINLM) パラメータは、レコードベースで SQL 以外のアダプタ (例、DFIX、FIX) に対して、等価結合フィールドの組み合わせの処理を制御します。この設定は、長さの異なる 2 つの文字フィールド、またはデータタイプと精度の異なる 2 つの数値フィールドを結合する場合の処理を制御します。

SQL データソースの場合、通常、JOIN は最適化 (SQL エンジンに送信して処理) されるか、SQL 処理規則に準拠するよう管理されます。

互換性はあるが同一ではない JOIN フィールドの処理では、次の 2 つのモードがサポートされます。

- **SQL 互換モード**　JOIN コマンドプロセッサは、結合されたフィールドの値の等価性を厳密に保証します。ホストからクロスリファレンスへの変換時に有効な文字の切り捨てが検出された場合、「ターゲットが見つかりません」エラーが生成され、結合は実行されません。結合されるホストフィールドの長さがクロスリファレンスフィールドよりも短い場合、ホストフィールドの値は、データタイプに応じた非有効文字でクロスリファレンスフィールドの長さに拡張された上で、結合処理が実行されます。
- **FOCUS レポートモード**　JOIN コマンドプロセッサは、結合されたフィールドの値の等価性を部分的に保証します。
    - 短いフィールドを長いフィールドに結合する場合は、検索範囲が作成され、ホストの値を前に持つすべてのクロスリファレンスの値を検索可能にします。
    - 長いフィールドを短いフィールドに結合する場合は、クロスリファレンスフィー

ルドの長さに合わせて、ホストの値の末尾が無条件に切り捨てられます。切り捨てにブランク以外の文字が含まれる場合は、一致は完了せず、比較は失敗し、レコードは拒否されます。

レコードベースデータソースの JOIN モードの制御

```
SET JOIN_LENGTH_MODE|JOINLM} = {SQL|RANGE}
```

説明

SQL

SQL 互換モードを設定します。ホストおよびクロスリファレンスフィールド値の厳密な等価性を保証します。これがデフォルト値です。

RANGE

FOCUS レポートモードを設定します。部分キー JOIN をサポートします。

共有フィールドのデータフォーマット

通常、ホストファイルとクロスリファレンスファイルで共有値を持つフィールドには、同一のデータフォーマットが指定されている必要があります。

複数のホストファイルフィールドを指定すると、JOIN コマンドにより複数フィールドが単一の JOIN フィールドとして処理されます。各フィールドフォーマットの長さを加算すると、その連結フィールドの長さが得られます。データフォーマットは次の規則に従う必要があります。

- ホストフィールドが文字フィールドの場合、クロスリファレンスフィールドも文字フィールドにして、長さを同一にする必要があります。

  これらのフォーマットには、異なる編集オプションを指定することができます。

  マスターファイルで FORMAT の値に TX が定義されたフィールドは、データソースを結合するために使用することはできません。

- ホストフィールドが数値フィールドの場合、マスターファイルの USAGE (または

FORMAT) 属性で指定したホストフィールドフォーマットのタイプ (I、P、F または D) は、USAGE (または FORMAT) 属性で指定したクロスリファレンスフィールドのフォーマットのタイプと一致する必要があります。詳細は、異なる数値データタイプを含むフィールドの結合を参照してください。

これらのフォーマットには、異なる編集オプションを指定することができます。また、異なる長さを指定することもできますが、次のような結果になります。

- (USAGE 属性により指定される) ホストフィールドのフォーマットがパック 10 進数 (P) または整数 (I) および (FOCUS データソースの USAGE 属性、または他のデータソースの ACTUAL 属性により指定される) クロスリファレンスフィールドフォーマットより長い場合、短い方のフィールドの桁数のみを使用してクロスリファレンスフィールドフォーマットと比較します。たとえば、5 桁のパック 10 進数フォーマットフィールドが 3 桁のパック 10 進数フォーマットフィールドに結合される場合、5 桁の数のホストレコードが検索されるとき、5 桁の数の最後の 3 桁と一致するすべてのクロスリファレンスレコードも検索されます。
- ホストフィールドのフォーマットが倍精度 (D) の場合、各フィールドの ACTUAL 属性により指定される 8 バイトが比較されます。

- ホストフィールドが日付フィールドのときは、クロスリファレンスフィールドも日付フィールドにする必要があります。日付フィールドおよび日付時間フィールドには同一の構成要素が含まれている必要があります。その際、順序が同じである必要はありません。
- コンポーネントフィールドの数が同一の場合、ホストおよびクロスリファレンスフィールドをマスターファイルのグループとして記述することができます。各グループの関連コンポーネントフィールド (例、ホストグループの 1 つ目のフィールドおよびクロスリファレンスグループの 1 つ目のフィールド) は上記の規則に従う必要があります。詳細は、グループフィールドの制限事項を参照してください。

  ホストフィールドがグループフィールドでない場合も、クロスリファレンスフィールドはグループにすることができます。ホストフィールドがグループの場合は、クロスリファレンスフィールドもグループにする必要があります。

異なる数値データタイプを含むフィールドの結合

異なる数値データタイプを含む複数のデータソースを結合することができます。たとえば、短いパック 10 進数フォーマットのフィールドを長いパック 10 進数フォーマットのフィールドに結合したり、整数フォーマットのフィールドをパック 10 進数フォーマットのフィール

ドに結合したりすることができます。これにより、JOIN 構造のデータソースからレポートを柔軟に作成することが可能になります。

- 短いフィールドを長いフィールドに結合するときは、クロスリファレンス値がホストフィールドの長さに追加され、(ALPHA フィールドに) ブランクまたは (数値フィールドに) 16 進数の 0 (ゼロ) が追加されます。この新しい値はクロスリファレンスファイルの検索に使用されます。
- 長いフィールドを短いフィールドに結合するときは、FROM 値は切り取られます。クロスリファレンスファイルの USAGE の長さのために値の一部が除外される場合、除外された値と一致するレコードのみがクロスリファレンスファイルに格納されます。

注意

- パック 10 進数フィールドでの比較を適切に実行するために、正の値のすべての記号は 16 進の C に、負の値のすべての記号は 16 進の D に変換されます。
- JOINOPT パラメータを使用して、ユニーク JOIN のラギング値を修正することもできます。

データタイプ変換付き JOIN の有効化

データタイプ変換を含む JOIN を有効にするには、次のコマンドを発行します。

```
SET JOINOPT = [GNTINT|OLD]
```

説明

GNTINT

データタイプ変換ありの JOIN を有効にします。

OLD

データタイプ変換ありの JOIN を無効にします。これがデフォルト値です。

データタイプ変換を含む JOIN の発行

短い 10 進数フォーマットのフィールドと長い 10 進数フォーマットのフィールドを結合できるため、次のようにマスターファイルで JOIN を定義することができます。

```
FILE=PACKED,SUFFIX=FIX,$
  SEGNAME=ONE,SEGTYPE=S0
   FIELD=FIRST,,P8,P4,INDEX=I,$
```

```
FILE=PACKED2,SUFFIX=FIX,$
  SEGNAME=ONE,SEGTYPE=S0
   FIELD=PFIRST,,P31,P16,INDEX=I,$
```

JOIN コマンドは次のようになります。

```
JOIN FIRST IN PACKED TO ALL PFIRST IN PACKED2 AS J1
```

パックフィールドを結合する場合でも、正の値の推奨記号フォーマットの X'C' および負の値の X'D' が必要です。その他すべての非推奨記号は X'C' と X'D' のいずれかに変換されます。

条件付き JOIN の使用

条件付き JOIN 構文を使用すると、フィールド間の等価性とは別の条件に基づいて、JOIN を設定することができます。さらに、ホストおよびクロスリファレンス JOIN フィールドに同一フォーマットを含めたり、クロスリファレンスフィールドにインデックスを付ける必要がなくなります。

条件付き JOIN は、FOCUS、ADABAS、IDMS およびすべてのリレーショナルデータソースでサポートされます。各データソースは複雑な WHERE 条件の処理能力において異なるため、条件付き JOIN 構文の最適化は JOIN に関わる特定データソースおよび WHERE 条件の複雑さにより異なります。

標準の ?JOIN コマンドにより、有効な JOIN が一覧表示され、WHERE 条件に基づくすべての JOIN が表示されます。

条件付き JOIN の作成

条件付き (WHERE に基づく) JOIN コマンドの構文は次のとおりです。

```
JOIN [LEFT_OUTER|RIGHT_OUTER|INNER] FILE hostfile AT hfld1     [WITH hfld2]
[TAG tag1]
     TO {UNIQUE|MULTIPLE}
```

```
      FILE crfile AT crfld [TAG tag2] [AS joinname]
      [WHERE expression1;
      [WHERE expression2;
      ...]
END
```

説明

INNER

INNER JOIN を指定します。JOIN コマンドで JOIN タイプを指定しない場合は、ALL パラメータ設定で指定した JOIN タイプが実行されます。

LEFT_OUTER

LEFT OUTER JOIN を指定します。JOIN コマンドで JOIN タイプを指定しない場合は、ALL パラメータ設定で指定した JOIN タイプが実行されます。

RIGHT_OUTER

RIGHT OUTER JOIN を指定します。SET SHORTPATH = SQL コマンドを有効にする必要があります。

hostfile

ホストマスターファイルです。

AT

親セグメント (ホストセグメント) を子セグメント (クロスリファレンスセグメント) に正しく接続します。AT パラメータとして使用されるフィールド値は、リンクには使用されません。これらはセグメントの参照としてのみ使用されます。

hfld1

セグメントをクロスリファレンスデータソースに結合するホストマスターファイルのフィールド名です。フィールド名は、参照するデータソースで最下位のセグメントである必要があります。

tag1

ホストデータソースのフィールドおよびエイリアスのユニーク修飾子として使用する、66 バイト以内のタグ名です (オプション)。通常はマスターファイル名です。

hfld2

DEFINE に基づく条件付き JOIN に関連付けるデータソースフィールドです。DEFINE ベースの条件付き JOIN の場合、KEEPDEFINES 設定を ON または ALL にする必要があり、JOIN コマンドを発行する前に、一時項目 (DEFINE) を作成する必要があります。

MULTIPLE

hostfile と crfile の関係として 1 対 n を指定します。ALL は MULTIPLE の同義語です。

UNIQUE

hostfile と crfile の関係として 1 対 1 を指定します。ONE は UNIQUE の同義語です。

注意：JOIN のタイプ (INNER または LEFT OUTER) に関係なく、UNIQUE はクロスリファレンスファイルの単一インスタンスのみを返し、クロスリファレンスファイルに一致するインスタンスが存在しない場合、デフォルト値 (文字フィールドはブランク、数値フィールドは 0 (ゼロ)) が表示されます。クロスリファレンスファイルには、ショートパスやミッシング値が存在することはありません。

crfile

クロスリファレンスマスターファイルです。

crfld

クロスリファレンスマスターファイルの JOIN フィールドの名前です。セグメント内の任意のフィールドを指定することができます。

tag2

クロスリファレンスデータソースのフィールドおよびエイリアスのユニーク修飾子として使用する、66 バイト以内のタグ名です (オプション)。通常はマスターファイル名です。

joinname

JOIN 構造に関連付けられた名前です。

expression1, expression2

任意の条件を指定します。式に含めるフィールドはすべて同一パス内に存在する必要があります。

WHERE 条件に、テーブル間の接続を含める必要があります。標準の JOIN と同様に、実際には、AT 参照はフィールド間の JOIN を実行しません。

JOIN に WHERE 条件を何も含めない場合、直積が生成されます。

END

コマンドを終了するには、END 句が必要です。この句は、独立した行に記述する必要があります。

注意：単一行の JOIN 構文はサポートされていません。

条件付き JOIN の使用

次の条件で、VIDEOTRK および MOVIES データソースを結合します。

- 取り扱い日 (VIDEOTRK) が、リリース日 (MOVIE) から 10 年を超えている。
- 両方のデータソースで映画コードが一致する。

JOIN は、参照する最下位セグメントで発生する必要があるため、VIDEOTRK データソースの MOVIECODE を含むセグメントで実行されます。

次のリクエストは、実行した JOIN でタイトル、最近の取り扱い日、各映画のリリース日を表示し、取り扱い日とリリース日の年数の差を計算します。

```
JOIN FILE VIDEOTRK AT MOVIECODE TAG V1 TO ALL
     FILE MOVIES   AT RELDATE   TAG M1 AS JW1
  WHERE DATEDIF(RELDATE, TRANSDATE,'Y') GT 10;
  WHERE V1.MOVIECODE EQ M1.MOVIECODE;
END
TABLE FILE VIDEOTRK
 SUM TITLE/A25 AS 'Title'
     TRANSDATE AS 'Last,Transaction'
     RELDATE AS 'Release,Date'
 COMPUTE YEARS/I5 = (TRANSDATE - RELDATE)/365;  AS 'Years,Difference'
 BY TITLE NOPRINT
 BY HIGHEST 1 TRANSDATE NOPRINT
END
```

出力結果は次のとおりです。

```
                         Last         Release   Years
Title                    Transaction  Date      Difference
-----                    -----------  -------   ----------
ALICE IN WONDERLAND      91/06/22     51/07/21          39
ALIEN                    91/06/18     80/04/04          11
ALL THAT JAZZ            91/06/25     80/05/11          11
ANNIE HALL               91/06/24     78/04/16          13
BAMBI                    91/06/22     42/07/03          49
```

```
BIRDS, THE                91/06/23     63/09/27         27
CABARET                   91/06/25     73/07/14         17
CASABLANCA                91/06/27     42/03/28         49
CITIZEN KANE              91/06/22     41/08/11         49
CYRANO DE BERGERAC        91/06/20     50/11/09         40
DEATH IN VENICE           91/06/26     73/07/27         17
DOG DAY AFTERNOON         91/06/23     76/04/04         15
EAST OF EDEN              91/06/20     55/01/12         36
GONE WITH THE WIND        91/06/24     39/06/04         52
JAWS                      91/06/27     78/05/13         13
MALTESE FALCON, THE       91/06/19     41/11/14         49
MARTY                     91/06/19     55/10/26         35
NORTH BY NORTHWEST        91/06/21     59/02/09         32
ON THE WATERFRONT         91/06/24     54/07/06         36
PHILADELPHIA STORY, THE   91/06/21     40/05/06         51
PSYCHO                    91/06/17     60/05/16         31
REAR WINDOW               91/06/17     54/12/15         36
SHAGGY DOG, THE           91/06/25     59/01/09         32
SLEEPING BEAUTY           91/06/24     75/08/30         15
TIN DRUM, THE             91/06/17     80/03/01         11
VERTIGO                   91/06/27     58/11/25         32
```

FULL OUTER JOIN

WebFOCUS の JOIN コマンドおよび条件付き JOIN コマンドには、FULL OUTER JOIN オプションがあります。

FULL OUTER JOIN を使用すると、ホストデータソースのすべての行と、ターゲットデータソースのすべての行が返されます。いずれかのデータソースの行に値が存在しない場合は、NULL 値が返されます。WebFOCUS レポート出力では、これらの値はデフォルト値に置き換えられ、文字フィールドではブランク、数値フィールドでは NODATA 記号になります。

FULL OUTER JOIN および RIGHT OUTER JOIN は、使用するデータソースでのサポートの有無に関係なく使用することができます。使用するデータソースがこれらの JOIN をサポートする場合は、JOIN の処理がデータベースエンジンに渡されます。サポートしない場合は、必要なデータがすべて返され、JOIN は WebFOCUS で処理されます。

注意：FULL OUTER JOIN を発行するには、SET SHORTPATH = SQL コマンドを有効にしておく必要があります。

FULL OUTER JOIN の指定

次の構文は、実フィールドに基づいて FULL OUTER JOIN (等価 JOIN) を生成します。

```
JOIN FULL_OUTER hfld1 [AND hfld2 ...] IN table1 [TAG tag1] TO
{UNIQUE|MULTIPLE} cfld [AND cfld2 ...] IN table2 [TAG tag2] [AS joinname]
END
```

説明

hfld1

クロスリファレンステーブルのフィールドと共有する値を含むホストテーブルのフィールド名です。このフィールドは「ホストフィールド」と呼ばれます。

AND hfld2...

ホストテーブルの追加フィールドを指定することができます。複数フィールドを指定する場合、AND で始まる句が必要です。

- 複数フィールドおよび連結 JOIN をサポートするリレーショナルアダプタでは、最大 20 個のフィールドを指定することができます。サポートされている JOIN 機能についての詳細は、使用するアダプタのマニュアルを参照してください。

IN table1

ホストテーブルの名前です。

TAG tag1

ホストテーブルのフィールドおよびエイリアスのユニーク修飾子として使用する、66 バイト以内のタグ名です。通常はマスターファイル名です。

ホストテーブルのタグ名は、JOIN 構造のすべての JOIN コマンドで同一にする必要があります。

TO [UNIQUE|MULTIPLE] crfld1

hfld1 (または連結されたホストフィールド) に一致する値を含む、クロスリファレンステーブルのフィールド名です。このフィールドは「クロスリファレンスフィールド」と呼ばれます。

注意：UNIQUE は 1 つのインスタンスのみを返し、一致するインスタンスがクロスリファレンステーブルに存在しない場合、NULL 値を返します。

crfld1 に *hfld1* 内の 1 つの値と共通の複数インスタンスが存在する可能性がある場合、MULTIPLE パラメータを使用します。ALL は MULTIPLE の同義語で、このパラメータを完全に省略すると、UNIQUE と同義になります。

AND crfld2...

hfld2 と共通の値を持つクロスリファレンステーブルのフィールド名です。

注意：*crfld2* には修飾名を使用することができます。このフィールドは、複数フィールド JOIN をサポートするアダプタのみで利用可能です。

IN crfile

クロスリファレンステーブルの名前です。

TAG tag2

クロスリファレンステーブルのフィールドおよびエイリアスのユニーク修飾子として使用する、66 バイト以内のタグ名です。通常はマスターファイル名です。再帰的 JOIN 構造では、タグ名が指定されていないと、すべてのフィールド名およびエイリアスの前に JOIN 名の先頭 4 バイトが付加されます。

ホストテーブルのタグ名は、JOIN 構造のすべての JOIN コマンドで同一にする必要があります。

AS joinname

オプションで、JOIN 構造に指定する 8 バイト以内の名前です。次のような場合、JOIN 構造に一意の名前を付ける必要があります。

- 後から実行する JOIN コマンドによる上書きを回避する。
- 後で特定の JOIN をクリアする。
- 再帰的構造を持つ。

注意：AS 句を使用して JOIN 構造に名前を指定しない場合、名前はブランクになります。JOIN に名前を付けないと、名前のない既存の JOIN が上書きされます。

END

JOIN コマンドの長さが 1 行を超える場合に必要です。この句により、コマンドが終了します。この句は、独立した行に記述する必要があります。

次の構文は、DEFINE ベースの FULL OUTER JOIN を生成します。

```
JOIN FULL_OUTER deffld WITH host_field ...
     IN table1 [TAG tag1]
```

```
      TO [UNIQUE|MULTIPLE]
      cr_field IN table2 [TAG tag2] [AS joinname]
END
```

説明

deffld

ホストファイル (ホストフィールド) の一時項目 (DEFINE) 名です。一時項目 (DEFINE) は、マスターファイルで定義することも、DEFINE コマンドを使用して定義することもできます。

WITH host_field

一時項目 (DEFINE) に関連付ける、ホストセグメントの任意の実フィールド名です。一時項目 (DEFINE) を検索するためには、関連付けを行います。

KEEPDEFINES パラメーターが ON または ALL に設定され、JOIN コマンドを発行する前に *deffld* が定義されている場合を除いて、WITH 句が必要です。

一時項目 (DEFINE) を含むセグメントを識別するには、DEFINE コマンドを発行した後に ? DEFINE クエリを使用します。

IN table1

ホストテーブルの名前です。

TAG tag1

ホストテーブルのフィールドおよびエイリアスのユニーク修飾子として使用する、66 バイト以内のタグ名です。通常はマスターファイル名です。

ホストテーブルのタグ名は、JOIN 構造のすべての JOIN コマンドで同一にする必要があります。

TO [UNIQUE|MULTIPLE] crfld1

一時項目 (DEFINE) と値が一致するクロスリファレンステーブルの実フィールド名です。マスターファイルで宣言されている実フィールドです。

注意：UNIQUE は 1 つのインスタンスのみを返し、一致するインスタンスがクロスリファレンステーブルに存在しない場合、NULL 値を返します。

crfld1 に *hfld1* 内の 1 つの値と共通の複数インスタンスが存在する可能性がある場合、MULTIPLE パラメータを使用します。ALL は MULTIPLE の同義語で、このパラメータを完全に省略すると、UNIQUE と同義になります。

IN crfile

クロスリファレンステーブルの名前です。

TAG tag2

クロスリファレンステーブルのフィールドおよびエイリアスのユニーク修飾子として使用する、66 バイト以内のタグ名です。通常はマスターファイル名です。再帰的 JOIN 構造では、タグ名が指定されていない場合、すべてのフィールド名およびエイリアスの前に JOIN 名の先頭 4 バイトが追加されます。

ホストファイルのタグ名は、JOIN 構造のすべての JOIN コマンドで同一にします。

AS joinname

JOIN 構造に指定する 8 バイト以内の名前です (オプション)。次のような場合、JOIN 構造に一意の名前を付ける必要があります。

- 後から実行する JOIN コマンドによる上書きを回避する。
- 後で特定の JOIN をクリアする。
- 再帰的構造を持ち、タグ名を指定しない場合。

JOIN 構造に AS 句により名前を指定しない場合、名前はブランクになります。JOIN に名前を付けないと、名前のない既存の JOIN が上書きされます。

END

JOIN コマンドの長さが 1 行を超える場合に必要です。この句により、コマンドが終了します。この句は、独立した行に記述する必要があります。

次の構文は、条件付き FULL OUTER JOIN を生成します。

```
JOIN FULL_OUTER FILE table1 AT hfld1 [WITH hfld2] [TAG tag1]
     TO {UNIQUE|MULTIPLE}
     FILE table2 AT crfld [TAG tag2] [AS joinname]
     [WHERE expression1;
     [WHERE expression2;
     ...]
END
```

説明

table1

ホストマスターファイルです。

AT

親セグメント (ホストセグメント) を子セグメント (クロスリファレンスセグメント) に正しく接続します。AT パラメータとして使用されるフィールド値は、リンクには使用されません。これらは単にセグメントの参照用として使用されます。

hfld1

ホストマスターファイルのフィールド名です。このフィールドのセグメントがクロスリファレンステーブルに結合されます。フィールド名は、参照するデータソースで最下位のセグメントである必要があります。

tag1

ホストテーブルのフィールドのユニーク修飾子およびエイリアスとして使用するオプションのタグ名です。

hfld2

DEFINE に基づく条件付き JOIN に関連付けるテーブルフィールドです。DEFINE ベースの条件付き JOIN の場合、KEEPDEFINES 設定を ON または ALL にする必要があり、JOIN コマンドを発行する前に、一時項目 (DEFINE) を作成する必要があります。

MULTIPLE

table1 と table 2 間の 1 対 n 関係を指定します。ALL は MULTIPLE の同義語です。

UNIQUE

table1 と table 2 間の 1 対 1 関係を指定します。ONE は UNIQUE の同義語です。

注意：UNIQUE を指定した JOIN では、クロスリファレンステーブルの 1 つのインスタンスのみが返されます。また、WHERE 式の評価に基づいて一致するインスタンスが存在しない場合は、NULL 値が返されます。

crfile

クロスリファレンスマスターファイルです。

crfld

クロスリファレンスマスターファイルの JOIN フィールドの名前です。セグメント内の任意のフィールドを指定することができます。

tag2

クロスリファレンステーブルのフィールドのユニーク修飾子およびエイリアスとして使用するオプションのタグ名です。

joinname

JOIN 構造に関連付けられた名前です。

expression1, expression2

任意の条件を指定します。式に含めるフィールドはすべて同一パス内に存在する必要があります。

END

コマンドを終了するには、END 句が必要です。この句は、独立した行に記述する必要があります。

Microsoft SQL Server テーブルの FULL OUTER JOIN の最適化

次のリクエストでは、結合する 2 つの Microsoft SQL Server テーブルを生成した後、その JOIN に対してリクエストを発行します。これらのテーブルは、wf_retail サンプルから生成されます。このサンプルは、Reporting Server Web コンソールの [WebFOCUS - Retail デモ] チュートリアルを使用して作成することができます。

次のリクエストは、WF_SALES テーブルを生成します。ID_PRODUCT フィールドは、FULL OUTER JOIN コマンドで使用されます。生成されるテーブルには、ID_PRODUCT の 2150 から 4000 までの値が格納されます。

```
TABLE FILE WF_RETAIL_LITE
SUM GROSS_PROFIT_US PRODUCT_CATEGORY PRODUCT_SUBCATEG
BY ID_PRODUCT
WHERE ID_PRODUCT FROM 2150 TO 4000
ON TABLE HOLD AS WF_SALES FORMAT SQLMSS
END
```

次のリクエストは、WF_PRODUCT テーブルを生成します。ID_PRODUCT フィールドは、FULL OUTER JOIN コマンドで使用されます。生成されるテーブルには、ID_PRODUCT の 3000 から 5000 までの値が格納されます。

```
TABLE FILE WF_RETAIL_LITE
SUM PRICE_DOLLARS PRODUCT_CATEGORY PRODUCT_SUBCATEG PRODUCT_NAME
BY ID_PRODUCT
WHERE ID_PRODUCT FROM 3000 TO 5000
```

```
ON TABLE HOLD AS WF_PRODUCT FORMAT SQLMSS
END
```

次のリクエストは、SET SHORTPATH=SQL コマンドおよび JOIN コマンドを発行し、結合済みテーブルの値を表示します。

```
 SET SHORTPATH = SQL
 SET TRACEUSER=ON
 SET TRACESTAMP=OFF
 SET TRACEOFF=ALL
 SET TRACEON = STMTRACE//CLIENT
 JOIN FULL_OUTER ID_PRODUCT IN WF_PRODUCT TAG T1
  TO ALL ID_PRODUCT IN WF_SALES TAG T2
 END
 TABLE FILE WF_PRODUCT
 PRINT T1.ID_PRODUCT AS 'Product ID'
 PRICE_DOLLARS AS Price
 T2.ID_PRODUCT AS 'Sales ID'
 GROSS_PROFIT_US
 BY T1.ID_PRODUCT NOPRINT
 ON TABLE SET PAGE NOPAGE
 ON TABLE SET STYLE *
 GRID=OFF,$
 ENDSTYLE
 END
```

トレースを確認すると、SQL が JOIN を処理できるよう、FULL OUTER JOIN が最適化 (SQL に変換) されていることが分かります。

```
 SELECT
 T1."ID_PRODUCT",
 T1."PRICE_DOLLARS",
 T2."ID_PRODUCT",
 T2."GROSS_PROFIT_US"
 FROM
 ( WF_PRODUCT T1
 FULL OUTER JOIN
 WF_SALES T2
 ON T2."ID_PRODUCT" = T1."ID_PRODUCT" )
 ORDER BY
 T1."ID_PRODUCT";
```

出力結果では、いずれかのテーブルに存在する ID_PRODUCT 値ごとに行が作成されています。ID_PRODUCT 値が 2150 から 2167 までの行は、WF_SALES テーブルのみに存在します。そのため、WF_PRODUCT のフィールドには、NODATA 記号が表示されています。ID_

PRODUCT 値が 4000 を超える行は、WF_PRODUCT テーブルのみに存在します。そのため、WF_SALES のフィールドには、NODATA 記号が表示されています。ID_PRODUCT 値が 3000 から 4000 までの行は、両方のテーブルに存在します。そのため、すべてのフィールドに値が表示されています。

Product ID	Price	Sales ID	Gross Profit
.	.	2151	$7,172.24
.	.	2152	$6,766.45
.	.	2153	$7,291.10
.	.	2154	$19,913.65
.	.	2155	$12,472.93
.	.	2156	$14,484.75
.	.	2157	$6,223.34
.	.	2158	$11,267.75
.	.	2159	$9,015.75
.	.	2160	$7,194.15
.	.	2161	$8,683.00
.	.	2162	$9,197.81
.	.	2163	$21,347.98
.	.	2164	$5,615.33
.	.	2165	$8,540.53
.	.	2166	$12,469.45
.	.	2167	$6,771.26
3001	47,988.00	3001	$12,786.30
3002	34,990.00	3002	$5,586.10
3003	30,591.00	3003	$12,399.15
3004	44,187.00	3004	$8,434.95
3005	30,057.00	3005	$8,987.80
3006	25,348.05	3006	$6,422.59
3007	23,607.00	3007	$8,354.60
3008	42,898.57	3008	$18,469.01
3009	48,139.00	3009	$22,542.75
3010	44,850.00	3010	$16,722.15
3011	28,078.44	3011	$18,204.82
3012	31,146.00	3012	$19,739.45
3013	27,208.00	3013	$17,983.00
3014	23,968.59	3014	$19,996.09
3015	29,061.45	3015	$17,420.72
4012	43,998.35	.	.
4013	24,749.89	.	.
4014	33,966.00	.	.
4015	20,998.95	.	.
4016	32,899.06	.	.

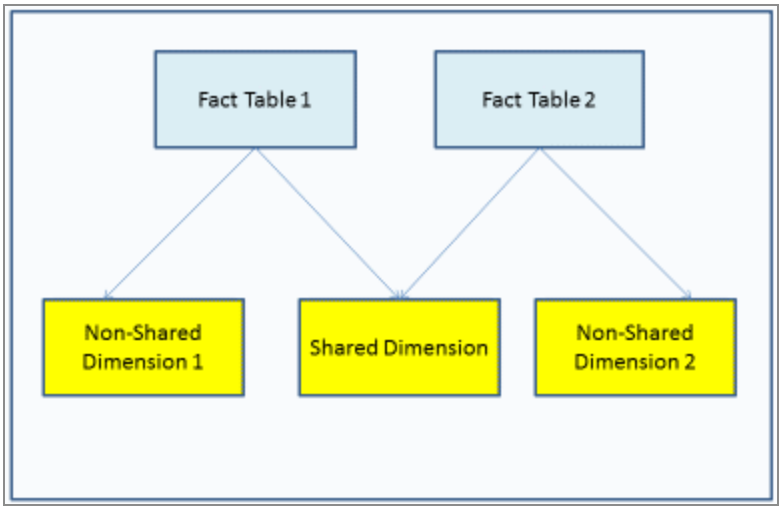
マルチファクトクラスタシノニムに対するレポートの作成

クラスタシノニムは、ベースシノニムを指定する CRFILE 属性参照により、各セグメントがクラスタに追加されたシノニムです。子セグメントは、JOIN WHERE 属性によって、親セグメントに結合されます。クラスタマスターファイルは、複数のルートセグメントを持つことができます。この場合、スタースキーマで見られるように、通常、ルートセグメントはファクトテーブルであり、子セグメントはディメンションテーブルです。この種の構造は、「マルチファクトクラスタ」呼ばれます。

ディメンションテーブルは、(共有ディメンションと呼ばれる) マルチファクトテーブルの子、または (非共有ディメンションと呼ばれる) シングルファクトテーブルにすることができ

ます。ほとんどの場合、ファクトテーブルは集計に使用され、ディメンションテーブルはソートに使用されます。

下図は、単純なマルチファクト構造を示しています。



マルチファクトクラスタマスターファイルに対するリクエストを作成する際の規則は、次のとおりです。

- 各ファクトテーブルから少なくとも 1 つのファクトを集計し (SUM)、ソート句を 1 つ以上含む場合に限り、ファクトテーブルに対してのみのレポートを作成できます。
- リクエストで最初のソートフィールドは、共有ディメンションのフィールドにする必要があります。
- リクエストで参照可能な共有ディメンションの数に制限はありません。
- 共通の親を持つ場合は、リクエストに複数の非共有ディメンションを含めることができます。異なる親を持つ複数の非共有ディメンションを、リクエストで参照することはできません。
- マルチファクトシノニムに対するレポートでは、MATCH FILE コマンドはサポートされません。

マルチファクトクラスタシノニムに対するレポートの作成

次のリクエストは、WF_RETAIL_LITE マルチファクトクラスタシノニムを使用し、WF_RETAIL_SALES セグメントの COGS_US メジャーおよび WF_RETAIL_SHIPMENTS セグメン

トの DAYSDELAYED メジャーを集計します。最初の BY フィールドである BRAND は、共有ディメンションの WF_RETAIL_PRODUCT に存在します。2 つ目の BY フィールドである TIME_QTR は、非共有ディメンションの WF_RETAIL_TIME_DELIVERED に存在します。

```
TABLE FILE WF_RETAIL_LITE
SUM COGS_US DAYSDELAYED
BY BRAND
BY WF_RETAIL_TIME_DELIVERED.TIME_QTR
WHERE BRAND EQ 'Denon' OR 'Grado'
WHERE DAYSDELAYED GT 1
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/ibi_themes/Warm.sty,$
END
```

出力結果は次のとおりです。DAYSDELAYED の合計は共有ディメンションの各値、共有ディメンションの各値内、および非共有ディメンションの各値で計算されています。

Brand	Cost of Goods	Quarter	Days Delayed
Denon	$25,970.00		12
		1	3
		2	4
		3	5
Grado	$21,930.00		5
		2	5

マルチファクトシノニムへの新しいファクトの追加 - JOIN AS_ROOT

JOIN AS_ROOT コマンドを使用すると、既存のファクトベースのクラスタ (スタースキーマ) に、新しいファクトテーブルが別のルートとして追加されます。ソースマスターファイルには、親ファクトセグメントと、少なくとも 1 つの子ディメンションセグメントが含まれています。JOIN AS_ROOT コマンドでは、子ディメンションセグメント (任意のレベル) から別のファクト親セグメントへのユニーク JOIN がサポートされます。

新しい親セグメントの追加

```
JOIN AS_ROOT sfld1 [AND sfld2 ...] IN [app1/]sfile TO UNIQUE tfld1 [AND tfld2
...] IN [app2/]tfile AS jname
END
```

説明

sfld1 [AND sfld2 ...]

ソースファイルの子 (ディメンション) セグメントのフィールドのうち、ターゲットファイルのフィールド値と照合するフィールドです。

[app1/]sfile

ソースファイルです。

TO UNIQUE tfld1 [AND tfld2 ...]

ターゲットファイルのフィールドのうち、ソースファイルの子セグメントのフィールド値と照合するフィールドです。JOIN は一意である必要があります。

[app2/]tfile

ターゲットファイルです。

jname

JOIN の名前です。

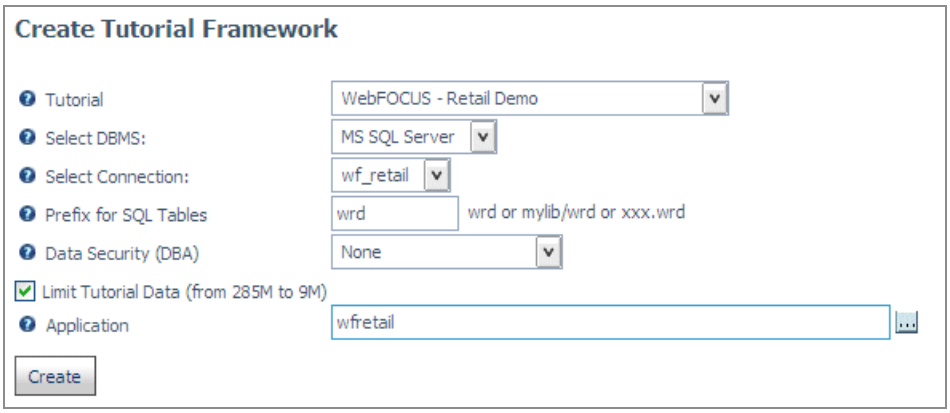
END

JOIN コマンドを終了するために必要です。

WebFOCUS Retail データソースの AS_ROOT と Excel ファイルの結合

次のリクエストは、WebFOCUS Retail データソースの PRODUCT_CATEGORY および PRODUCT_SUBCATEGORY フィールドを、PROJECTED という名前の Excel ファイルに結合します。

Reporting Server Web コンソールで WebFOCUS Retail データソースを生成するには、[アプリケーション] ページで [新規]、[チュートリアル] を順に選択します。



[WebFOCUS - Retail デモ] を選択します。構成済みリレーショナルアダプタを選択し (リレーショナルアダプタが構成されていない場合は [Flat File] オプションを選択)、[チュートリアルデータのデータ量の制限] のチェックをオンにした上で、[作成] をクリックします。

Excel ファイルマスターファイルは、次のとおりです。

```
FILENAME=PROJECTED, SUFFIX=DIREXCEL,
 DATASET=app2/projected.xlsx, $
  SEGMENT=PROJECTED, SEGTYPE=S0, $
    FIELDNAME=PRODUCT_CATEGORY, ALIAS='Product  Category', USAGE=A16V,
ACTUAL=A16V,
      MISSING=ON,
      TITLE='Product  Category',
      WITHIN='*PRODUCT', $
    FIELDNAME=PRODUCT_SUBCATEGORY, ALIAS='Product     Subcategory',
USAGE=A25V, ACTUAL=A25V,
      MISSING=ON,
      TITLE='Product     Subcategory',
      WITHIN=PRODUCT_CATEGORY, $
    FIELDNAME=PROJECTED_COG, ALIAS='              Projected COG',
USAGE=P15.2C, ACTUAL=A15,
      MISSING=ON,
      TITLE='              Projected COG', MEASURE_GROUP=PROJECTED,
      PROPERTY=MEASURE,  $
    FIELDNAME=PROJECTED_SALE_UNITS, ALIAS='             Projected Sale
Units', USAGE=I9, ACTUAL=A11,
      MISSING=ON,
      TITLE='             Projected Sale Units', MEASURE_GROUP=PROJECTED,
      PROPERTY=MEASURE,  $
 MEASUREGROUP=PROJECTED, CAPTION='PROJECTED', $
 DIMENSION=PRODUCT, CAPTION='Product', $
  HIERARCHY=PRODUCT, CAPTION='Product', HRY_DIMENSION=PRODUCT, HRY_
STRUCTURE=STANDARD, $
```

下図は、Excel ファイルのデータを示しています。

	A	B	C	D
1	Product Category	Product Subcategory	Projected COG	Projected Sale Units
2	Accessories	Charger	$2,068,508.00	75,279
3	Accessories	Headphones	$52,061,301.00	163,152
4	Accessories	Universal Remote Controls	$36,297,267.00	127,286
5	Camcorder	Handheld	$20,733,053.00	178,704
6	Camcorder	Professional	$35,440,708.00	9,095
7	Camcorder	Standard	$49,442,067.00	137,489
8	Computers	Smartphone	$44,420,201.00	146,858
9	Computers	Tablet	$26,047,885.00	105,053
10	Media Player	Blu Ray	$182,459,862.00	485,131
11	Media Player	DVD Players	$3,756,254.00	13,346
12	Media Player	DVD Players - Portable	$306,576.00	3,981
13	Media Player	Streaming	$5,108,342.00	48,630
14	Stereo Systems	Boom Box	$840,373.00	6,687
15	Stereo Systems	Home Theater Systems	$56,829,817.00	285,041
16	Stereo Systems	Receivers	$40,620,030.00	107,537
17	Stereo Systems	Speaker Kits	$81,962,756.00	174,036
18	Stereo Systems	iPod Docking Station	$26,310,783.00	221,723
19	Televisions	CRT TV	$1,928,416.00	3,268
20	Televisions	Flat Panel TV	$59,540,624.00	66,119
21	Televisions	Portable TV	$545,348.00	5,696
22	Video Production	Video Editing	$40,380,803.00	142,594

次のリクエストは、wf_retail データソースの wf_retail_product セグメントを新しいルートとして Excel ファイルに結合し、両方の親セグメントからレポートを作成します。

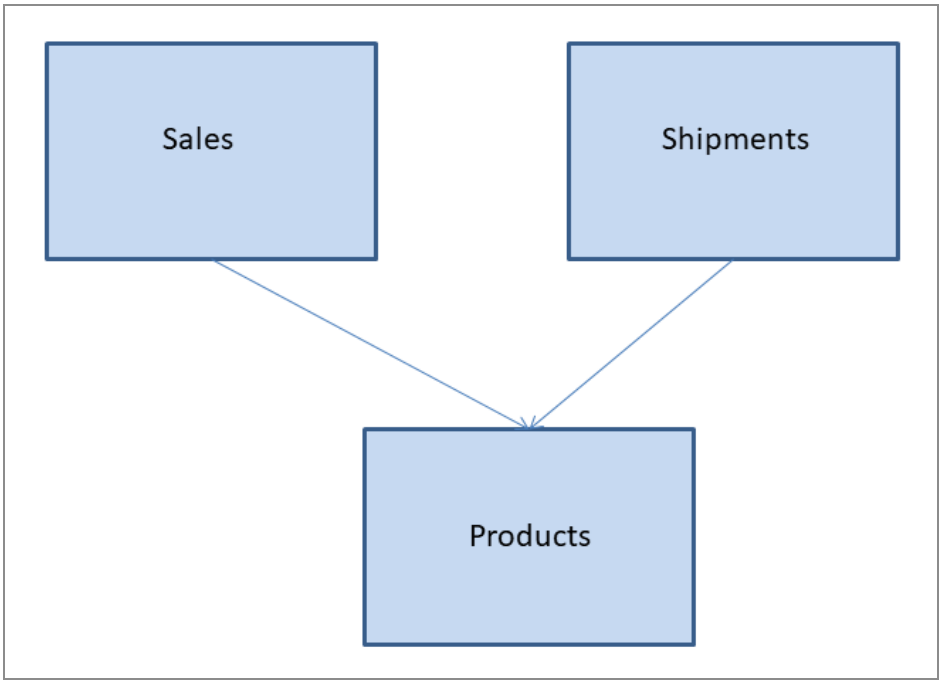
```
JOIN AS_ROOT PRODUCT_CATEGORY AND PRODUCT_SUBCATEG IN WF_RETAIL
  TO UNIQUE PRODUCT_CATEGORY AND PRODUCT_SUBCATEGORY IN PROJECTED
  AS J1
END
TABLE FILE WF_RETAIL
SUM PROJECTED_SALE_UNITS REVENUE_US
BY PRODUCT_CATEGORY
ON TABLE SET PAGE NOPAGE
END
```

出力結果は次のとおりです。

Product Category	Projected Sale Units	Revenue
Accessories	365717	$41,817.34
Camcorder	325288	$49,768.33
Computers	251911	$18,298.75
Media Player	551088	$118,615.17
Stereo Systems	795024	$111,768.71
Televisions	75083	$27,209.29
Video Production	142594	$25,837.37

クラスタシノニムコンテキストでの OUTER JOIN の作成

複数ルートセグメントおよび共有ディメンションに対してレポートを作成する場合、クラスタシノニム内に複数のコンテキストが作成されます。たとえば、下図では、Sales および Products は 1 つのコンテキストを構成する一方で、Shipments および Products は 2 つ目のコンテキストを構成します。

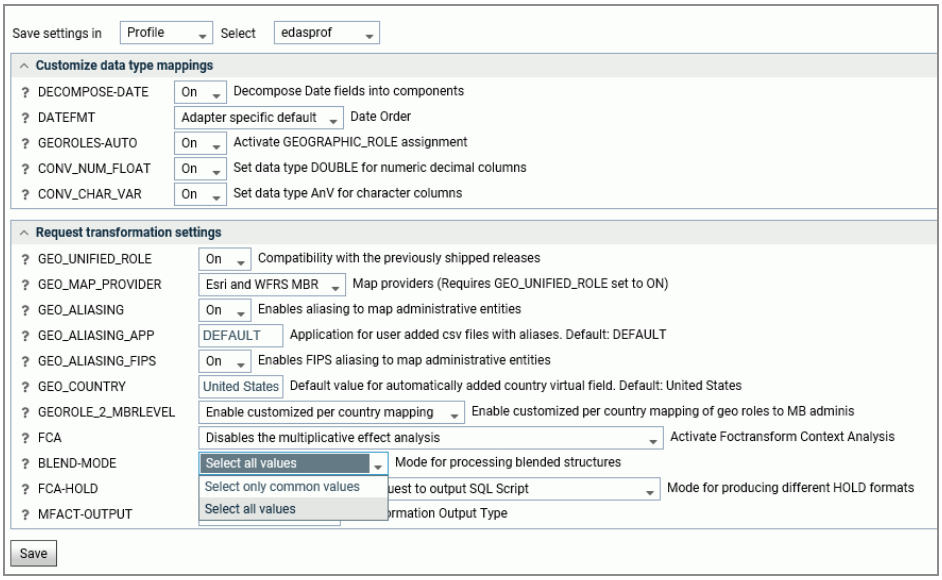


リクエストに両方のコンテキストのフィールドが含まれる場合、デフォルト設定では、単一の INNER JOIN が SQL エンジンに渡されます。これにより、両方のコンテキストの共有ディメンションに存在する一致値のみが取得されます。

INNER JOIN の代わりに BLEND-MODE パラメータを使用して FULL OUTER JOIN を生成することで、両方のコンテキストからすべての値を取得することができます。

クラスタシノニムコンテキストでの JOIN 処理の制御

Reporting Server の Web コンソールで混合モードパラメータを設定し、この設定をプロファイルまたはプロシジャに格納することができます。[データの取得] ページで [共通アダプタの設定] をクリックし、[リクエスト変換の設定] セクションの [BLEND-MODE] ドロップダウンリストから [すべての値を選択] を選択します。



また、混合モードパラメータを設定するには、次の構文を使用することもできます。

```
ENGINE INT SET BLEND-MODE {COMMON-VALUES|ALL-VALUES}
```

説明

COMMON-VALUES

複数のクラスタシノニムコンテキストでの単一 INNER JOIN を生成し、共有ディメンションフィールドでの一致値のみを取得します。これがデフォルト値です。

ALL-VALUES

複数クラスタシノニムコンテキストでの単一 FULL OUTER JOIN を生成し、共有ディメンションフィールドでのすべての値を取得します。ミッシング値は、共有ディメンションフィールドで一致する値が存在しないコンテキストのフィールドから取得されます。

クラスタシノニムコンテキストでの JOIN 処理の制御

次の Excel ファイル (excelroot.xlsx) は、Excel アダプタを使用してサーバにアップロードされ、WF_RETAIL マスターファイルのルートとして結合されます。2 つのコンテキストが作成されます。次に、この 2 つのルートおよび共有ディメンションに対して、リクエストが発行されます。

	A	B	C	D
1	Product Category	Product Subcategory	Projected COG	Projected Sale Units
2	Accessories	Charger	$2,068,508.00	75,279
3	Accessories	Headphones	$52,061,301.00	163,152
4	Accessories	Universal Remote Controls	$36,297,267.00	127,286
5	Computers	Smartphone	$44,420,201.00	146,858
6	Computers	Tablet	$26,047,885.00	105,053
7	Media Player	Blu Ray	$182,459,862.00	485,131
8	Media Player	DVD Players	$3,756,254.00	13,346
9	Media Player	DVD Players - Portable	$306,576.00	3,981
10	Media Player	Streaming	$5,108,342.00	48,630
11	Displays	CRT TV	$1,928,416.00	3,268
12	Televisions	Flat Panel TV	$59,540,624.00	66,119
13	Televisions	Portable TV	$545,348.00	142,594

このファイルの製品区分 (PRODUCT_CATEGORY) Camcoder、Stereo System、Video Production にはデータが含まれていません。製品区分 Display は、WF_RETAIL には存在しません。

この Excel ファイルに対して、次のマスターファイルが生成されます。

```
FILENAME=EXCELROOT, SUFFIX=DIREXCEL,
 DATASET=ibisamp/excelroot.xlsx, $
  SEGMENT=EXCELROOT, SEGTYPE=S0, $
    FIELDNAME=PRODUCT_CATEGORY, ALIAS='Product Category', USAGE=A15V,
ACTUAL=A15V,
      MISSING=ON,
      TITLE='Product Category', $
    FIELDNAME=PRODUCT_SUBCATEGORY, ALIAS='Product Subcategory', USAGE=A31V,
```

```
ACTUAL=A31V,
      MISSING=ON,
      TITLE='Product Subcategory', $
    FIELDNAME=PROJECTED_COG, ALIAS='Projected COG', USAGE=D15.2:C,
ACTUAL=A64V,
      MISSING=ON,
      TITLE='Projected COG',
      CURRENCY_DISPLAY=LEFT_FLOAT,  CURRENCY_ISO_CODE=USD,  $
    FIELDNAME=PROJECTED_SALE_UNITS, ALIAS='Projected Sale Units', USAGE=I9,
ACTUAL=A11V,
      MISSING=ON,
      TITLE='Projected Sale Units', $
```

次のリクエストは、Excel ファイルをルートとして結合し、ルートおよび共有ディメンションの両方のフィールドが含まれたレポートを生成します。BLEND-MODE のデフォルト値を使用するため、PRODUCT_CATEGORY で共通する値のみを取得する INNER JOIN が生成されます。

```
JOIN AS_ROOT PRODUCT_CATEGORY AND PRODUCT_SUBCATEG IN ibisamp/WF_RETAIL
  TO PRODUCT_CATEGORY AND PRODUCT_SUBCATEGORY IN ibisamp/EXCELROOT
  AS J1
  END
TABLE FILE ibisamp/WF_RETAIL
SUM COGS_US PROJECTED_SALE_UNITS
BY PRODUCT_CATEGORY
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
GRID=OFF,$
END
```

出力結果は次のとおりです。

Product Category	Cost of Goods	Projected Sale Units
Accessories	$342,877.00	365717
Computers	$109,281.00	251911
Media Player	$779,593.00	551088
Televisions	$227,820.00	208713

次のリクエストは、ENGINE INT SET BLEND-MODE ALL-VALUES コマンドを発行し、PRODUCT_CATEGORY のすべての値を返す FULL OUTER JOIN を生成します。

```
ENGINE INT SET BLEND-MODE ALL-VALUES
JOIN AS_ROOT PRODUCT_CATEGORY AND PRODUCT_SUBCATEG IN ibisamp/WF_RETAIL
  TO PRODUCT_CATEGORY AND PRODUCT_SUBCATEGORY IN ibisamp/EXCELROOT
  AS J1
  END
TABLE FILE ibisamp/WF_RETAIL
SUM COGS_US PROJECTED_SALE_UNITS
BY PRODUCT_CATEGORY
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
GRID=OFF,$
END
```

出力結果は次のとおりです。ミッシング値を示す記号が表示されています。

- 製品区分 (PRODUCT_CATEGORY) Camcorder、Stereo Systems、Video Production に対応する行の 販売予定台数 (PROJECTED_SALE_UNITS) フィールドは、Excel ファイルには表示されていません。
- 製品区分 (PRODUCT_CATEGORY) Display に対応する行の値は、WF_RETAIL データソースに存在しません。

Product Category	Cost of Goods	Projected Sale Units
Accessories	$89,753,898.00	1964918
Camcorder	$104,866,857.00	.
Computers	$69,807,664.00	2519110
Displays	.	3268
Media Player	$190,240,481.00	10585039
Stereo Systems	$205,113,863.00	.
Televisions	$61,551,109.00	1033209
Video Production	$40,105,657.00	.

マルチファクトシノニムからの JOIN の作成

ターゲットシノニムの単一セグメントへの JOIN を作成する際に、ソースとしてマルチ親シノニムがサポートされます。

マルチ親シノニムからの JOIN には、次の条件が適用されます。

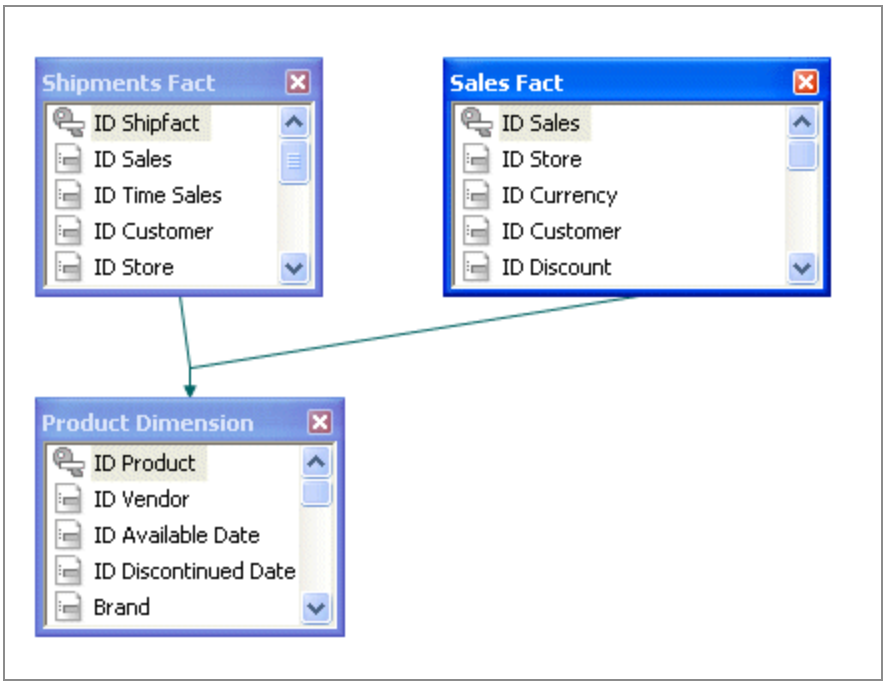
- 条件付き JOIN はサポートされません (JOIN WHERE)。
- JOIN は一意である必要があります。つまり、TO ALL 句および TO MULTIPLE 句はサポートされません。
- JOIN のターゲットをマルチ親シノニムにすることはできません。
- JOIN のターゲットは、単一セグメントシノニム内の単一セグメントにするか、単一親マルチセグメントシノニム内の単一セグメントにする必要があります。
- JOIN に含めるすべてのフィールドは、ソースシノニムとターゲットシノニムの両方で、単一セグメントに属している必要があります。ソースシノニムの任意の単一セグメントを JOIN に使用することができます。

マルチファクトシノニムからの JOIN の作成

次のマスターファイルには、WebFOCUS Retail チュートリアルに基づいて、マルチ親構造が記述されています。2 つのファクトテーブル (wf_retail_sales および wf_retail_shipments) は、wf_retail_product ディメンションテーブルの親です。

```
FILENAME=WF_RETAIL_MULTI_PARENT, $
  SEGMENT=WF_RETAIL_SHIPMENTS, CRFILE=WFRETAIL/FACTS/WF_RETAIL_SHIPMENTS,
CRINCLUDE=ALL,
    DESCRIPTION='Shipments Fact', $
  SEGMENT=WF_RETAIL_SALES, PARENT=., CRFILE=WFRETAIL/FACTS/WF_RETAIL_SALES,
CRINCLUDE=ALL,
    DESCRIPTION='Sales Fact', $
  SEGMENT=WF_RETAIL_PRODUCT, CRFILE=WFRETAIL/DIMENSIONS/WF_RETAIL_PRODUCT,
CRINCLUDE=ALL,
    DESCRIPTION='Product Dimension', $
   PARENT=WF_RETAIL_SHIPMENTS, SEGTYPE=KU,
    JOIN_WHERE=WF_RETAIL_SHIPMENTS.ID_PRODUCT EQ WF_RETAIL_PRODUCT.ID_
PRODUCT;, $
   PARENT=WF_RETAIL_SALES, SEGTYPE=KU,
    JOIN_WHERE=WF_RETAIL_SALES.ID_PRODUCT EQ WF_RETAIL_PRODUCT.ID_PRODUCT;, $
```

下図は、これらのテーブル間の JOIN を、データ管理コンソールのシノニムエディタで表示したものです。



次のリクエストは、ID_VENDOR に基づいて wf_retail_vendor ディメンションテーブルに Product セグメントを結合し、結合後の JOIN 構造に対してリクエストを発行します。

```
JOIN ID_VENDOR IN WF_RETAIL_MULTI_PARENT TO ID_VENDOR IN WF_RETAIL_VENDOR AS
J1
TABLE FILE WF_RETAIL_MULTI_PARENT
SUM COGS_US DAYSDELAYED
BY PRODUCT_CATEGORY
BY VENDOR_NAME
WHERE PRODUCT_CATEGORY LT 'S'
ON TABLE SET PAGE NOPAGE
END
```

出力結果は次のとおりです。

Product Category	Vendor Name	Cost of Goods	Days Delayed
Accessories	Audio Technica	$38,000.00	28
	Denon	$25,970.00	17
	Grado	$21,930.00	15
	Logitech	$61,432.00	114
	Niles Audio	$73,547.00	150
	Pioneer	$16,720.00	71
	Samsung	$5,405.00	83
	Sennheiser	$78,113.00	128
	Sony	$21,760.00	157
Camcorder	Canon	$110,219.00	97
	JVC	$72,292.00	75
	Panasonic	$22,356.00	91
	Sanyo	$31,590.00	179
	Sony	$216,748.00	333
Computers	Samsung	$33,129.00	156
	Sony	$76,152.00	186
Media Player	JVC	$87,057.00	267
	LG	$3,830.00	13
	Panasonic	$143,600.00	171
	Pioneer	$169,810.00	206
	Roku	$10,248.00	85
	Samsung	$151,620.00	191
	Sharp	$66,024.00	157
	Sony	$142,190.00	121
	Toshiba	$5,214.00	7

クラスタシノニム間の JOIN のナビゲーション

クラスタシノニムの JOIN のデフォルト設定では、セグメントの階層は、結合されたすべてのファイルから作成され、作成後の階層は、上下、左右の順序でナビゲートされます。

そのため、ホストシノニムから INNER JOIN を含むクラスタへの LEFT OUTER JOIN を指定すると、INNER JOIN は最後に実行されるため、LEFT OUTER JOIN の目的に反して、ホストファイルから行が削除される場合があります。SET FOCTRANSFORM = NESTED_CLUSTERS/ON コマンドを使用すると、ターゲットクラスタでの JOIN を、ホストシノニムとターゲットシノニム間の JOIN の前に実行することができます。この設定を使用する場合、SQL スクリプトを使用して、ホストファイルへの JOIN を実装する前に、ターゲットクラスタのテーブルの JOIN が実行されます。LEFT OUTER JOIN が最後に実行されるため、ホストシノニムのすべての行が保持されます。

構文は次のとおりです。

```
SET FOCTRANSFORM = {NESTED_CLUSTERS/OFF|NESTED_CLUSTERS/ON}
```

説明

NESTED_CLUSTERS/OFF

セグメントナビゲーション順序の左右、上下が保持されます。これがデフォルト値です。

NESTED_CLUSTERS/ON

ホストシノニムから結果への JOIN の前に、ターゲットクラスタの JOIN を実行します。

クラスタシノニム JOIN 使用上の注意

- SET FOCTRANSFORM = NESTED_CLUSTERS 機能を使用するには、JOIN を最適化する必要があります。FOCTRANSFORM = NESTED_CLUSTERS/OFF 設定で INNER JOIN および OUTER JOIN の組み合わせを使用する場合、リクエストを最適化するには、SET SHORTPATH = SQL コマンドを有効にしておく必要があります。SHORTPATH = SQL 設定は、FOCTRANSFORM = NESTED_CLUSTERS/ON 設定による最適化には影響しません。
- クラスタシノニムでルート以外のセグメントは結合できません。ルート以外のセグメントで JOIN を発行すると、次のメッセージが表示されて、リクエストは終了します。

  ```
  (FOC906) NESTED_CLUSTERS でルート以外のセグメント segname への JOIN は許可されていません。
  ```

クラスタシノニム間の JOIN のナビゲーション

この例では、「https://www.citibikenyc.com/system-data」からアップロードされた citibike の走行データ、および走行時に使用したステーションの郵便番号ファイル (https://techsupport.informationbuilders.com/public/station_zip.csv からダウンロード可) から生成された SQL Server データソースを使用します。

「station_trip_cls」という名前のクラスタシノニムは、ステーションの郵便番号データソースを一部の走行データ (数行のみを含む) と結合します。以下は、このクラスタシノニムで定義された INNER JOIN を示しています。

```
FILENAME=STATION_TRIP_CLS, $
  SEGMENT=STATION_ZIP_OLEDB, CRFILE=CITIBIKE/STATION_ZIP_OLEDB,
CRINCLUDE=ALL, $
  SEGMENT=CITIBIKE_PARTIAL_OLEDB, SEGTYPE=KU, PARENT=STATION_ZIP_OLEDB,
    CRFILE=CITIBIKE/CITIBIKE_PARTIAL_OLEDB, CRINCLUDE=ALL, CRJOINTYPE=INNER,
    JOIN_WHERE=STATION_ID EQ START_STATION_ID;, $
```

次のリクエストは、大規模な走行データファイルからクラスタへの LEFT OUTER JOIN を発行します。

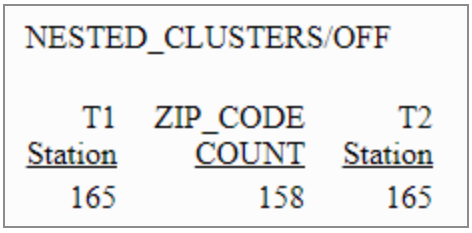
```
SET FOCTRANSFORM = NESTED_CLUSTERS/&VALUE
SET SHORTPATH = SQL
JOIN LEFT_OUTER START_STATION_ID IN CITIBIKE_TRIPDATA TAG T1 TO ALL STATION_
ID IN STATION_TRIP_CLS TAG T2 AS J1
TABLE FILE CITIBIKE_TRIPDATA
" NESTED_CLUSTERS/&VALUE"
" "
SUM CNT.T1.START_STATION_ID AS T1,Station CNT.ZIP_CODE CNT.T2.START_STATION_
ID AS T2,Station
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

このリクエストを &VALUE を OFF に設定して実行すると、次のトレースが生成されます。

```
SELECT
   COUNT(T1."START_STATION_ID"),
   COUNT(T2."ZIP_CODE"),
   COUNT(T3."START_STATION_ID")
```

```
    FROM
    ( ( citibike_tripdata_mssqloledb T1
    LEFT OUTER JOIN
    station_zip_oledb T2
    ON T2."STATION_ID" = T1."START_STATION_ID" )
    INNER JOIN
    citibike_partial_msoledb T3
    ON (T3."START_STATION_ID" = T2."STATION_ID") );
```

出力結果は次のとおりです。INNER JOIN が最後に実行されるため、ホストファイルのステーション数が削減され、クラスタ内と同数になります。



NESTED_CLUSTERS/OFF

T1 Station	ZIP_CODE COUNT	T2 Station
165	158	165

このリクエストを &VALUE を ON に設定して実行すると、次のトレースが生成されます。ホストファイルとクラスタ内の JOIN で使用する 2 つの SQL スクリプトが生成されます。次に、INNER JOIN の結果に対して LEFT OUTER JOIN が実行されます。

```
SELECT
   T1."START_STATION_ID" AS "SK001_START_STATION_ID",
    COUNT(T1."START_STATION_ID") AS "VB001_CNT_START_STATION_ID"
    FROM
   citibike_tripdata_mssqloledb T1
    GROUP BY
   T1."START_STATION_ID";
   (FOC2546) SQL SCRIPT __CITIBIKE_TRIPDATA_OLEDB_CITIBIKE_TRIPDATA_OLEDB.SQL
CREATED SUCCESSFULLY (BUT NOT EXECUTED)
   _EDATEMP/__citibike_tripdata_oledb_citibike_tripdata_oledb HELD AS SQL_
SCRIPT
    SELECT
  T1."STATION_ID" AS "SK001_STATION_ID",
    COUNT(T1."ZIP_CODE") AS "VB001_CNT_ZIP_CODE",
    COUNT(T2."START_STATION_ID") AS "VB002_CNT_START_STATION_ID"
    FROM
   station_zip_oledb T1,
   citibike_partial_msoledb T2
    WHERE
   (T2."START_STATION_ID" = T1."STATION_ID")
    GROUP BY
   T1."STATION_ID";
```

```
  (FOC2546) SQL SCRIPT __CITIBIKE_TRIPDATA_OLEDB_STATION_PARTIAL_OLEDB_
CLS.SQL CREATED SUCCESSFULLY (BUT NOT EXECUTED)
  _EDATEMP/__citibike_tripdata_oledb_station_partial_oledb_cls HELD AS SQL_
SCRIPT
```

```
    SELECT
    SUM(T1."VB001_CNT_START_STATION_ID"),
    SUM(T2."VB001_CNT_ZIP_CODE"),
    SUM(T2."VB002_CNT_START_STATION_ID")
    FROM
   (
   ( /* vvv */
      SELECT
     T1."START_STATION_ID" AS "SK001_START_STATION_ID",
      COUNT(T1."START_STATION_ID") AS
     "VB001_CNT_START_STATION_ID"
      FROM
     citibike_tripdata_mssqloledb T1
      GROUP BY
     T1."START_STATION_ID"
   ) /* ^^^ */ T1
    LEFT OUTER JOIN
   ( /* vvv */
      SELECT
     T1."STATION_ID" AS "SK001_STATION_ID",
      COUNT(T1."ZIP_CODE") AS "VB001_CNT_ZIP_CODE",
      COUNT(T2."START_STATION_ID") AS
     "VB002_CNT_START_STATION_ID"
      FROM
     station_zip_oledb T1,
     citibike_partial_msoledb T2
      WHERE
     (T2."START_STATION_ID" = T1."STATION_ID")
      GROUP BY
     T1."STATION_ID"
   ) /* ^^^ */ T2
    ON T2."SK001_STATION_ID" = T1."SK001_START_STATION_ID" );
```

出力結果は次のとおりです。LEFT OUTER JOIN が最後に実行されたため、ホストファイルの元のステーション数が保持されています。

NESTED_CLUSTERS/ON

T1 Station	ZIP_CODE COUNT	T2 Station
6680	8	9

データベース間の JOIN の最適化

異なるリレーショナルデータベースシステムからのテーブル結合時に、特定の条件下で取得パフォーマンスが最適化されました。

パフォーマンスの最適化の 1 つは、JOIN の実行前またはサブセレクトの発行前にクロスリファレンステーブルからデータを抽出することによって実現されています。この最適化プロセスは、次のコマンドを発行して無効にすることができます。

```
SQL SET HOLDSQLJOIN = OFF
END
```

デフォルト設定では、このパラメータは ON に設定されています。

次のパフォーマンスの最適化が実装されています。

- 集計以外のクエリでは、クロスリファレンス (TO) テーブルが、内部バイナリフォーマットのファイルとして保存されます。このため、異なるデータベースシステムのテーブルに結合するより高速で実行されます。
- 集計クエリの場合、集計関数 (MIN、MAX、SUM、AVG、COUNT) により結合されたクロスリファレンステーブルでは、検索がサブセレクトのリレーショナルデータベースに渡されます。これにより、小規模のアンサーセットが取得され、パフォーマンスが改善されます。
- 2 つのフィールドが等しい、または 2 つとも NULL であることを検証する句を含むリクエストでは、TO テーブルが内部バイナリフォーマットで保存され、パフォーマンスが改善されます。

生成されたクエリは、セッションログでもトレースファイルでも確認することができます。

次の SQL リクエストは、「citibike_mssql」という名前の Microsoft SQL Server テーブルを「station_zip_mysql」という名前の MySQL テーブルに結合します。

```
SQL
SELECT
```

```
   T1.TRIPDURATION,
   T1.START_STATION_NAME ,
   T1.END_STATION_NAME ,
   T1.BIKEID ,
   T1.BIRTH_YEAR ,
   T1.GENDER ,
   T1.STARTTIME ,
   T1.STOPTIME ,
   T1.USERTYPE ,   T2.ZIP_CODE ,
   T2.COUNTY ,
   T2.CITY
FROM
   citibike.citibike_mssql T1
     INNER JOIN /*Join 1*/
    citibike.station_zip_mysql T2
       ON
      T1.START_STATION_ID = T2.STATION_ID
;
TABLE
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

このリクエストを実行すると、セッションログに生成されたリクエストが表示されます。最初に、station_zip_mysql テーブルが中間フォーマットで保存されます。

```
TABLEF FILE STATION_ZIP_MYSQL
 PRINT
   ZIP_CODE
   COUNTY
   CITY
   STATION_ID
ON TABLE SET CARTESIAN ON
ON TABLE SET ASNAMES ON
ON TABLE SET HOLDLIST PRINTONLY
 ON TABLE HOLD
  AS SQLHLD01
  FORMAT DATREC
END
```

次に、citibike_mssql テーブルが HOLD ファイルに結合されます。

```
JOIN INNER
   CITIBIKE_MSSQL2.START_STATION_ID
```

```
  IN CITIBIKE_MSSQL
 TO ALL
   E04
  IN SQLHLD01
 AS SQLJNM01
END
```

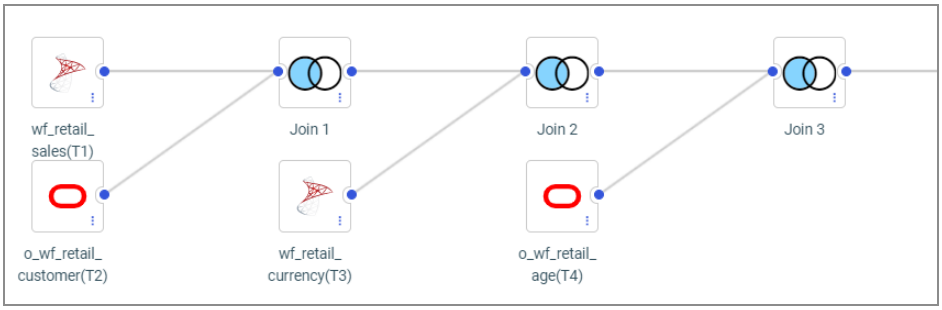
最終的に、この JOIN 構造の検索が実行され、出力が生成されます。

```
TABLEF FILE CITIBIKE_MSSQL
 PRINT
   CITIBIKE_MSSQL.TRIPDURATION
   CITIBIKE_MSSQL.START_STATION_NAME
   CITIBIKE_MSSQL.END_STATION_NAME
   CITIBIKE_MSSQL.BIKEID
   CITIBIKE_MSSQL.BIRTH_YEAR
   CITIBIKE_MSSQL.GENDER
   CITIBIKE_MSSQL.STARTTIME
   CITIBIKE_MSSQL.STOPTIME
   CITIBIKE_MSSQL.USERTYPE
   SQLHLD01.E01
   SQLHLD01.E02
   SQLHLD01.E03
   SQLHLD01.E04 NOPRINT
ON TABLE SET CARTESIAN ON
ON TABLE SET ASNAMES ON
ON TABLE SET HOLDLIST PRINTONLY
END
```

次の変換によって、その他のパフォーマンスの改善が実現されています。

- FROM 句に同一の DBMS による複数のテーブルクラスタが 1 つ以上存在する場合、これらのテーブル間の JOIN は、対応する DBMS に渡されます。
- LEFT OUTER JOIN または RIGHT OUTER JOIN の OUTER テーブルのフィールドに WHERE 条件が設定されている場合、この条件は、NULL 値で常に失敗し、この JOIN は INNER JOIN に変換されます。
- FULL OUTER JOIN の LEFT テーブル、RIGHT テーブル、または両方のテーブルのフィールドに WHERE 条件が設定されている場合、この条件は、NULL 値で常に失敗し、この JOIN は LEFT JOIN、RIGHT JOIN、または INNER JOIN に変換されます。
- マージサブセレクト句の IF FROM、WHERE、GROUP BY、および HAVING は同一で、マージサブセレクトステートメントに対して作成された一時テーブルを再使用します。

たとえば、次のフローは Microsoft SQL Server テーブルと Oracle テーブル間で 3 つの LEFT OUTER JOIN を作成します。Oracle のシノニムには、先頭に「o_」の文字が追加されています。



次の SQL ステートメントは、このフローで生成された JOIN に対応しています。

```
SQL
SELECT
   T1.ID_SALES ,
   T1.ID_STORE ,
   T1.ID_CURRENCY ,
   T1.ID_CUSTOMER ,
   T1.ID_DISCOUNT ,
   T1.ID_PRODUCT ,
   T1.ID_TIME ,
   T1.COGS_LOCAL ,
   T1.COGS_US ,
   T1.DISCOUNT_LOCAL ,
   T1.DISCOUNT_US ,
   T1.GROSS_PROFIT_LOCAL ,
   T1.GROSS_PROFIT_US ,
   T1.MSRP_LOCAL ,
   T1.MSRP_US ,
   T1.QUANTITY_SOLD ,
   T1.REVENUE_LOCAL ,
   T1.REVENUE_US ,
   T2.ID_AGE ,
   T2.ID_EDUCATION ,
   T2.ID_GEOGRAPHY ,
   T2.ID_INCOME ,
   T2.ID_INDUSTRY ,
   T2.ID_MARITAL_STATUS ,
   T2.ID_OCCUPATION ,
   T2.ID_TIME_MIN ,
   T2.ID_TIME_MAX ,
   T2.EMAIL_ADDRESS ,
   T2.FIRSTNAME ,
   T2.FULLNAME ,
```

```
   T2.GENDER ,
   T2.LASTNAME ,
   T2.INCOME ,
   T3.CURRENCY_NAME ,
   T3.CURRENCY_RATE ,
   T4.AGE ,
   T4.AGE_RANGE ,
   T4.AGE_GROUP
```

```
FROM
   ((("ibisamp/facts".wf_retail_sales T1
     LEFT OUTER JOIN /*Join 1*/
    "ibisamp/dimensions".o_wf_retail_customer T2
       ON
      T1.ID_CUSTOMER = T2.ID_CUSTOMER )
     LEFT OUTER JOIN /*Join 2*/
    "ibisamp/dimensions".wf_retail_currency T3
       ON
      T1.ID_CURRENCY = T3.ID_CURRENCY )
     LEFT OUTER JOIN /*Join 3*/
    "ibisamp/dimensions".o_wf_retail_age T4
       ON
      T2.ID_AGE = T4.ID_AGE )
 WHERE
   T1.REVENUE_US > 600  AND
   T2.LASTNAME LIKE 'C%'  AND
   T3.CURRENCY_NAME = 'US Dollar'  AND
   T4.AGE  BETWEEN 35 AND 40
;
END
```

セッションログには、生成された JOIN が表示されます。LEFT OUTER JOIN は INNER JOIN に変換されています。以下は、リストの一部を示しています。

```
JOIN INNER
   SQLAPP01.ID_AGE
  IN 'ibisamp/dimensions/o_wf_retail_customer' TAG SQLAPP01
 TO ALL
   ID_AGE
  IN 'ibisamp/dimensions/o_wf_retail_age' TAG SQLJTG01
 AS SQLJNM01
END
DEFINE FILE 'ibisamp/dimensions/o_wf_retail_customer' TEMP
 SQLDEF01/I1 WITH SQLJTG01.O_WF_RETAIL_AGE.ID_AGE = 1;
END
```

```
TABLEF FILE 'ibisamp/dimensions/o_wf_retail_customer'
 PRINT
   SQLAPP01.ID_AGE
   SQLAPP01.ID_EDUCATION
   SQLAPP01.ID_GEOGRAPHY
   SQLAPP01.ID_INCOME
   SQLAPP01.ID_INDUSTRY
   SQLAPP01.ID_MARITAL_STATUS
   SQLAPP01.ID_OCCUPATION
   SQLAPP01.ID_TIME_MIN
   SQLAPP01.ID_TIME_MAX
   SQLAPP01.EMAIL_ADDRESS
   SQLAPP01.FIRSTNAME
   SQLAPP01.FULLNAME
   SQLAPP01.GENDER
   SQLAPP01.LASTNAME
   SQLAPP01.INCOME
   SQLJTG01.AGE
   SQLJTG01.AGE_RANGE
   SQLJTG01.AGE_GROUP
   SQLAPP01.ID_CUSTOMER
   SQLDEF01 AS (,'SQL$$HIDDEN01',)
   SQLJTG01.ID_AGE NOPRINT
```

```
WHERE ( SQLAPP01.LASTNAME LIKE 'C%' ) ;
 WHERE (  ( SQLJTG01.AGE FROM 35 TO 40
 AND SQLJTG01.AGE NE MISSING) ) ;
ON TABLE SET HOLDATTRS ON
ON TABLE SET CARTESIAN ON
ON TABLE SET ASNAMES MIXED
ON TABLE SET HOLDLIST PRINTONLY
 ON TABLE HOLD
  AS SQLHLD01
  FORMAT DATREC
END
```

同一の DBMS を使用する JOIN は、その DBMS に渡されます。以下は、リストの一部を示しています。

```
   SELECT
  T1."ID_CUSTOMER",
  T1."ID_AGE",
  T1."ID_EDUCATION",
  T1."ID_GEOGRAPHY",
  T1."ID_INCOME",
```

```
T1."ID_INDUSTRY",
T1."ID_MARITAL_STATUS",
T1."ID_OCCUPATION",
T1."ID_TIME_MIN",
T1."ID_TIME_MAX",
T1."EMAIL_ADDRESS",
T1."FIRSTNAME",
T1."FULLNAME",
T1."GENDER",
T1."LASTNAME",
T1."INCOME",
T2."ID_AGE",
T2."AGE",
T2."AGE_RANGE",
T2."AGE_GROUP"
 FROM
wf_retail_customer_t T1,
wf_retail_age_t T2
 WHERE
(T2."ID_AGE" = T1."ID_AGE") AND
(T1."LASTNAME" LIKE 'C%') AND
(T2."AGE" IS NOT NULL) AND
(T2."AGE" BETWEEN 35 AND 40);
```

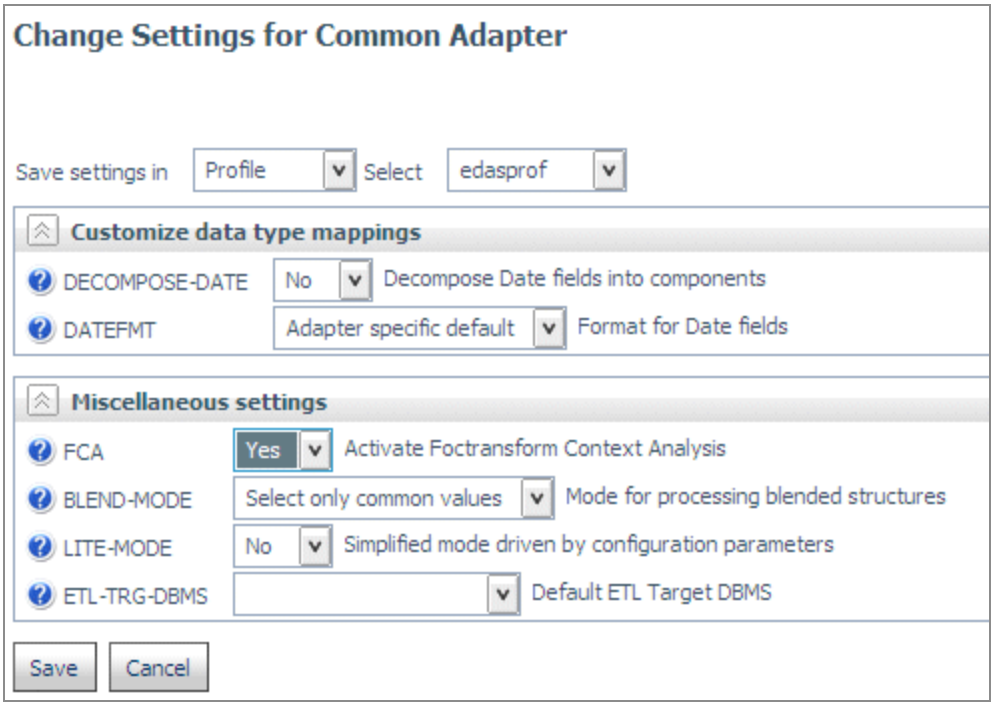
ファントラップ時のスタースキーマコンテキスト分析の呼び出し

スタースキーマのセグメントの 1 つに集計ファクトが含まれ、下位セグメントの 1 つに関連する詳細ファクトが含まれている場合、リクエストを発行してこれらの両方のレベルで集計を実行し、上位側でソートした結果を返す場合、乗算が発生する場合があります。つまり、集計済みのファクト値が再集計され、その結果、乗算された値が返される場合があります。

アダプタが乗算を検知すると、リクエスト処理を操作し、乗算を回避するために最適化をオフにします。ただし、リクエストが最適化されていない場合、パフォーマンスは低下します。

このバージョンでは、新しいコンテキスト分析処理が導入され、この処理によって乗算が検知され、セグメントコンテキストごとに正しい値を取得する SQL スクリプトコマンドが生成されます。これらのスクリプトは、最適化された SQL ステートメントのサブクエリとして RDBMS に渡されます。

コンテキスト分析機能を有効にするには、Reporting Server Web コンソールの [アダプタ] ページで、[共通アダプタ設定の変更] をクリックします。下図のように、[リクエスト変換の設定] セクションで [FCA] パラメータのドロップダウンリストから [はい] を選択し、[保存] をクリックします。



JOIN 条件への DBA 制限の追加 - SET DBAJOIN

複数セグメント構造のリクエストに DBA 制限を適用すると、デフォルト設定で、その制限がリクエストの WHERE 条件として追加されます。DBAJOIN パラメータを ON に設定すると、DBA 制限が、それらが指定されているファイルまたはセグメントに対して内部的な制限として処理され、JOIN 構文に追加されます。

注意：DBAJOIN OFF の場合、DBA 制限は、取得されるレコードインスタンス全体に適用されます。そのため、レコードインスタンスの一部が制限された場合、そのレコードインスタンス全体が除外されます。DBAJOIN ON の場合、DBA 制限は、データ値が表示されるセグメントのみに適用され、レコードインスタンスの残りの部分 (存在する場合) は表示されます。

制限の対象とするファイルまたはセグメントに構造上の親が含まれ、その JOIN が OUTER JOIN または UNIQUE JOIN の場合、この違いが重要になります。

制限がレポートフィルタとして処理される場合、そのフィルタに一致しない下位セグメントインスタンスは、ホストセグメントとともにレポート出力から省略されます。ホストセグメントが省略されるため、出力には OUTER JOIN または UNIQUE JOIN が正しく反映されません。

制限が JOIN 条件として処理される場合、その JOIN 条件に一致しない下位セグメントインスタンスはミッシング値として表示され、レポート出力にはホスト行がすべて表示されます。

JOIN 条件への DBA 制限の追加

```
SET DBAJOIN = {OFF|ON}
```

説明

OFF

レポートリクエストで DBA 制限を WHERE フィルタとして処理します。デフォルト値は OFF です。

ON

DBA 制限を JOIN 条件として処理します。

リレーショナルテーブルでの DBAJOIN 設定の使用

次のリクエストは、2 つのテーブル (EMPINFOSQL および EDINFOSQL) を作成します。

```
TABLE FILE EMPLOYEE
SUM LAST_NAME FIRST_NAME CURR_JOBCODE
BY EMP_ID
ON TABLE HOLD AS EMPINFOSQL FORMAT SQLMSS
END
-RUN
TABLE FILE EDUCFILE
SUM COURSE_CODE COURSE_NAME
BY EMP_ID
```

```
ON TABLE HOLD AS EDINFOSQL FORMAT SQLMSS
END
```

生成された EMPINFOSQL マスターファイルの末尾に、次の DBA 属性を追加します。指定された制限により、USER2 は値が 300 以上のコースコードを取得することはできません。

```
END
DBA=USER1,$
USER=USER2, ACCESS = R, $
FILENAME=EDINFOSQL,$
USER=USER2, ACCESS = R, RESTRICT = VALUE, NAME=SYSTEM, VALUE=COURSE_CODE LT
300;,$
```

生成された EDINFOSQL マスターファイルの末尾に、次の DBA 属性を追加します。

```
END
DBA=USER1,DBAFILE=EMPINFOSQL,$
```

次のリクエストを発行します。

```
SET USER=USER2
SET DBAJOIN=OFF
JOIN LEFT_OUTER EMP_ID IN EMPINFOSQL TO MULTIPLE EMP_ID IN EDINFOSQL AS J1
TABLE FILE EMPINFOSQL
PRINT LAST_NAME FIRST_NAME COURSE_CODE COURSE_NAME
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
GRID=OFF,$
END
```

下図のレポート出力では、値が 300 以上のコースコードのホスト行およびその下位行がすべて省略されています。

LAST_NAME	FIRST_NAME	COURSE_CODE	COURSE_NAME
STEVENS	ALFRED	101	FILE DESCRPT & MAINT
SMITH	MARY	103	BASIC REPORT PREP FOR PROG
JONES	DIANE	203	FOCUS INTERNALS
SMITH	RICHARD	108	BASIC RPT NON-DP MGRS
MCKNIGHT	ROGER	101	FILE DESCRPT & MAINT

生成された SQL では、SELECT ステートメントの WHERE 記述子に DBA 制限が追加されています。

```
SELECT
  T1."EID",
  T1."LN",
  T1."FN",
  T2."CC",
  T2."CD"
   FROM
  EMPINFOSQL T1,
  EDINFOSQL T2
   WHERE
  (T2."EID" = T1."EID") AND
  (T2."CC" < '300;');
```

SET DBAJOIN=ON に設定して、上記のリクエストを再実行します。下図のレポート出力では、すべてのホスト行が表示されていますが、DBA 制限を満たさなかった下位セグメントインスタンスでは、コースコードの代わりにミッシング値が表示されています。

LAST_NAME	FIRST_NAME	COURSE_CODE	COURSE_NAME
STEVENS	ALFRED	101	FILE DESCRPT & MAINT
SMITH	MARY	103	BASIC REPORT PREP FOR PROG
JONES	DIANE	203	FOCUS INTERNALS
SMITH	RICHARD	108	BASIC RPT NON-DP MGRS
BANNING	JOHN	.	.
IRVING	JOAN	.	.
ROMANS	ANTHONY	.	.
MCCOY	JOHN	.	.
BLACKWOOD	ROSEMARIE	.	.
MCKNIGHT	ROGER	101	FILE DESCRPT & MAINT
GREENSPAN	MARY	.	.
CROSS	BARBARA	.	.

生成された SQL では、DBA 制限が JOIN に追加されています。WHERE 記述子はありません。

```
SELECT
  T1."EID",
  T1."LN",
  T1."FN",
  T2."EID",
  T2."CC",
  T2."CD"
   FROM
  ( EMPINFOSQL T1
```

```
    LEFT OUTER JOIN EDINFOSQL T2
    ON T2."EID" = T1."EID" AND
       (T2."CC" < '300;') );
```

JOIN 解析中の一時項目 (DEFINE) の保持

JOIN の解析中に一時項目 (DEFINE) を保存する 2 つの方法があります。1 つ目は KEEPDEFINES の使用、2 つ目は DEFINE FILE SAVE および DEFINE FILE RETURN の使用によるものです。

KEEPDEFINES による一時項目 (DEFINE) の保持

KEEPDEFINES パラメータにより、JOIN コマンドの実行後にホストまたは JOIN 構造に対して DEFINE で作成した一時項目を保持するか、クリアするかを決定します。これは、DEFINE コマンドが JOIN コマンドの前に来るときに適用されます。

前の一時項目 (DEFINE) により、「コンテキスト」と呼ばれるものが構築されます。新しいコンテキストでは、レイヤまたはコマンド環境が新しく作成されます。新しい環境が開始されたときには、前のレイヤで定義されていたすべての一時項目 (DEFINE) は、新しいレイヤで使用できます。一時項目 (DEFINE) の定義を上書きまたはクリアした場合、その影響を受けるのは現在のレイヤのみです。前のレイヤに戻るときは、その一時項目 (DEFINE) 定義に変化はありません。

JOIN コマンドの後に発行した新しい DEFINE フィールドにより、もう 1 つのコンテキストが構築され、それによりコンテキスト群が生成されます。各コンテキストでは、すべての前のコンテキスト中の一時項目 (DEFINE) へのアクセスが可能です。

- デフォルト設定では、KEEPDEFINES は OFF になっています。この設定により JOIN コマンドは前の一時項目 (DEFINE) を削除します。
- KEEPDEFINES を ON に設定後、後続の JOIN コマンドの解析中に一時項目 (DEFINE) が復活します。
- KEEPDEFINES を ALL に設定すると、JOIN コマンドの解析中に、後続のすべての一時項目 (DEFINE) および以前の一時項目 (DEFINE) が復元されます。

JOIN CLEAR *as_name* コマンドは、JOIN *as_name* の発行後に作成したすべてのコンテキストを削除します。

DEFINE に基づく条件付き JOIN では、KEEPDEFINES 設定を ON にする必要があります。さらに、DEFINE に基づく条件付き JOIN コマンドの発行前に、すべての一時項目 (DEFINE) を作成しておく必要があります。これは、JOIN コマンドの後に一時項目 (DEFINE) を作成する従来の DEFINE に基づく JOIN と異なります。さらに、一時項目 (DEFINE) は JOIN 構文または WHERE 構文の一部にすることができます。

新しいファイルコンテキストが作成されるため、JOIN コマンドの後に発行した DEFINE コマンドにより JOIN の前に作成した一時項目 (DEFINE) が置き換えられたり、クリアされたりすることはありません。

KEEPDEFINES の使用

```
SET KEEPDEFINES = {ON|OFF|ALL}
```

説明

ON

JOIN コマンドの実行後、後続の一時項目 (DEFINE) を保持します。

OFF

JOIN コマンドの実行後に一時項目 (DEFINE) をクリアします。デフォルト値は OFF です。

ALL

JOIN コマンドの実行後、リクエスト内でそれ以前および以降の一時項目 (DEFINE) のすべてを保持します。

KEEPDEFINES の使用上の注意

- KEEPDEFINES ON を設定する前に定義した一時項目 (DEFINE) は、JOIN コマンドの発行後は保持されません。KEEPDEFINES を ALL に設定すると、以前に定義した一時項目 (DEFINE) がすべて保持されます。

- 一時項目 (DEFINE) およびネストされた JOIN を指定して KEEPDEFINE コマンドを使用する場合は、各 JOIN 構造に固有の名前を割り当ててください。各 JOIN に固有の AS 名を追加することで、後続の JOIN による前の JOIN ステートメントまたは DEFINE ステートメントの上書きを防止することができます。

KEEPDEFINES による JOIN 解析中の一時項目 (DEFINE) の保持

1 つ目の一時項目 (DEFINE) の DAYSKEPT は、KEEPDEFINES を ON に設定後、任意の JOIN の発行前に定義されます。DAYSKEPT は、ビデオテープの返却日と貸出日の日数の差です。

```
SET KEEPDEFINES = ON
DEFINE FILE VIDEOTRK
DAYSKEPT/I5 = RETURNDATE - TRANSDATE;
END
```

?DEFINE クエリは、これがこの時点で唯一定義された一時項目 (DEFINE) であることを示します。

```
?DEFINE
```

```
FILE     FIELD NAME                  FORMAT  SEGMENT   VIEW       TYPE
VIDEOTRK DAYSKEPT                    I5            4
```

次のリクエストは、貸し出し期間が 2 日のすべてのトランザクションを表示します。

```
TABLE FILE VIDEOTRK
PRINT MOVIECODE TRANSDATE RETURNDATE DAYSKEPT
COMPUTE ACTUAL_DAYS/I2 = RETURNDATE-TRANSDATE;
WHERE DAYSKEPT EQ 2
END
```

出力の最初の数行に、各返却日は取り扱い日の 2 日後であることが表示されます。

```
MOVIECODE  TRANSDATE  RETURNDATE  DAYSKEPT  ACTUAL_DAYS
---------  ---------  ----------  --------  -----------
001MCA     91/06/27   91/06/29           2            2
```

```
692PAR     91/06/27   91/06/29            2             2
259MGM     91/06/19   91/06/21            2             2
```

次に、VIDEOTRK データソースを MOVIES データソースに結合します。?DEFINE クエリは、JOIN により DAYSKEPT 一時項目 (DEFINE) がクリアされなかったことを示します。

```
JOIN  MOVIECODE IN VIDEOTRK TO ALL MOVIECODE IN MOVIES AS J1
?DEFINE
```

```
FILE     FIELD NAME                   FORMAT  SEGMENT   VIEW       TYPE
VIDEOTRK DAYSKEPT                     I5            4
```

次に、新しい一時項目 (DEFINE) である YEARS が、VIDEOTRK と MOVIES の間の JOIN に対して定義されます。

```
DEFINE FILE VIDEOTRK
YEARS/I5 = (TRANSDATE - RELDATE)/365;
END
```

?DEFINE クエリは、JOIN の前に作成した一時項目 (DEFINE) が、別のコンテキストにあるために新しい一時項目 (DEFINE) によりクリアされなかったことを示します。

```
?DEFINE
```

```
FILE     FIELD NAME                    FORMAT  SEGMENT   VIEW     TYPE
VIDEOTRK DAYSKEPT                      I5            4
VIDEOTRK YEARS                         I5            5
```

次に、実際の日数プラス 1 日になるように DAYSKEPT フィールドが再定義されます。

```
DEFINE FILE VIDEOTRK
DAYSKEPT/I5 = RETURNDATE - TRANSDATE + 1;
END
```

?DEFINE クエリは、DAYSKEPT 一時項目 (DEFINE) の 2 つのバージョンがあることを示します。ただし、YEARS は DAYSKEPT の新しいバージョンと同一のコンテキスト (JOIN の後) にあり、DEFINE コマンドにより ADD オプションが指定されれなかったため、クリアされまています。

```
?DEFINE
```

```
FILE     FIELD NAME                    FORMAT  SEGMENT   VIEW     TYPE
VIDEOTRK DAYSKEPT                      I5            4
VIDEOTRK DAYSKEPT                      I5            4
```

同一のリクエストを DAYSKEPT の新しい定義に使用します。返却日と取り扱い日の日数の差は、実際には DAYSKEPT の定義変更のため、2 日ではなく 1 日であることに注目します。

```
MOVIECODE  TRANSDATE  RETURNDATE  DAYSKEPT  ACTUAL_DAYS
---------  ---------  ----------  --------  -----------
040ORI     91/06/20   91/06/21           2            1
505MGM     91/06/21   91/06/22           2            1
710VES     91/06/26   91/06/27           2            1
```

次に、J1 をクリアします。DAYSKEPT の再定義もクリアされます。

```
JOIN CLEAR J1
?DEFINE
```

```
FILE     FIELD NAME                    FORMAT  SEGMENT   VIEW         TYPE
VIDEOTRK DAYSKEPT                      I5            4
```

レポート出力には、DAYSKEPT の元の定義が有効になったことが表示されます。

```
MOVIECODE  TRANSDATE  RETURNDATE  DAYSKEPT  ACTUAL_DAYS
---------  ---------  ----------  --------  -----------
001MCA     91/06/27   91/06/29           2            2
692PAR     91/06/27   91/06/29           2            2
259MGM     91/06/19   91/06/21           2            2
```

DEFINE FILE SAVE と RETURN による一時項目 (DEFINE) の保持

DEFINE FILE SAVE コマンドにより一時項目 (DEFINE) に対して新しいコンテキストが形成され、これは後で DEFINE FILE RETURN により削除することができます。詳細は、一時フィールドの作成を参照してください。

DEFINE FILE SAVE および RETURN による一時項目 (DEFINE) の保持

次のコマンドを使用して、ファイルコンテキスト内で一時項目 (DEFINE) を保持することができます。

```
SET KEEPDEFINES=ON
```

次のコマンドは、一時項目 (DEFINE) A を VIDEOTRK データソースに定義し、使用中のコンテキストに配置します。

```
DEFINE FILE VIDEOTRK
 A/A5='JAWS';
 END
```

次のコマンドは、新しいコンテキストを作成し、そのコンテキストに一時項目 (DEFINE) B を保存します。

```
DEFINE FILE VIDEOTRK SAVE
 B/A5='ROCKY';
 END
?DEFINE
```

? FILTER クエリの一時項目 (DEFINE) A および B が一覧表示されます。

```
FILE     FIELD NAME               FORMAT  SEGMENT   VIEW         TYPE
VIDEOTRK A                        A5
VIDEOTRK B                        A5
```

次の DEFINE コマンドは、一時項目 (DEFINE) C を作成します。ADD オプションを DEFINE コマンドで使用しなかったため、前に定義したすべての一時項目 (DEFINE) はクリアされます。

```
DEFINE FILE VIDEOTRK
 C/A10='AIRPLANE';
 END
?DEFINE
```

? FILTER クエリのC が定義した唯一の一時項目 (DEFINE) であることを示します。

```
FILE     FIELD NAME                FORMAT  SEGMENT   VIEW         TYPE
VIDEOTRK C                         A10
```

次の JOIN コマンドは、新しいコンテキストを作成します。KEEPDEFINES が ON に設定されているため、一時項目 (DEFINE) C は JOIN コマンドによりクリアされません。

```
JOIN MOVIECODE IN VIDEOTRK TAG V1 TO MOVIECODE IN MOVIES TAG M1 AS J1
?DEFINE
```

? FILTER クエリのフィールド C が定義されたままであることを示します。

```
FILE     FIELD NAME                FORMAT  SEGMENT   VIEW         TYPE
VIDEOTRK C                         A10
```

次の DEFINE コマンドは、JOIN コマンドで作成した新しいコンテキストに、一時項目 (DEFINE) D を作成します。

```
DEFINE FILE VIDEOTRK SAVE
 D/A10='TOY STORY';
 END
?DEFINE
```

? FILTER クエリの一時項目 (DEFINE) C および D が定義されていることを示します。

```
FILE     FIELD NAME                FORMAT  SEGMENT   VIEW         TYPE
VIDEOTRK C                         A10
VIDEOTRK D                         A10
```

DEFINE FILE RETURN コマンドにより、使用中のコンテキスト (JOIN の後) で作成した一時項目 (DEFINE) をクリアします。

```
DEFINE FILE VIDEOTRK RETURN
END
?DEFINE
```

? FILTER クエリの一時項目 (DEFINE) D がクリアされ、C がまだ定義されていることを示します。

```
FILE     FIELD NAME                FORMAT  SEGMENT   VIEW         TYPE
VIDEOTRK C                         A10
```

DEFINE FILE SAVE コマンドによりフィールド C が作成されなかったため、次の DEFINE FILE RETURN コマンドは一時項目 (DEFINE) C をクリアしません。

```
DEFINE FILE VIDEOTRK RETURN
END
?DEFINE
```

? FILTER クエリの一時項目 (DEFINE) C が定義されたままであることを示します。

```
FILE     FIELD NAME                FORMAT  SEGMENT   VIEW         TYPE
VIDEOTRK C                         A10
```

注意：DEFINE FILE RETURN は、DEFINE FILE SAVE が有効な場合にのみ有効になります。

条件付き JOIN 式によるセグメントの選別

条件付き JOIN コマンドは、JOIN セグメント、親セグメント、またはその親パスの上位のすべてのフィールドを参照することができます。

これらの JOIN 式が有効なときは、JOIN の存在するセグメント (子または JOIN セグメント) が選別されます。つまり、式により定義したテストをパスする子セグメントがない場合、その JOIN は子セグメントが存在しないときの SET ALL=OFF または SET ALL=ON の規則に従います。TABLE コマンドの WHERE 句と異なり、SET ALL=ON のときは、親セグメントは、JOIN_WHERE による自動選別の対象外になります。

JOIN の WHERE 条件の解析

WHERE 条件は TABLE リクエストがクロスリファレンスセグメントまたはその子に参照された場合のみ、JOIN で有効になります。そのような参照がない場合、WHERE は有効になりません。

AT 属性は、適切な親セグメントまたはホストの適切な子またはクロスリファレンスセグメントへの結合に使用されます。AT パラメータとして使用されるフィールド値は、リンクには使用されません。これらは単にセグメントの参照用に使用します。

注意：有効な WHERE 条件がない場合は直積が与えられます。

JOIN 構造の表示

2 つのデータソースを結合すると、1 つの論理構造として処理されるようになります。この構造は、クロスリファレンスファイルの構造をホストファイルの構造へ追加することにより作成されます。共有値フィールドを含むクロスリファレンスファイルのセグメントは、その共有値フィールドを含むホストファイルの子セグメントになります。

JOIN 構造の表示

JOIN 構造を表示するには、次のコマンドを発行します。

```
CHECK FILE hostfile PICTURE
```

説明

hostfile

ホストファイル名です。

JOIN 構造の表示

ホストファイルに属するセグメントはアスタリスク (*) で示される標準のセグメントとして表示され、クロスリファレンスファイルに属するセグメントは点で示される仮想セグメントとして表示されます。クロスリファレンスファイルのセグメントには、各セグメント以下のクロスリファレンスファイル名でラベルも付けられます。

```
JOIN PIN IN EMPDATA TO PIN IN SALHIST
CHECK FILE EMPDATA PICTURE
0 NUMBER OF ERRORS=     0
  NUMBER OF SEGMENTS=   2  ( REAL=    1  VIRTUAL=   1 )
  NUMBER OF FIELDS=    14  INDEXES=   1  FILES=     2
  NUMBER OF DEFINES=    1
  TOTAL LENGTH OF ALL FIELDS=  132
1SECTION 01.01
              STRUCTURE OF FOCUS    FILE EMPDATA  ON 03/05/01 AT 12.22.49
```

```
          EMPDATA
 01       S1
**************
*PIN          **I
*LASTNAME     **
*FIRSTNAME    **
*MIDINITIAL   **
*             **
***************
 **************
       I
       I
       I
       I SLHISTRY
 02    I KU
..............
:PIN          :K
:EFFECTDATE   :
:OLDSALARY    :
:             :
:             :
:............:
 JOINED  SALHIST
```

クロスリファレンスファイル構造の最上位セグメントに、共有値フィールドが含まれています。このセグメントがルートセグメントでない場合、そのクロスリファレンスファイル構造は代替ファイルビューのように反転します。

JOIN 構造のクロスリファレンスファイルセグメントのタイプは次のとおりです。

- ユニーク JOIN 構造では、最上位のクロスリファレンスファイルセグメントにセグメントタイプ KU があります。そのユニーク子セグメントにはセグメントタイプ KLU があり、非ユニーク子セグメントにはセグメントタイプ KL があります。
- 非ユニーク JOIN 構造には、最上位のクロスリファレンスファイルセグメントにセグメントタイプ KM があります。そのユニーク子セグメントにはセグメントタイプ KLU があり、非ユニーク子セグメントにはセグメントタイプ KL があります。

ホストファイル構造に変更はありません。そのクロスリファレンスファイルは、独立して使用することができます。

JOIN 構造の一覧表示

JOIN データソースを一覧表示するには、次のコマンドを発行します。

```
?JOIN
```

これにより、有効なすべての JOIN コマンドが表示されます。以下はその例です。

```
  JOINS CURRENTLY ACTIVE

HOST                        CROSSREFERENCE
FIELD      FILE      TAG    FIELD      FILE     TAG    AS       ALL  WH
-----      ----      ---    -----      ----     ---    --       ---  --
JOBCODE    EMPLOYEE         JOBCODE    JOBFILE                   N    N
```

JOIN 構造に JOIN 名がない場合、AS 句は省略されます。2 つのデータソースが複数の JOIN コマンドにより結合される場合は、発行した 1 つ目のコマンドのみが表示されます。WH フィールドの N は JOIN が条件付きでないことを示します。Y は JOIN が条件付きであることを示します。

JOIN 構造のクリア

特定の JOIN 構造、または存在するすべての構造をクリアすることができます。クリアにより、割り当てた JOIN は無効になります。条件付きの JOIN をクリアすると、同一のホストファイルを使用してその JOIN 以降に発行したすべての JOIN もクリアされます。

ヒント：クリアする前に JOIN を一覧表示する、またはすべての有効 JOIN 構造の詳細を表示するには、クエリコマンド ? JOIN を発行します。詳細は、JOIN 構造の一覧表示を参照してください。

JOIN のクリア

JOIN 構造をクリアするには、次のコマンドを発行します。

```
JOIN CLEAR {joinname|*}
```

説明

joinname

クリアする JOIN 構造の AS 名です。

*

すべての JOIN 構造をクリアします。

条件付き JOIN のクリア

JOIN CLEAR コマンドを発行することにより JOIN をクリアすることができます。JOIN CLEAR コマンドの影響は条件付き JOIN が存在するかどうかにより異なります。

- クリアする JOIN の後に条件付き JOIN を特定して発行する、またはクリアする JOIN が条件付き JOIN の場合、JOIN CLEAR *as_name* コマンドによりすべての JOIN が指定した JOIN の後で削除されます。
- クリアする JOIN の後に条件付き JOIN が発行されない場合、指定の JOIN のみがクリアされます。クリアする JOIN のコンテキストで保存したすべての一時項目 (DEFINE) もクリアされます。条件付き JOIN の位置により、通常の JOIN にはクリアされるものとされないものがあります。JOIN CLEAR * コマンドにより、発行したすべての JOIN およびその関連一時項目 (DEFINE) がクリアされます。ただし、NULL コンテキストのすべての一時項目 (DEFINE) は保持されます。

注意：NULL コンテキストは、JOIN が発行される前の、データソースのコンテキストです。

JOIN のクリア

次のリクエストは VIDEOTRK をホストデータソースとして使用し、3 つの JOIN を作成します。最初の 2 つは条件付き (JW1、JW2) で、3 つ目の JOIN は無条件 (J1) です。

```
JOIN FILE VIDEOTRK AT PRODCODE TO ALL
     FILE GGSALES  AT PCD AS JW1
WHERE PRODCODE NE PCD;
END
JOIN  FILE VIDEOTRK AT TRANSDATE TO ALL
      FILE MOVIES   AT RELDATE   AS JW2
WHERE (TRANSDATE - RELDATE)/365 GT 10;
END
JOIN MOVIECODE IN VIDEOTRK TO MOVIECODE IN MOVIES AS J1
```

次のリクエストは MOVIES をホストデータソースとして使用して、条件付き JOIN (JW3) を作成します。

```
JOIN  FILE MOVIES   AT MOVIECODE TO ONE
      FILE VIDEOTRK AT TRANSDATE AS JW3
WHERE (TRANSDATE - RELDATE)/365 LT 2;
END
```

最終リクエストにより、VIDEOTRK をホストデータソースとして使用する 4 つ目の条件付き JOIN (JW4) を作成します。

```
JOIN  FILE VIDEOTRK AT LASTNAME  TO ALL
      FILE EMPLOYEE AT LAST_NAME AS JW4
WHERE LASTNAME GE LAST_NAME;
END
```

これらの JOIN の実行後の ? JOIN クエリの出力結果は次のとおりです。

```
?JOIN
 JOINS CURRENTLY ACTIVE

HOST                          CROSSREFERENCE
FIELD       FILE     TAG    FIELD       FILE      TAG   AS      ALL  WH
-----       ----     ---    -----       ----      ---   --      ---  --
PRODCODE    VIDEOTRK        PCD         GGSALES         JW1      Y    Y
TRANSDATE   VIDEOTRK        RELDATE     MOVIES          JW2      Y    Y
MOVIECODE   VIDEOTRK        MOVIECODE   MOVIES          J1       N    N
MOVIECODE   MOVIES          TRANSDATE   VIDEOTRK        JW3      N    Y
LASTNAME    VIDEOTRK        LAST_NAME   EMPLOYEE        JW4      Y    Y
```

JW2 をクリアすると、JW2 の後に発行した同一ホストデータソースを使用するすべての JOIN がクリアされます。JW1 は JW2 の前に発行されたために残り、JW3 は異なるホストデータソースを使用するために残ります。

```
JOIN CLEAR JW2
?JOIN
 JOINS CURRENTLY ACTIVE

HOST                          CROSSREFERENCE
FIELD        FILE     TAG    FIELD        FILE     TAG      AS   ALL WH
-----        ----     ---    -----        ----     ---      --   --- --
PRODCODE     VIDEOTRK        PCD          GGSALES          JW1   Y   Y
MOVIECODE    MOVIES          TRANSDATE    VIDEOTRK         JW3   N   Y
```

データソースの結合 (MATCH)

データ構造の内容を MATCH コマンドで結合したり、データソースを MORE 句で連結することで、結合されたレポートデータを使用して、レポートを実行することができます。

データの結合

MATCH コマンドを使用すると、統合およびソートの対象レコードを指定して、複数のデータソースを統合することができます。コマンドにより新しいデータソースである HOLD ファイルが作成されます。選択したレコードの結合はこのファイルで実行されます。新しく作成されたデータソースは、他の HOLD ファイル同様に、レポートの作成に使用することが可能です。

MATCH コマンドでソートフィールドを指定することで、新しいデータソースに統合するレコードを選択します。ソートフィールド群は BY 句により指定します。第 1 データソースに対するソートフィールド群を指定し、さらに第 2 データソースに対するソートフィールド群を指定します。MATCH コマンドは、両方のデータソースに共通して指定したすべてのソートフィールドを比較し、第 2 データソースのソート値と一致するソート値を持つ、第 1 データソースのすべてのレコードを新しい HOLD ファイルに統合します。最大 128 個のソートセットを指定することができます。この値には、共通ソートフィールドの数も含まれます。

値が共通するデータソースレコードの結合だけではなく、他の関係に基づいてレコードを結合することも可能です。たとえば、他方のデータソースに一致するソート値がないレコードを各データソースから検索し、これらを結合することができます。もう 1 つの結合タイプとしては、第 1 データソースのすべてのレコードを、第 2 データソースで一致する任意のレコードと結合する方法があります。

1 つの MATCH ウィザードで最大 16 組のデータ結合を実行することができます。たとえば、異なるデータソース間の結合や同一のデータソース内の結合などが実行可能です。

注意：最大 16 組という制限は、最も複雑なリクエストに適用されます。単純なリクエストでは、それ以上のデータソースの結合が可能な場合があります。

データソースの結合

MATCH コマンドの構文は TABLE コマンドの構文と類似しています。

```
MATCH FILE file1
.
.
.
RUN
FILE file2
.
.
.
[AFTER MATCH merge_phrase]
RUN
FILE file3
.
.
.
[AFTER MATCH merge_phrase]
END
```

説明

file1

MATCH が要求されたレコードを検索する1 つ目のデータソースです。

merge_phrase

ファイルから検索したレコードの比較方法を指定します。

file2/file3

MATCH が要求されたレコードを検索する追加データソースです。

RUN コマンドは、各 AFTER MATCH コマンドの後に続ける必要があります (最後を除く)。END コマンドは、最後の AFTER MATCH コマンドの後に続ける必要があります。

MATCH は、HOLD ファイルを作成します。ワイルドカード文字 (*) を含む PRINT コマンドを使用して、HOLD ファイルの内容を表示することができます。

MATCH 処理のタイプ

MATCH 処理には、グループ化および非グループ化という、2 つのタイプがあります。グループか処理は、新しい MATCH 処理で、この処理がデフォルト設定で使用されます。非グループ化処理は、従来の MATCH 処理です。従来の処理を呼び出す必要がある場合は、SET MATCHCOLUMNORDER = UNGROUPED コマンドを使用します。

すべての MATCH リクエストには、次のことが適用されます。

- ソースとターゲット (OLD および NEW) は、共通の上位 BY フィールドの一致に基づいて結合されます。OLD ファイルと NEW ファイルの間に実際の一致が成立するには、上位の BY フィールドが同一である必要があります。共通の上位 BY フィールドが存在しない場合、レコード単位での連結が実行されます。
- OLD 側および NEW 側から選択される出力は、AFTER MATCH コマンドに基づきます。

MATCH の出力段階は、処理方法 (グループ化、非グループ化) によって異なります。

- グループ化処理では、出力ファイルには、リクエストで指定した順序で、各共通ソートキー (BY フィールド) が表示フィールドに続きます。結果として、非グループ化処理では、Flat File 出力に限定されます。
- 一方、グループ化処理では、リクエストのフィールドは、出力の最上位共通ソートフィールドでグループ化されます。これにより、結合の各側で複数表示コマンドの使用が可能になることから、FOCUS または XFOCUS フォーマットの複数セグメント階層出力ファイルの生成が可能になります。実際には、対応するマスターファイルが存在する場合、出力ファイルは任意のフォーマットで作成することができます。

MATCH の両側 (OLD および NEW) で同一ソートキーを使用する標準の MATCH リクエストの場合、グループ化処理と非グループ化処理は、同一の出力を生成します。

ただし、結合の各側で複数の表示コマンドまたは異なるソートフィールド使用するリクエストでは、ソートキーによるフィールドのグループ化によって、従来の処理とは異なるフィールド順序の出力が生成されます。新しい処理が従来の結果とは異なる出力を生成し、これが期待する結果ではない場合は、SET MATCHCOLUMNORDER コマンドを使用して、従来の結果に戻すことができます。

たとえば、MATCH を使用して、異なるソートに基づく集計列を含む製品リストの出力を作成する場合、MATCHCOLUMNORDER=UNGROUPED に設定することで、列出力の順序を、従来と同一にすることができます。

MATCH によるデータの結合方法は、リクエスト、BY フィールド、表示コマンド、処理タイプ、使用する結合句で指定するデータソースの順序によって異なります。ただし、一般的な処理は次のとおりです。

1. 指定された 1 つ目のデータソースから、要求されたレコードを検索し、一時作業領域

へ書き込みます。

2. 指定された 2 つ目のデータソースから、要求されたレコードを検索し、一時作業領域へ書き込みます。
3. 結合句の指定 (例、OLD-OR-NEW) に基づいて、取得されたレコードの共通上位ソートフィールドが比較されます。
4. デフォルト設定のグループ化処理が有効な場合、フィールドが共通ソートフィールド下に再編成されます。
5. それ以上の MATCH 処理がある場合、結合した比較結果を一時データソースに書き込みます。END が出現するまで、指定されたすべてのデータソースで、処理を繰り返します。
6. 最終レコードを HOLD ファイルに書き込みます。

MATCH 処理の制御

```
SET MATCHCOLUMNORDER = {GROUPED|UNGROUPED}
```

説明

GROUPED

出力ファイルのフィールドを、共通上位ソートフィールド下にグループ化します。これがデフォルト値です。

UNGROUPED

出力ファイルのフィールドを、共通上位ソートフィールドでグループ化せず、MATCH リクエストで指定されたとおりに配置します。

MATCH リクエスト使用上の注意

- 非グループ化処理では、MATCH で生成される HOLD ファイルのフォーマットを指定することはできません。HOLD ファイルは、HOLDFORMAT パラメータ値に応じて、単一セグメントの BINARY または ALPHA フォーマットで作成されます。この結合プロセスによって、元のデータソースが変更されることはありません。

- MATCH リクエストから作成される HOLD マスターファイルでは、連続番号付きのエイリアス (E01、E02...) が割り当てられます。MATCH コマンドで同一フィールド名が複数回使用される場合、HOLD ファイルを使用するリクエストで、フィールド名の代わりにこれらのエイリアスを参照することで、違いを区別することができます。

  グループ化処理では、マスターファイルのフィールドが再編成されるため、このエイリアス名は、非グループ化処理で割り当てられたものとは異なるフィールドを表します。これにより、処理方法を変更すると、生成される結果が異なる場合があります。

  エイリアス名の使用を回避するには、MATCH リクエストで AS 句を使用して、固有のフィールド名を作成します (常に同一名にする必要がある共通上位 BY フィールドを除く)。

- ACROSS、BY HIGHEST/LOWEST n、IN-GROUPS-OF、WHERE TOTAL、IF TOTAL 句、および COMPUTE コマンドは、MATCH リクエストで使用することができません。ただし、DEFINE コマンドは使用することができます。

- MATCH リクエストごとに、合計 128 個の BY 句、および最大数の表示フィールドを使用できます。ソートフィールド群の個数には、共通ソートフィールドの数も含まれます。表示フィールドの最大数は、いくつかの要因の組み合わせにより決定されます。

  詳細は、レポートデータの表示を参照してください。

- MATCH リクエストで使用する各ファイルには、少なくとも 1 つの BY フィールドが必要です。

- SET HOLDLIST パラメータを MATCH と併用すると、HOLDLIST を ALL に設定したように動作します。

- MATCH リクエストでは、演算接頭語の DST.、DST.CNT.、RNK.、ST.、CT. はサポートされません。

データソースの結合 (MATCH)

次のリクエストでは、上位ソートフィールドが、両方のファイルで共通であるため、グループ化処理と非グループ化処理のどちらを使用した場合でも、結果は同一になります。

```
MATCH FILE EDUCFILE
SUM COURSE_CODE
BY EMP_ID
RUN
FILE EMPLOYEE
SUM LAST_NAME AND FIRST_NAME
BY EMP_ID BY CURR_SAL
```

```
AFTER MATCH HOLD OLD-OR-NEW
END
-*****************************
-*  PRINT CONTENTS OF HOLD FILE
-*****************************
TABLE FILE HOLD
PRINT *
END
```

この例で使用した結合句は OLD-OR-NEW です。つまり、1 つ目の (old) データソースからのレコードと、2 つ目の (new) データソースからのレコードの両方が HOLD ファイルに表示されます。

出力結果は次のとおりです。

```
EMP_ID     COURSE_CODE         CURR_SAL  LAST_NAME        FIRST_NAME
------     -----------         --------  ---------        ----------
071382660  101              $11,000.00  STEVENS          ALFRED
112847612  103              $13,200.00  SMITH            MARY
117593129  203              $18,480.00  JONES            DIANE
119265415  108               $9,500.00  SMITH            RICHARD
119329144                   $29,700.00  BANNING          JOHN
123764317                   $26,862.00  IRVING           JOAN
126724188                   $21,120.00  ROMANS           ANTHONY
212289111  103                   $.00
219984371                   $18,480.00  MCCOY            JOHN
315548712  108                   $.00
326179357  301              $21,780.00  BLACKWOOD        ROSEMARIE
451123478  101              $16,100.00  MCKNIGHT         ROGER
543729165                    $9,000.00  GREENSPAN        MARY
818692173  302              $27,062.00  CROSS            BARBARA
```

グループ化と非グループ化処理の比較

次の MATCH リクエストには、各ファイルに対して、2 つの SUM コマンドおよび 1 つの PRINT コマンドが含まれています。すべてのソートフィールドは、両方のファイルで共通です。従来の処理を呼び出す SET MATCHCOLUMNORDER = UNGROUPED コマンドが発行されています。

```
SET MATCHCOLUMNORDER = UNGROUPED
MATCH FILE GGSALES
SUM DOLLARS
```

```
BY ST
SUM BUDDOLLARS BY ST BY CITY
PRINT UNITS BY ST BY CITY BY CATEGORY
RUN
FILE GGSALES
SUM DOLLARS
BY ST
SUM BUDDOLLARS BY ST BY CITY
PRINT BUDUNITS BY ST BY CITY BY CATEGORY
AFTER MATCH HOLD OLD-OR-NEW
END
TABLE FILE HOLD
PRINT *
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, SIZE=9,$
ENDSTYLE
END
```

HOLD マスターファイルは次のとおりです。ソートフィールドは両方のファイルで共通するため、2 つのファイルはこれらのフィールドに基づいて結合されます。ただし、マスターファイルでのフィールドの順序は、リクエストでの順序に従います。最上位ソートフィールドの後にその表示フィールドが続き、次のソートフィールドの後にその表示ソートフィールドが続きます (以降同様)。

```
FILENAME=HOLD, SUFFIX=FIX     , IOTYPE=BINARY, $
  SEGMENT=HOLD, SEGTYPE=S1, $
    FIELDNAME=ST, ALIAS=E01, USAGE=A02, ACTUAL=A04, $
    FIELDNAME=DOLLARS, ALIAS=E02, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=CITY, ALIAS=E03, USAGE=A20, ACTUAL=A20, $
    FIELDNAME=BUDDOLLARS, ALIAS=E04, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=CATEGORY, ALIAS=E05, USAGE=A11, ACTUAL=A12, $
    FIELDNAME=UNITS, ALIAS=E06, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=DOLLARS, ALIAS=E07, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=BUDDOLLARS, ALIAS=E08, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=BUDUNITS, ALIAS=E09, USAGE=I08, ACTUAL=I04, $
```

下図は、出力結果の一部を示しています。

ST	DOLLARS	CITY	BUDDOLLARS	CATEGORY	UNITS	DOLLARS	BUDDOLLARS	BUDUNITS
CA	7642261	Los Angeles	3669484	Coffee	1667	7642261	3669484	1464
CA	7642261	Los Angeles	3669484	Coffee	1663	7642261	3669484	1464
CA	7642261	Los Angeles	3669484	Coffee	1519	7642261	3669484	1464
CA	7642261	Los Angeles	3669484	Coffee	1452	7642261	3669484	1520
CA	7642261	Los Angeles	3669484	Coffee	1663	7642261	3669484	1463
CA	7642261	Los Angeles	3669484	Coffee	1508	7642261	3669484	1539
CA	7642261	Los Angeles	3669484	Coffee	1615	7642261	3669484	1759
CA	7642261	Los Angeles	3669484	Coffee	1449	7642261	3669484	1494
CA	7642261	Los Angeles	3669484	Coffee	1208	7642261	3669484	1038
CA	7642261	Los Angeles	3669484	Coffee	1272	7642261	3669484	1512

UNGROUPED 設定を GROUPED に変更すると、次のマスターファイルが生成されます。両方のファイルで同一の共通ソートフィールドを持つフィールドは、マスターファイルのこれらのソートフィールドの下に移動されています。

```
FILENAME=HOLD, SUFFIX=FIX     , IOTYPE=BINARY, $
  SEGMENT=HOLD, SEGTYPE=S1, $
    FIELDNAME=ST, ALIAS=E01, USAGE=A02, ACTUAL=A04, $
    FIELDNAME=DOLLARS, ALIAS=E02, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=DOLLARS, ALIAS=E03, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=CITY, ALIAS=E04, USAGE=A20, ACTUAL=A20, $
    FIELDNAME=BUDDOLLARS, ALIAS=E05, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=BUDDOLLARS, ALIAS=E06, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=CATEGORY, ALIAS=E07, USAGE=A11, ACTUAL=A12, $
    FIELDNAME=UNITS, ALIAS=E08, USAGE=I08, ACTUAL=I04, $
    FIELDNAME=BUDUNITS, ALIAS=E09, USAGE=I08, ACTUAL=I04, $
```

下図は、出力結果の一部を示しています。

ST	DOLLARS	DOLLARS	CITY	BUDDOLLARS	BUDDOLLARS	CATEGORY	UNITS	BUDUNITS
CA	7642261	7642261	Los Angeles	3669484	3669484	Coffee	1667	1464
CA	7642261	7642261	Los Angeles	3669484	3669484	Coffee	1663	1464
CA	7642261	7642261	Los Angeles	3669484	3669484	Coffee	1519	1464
CA	7642261	7642261	Los Angeles	3669484	3669484	Coffee	1452	1520
CA	7642261	7642261	Los Angeles	3669484	3669484	Coffee	1663	1463
CA	7642261	7642261	Los Angeles	3669484	3669484	Coffee	1508	1539
CA	7642261	7642261	Los Angeles	3669484	3669484	Coffee	1615	1759
CA	7642261	7642261	Los Angeles	3669484	3669484	Coffee	1449	1494
CA	7642261	7642261	Los Angeles	3669484	3669484	Coffee	1208	1038
CA	7642261	7642261	Los Angeles	3669484	3669484	Coffee	1272	1512

MATCHCOLUMNORDER で GROUPED を使用するリクエストでは、HOLD コマンドを次のように変更することで、FORMAT FOCUS 出力ファイルを生成することができます。

```
AFTER MATCH HOLD FORMAT FOCUS OLD-OR-NEW
```

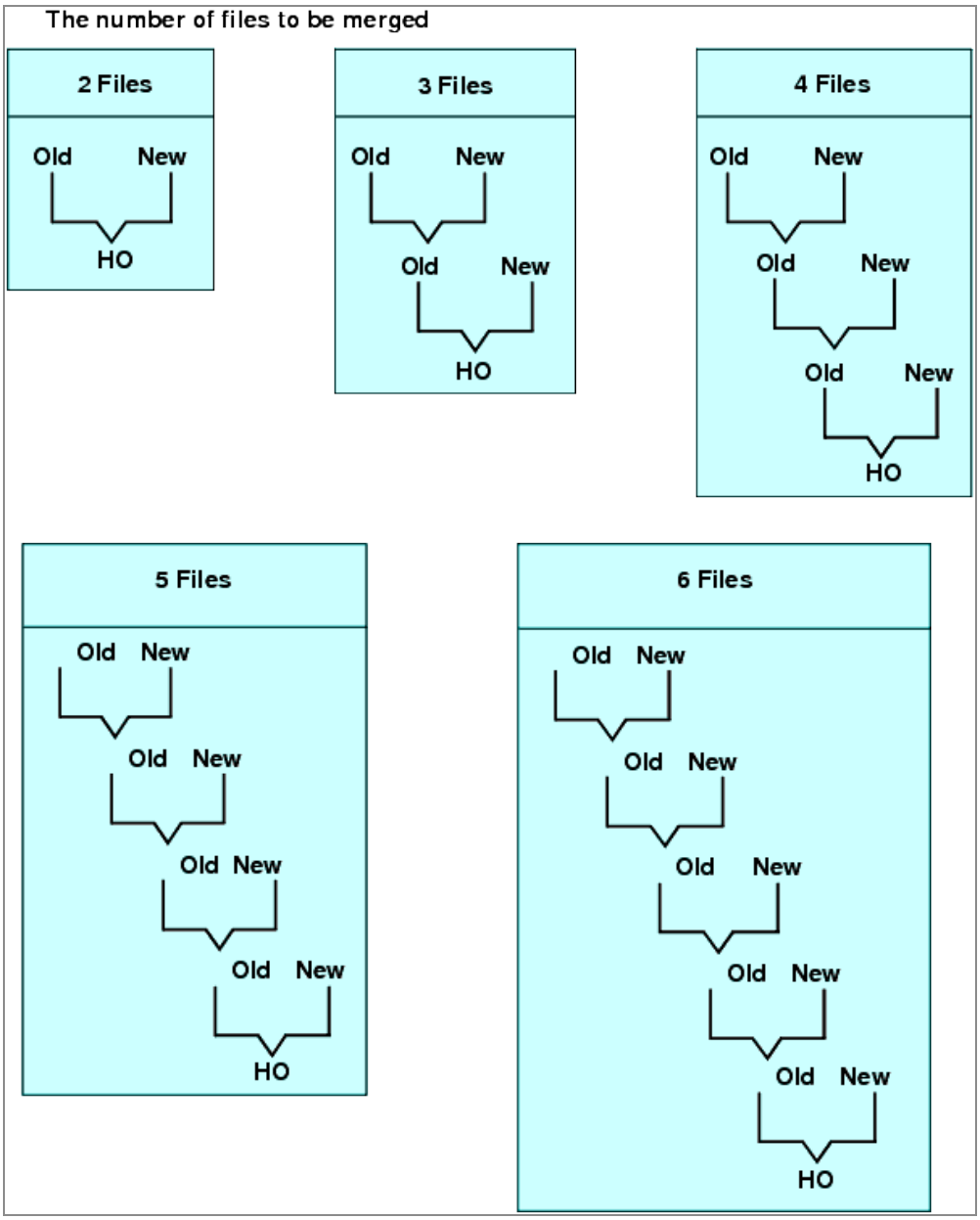
次の階層複数セグメントマスターファイルが生成されます。

```
FILENAME=HOLD    , SUFFIX=FOC     , $
  SEGMENT=SEG01, SEGTYPE=S1, $
    FIELDNAME=ST, ALIAS=E01, USAGE=A02,
      TITLE='State', DESCRIPTION='State', $
    FIELDNAME=DOLLARS, ALIAS=E02, USAGE=I08,
      TITLE='Dollar Sales', DESCRIPTION='Total dollar amount of reported
sales', $
    FIELDNAME=DOLLARS, ALIAS=E03, USAGE=I08,
      TITLE='Dollar Sales', DESCRIPTION='Total dollar amount of reported
sales', $
  SEGMENT=SEG02, SEGTYPE=S1, PARENT=SEG01, $
    FIELDNAME=CITY, ALIAS=E04, USAGE=A20,
      TITLE='City', DESCRIPTION='City', $
    FIELDNAME=BUDDOLLARS, ALIAS=E05, USAGE=I08,
      TITLE='Budget Dollars', DESCRIPTION='Total sales quota in dollars', $
    FIELDNAME=BUDDOLLARS, ALIAS=E06, USAGE=I08,
      TITLE='Budget Dollars', DESCRIPTION='Total sales quota in dollars', $
  SEGMENT=SEG03, SEGTYPE=S2, PARENT=SEG02, $
    FIELDNAME=CATEGORY, ALIAS=E07, USAGE=A11,
      TITLE='Category', DESCRIPTION='Product category', $
    FIELDNAME=FOCLIST, ALIAS=E08, USAGE=I5, $
    FIELDNAME=UNITS, ALIAS=E09, USAGE=I08,
      TITLE='Unit Sales', DESCRIPTION='Number of units sold', $
  SEGMENT=SEG04, SEGTYPE=S1, PARENT=SEG03, $
    FIELDNAME=FOCLIST, ALIAS=E10, USAGE=I5, $
    FIELDNAME=BUDUNITS, ALIAS=E11, USAGE=I08,
      TITLE='Budget Units', DESCRIPTION='Number of units budgeted', $
```

結合句

MATCH 論理は「OLD」/「NEW」というデータソースの概念に基づいています。「OLD」はリクエストで指定した 1 つ目のデータソースで、「NEW」は 2 つ目のデータソースです。各結合の結果により END コマンドが出現するまで HOLD ファイルが作成されます。

下図は、一般的な結合プロセスを示しています。



結合句の指定

```
AFTER MATCH HOLD [AS 'name'] mergetype
```

説明

AS 'name'

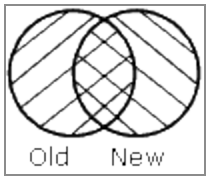
MATCH コマンドで作成する抽出データソースの名前を指定します。デフォルト値は HOLD です。

mergetype

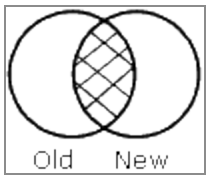
ファイルから検索したレコードの比較方法を指定します。

各句の結果は、ベン図を使用した図で表わされます。この図では、左の円が「OLD」データソース、右の円が「NEW」データソース、斜線部分が HOLD ファイルに書き込まれるデータを、それぞれ表します。

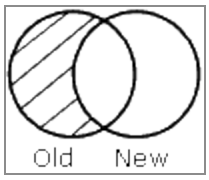
OLD-OR-NEW 「OLD」データソースのレコードすべて、および「NEW」データソースのレコードすべてを HOLD ファイルに格納することを指定します。AFTER MATCH 行を省略する場合、これがデフォルト値です。



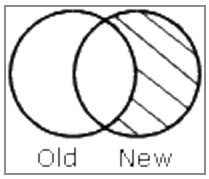
OLD-AND-NEW 「OLD」データソースと「NEW」データソースの両方に存在するレコードを、HOLD ファイルに格納することを指定します (交差集合)。



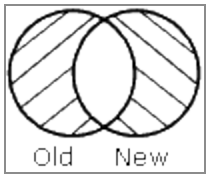
OLD-NOT-NEW 「OLD」データソースのみに存在するレコードを、HOLD ファイルに格納することを指定します。



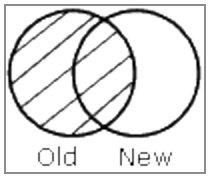
NEW-NOT-OLD 「NEW」データソースのみに存在するレコードを、HOLD ファイルに格納します。



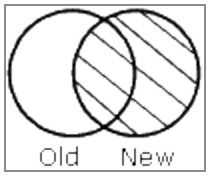
OLD-NOR-NEW 「OLD」データソースに存在するが「NEW」データソースに存在しないレコード、または「NEW」データソースに存在するが「OLD」データソースに存在しないレコードのみを、HOLD ファイルに格納することを指定します (両方のデータソースからの不一致レコードの完全集合)。



OLD 「OLD」データソースのレコードすべて、および「NEW」データソースの一致レコードを、HOLD ファイルに結合することを指定します。



NEW 「OLD」データソースのレコードすべて、および「NEW」データソースの一致レコードを、HOLD ファイルに結合することを指定します。



共通上位ソートフィールドによる MATCH プロセス

両方のデータソースに使用する (「共通上位ソートフィールド」と呼ばれる) 1 つ目の BY フィールドが同一になるよう MATCH を構築すると、その MATCH により共通上位ソートフィールドの値が比較されます。一連のソートフィールド群がすべて両方のファイルで共通の場合は、すべてが比較されます。

少なくとも 1 組のソートフィールド群が必要です。フィールドフォーマットは同一の必要があります。DEFINE コマンドを使用してフィールドフォーマットを再定義できる場合もあります。フィールド名が異なる場合は、AS 句を使用して 2 つ目のソートフィールドが 1 つ目と一致するように名前を変更します。MATCH リクエストで AS 句を使用すると、指定したフィールドは、結果の HOLD ファイルで再び自動的に名前が変更されます。

共通のソートフィールドでファイルを結合する場合は、次の前提条件が適用されます。

- ソートフィールドの 1 つがもう一方の部分の場合、1 対 n の関係が想定されます。
- ソートフィールドの 1 つが一方の部分でない場合、1 対 1 の関係が想定されます。1 つの一致レコードのみが検索されます。

共通上位ソートフィールドによる MATCH プロセス

共通上位ソートフィールドを明確に理解するために、次のデータソースのいくつかのデータについて考察します。

EMPLOYEE データソース		EDUCFILE データソース	
EMP_ID	**LAST_NAME**	**EMP_ID**	**COURSE_CODE**
071382660	STEVENS	071382660	101
119329144	BANNING	212289111	103
112847612	SMITH	112847612	103

MATCH リクエストは次のとおりです。

```
MATCH FILE EMPLOYEE
SUM LAST_NAME BY EMP_ID
RUN
FILE EDUCFILE
SUM COURSE_CODE BY EMP_ID
AFTER MATCH HOLD OLD-OR-NEW
END
```

次の MATCH プロセスが実行されます。

- 共通上位ソートフィールド (EMP_ID) があるため、EMPLOYEE および EDUCFILE ファイルのレコードの EMP_ID 値を一致させることにより、MATCH 論理を開始します。
- EMP_ID の値が 071382660 であるレコードが両方のファイルにあります。一致するため、このレコードは HOLD ファイルに書き込まれます。

```
Record n:071382660 STEVENS 101
```

- EMP_ID の値が 112847612 であるレコードが両方のファイルにあります。一致するため、このレコードは HOLD ファイルに書き込まれます。

```
Record n:112847612 SMITH 103
```

- EMPLOYEE ファイルの EMP_ID の値は 119329144 であり、EDUCFILE ファイルの EMP_ID の値は 212289111 であるため、レコードは一致しません。低い方の値を含むレコードが HOLD ファイルに書き込まれ、ミッシング値にはブランクが挿入されます。

```
Record n:119329144 BANNING
```

- 同様に、212289111 番目のレコードは EDUCFILE ファイルにのみ存在し、次のように書き込まれます。

```
Record n:212289111 103
```

次のコードは、HOLD ファイルのレコードからレポートを作成します。

```
TABLE FILE HOLD
PRINT *
END
```

出力結果は次のとおりです。

EMP_ID	LAST_NAME	COURSE_CODE
071382660	STEVENS	101

112847612	SMITH	103
117593129	JONES	203
119265415	SMITH	108
119329144	BANNING	
123764317	IRVING	
126724188	ROMANS	
212289111		103
219984371	MCCOY	
315548712		108
326179357	BLACKWOOD	301
451123478	MCKNIGHT	101
543729165	GREENSPAN	
818692173	CROSS	302

共通上位ソートフィールドによる結合

次のリクエストは、EMPLOYEE データソースと EMPDATA データソースのデータを結合します。ソートフィールドは EID および PIN です。

```
MATCH FILE EMPLOYEE
PRINT LN FN DPT
BY EID
RUN
FILE EMPDATA
PRINT LN FN DEPT
BY PIN
AFTER MATCH HOLD OLD-OR-NEW
END

TABLE FILE HOLD
PRINT *
END
```

共通上位ソートフィールドのない結合

共通上位ソートフィールドがない場合は、MATCH はレコード単位で実行されます。次のリクエストは、一致データを検索し、HOLD ファイルを作成します。

```
MATCH FILE EMPLOYEE
PRINT LAST_NAME AND FIRST_NAME
BY EMP_ID
RUN
FILE EMPDATA
PRINT PIN
BY LASTNAME
BY FIRSTNAME
AFTER MATCH HOLD OLD-OR-NEW
END

TABLE FILE HOLD
PRINT *
END
```

2 つのデータソースから検索したレコードは、HOLD ファイルに書き込まれます。値は比較されません。出力結果は次のとおりです。

EMP_ID	LAST_NAME	FIRST_NAME	LASTNAME	FIRSTNAME	PIN
071382660	STEVENS	ALFRED	ADAMS	RUTH	000000040
112847612	SMITH	MARY	ADDAMS	PETER	000000050
117593129	JONES	DIANE	ANDERSON	TIM	000000100
119265415	SMITH	RICHARD	BELLA	MICHAEL	000000020
119329144	BANNING	JOHN	CASSANOVA	LOIS	000000030
123764317	IRVING	JOAN	CASTALANETTA	MARIE	000000270
126724188	ROMANS	ANTHONY	CHISOLM	HENRY	000000360
219984371	MCCOY	JOHN	CONRAD	ADAM	000000250

326179357	BLACKWOOD	ROSEMARIE	CONTI	MARSHALL	000000410
451123478	MCKNIGHT	ROGER	CVEK	MARCUS	000000130
543729165	GREENSPAN	MARY	DONATELLO	ERICA	000000320
818692173	CROSS	BARBARA	DUBOIS	ERIC	000000210
			ELLNER	DAVID	000000380
			FERNSTEIN	ERWIN	000000350
			GORDON	LAURA	000000180
			GOTLIEB	CHRIS	000000340
			GRAFF	ELAINE	000000390

			HIRSCHMAN	ROSE	000000160
			KASHMAN	YOLANDA	000000240
			LASTRA	KAREN	000000200
			LEWIS	CASSANDRA	000000220
. . .					

MATCH プロセスの細部調整

PRINT および SUM コマンドを使用して MATCH プロセスを細部調整することができます。これらの相違を理解するには、1 対 n の関係を理解する必要があります。SUM は複数のレコードから単一のレコードを生成し、PRINT は各個別レコードを表示します。BY フィールドの適切な選択により、SUM コマンドのみを使用して PRINT で生成する結果と同一の結果を得ることができます。

MATCH プロセスでの表示コマンドの使用

MATCH プロセスの PRINT および SUM の効果を理解するために、データソース A と B、および一連のリクエストについて考察します。

```
      A                B

F1   F2   F3       F1   F4   F5

1    x    100      1    a    10
2    y    200      1    b    20
                   2    c    30
                   2    d    40
```

リクエスト 1　このリクエストは、ファイル A の F2 フィールドおよび F3 フィールドを集計し、ファイル B の F4 フィールドおよび F5 フィールドを集計後、F1 を共通上位ソートフィールドとして使用します。

```
MATCH FILE A
SUM F2 AND F3 BY F1
RUN
FILE B
SUM F4 AND F5 BY F1
AFTER MATCH HOLD OLD-OR-NEW
END
```

HOLD ファイルに含まれるデータは次のとおりです。

```
F1    F2    F3    F4    F5

1     x     100   b     30
2     y     200   d     70
```

結果ファイルには、共通の各上位ソートフィールドに対して、1 つのレコードのみが含まれます。

リクエスト 2　このリクエストは、ファイル A の F2 フィールドおよび F3 フィールドを集計し、ファイル B の F4 フィールドおよび F5 フィールドを表示後、F1 を共通上位ソートフィールドとして使用します。

```
MATCH FILE A
SUM F2 AND F3 BY F1
RUN
FILE B PRINT F4 AND F5 BY F1
AFTER MATCH HOLD OLD-OR-NEW
END
```

HOLD ファイルに含まれるデータは次のとおりです。

```
F1   F2   F3   F4   F5

1    x    100  a    10
1    x    100  b    20
2    y    200  c    30
2    y    200  d    40
```

ファイル A のレコードが、ファイル B の各レコードに対して複製されます。

リクエスト 3　このリクエストは、ファイル A の F2 フィールドおよび F3 フィールドを表示し、ファイル B の F4 フィールドおよび F5 フィールドを集計後、F1 を共通上位ソートフィールドとして使用します。

```
MATCH FILE A
PRINT F2 AND F3 BY F1
RUN
FILE B
SUM F4 AND F5 BY F1
AFTER MATCH HOLD OLD-OR-NEW
END
```

HOLD ファイルに含まれるデータは次のとおりです。

```
F1   F2   F3   F4   F5

1    x    100  b    30
2    y    200  d    70
```

ファイル A の各レコードは含まれますが、B ファイルからは、各共通上位ソートフィールドに対する最終レコードのみがに含まれます。

リクエスト 4　このリクエストは、ファイル A の F2 フィールドおよび F3 フィールドを表示し、ファイル B の F4 フィールドおよび F5 フィールドを表示後、F1 を共通上位ソートフィールドとして使用します。

```
MATCH FILE A
PRINT F2 AND F3 BY F1
RUN
FILE B PRINT F4 AND F5 BY F1
AFTER MATCH HOLD OLD-OR-NEW
END
```

HOLD ファイルに含まれるデータは次のとおりです。

```
F1   F2   F3   F4   F5

1    x    100  a    10
1         0    b    20
2    y    200  c    30
2         0    d    40
```

F2 のブランク値および F3 の 0 (ゼロ) に注目します。

リクエスト 5　このリクエストは、ファイル A の F2 フィールドおよび F3 フィールドを集計後、ファイル B の F5 フィールドを集計して共通上位ソートフィールドの F1 フィールドと F4 でソートします。

```
MATCH FILE A
SUM F2 AND F3 BY F1
RUN
FILE B
SUM F5 BY F1 BY F4
AFTER MATCH HOLD OLD-OR-NEW
END
```

HOLD ファイルに含まれるデータは次のとおりです。

```
F1   F2   F3   F4   F5

1    x    100  a    10
1    x    100  b    20
2    y    200  c    30
2    y    200  d    40
```

ファイル A のレコードは、ファイルB のレコードと一致するものがすべて含められます。

ユニバーサル連結

ユニバーサル連結により、単一リクエストで異なるデータソースの検索を実行することができます。ソースに関わらず、すべてのデータが単一ファイルのものであるかのように表示されます。MORE 句により、同一フォーマットの関連フィールドを共有する限り、データソースのすべてのタイプ (FOCUS、DB2 など) を連結することができます。WHERE および IF 選別テストを MORE と共用することができます。詳細は、レコードの選択を参照してください。

MORE を使用するには、リクエストを次の項目に分割する必要があります。

- 1 つ目のデータソースを検索し、すべてのデータのデータフィールド、ソート条件および出力フォーマットを定義する 1 つの主要リクエスト。
- 主要リクエストのデータに連結するデータソースおよびフィールドを定義するサブリクエスト。主要リクエストにより表示、ソートするフィールドは、連結する各データソースに存在する必要があります。それ以外の場合は、一時項目 (DEFINE) として作成する必要があります。

検索時に、データは各データソースから順に収集され、主要リクエストで指定したようにすべてのデータがソートされ、出力のフォーマット設定が実行されます。

データソースの連結

TABLE および MATCH コマンドでアクセスできる MORE 句により、ソースからのデータと異なるマスターファイルを連結する方法を指定します。

```
{TABLE|MATCH}  FILE file1main request
MORE
FILE file2
  subrequest
MORE
FILE file3
  subrequest
MORE
   .
   .
   .
{END|RUN}
```

説明

TABLE|MATCH

データソースを連結するリクエストを開始します。

file1

1 つ目のデータソース名です。

main request

END または RUN なしで 1 つ目のデータソースを検索し、すべてのデータのデータフィールド、ソート条件、および出力フォーマットを定義するリクエストです。主要リクエストの WHERE および IF 条件は、*file1* のみに適用されます。

TABLE コマンドでファイルを連結するときは、1 つ目のデータソースの一時項目 (COMPUTE) も定義することができます。

MORE

サブリクエストを開始します。サブリクエスト数には、使用可能なメモリ以外に制限はありません。

FILE file2

file2 を連結用の 2 つ目のデータソースとして定義します。

subrequest

サブリクエストです。サブリクエストには WHERE および IF 句のみを含めることができます。

END|RUN

リクエストを終了します。

データソースの連結

EMPLOYEE および EXPERSON データソースの両方に従業員情報が含まれています。これらの共通データを連結して単一ファイルにすることができます。

- EMPLOYEE にはフィールド値の EMP_ID=123456789、および CURR_SAL=50.00 が含まれます。
- EXPERSON にはフィールド値の SSN=987654321、および WAGE=100.00 が含まれます。

次の注釈付きリクエストは 2 つのデータソースを連結します。

```
   DEFINE FILE EXPERSON
1.EMP_ID/A9 = SSN;
   CURR_SAL/D12.2 = WAGE;
   END
2.TABLE FILE EMPLOYEE
   PRINT CURR_SAL
   BY EMP_ID
3.MORE
   FILE EXPERSON
   END
```

1. このリクエストでは主要リクエストで使用するものと一致させるために、EXPERSON データソースのフィールド名およびフォーマットを再びマッピングする必要があります。
2. 主要リクエストは、連結の 1 つ目のデータソースに EMPLOYEE という名前を付けます。また、両方のデータソースに、表示フィールドおよびソートフィールドを定義します。
3. MORE 句は、次のデータソース EXPERSON を連結するサブリクエストを開始します。サブリクエストでは表示コマンドは使用できません。IF および WHERE 条件がサブリクエストで使用できる唯一のレポートコンポーネントです。

フィールド名およびフォーマットの一致

主要リクエストで参照されているすべてのフィールドは、連結されたすべてのファイルに同一の名前およびフォーマットで存在しているか、一時項目 (DEFINE) を使用してこれらの名前およびフォーマットに再マッピングする必要があります。参照フィールドには、COMPUTE コマンド、見出し、集計句、ソート句、PRINT、LIST、SUM、COUNT コマンドなどがあります。

下表は、正しいフォーマットの一致を示しています。

処理フォーマットのタイプ	一致
A	フォーマットのタイプおよび長さが一致している。
I, F, D	フォーマットのタイプが一致している。
P	フォーマットのタイプおよび精度が一致している。
DATE (新)	フォーマット情報 (タイプ、長さ、コンポーネント、順序) が常に一致している。
DATE (旧)	編集オプションが一致している。
DATE -TIME	フォーマット情報 (タイプ、長さ、構成要素、順序) が常に一致している。

テキスト (TX) フィールドと CLOB フィールド (サポートされている場合) を連結することはできません。

フィールド名およびフォーマットの一致

次の注釈付きの例は EMPDATA および SALHIST データソースからのデータを連結します。

```
   DEFINE FILE EMPDATA
1.NEWID/A11=EDIT (ID,'999-99-9999');
   END

   DEFINE FILE SALHIST
2.NEWID/A11=EDIT (ID,'999-99-9999');
   CSAL/D12.2M=OLDSALARY;
   END

3.TABLE FILE EMPDATA
   HEADING
   "EMPLOYEE SALARIES"
   " "
   PRINT CSAL
   BY NEWID
4.WHERE CSAL GT 65000
5.MORE
   FILE SALHIST
6.WHERE OLDSALARY GT 65000
   END
```

1. 主要リクエストで参照したソートフィールドと同一名、同一フォーマットの NEWID を、EMPDATA データソースに定義します。
2. 主要リクエストで参照したソートフィールドと同一名、同一フォーマットの NEWID を、SALHIST データソースに定義します。
3. 主要リクエストです。このリクエストには、結果レポートのすべてのフォーマットを含めるとともに、連結する 1 つ目のファイル名を指定します。このリクエストには、表示およびソートに関するすべての情報も含まれます。表示フィールドおよびソートフィールドは、各ファイルに実フィールドまたは一時項目 (DEFINE) として存在している必要があります。
4. 主要リクエストの WHERE 条件は、EMPDATA データソースにのみ適用されます。
5. MORE 句は、SALHIST データソースを EMPDATA データソースに連結します。

6. この WHERE 条件は、SALHIST データソースにのみ適用されます。ここでは、EMPDATA データソースで定義されていないフィールドが参照されています。

出力結果は次のとおりです。

EMPLOYEE SALARIES	
NEWID	SALARY
000-00-0030	$70,000.00
	$70,000.00
000-00-0070	$83,000.00
	$83,000.00
	$79,100.00
000-00-0200	$115,000.00
	$115,000.00
	$102,500.00

	$89,500.00
000-00-0230	$80,500.00
	$80,500.00
	$75,000.00
	$70,800.00
000-00-0300	$79,000.00
	$79,000.00
	$75,000.00
	$70,000.00

データを連結するときは、レコードセットが単純に追加されるだけで、ファイル間でのグループ化や集計は実行されません。そのため、重複したソートフィールドが存在する場合は、レコードセットがレポート出力に 2 回表示されます。

連結したデータソースの統合

MATCH リクエストで MORE 句を使用して、最大で 16 組の連結データソースを結合することができます。

主要リクエストのすべての MATCH 必要事項を満たす必要があります。結合するすべてのデータソースは、少なくとも共通フォーマットのフィールドの 1 つでソートする必要があります。

MATCH リクエストの結果として、結合されたデータが HOLD ファイルに格納されます。AFTER MATCH コマンドを使用して、これらのファイルの結合方法を指定することができます。たとえば、レコードを次のように保持することができます。

- 両方のファイルからのすべてのレコード (OLD-OR-NEW)。これがデフォルト値です。
- 両方のファイルに共通するレコードのみ (OLD-AND-NEW)。
- 1 つ目のファイルからのレコードで、2 つ目のファイルに一致のないもの (OLD-NOT-NEW)。
- 2 つ目のファイルからのレコードで、1 つ目のファイルに一致のないもの (NEW-NOT-OLD)。
- 両方のファイルからの一致のないすべてのレコード。つまり、一方のファイルのみにあり両方にないレコード (OLD-NOR-NEW)。
- 1 つ目のファイルからのすべてのレコードで、2 つ目のファイルに一致するすべてのレコードを含むもの (OLD)。
- 2 つ目のファイルからのすべてのレコードで、1 つ目のファイルからのすべての一致レコードを含むもの (NEW)。

連結したデータソースの結合

```
1.MATCH FILE file1main request
   MORE
2.FILE file2subrequest
   MORE
3.FILE file3subrequest
   RUN
4.FILE file4main request
5. [AFTER MATCH merge_phrase]
   MORE
6.FILE file5subrequest
   MORE
7.FILE file6subrequest
   RUN
8.FILE file7main request
9. [AFTER MATCH merge_phrase]
```

```
   MORE
10.FILE file8subrequest
   MORE
11.FILE file9subrequest
   END
```

1. MATCH の 1 つ目のアンサーセットを開始します。file1 が 1 つ目のアンサーセット内の 1 つ目のデータソースです。
2. 1 つ目の MATCH アンサーセットで、file2 を file1 に連結します。
3. 1 つ目の MATCH アンサーセットで file3 を file1 および file2 に連結します。
4. MATCH の 2 つ目のアンサーセットを開始します。File4 が 2 つ目のアンサーセットの 1 つ目のデータソースです。
5. 1 つ目のアンサーセットで連結されたすべてのデータは、2 つ目のアンサセットの AFTER MATCH merge_phrase を使用して、2 つ目のアンサセットで連結されたデータと結合されます。
6. 2 つ目の MATCH アンサーセットで、file5 を file4 に連結します。
7. 2 つ目の MATCH アンサーセットで file6 を file4 および file5 に連結します。
8. MATCH の 3 つ目のアンサーセットを開始します。File7 が 3 つ目のアンサーセットの 1 つ目のデータソースです。
9. HOLD ファイルになった 1 つ目および 2 つ目のアンサーセットで連結されたすべてのデータは、3 つ目のアンサーセットの AFTER MATCH merge_phrase を使用して、3 つ目のアンサーセットで連結されたデータと結合されます。結合データの最終セットは HOLD ファイルに保存されます。
10. 3 つ目の MATCH アンサーセットで、file8 を file7 に連結します。
11. 3 つ目の MATCH アンサーセットで、file9 を file7 および file8 に連結します。

MATCH リクエストでのソートフィールドの使用

MATCH のデータソースにより、同一名、同一フォーマットの共通上位のソートフィールドが共有される場合、MATCH プロセスにより各ファイルからの一致するソートフィールド値を含むレコードが統合されます。MATCH の 2 つのデータソースに、名前が異なる同一の

ソートフィールドが存在する場合、AS 句を使用して、いずれかの名前を変更することができます。

MATCH で指定したファイルが、上位のソートフィールドを共有していない場合は、フィールドは比較されません。代わりに、各データソースの 1 つ目のレコードからのフィールドが結合され、HOLD ファイルの 1 つ目のレコードを作成されます。残りすべてのレコードも同様に作成されます。

共通上位ソートフィールドによる連結したデータソースの結合

次の注釈付きのストアドプロシジャの例は、共通ソートフィールドを使用した、MORE を含む MATCH を示しています。

```
1.DEFINE FILE EMPDATA
   CURR_SAL/D12.2M = CSAL;
   FIRST_NAME/A10 = FN;
   EID/A9 = PIN;
   END

   -*Start MATCH.

2.MATCH FILE EMPLOYEE
      SUM CURR_SAL AS 'CURRENT'
          FIRST_NAME AS 'FIRST'
      BY EID AS 'SSN'
   -*Concatenate file EMPDATA to EMPLOYEE to form first MATCH answer set.
3.    MORE
      FILE EMPDATA
      RUN
   -*Second MATCH answer set:

4.FILE TRAINING
      PRINT EXPENSES
5.    BY PIN AS 'SSN'
6.    AFTER MATCH HOLD OLD-OR-NEW
   END

   -*Print merged file:

7.TABLE FILE HOLD
      PRINT *
   END
```

1. EMPLOYEE に連結するために必要な EMPDATA フィールドを定義します。
2. 連結で MATCH および主要リクエストを開始します。主要リクエストは、連結したファイルのすべての表示およびソートを定義します。ソートフィールドは結果ファイルで「SSN」と呼ばれます。
3. EMPDATA ファイルを EMPLOYEE に連結します。この連結ファイルが MATCH の OLD ファイルになります。
4. MATCH で NEW ファイルを作成します。
5. AS 句を使用して、NEW ファイルのソートフィールド名を OLD ファイルのソートフィールドの名前と同一のものに変更します。
6. 結合プロシジャを定義します。NEW ファイル、OLD ファイル、および両方のファイルからのすべてのレコードが、最終 HOLD ファイルに格納されます。
7. 結合ファイルの値を表示します。

次は出力結果の 1 ページ目です。

```
SSN               CURRENT  FIRST        EXPENSES
---               -------  -----        --------
000000010       $55,500.00  DANIEL       2,300.00
000000020       $62,500.00  MICHAEL             .
000000030       $70,000.00  LOIS         2,600.00
000000030       $70,000.00  LOIS         2,300.00
000000040       $62,500.00  RUTH         3,400.00
000000050       $54,100.00  PETER        3,300.00
000000060       $55,500.00  DORINA              .
000000070       $83,000.00  EVELYN              .
000000080       $43,400.00  PAMELA       3,200.00
000000080       $43,400.00  PAMELA       3,350.00
000000090       $33,000.00  MARIANNE            .
000000100       $32,400.00  TIM          3,100.00
000000110       $19,300.00  ANTHONY      1,800.00
000000110       $19,300.00  ANTHONY      2,500.00
000000110       $19,300.00  ANTHONY      2,400.00
000000120       $49,500.00  KATE         2,200.00
000000130       $62,500.00  MARCUS              .
```

共通ソートフィールドのない連結データソースの結合

次の例では、結合されたデータソースには共有ソートフィールドはありません。

```
DEFINE FILE EMPDATA
CURR_SAL/D12.2M = CSAL;
FIRST_NAME/A10 = FN;
EID/A9 = PIN;
END

-*Start MATCH

MATCH FILE EMPLOYEE
SUM CURR_SAL AS 'CURRENT'
    FIRST_NAME AS 'FIRST'
BY EID AS 'SSN'

-*Concatenate EMPDATA to EMPLOYEE to form the first MATCH answer set

MORE
FILE EMPDATA
RUN

-*Second MATCH answer set:

FILE TRAINING
PRINT EXPENSES
BY PIN AS 'EID'
AFTER MATCH HOLD OLD-OR-NEW
END

-*Print merged file:

TABLE FILE HOLD
PRINT *
END
```

AS 句によりアンサーセットが変更されます。同一名のソートフィールドは存在しないため、フィールドは一致レコードとは関係なく結合されます。

次は出力結果の 1 ページ目です。

```
SSN               CURRENT  FIRST       EID         EXPENSES
---               -------  -----       ---         --------
000000010       $55,500.00  DANIEL      000000010   2,300.00
000000020       $62,500.00  MICHAEL     000000030   2,600.00
000000030       $70,000.00  LOIS        000000030   2,300.00
000000040       $62,500.00  RUTH        000000040   3,400.00
000000050       $54,100.00  PETER       000000050   3,300.00
000000060       $55,500.00  DORINA      000000080   3,200.00
000000070       $83,000.00  EVELYN      000000080   3,350.00
000000080       $43,400.00  PAMELA      000000100   3,100.00
000000090       $33,000.00  MARIANNE    000000110   1,800.00
000000100       $32,400.00  TIM         000000110   2,500.00
000000110       $19,300.00  ANTHONY     000000110   2,400.00
000000120       $49,500.00  KATE        000000120   2,200.00
000000130       $62,500.00  MARCUS      000000140   3,600.00
000000140       $62,500.00  VERONICA    000000150   3,400.00
000000150       $40,900.00  KARL        000000160   1,000.00
000000160       $62,500.00  ROSE        000000180   1,250.00
000000170       $30,800.00  WILLIAM     000000190   3,150.00
```

直積

直積により、非関連レコード、またはマルチパスリクエストのデータインスタンスのすべての組み合わせを含むレポートの作成が可能になります。つまり、親セグメントの 1 つのパスに 3 つの子インスタンス、もう 1 つのパスに 2 つの子インスタンスがある場合、CARTESIAN が ON のときは、その親セグメントおよび両方の子を参照するリクエストにより 16 のレコードが作成されます。CARTESIAN が OFF の場合は、同じリクエストにより 3 つのレコードのみが作成されます。

独立パスの子セグメントに適用する選別テストの制御についての詳細は、レコードの選択を参照してください。

直積の有効化と無効化

```
SET CARTESIAN = {OFF|ON}
```

説明

OFF

直積を無効にします。OFF がデフォルト値です。

ON

直積を有効にし、非関連レコードのすべての可能な組み合わせを作成します。

リクエストで SET CARTESIAN も発行することができます。

直積使用上の注意

- 直積は、セグメントのフィールドが参照されるかどうかに関わらず、すべてのパスに共通する最下位のセグメントで実行されます。
- 直積を含むリクエストでは、ショートパスは表示されません。
- SET CARTESIAN パラメータは ACROSS を指定すると無効化され、警告メッセージが発行されます。
- SUM 表示コマンドおよび TOT. 演算接頭語は直積に影響しません。
- SUM、COMPUTE、および WITHIN と PRINT 表示コマンドの組み合わせは 直積を実行します。
- ON TABLE COLUMN-TOTAL は直積を自動的に生成します。
- NOSPLIT は、SET CARTESIAN パラメータとの組み合わせで指定すると無効化され、警告メッセージは発行されません。
- MATCH は SET CARTESIAN パラメータではサポートされません。MATCH を要求すると警告メッセージは発行されず、そのリクエストは CARTESIAN を OFF に設定した場合と同様に処理されます。
- TABLEF は SET CARTESIAN パラメータではサポートされません。

直積によるレポートの実行

CARTESIAN を ON に設定すると、次のマルチパスリクエストにより、各車の MODEL および STANDARD のすべての可能な組み合わせを含むレポートが作成されます。

```
SET CARTESIAN=ON
TABLE FILE CAR
PRINT MODEL STANDARD
BY CAR
IF CAR EQ 'JAGUAR'
END
```

ASCII 環境での出力は次のとおりです。

CAR	MODEL	STANDARD
JAGUAR	V12XKE AUTO	4 WHEEL DISC BRAKES
	V12XKE AUTO	POWER STEERING
	V12XKE AUTO	RECLINING BUCKET SEATS
	V12XKE AUTO	WHITEWALL RADIAL PLY TIRES
	V12XKE AUTO	WRAP AROUND BUMPERS
	XJ12L AUTO	4 WHEEL DISC BRAKES
	XJ12L AUTO	POWER STEERING

	XJ12L AUTO	RECLINING BUCKET SEATS
	XJ12L AUTO	WHITEWALL RADIAL PLY TIRES
	XJ12L AUTO	WRAP AROUND BUMPERS

CARTESIAN を OFF に設定 (デフォルト) すると、同じリクエストの結果は論理関係のない MODEL および STANDARD のリストを含む、CAR データソースからのレポートになります。

ASCII 環境での出力は次のとおりです。

CAR	MODEL	STANDARD
JAGUAR	V12XKE AUTO	4 WHEEL DISC BRAKES
	XJ12L AUTO	POWER STEERING
	.	RECLINING BUCKET SEATS
	.	WHITEWALL RADIAL PLY TIRES
	.	WRAP AROUND BUMPERS

レポートフォーマットの概要

効果的なレポートを作成するには、以下が必要です。

- **コンテンツ**　レポート内のデータです。
- **フォーマット**　データを視覚的に印象付けるための表現方法です。

フォーマットオプションは、レポートに適切な外観、その視覚的効果および印象を与えるために使用します。たとえば、以下を制御することができます。

- データの外観。重要な値を強調するために変更することができます。
- データの枠組みに使用する見出し、脚注、その他のテキスト。
- ページまたは画面上のレポートのレイアウト。異なる表示環境や異なるニーズに対応するための調整が可能です。

この章では、表形式レポート (Financial Modeling Language レポートを含む) および自由形式レポートの概要について説明します。グラフのフォーマットについての詳細は、グラフの作成を参照してください。

利用可能なフォーマット

レポートには、さまざまなフォーマットを適用することができます。

- **レポートのデータの外観**　フォント、サイズ、スタイル (斜体、太字、下線)、色 (前景および背景)、位置、位置揃えなど、レポートの外観を調整することができます。また、データの周囲に枠線や線を描画することもできます。これらのプロパティを使用することにより、重要な値を強調し、重要なデータの関係に注目を集めることができます。

  また、国別の慣習に対応するために、小数点の位置を表す文字にピリオド (.) またはカンマ (,) を使用することができます。ミッシング値 (ミッシングデータ) を表す文字を選択することもできます。詳細は、レポートデータのフォーマットを参照してください。

- **枠線を適用することによるデータの強調**　見出し、脚注、およびカスタマイズされたフィールドタイトルで枠線を使用することにより、データを強調することができま

す。見出しと脚注にフィールドとイメージを挿入することもできます。データの場合と同様に、見出し、脚注、フィールドタイトルのフォーマット (フォント、サイズ、スタイル、色、位置、位置揃え) を指定したり、枠線や線を適用したりすることができます。枠線を使用することにより、データを強調することができます。

- **レポートのレイアウト**　画面または出力ページでレポートのレイアウトを調整します。レポートマージン、見出しおよび脚注の位置、背景イメージ (透かし)、レポートフィールドの配列 (列間のブランクの調整、列幅と順序の調整、列の積み重ねによるレポート幅の調整) を選択することができます。色や線を使用することにより、異なる列、行、ソートグループを視覚的に区別することができます。必要に応じて、レポートの一部または全体に境界線を適用することもできます。

  サイズおよび解像度の異なる画面などの異なる表示環境や、異なるサイズの出力ページ用に最適化するためにレポートのレイアウトを指定することができます。ラベルを出力するため、1 ページに複数のレポート枠を作成することができます。1 つのファイルに複数のレポートを組み合わせ、これらをグループとして表示または出力することができます。詳細は、レポートページのレイアウトを参照してください。

- **条件付きフォーマット**　レポートのデータに基づいて、レポートのフォーマットを条件付きで設定することができます。指定したレポートコンポーネントのインスタンス (例、ソートフィールドの各値) ごとに、実行時に自動的に評価される条件を指定します。フォーマットオプションは、条件が真値であるレポートコンポーネントの各インスタンスに適用されます。たとえば、セールスレポートで、ノルマを達成した営業社員の名前を太字に変更し、異なる色で表示することにより、注目を集めることができます。詳細は、レポートフォーマット設定の管理を参照してください。

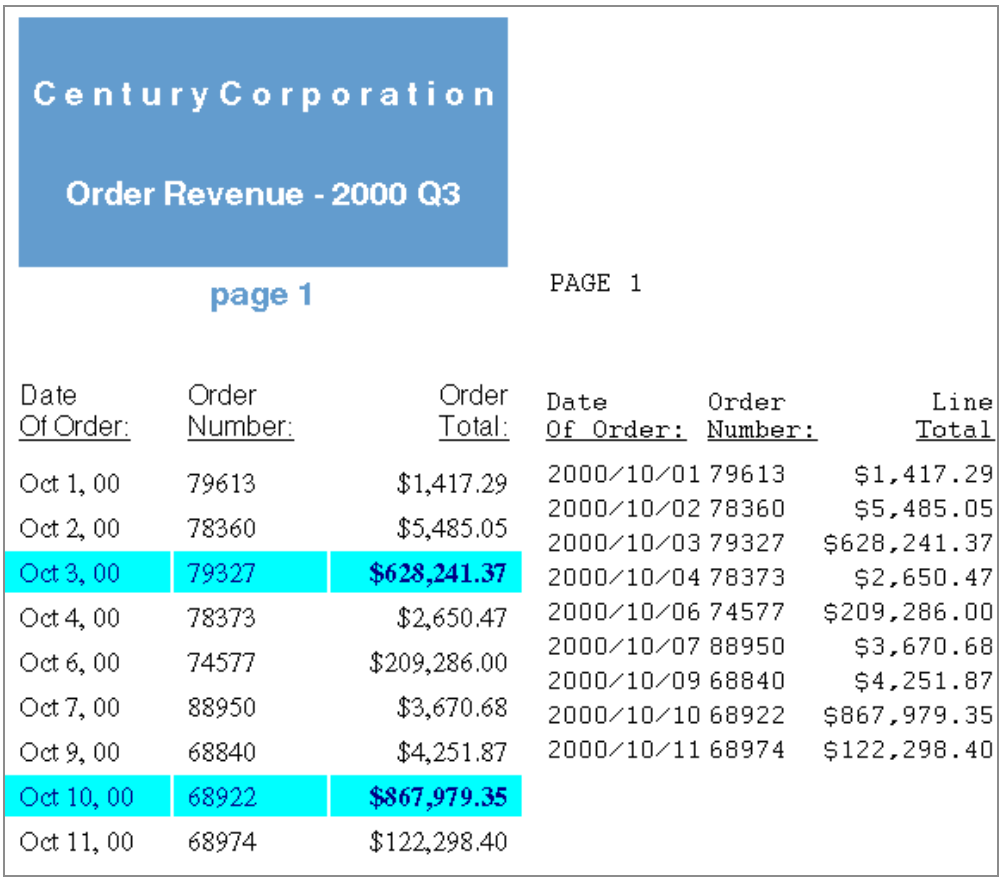
- **表示フォーマットの選択**　HTML (デフォルト値)、PDF (Portable Document Format)、Excel 2000 など、レポートの表示および処理に適切な表示フォーマットを選択します。

- **レポートへのアクセス**　ブラウザタイプや画面の設定には関係なくすべてのユーザのレポートへのアクセスを可能にします。たとえば、レポートのフォント、色、およびレイアウト、また特別な表示が必要なユーザに対応するためのその他のフォーマットを設計したり、音声または点字ベースのブラウザを使用するユーザが情報を把握することができるよう、テーブルとグラフのテキスト説明を追加したりすることができます。レポートは、米国リハビリテーション法 (U.S. Rehabilitation Act) 第 508 条 (Section 508) などのアクセシビリティのガイドラインに準拠する必要があります。

レポートフォーマットの利点

次の 2 つのレポートは、Century Corporation の 2002 年第 3 四半期の Date Of Order、Order Number、Order Total を示します。左側はフォーマット設定後のレポート、右側は

フォーマット設定前のレポートです。



Century Corporation

Order Revenue - 2000 Q3

page 1

Date Of Order:	Order Number:	Order Total:
Oct 1, 00	79613	$1,417.29
Oct 2, 00	78360	$5,485.05
Oct 3, 00	79327	$628,241.37
Oct 4, 00	78373	$2,650.47
Oct 6, 00	74577	$209,286.00
Oct 7, 00	88950	$3,670.68
Oct 9, 00	68840	$4,251.87
Oct 10, 00	68922	$867,979.35
Oct 11, 00	68974	$122,298.40

PAGE 1

Date Of Order:	Order Number:	Line Total
2000/10/01	79613	$1,417.29
2000/10/02	78360	$5,485.05
2000/10/03	79327	$628,241.37
2000/10/04	78373	$2,650.47
2000/10/06	74577	$209,286.00
2000/10/07	88950	$3,670.68
2000/10/09	68840	$4,251.87
2000/10/10	68922	$867,979.35
2000/10/11	68974	$122,298.40

左側のレポートに適用されたフォーマットについて考察します。

- **強調**　見出しの追加および色の使用。
- **レポートの重要性の明示**　見出しの使用、およびデフォルトのフィールドタイトル「Line Total (行の合計)」の「Order Total (注文の合計)」への変更。
- **読みやすさの向上**　行間の拡大、「Date of Order (発注日付)」のフォーマット変更、およびプロポーショナルフォントの使用。
- **重要なデータの強調**　$500,000 を超過した行の背景色およびフォント色の変更。また、「Order Total (注文の合計)」のフォントスタイルの太字への変更。

レポートフォーマットの指定

レポートのフォーマットは、スタイルシートを使用して指定することができます。スタイルシートは、レポートの外観を定義する構文のセットです。フォーマットタイプによっては、

SET パラメータや TABLE コマンドなどによるスタイルシートへの補足が必要な場合があります。ここでは、指定されたフォーマットタイプを得るための必要事項について説明します。

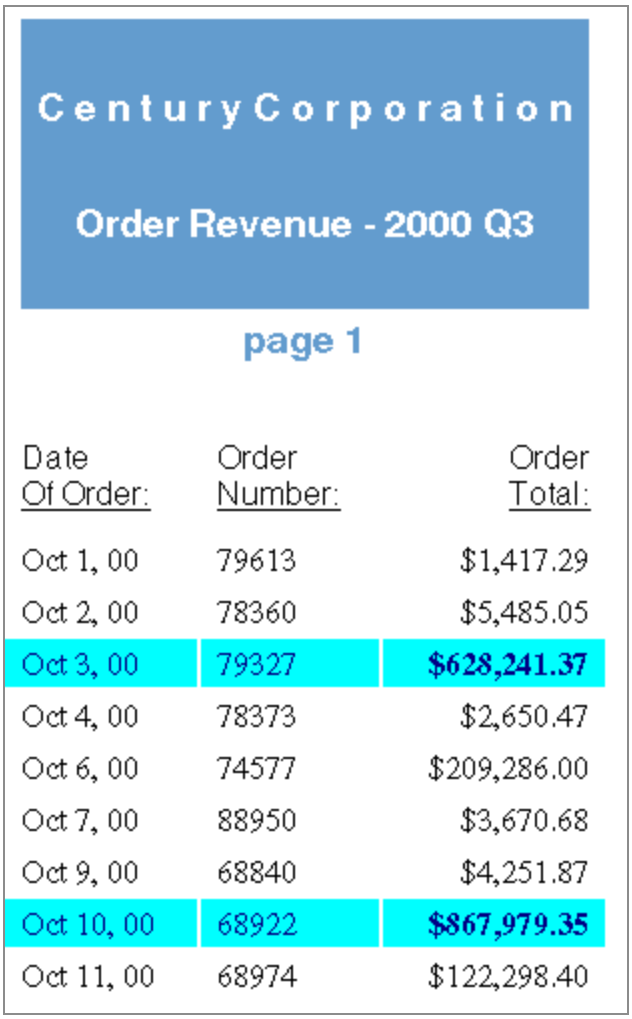
スタイルシートの利点　フォーマットタイプによっては、スタイルシートと別機能のいずれかを選択することができます。スタイルシートではフォーマットロジックの集約および再利用が可能なため、スタイルシートの使用が通常優先されます。スタイルシートを使用することにより、次の利点が得られます。

- **生産性**　数行のコード (1 つのスタイルシート) の使用により多数のレポートにフォーマットを設定できるため、各レポートに要する開発時間を短縮することができます。
- **簡単なメンテナンス**　スタイルシートを 1 つ編集するだけで、数多くのレポートのフォーマットを一度に変更することができます。
- **一貫性のある外観**　同一スタイルシートをレポートに適用することにより、レポートの外観に一貫性を持たせることができます。
- **すばやいフォーマット変更**　レポートに適用されたスタイルシートを変更するだけで、レポートの外観をすばやく変更することができます。
- **優先度**　既存のスタイルシートを適用することによりレポートの表現をすばやく得ることができるため、レポートの内容に集中することができます。

レポートのフォーマットには、さまざまなスタイルシートが使用できます。これらのスタイルシートについての詳細は、スタイルシートタイプの選択を参照してください。

注文の収入レポートへのフォーマットの指定

次のレポートには、Century Corporation の 2002 年第 3 四半期においての Date Of Order、Order Number、Order Total が表示されています。



Century Corporation

Order Revenue - 2000 Q3

page 1

Date Of Order:	Order Number:	Order Total:
Oct 1, 00	79613	$1,417.29
Oct 2, 00	78360	$5,485.05
Oct 3, 00	79327	$628,241.37
Oct 4, 00	78373	$2,650.47
Oct 6, 00	74577	$209,286.00
Oct 7, 00	88950	$3,670.68
Oct 9, 00	68840	$4,251.87
Oct 10, 00	68922	$867,979.35
Oct 11, 00	68974	$122,298.40

レポートのフォーマットは、WebFOCUS スタイルシート、およびレポートプロシジャ内のコマンドのフォーマットに従って設定されます。以下は、プロシジャ (Revenue.fex) とスタイルシートファイル (OrderRev.sty) です。

Revenue.fex

```
   TABLE FILE CENTORD
1.HEADING
1. " "
1."C e n t u r y C o r p o r a t i o n"
1. " "
1."Order Revenue - 2000 Q3"
1. " "
1. "page <TABPAGENO"
1. " "
2.SUM ORDER_DATE/MtDY ORDER_NUM LINEPRICE AS 'Order,Total:'
```

```
   BY LOWEST 9 ORDER_DATE NOPRINT
   WHERE (ORDER_DATE GE '2000/10/01') AND (ORDER_DATE LE '2000/12/31');
   ON TABLE SET ONLINE-FMT PDF
ON TABLE SET SQUEEZE ON
ON TABLE SET STYLESHEET OrderRev
   END
```

OrderRev.sty

```
TYPE=Report, GRID=Off, UNITS=Inches, TOPGAP=0.06, BOTTOMGAP=0.06, $
TYPE=Data, FONT='Times', $
TYPE=Data, BACKCOLOR=Aqua, COLOR=Navy,
      WHEN=LinePrice GT 500000, $
TYPE=Data, COLUMN=LINEPRICE, BACKCOLOR=Aqua, COLOR=Navy, STYLE=Bold,
      WHEN=LinePrice GT 500000, $
TYPE=Title, FONT='Helvetica', $
TYPE=Heading, FONT='Helvetica', STYLE=Bold, SIZE=14, JUSTIFY=Center,
                        BACKCOLOR=Dark Turquoise, COLOR=White, $
TYPE=Heading, LINE=6, BACKCOLOR=White, COLOR=Dark Turquoise, $
TYPE=Heading, LINE=7, BACKCOLOR=White, $
```

1. レポートにページ見出しを追加します。
2. 発注日の日付フォーマットを変更します。この例では、2000/10/07 が Oct. 7, 00 に変更されます。
3. ページマージンではなく、レポートマージンで見出しの位置を揃えます。
4. スタイルシートファイルを識別して、レポートのフォーマットを設定します。
5. レポートの行間を拡大します。
6. レポートのデータにプロポーショナル Serif フォントを使用します。
7. $500,000 を超過した注文の合計を強調するためにそのフォーマットを変更します。ここでは、フォントに紺色を、背景にアクアマリン色を適用し、注文の合計値のフォントスタイルを太字に変更します。
8. レポートのフィールドタイトルにプロポーショナル Sans Serif フォントを使用します。
9. レポートの見出しを中央揃えにし、Sans Serif フォントのサイズを拡大します。背景色を青緑に、文字を白に変更し、ページ番号 (見出しの 6 行目) にはその逆の色の組み合わせを適用します。

これは、フォーマット設定手順の概要です。詳細は、各フォーマットについての説明を参照してください。

スタイルシートタイプの選択

レポートのフォーマットを設定する場合、次の 2 種類のスタイルシートからいずれかを選択することができます。

- **WebFOCUS スタイルシート**　スタイルシートとも呼ばれます。WebFOCUS スタイルシート言語で記述されたスタイルシートです。WebFOCUS スタイルシートでは、HTML、PDF、Excel 2000 などの表示フォーマットを使用できるため、レポートのフォーマットを柔軟に設定することができます。スタイルシートは個別のファイルとして保存することも、リクエスト内に保存することもできます。個別のファイルとして保存する場合は、複数のレポートへの適用が可能です。

  HTML フォーマットでレポートを生成している場合、WebFOCUS スタイルシートで内部カスケードスタイルシート (CSS) を動的に生成することにより、そのパフォーマンスを向上させ、利用可能なフォーマットオプション数を増加することができます。CSS は、HTML ドキュメント用に設計された標準スタイルシート言語です。WebFOCUS により生成された内部カスケードスタイルシート (CSS) は、レポート出力の内部に存在します。個別のファイルとしては保存されません。内部カスケードスタイルシートについての詳細は、HTML レポート内部カスケードスタイルシートの生成を参照してください。

- **外部カスケードスタイルシート**　HTML ドキュメント用に設計された標準スタイルシート言語です。外部カスケードスタイルシートは、HTML フォーマットの WebFOCUS レポートに適用することができます。外部カスケードスタイルシートは、フォーマットを設定するドキュメント内には保存されず、個別ファイルとして保存されたものです。

これらのスタイルシートのどちらのタイプを選択するかについては、次の事項について考察します。

- WebFOCUS スタイルシートを選択する場合

  表示フォーマットでレポートを表示　PDF や Excel 2000 などの表示フォーマットでレポートを表示する場合です。WebFOCUS スタイルシートは、さまざまな表示フォーマットでサポートされますが、カスケードスタイルシートは、HTML フォーマットのレポートでのみ使用可能です。

- 外部カスケードスタイルシートを選択する場合

  カスケードスタイルシートを採用　HTML ドキュメントのフォーマットにカスケードスタイルシートを使用しており、レポートに同じガイドラインを適用する場合です。

  別のタイプの HTML ドキュメントに同一フォーマットを適用　別のタイプの HTML ドキュメントに同一フォーマットを適用する場合です。

標準およびレガシーフォーマット

多くの場合、新しいバージョンの WebFOCUS では、強化されたレポートフォーマット設定方法が提供されています。新機能には、既存の機能がオプション数や機能性の面で向上したものがあります。新機能が標準となり、以前の機能は「レガシー」機能と見なされます。

新しいアプリケーションを作成し、標準機能とレガシー機能のいずれかを選択できる場合、標準機能の選択をお勧めします。レガシー機能が組み込まれた以前のアプリケーションの保守が必要な場合、時間を節約するためにレガシー機能を保持することも、新しい機能を使用できるよう、標準機能へアプリケーションを変換することもできます。

標準フォーマットとレガシーフォーマットの相違の例として、ページ上にレポートをレイアウトする方法について考察します。

- **標準**　TOPMARIGN、BOTTOMMARGIN、LEFTMARGIN、RIGHTMARGIN を使用して、ページマージンを指定します。これらのキーワードは、スタイルシート属性または SET コマンドパラメータとして適用することができます。UNITS (スタイルシート属性または SET コマンドパラメータとしても発行可能) により、マージンの単位 (インチ、センチメートル、ポイント) を指定することができます。この作業は簡単で、その他のドキュメントタイプの設計と同様にレポートを設計することができます。
- **レガシー**　ページ上のレポート出力の縦の長さ (SET コマンドの LINES パラメータを使用したレポート行数で測定)、および横幅 (SET コマンドの WIDTH パラメータを使用したバイト数で測定) を指定することができます。上下のマージンは、それぞれページとレポートの縦の長さ (行数で測定) の差を 2 分割し、左右のマージンは、それぞれページとレポートの横幅 (バイト数で測定) の差を 2 分割します。このレガシー方法でのフォントの使用は、Courier などの等幅フォントのみに限定されます。

すばやく簡単なフォーマット設定のためのテクニック

時間を節約し労力を低減するため、複数のフォーマット設定方法を適用することができます。これらの方法の多くでは、WebFOCUS に備わっているコードの使用、またはユーザが作成したコードの利用が可能です。

- **継承および上書き**　各レポートコンポーネントは、スタイルシート属性をその「親」レポートコンポーネントから継承します。この機能により、親コンポーネントに構文を 1 回追加するだけで、共通フォーマットを定義することができます。子孫コンポーネントにはフォーマットが自動的に継承され、ユーザは必要に応じて継承された値を上書

きすることができます。継承を利用できるようにスタイルシートを設計することにより、記述するコード数を削減し、複数のレポートコンポーネントへのフォーマットをすばやく更新することができます。

たとえば、レポートのすべてのデータに青色を指定すると、フィールドデータはすべて青色で表示されます。BY フィールドにオレンジ色を指定すると、ソートフィールドの青色は上書きされ、オレンジ色で表示されます。EMP_ID ソートフィールドに緑色を指定すると、EMP_ID ソートフィールドのオレンジ色は上書きされ、緑色で表示されます。詳細は、WebFOCUS スタイルシート属性の継承を参照してください。

- **マクロ**　1 つのスタイルシート内の複数の構文に同一の属性と値を指定する場合、「マクロ」を作成することにより、毎回コードを記述することなく、スタイルシート内で属性を繰り返し適用することができます。その後、値を変更する必要がある場合、マクロで変更を一度だけ行います。これにより、スタイルシート内で変更が自動的に適用されます。

  たとえば、重要なフィールドのタイトル、しきい値を超過するデータ値、ソート項目を指定している見出しなど、強調するレポートコンポーネントが複数あり、これらをすべて太字に変更し、紫色で表示するには、まずフォントスタイルを太字に、色を紫色に設定するマクロを定義し、そのマクロをこれらのレポートコンポーネントに適用します。詳細は、マクロを使用した WebFOCUS スタイルシート宣言の再利用を参照してください。

- **デフォルト**　WebFOCUS スタイルシート属性の多くには、デフォルト値が指定されています。各スタイルシート属性を明示的に指定する代わりに、属性値を省略し、そのデフォルト値を受け入れることもできます。たとえば、フォントを指定する代わりにデフォルトのフォントを使用することができます。各属性のデフォルト値は、属性の構文と同じ場所に記述されています。

レポートからその他のリソースへのナビゲーション

レポートのある点から、同一レポートの別の部分、その他のレポート、インターネットリソースへ移動することができます。ナビゲーションはフォーマットとしては考慮されませんが、スタイルシートを使用することにより、ある種のナビゲーションをサポートすることができます。

ユーザに次のナビゲーションを提供することができます。

- **ドリルダウン**　関連するレポートへのドリルダウンが可能です。詳細は、別レポートへのリンクを参照してください。

- **Web ページへのリンク**　Email メッセージの構成と送信、およびその他のインターネットリソースへの接続が可能です。詳細は、URL へのリンクを参照してください。
- **JavaScript 関数の実行**　レポートデータの追加分析の実行が可能です。詳細は、JavaScript 関数へのリンクを参照してください。
- **目次を使用したナビゲーション**　必要なデータへ直接移動することができます。レポートは、ソート値に基づいて動的に目次を生成します。目次で項目を選択することにより、レポートの一部分または全体を表示することができます。
- **レポートページ間のナビゲーション**　現在のレポートページの上部、またはレポートの開始位置や終了位置へ移動することができます。

WebFOCUS スタイルシートの作成と管理

スタイルシートを使用してレポートのフォーマットを設定し、重要な情報を強調して魅力的なレポートを生成することができます。スタイルシートを使用すると、レポートのさまざまな特性を明確に表現したり、レポートコンポーネントのフォーマット設定を個別に行ったりすることができます。

スタイルシートは、次の場合に使用することができます。

- 特定のレポートコンポーネントにフォーマットを個別に設定する。
- グラフ要素を組み込む。
- 動的なハイパーリンクを定義する。
- 指定した条件を満たすデータに対してフォーマットを設定する。
- マクロを作成して、フォーマット指定を効率的に適用する。

また、HTML レポート用に外部カスケードスタイルシートを使用したり、内部カスケードスタイルシートを有効にしたりすることもできます。詳細は、レポートフォーマット設定の管理を参照してください。

特に明記していない限り、この章で言及するスタイルシートは、WebFOCUS スタイルシートのことを指します。

WebFOCUS スタイルシートの作成

スタイルシートは、次の方法で作成することができます。

- **リクエスト内で作成**　インラインスタイルシートとして作成します。この方法は、スタイルシートを 1 つのレポートのみに適用する場合に役立ちます。詳細は、WebFOCUS スタイルシートの作成を参照してください。
- **リクエスト外で作成**　別のファイルとして作成します。この方法では、1 つのスタイルシートを複数のレポートに適用することができます。詳細は、WebFOCUS スタイルシートファイルの作成と適用を参照してください。

注意：スタイルシートファイルを別のスタイルシートに含めることもできます。これにより、スタイルシートファイル内のスタイルを適用する一方で、特定の属性を上書きすることができます。詳細は、スタイルシートファイルへの別スタイルシートの追加を参照してください。

リクエストでの WebFOCUS スタイルシートの作成

スタイルシートは、リクエスト内で作成することができます。この方法では、レポートのフォーマット設定をリクエストで直接作成して管理することができます。このタイプのスタイルシートは、「インラインスタイルシート」と呼ばれます。

リクエストでの WebFOCUS スタイルシートの作成

```
ON TABLE SET STYLE[SHEET] *
declaration
[declaration]
.
.
.
[ENDSTYLE]
```

説明

SHEET

この部分を省略してコマンドの長さを短くすることができます。コマンドの動作には影響しません。

declaration

スタイルシート宣言です。通常、スタイルシート宣言で指定するのは、フォーマットを設定するレポートコンポーネントおよびそれに適用するフォーマットです。構文についての詳細は、一般的な WebFOCUS スタイルシート構文を参照してください。

ENDSTYLE

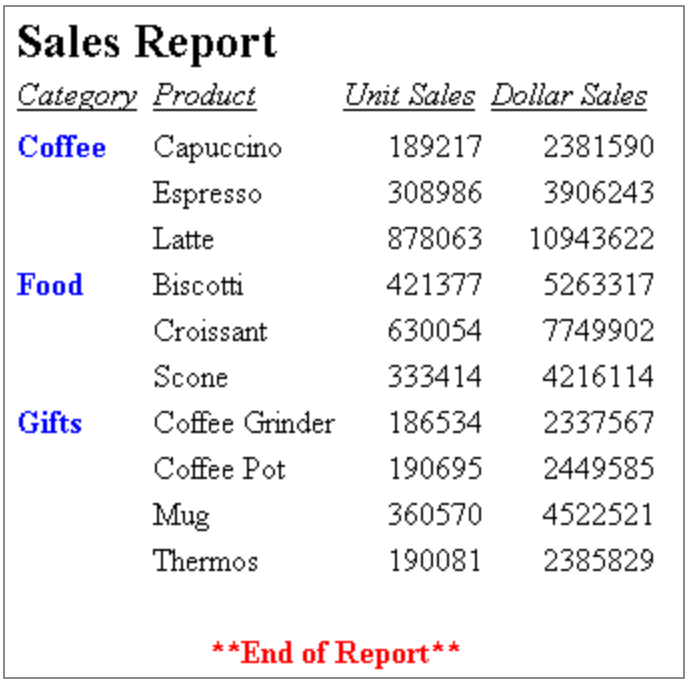
インラインスタイルシートの終了を表します。リクエストで ENDSTYLE の直後に END が続く場合は、その ENDSTYLE を省略することができます。

リクエストでの WebFOCUS スタイルシートの作成

次の例は、インラインスタイルシートを使用する方法を示しています。リクエストの太字の部分が、この例に関連するスタイルシートです。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS BY CATEGORY BY PRODUCT
HEADING
"Sales Report"
FOOTING CENTER
"**End of Report**"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=HEADING, FONT=ARIAL, SIZE=12, STYLE=BOLD, $
TYPE=TITLE, STYLE=ITALIC, $
TYPE=DATA, COLUMN=N1, STYLE=BOLD, COLOR=BLUE, $
TYPE=FOOTING, COLOR=RED, STYLE=BOLD, $
ENDSTYLE
END
```

出力結果は次のとおりです。



Sales Report

Category	*Product*	*Unit Sales*	*Dollar Sales*
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

****End of Report****

スタイルシートファイルへの別スタイルシートの追加

```
INCLUDE = stysheet,$
```

説明

stysheet

追加するスタイルシートファイルです。

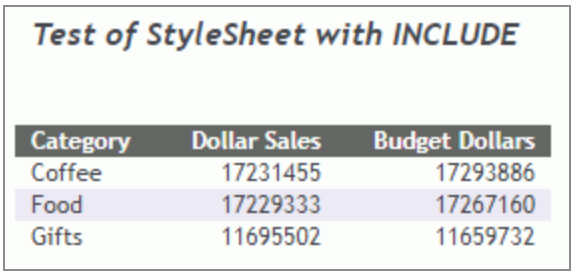
スタイルシート宣言は、スタイルシートで検出された順序で適用されます。このため、スタイルシートファイルを追加し、このファイル内の属性のいくつかを上書きする場合は、最初に INCLUDE ステートメントを配置し、その下に特定の属性を上書きする宣言を記述します。

スタイルシートファイルへの別スタイルシートの追加

次のリクエストは、インラインレポートスタイルシートに、WebFOCUS に付属のスタイルシートを追加し、見出しスタイルを上書きして太字、斜体に設定します。

```
HEADING CENTER
"Test of Stylesheet with Include"
" "
SUM DOLLARS BUDDOLLARS
BY CATEGORY
ON TABLE HOLD AS STYLE2 FORMAT HTML
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/javaassist/intl/EN/ENIADefault_combine.sty,$
TYPE=HEADING, STYLE = BOLD+ITALIC,$
END
```

出力結果は次のとおりです。



Test of StyleSheet with INCLUDE

Category	Dollar Sales	Budget Dollars
Coffee	17231455	17293886
Food	17229333	17267160
Gifts	11695502	11659732

WebFOCUS スタイルシートファイルの作成と適用

スタイルシートを別のファイルとして作成し、それを任意のレポートに適用することができます。スタイルシートファイルは、構文と追加コメントのみで構成されます。インラインスタイルシートとは異なり、スタイルシートファイルでは ON TABLE SET STYLESHEET および ENDSTYLE コマンドは使用しません。スタイルシートファイルをレポートに適用するには、SET STYLESHEET コマンドを使用します。詳細は、WebFOCUS スタイルシートの作成を参照してください。スタイルシート宣言についての詳細は、一般的な WebFOCUS スタイルシート構文を参照してください。

新しいスタイルシートを作成する代わりに、WebFOCUS から提供されるサンプルスタイルシートファイルのいずれかをテンプレートとして使用することもできます。

新しいスタイルシートファイルを作成する場合でも、既存のファイルをコピーしてカスタマイズする場合でも、作成したファイルを正しい場所に保存する必要があります。詳細は、WebFOCUS スタイルシートの作成を参照してください。

WebFOCUS スタイルシートファイルの名前を付けて保存

作成したスタイルシートは、その使用目的に応じて異なる場所に保存します。

- **WebFOCUS で使用**　WebFOCUS リポジトリにスタイルシートファイルをアップロードし、ユーザがコンテンツの作成権限を所有するフォルダに保存します。また、スタイルシートを参照するプロシジャの実行ユーザが、そのフォルダで読み取り権限を所有している必要があります。

  セルフサービスのアプリケーション用にスタイルシートを作成する場合は、スタイルシートファイルを apps¥baseapp ディレクトリに展開して、複数のアプリケーションで共有させることができます。また、スタイルシートファイルを特定のレポートプロシジャのみに適用する場合は、そのプロシジャと同一の場所にファイルを展開することもできます。

作成したスタイルシートファイルには、「filename.sty」という名前を付ける必要があります。ここで、filename には文字、数字、アンダスコア (_) が使用できるほか、ファイルを保存するオペレーティングシステムで有効なものを組み合わせて使用することができます。

WebFOCUS スタイルシートファイルのレポートへの適用

リクエストの先頭でスタイルシートファイルを適用するには、次の構文を使用します。

```
SET STYLE[SHEET] = stylesheet
```

リクエスト内にスタイルシートファイルを適用するには、次の構文を使用します。

```
ON TABLE SET STYLE[SHEET] stylesheet
```

説明

SHEET

この部分を省略してコマンドの長さを短くすることができます。コマンドの動作には影響しません。

stylesheet

スタイルシートファイルの名前です。ファイル拡張子は含めません。

一般的な WebFOCUS スタイルシート構文

スタイルシートは、フォーマットを設定するレポートコンポーネントを選択する構文とそのコンポーネントに適用するフォーマット形式で構成されます。通常、構文は TYPE 属性で開始し、その後にレポートコンポーネントに割り当てる attribute=value を指定します。スタイルシートにコメントを追加して、そのスタイルシートのコンテキストを記述することもできます。コメントは、スタイルシートの動作には影響しません。詳細は、WebFOCUS スタイルシートへのコメントの追加を参照してください。

レポートコンポーネントの特定についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

WebFOCUS スタイルシート宣言の指定

スタイルシート宣言では、次の形式で一連の属性を指定します。

```
attribute = value, [attribute = value, ...] $
```

説明

attribute

TYPE、COLUMN、COLOR、FONT などの属性です。

value

属性に割り当てる値です。

サンプルの WebFOCUS スタイルシート

次のリクエストでは、インラインスタイルシートを使用します。スタイルシートは、ON TABLE SET STYLESHEET * で開始され、ENDSTYLE で終了します。

```
TABLE FILE CENTORD
HEADING
" "
"C e n t u r y  C o r p o r a t i o n"
" "
"Order Revenue - 2000 Q3"
" "
"page <TABPAGENO"
" "
SUM ORDER_DATE/MtDY ORDER_NUM LINEPRICE AS 'Order,Total:'
BY LOWEST 9 ORDER_DATE NOPRINT
WHERE (ORDER_DATE GE '2000/10/01') AND (ORDER_DATE LE '2000/12/31');
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON

ON TABLE SET STYLESHEET *
TYPE=Report, GRID=Off, UNITS=Inches, $
TYPE=Data, FONT='Times', $
TYPE=Data, BACKCOLOR=Aqua, COLOR=Navy,
     WHEN=LinePrice GT 500000, $
TYPE=Data, COLUMN=LinePrice, BACKCOLOR=Aqua, COLOR=Navy, STYLE=Bold,
     WHEN=LinePrice GT 500000, $
TYPE=Title, FONT='Helvetica', $
TYPE=Heading, FONT='Helvetica', STYLE=Bold, SIZE=14, JUSTIFY=Center,
     COLOR=White, BACKCOLOR=Dark Turquoise, $
TYPE=Heading, LINE=6, BACKCOLOR=White, COLOR=Dark Turquoise, $
TYPE=Heading, LINE=7, BACKCOLOR=White, $
ENDSTYLE

END
```

WebFOCUS スタイルシートの読みやすさの向上

さまざまな方法を駆使してスタイルシートを構成することで、スタイルシートをより読みやすくすることができます。次の方法のいずれかを使用したり、複数の方法を組み合わせたりすることができます。

- 先頭にブランクまたはタブを配置してからフィールドの構文を開始します。
- 2 つの構文間にブランク行を追加します。
- すべて大文字、すべて小文字、大文字小文字の組み合わせを使用して構文を作成します。
- 1 つのレポートコンポーネントのフォーマット設定に、複数の構文を使用します。
- 属性、等号 (=)、値、カンマ (,)、ドル記号 ($) の間にブランクまたはタブを追加します。
- 1 行に収まらない構文を分割します。構文を分割しても、終了を表すドル記号 ($) まで処理されます。たとえば、1 つの構文を次のように分割することができます。

```
TYPE=HEADING, FONT=ARIAL,
SIZE=14, STYLE=BOLD, $
```

- 1 行に収まらない attribute=value を分割します。1 行に収まらない属性または値を分割する場合、最初の行の末尾に継続構文として円記号 (¥) を使用します。以下はその例です。

```
TYPE=TITLE, COLUMN=N2, STY¥
LE=BOLD+ITALIC, COLOR=BLUE, $
```

WebFOCUS スタイルシートへのコメントの追加

スタイルシートにコメントを追加して、構文にコンテキストを含めます。コメントは、スタイルシートの動作には影響しません。

次の行にコメントを追加することができます。

- **構文の行**　ドル記号 ($) の後に任意のテキストを追加します。以下はその例です。

```
TYPE=HEADING, STYLE=BOLD, COLOR=BLUE, SIZE=14, $ Sample comment
```

- **コメント専用の行**　ドル記号 ($) またはハイフンとアスタリスクの組み合わせ (-*) のいずれかで行を開始し、その後に任意のテキストを追加します。以下はその例です。

```
-* This is a sample comment
$ This is another sample comment
```

注意：コメントは、スタイルシート内に限らず、リクエスト内の任意の場所に追加することができます。

マクロを使用した WebFOCUS スタイルシート宣言の再利用

スタイルシート宣言に特定の属性群をよく使用する場合は、これらの属性群を 1 つのグループにまとめたスタイルシートマクロを作成しておき、そのマクロをスタイルシートで再利用することにより、繰り返しコーディングする手間を省くことができます。

WebFOCUS スタイルシートマクロの定義

スタイルシートマクロは、そのマクロを参照するスタイルシートで定義し、スタイルシートにマクロ定義を最初に配置した後にそのマクロを使用する必要があります。

マクロを定義するには、使用するスタイル属性が後に続く DEFMACRO 属性を使用します。

WebFOCUS スタイルシートマクロの定義

```
DEFMACRO = macroname, attribute1 = value1, [attribute2 = value2,]... $
```

説明

macroname

作成するマクロに割り当てる名前です。

attribute

スタイルシート属性です。たとえば、レポートコンポーネントのフォーマット設定、グラフの挿入、ハイパーリンクの定義、条件付きフォーマット設定 (WHEN) の条件の適用などに使用する属性があります。

value

属性に割り当てる値です。

WebFOCUS スタイルシートマクロの適用

スタイルシートマクロは、そのマクロで定義したすべてのフォーマット設定を、構文で指定したレポートコンポーネントに適用します。マクロを適用するには、MACRO 属性を使用します。1 つの構文に適用できるマクロは 1 つのみです。

スタイルシートマクロをレポートコンポーネントに適用する際に、マクロで定義した属性のいずれかを上書きすることができます。その場合は、構文に MACRO 属性を使用した後、同一の属性に新しい値を指定します。

WebFOCUS スタイルシートマクロの適用

```
TYPE=type, [subtype,] MACRO=macroname, [condition,] $
```

説明

type

マクロを適用するレポートコンポーネントです。任意のレポートコンポーネントを指定することができます。

subtype

追加の属性です。マクロを適用するレポートコンポーネントを選択する際に必要です。たとえば、COLUMN、ACROSS、ITEM などがあります。異なるタイプのレポートコンポーネントを指定する方法についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

macroname

指定したレポートコンポーネントに適用するマクロの名前です。マクロは、同一のスタイルシートで定義する必要があります。

condition

この構文を条件付きにする場合に、オプションとして指定する WHEN 属性です。条件付き構文についての詳細は、レポートフォーマット設定の管理を参照してください。

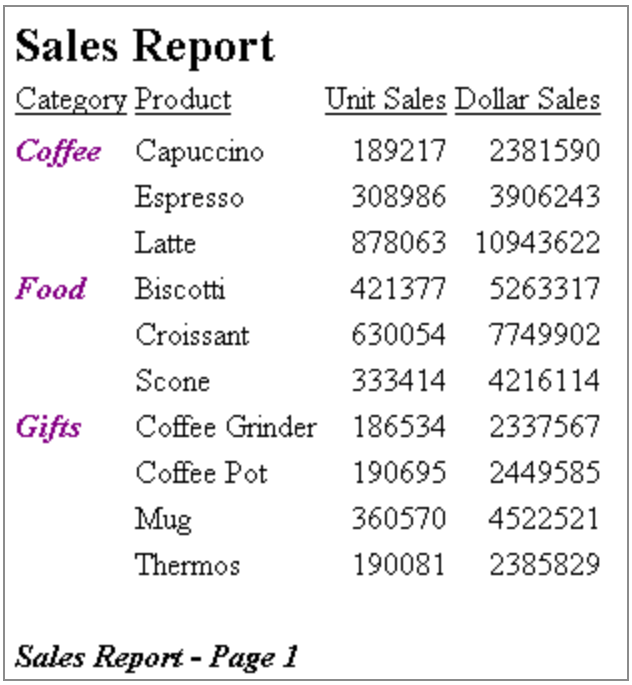
WebFOCUS スタイルシートマクロの定義、適用、上書き

次の注釈付きの例は、スタイルシートでマクロを定義、適用、上書きする方法を示しています。

```
   TABLE FILE GGSALES
    SUM UNITS DOLLARS
    BY CATEGORY BY PRODUCT
    HEADING
    "Sales Report"
    FOOTING
    "Sales Report - Page <TABPAGENO"
    ON TABLE SET STYLE *
    TYPE=REPORT, GRID=OFF,$
1.DEFMACRO=A, STYLE=BOLD, SIZE=12, $
2.DEFMACRO=BI, STYLE=BOLD+ITALIC, COLOR=PURPLE, $
3.TYPE=HEADING, MACRO=A, $
4.TYPE=FOOTING, MACRO=BI, COLOR=BLACK, $
5.TYPE=DATA, COLUMN=N1, MACRO=BI, $
   ENDSTYLE
   END
```

1. マクロ A を定義します。
2. マクロ BI を定義します。
3. マクロ A を見出しに適用します。
4. マクロ BI を脚注に適用し、COLOR=BLACK 属性を使用してマクロの一部を上書きします。
5. マクロ BI を BY フィールドである CATEGORY (TYPE=DATA、COLUMN=N1 で指定) のデータに適用します。

出力結果は次のとおりです。



Sales Report

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

Sales Report - Page 1

WebFOCUS スタイルシート属性の継承

スタイルシート属性は、親のレポートコンポーネントから子のレポートコンポーネントに継承されます。継承された属性を上書きするには、子のコンポーネントの構文でその属性を明示的に指定し、新しい値を割り当てます。各コンポーネントは属性を自動的に継承するため、新しく指定する属性は、コンポーネントが継承した属性と異なるもの、または補足するものに限られます。

属性が継承されることから、1 つの構文に共通のフォーマット設定を定義すると、その設定がすべての子のコンポーネントに自動的に適用されます。ただし、これらのコンポーネントの中で、異なる属性値を指定し、継承した値を上書きした場合を除きます。これにより、繰り返しコーディングする手間が省けるとともに、簡素なスタイルシートを作成することができます。

たとえば、次のようにすべてのレポートタイトルを青色の太字で表示するよう指定することができます。

```
TYPE=TITLE, COLOR=BLUE, STYLE=BOLD, $
```

デフォルト設定により、青色の太字で表示するフォーマット設定が各フィールドタイトルに継承されます。ここで、次のようにフィールドの 1 つを指定して、青色は継承するが、太字は無効にし、さらに背景色に黄色を追加するようフォーマット設定を変更することができます。

```
TYPE=TITLE, COLUMN=N2, STYLE=-BOLD, BACKCOLOR=YELLOW, $
```

WebFOCUS スタイルシート継承の階層

各レポートコンポーネントは、階層に基づいてスタイルシートを継承します。スタイルシート宣言で TYPE=REPORT を指定した場合、階層のルートはレポート全体になります。スタイルシート宣言で TYPE を指定しない場合、デフォルトの TYPE=REPORT となり、この属性がレポート全体に適用されます。レポート全体に対して属性が指定されていない場合は、HTML または PDF などのレポート表示フォーマットに基づいて各属性にそれぞれのデフォルト値が適用されます。

各レポートコンポーネントは、その親のコンポーネントから属性を継承します。コンポーネント X がコンポーネント Y の親になる条件は、コンポーネント Y を指定するタイプ関連属性の一部がコンポーネント X を指定していること、およびこれらに共通するタイプ関連属性に同一の値が指定されていることです。以下はその例です。

- TYPE=x、subtype=y、elementtype=z で指定されたコンポーネント x は、TYPE=x、subtype=y で指定されたコンポーネントの子となり、親のコンポーネントから属性を継承します。
- TYPE=x、subtype=y で指定されたコンポーネントは、TYPE=x で指定されたコンポーネントの子となり、親のコンポーネントから属性を継承します。
- TYPE=x で指定されたコンポーネントは、レポート全体の (TYPE=REPORT) の子となり、レポート全体から属性を継承します。ここで、x は REPORT 以外の任意の値を表します。
- TYPE=DATA、subtype=z で指定されたコンポーネントは、TYPE=REPORT、subtype=z からは継承されません。ここでは、個別コンポーネント (subtype=z) は、同一タイプのスタイルからのみ継承されるという規則が適用されます。唯一の例外は、最上位レベルの TYPE=REPORT コンポーネント (COLUMN などのその他のコンポーネントを伴わない TYPE=REPORT 単独のデフォルトスタイル) からは、いずれのスタイルも継承可能であるという点です。

  たとえば、次の構文で、DATA レベルでの COLUMN への特定のスタイルを定義することで、TYPE=REPORT からのスタイル全体 (FONT=TAHOMA) が適用されますが、

TYPE=REPORT, COLUMN=N2 (COLOR=GREEN) は適用されません。COLOR=GREEN を DATA COLUMN=N2 に適用するには、明示的にこの色を指定する必要があります。

```
TYPE=REPORT, FONT=TAHOMA, $
TYPE=REPORT, COLUMN=N2, COLOR=GREEN, $
TYPE=DATA, COLUMN=N2, STYLE=+UNDERLINE+BOLD, $
```

外部カスケードスタイルシート (CSS) を使用する場合、レポートコンポーネントは、親のレポートコンポーネントからではなく、親の HTML 要素からフォーマット設定を継承します。

継承した WebFOCUS スタイルシート属性の補足

次の例は、継承したスタイルシート属性を補足する方法を示しています。リクエストの太字の部分が、この例で説明するスタイルシート宣言です。

このレポートには、2 行のページ見出しがあります。最初のスタイルシート宣言で、レポートコンポーネントの HEADING がフォントサイズ 12 ポイントの太字でフォーマット設定されます。この構文により、見出しの両方の行がこのスタイルでフォーマット設定されます。

2 つ目の構文を追加して見出しの行番号および追加のスタイル特性を指定し、見出しの 2 行目のフォーマットを補足します。ここでは、構文中の TYPE=HEADING、LINE=2、STYLE=ITALIC を追加しています。見出しの 2 行目は、最初の HEADING 構文から太字スタイルおよび 12 ポイントのフォントサイズを継承し、さらに 2 行目の構文で斜体スタイルが定義されています。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY CATEGORY BY PRODUCT
HEADING
"Sales Report:"
"First Quarter"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=HEADING, STYLE=BOLD, SIZE=12, $
TYPE=HEADING, LINE=2, STYLE=ITALIC, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Sales Report:
First Quarter

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

継承した WebFOCUS スタイルシート属性の上書き

次の例は、スタイルシートの継承を上書きする方法を示しています。リクエストの太字の部分が、この例で説明するスタイルシート宣言です。

```
   TABLE FILE GGSALES
HEADING
"Sales Report"
SUM UNITS DOLLARS
BY CATEGORY BY PRODUCT BY DATE NOPRINT
WHERE DATE GE 19960101 AND DATE LE 19960401
ON TABLE SET STYLEMODE PAGED
ON TABLE SET LINES 20
FOOTING
"Page <TABPAGENO of <TABLASTPAGE"
ON TABLE SET STYLE *
   TYPE=REPORT, GRID=OFF, $
1.TYPE=REPORT, BACKCOLOR=BLUE, COLOR=WHITE, $
2.TYPE=HEADING, BACKCOLOR=WHITE, COLOR=BLACK, STYLE=BOLD, SIZE=12, $
3.TYPE=FOOTING, SIZE=11, STYLE=BOLD+ITALIC, BACKCOLOR=WHITE,
    COLOR=BLACK, $
4.TYPE=FOOTING, OBJECT=FIELD, ITEM=1, STYLE=-ITALIC, $
```

```
ENDSTYLE
END
```

1. レポート全体 (すべてのコンポーネント) にフォーマットを設定し、青色の背景色および白色のフォントで表示します。
2. 白色の背景色および黒色のフォントを指定して、ページ見出し (TYPE=REPORT 構文で定義) に継承されたフォーマット設定を上書きします。
3. ページ脚注をフォントサイズ 11 ポイントの太字の斜体でフォーマット設定し、BACKCOLOR=WHITE および COLOR=BLACK を指定してレポートの色設定を上書きします。
4. <TABPAGENO システム変数はページ脚注の一部であるため、最初の TYPE=FOOTING 構文で指定されたフォーマット設定のすべてを継承します。この構文では OBJECT=FIELD、ITEM=1 が指定され、斜体 (STYLE=-ITALIC) が削除され、ページ脚注に継承されたフォーマット設定が上書きされます。なお、脚注に 2 つの埋め込みフィールドがあることから、ITEM=1 を指定する必要があります。

出力結果は次のとおりです。

Sales Report

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	5507	71728
		5219	67429
		6343	87290
		4170	58692
	Espresso	14499	182900
		14015	169023
		14871	184474
		11613	156396
	Latte	31469	401873
		38725	490645
		42717	520948
		36374	468137

Page 1 of 4

JAWarm スタイルシートを使用したレポートの作成

JAWarm スタイルシート (JAWarm.sty) は、WebFOCUS ツールセットの新しいデフォルトスタイルシートです。作成するレポートおよびグラフのスタイルオプションで、このスタイルシートのクリーンでシンプルなレイアウトとデザインを活用することができます。統一されたテキストと配色、共通の外観が提供されるため、すべてのレポートおよびグラフ全体のアクティビティが予測しやすくなります。

また、JAFlat スタイルシートは、JAWarm と同等の外観を持ちますが、配色が異なります。Analytic Document では、このスタイルシートの使用をお勧めします。

ここでは、JAWarm スタイルシートの仕様、およびこのスタイルシートのレポートおよび Analytic Document への適用方法について説明します。

レポートのスタイル設定

ここでは、JAWarm スタイルシートの仕様、およびこのスタイルシートのレポートスタイルへの適用方法について説明します。

データ、レポート、タイトルのスタイル設定

下図は、データ、レポート、タイトル要素へのスタイル設定を示しています。

Product Category	Product Subcategory	Revenue	Quantity Sold	Gross Profit
Accessories	Charger	$4,022,834.91	105,257	$1,970,123.91
	Headphones	$76,186,587.97	228,349	$24,523,023.97
	Universal Remote Controls	$49,398,915.65	178,061	$13,361,292.65
Camcorder	Handheld	$41,970,570.97	250,167	$21,393,654.97
	Professional	$44,053,830.75	12,872	$8,835,522.75
	Standard	$68,441,300.52	192,205	$19,369,667.52

タイトル

- 太字

レポート

- Arial
- 9 pt
- 文字色 - RGB (20, 20, 20)
- タイトル行のスキップ
- ハイパーリンク色 - RGB (51, 102, 255)
- ページ色 - 白

データ

- 上部境界 - RGB (219, 219, 219)
- 下部境界 - RGB (219, 219, 219)
- 上部間隔 - 0.05
- 下部間隔 - 0.05

見出しおよび脚注スタイル

見出しと脚注は、レポートとページ内で下図のように定義されています。

Product Category	Product Subcategory	Revenue	Quantity Sold	Gross Profit
Accessories	Charger	$4,022,834.91	105,257	$1,970,123.91
	Headphones	$76,186,587.97	228,349	$24,523,023.97
	Universal Remote Controls	$49,398,915.65	178,061	$13,361,292.65
Camcorder	Handheld	$41,970,570.97	250,167	$21,393,654.97
	Professional	$44,053,830.75	12,872	$8,835,522.75
	Standard	$68,441,300.52	192,205	$19,369,667.52

JAWarm スタイルシートのスタイル設定は、次のとおりです。

見出し

- 14 pt
- 太字
- 文字色 - RGB (75, 75, 75)
- 左揃え

ページ見出し

- 12 pt
- 太字
- 文字色 - RGB (75, 75, 75)
- 左揃え

ページ脚注

- 10 pt
- 文字色 - RGB (102, 102, 102)

ページ脚注

- 10 pt
- 文字色 - RGB (102, 102, 102)

中間見出しおよび中間脚注のスタイル設定

下図は、中間見出し要素ののスタイル設定を示しています。

Product Subcategory	Revenue	Quantity Sold	Gross Profit
Subhead: **Accessories**			
Charger	$4,022,834.91	105,257	$1,970,123.91
Headphones	$76,186,587.97	228,349	$24,523,023.97
Universal Remote Controls	$49,398,915.65	178,061	$13,361,292.65
Subhead: **Camcorder**			
Handheld	$41,970,570.97	250,167	$21,393,654.97
Professional	$44,053,830.75	12,872	$8,835,522.75
Standard	$68,441,300.52	192,205	$19,369,667.52

中間見出し

- 背景色 - RGB (246, 246, 246)
- 上部境界 - RGB (219, 219, 219)

中間見出しデータ

- 太字

下図は、中間脚注要素ののスタイル設定を示しています。

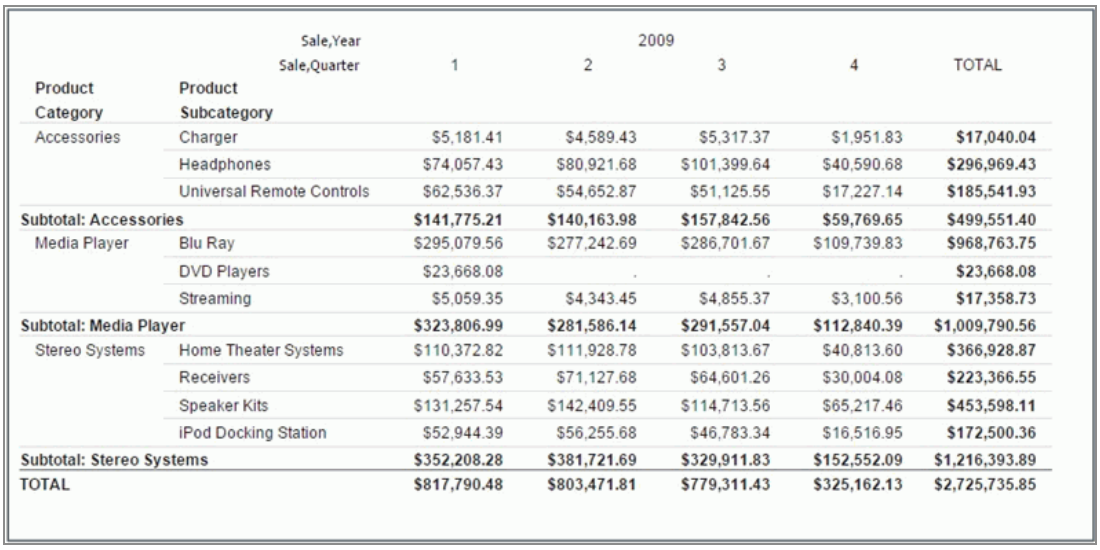
Product Subcategory	Revenue	Quantity Sold	Gross Profit
Charger	$4,022,834.91	105,257	$1,970,123.91
Headphones	$76,186,587.97	228,349	$24,523,023.97
Universal Remote Controls	$49,398,915.65	178,061	$13,361,292.65
Subfoot: **Accessories**			
Handheld	$41,970,570.97	250,167	$21,393,654.97
Professional	$44,053,830.75	12,872	$8,835,522.75
Standard	$68,441,300.52	192,205	$19,369,667.52
Subfoot: **Camcorder**			

中間脚注データ

- 太字

ACROSS のスタイル設定

下図は、レポートの ACROSS のスタイル設定を示しています。



	Sale,Year	2009				
	Sale,Quarter	1	2	3	4	TOTAL
Product Category	Product Subcategory					
Accessories	Charger	$5,181.41	$4,589.43	$5,317.37	$1,951.83	**$17,040.04**
	Headphones	$74,057.43	$80,921.68	$101,399.64	$40,590.68	**$296,969.43**
	Universal Remote Controls	$62,536.37	$54,652.87	$51,125.55	$17,227.14	**$185,541.93**
Subtotal: Accessories		**$141,775.21**	**$140,163.98**	**$157,842.56**	**$59,769.65**	**$499,551.40**
Media Player	Blu Ray	$295,079.56	$277,242.69	$286,701.67	$109,739.83	**$968,763.75**
	DVD Players	$23,668.08	.	.	.	**$23,668.08**
	Streaming	$5,059.35	$4,343.45	$4,855.37	$3,100.56	**$17,358.73**
Subtotal: Media Player		**$323,806.99**	**$281,586.14**	**$291,557.04**	**$112,840.39**	**$1,009,790.56**
Stereo Systems	Home Theater Systems	$110,372.82	$111,928.78	$103,813.67	$40,813.60	**$366,928.87**
	Receivers	$57,633.53	$71,127.68	$64,601.26	$30,004.08	**$223,366.55**
	Speaker Kits	$131,257.54	$142,409.55	$114,713.56	$65,217.46	**$453,598.11**
	iPod Docking Station	$52,944.39	$56,255.68	$46,783.34	$16,516.95	**$172,500.36**
Subtotal: Stereo Systems		**$352,208.28**	**$381,721.69**	**$329,911.83**	**$152,552.09**	**$1,216,393.89**
TOTAL		**$817,790.48**	**$803,471.81**	**$779,311.43**	**$325,162.13**	**$2,725,735.85**

ACROSS タイトル

- BY フィールドで右揃え

ACROSS データ

- 中央揃え

行合計データ

- 太字

中間合計および列合計のスタイル設定

中間合計

- 太字

列合計

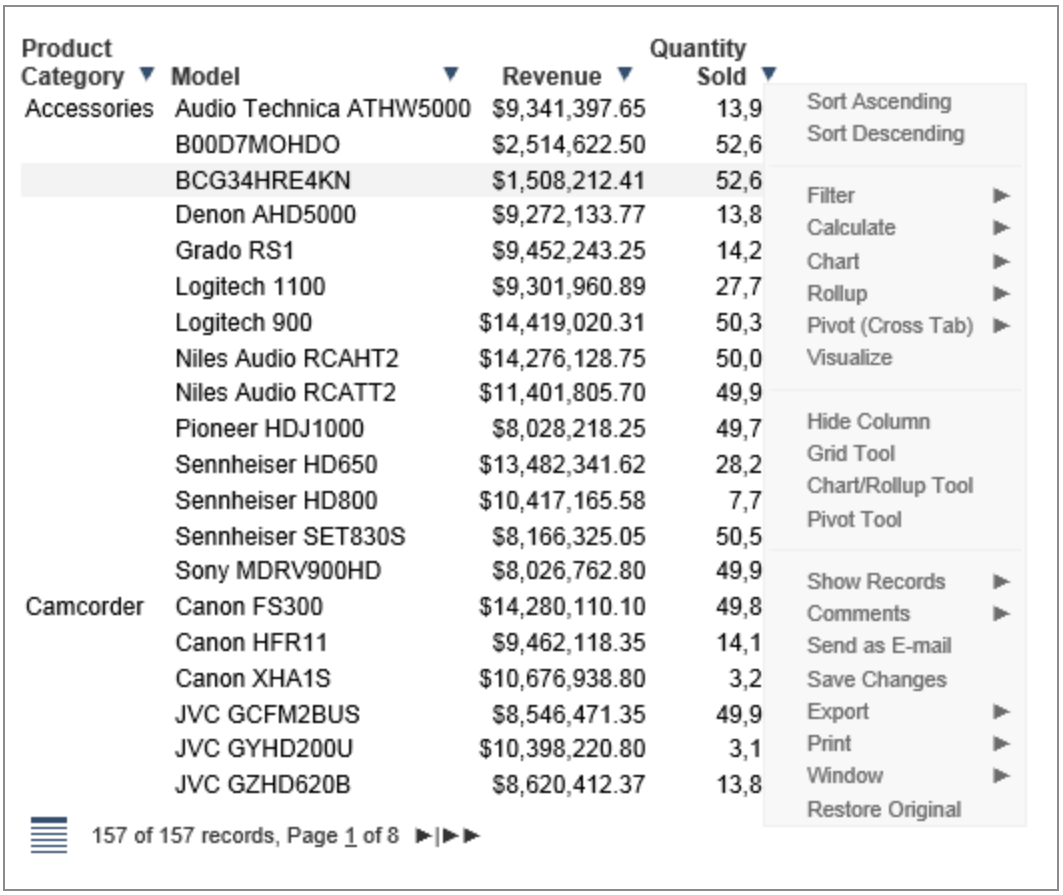
- 太字
- 上部境界 - RGB (102, 102, 102)

Analytic Document

ここでは、JAWarm スタイルシートの仕様、およびこのスタイルシートの Analytic Document への適用方法について説明します。

WebFOCUS Analytic Document のページ、メニュー、Hover テキストへのスタイル設定

下図は、WebFOCUS Analytic Document のページ、メニュー、Hover テキスト要素へのスタイル設定を示しています。



ページオプション

- 位置 - 下部
- 背景色 - なし
- 左揃え

メニュー

- テキスト色 - RGB (#6B6B6B)
- 背景色 - RGB (#F8F8F8)
- Hover テキスト色 - RGB (#495263)
- Hover 背景色 - RGB (#DFDFDF)

Hover

- 背景色 - RGB (243, 243, 243)

AR アイコンセット

- 青

JAWarm.sty 使用上の注意

- AHTML では、境界線はサポートされません。
- ソートオブジェクトには、複数の BY フィールドを含めることができます。プロシジャのソートオブジェクトが最初の BY フィールドの場合、JAwarm スタイルシートはその最初の BY フィールドに対して属性を定義します。そのため、このプロシジャで JAWarm スタイルシートはサポートされません。
- ソートオブジェクトでは、スタイルシートの BY フィールドへの参照はサポートされません。JAWarm スタイルシートは、最初の BY 中間見出しにスタイルを設定して区切りを区別します。
- 各データ行を区別するため、横方向の境界線が使用されます。レポートに境界線が存在する場合、次の動作が変更されます。
    - 見出しとタイトルの間のブランク行
    - データ末尾と脚注の間のブランク行
- FML バーには、バーラインが表示されません。

注意：上記の問題を回避するには、JAWarm スタイルシートの代わりに、jadeflt スタイルシートを使用してください。

レポートフォーマット設定の管理

レポートのフォーマットを設定する際に、フォーマット設定の適用方法を制御することができます。次の処理を実行することができます。

- **HTML レポート用の内部カスケードスタイルシートを生成する**　パフォーマンスが向上するとともに、さまざまなフォーマット設定オプションを使用することができます。
- **フォーマット設定を条件付きで適用する**　レポートの値に基づいてレポートの外観を条件付きで設定することができます。また、その条件に基づいてその他のレポートまたはインターネットリソースへのリンクを設定することもできます。たとえば、毎月の注文レポートを作成する場合、未払いの注文をすべて赤色で表示するとともに、顧客の最近の支払い履歴レポートにリンクさせるよう条件を指定することができます。
- **レポートの測定単位を指定する**　レポートのサイズとして上側マージンなどがあります。測定単位として、インチ、センチメートル、ポイントのいずれかを選択することができます。
- **空レポートの表示を制御する**　リクエストから返されるレコードがない場合に、データのないレポートを表示するか、出力結果がないことを告げるメッセージを表示するかのいずれかを選択することができます。

HTML レポート内部カスケードスタイルシートの生成

HTML フォーマットでレポートを作成する場合、レポートのフォーマットを指定するコードが生成されます。この HTML コードの一部として、内部カスケードスタイルシートを生成するようレポートを設定することができます。この方法には次のような利点があります。

- **パフォーマンスの向上**　HTML ファイルサイズの大幅な縮小、転送帯域幅の減少、大規模レポートの高速表示によりパフォーマンスが向上します。
- **フォーマット設定オプションの提供**　さまざまなフォーマット設定オプションを HTML レポートで使用することができます。WebFOCUS スタイルシート属性の中には、内部カスケードスタイルシートを生成するレポートの HTML 表示フォーマットでのみサポートされるものがあります。

内部カスケードスタイルシートにより HTML で有効になる属性として、UNITS、BOTTOMMARGIN、TOPMARGIN、LEFTMARGIN、RIGHTMARGIN、SIZE、POSITION、WRAP、PAGECOLOR があります。また、ほとんどのレポートコンポーネントで下線の追加と削除を行えるとともに、イメージの開始位置およびサイズを指定することができるようになります。UNITS 属性についての詳細は、測定単位の選択を参照してください。その他の属性についての詳細は、レポートページのレイアウトを参照してください。

内部カスケードスタイルシートを生成した場合でも、同一のリクエストに外部カスケードスタイルシートを適用することができます。フォーマット設定の命令が競合する場合は、内部カスケードスタイルシートが外部カスケードスタイルシートに優先します。

注意：一般的に、1 つのレポートまたはスタイルシート内に、ネイティブの WebFOCUS スタイルシート属性と外部カスケードスタイルシート (CSS) クラスの両方を指定しないことをお勧めします。

内部カスケードスタイルシートの生成

HTML フォーマットのレポートでは HTML コードが生成されますが、そのコードの一部として内部カスケードスタイルシートを作成することができます。次の方法を使用します。

リクエスト外で設定　次の構文を使用します。

```
SET HTMLCSS = {ON|OFF}
```

リクエスト内で設定　次の構文を使用します。

```
ON TABLE SET HTMLCSS {ON|OFF}
```

説明

ON

HTML 出力結果に、レポートの外観の大部分を制御する内部カスケードスタイルシートを生成します。デフォルト値は ON です。

OFF

内部カスケードスタイルシートを生成しません。ただし、レポートの作成に使用した HTML テーブルの各セルに、フォーマット設定用のタグが配置されます。

内部カスケードスタイルシートの要件

内部カスケードスタイルシート (CSS) でフォーマットを設定したレポートを表示する場合は、カスケードスタイルシートをサポートする Web ブラウザを使用する必要があります。Microsoft Internet Explorer バージョン 5.0 以降では、カスケードスタイルシートがサポートされています。

なお、内部カスケードスタイルシートを使用した場合にレポートがどのようにフォーマット設定されるかについては、WebFOCUS ではなく、Web ブラウザのカスケードスタイルシートのサポートおよび実装状況により異なります。各種 Web ブラウザの中には、最新の CSS バージョンを完全にサポートしないものや、CSS 機能を異なる方法で実装するものがあります。

測定単位の選択

PDF のレポートを作成する場合と同様に、内部カスケードスタイルシートを生成する HTML レポートのページマージンおよび列幅の測定単位を選択することができます。さらに、PDF のレポート内の位置を指定する測定単位を選択することもできます。選択できる測定単位として、インチ、センチメートル、ポイントがあります。

測定単位を変更すると、現在の使用単位がすべて新しい単位に自動変換されます。たとえば、単位がインチで、レポートの上側マージンが 1 に設定されている場合、測定単位をセンチメートルに変更すると、上側マージンの長さは自動的に 1 センチメートルに変換されます。

測定単位は、次の方法で設定します。

- **スタイルシート**　UNITS 属性を使用します。
- **SET コマンド**　UNITS パラメータを使用します。

測定単位の設定

測定単位を設定するには、次の構文を使用します。

スタイルシートで設定　次の属性を追加します。

```
UNITS = units
```

リクエスト外で設定　次の構文を使用します。

```
SET UNITS = units
```

リクエスト内で設定　次の構文を使用します。

```
ON TABLE SET UNITS units
```

説明

units

測定単位です。選択できる値には次のものがあります。

- **INCHES**　測定単位としてインチを指定します。これがデフォルト値です。
- **CM**　測定単位としてセンチメートルを指定します。
- **PTS**　測定単位としてポイントを指定します。ポイントは、活字表記では一般的な測定単位です。

スタイルシートでの条件付きフォーマット設定、表示、リンク

レポートの値に基づいて、レポートコンポーネントのフォーマット、グラフ表示、レポートへのリンクの組み込みを、条件付きで設定することができます。条件付きスタイルを使用して、次のことを行えます。

- レポートの特定の項目を強調する。
- 重要な値と値の差を強調する。
- ユーザがレポート内またはレポート間を移動して情報を参照できるようリソースをカスタマイズする。

レポートを条件付きでフォーマット設定するには、スタイルシートに WHEN 属性を追加します。WHEN 属性で指定した条件がレポートコンポーネントの各インスタンスに対して評価されます。レポートコンポーネントとは、表形式レポートフィールドの各セル、グラフ内の各要素、自由形式レポートの各ページを指します。スタイルシートは、条件を満たすすべてのインスタンスに適用されますが、条件を満たさないすべてのインスタンスでは無視されます。

また、連続条件付きのフォーマット設定を適用することもできます。

注意：条件付きスタイル設定 (WHEN) を使用してスタイルを定義する際に、TABPAGENO および TABLASTPAGE 変数を使用することはできません。

連続条件付きフォーマット設定の適用

1 つのレポートコンポーネントに連続条件のロジックを適用するには、一連の構文を作成し、各構文にそれぞれ異なる条件を設定します。このスタイルシートは、ネストされた一連の IF-THEN-ELSE ステートメントに相当します。1 つのレポートコンポーネント (例、同一フィールド) が複数の条件付き構文で指定され、かつ 1 つのフィールドがこれらの条件で評価される場合は、1 つのグループとして一括処理されます。レポートコンポーネントの各インスタンス (例、フィールドの各セル) は次のように処理されます。

1. グループ内の一連の条件付き構文は、スタイルシートで指定した順序で、そのいずれかの条件が満たされるまで評価されます。続いて、条件を満たす構文がレポートコンポーネントのインスタンスに適用されます。グループ内のその他の条件付き構文および同一のレポートコンポーネントと属性を指定する条件なしの構文は、そのインスタンスでは無視されます。
2. ただし、そのインスタンスで条件付き構文がまったく満たされない場合は、同一の属性が指定されたレポートコンポーネントに対する最初の条件なしの構文が、そのインスタンスに適用されます。
3. 手順 1 または 2 でインスタンスにまだ適用されていない属性がレポートコンポーネントで指定されている場合、そのレポートコンポーネントに対する条件なしの宣言がインスタンスに適用されます。
4. このレポートコンポーネントの次のインスタンス (例、フィールドの次のセル) に対して、この処理全体が繰り返し実行されます。

スタイルシートでの条件付きのフォーマット設定、表示、リンク

```
TYPE=type, [subtype,] attributes, WHEN=field1 operator {field2|value},$
```

または

```
TYPE=type, [subtype,] attributes, WHEN=FORECAST, $
```

説明

type

TYPE 属性の値です。任意のレポートコンポーネントを指定することができます。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

subtype

追加の属性です。宣言を適用するレポートコンポーネントを選択する際に必要です。たとえば、COLUMN、ACROSS、ITEM などがあります。

attributes

スタイルシートで指定する属性で、WHEN 属性を指定することにより条件付きとなります。ここには、ほとんどのフォーマット設定、イメージ、ハイパーリンクの属性を使用することができます。

field1, field2

比較するレポートフィールドを指定します。次のフィールドを指定することができます。

- グラフレポートまたは表形式レポートの表示フィールド名、BY フィールド名、ACROSS フィールド名。ACROSS 値を条件式の一部として使用することで、テーブル内のセルごとにスタイル属性を定義することができます。
- グラフレポートまたは表形式レポートの列参照。
- 自由形式レポートの見出しまたは脚注に埋め込まれたフィールド名。

レポートに表示したくないフィールドを使用する場合は、リクエストでそのフィールドを指定し、NOPRINT オプションを使用してそのフィールドを非表示にします (例、PRINT fieldname NOPRINT)。

レポートのフィールドに演算接頭語を適用する場合は、次のことを行えます。

- WHEN 属性に同一の演算接頭語を使用することができます。WHEN 属性でフィールドを参照する場合は、フィールドの名前を指定する必要があります (例、WHEN=AVE.PRICE GT 300)。
- WHEN 属性でフィールドを参照する場合に、フィールドの位置を指定して、演算接頭語を省略することができます (例、WHEN=N3 GT 300)。この方法は、ST. および CT. 演算接頭語ではサポートされません。

- WHEN 属性で複合ブール式を使用することはできません。

このフィールドを、パック 10 進数 (P) の数値フィールドにすることはできません。

operator

条件の適合性を判断する方法を定義します。使用できる関係演算子には次のものがあります。

EQ - 左辺と右辺の値が等しい場合に条件が満たされます。比較する値が文字の場合は、その大文字と小文字が一致しなければなりません。

NE - 左辺と右辺の値が等しくない場合に条件が満たされます。

LT - 左辺の値が右辺の値より小さい場合に条件が満たされます。

LE - 左辺の値が右辺の値以下の場合に条件が満たされます。

GT - 左辺の値が右辺の値より大きい場合に条件が満たされます。

GE - 左辺の値が右辺の値以上の場合に条件が満たされます。

value

数値、文字列、日付などの定数です。文字列や日付などの数値以外の定数は、一重引用符 (') で囲む必要があります。

ここでは、関数または演算子を使用して値を指定することはできませんが、関数または演算子を使用して一時項目 (COMPUTE または DEFINE) を定義し、その一時項目をレポートに使用し、定数の代わりとして指定することができます。

FORECAST

FORECAST コマンドを使用して生成するフィールドを指定します。

連続条件付きのフォーマット設定の使用

次の例は、連続条件付きのフォーマット設定をレポートに適用する方法を示しています。このレポートでは、連続条件付きのロジックを使用し、注文合計 (LINEPRICE) の値に基づいてそれぞれの行にフォーマットを設定します。

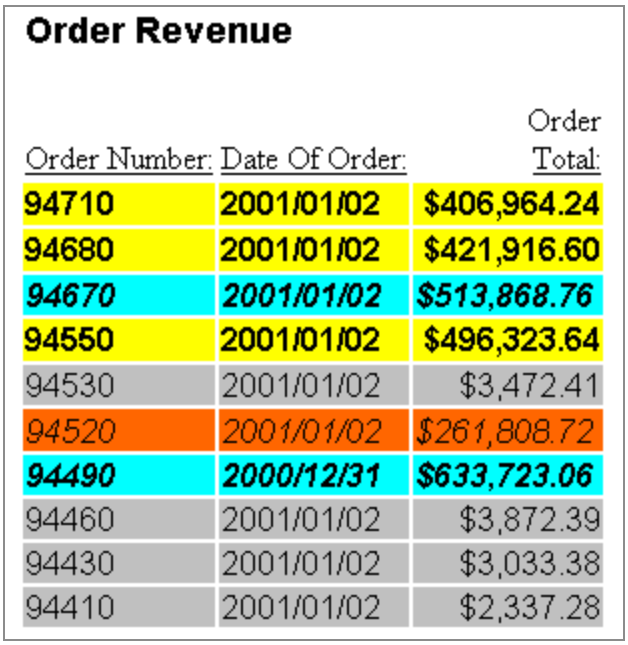
```
TABLE FILE CENTORD
HEADING
"Order Revenue"
" "
SUM ORDER_DATE LINEPRICE AS 'Order,Total:'
BY HIGHEST 10 ORDER_NUM
```

```
   ON TABLE SET PAGE-NUM OFF

   ON TABLE SET STYLESHEET *
   TYPE=REPORT, GRID=OFF, $
1.TYPE=DATA, BACKCOLOR=AQUA, STYLE=BOLD+ITALIC,
   WHEN=LINEPRICE GT 500000, $
2.TYPE=DATA, BACKCOLOR=YELLOW, STYLE=BOLD,
   WHEN=LINEPRICE GT 400000, $
3.TYPE=DATA, BACKCOLOR=ORANGE, STYLE=ITALIC,
   WHEN=LINEPRICE GT 100000, $
4.TYPE=DATA, BACKCOLOR=SILVER, FONT='Arial', $
   TYPE=HEADING, FONT='Arial', STYLE=BOLD, SIZE=11, $
   ENDSTYLE

   END
```

このフォーマット設定の要点は次のとおりです。

1. 1 つ目の条件付き宣言で、注文合計が 500,000 より大きい行のフォーマットを設定します。
2. 2 つ目の条件付き宣言で、注文合計が 400,000 より大きく 500,000 以下の行のフォーマットを設定します。ここで「500,000以下」という条件を設定しているのは、注文合計が 500,000 より大きい行が 1 つ目の条件付き宣言で、すでにフォーマット設定されているためです。
3. 3 つ目の条件付き宣言で、注文合計が 100,000 より大きく 400,000 以下の行のフォーマットを設定します。ここで「400,000 以下」という条件を設定しているのは、注文合計が 400,000 より大きい行が最初の 2 つの条件付き宣言ですでにフォーマット設定されているためです。
4. 条件付き宣言文の後に続く条件なしの宣言では、次のフォーマットが設定されます。
   - 背景色を指定します。背景色は条件付き宣言でも指定されています。ここでは、条件付き宣言でフォーマットがまだ設定されていない注文合計が 100,000 以下の行に背景色 (銀色) を適用します。
   - フォントを指定します。フォントは条件付き宣言では指定されていません。ここでは、すべてのデータ行にフォント (Arial) を適用します。

出力結果は次のとおりです。



Order Revenue

Order Number:	Date Of Order:	Order Total:
94710	2001/01/02	$406,964.24
94680	2001/01/02	$421,916.60
94670	*2001/01/02*	*$513,868.76*
94550	2001/01/02	$496,323.64
94530	2001/01/02	$3,472.41
94520	*2001/01/02*	*$261,808.72*
94490	*2000/12/31*	*$633,723.06*
94460	2001/01/02	$3,872.39
94430	2001/01/02	$3,033.38
94410	2001/01/02	$2,337.28

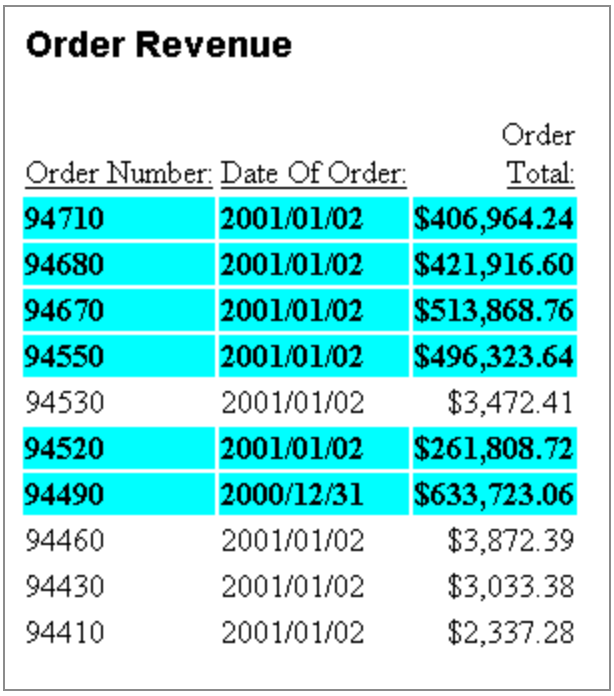
基本的な条件付きフォーマット設定の適用

次の例は、条件付きフォーマット設定をレポートに適用する方法について示しています。この条件付きフォーマット設定では、合計が 200,000 を超える注文を強調して表示します。

ここでは、構文内で特定のフィールドが指定されていないことから、すべての行にこのフォーマット設定が適用されます。

```
TABLE FILE CENTORD
HEADING
"Order Revenue"
" "
SUM ORDER_DATE LINEPRICE AS 'Order,Total:'
BY HIGHEST 10 ORDER_NUM
ON TABLE SET PAGE-NUM OFF

ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=DATA, BACKCOLOR=AQUA, STYLE=BOLD, WHEN=LINEPRICE GT 200000, $
TYPE=HEADING, FONT='Arial', STYLE=BOLD, SIZE=11, $
ENDSTYLE

END
```

出力結果は次のとおりです。



Order Revenue

Order Number:	Date Of Order:	Order Total:
94710	2001/01/02	$406,964.24
94680	2001/01/02	$421,916.60
94670	2001/01/02	$513,868.76
94550	2001/01/02	$496,323.64
94530	2001/01/02	$3,472.41
94520	2001/01/02	$261,808.72
94490	2000/12/31	$633,723.06
94460	2001/01/02	$3,872.39
94430	2001/01/02	$3,033.38
94410	2001/01/02	$2,337.28

フィールドの条件付きフォーマット設定

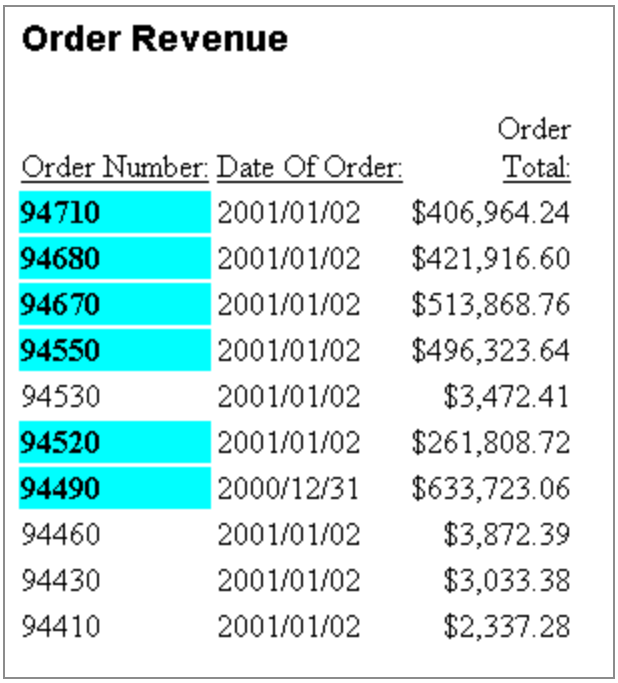
次の例は、特定の条件が指定されていないフィールドに対して、条件付きフォーマット設定を適用し、そのフィールドを強調表示する方法を示しています。WHEN 条件の指定により、合計が 200,000 を超える注文は、太字および水色の背景色で表示されます。

ここでは、WHEN 条件 (LINEPRICE) で評価するフィールドとフォーマット設定するフィールド (ORDER_NUM) は一致しません。これらのフィールドを一致させる必要はありません。

```
TABLE FILE CENTORD
HEADING
"Order Revenue"
" "
SUM ORDER_DATE LINEPRICE AS 'Order,Total:'
BY HIGHEST 10 ORDER_NUM
ON TABLE SET PAGE-NUM OFF

ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=DATA, COLUMN=ORDER_NUM,
   BACKCOLOR=AQUA, STYLE=BOLD, WHEN=LINEPRICE GT 200000, $
TYPE=HEADING, FONT='Arial', STYLE=BOLD, SIZE=11, $
ENDSTYLE
```

```
END
```

出力結果は次のとおりです。



Order Revenue

Order Number:	Date Of Order:	Order Total:
94710	2001/01/02	$406,964.24
94680	2001/01/02	$421,916.60
94670	2001/01/02	$513,868.76
94550	2001/01/02	$496,323.64
94530	2001/01/02	$3,472.41
94520	2001/01/02	$261,808.72
94490	2000/12/31	$633,723.06
94460	2001/01/02	$3,872.39
94430	2001/01/02	$3,033.38
94410	2001/01/02	$2,337.28

ACROSS 値に基づく条件付きスタイル設定

次の例は、ACROSS フィールド名または ACROSS 列要素 (A1、A2) のいずれかを使用して ACROSS 値を参照する方法を示しています。

この例では、条件付きスタイル設定に ACROSS 値を使用して、Coffee カテゴリのすべての ACROSS 列に異なる背景色を設定し、Espresso カテゴリの ACROSS 列に新しいフォントスタイルを設定しています。

```
SET ACROSSTITLE=SIDE
TABLE FILE GGSALES
SUM DOLLARS/I8M AS ''
BY REGION
BY ST
BY CITY
ACROSS CATEGORY
ACROSS PRODUCT
WHERE CATEGORY EQ 'Coffee' OR 'Food';
```

```
ON TABLE SET PAGE-NUM NOPAGE
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
SQUEEZE=ON,UNITS=IN,ORIENTATION=PORTRAIT,$
TYPE=REPORT,FONT='ARIAL',SIZE=10,BORDER=LIGHT,$
TYPE=ACROSSTITLE,COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=ACROSSVALUE,COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=TITLE,COLOR=WHITE, BACKCOLOR=GREY,$
TYPE=DATA, ACROSSCOLUMN=DOLLARS, BACKCOLOR=THISTLE, WHEN=CATEGORY EQ
'Coffee',$
TYPE=DATA, ACROSSCOLUMN=DOLLARS, STYLE=BOLD+ITALIC, WHEN=A2 EQ 'Espresso', $
ENDSTYLE
END
```

出力結果は次のとおりです。

		Category	Coffee			Food		
		Product	Capuccino	Espresso	Latte	Biscotti	Croissant	Scone
Region	State	City						
Midwest	IL	Chicago	.	***$420,439***	$978,340	$378,412	$549,366	$595,069
	MO	St. Louis	.	***$419,143***	$966,981	$368,077	$613,871	$481,953
	TX	Houston	.	***$455,365***	$938,245	$345,238	$587,887	$418,398
Northeast	CT	New Haven	$158,995	***$279,373***	$926,052	$589,355	$551,489	$283,874
	MA	Boston	$174,344	***$248,356***	$917,737	$570,391	$497,234	$332,486
	NY	New York	$208,756	***$322,378***	$928,026	$642,259	$622,095	$290,811
Southeast	FL	Orlando	$317,027	***$256,539***	$889,887	$511,597	$602,076	$311,836
	GA	Atlanta	$352,161	***$317,389***	$907,365	$555,231	$661,806	$273,420
	TN	Memphis	$274,812	***$279,644***	$820,584	$438,889	$638,477	$315,399
West	CA	Los Angeles	$306,468	***$267,809***	$809,647	$266,030	$800,084	$315,584
		San Francisco	$279,830	***$338,270***	$935,862	$269,518	$824,457	$292,839
	WA	Seattle	$309,197	***$301,538***	$924,896	$328,320	$801,060	$304,445

ピアグラフの条件付きフォーマット設定

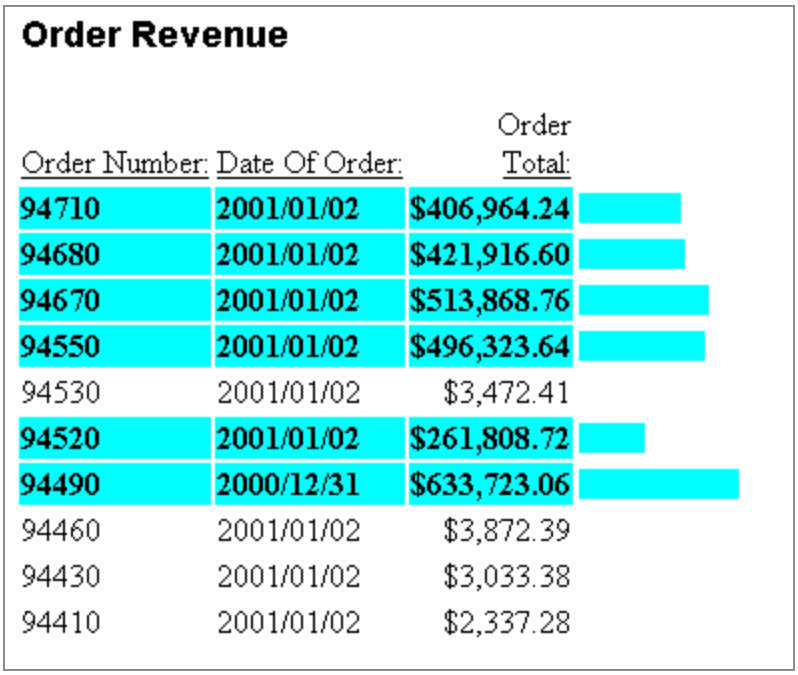
次の例は、ピアグラフに条件付きフォーマットを設定する方法を示しています。このリクエストは、ピアグラフを組み込んで、LINEPRICE フィールドのデータを図で表現します。条件付きフォーマット設定を使用して、合計が 200,000 を超える注文を強調表示します。データフィールド (TYPE=DATA) およびピアグラフ (GRAPHTYPE=DATA) の両方に条件付きのフォーマット設定を適用します。

なお、ピアグラフは HTML レポートのみでサポートされます。

```
TABLE FILE CENTORD
HEADING
"Order Revenue"
```

```
" "
SUM ORDER_DATE LINEPRICE AS 'Order,Total:'
BY HIGHEST 10 ORDER_NUM
ON TABLE SET PAGE-NUM OFF

ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=DATA, BACKCOLOR=AQUA, STYLE=BOLD, WHEN=LINEPRICE GT 200000, $
GRAPHTYPE=DATA, COLUMN=LINEPRICE, $
GRAPHTYPE=DATA, GRAPHCOLOR=AQUA, WHEN=LINEPRICE GT 200000,$
TYPE=HEADING, FONT='Arial', STYLE=BOLD, SIZE=11, $
ENDSTYLE

END
```

出力結果は次のとおりです。



Order Revenue

Order Number:	Date Of Order:	Order Total:
94710	2001/01/02	$406,964.24
94680	2001/01/02	$421,916.60
94670	2001/01/02	$513,868.76
94550	2001/01/02	$496,323.64
94530	2001/01/02	$3,472.41
94520	2001/01/02	$261,808.72
94490	2000/12/31	$633,723.06
94460	2001/01/02	$3,872.39
94430	2001/01/02	$3,033.38
94410	2001/01/02	$2,337.28

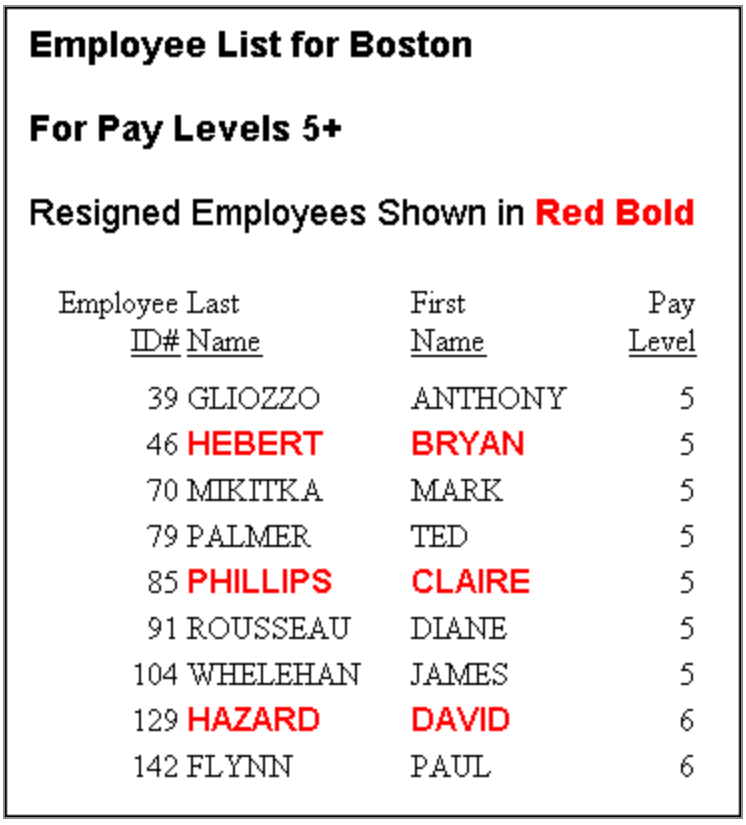
非表示 (NOPRINT) のフィールド値を基準にした条件付きフォーマット設定の適用

次の例は、非表示 (NOPRINT) のフィールド値を基準にして条件付きフォーマットを設定する方法を示しています。このレポートでは、退職した従業員を強調して表示する条件付きフォーマット設定を使用します。

ここでは、WHEN 属性の条件が、レポートには表示されないフィールド (STATUS) を評価します。条件で評価するフィールドはリクエストで指定されていなければなりませんが、次のリクエストで示すように、NOPRINT オプションを使用してそのフィールドをレポート上で非表示にすることができます。

```
TABLE FILE CENTHR
HEADING
"Employee List for Boston"
" "
"For Pay Levels 5+"
" "
"Resigned Employees Shown in <0>Red Bold"
" "
PRINT LNAME FNAME PAYSCALE STATUS NOPRINT
BY ID_NUM
WHERE PLANT EQ 'BOS' AND PAYSCALE GE 5
ON TABLE SET PAGE-NUM OFF

ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=DATA, COLUMN=LNAME,
   COLOR=RED, FONT='Arial', STYLE=BOLD, WHEN=STATUS EQ 'RESIGNED', $
TYPE=DATA, COLUMN=FNAME,
   COLOR=RED, FONT='Arial', STYLE=BOLD, WHEN=STATUS EQ 'RESIGNED', $
TYPE=HEADING, FONT='Arial', STYLE=BOLD, SIZE=11, $
TYPE=HEADING, LINE=5, STYLE=-BOLD, $
TYPE=HEADING, LINE=5, ITEM=2, STYLE=BOLD, COLOR=RED, $
ENDSTYLE

END
```

出力結果は次のとおりです。



Employee List for Boston

For Pay Levels 5+

Resigned Employees Shown in **Red Bold**

Employee ID#	Last Name	First Name	Pay Level
39	GLIOZZO	ANTHONY	5
46	**HEBERT**	**BRYAN**	5
70	MIKITKA	MARK	5
79	PALMER	TED	5
85	**PHILLIPS**	**CLAIRE**	5
91	ROUSSEAU	DIANE	5
104	WHELEHAN	JAMES	5
129	**HAZARD**	**DAVID**	6
142	FLYNN	PAUL	6

ソートグループへの条件付きフォーマット設定の適用

次の例は、条件付きフォーマット設定をソートグループに適用する方法を示しています。このレポートでは、退職した従業員を強調して表示する条件付きフォーマット設定を使用します。

1 つの条件付き宣言で、すべてのソートグループ行にフォーマット設定を適用することができます。これを行うには、WHEN 属性の条件でソートフィールド (STATUS) の値を評価します。

```
TABLE FILE CENTHR
HEADING
"Employee List for Boston"
" "
"For Pay Levels 5+"
" "
PRINT LNAME FNAME PAYSCALE
```

```
BY STATUS SKIP-LINE
WHERE PLANT EQ 'BOS' AND PAYSCALE GE 5
ON TABLE SET PAGE-NUM OFF

ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=DATA,
   COLOR=RED, FONT='Arial', STYLE=BOLD, WHEN=STATUS EQ 'RESIGNED',$
TYPE=HEADING, FONT='Arial', STYLE=BOLD, SIZE=11, $
ENDSTYLE

END
```

出力結果は次のとおりです。

Employee List for Boston

For Pay Levels 5+

Current Status	Last Name	First Name	Pay Level
DECLINED	ROUSSEAU	DIANE	5
EMPLOYED	MIKITKA	MARK	5
	WHELEHAN	JAMES	5
	FLYNN	PAUL	6
RESIGNED	HEBERT	BRYAN	5
	PHILLIPS	CLAIRE	5
	HAZARD	DAVID	6
TERMINAT	GLIOZZO	ANTHONY	5
	PALMER	TED	5

1 つの条件付きフォーマット設定をすべてのフィールドに適用するのではなく、2 つのフィールドに限定して適用する場合、このリクエストの例では 2 つの宣言を使用して、各宣言がそれぞれ異なるフィールド (LNAME と FNAME) を指定するようにします。

```
TABLE FILE CENTHR
HEADING
"Employee List for Boston"
" "
"Pay Levels 5+"
```

```
" "
PRINT LNAME FNAME PAYSCALE
BY STATUS SKIP-LINE
WHERE PLANT EQ 'BOS' AND PAYSCALE GE 5
ON TABLE SET PAGE-NUM OFF

ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=DATA, COLUMN=LNAME,
   COLOR=RED, FONT='Arial', STYLE=BOLD, WHEN=STATUS EQ 'RESIGNED', $
TYPE=DATA, COLUMN=FNAME,
   COLOR=RED, FONT='Arial', STYLE=BOLD, WHEN=STATUS EQ 'RESIGNED', $
TYPE=HEADING, FONT='Arial', STYLE=BOLD, SIZE=11, $
ENDSTYLE

END
```

出力結果は次のとおりです。

Employee List for Boston

For Pay Levels 5+

Current Status	Last Name	First Name	Pay Level
DECLINED	ROUSSEAU	DIANE	5
EMPLOYED	MIKITKA	MARK	5
	WHELEHAN	JAMES	5
	FLYNN	PAUL	6
RESIGNED	HEBERT	BRYAN	5
	PHILLIPS	CLAIRE	5
	HAZARD	DAVID	6
TERMINAT	GLIOZZO	ANTHONY	5
	PALMER	TED	5

予測値への条件付きフォーマット設定の適用

次の例は、条件付きフォーマット設定をレポートの予測値に適用する方法を示しています。

```
DEFINE FILE GGSALES
SDATE/YYM = DATE;
SYEAR/Y = SDATE;
SMONTH/M = SDATE;
PERIOD/I2 = SMONTH;
END

TABLE FILE GGSALES
SUM UNITS DOLLARS
BY CATEGORY BY PERIOD
WHERE SYEAR EQ 97 AND CATEGORY EQ 'Coffee'
ON PERIOD RECAP MOVAVE/D10.1= FORECAST(DOLLARS,1,3,'MOVAVE',3);
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=REPORT, BACKCOLOR=SILVER, WHEN=FORECAST, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Category	PERIOD	Unit Sales	Dollar Sales	MOVAVE
Coffee	1	61666	801123	801,123.0
	2	54870	682340	741,731.5
	3	61608	765078	749,513.7
	4	57050	691274	712,897.3
	5	59229	720444	725,598.7
	6	58466	742457	718,058.3
	7	60771	747253	736,718.0
	8	54633	655896	715,202.0
	9	57829	730327	711,158.7
	10	57012	724412	703,545.0
	11	51110	620264	691,667.7
	12	58981	762328	702,334.7
	13	0	0	694,975.6
	14	0	0	719,879.4
	15	0	0	705,729.9

集計行、下線、ブランク行、改ページの追加

リクエストに ON 句を使用することで、ページ番号を振り直せるだけではなく、ソートオプションを追加することもできます。このオプションには、中間合計およびその他の集計行、ソート項目を指定している見出しおよび脚注、下線、ブランク行、改ページなどがあります。ON 句では、ソートフィールドまたは表示フィールドの値が変わるたびに開始されるオプションを指定します。ソートオプション (集計行、下線、ブランク行、改ページ) は、各ソートグループに適用されます。

リクエストを柔軟にするため、他の状況では必要のないオプションを追加することができます。この場合、さらにユーザの選択により、各リクエストの実行オプションを制御します。

デフォルト設定では、ON 句で参照されるフィールドがリクエストに存在しない場合や、そのオプションが指定したフィールドでサポートされていない場合は、次のメッセージが生成されて処理が停止します。

```
(FOC013) ON フィールドが BY で指定されていません：sortfield
```

SET ONFIELD=IGNORE コマンドを使用して、存在しないフィールドや指定したオプションでサポートされていないフィールドを参照している ON 句を WebFOCUS が無視するよう指定することができます。

使用されるフィールドは、すべてデータソースのマスターファイルに存在する必要があり、存在しない場合は、処理が停止します。

```
(FOC003) THE FIELDNAME IS NOT RECOGNIZED: field
```

集計行、下線、ブランク行、改ページの表示

```
{BY|ON} sortfieldoption [[AND] option ...]
```

説明

BY|ON

この 2 つは構文上の相違はありますが、機能的には同一です。BY を指定した場合は、ソート句の一部としてソートオプションを指定することができます。ON を指定した場合は、ソート句とは別にソートオプションを指定することができます。

sortfield

BY フィールド名です。

option

ソートオプションには、PAGE-BREAK、PAGE-BREAK REPAGE、REPAGE、RECAP、RECOMPUTE、SKIP-LINE、SUBFOOT、SUBHEAD、SUBTOTAL、SUB-TOTAL、SUMMARIZE、UNDER-LINE があります。

AND

読みやすさを向上するために、2 つのソートフィールドオプション間に追加することができます。

ON 句処理の制御

```
SET ONFIELD = {ALL|IGNORE}
```

```
ON TABLE SET ONFIELD {ALL|IGNORE}
```

説明

ALL

ON 句で参照されているフィールドがリクエストに存在しない場合、メッセージが発行されて処理が停止します。デフォルト値は ALL です。

IGNORE

リクエストに存在しないフィールドを参照している ON 句、および指定したフィールドタイプでサポートされていないオプションが含まれる ON 句を無視します。

存在しないフィールドの ON 句の無視

次のリクエストは、EMPDATA データソースが使用され、DEPT、DIV、PIN フィールドの ON 句が含まれています。PIN はソートフィールドですが、その他のソートフィールドは、実行時に変数を「&F1」として入力する必要があります。

```
SET USER = EUSER
 TABLE FILE EMPDATA
   SUM SALARY
   BY &F1
   BY PIN
      ON DEPT SKIP-LINE NOSPLIT
      ON &F1  SUBTOTAL
      ON DIV   PAGE-BREAK
      ON TABLE SET ONFIELD ALL
 END
```

変数「&F1」に値「DEPT」を入力してリクエストを実行します。次のメッセージが生成されます。

```
 エラーのある行       8  プロシジャ名  IGNORE3 *
(FOC013) ON フィールドが BY で指定されていません :DIV
 コマンドの終わりまで処理をバイパスします
(FOC009) リクエストが完結していません。
```

さらに、ONFIELD パラメータの値を IGNORE に変更し、変数「&F1」に値「DEPT」を入力してリクエストを再度実行します。以下は、出力結果の一部を示しています。

```
DEPT                 PIN                 SALARY
----                 ---                 ------

ACCOUNTING           000000070       $83,000.00
                     000000100       $32,400.00
                     000000300       $79,000.00
                     000000370       $62,500.00
                     000000400       $26,400.00

*TOTAL ACCOUNTING                   $283,300.00


ADMIN SERVICES       000000170       $30,800.00
                     000000180       $25,400.00
```

```
*TOTAL ADMIN SERVICES                $56,200.00
```

集計行、下線、ブランク行、改ページの条件付き追加

リクエストに WHEN 句を使用すると、条件付きでページ番号を振り直せるだけでなく、ソートオプションを条件付きで追加することもできます。このオプションには、中間合計およびその他の集計行、ソート項目を指定している見出しおよび脚注、下線、ブランク行、改ページなどがあります。WHEN 句を使用して、BY フィールド値を評価する条件を指定します。ソートオプション (集計行、下線、ブランク行、改ページ) は、条件を満たすすべてのソートグループに適用されますが、条件を満たさないソートグループでは無視されます。

WHEN 句は、ON ソートフィールド句および BY ソートフィールド句の一部として指定します。WHEN 句は、それぞれのソートフィールド句に対して個別に指定することができます。以下はその例です。

```
ON ORDER_NUM UNDER-LINE WHEN QUANTITY GT 5
ON COUNTRY PAGE-BREAK WHEN LINEPRICE GT 200000
```

複数のソートオプションが ソートフィールド句に指定されている場合は、各オプションに対して異なる WHEN 句を指定することができます。以下はその例です。

```
ON ORDER_NUM SKIP-LINE WHEN QUANTITY GT 5; UNDER-LINE WHEN QUANTITY GT 10
```

集計行、下線、ブランク行、改ページの条件付き表示

```
{BY|ON} sortfield [option WHEN condition [;] [AND]]...
```

説明

BY|ON

この 2 つは構文上の相違はありますが、機能的には同一です。BY を指定した場合は、ソート句の一部としてソートオプションを指定することができます。ON を指定した場合は、ソート句とは別にソートオプションを指定することができます。

sortfield

BY フィールド名です。

option

ソートオプションには、PAGE-BREAK、PAGE-BREAK REPAGE、REPAGE、RECAP、RECOMPUTE、SKIP-LINE、SUBFOOT、SUBHEAD、SUBTOTAL、SUB-TOTAL、SUMMARIZE、UNDER-LINE があります。

SUBHEAD または SUBFOOT を指定する場合は、見出しまたは脚注テキストの次の行に WHEN 句を記述する必要があります。

condition

論理式です。詳細は、式の使用を参照してください。

文字列や日付などの数値以外の定数は、一重引用符 (') で囲む必要があります。

条件が数値詳細フィールドを評価する場合は、詳細値が個別に評価されるのではなく、ソートグループの詳細フィールド値の合計が評価されます。たとえば、次のようなリクエストを想定します。

```
TABLE FILE CENTHR
PRINT ID_NUM SALARY
BY PLANT
ON PLANT UNDER-LINE
WHEN SALARY GT 2000000
END
```

この条件では、PLANT の値ごとに SALARY 値の合計が評価されます。

ソートグループに複数回出現する文字フィールドを評価する場合は、それぞれのソートグループのフィールドの最後の値が評価されます。

レポートに演算子とフィールドが使用されていない場合でも、演算接頭語を条件内のフィールドに適用することができます (例、WHEN AVE.PRICE GT 300)。集計は、ソートフィールドのそれぞれの値に対して実行されます。

BY または ON 句に複数のオプションが指定されている場合は、WHEN 条件はその直後にあるオプションのみに適用されます。

;

BY または ON 句の複数のオプションに対して WHEN 句が指定されている場合に必要です。その他の状況ではオプションとして使用したり、読みやすさを向上するために使用したりします。

AND

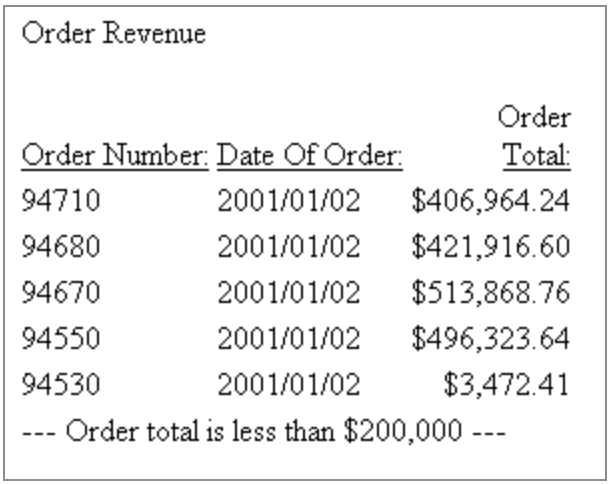
読みやすさを向上するために、2 つのソートフィールドオプション間に追加することができます。

ソートオプションでの WHEN 条件の使用

次の例は、レポートに中間脚注を条件付きで表示する方法を示しています。このレポートでは、条件付きのソート項目を指定している脚注を使用して、合計が 200,000 未満の注文を強調して表示します。

```
TABLE FILE CENTORD
HEADING
"Order Revenue"
" "
SUM ORDER_DATE LINEPRICE AS 'Order,Total:'
BY HIGHEST 5 ORDER_NUM

ON ORDER_NUM
    SUBFOOT
       "--- Order total is less than $200,000 ---"
       " "
       WHEN LINEPRICE LT 200000

ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。



Order Revenue

Order Number:	Date Of Order:	Order Total:
94710	2001/01/02	$406,964.24
94680	2001/01/02	$421,916.60
94670	2001/01/02	$513,868.76
94550	2001/01/02	$496,323.64
94530	2001/01/02	$3,472.41

--- Order total is less than $200,000 ---

複数ソートオプションでの WHEN 条件の使用

次の例は、レポートコンポーネントに複数の条件を適用する方法を示しています。このレポートでは、条件付きのソート項目を指定している脚注を使用して、合計が 200,000 を超える注文と 200,000 以下の注文を区別します。

ここでは、1 つのソート句 (ON ORDER_NUM) が、複数のソートオプション (2 つの異なる SUBFOOT 句) を指定し、それぞれのオプションに個別の WHEN 句が指定されています。

```
TABLE FILE CENTORD
HEADING
"Order Revenue"
" "
SUM ORDER_DATE LINEPRICE AS 'Order,Total:'
BY HIGHEST 5 ORDER_NUM
ON ORDER_NUM
   SUBFOOT
      "--- Order total is less than $200,000 ---"
      " "
      WHEN LINEPRICE LT 200000;
   SUBFOOT
      "+++ Order total is greater than or equal to $200,000 +++"
      " "
      WHEN LINEPRICE GE 200000;

ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Order Revenue

Order Number:	Date Of Order:	Order Total:
94710	2001/01/02	$406,964.24
+++ Order total is greater than or equal to $200,000 +++		
94680	2001/01/02	$421,916.60
+++ Order total is greater than or equal to $200,000 +++		
94670	2001/01/02	$513,868.76
+++ Order total is greater than or equal to $200,000 +++		
94550	2001/01/02	$496,323.64
+++ Order total is greater than or equal to $200,000 +++		
94530	2001/01/02	$3,472.41
--- Order total is less than $200,000 ---		

空レポートの表示制御

リクエストから返されるレコードがない場合、たとえば選択条件を満たすレコードがない場合、またはデータソースにレコードが存在しない場合は、レポートに表示する内容として次のものを選択することができます。

- **空レポート**　データは存在しないが、フィールドタイトル、レポートの見出し (リクエストで指定されている場合)、ページの見出し (指定されている場合) が表示されます。

  空レポートを表示するには、EMPTYREPORT パラメータを ON に設定します。

- **メッセージ**　レポートの出力結果がないことを知らせます。

  これがデフォルト値です。EMPTYREPORT パラメータを OFF に設定すると、このデフォルトに戻ります。

この設定は、表形式レポートに適用されますが、自由形式レポートおよびグラフには適用されません。

SET EMPTYREPORT = OFF は、DOC フォーマットではサポートされません。

WHERE TOTAL ステートメントを使用したレポートを作成し、そのテストでレコード件数が 0 (ゼロ) の場合は、空レポートが表示されます。

空レポートの表示制御

リクエストで返されるレコードがない場合、EMPTYREPORT パラメータを使用してレポートに表示する内容を制御することができます。SET コマンドを次の方法で発行します。

リクエスト外　次の構文を使用します。

```
SET EMPTYREPORT = {ANSI|ON|OFF}
```

リクエスト内　次の構文を使用します。

```
ON TABLE SET EMPTYREPORT {ANSI|ON|OFF}
```

説明

ANSI

COUNT がリクエストされている場合は、単一行レポートを生成し、ミッシングデータ文字またはゼロを表示します。いずれの場合でも、&RECORDS が 0 になり、&LINES は 1 になります。

SQL トランスレータを呼び出す場合は、EMPTYREPORT のデフォルト設定が OFF から ANSI に自動的に置換されます。

ON

データは存在しないが、フィールドタイトル、レポートの見出し (リクエストで指定されている場合)、ページの見出し (指定されている場合) がレポートに表示されます。

OFF

メッセージが表示されて、レポートの出力結果がないことを知らせます。デフォルト値は OFF です。

空レポートの表示制御

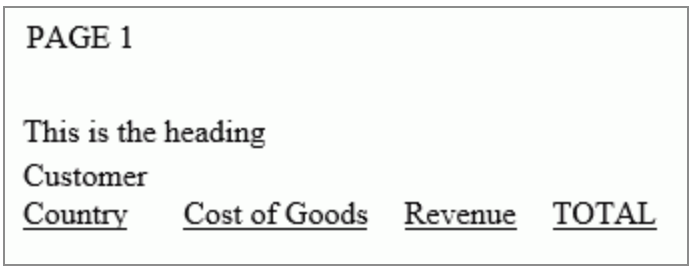
次のリクエストではレコードは取得されず、EMPTYREPORT パラメータは OFF に設定されています。

```
SET EMPTYREPORT=OFF
TABLE FILE WF_RETAIL_LITE
HEADING
"This is the heading"
SUM COGS_US REVENUE_US COLUMN-TOTAL ROW-TOTAL
BY COUNTRY_NAME
WHERE COUNTRY_NAME EQ 'Louisiana'
FOOTING
"This is the footing"
ON TABLE SET STYLE *
GRID=OFF,$
END
```

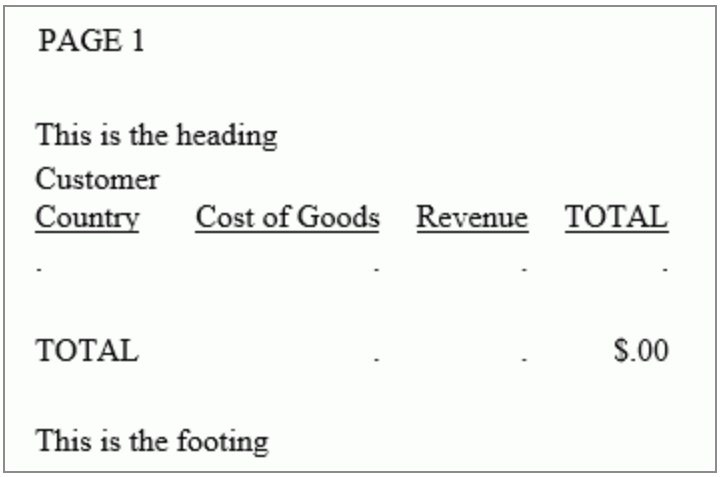
次の出力が生成されます。

```
0 NUMBER OF RECORDS IN TABLE=        0  LINES=      0
```

EMPTYREPORT を ON に変更した場合の出力結果は次のとおりです。



EMPTYREPORT を ANSI に変更した場合の出力結果は次のとおりです。


PAGE 1

This is the heading

Customer Country	Cost of Goods	Revenue	TOTAL
.	.	.	.
TOTAL	.	.	$.00

This is the footing

スタイルシートデフォルト値のみによるレポートフォーマット設定

デフォルトのスタイルシート値のみを使用して、レポートのフォーマットを設定することができます。この方法では、特定のフォーマット設定を指定したり、スタイルシートを利用したりすることはできません。

HTML や PDF などの表示フォーマットには、それぞれ独自のデフォルト値が設定されています。たとえば、HTML のデフォルトはプロポーショナルフォント、PDF のデフォルトは等幅フォントです。特定のスタイルシート属性のデフォルト値についての詳細は、対応する属性のマニュアルを参照してください。

スタイルシートデフォルト値のみによるレポートフォーマット設定

デフォルトのスタイルシート値のみを使用してフォーマットを設定するには、次の方法を使用します。

- **プロシジャ内のすべてのレポートを対象**　プロシジャの先頭で次のコマンドを発行します。

```
SET STYLE[SHEET] = ON
```

- **1 つのリクエストを対象**　リクエストで次のコマンドを発行します。

```
ON TABLE SET STYLE[SHEET] ON
```

説明

SHEET

この部分を省略してコマンドの長さを短くすることができます。コマンドの動作には影響しません。

WebFOCUS スタイルシートを使用したレポートコンポーネントの選択

レポートは、さまざまなタイプのコンポーネントで構成されており、スタイルシートでこれらのコンポーネントを個別に選択してフォーマットを設定することができます。

レポートの各コンポーネントには、スタイルシートが継承されます。詳細は、WebFOCUS スタイルシートの作成と管理を参照してください。

グラフ内でコンポーネントを選択する方法についての詳細は、グラフの作成を参照してください。

特に明記していない限り、この章で言及するスタイルシートは、WebFOCUS スタイルシートのことを指します。

レポート全体、列全体、行全体の選択

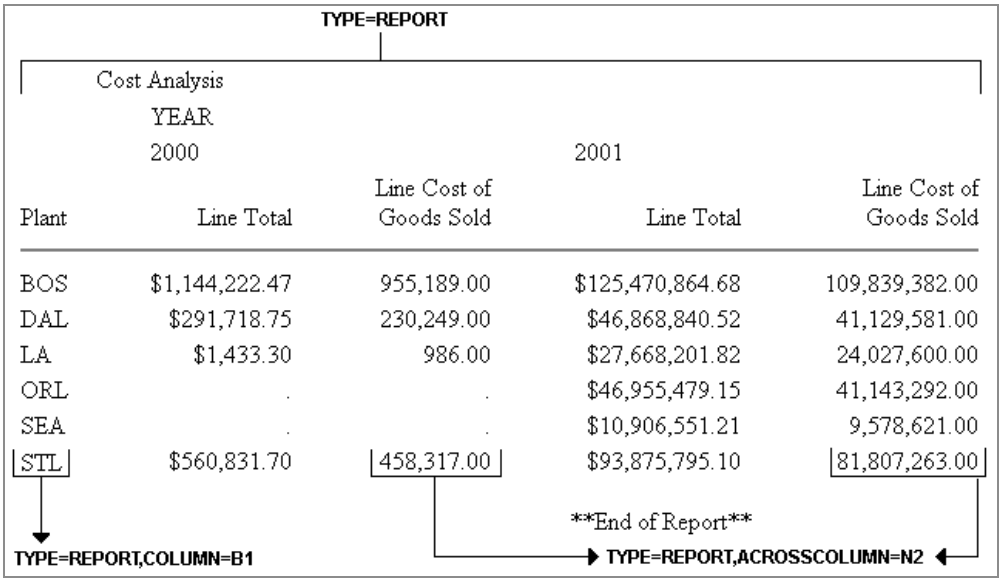
次のコンポーネントに対してフォーマットを設定することができます。

- **レポート全体**詳細は、レポート全体、列全体、行全体の選択を参照してください。
- **列全体**　レポート内の列全体のフォーマットを設定します。タイトルとデータで構成されます (ROW-TOTAL 列を含む)。詳細は、レポート全体、列全体、行全体の選択を参照してください。
- **行全体**　レポート内の行全体のフォーマットを設定します。行のラベルテキストとデータで構成されます。これらの行には、FML 行、合計行、中間合計行が含まれます。詳細は、レポート全体、列全体、行全体の選択を参照してください。

  ACROSS フィールド全体のタイトル行または値行に表示されるデータは 1 種類しかありませんが、これらの行をスタイルシートで選択することもできます。詳細は、見出し、脚注、タイトル、FML テキストの選択を参照してください。

次の例は、REPORT コンポーネントおよび COLUMN 属性と ACROSSCOLUMN 属性がレポート内に表示される位置、およびこれらのコンポーネントの識別に使用する TYPE の値を示しています。この例では、COLUMN の値は B1、ACROSSCOLUMN の値は N2 ですが、それ以外の値を使用してコンポーネントを選択することもできます。

```
TABLE FILE CENTORD
SUM LINEPRICE LINE_COGS AS 'Line Cost of,Goods Sold'
BY PLANT AS 'Plant'
ACROSS YEAR
WHERE YEAR EQ 2000 or 2001
HEADING
"Cost Analysis"
FOOTING CENTER
"**End of Report**"
ON TABLE SET PAGE-NUM OFF
END
```



注意：このリクエストは、レポート内の各コンポーネントの表示位置を単に示すことが目的であるため、ここではスタイルシートは省略されています。

レポート全体の選択

スタイルシートでレポート全体を選択するには、次の属性と値を使用します。

```
TYPE=REPORT
```

レポート全体の選択

次の例は、レポート全体を選択してフォーマットを設定する方法を示しています。関連するスタイルシートは、リクエスト内で太字で表示されています。

```
TABLE FILE CENTINV
HEADING
"Excess Stock Report"
SUM QTY_IN_STOCK
BY PRODNAME
WHERE QTY_IN_STOCK GT 10000
FOOTING CENTER
"**End of Report**"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
TYPE=REPORT, STYLE=BOLD,$
ENDSTYLE
END
```

出力結果は次のとおりです。

Excess Stock Report

Product Name:	Quantity In Stock:
2 Hd VCR LCD Menu	43068
250 8MM Camcorder 40 X	60073
330DX Digital Camera 1024K P	12707
750SL Digital Camcorder 300 X	10758
AR2 35MM Camera 8 X	11499
AR3 35MM Camera 10 X	12444
Combo Player - 4 Hd VCR + DVD	13527
QX Portable CD Player	22000
ZC Digital PDA - Standard	33000
ZT Digital PDA - Commercial	21000

End of Report

列全体の選択

```
TYPE=REPORT, coltype=column
```

説明

coltype

列のタイプを指定します。次のいずれかの値を使用することができます。

- **COLUMN** 指定する値には、ソート項目 (BY で生成)、表示項目 (PRINT、LIST、SUM、COUNT で生成)、一時項目 (COMPUTE で生成)、行合計の項目 (ROW-TOTAL で生成) があります。
- **ACROSSCOLUMN** ACROSS 行に繰り返し表示する列のインスタンスすべてを指定します。この指定により、フィールドタイトルの上に表示される ACROSS フィールド値にもフォーマット設定が適用されます。

column

1 つまたは複数の列を指定します。ACROSSCOLUMN を選択する場合、有効な識別子は Nn と Pn のみです。これらの識別子は、表示フィールドではなく、ACROSS フィールドの個数のみを集計します。

下表は、スタイルシートでフィールドの選択に使用できる識別子の一覧です。

識別子	説明
Nn	レポート内の位置に基づいて列を選択します。n の値を特定するには、BY フィールド、表示フィールド、ROW-TOTAL フィールドの数を左から右に数えます。この場合、NOPRINT フィールドもその数に含められます。ACROSSCOLUMN の場合、ACROSS の個数のみが集計されます。
Pn	レポート内の位置に基づいて列を選択します。n の値を特定するには、BY フィールド、表示フィールド、ROW-TOTAL フィールドの数を左から右に数えます。ただし、NOPRINT フィールドはその数に含められません。ACROSSCOLUMN の場合、ACROSS の個数のみが集計されます。

識別子	説明
`Cn`	レポート内の位置に基づいて表示する列を選択します。n の値を特定するには、表示フィールドのみの数を左から右に数えます。この場合、NOPRINT フィールドもその数に含められます。ただし、BY フィールドおよび ROW-TOTAL フィールドはこの数に含められません。 すべての表示フィールドを選択するには C* を使用します。
`Bn`	レポート内の位置に基づいて BY フィールドの列を選択します。n の値を特定するには、BY フィールドのみの数を左から右に数えます。この場合、NOPRINT フィールドもその数に含められます。 すべての BY フィールドを選択するには B* を使用します。
`field`	フィールド名に基づいて列を選択します。 1 つのフィールドが複数回出現する場合は、field(n) を使用してそのフィールドの特定の出現回数を選択するか、field(*) を使用してすべての出現回数を選択します。
`ROWTOTAL`	ROW-TOTAL で生成された行合計のフィールドを選択します。ACROSS とともに複数の表示コマンドを使用した場合、ROWTOTAL は複数の合計フィールドを生成します。ROWTOTAL(n) を使用して、特定の合計フィールドを選択します。ROWTOTAL(field) を使用して、特定のフィールドに対する行合計のフィールドを選択します。 ROWTOTAL(*) を使用して、レポート内の行合計のフィールドをすべて選択します。

注意：1 つのスタイルシート内では、すべてのフィールドが同様に指定されなければなりません。

列全体の選択

次の例は、レポート内のフィールドタイトルと列データを含む列全体を選択する方法を示しています。リクエストの太字の部分が、この例に関連するスタイルシートです。

注意：スタイルシートの COLUMN 属性に対して P1 値、B1 値、フィールド名 (PRODNAME) のいずれかを代用して、同一の結果を得ることができます。

```
TABLE FILE CENTINV
HEADING
"Excess Stock Report"
SUM QTY_IN_STOCK
BY PRODNAME
WHERE QTY_IN_STOCK GT 10000
FOOTING CENTER
"**End of Report**"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
TYPE=REPORT, COLUMN=N1, STYLE=ITALIC,$
ENDSTYLE
END
```

出力結果は次のとおりです。

```
Excess Stock Report

Product                                Quantity
Name:                                  In Stock:
2 Hd VCR LCD Menu                         43068
250 8MM Camcorder 40 X                    60073
330DX Digital Camera 1024K P              12707
750SL Digital Camcorder 300X'             10758
AR2 35 MM Camera 8 X                      11499
AR3 35MM Camera 10 X                      12444
Combo Player - 4 Hd VCR + DVD             13527
QX Portable CD Player                     22000
ZC Digital PDA - Standard                 33000
ZT Digital PDA - Commercial               21000

                **End of Report**
```

ACROSS フィールド全体の選択

次の例は、ACROSS フィールドを選択する方法を示しています。ACROSSCOLUMN を選択してフォーマットを設定する場合、レポート出力結果のすべてのフィールドに対して、すべてのデータ値、フィールドタイトル、フィールドに関連する ACROSS フィールド値にフォーマット設定が適用されます。関連するスタイルシートは、リクエスト内で太字で表示されています。

注意：ACROSSCOLUMN 属性に対して P1 値と P2 値をそれぞれこの順序で代用しても、同一の結果を得ることができます。

```
TABLE FILE CENTORD
SUM LINEPRICE LINE_COGS AS 'Line Cost of,Goods Sold'
BY PLANT AS 'Plant'
ACROSS YEAR
WHERE YEAR EQ 2000 OR 2001
HEADING
"Cost Analysis"
FOOTING CENTER
"**End of Report**"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
TYPE=REPORT, ACROSSCOLUMN=N1, STYLE=ITALIC,$
TYPE=REPORT, ACROSSCOLUMN=N2, STYLE=BOLD,$
ENDSTYLE
END
```

出力結果は次のとおりです。

```
Cost Analysis
     YEAR
     2000                                2001
                          Line cost of                       Line cost of
Plant          Line Total    Goods Sold       Line Total      Goods Sold
-------------------------------------------------------------------------
BOS        $1,144,222.47    955,189.00   $125,470,864.68 109,839,382.00
DAL          $291,718.75    230,249.00    $46,868,840.52  41,129,581.00
LA             $1,433.30        986.00    $27,668,201.82  24,027,600.00
ORL                    .             .    $46,955,479.15  41,143,292.00
SEA                    .             .    $10,906,551.21   9,578,621.00
STL          $560,831.70    458,317.00    $93,875,795.10  81,807,263.00

                        **End of Report**
```

FML 行全体の選択

```
TYPE=REPORT, LABEL=label
```

説明

label

明示的な行ラベルです。

FML 行全体の選択

次の例は、行のラベルとデータで構成された FML 行全体を選択する方法を示しています。関連するスタイルシートは、リクエスト内で太字で表示されています。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' LABEL COH OVER
1020 AS 'DEMAND DEPOSITS' LABEL DD OVER
1030 AS 'TIME DEPOSITS' LABEL TD OVER
BAR OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=REPORT, LABEL=COH, STYLE=ITALIC, $
TYPE=REPORT, LABEL=DD, STYLE=ITALIC, $
TYPE=REPORT, LABEL=TD, STYLE=ITALIC, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	AMOUNT
CASH ON HAND	*8,784*
DEMAND DEPOSITS	*4,494*
TIME DEPOSITS	*7,961*
TOTAL CASH	21,239

合計または中間合計行全体の選択

```
TYPE=type, [BY=sortcolumn]
```

説明

type

中間合計または合計を選択します。次のオプションから選択します。

GRANDTOTAL - 総合計です。COLUMN-TOTAL、SUBTOTAL、SUB-TOTAL、RECOMPUTE、SUMMARIZE のいずれかで生成されます。

SUBTOTAL - 中間合計です。SUBTOTAL、SUB-TOTAL、RECOMPUTE、SUMMARIZE のいずれかで生成されます。

RECAP - 中間合計の計算です。ON sortfield RECAP または ON sortfield COMPUTE のいずれかで生成されます。

BY

複数の中間合計コマンドがそれぞれ異なる BY フィールドの列に関連付けられている場合に、どの中間合計コマンドにフォーマットを設定するかを選択することができます。

sortcolumn

レポートに複数の中間合計コマンドが使用されている場合に、そのコマンドの 1 つと関連付けられた BY フィールドを選択します。ソートフィールドを選択するには、フィールド名を使用します。

合計行全体の選択

次の例は、スタイルシートで COLUMN-TOTAL 行全体を選択する方法を示しています。リクエストの太字の部分が、この例に関連するスタイルシートです。

```
TABLE FILE SALES
SUM RETURNS DAMAGED AND ROW-TOTAL AND COLUMN-TOTAL
BY PROD_CODE
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=GRANDTOTAL, STYLE=BOLD, SIZE=12, $
ENDSTYLE
END
```

出力結果は次のとおりです。

PROD_CODE	RETURNS	DAMAGED	TOTAL
B10	13	10	23
B12	4	3	7
B17	4	2	6
B20	1	2	3
C13	3	0	3
C17	0	0	0
C7	5	4	9
D12	3	2	5
E1	4	7	11
E2	9	4	13
E3	12	11	23
TOTAL	**58**	**45**	**103**

行合計の選択

次の例は、行合計を選択する方法を示しています。リクエストの太字の部分が、この例に関連するスタイルシートです。なお、行合計のインスタンスにフォーマットを設定する場合は、スタイルシートに WHEN ステートメントを追加することができます。詳細は、レポートフォーマット設定の管理を参照してください。

```
TABLE FILE SALES
SUM RETURNS DAMAGED AND ROW-TOTAL
BY PROD_CODE AS 'PRODUCT,CODE'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=REPORT, COLUMN=ROWTOTAL, STYLE=BOLD, $
ENDSTYLE
END
```

出力結果は次のとおりです。

PRODUCT CODE	RETURNS	DAMAGED	TOTAL
B10	13	10	**23**
B12	4	3	**7**
B17	4	2	**6**
B20	1	2	**3**
C13	3	0	**3**
C17	0	0	**0**
C7	5	4	**9**
D12	3	2	**5**
E1	4	7	**11**
E2	9	4	**13**
E3	12	11	**23**

SUBTOTAL および GRANDTOTAL 行のタグの選択

ここで説明する「タグ」とは、レポートの SUBTOTAL および GRANDTOTAL 行の左端に表示されるテキストのことです。このタグは、これらの行に表示されたデータのタイプを識別するために使用されます。このタグの生成に使用するテキストは、SUBTOTAL 構文に AS 名を追加することでカスタマイズすることができます。

中間合計タグおよび総合計タグのスタイルは、これらの行の残りの部分とは独立して定義することができます。タグに使用可能なテキスト属性にはフォント、色、サイズ、スタイルがあり、これらの属性を使用することで、タグを区別したり、強調したりできます。また、タグをドリルダウンリンクとして表示するためのスタイルを適用することもできます。

サポートされるスタイルはテキスト属性のみです。境界、背景色、位置揃えなど、セルや列に適用される機能はサポートされません。

この機能は、PDF、DHTML、HTML、AHTML、PPTX、XLSX、EXL2K フォーマットで使用できます。

中間合計タグおよび総合計タグのスタイル設定

```
TYPE={SUBTOTAL|GRANDTOTAL}, OBJECT=TAG,
   [FONT=font], [SIZE=size], [STYLE=style],
   [COLOR={color|RGB({rgb|#hexcolor})],
   [drilltype=drillparms], $
```

説明

font

フォントの名前です。

size

フォントのポイントサイズです。

style

フォントのスタイルです (例、bold、italic、bold+italic)。

color

色の名前です。

rgb

赤、緑、青の組み合わせを使用して、色を指定します。

(r, g, b) は、赤、緑、青の順序で指定する輝度です。0 (ゼロ) から 255 までの値が使用されます。0 (ゼロ) は最も輝度が低く、255 は最も輝度が高いことを示します。色の構成要素の輝度をすべて同じ値に設定すると、灰色になります。

#hexcolor

色の 16 進数値です。たとえば、赤色の 16 進数値は FF0000 です。16 進数値では大文字と小文字は区別されません。また、先頭にシャープ記号 (#) を付ける必要があります。

drilltype

有効なドリルダウン属性です (例、URL=、FOCEXEC=)。ドリルダウンリンクについての詳細は、レポートの別リソースへのリンクを参照してください。

drillparms

ドリルダウンのタイプを示す有効な属性値です。

SUBTOTAL および GRANDTOTAL タグのスタイル設定

次のリクエストは、GGSALES データソースを使用し、中間合計行および総合計行を生成します。中間合計行のタグは、斜体の白色です。総合計行のタグには、URL へのドリルダウンリンクが指定されています。

```
TABLE FILE GGSALES
SUM UNITS/D8C DOLLARS/D12CM BUDUNIT/D8C BUDDOLLARS/D12CM
BY REGION
BY CATEGORY
ON REGION SUBTOTAL
HEADING
"Gotham Grinds Sales Report"
ON TABLE SET HTMLCSS ON
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET DROPBLNKLINE ALL
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/javaassist/intl/EN/ENIADefault_combine.sty,$
TYPE=SUBTOTAL, OBJECT=TAG,STYLE=ITALIC,COLOR=WHITE,$
TYPE=GRANDTOTAL, BACKCOLOR='LIGHT GREY',$
TYPE=GRANDTOTAL, OBJECT=TAG,URL='http://www.informationbuilders.com',$
ENDSTYLE
END
```

出力結果は次のとおりです。

Gotham Grinds Sales Report

Region	Category	Unit Sales	Dollar Sales	Budget Units	Budget Dollars
Midwest	Coffee	332,777	$4,178,513	335,526	$4,086,032
	Food	341,414	$4,338,271	339,263	$4,220,721
	Gifts	230,854	$2,883,881	232,318	$2,887,620
*TOTAL Midwest		905,045	$11,400,665	907,107	$11,194,373
Northeast	Coffee	335,778	$4,164,017	335,920	$4,252,462
	Food	353,368	$4,379,994	351,431	$4,453,907
	Gifts	227,529	$2,848,289	227,008	$2,870,552
*TOTAL Northeast		916,675	$11,392,300	914,359	$11,576,921
Southeast	Coffee	350,948	$4,415,408	355,693	$4,431,429
	Food	349,829	$4,308,731	351,509	$4,409,288
	Gifts	234,455	$2,986,240	235,045	$2,967,254
*TOTAL Southeast		935,232	$11,710,379	942,247	$11,807,971
West	Coffee	356,763	$4,473,517	358,784	$4,523,963
	Food	340,234	$4,202,337	335,361	$4,183,244
	Gifts	235,042	$2,977,092	236,636	$2,934,306
*TOTAL West		932,039	$11,652,946	930,781	$11,641,513
TOTAL		3,688,991	$46,156,290	3,694,494	$46,220,778

データの選択

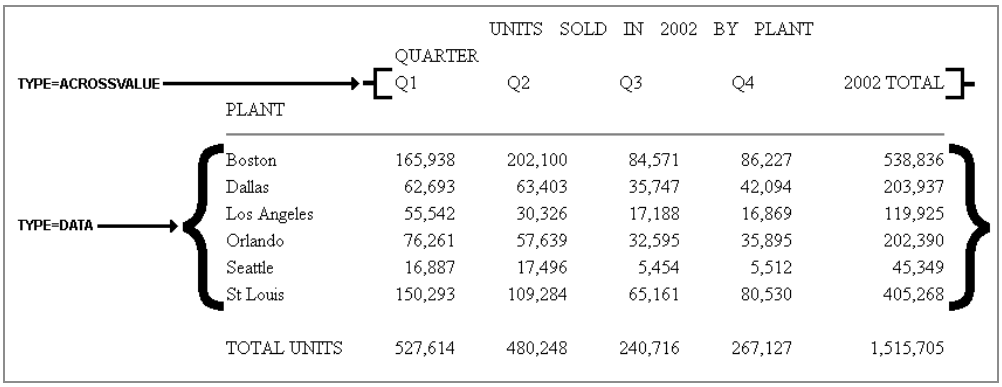
レポートのさまざまな種類のデータを選択してフォーマットを設定することができます。データの種類には次のものがあります。

- **レポートの全データ**　詳細は、データの選択を参照してください。
- **データフィールド**　ソートフィールドと表示フィールドがあります。詳細は、データの選択を参照してください。
- **ソート行**　ACROSS フィールドの値です。詳細は、Identify a Row of Horizontal Sort (ACROSS) Dataを参照してください。
- **合計と中間合計**　詳細は、データの選択を参照してください。

次の例は、DATA および ACROSSVALUE の各コンポーネントがレポート内で表示される位置およびこれらのコンポーネントの選択に使用する TYPE の値を示しています。

```
TABLE FILE CENTORD
HEADING CENTER
"UNITS SOLD IN 2002 BY PLANT"
SUM QUANTITY AND ROW-TOTAL AS '2002 TOTAL'
```

```
ACROSS QUARTER
BY PLANTLNG AS 'PLANT'
WHERE YEAR EQ 2002
ON TABLE COLUMN-TOTAL AS 'TOTAL UNITS'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
END
```



注意：このリクエストは、レポート内の各コンポーネントの表示位置を単に示すことが目的であるため、ここではスタイルシートは省略されています。

全データの選択

スタイルシートのすべてのデータを選択するには、次の属性および値を使用します。ただし、列合計、総合計、中間合計、ACROSS フィールドの値は、個別に選択する必要があります。

```
TYPE = DATA
```

レポート全データの選択

次の例は、レポートのすべてのデータを選択する方法を示しています。リクエストの太字の部分が、この例に関連するスタイルシートです。

```
TABLE FILE CENTORD
HEADING CENTER
"UNITS SOLD IN 2002 BY PLANT"
SUM QUANTITY AND ROW-TOTAL AS '2002 TOTAL'
ACROSS QUARTER
BY PLANTLNG AS 'PLANT'
WHERE YEAR EQ 2002
ON TABLE COLUMN-TOTAL AS 'TOTAL UNITS'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=DATA, STYLE=BOLD, $
ENDSTYLE
END
```

出力結果は次のとおりです。

UNITS SOLD IN 2002 BY PLANT

PLANT	QUARTER Q1	Q2	Q3	Q4	2002 TOTAL
Boston	**165,938**	**202,100**	**84,571**	**86,227**	**538,836**
Dallas	**62,693**	**63,403**	**35,747**	**42,094**	**203,937**
Los Angeles	**55,542**	**30,326**	**17,188**	**16,869**	**119,925**
Orlando	**76,261**	**57,639**	**32,595**	**35,895**	**202,390**
Seattle	**16,887**	**17,496**	**5,454**	**5,512**	**45,349**
St Louis	**150,293**	**109,284**	**65,161**	**80,530**	**405,268**
TOTAL UNITS	527,614	480,248	240,716	267,127	1,515,705

データフィールドの選択

```
TYPE=DATA, COLUMN=column
```

説明

column

フォーマットを設定する 1 つまたは複数のフィールドを指定します。値のリストは、レポート全体、列全体、行全体の選択を参照してください。

データフィールドの選択

次の例は、データフィールドを選択する方法を示しています。リクエストの太字の部分が、この例に関連するスタイルシートです。

なお、Nn を使用してフィールドを選択する場合は、NOPRINT フィールドもその数に含まれます。このレポートでは Product Name フィールドが先頭フィールドですが、NOPRINT フィールドが存在するため、このフィールドは N2 として認識されます。

```
TABLE FILE CENTORD
PRINT QUANTITY LINEPRICE LINE_COGS
BY ORDER_NUM NOPRINT
BY PRODNAME
WHERE ORDER_NUM EQ '48045'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
TYPE=DATA, COLUMN=N2, STYLE=ITALIC,$
ENDSTYLE
END
```

出力結果は次のとおりです。

```
Cost Analysis
     YEAR
     2000                                    2001
                           Line cost of                          Line cost of
Plant          Line Total    Goods Sold       Line Total         Goods Sold
-------------------------------------------------------------------------------
BOS        $1,144,222.47    955,189.00    $125,470,864.68  109,839,382.00
DAL          $291,718.75    230,249.00     $46,868,840.52   41,129,581.00
LA             $1,433.30        986.00     $27,668,201.82   24,027,600.00
ORL                    .            .     $46,955,479.15   41,143,292.00
SEA                    .            .     $10,906,551.21    9,578,621.00
STL          $560,831.70    458,317.00     $93,875,795.10   81,807,263.00

                        **End of Report**
```

ACROSS フィールドデータ行の選択

```
TYPE=ACROSSVALUE, [ACROSS={fieldname|{N|A}n}]
```

説明

ACROSS

リクエストで複数の ACROSS フィールドを使用している場合は、ACROSS 識別子を使用して特定のフィールドを選択することができます。リクエストに複数の ACROSS フィールドが使用されている場合に限り、ACROSS 識別子を使用する必要があります。

fieldname

ACROSS フィールドのフィールド名を指定します。

n

横方向ソート行の位置に基づいて、ACROSS フィールドを指定します。同一スタイルシート内の fieldname 指定と組み合わせることはできません。

ACROSS フィールドデータ行の選択

次の例は、ACROSS フィールドのデータ行を選択する方法を示しています。リクエストの太字の部分が、この例に関連するスタイルシートです。

```
TABLE FILE CENTORD
HEADING
"Units Sold"
SUM QUANTITY
BY PRODNAME
ACROSS PLANT AS 'Manufacturing Plant'
WHERE PRODTYPE EQ 'Digital'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=HEADING, SIZE=12, $
TYPE=ACROSSVALUE, ACROSS=PLANT, STYLE=BOLD, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Units Sold

Product Name:	Manufacturing Plant BOS	DAL	LA	ORL	SEA	STL
330DX Digital Camera 1024K P	3,949	2,767	562	561	2	1,132
650DL Digital Camcorder 150 X	44,930	14,548	7,995	17,972	4,967	31,436
750SL Digital Camcorder 300 X	4,098	2	6	283	642	1,644
Combo Player - 4 Hd VCR + DVD	27,687	15,985	9,634	13,816	2,359	22,097
DVD Upgrade Unit for Cent. VCR	47,112	25,602	16,152	23,429	4,539	32,875
QX Portable CD Player	108,696	35,355	26,215	46,388	8,056	83,564
R5 Micro Digital Tape Recorder	188,384	73,255	41,169	63,120	15,385	142,714
ZC Digital PDA - Standard	45,890	13,154	4,664	10,249	6,439	30,213
ZT Digital PDA - Commercial	181,750	70,111	41,149	61,244	15,039	130,942

注意：スタイルシートの ACROSS 属性に対して N1 値を代用しても同一の結果を得ることができます。たとえば、TYPE=ACROSSVALUE、ACROSS=N1、STYLE=BOLD、$ と記述します。

ACROSS フィールドデータ合計行 (ACROSS-TOTAL) の選択

次の例は、ACROSSVALUE コンポーネントと数値列参照 (Nn) を使用して、ACROSS フィールドのデータ行の合計 (ACROSS-TOTAL) を選択する方法を示しています。リクエストの太字の部分が、この例に関連するスタイルシートです。

```
TABLE FILE CENTORD
SUM QUANTITY
BY PRODNAME
ACROSS PLANT AS 'Manufacturing Plant'
ACROSS-TOTAL AS 'Plant Totals'
WHERE PRODTYPE EQ 'Digital'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=ACROSSVALUE, COLUMN=N8, STYLE=ITALIC, COLOR='RED', $
```

```
ENDSTYLE
END
```

次の出力結果では、ACROSS-TOTAL 値である Plant Totals が赤色の斜体で表示されています。

	Manufacturing Plant						
Product Name:	BOS	DAL	LA	ORL	SEA	STL	*Plant Totals*
330DX Digital Camera 1024K P	3,949	2,767	562	561	2	1,132	8,973
650DL Digital Camcorder 150 X	44,930	14,548	7,995	17,972	4,967	31,436	121,848
750SL Digital Camcorder 300 X	4,098	2	6	283	642	1,644	6,675
Combo Player - 4 Hd VCR + DVD	27,687	15,985	9,634	13,816	2,359	22,097	91,578
DVD Upgrade Unit for Cent. VCR	47,112	25,602	16,152	23,429	4,539	32,875	149,709
QX Portable CD Player	108,696	35,355	26,215	46,388	8,056	83,564	308,274
R5 Micro Digital Tape Recorder	188,384	73,255	41,169	63,120	15,385	142,714	524,027
ZC Digital PDA - Standard	45,890	13,154	4,664	10,249	6,439	30,213	110,609
ZT Digital PDA - Commercial	181,750	70,111	41,149	61,244	15,039	130,942	500,235

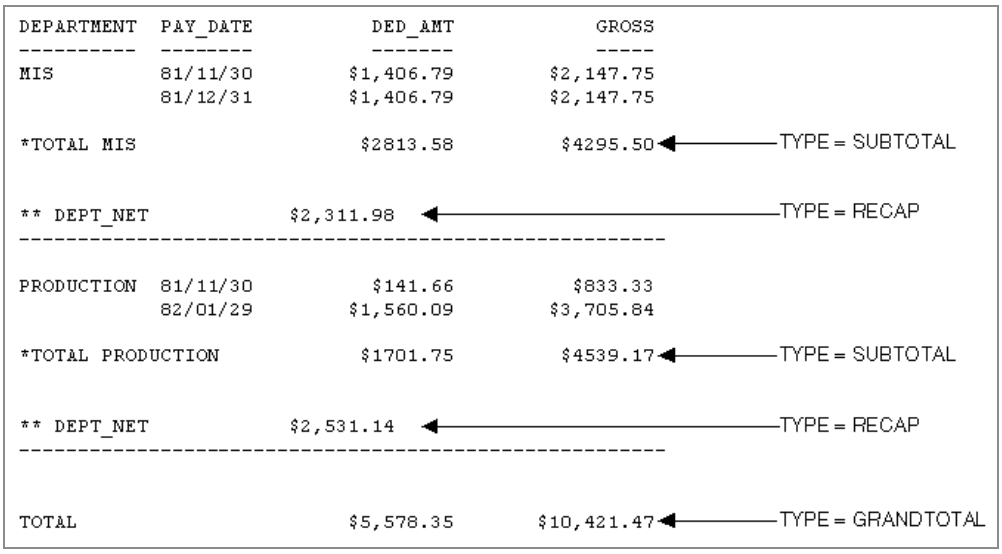
合計と中間合計の選択

総合計、中間合計、中間合計の計算 (ON sortfield RECAP または ON sortfield COMPUTE で生成)、列合計、行合計をスタイルシートで個別に選択してフォーマットを設定することができます。行合計を選択する方法についての詳細は、Identifying an Entire Report, Column, or Rowを参照してください。

次の例は、これらのコンポーネントがレポート内に表示される位置およびこれらのコンポーネントの選択に使用する TYPE の値を示しています。

```
TABLE FILE EMPLOYEE
SUM DED_AMT AND GROSS
BY DEPARTMENT BY PAY_DATE
ON DEPARTMENT RECAP DEPT_NET/D8.2M = GROSS-DED_AMT;
WHEN PAY_DATE GT 820101
ON DEPARTMENT SUBTOTAL
END
```

次の図は各コンポーネントを示しています。



注意：このリクエストは、異なるタイプの合計と中間合計を選択する方法を単に示すことが目的であるため、ここではスタイルシートは省略されています。

総合計、中間合計、中間合計計算の指定

```
TYPE=type, [BY=sortfield] [coltype=column]
```

説明

type

中間合計または合計を選択します。次のオプションから選択します。

GRANDTOTAL - 総合計です。COLUMN-TOTAL、SUBTOTAL、SUB-TOTAL、RECOMPUTE、SUMMARIZE のいずれかで生成されます。

SUBTOTAL - 中間合計です。SUBTOTAL、SUB-TOTAL、RECOMPUTE、SUMMARIZE のいずれかで生成されます。

RECAP - 中間合計の計算です。ON sortfield RECAP または ON sortfield COMPUTE のいずれかで生成されます。

BY

リクエストに複数の BY フィールドが使用され、それぞれのフィールドに複数の中間合計コマンドが関連付けられている場合は、BY 識別子を使用して特定のフィールドを選択す

ることができます。この方法は、それぞれの中間合計を異なる方法でフォーマット設定する場合や、特定の中間合計のみにフォーマットを設定する場合に役立ちます。

リクエストに複数の BY フィールドが使用されている場合に限り、BY 識別子を使用する必要があります。

sortfield

レポートの複数の中間合計コマンドに関連付けられている BY フィールドを指定します。この値としてフィールド名 (BY=fieldname) を使用します。

coltype

フォーマット設定を適用する特定の列を選択します。宣言に COLUMN または ACROSSCOLUMN 識別子を使用する場合、フォーマット設定は中間合計値のみに適用され、ラベルテキストには適用されません。選択できる値には次のものがあります。

COLUMN - PRINT、LIST、SUM、COUNT で生成される表示フィールドまたは COMPUTE で生成される一時項目です。

ACROSSCOLUMN - 表示フィールドまたは一時項目の各インスタンスが、ACROSS フィールド行に繰り返し表示されます。

1 つのコマンドで複数のフィールドが合計または中間合計され、スタイルシートでフィールドが指定されていない場合は、合計または中間合計のすべてのフィールドにフォーマット設定が適用されます。フォーマット設定は、合計値および中間合計値のラベルテキストにも適用されます。

column

フォーマットを設定する合計または中間合計のフィールドを指定します。値のリストは、レポート全体、列全体、行全体の選択を参照してください。

総合計の選択

次の例は、リクエストで総合計を選択する方法を示しています。この例では、LINE_COGS フィールドの総合計値のみにフォーマットを設定するため、スタイルシートで COLUMN 属性を指定しています。このリクエストの総合計は、COLUMN-TOTAL で生成されます。リクエストの太字の部分が、この例に関連するスタイルシートです。

注意

- 総合計行全体にスタイルを設定するには、スタイルシートから COLUMN 属性を削除します。
- スタイルシートの COLUMN 属性に対して N5 値、P5 値、C3 値を代用して、同一の結

果を得ることができます。

```
TABLE FILE CENTORD
SUM QUANTITY LINEPRICE LINE_COGS AND COLUMN-TOTAL
BY ORDER_NUM BY PRODNAME
WHERE ORDER_NUM EQ '48053' OR '48798'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
TYPE=GRANDTOTAL, COLUMN=LINE_COGS, STYLE=BOLD, SIZE=11,$
ENDSTYLE
END
```

出力結果は次のとおりです。

Order Number:	Product Name:	Quantity:	Line Total	Line Cost Of Goods Sold
48053	150 8MM Camcorder 20 X	1	$338.28	240.00
	Combo Player - 4 Hd VCR + DVD	1	$415.73	289.00
	DVD Upgrade Unit for Cent. VCR	1	$210.42	139.00
	R5 Micro Digital Tape Recorder	1	$88.52	69.00
	ZT Digital PDA - Commercial	1	$523.75	349.00
48798	150 8MM Camcorder 20 X	1	$319.89	240.00
	Combo Player - 4 Hd VCR + DVD	1	$418.26	289.00
	DVD Upgrade Unit for Cent. VCR	1	$200.63	139.00
	R5 Micro Digital Tape Recorder	1	$92.09	69.00
	ZT Digital PDA - Commercial	1	$483.52	349.00
TOTAL		10	$3,091.09	**2,172.00**

中間合計の選択

次の例は、リクエストで中間合計を選択する方法を示しています。この例では、QUANTITY および LINE_COGS フィールドの中間合計値のみにフォーマットを設定するため、スタイルシートで COLUMN 属性を指定しています。

また、3 つの BY フィールドのうちの 2 つ (PLANT と ORDER_NO) に関連付けられた SUBTOTAL コマンドが 2 つあるため、各宣言で BY 属性を指定することにより、フォーマット設定が正しい値に適用されるようにします。関連するスタイルシートは、リクエスト内で太字で表示されています。

注意

- すべての中間合計行にスタイルを設定するには、スタイルシートから COLUMN および

BY 属性を削除します。

- COLUMN=LINE_COGS 属性に対して COLUMN=N6、COLUMN=P6、COLUMN=C3 の値を代用して、同一の結果を得ることができます。
- COLUMN=QUANTITY 属性に対して COLUMN=N4、COLUMN=P4、COLUMN=C1 の値を代用して、同一の結果を得ることができます。

```
TABLE FILE CENTORD
SUM QUANTITY LINEPRICE LINE_COGS AS 'Line Cost of, Goods Sold'
BY PLANT
BY ORDER_NUM
BY PRODNAME
ON PLANT SUBTOTAL
ON ORDER_NUM SUBTOTAL
WHERE ORDER_NUM EQ '35774' OR '48041'
WHERE PLANT EQ 'BOS'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
TYPE=SUBTOTAL, BY=PLANT, COLUMN=LINE_COGS, STYLE=BOLD+ITALIC,
    COLOR=BLUE,$
TYPE=SUBTOTAL, BY=ORDER_NUM, COLUMN=QUANTITY, STYLE=BOLD, SIZE=11,$
ENDSTYLE
END
```

出力結果は次のとおりです。

Manufacturing Plant	Order Number:	Product Name:	Quantity:	Line Total	Line Cost of Goods Sold
BOS	35774	110 VHS-C Camcorder 20 X	146	$38,817.56	36,354.00
		120 VHS-C Camcorder 40 X	146	$40,426.90	37,814.00
		2 Hd VCR LCD Menu	279	$39,693.32	35,991.00
		650DL Digital Camcorder 150 X	146	$116,907.15	103,660.00
*TOTAL ORDER_NUM 35774			**717**	$235,844.93	213,819.00
	48041	150 8MM Camcorder 20 X	1	$307.43	240.00
		Combo Player - 4 Hd VCR + DVD	1	$398.95	289.00
		DVD Upgrade Unit for Cent. VCR	1	$194.40	139.00
		R5 Micro Digital Tape Recorder	1	$93.91	69.00
		ZT Digital PDA - Commercial	1	$528.38	349.00
*TOTAL ORDER_NUM 48041			**5**	$1,523.07	1,086.00
*TOTAL PLANT BOS			722	$237,368.00	***214,905.00***
TOTAL			722	$237,368.00	214,905.00

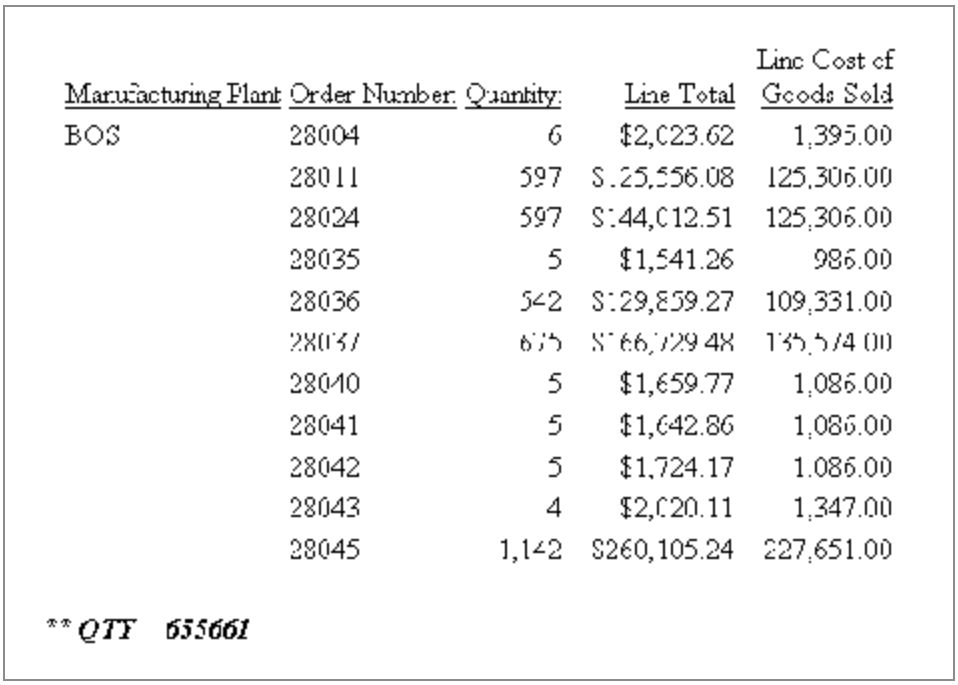
中間合計の計算 (RECAP/COMPUTE) の選択

次の例は、RECAP または COMPUTE 句で生成される中間合計の計算を選択する方法を示しています。この例では、ON PLANT RECAP QTY/F6=QUANTITY を使用して中間合計の計算を生成します。リクエストの太字の部分が、この例に関連するスタイルシートです。

注意：リクエストに複数の RECAP または COMPUTE フィールドがある場合にこれらのフィールドを区別するには、スタイルシートに BY=fieldname を追加します。

```
TABLE FILE CENTORD
SUM QUANTITY LINEPRICE LINE_COGS AS 'Line Cost of, Goods Sold'
BY PLANT BY ORDER_NUM
ON PLANT RECAP QTY/F6=QUANTITY;
WHERE PLANT EQ 'BOS'
WHERE ORDER_NUM LT '56098'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=RECAP, STYLE=BOLD+ITALIC, $
ENDSTYLE
END
```

出力結果は次のとおりです。



Manufacturing Plant	Order Number	Quantity	Line Total	Line Cost of Goods Sold
BOS	28004	6	$2,023.62	1,395.00
	28011	597	$ 25,556.08	125,306.00
	28024	597	$ 44,012.51	125,306.00
	28035	5	$1,541.26	986.00
	28036	542	$ 29,859.27	109,331.00
	28037	675	$ 66,729.48	135,574.00
	28040	5	$1,659.77	1,086.00
	28041	5	$1,642.86	1,086.00
	28042	5	$1,724.17	1,086.00
	28043	4	$2,020.11	1,347.00
	28045	1,142	$260,105.24	227,651.00

** QTY 655661

見出し、脚注、タイトル、FML テキストの選択

レポートのデータは、見出し、脚注、タイトルで全体が構成されます。これらの要素はデータのコンテキストを提供します。レポートに表示する見出し、脚注、タイトルのさまざまな種類を選択してフォーマットを設定することができます。その種類には次のものがあります。

- レポート、ページ、ソート項目を指定している見出し
- レポート、ページ、ソート項目を指定している脚注
- フィールドタイトル
- ACROSS フィールドのタイトルと値
- FML (Financial Modeling Language) レポートのテキスト

フィールドタイトルと行タイトルの選択

スタイルシートでレポートのフィールドタイトルおよび ACROSS フィールド値を選択してフォーマットを設定することができます。次の例は、フィールドタイトルおよび横方向ソートの値がレポート内に表示される位置、およびこれらの選択に使用する TYPE 値を示しています。

```
TABLE FILE EMPLOYEE
SUM GROSS AND DED_AMT
ACROSS DEPARTMENT BY PAY_DATE
END
```

```
PAGE     1


            DEPARTMENT ◄──────────────────────────────────────────── TYPE = ACROSSTITLE
            MIS                                PRODUCTION ◄──────── TYPE = ACROSSVALUE
  PAY_DATE             GROSS          DED_AMT             GROSS          DED_AMT ◄── TYPE = TITLE
  ---------------------------------------------------------------------------
  81/11/30         $2,147.75        $1,406.79           $833.33          $141.66
  81/12/31         $2,147.75        $1,406.79           $833.33          $141.66
  82/01/29         $3,247.75        $1,740.89         $3,705.84        $1,560.09
  82/02/26         $3,247.75        $1,740.89         $4,959.84        $2,061.69
  82/03/31         $3,247.75        $1,740.89         $4,959.84        $2,061.69
  82/04/30         $5,890.84        $3,386.73         $4,959.84        $2,061.69
  82/05/28         $6,649.50        $3,954.35         $7,048.84        $3,483.88
  82/06/30         $7,460.00        $4,117.03         $7,048.84        $3,483.88
  82/07/30         $7,460.00        $4,117.03         $7,048.84        $3,483.88
  82/08/31         $9,000.00        $4,575.72         $9,523.84        $4,911.12
 PAGE     1
```

注意：このリクエストは、レポートのフィールドタイトルと ACROSS フィールドの値を選択する方法を示すことが目的であるため、ここではスタイルシートは省略されています。

フィールドタイトルの選択

```
TYPE=TITLE, [COLUMN=column]
```

説明

COLUMN

1 つまたは複数のフィールドタイトルを指定します。この属性および値を指定しない場合は、レポートのすべてのフィールドタイトルにフォーマット設定が適用されます。

column

タイトルのフォーマットを設定するフィールドを指定します。フィールド値についての詳細は、レポート全体、列全体、行全体の選択を参照してください。

ACROSS フィールドのタイトルおよび値の選択

```
TYPE={ACROSSTITLE|ACROSSVALUE}, [ACROSS=column]
```

説明

ACROSSTITLE

ACROSS フィールドのタイトルを指定します。

ACROSSVALUE

ACROSS フィールドの値を指定します。

ACROSS フィールドの値は厳密にはタイトルではありませんが、多くの場合、この値はその下に表示されるフィールドタイトルを分類するタイトルとして機能します。

ACROSS

特定の ACROSS フィールドのタイトルまたは値を指定します。この属性および値を指定しない場合は、レポートの ACROSS フィールドのすべてのタイトルまたは値にフォーマット設定が適用されます。

column

フォーマットを設定するタイトルまたは値の ACROSS フィールドを指定します。この属性に割り当て可能な値についての詳細は、Identify a Row of Horizontal Sort (ACROSS) Dataを参照してください。

フィールドタイトルおよび ACROSS フィールド値の選択

次の例は、BY フィールドのタイトル、ACROSS フィールドのタイトル、ACROSS フィールドの値を選択する方法を示しています。BY フィールドのタイトル (TYPE=TITLE) は、Manufacturing Plant、Quantity Sold、Product Cost です。ACROSS フィールドのタイトル (TYPE=ACROSSTITLE) は Year です。ACROSS フィールドの値 (TYPE=ACROSSVALUE) は 2001、2002、TOTAL です。

ここでは、ドリルダウンの値を、TOTAL の ROW-TOTAL ではなく、2001 と 2002 の ACROSS 値ごとに割り当てる方法について説明しています。リクエストの太字の部分が、この例で説明するスタイルシートです。

```
TABLE FILE CENTORD
SUM QUANTITY AS 'Quantity,Sold' LINE_COGS/I9 AS 'Product,Cost'
BY PLANT
ACROSS YEAR
WHERE YEAR EQ '2001' OR '2002'
HEADING
"Plant Production Cost Analysis"
ON TABLE ROW-TOTAL AS 'TOTAL'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
TYPE=TITLE, STYLE=BOLD, $
TYPE=ACROSSTITLE, STYLE=BOLD, $
TYPE=ACROSSVALUE, STYLE=BOLD+ITALIC, COLOR=BLUE, FOCEXEC=DETAILS, $
TYPE=ACROSSVALUE, COLUMN=N4, STYLE=BOLD, COLOR=RED, $
TYPE=ACROSSVALUE, COLUMN=ROWTOTAL(1), COLOR='BLACK',FOCEXEC=NONE, $
ENDSTYLE
END
```

レポートの出力結果は次のとおりです。

Plant Production Cost Analysis

	YEAR					
	2001		2002		TOTAL	
Manufacturing Plant	Quantity Sold	Product Cost	Quantity Sold	Product Cost	Quantity Sold	Product Cost
BOS	491,080	109839382	538,836	120518527	1,029,916	230357909
DAL	185,785	41129581	203,937	45147907	389,722	86277488
LA	109,326	24027600	119,925	26356066	229,251	50383666
ORL	184,519	41143292	202,390	45127226	386,909	86270518
SEA	41,331	9578621	45,349	10510177	86,680	20088798
STL	369,456	81807263	405,268	89734521	774,724	171541784

FML レポートのテキストの選択

```
TYPE=FREETEXT, LABEL={Rn|label}
```

説明

n

暗示指定の行ラベルです。n の値を特定するには、目的とする行を含めて、その行までの行数を数えます。

label

明示的な行ラベルです。

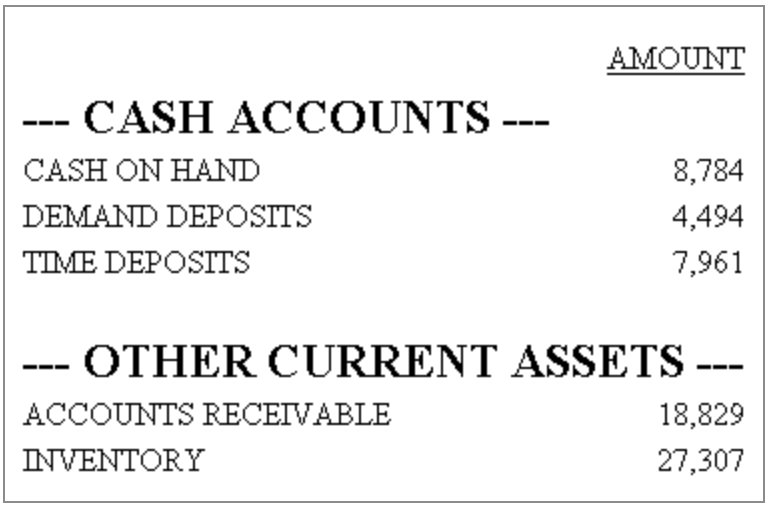
FML レポートのテキストの選択

次の例は、FML レポートのテキストを選択する方法を示しています。この例で選択するテキストは、「CASH ACCOUNTS」と「OTHER CURRENT ASSETS」です。リクエストの太字の部分が、この例に関連するスタイルシートです。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
" --- CASH ACCOUNTS ---" LABEL CA            OVER
```

```
1010 AS 'CASH ON HAND'                          OVER
1020 AS 'DEMAND DEPOSITS'                       OVER
1030 AS 'TIME DEPOSITS'                         OVER
" "                                             OVER
" --- OTHER CURRENT ASSETS ---" LABEL OCA      OVER
1100 AS 'ACCOUNTS RECEIVABLE'                   OVER
1200 AS 'INVENTORY'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=FREETEXT, LABEL=CA, STYLE=BOLD, SIZE=12, $
TYPE=FREETEXT, LABEL=OCA, STYLE=BOLD, SIZE=12, $
ENDSTYLE
END
```

出力結果は次のとおりです。



	AMOUNT
--- CASH ACCOUNTS ---	
CASH ON HAND	8,784
DEMAND DEPOSITS	4,494
TIME DEPOSITS	7,961
--- OTHER CURRENT ASSETS ---	
ACCOUNTS RECEIVABLE	18,829
INVENTORY	27,307

見出しまたは脚注の選択

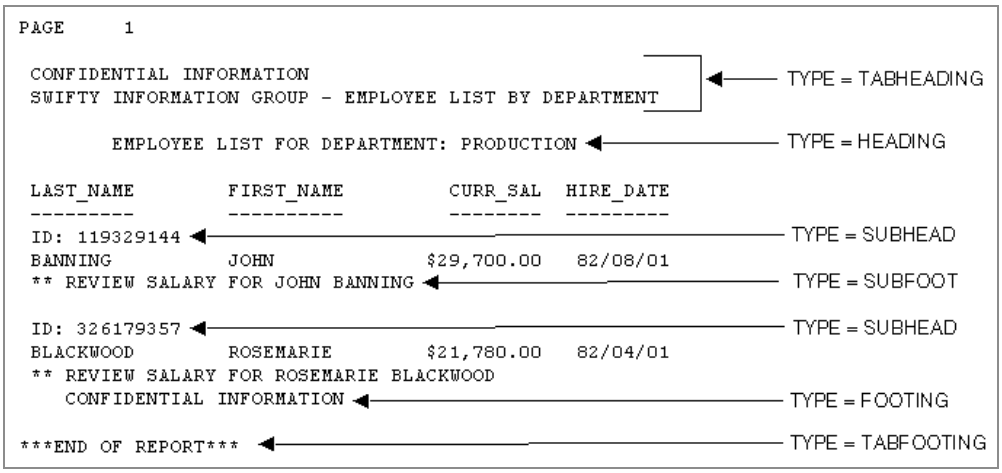
スタイルシートでレポートの見出しと脚注、特定の行、テキスト、これらのコンポーネント内のフィールドを選択してフォーマットを設定することができます。

TABLE リクエストでは、複数ページの見出しおよび脚注を指定することができます。見出しまたは脚注のそれぞれについて、取得されるデータに WHEN 句を使用して、レポート出力に見出しまたは脚注を表示するかどうかを指定することができます。CONDITION スタイルシート属性を使用して、特定の WHEN 句を識別することができ、見出しと脚注のそれぞれに、個別にスタイルを設定することができます。

次のコードおよび出力結果の例は、レポートの見出し (TABHEADING)、ページの見出し、(HEADING)、ソート項目を指定している見出し (SUBHEAD)、ソート項目を指定している脚注 (SUBFOOT)、レポートの脚注 (TABFOOTING) のそれぞれがレポート内で表示される位置およびこれらのコンポーネントの選択に使用する TYPE の値を示しています。

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL HIRE_DATE
BY LAST_NAME
BY FIRST_NAME
ON TABLE SUBHEAD
"CONFIDENTIAL INFORMATION"
"SWIFTY INFORMATION GROUP - EMPLOYEE LIST BY DEPARTMENT"
HEADING CENTER
"</1>EMPLOYEE LIST FOR DEPARTMENT: <DEPARTMENT"
ON LAST_NAME SUBHEAD
"ID: <EMP_ID"
ON LAST_NAME SUBFOOT
"** REVIEW SALARY FOR <FIRST_NAME <LAST_NAME"
FOOTING
"CONFIDENTIAL INFORMATION"
ON TABLE SUBFOOT
"</1>***END OF REPORT***"
ON TABLE SET ONLINE-FMT STANDARD
END
```

次の出力結果は、上記のコード例に対応しています。



注意：このリクエストは、異なるタイプの見出しと脚注を選択する方法を示すことを目的としているため、ここではスタイルシートは省略されています。

見出しおよび脚注の選択

```
TYPE=headfoot, [BY=sortcolumn]
```

説明

headfoot

見出しまたは脚注を指定します。次のオプションから選択します。

- **TABHEADING** レポート見出しです。レポートの先頭に一度だけ表示され、ON TABLE SUBHEAD で生成されます。
- **TABFOOTING** レポート脚注です。レポートの末尾に一度だけ表示され、ON TABLE SUBFOOT で生成されます。
- **HEADING** ページ見出しです。各レポートページの上部に表示され、HEADING で生成されます。
- **FOOTING** ページ脚注です。各レポートページの下部に表示され、FOOTING で生成されます。
- **SUBHEAD** ソート項目を指定している見出しです。BY フィールドグループの先頭に表示され、ON sortfield SUBHEAD で生成されます。
- **SUBFOOT** ソート項目を指定している脚注です。BY ソートグループの末尾に表示され、ON sortfield SUBFOOT で生成されます。

BY

ソート項目を指定している見出しまたは脚注が複数存在し、それぞれ異なる BY フィールド列に関連付けられている場合に、フォーマットの設定対象を指定することができます。

ソート項目を指定している見出しまたは脚注が複数存在し、それぞれ異なる BY フィールド列に関連付けられている場合、この属性を指定しないと、これらすべてにフォーマットが設定されます。

sortcolumn

レポートでソート項目を指定している見出しまたは脚注の 1 つに関連付ける BY フィールド列を指定します。

ページ見出しとレポート脚注の選択

次の例は、レポートの各ページの上部に表示されるページ見出しおよびレポートの最終ページのみに表示されるレポート脚注を選択する方法を示しています。関連するスタイルシートは、リクエスト内で太字で表示されています。

```
TABLE FILE CENTORD
HEADING
"Sales Quantity and Amount by Plant"
SUM QUANTITY LINEPRICE
BY PLANT
ON TABLE SUBFOOT
" "
"***End of Report***"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=HEADING, FONT=TIMES, SIZE=12, STYLE=BOLD,$
TYPE=TABFOOTING, JUSTIFY=CENTER, STYLE=BOLD, SIZE=11,$
ENDSTYLE
END
```

出力結果は次のとおりです。

Sales Quantity and Amount by Plant

Manufacturing Plant	Quantity:	Line Total
BOS	1,033,818	$262,433,960.63
DAL	390,844	$97,751,846.65
LA	229,256	$57,507,080.82
ORL	386,909	$97,616,855.94
SEA	86,680	$22,742,743.88
STL	776,743	$195,970,536.09

*****End of Report*****

見出しと脚注の特定行の選択

```
TYPE=type, LINE=line_#
```

説明

type

見出しまたは脚注のタイプを指定します。選択する値には、HEADING、FOOTING、TABHEADING、TABFOOTING、SUBHEAD、SUBFOOT があります。詳細は、見出し、脚注、タイトル、FML テキストの選択を参照してください。

line_#

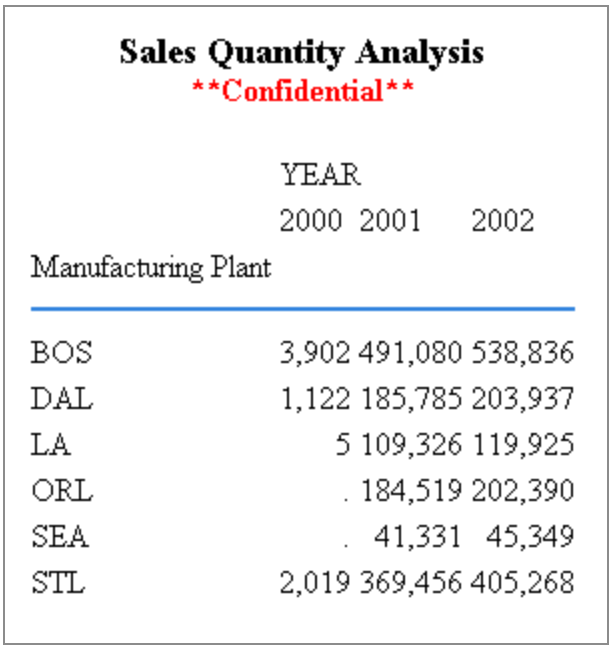
見出しまたは脚注の位置に基づいて行を選択します。

見出しの特定行の選択

次の例は、見出しの特定行のフォーマットを設定する方法を示しています。見出しの 1 行目 (Sales Quantity Analysis) は、太字のフォントサイズ 11 ポイントでフォーマットが設定されます。見出しの 2 行目 (**Confidential**) は、赤色の太字でフォーマットが設定されます。関連するスタイルシートは、リクエスト内で太字で表示されています。

```
TABLE FILE CENTORD
HEADING
"Sales Quantity Analysis"
"**Confidential**"
" "
SUM QUANTITY
ACROSS YEAR
BY PLANT
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=HEADING, LINE=1, SIZE=11, STYLE=BOLD,$
TYPE=HEADING, LINE=2, COLOR=RED, STYLE=BOLD,$
TYPE=HEADING, JUSTIFY=CENTER,$
ENDSTYLE
END
```

出力結果は次のとおりです。



Sales Quantity Analysis
****Confidential****

Manufacturing Plant	YEAR 2000	2001	2002
BOS	3,902	491,080	538,836
DAL	1,122	185,785	203,937
LA	5	109,326	119,925
ORL	.	184,519	202,390
SEA	.	41,331	45,349
STL	2,019	369,456	405,268

見出しまたは脚注テキストの選択

```
TYPE=type, [LINE=line_#], [OBJECT=TEXT], ITEM=item_#
```

説明

type

見出しまたは脚注のタイプを指定します。選択する値には、HEADING、FOOTING、TABHEADING、TABFOOTING、SUBHEAD、SUBFOOT があります。詳細は、見出し、脚注、タイトル、FML テキストの選択を参照してください。

line_#

見出しまたは脚注の位置に基づいて行を選択します。複数行の見出しまたは脚注がある場合に限り、LINE 属性を使用する必要があります。

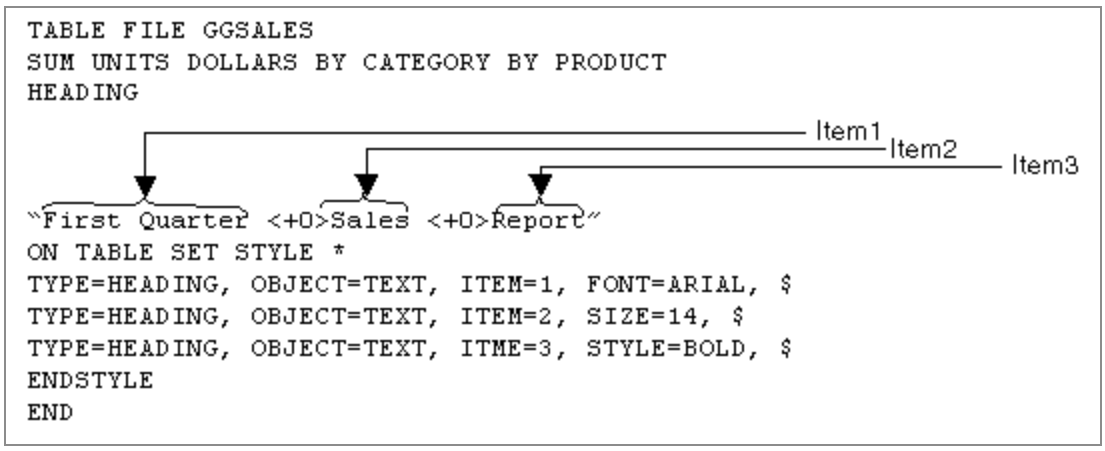
TEXT

フォーマット設定の対象となるのは、テキストおよびダイアログマネージャ変数 (&変数と表記される) に限られます。1 つの見出しまたは脚注にテキストと埋め込みフィールドの両方のスタイルを設定する場合に限り、OBJECT=TEXT を使用する必要があります。

item_#

行内の位置に基づいて項目を指定します。

1 行で表示された一連のテキストの中で、特定のテキストのみを部分的にフォーマット設定する必要がある場合は、スポットマーカーを使用して見出しまたは脚注を複数の部分に分割することができます。その場合は、指定するテキストの後にスポットマーカーを追加します。スポットマーカーの <+0> を使用しても、見出しまたは脚注にブランクは挿入されません。スポットマーカーを使用すると、テキストは次のように分割されます。



注意：終了スポットマーカーの直後に開始スポットマーカーを追加すると (><)、この 2 つのスポットマーカー間にブランクが 1 つ挿入されます (> <)。フォーマット設定を適用する際は、このことを考慮する必要があります。

OBJECT 属性を使用した場合と使用しない場合で位置の値が異なります。位置の値は次のようになります。

- OBJECT=TEXT を使用する場合は、テキストのみを左から右に数えます。
- OBJECT を使用しない場合は、テキストと埋め込みフィールド値の両方を左から右に数えます。

スポットマーカーによる見出しテキストの選択

次の例は、スポットマーカーを使用して見出しのテキストに異なるフォーマット設定を適用する方法を示しています。この例では、ブランクを追加しないスポットマーカーの <+0> を使用します。関連するスタイルシートは、リクエスト内で太字で表示されています。

```
TABLE FILE CENTORD
HEADING
```

```
"Third Quarter,<+0>2002:<+0> Sales Quantity Analysis"
SUM QUANTITY  BY PLANT
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=HEADING, OBJECT=TEXT, ITEM=1, STYLE=BOLD+UNDERLINE, SIZE=12, $
TYPE=HEADING, OBJECT=TEXT, ITEM=2, COLOR=BLUE, SIZE=12,
     STYLE=BOLD+UNDERLINE, $
TYPE=HEADING, OBJECT=TEXT, ITEM=3, STYLE=ITALIC,$
ENDSTYLE
END
```

出力結果は次のとおりです。

Third Quarter,2002: *Sales Quantity Analysis*

Manufacturing Plant	Quantity:
BOS	1,033,818
DAL	390,844
LA	229,256
ORL	386,909
SEA	86,680
STL	776,743

見出しと脚注の埋め込みフィールドの選択

```
TYPE=type, [LINE=line_#], OBJECT=FIELD, [ITEM=item #]
```

説明

type

見出しまたは脚注のタイプを指定します。選択する値には、HEADING、FOOTING、TABHEADING、TABFOOTING、SUBHEAD、SUBFOOT があります。詳細は、見出し、脚注、タイトル、FML テキストの選択を参照してください。

line_#

見出しまたは脚注の位置に基づいて行を選択します。複数行の見出しまたは脚注がある場合に限り、LINE 属性を使用する必要があります。

item_#

行内の位置に基づいて項目を指定します。

見出しまたは脚注に複数の埋め込みフィールドがある場合は、フォーマットを設定するフィールドを項目番号で指定する必要があります。項目数を左から右に数えます。テキストフィールドはこの数に含めません。見出しまたは脚注の中に埋め込みフィールドが 1 つのみ存在する場合は、項目番号を指定する必要はありません。

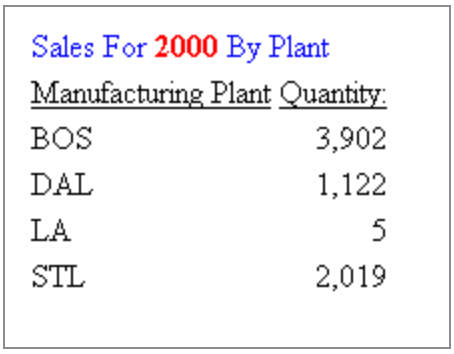
注意：BORDER オプションは、HEADING、FOOTING、SUBHEAD、SUBFOOT の特定の ITEM ではサポートされません。

見出し埋め込みフィールドの選択

次の例は、見出しの埋め込みのフィールドに、フォーマットを設定する方法を示しています。なお、見出しには埋め込みフィールドが 1 つしか存在しないため、スタイルシートで項目番号を指定していません。リクエストの太字の部分が、この例に関連するスタイルシートです。

```
TABLE FILE CENTORD
HEADING
"Sales For <YEAR By Plant"
SUM QUANTITY  BY PLANT
WHERE YEAR EQ 2000
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=HEADING, OBJECT=TEXT, COLOR=BLUE,$
TYPE=HEADING, OBJECT=FIELD, COLOR=RED, STYLE=BOLD,$
ENDSTYLE
END
```

出力結果は次のとおりです。



Sales For 2000 By Plant

Manufacturing Plant	Quantity:
BOS	3,902
DAL	1,122
LA	5
STL	2,019

WHEN 句に基づく特定の見出しまたは脚注の選択

```
TYPE = {HEADING|FOOTING}, CONDITION = n, ... ,$
```

説明

n

見出しまたは脚注で先頭から末尾までの WHEN 条件の数です。指定しない場合は、フォーマット設定はすべての見出しおよび脚注に適用されます。

WHEN による複数見出しのスタイル設定

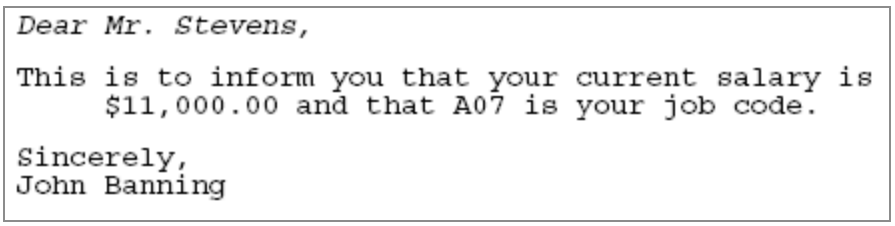
次のリクエストにより、各従業員の給与とジョブコード情報が、ページ単位で表示されます。最初の WHEN 条件は、従業員が女性の場合に適用されます。2 つ目の WHEN 条件は、従業員が男性の場合に適用されます。3 つ目の WHEN 条件は、部門が MIS の場合に適用されます。4 つ目の WHEN 条件は、部門が PRODUCTION の場合に適用されます。スタイルシートには、2 つ目と 3 つ目の条件のスタイル要素が含まれています。

```
DEFINE FILE EMPLOYEE
GENDER/A1 = DECODE FIRST_NAME(ALFRED 'M' RICHARD 'M' JOHN 'M'
  ANTHONY 'M' ROGER 'M' MARY 'F' DIANE 'F' JOAN 'F' ROSEMARIE 'F'
  BARBARA 'F');
MIXEDNAME/A15 = LCWORD(15, LAST_NAME, MIXEDNAME);
NAME/A16 = MIXEDNAME||',';
END
```

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME NOPRINT GENDER NOPRINT NAME NOPRINT
HEADING
"Dear Ms. <NAME"
   WHEN GENDER EQ 'F';
HEADING
"Dear Mr. <NAME>"
      WHEN GENDER EQ 'M';
HEADING
" "
HEADING
"This is to inform you that your current salary is "
"<CURR_SAL and <CURR_JOBCODE>is your job code."
" "
"Sincerely,"
HEADING
"Barbara Cross "
  WHEN DEPARTMENT EQ 'MIS';
HEADING
"John Banning   "
  WHEN DEPARTMENT EQ 'PRODUCTION' ;
WHERE LAST_NAME NE 'BANNING' OR 'CROSS'
BY EMP_ID NOPRINT PAGE-BREAK
ON TABLE SET PAGE NOPAGE
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=HEADING, CONDITION=2, STYLE=ITALIC,$
TYPE=HEADING, CONDITION=3, STYLE=BOLD,$
ENDSTYLE
END
```

リクエストのスタイルシートで、最初の条件で表示される見出し行には斜体を設定し、3 つ目の条件で表示される見出し行には太字を設定します。

出力で最初のページは男性従業員向けであることから、あいさつ文は斜体で表示されます。



出力で 2 ページ目は MIS 部門の従業員向けであることから、署名行は太字で表示されます。

```
Dear Ms. Smith,

This is to inform you that your current salary is
     $13,200.00 and that B14 is your job code.

Sincerely,
Barbara Cross
```

ページ番号、下線、ブランク行の選択

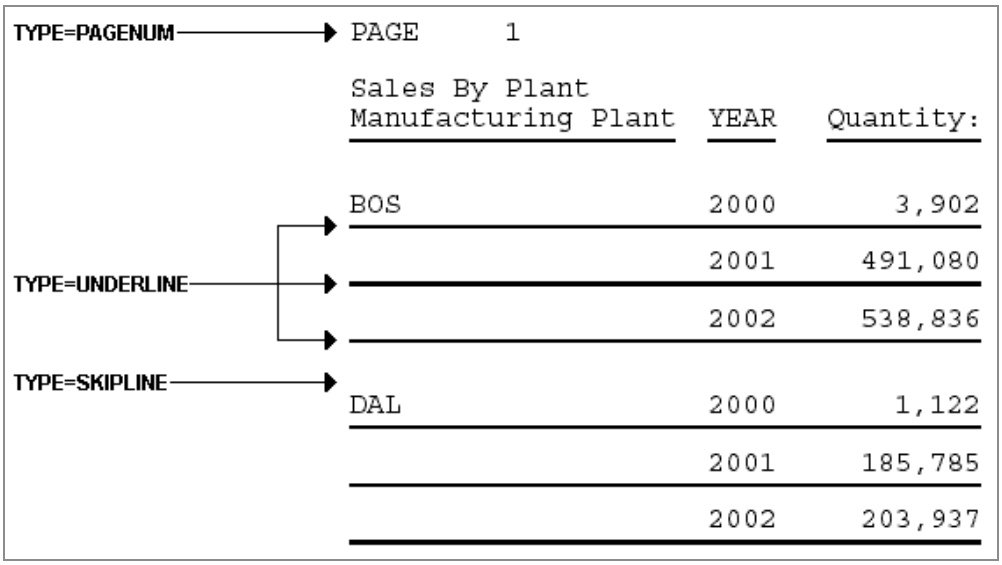
レポートのページ番号、下線、ブランク行を選択してフォーマットを設定するには、PAGENUM、SKIPLINE、UNDERLINE 属性を使用します。

なお、HTML レポートではブランク行と下線を挿入することはできますが、フォーマット設定はサポートされません。

次のコードおよび出力結果の例は、PAGENUM、UNDERLINE、SKIPLINE の各コンポーネントがレポート内に表示される位置およびこれらのコンポーネントの選択に使用する TYPE 値を示しています。

```
SET ONLINE-FMT=PDF
TABLE FILE CENTORD
HEADING
"Sales By Plant"
SUM QUANTITY
BY PLANT BY YEAR
WHERE PLANT EQ 'BOS' OR 'DAL'
ON YEAR UNDER-LINE
ON PLANT SKIP-LINE
END
```

次の出力結果は、上記のコード例に対応しています。



注意：このリクエストは、レポート内の各コンポーネントの表示位置を単に示すことが目的であるため、ここではスタイルシートは省略されています。

ページ番号、下線、ブランク行の選択

```
TYPE=type
```

説明

type

レポートコンポーネントを指定します。次のオプションから選択します。

PAGENUM - ページ番号を指定します。

SKIPLINE - ON field SKIP-LINE で生成されるブランク行を表します。HTML フォーマットのレポートではサポートされません。

UNDERLINE - FML レポートにおいて ON field UNDER-LINE または BAR で生成された下線を指定します。HTML フォーマットのレポートではサポートされません。

下線とページ番号の選択

次の例は、リクエストで下線とページ番号を選択する方法を示しています。リクエストの太字の部分が、この例に関連するスタイルシートです。

なお、HTML レポートでは下線のフォーマット設定はサポートされないため、このレポートは PDF フォーマットで作成します。

```
SET ONLINE-FMT=PDF
TABLE FILE CENTORD
HEADING
"Sales By Plant"
SUM QUANTITY
BY PLANT BY YEAR
WHERE PLANT EQ 'BOS' OR 'DAL' OR 'LA'
ON PLANT UNDER-LINE SKIP-LINE
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=HEADING, OBJECT=TEXT, COLOR=BLUE, FONT=ARIAL,$
TYPE=PAGENUM, STYLE=ITALIC, SIZE=8,$
TYPE=UNDERLINE, COLOR=RED,$
ENDSTYLE
END
```

出力結果は次のとおりです。

PAGE 1

Sales By Plant

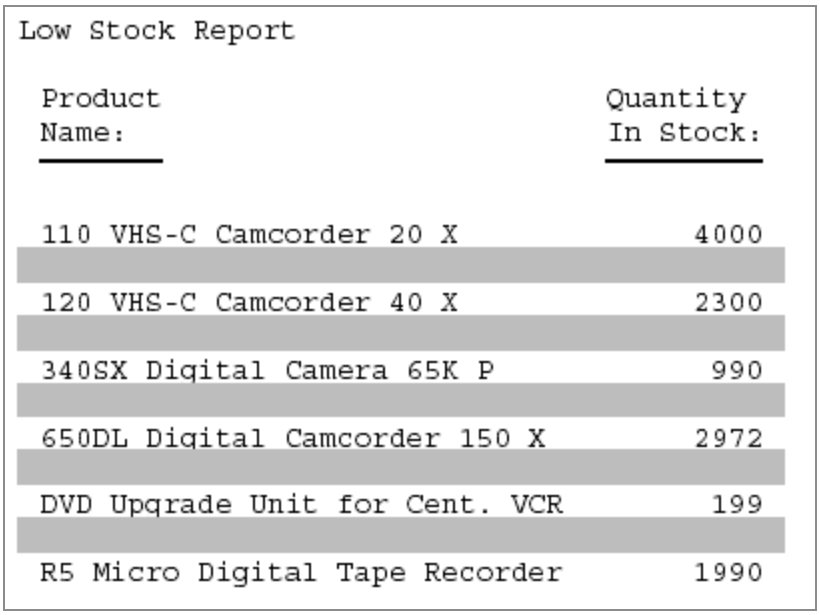
Manufacturing Plant	YEAR	Quantity:
BOS	2000	3,902
	2001	491,080
	2002	538,836
DAL	2000	1,122
	2001	185,785
	2002	203,937
LA	2000	5
	2001	109,326
	2002	119,925

ブランク行の選択

次の例は、レポートのブランク行を選択する方法を示しています。リクエストの太字の部分が、この例に関連するスタイルシートです。

```
SET ONLINE-FMT=PDF
TABLE FILE CENTINV
HEADING
"Low Stock Report"
" "
SUM QTY_IN_STOCK
WHERE QTY_IN_STOCK LT 5000
BY PRODNAME
ON PRODNAME SKIP-LINE
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=SKIPLINE, BACKCOLOR=SILVER, $
ENDSTYLE
END
```

出力結果は次のとおりです。



Low Stock Report

Product Name:	Quantity In Stock:
110 VHS-C Camcorder 20 X	4000
120 VHS-C Camcorder 40 X	2300
340SX Digital Camera 65K P	990
650DL Digital Camcorder 150 X	2972
DVD Upgrade Unit for Cent. VCR	199
R5 Micro Digital Tape Recorder	1990

外部カスケードスタイルシートの使用

カスケードスタイルシート (CSS) は、HTML ドキュメントのフォーマットを設定する標準的な方法です。外部カスケードスタイルシート (CSS) を使用して WebFOCUS レポートの HTML 出力にフォーマットを設定するには、CSS をレポートへリンクし (WebFOCUS スタイルシートの CSSURL 属性の使用)、必要に応じて CSS のクラスを特定のレポートコンポーネントに適用します (CLASS 属性の使用)。

カスケードスタイルシートの概要

カスケードスタイルシート (CSS) は、HTML ドキュメントへのフォーマットの指定を可能にする HTML の拡張機能です。WebFOCUS では、次の 2 種類の CSS を使用することができます。

- **内部カスケードスタイルシート**　フォーマットを設定する HTML ドキュメントに内部格納されるカスケードスタイルシートです。

- **外部カスケードスタイルシート**　複数のドキュメントで共有可能な個別のファイルに格納されるカスケードスタイルシートです。外部 CCS ファイルは、ブラウザにアクセス可能な任意の Web サーバ上に配置することができます。このファイルの格納先は、WebFOCUS スタイルシート CSSURL 属性、CSSURL SET パラメータ、LINK 要素 (特別な場合) のいずれかを使用して指定します。

  カスケードスタイルシート内にクラスを定義し、いずれかの CSS クラスを適用することにより、レポートコンポーネントにフォーマットを設定します。

カスケードスタイルシートは、複数の異なるスタイルシートを 1 つのドキュメントに対して同時に使用することができるため、「カスケード」と呼ばれます。たとえば、1 つ目のスタイルシートをドキュメント自体に関連付け、2 つ目のスタイルシートを 1 つ目のスタイルシートにリンクさせ、さらに 3 つ目のスタイルシートをドキュメントの表示先となるブラウザに関連付けることができます。複数のスタイルシートが有効な場合、これらのスタイルシートは、ブラウザに設定された定義済みの順序でドキュメントに適用されます。これらのフォーマット設定の流れは、スタイルシート間で連続すると見なされます。

外部カスケードスタイルシートの使用は、次の場合に適しています。

- **レポートの開発**　レポートのフォーマット設定のためのより優れた手段としてレポートの開発に使用することができます。
- **ガイドライン担当**　既存のカスケードスタイルシートをレポートに適用することができるため、Web ドキュメント外観のガイドラインの決定に使用することができます。

CSS の詳細　この WebFOCUS マニュアルでは、CSS の知識があることを前提にしています。CSS についての説明はこのマニュアルの対象外ですが、情報のソースが多く記述されています。オンラインでは、カスケードスタイルシートのホームページ (http://www.w3.org/Style/CSS/) を参照してください。

カスケードスタイルシートのルールおよびクラス

カスケードスタイルシート (CSS) は、「ルール」と呼ばれるステートメント内にフォーマットを定義します。以下は、HTML ページの本文の背景色を黄色にする単純なルールの例です。

```
BODY {background: yellow}
```

各ルールには、セレクタ (この場合、BODY) と宣言 (この場合、background: yellow) が 1 つずつ含まれています。宣言には、プロパティ (background) とプロパティに割り当てられた値 (yellow) が含まれています。

宣言によりフォーマットが定義され、セレクタにより適用されるフォーマットが決定されます。セレクタには、任意の HTML 要素を使用できます。セレクタには、クラスも使用できます。クラスは、それに対するルールを作成することにより定義することができます。要素のクラスにルールを作成することにより、1 つの要素に異なるフォーマットを定義することができます。

たとえば、レポートのソートフィールド、集計フィールド、詳細フィールドのそれぞれに表示されるテキストに異なる色を適用することを想定します。これには、BODY 要素に 3 つのクラス (sortColumn、aggregateColumn、detailColumn) を作成します。

```
BODY.sortColumn {color: blue}
BODY.aggregateColumn {color: green}
BODY.detailColumn {color: black}
```

1 つの要素に制限されない汎用クラスを定義することもできます。以下はその例です。

```
.pageFooting {font-weight: bolder}
```

汎用クラスを使用することにより、WebFOCUS の任意のレポートコンポーネントにフォーマットを指定することができます。

外部カスケードスタイルシートを使用する理由

レポートのフォーマット設定に WebFOCUS スタイルシートを使用している場合は、これらのスタイルシートを外部カスケードスタイルシートと組み合わせることにより、次の利点が得られます。

- **フォーマットオプションの拡張**　カスケードスタイルシートに指定できるフォーマットの大部分は、レポートに適用することができます。これにより、ネイティブ WebFOCUS スタイルシート属性では使用できないフォーマットオプションを利用することができます。たとえば、ブラウザベースの測定機能を使用することにより、CSS の相対測定単位である「em」に指定されたサイズを持つフォントサイズ、マージン、およびその他の要素のサイズをレポートを参照するユーザが制御することができます。また、CSS を使用して行の高さと文字間隔を制御したり、レポート内での項目の配置を制御したりすることができます。
- **パフォーマンスの向上**　カスケードスタイルシートにより、WebFOCUS は HTML 出力を容易に生成することができます。これにより、ブラウザへのレポートの送信にネットワークが使用する情報量が少なくなり、ダウンロード時間が短縮されます。
- **工数の削減**　カスケードスタイルシートをすでに導入している企業は、WebFOCUS レポートにもカスケードスタイルシートを適用することができます。これにより、フォーマットの指定と保守にかかる工数を削減することができます。
- **標準化**　WebFOCUS スタイルシートから複製を作成する代わりに、1 つのカスケードスタイルシートセットにすべての Web ドキュメントフォーマットを指定することができるため、企業でのレポートの外観の標準化が容易になります。

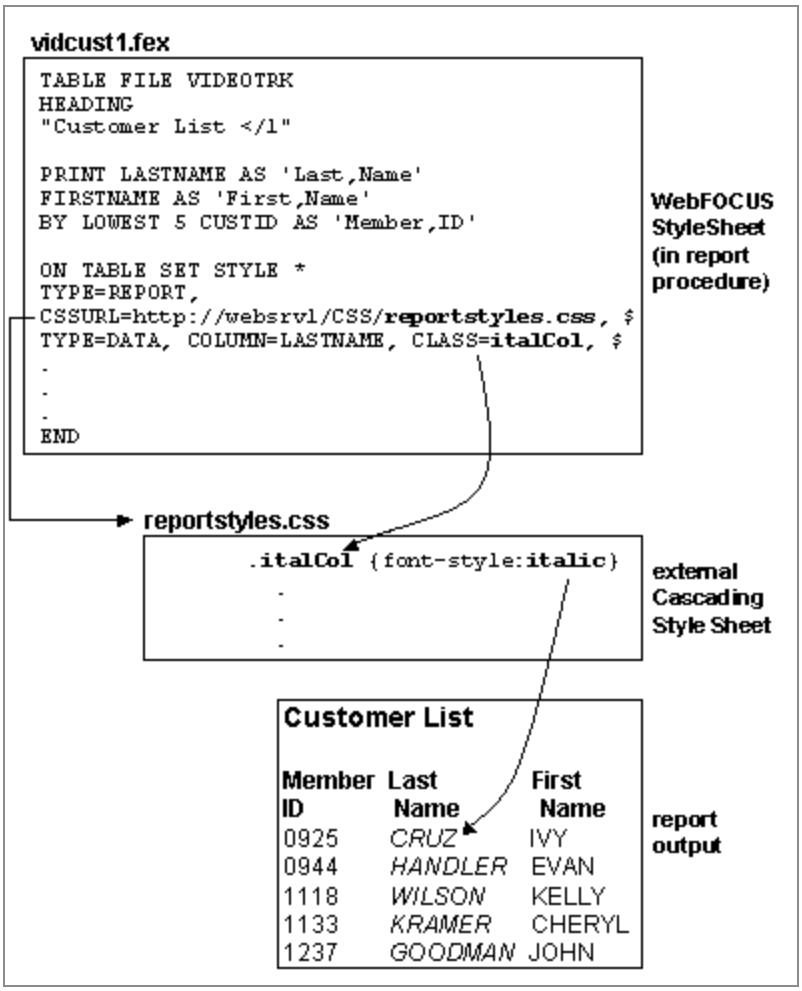
外部カスケードスタイルシートによるレポートフォーマット設定

外部カスケードスタイルシート (CSS) によるレポートのフォーマット設定には、次の 3 つの項目が必要です。

- **外部カスケードスタイルシート**　レポートに適用するフォーマットを指定します。
- **WebFOCUS スタイルシート**　外部カスケードスタイルシートを適用する際に、レポートの特定のコンポーネントにフォーマットを適用します。ただし、レポート全体にフォーマットを適用する場合、WebFOCUS スタイルシートは必要ありません。

  WebFOCUS スタイルシートを使用して、外部カスケードスタイルシート外でフォーマットを指定することは可能ですが、制限事項があります。
- **外部カスケードスタイルシートへのリンク**　レポートを外部カスケードスタイルシートにリンクします。

これらの項目の動作の連携についての詳細は、下図の例を参照してください。



外部カスケードスタイルシートを使用してレポートにフォーマットを設定するには

外部カスケードスタイルシート (CSS) を使用してレポートにフォーマットを設定するには、次の手順を実行します。

手順

1. **CSS でのレポートフォーマットの指定**　使用するカスケードスタイルシート、レポートのデフォルトフォーマットを指定するルール、レポートコンポーネントへの適用に最適なクラスを決定します。また、新しい CSS を作成している場合、または既存の CSS を拡張している場合は、新しいルールを定義して、レポートのフォーマットを指

定します。フォーマットの指定は、指定先によって次のようになります。

- **レポートコンポーネント**　ルールは、任意の汎用クラス (要素に対して宣言されていないクラス) に使用できます。

  グラフは、その他のレポートタイプと異なります。グラフが表示されるページ、またはグラフの見出しおよび脚注にフォーマットを指定するには、BODY 要素のルールを使用することができますが、クラスのルールは使用できません。その他のグラフコンポーネントにフォーマットを設定することはできません。

- **レポート全体**　BODY または TD 要素のルール (これらの要素のクラスではありません) を使用し、手順 2 を省略します。これは、レポートのデフォルトフォーマットを指定する上で効果的な方法であり、CSS クラスをレポート全体に適用するよりも効率よくレポートを出力することができます。

  グラフは、その他のレポートタイプと異なります。BODY 要素のルールは使用できますが、TD 要素のルールは使用できません。BODY のルールは、グラフが表示されるページとその見出しおよび脚注にフォーマットを設定しますが、グラフ自体にはフォーマットを設定しません。

2. **レポートコンポーネントへのクラスの割り当て**WebFOCUS スタイルシートでは、フォーマットを設定するレポートコンポーネントそれぞれにカスケードスタイルシートを割り当てます。CLASS 属性を使用してクラスを指定します。それぞれのコンポーネントに異なるクラスを割り当てることができます。また、1 つのクラスを複数のコンポーネントに割り当てることもできます。

3. **CSS へのリンク**　CSS ファイルの URL を CSSURL WebFOCUS スタイルシート属性、または CSSURL SET パラメータに割り当てることにより、外部カスケードスタイルシートへのリンクを作成します。

   例外として、-HTMLFORM コマンドを使用して既存の HTML ページにレポート出力を埋め込む場合は、CSSURL は設定せずに、この HTML ページに LINK 要素を含めます。

フィールドタイトルクラスの CSS ルール

このカスケードスタイルシート (CSS) ルールは、ColumnTitle 汎用クラス (要素に関連付けられていないクラス) を次のように宣言します。

```
.ColumnTitle {font-family:helvetica; font-weight:bold; color:blue;}
```

TD 要素の CSS ルール

TD 要素のカスケードスタイルシート (CSS) ルールは、要素のフォントファミリーを次のように指定します。

```
TD {font-family:helvetica}
```

これは TD 要素のルールであるため、そのフォーマットはレポートコンポーネントだけではなく、レポート全体に適用されます。

レポート内の ACROSS 値への CSS クラスの適用

この WebFOCUS スタイルシートは、ColumnTitle クラスで指定されたフォーマットを適用することにより、ACROSS 値にフォーマットを設定します。

```
TYPE=AcrossValue, CLASS=ColumnTitle, $
```

レポートスタイルの外部カスケードスタイルシートへのリンク

この WebFOCUS スタイルシートは、レポートスタイルの外部カスケードスタイルシートへのリンクを作成します。

```
TYPE=REPORT, CSSURL=http://webserv1/css/reportstyles.css
```

または

```
TYPE=REPORT, CSSURL=IBFS:/WFC/Repository/css/reportstyles.css
```

これは、SET コマンドを使用しても実行可能です。

```
SET CSSURL=http://webserv1/css/reportstyles.css
```

リクエスト内では、次のように指定します。

```
ON TABLE SET CSSURL=http://webserv1/css/reportstyles.css
```

-HTMLFORM を使用して既存の HTML ページにレポート出力を埋め込む場合、CSSURL を設定する代わりに、レポートの埋め込み先の HTML ページに LINK 要素をコーディングしてリンクを指定します。

```
<HEAD>
<TITLE>Accounts Receivable Report</TITLE>
<LINK REL="STYLESHEET" HREF="http://srv3/css/reports.css"
TYPE="text/css">
</HEAD>
```

外部カスケードスタイルシート (CSS) によるレポートフォーマット

次のレポートは、Gotham Grinds から現在提供されている製品を表示しており、そのフォーマット設定には外部カスケードスタイルシート (CSS) が使用されています。レポートのフォーマットは次のとおりです。

- デフォルトのフォントファミリーは Arial です。
- レポートの見出しのフォントファミリーが Times New Roman で上書きされます。また、見出しのフォントサイズは大きく、中央揃えです。
- すべてのフィールドタイトルは太字で、その背景色は水色です。
- 製品の単価が $27 未満の場合、製品の行は緑色の斜体で表示されます。

リクエストおよびインライン WebFOCUS スタイルシートは、curprods.fex プロシジャでは次のように表されます。プロシジャの後には、「report01.css」という名前の外部カスケードスタイルシートが示されています。

curprods.fex

```
    TABLE FILE GGPRODS
    HEADING
    "</1 Current Products</1"
    PRINT PRODUCT_DESCRIPTION UNIT_PRICE
    BY PRODUCT_ID
    ON TABLE SET PAGE-NUM OFF

ON TABLE SET STYLE *
TYPE=REPORT, CSSURL=http://websrv2/css/report01.css, $
TYPE=HEADING, CLASS=headText, $
TYPE=TITLE, CLASS=reportTitles, $
TYPE=DATA, CLASS=lowCost, WHEN=N3 LT 27, $
ENDSTYLE
    END
```

注意：report01.css ファイルの格納先 WebFOCUS リポジトリパスを指定するには、リクエストの TYPE=REPORT の CSSURL パラメータで、次の構文を使用します。

```
TYPE=REPORT, CSSURL=IBFS:/WFC/Repository/css/report01.css, $
```

ここで、「CSS」は WebFOCUS リポジトリで report01.css が格納されているフォルダを示します。

report01.css

```
BODY  {font-family:Arial, sans-serif}
TABLE {border:0}
TD    {border:0}
 .reportTitles {font-weight:bolder; background:lightblue;}
.lowCost {color:green; font-style:italic;}
.headText {font-family:Times New Roman, serif; font-size:larger;
              text-align:center}
```

1. インライン WebFOCUS スタイルシートを開始します。
2. 外部カスケードスタイルシート report01.css にリンクします。
3. headText クラスの CSS ルールを使用してレポートの見出しにフォーマットを設定します。
4. reportTitles クラスの CSS ルールを使用して、レポートのフィールドタイトルにフォーマットを設定します。

5. 製品の単価が $27 未満の各レポート行に、lowCost クラスの CSS ルールを使用したフォーマットを設定します。
6. インライン WebFOCUS を終了します。
7. この BODY 要素の CSS ルールは、フォントファミリーに Arial を指定します。Arial フォントが利用できない場合は、フォントファミリー sans serif を指定します。

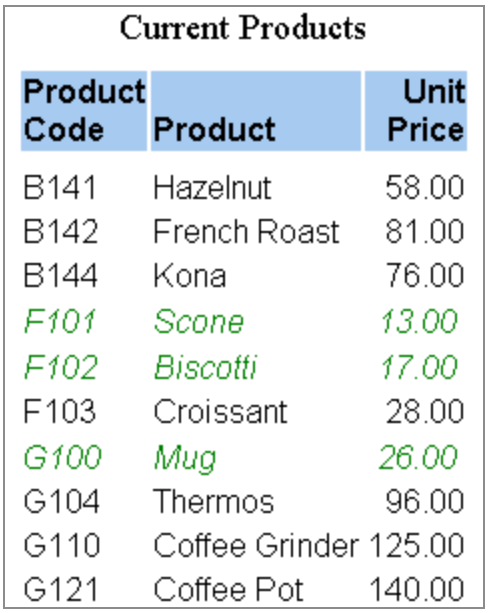
   このルールは BODY のルールであるため、レポート全体に適用されます。レポートのすべてのテキストには、デフォルト値の Arial が指定されます。前述のプロシジャで headText クラスルール (11 を参照) で設定が上書きされるよう、特定のレポートコンポーネントにそのコンポーネントの汎用クラスルールを適用することによりこの設定を上書きすることができます。
8. TABLE 要素および TD 要素の CSS ルールは、レポートのデフォルト罫線を削除します。
9. 汎用クラス reportTitles の CSS ルールは、フォントを相対的に太くし、背景色を水色に指定します。

   WebFOCUS スタイルシートは、これをレポートのフィールドタイトル (4 を参照) に適用します。
10. 汎用クラス lowCost の CSS ルールは、テキスト色を緑色に、フォントスタイルを斜体に変更します。

    WebFOCUS スタイルシートは、製品の単価が $27 未満のレポート行にこのルールを適用します (5 を参照)。
11. 汎用クラス headText の CSS ルールは、フォントファミリーに Times New Roman を指定します。Times New Roman が使用できない場合は、フォントファミリー Serif を指定します。また、フォントサイズを大きくし、中央揃えにします。

    WebFOCUS スタイルシートは、このルールをレポートの見出しに適用します。BODY 要素のルールに指定されたデフォルトのフォントファミリーを上書きします (7 を参照)。

プロシジャは、レポートに次のフォーマットを設定しています。



Current Products

Product Code	Product	Unit Price
B141	Hazelnut	58.00
B142	French Roast	81.00
B144	Kona	76.00
F101	*Scone*	*13.00*
F102	*Biscotti*	*17.00*
F103	Croissant	28.00
G100	*Mug*	*26.00*
G104	Thermos	96.00
G110	Coffee Grinder	125.00
G121	Coffee Pot	140.00

外部カスケードスタイルシートの使用

レポートのフォーマット設定に外部カスケードスタイルを使用する場合、次の事項を理解しておく必要があります。

- 既存または新しい外部カスケードスタイルシート (CSS) の選択。
- 外部カスケードスタイルシート (CSS) の保存先。
- 1 つのレポートへの複数のカスケードスタイルシートの適用。
- 外部カスケードスタイルシート (CSS) の編集。
- レポートフォーマットの指定に使用する CSS ルールおよびクラス。
- カスケードスタイルシートクラス名の指定。
- WebFOCUS スタイルシートや TABLE 言語指定などのフォーマット設定方法と外部カスケードスタイルシート (CSS) との組み合わせ。

外部カスケードスタイルシートの選択

外部カスケードスタイルシート (CSS) を使用してレポートにフォーマットを設定する際、以下を選択することができます。

- **既存の CSS を変更せずに適用**　外部カスケードスタイルシートは、その他のドキュメントに使用することができ、レポートにフォーマットを設定するルール以外のルールを含めることもできます。たとえば、CSS には、WebFOCUS アプリケーションにより使用される Web ページ内のその他の要素にフォーマットを設定するルールや、その他の Web ページのルールを含めることができます。これにより、1 つのカスケードスタイルシートですべての Web ドキュメントにフォーマットを設定することができます。
- **既存の CSS の編集**　ルールの追加または変更に使用します。たとえば、カスケードスタイルシートを編集し、レポートコンポーネントにフォーマットを設定する新しい汎用クラスを追加することができます。
- **新規 CSS の作成**　新しいカスケードスタイルシートを作成し、レポートにフォーマットを設定することができます。

外部カスケードスタイルシートを作成するには、テキストエディタまたは他社製 Web 開発ツールを使用します。

外部カスケードスタイルシートの保存先

外部カスケードスタイルシート (CSS) は、任意の Web サーバプラットフォームに格納することができます。ただし、CSSURL (スタイルシート属性または SET パラメータ) で相対 URL を指定する場合、カスケードスタイルシートは、WebFOCUS で使用される Web サーバに保存する必要があります。

複数外部カスケードスタイルシートの使用

各レポートプロシジャは、それぞれを 1 つの外部カスケードスタイルシート (CSS) にリンクできますが、複数の CSS をインポートする CSS にプロシジャをリンクすることにより、複数のカスケードスタイルシートを使用することができます。CSS 間のインポートについての詳細は、使用する CSS のマニュアルを参照してください。

外部カスケードスタイルシートの編集

テキストエディタまたは他社製 Web 開発ツールを使用することにより、外部カスケードスタイルシート (CSS) を編集することができます。

レポートのフォーマットがカスケードスタイルシートのみで指定されている場合、CSS を編集した後に別のユーザがレポートを表示すると、レポートを再実行することなく、レポートに CSS への変更内容が反映されます。

ただし、レポートに変更内容が反映されない場合、Web ブラウザがキャッシュに格納された旧バージョンの CSS を使用している可能性があります。レポートを表示しているユーザは、Web サーバから CSS ファイルを再ロードする必要があります。これを行うには、Microsoft Internet Explorer でブラウザの [更新] ボタンをクリックして、レポートのフォーマット設定に最新バージョンの CSS が使用されるようにします。

カスケードスタイルシートルールの選択

異なるタイプのルールを使用してレポートの異なる部分にフォーマットを設定することができます。

フォーマットの対象	使用するルール
レポート全体	BODY または TD
任意のレポートコンポーネント	汎用クラス (宣言された要素を含まないクラス)

BODY または TD のルールを選択する際は、次の点に注意する必要があります。

- **BODY**　レポートが表示される Web ページ全体 (レポート自体も含む) にデフォルトのフォーマットを指定します。これは、すべての CSS 動作のように、各ユーザの Web ブラウザにより実装され、ブラウザに依存する CSS 継承に依存することに注意してください。

  グラフはその他のタイプのレポートとは異なります。BODY のルールは、グラフが表示されるページとその見出しおよび脚注にフォーマットを設定しますが、グラフ自体にはフォーマットを設定しません。

- **TD**　レポート、およびページ上に含まれるテーブルセルのみにデフォルトのフォーマットを指定します。TD は、テーブルデータ (テーブルセル) 要素です。WebFOCUS は、ほとんどの HTML レポート出力を、個別のセル内にそれぞれのレポート項目を配置する HTML テーブルとして生成します。これにより、TD のルールを使用してレポート全体にフォーマットを設定することができます。

  グラフはその他のタイプのレポートとは異なります。グラフのデフォルトフォーマットを指定するには、TD のルールではなく、BODY のルールを使用します。BODY の

ルールを使用したグラフのフォーマット設定については、上記の説明を参照してください。

レポートコンポーネントのフォーマット設定にクラスのルールを使用するには、CLASS 属性を使用して WebFOCUS スタイルシート内のコンポーネントにクラスを割り当てる必要があります。

1 つのレポートコンポーネントに複数の CSS プロパティを適用する場合、これらのプロパティを 1 つのクラス内で宣言することをお勧めします。これにより、1 つのプロパティを 1 つのクラスで宣言するよりも効率よく出力を生成することができます。

各カスケードスタイルシートの所有者は、その CSS 内でレポートのフォーマット設定に使用できるクラスの一覧にアクセス可能な状態にしておくことをお勧めします。これにより、レポートの開発者が適切なクラスを選択することができます。

ルールの例についての詳細は、次のとおりです。

カスケードスタイルシートクラス名の指定

新しいクラスに名前を指定する際は、大文字と小文字が区別される場合があることに注意してください (Web ブラウザにより異なる) 。

新しいクラスを作成する際は、レポートコンポーネントに適用する外観ではなく、機能に基づいた名前を指定することをお勧めします。これにより、レポートコンポーネントの外観を変更した場合にも、名前と意味の整合性が取れます。たとえば、すべてのレポートタイトルを赤色にするには、タイトルにフォーマットを設定するために宣言するクラス名には「Red」ではなく「Title」と指定します。

外部カスケードスタイルシートフォーマットの適用

外部カスケードスタイルシート (CSS) フォーマットの適用対象は次のとおりです。

- **レポートコンポーネント**　WebFOCUS スタイルシート CLASS 属性を使用して、レポートコンポーネントにカスケードスタイルシートクラスを割り当てます (例、フィールドを斜体で表示)。

  表形式または自由形式レポートにフォーマットを設定する際は、CSS クラスを割り当

てることにより、任意のレポートコンポーネントにフォーマットを設定することができます。

グラフレポートにフォーマットを設定する際は、グラフの見出しおよび脚注にフォーマットを設定し、グラフが表示されるページの背景色および背景イメージを BODY 要素のルールに指定することができます。グラフの見出しと脚注にフォーマットを設定する際は、個別の行、文字列、およびフィールド値にはフォーマットを設定することができないことに注意してください。グラフの見出しと脚注を中央揃えにする場合、スタイルシート内ではなく、HEADING または FOOTING コマンドの CENTER オプションを使用することをお勧めします。

WebFOCUS スタイルシートの CLASS 属性を操作する場合は、WebFOCUS テキストエディタで WebFOCUS スタイルシートを編集する必要があります。

- **レポート全体**　外部カスケードスタイルシート (CSS) 内のフォーマットを BODY または TD 要素のルールに指定します (例、レポート全体を斜体で表示)。グラフでは、BODY 要素内のルールにのみフォーマットを指定します。これにより、グラフが表示されるページと、グラフの見出しおよび脚注にフォーマットが設定されますが、グラフ自体にはフォーマットは設定されません。また、レポートを CSS にリンクする必要があります。要素のクラスにはルールは必要なく、WebFOCUS スタイルシート宣言も必要ありません。

レポートのフォーマット設定に外部カスケードスタイルシートを使用する際は、内部カスケードスタイルシートを生成しない限り、レポートフォーマットの指定に WebFOCUS スタイルシートを使用しないことをお勧めします。

CLASS 属性を使用した CSS フォーマットの適用

外部カスケードスタイルシート (CSS) クラスをレポートコンポーネントに適用するには、WebFOCUS スタイルシートに次の構文を使用します。

```
TYPE = type, [subtype,] CLASS = classname, [when,] [link,] $
```

説明

type

クラスのフォーマットを適用するレポートコンポーネントを識別します。表形式および自由形式レポートでは、任意のコンポーネントを指定することができます。詳細は、

WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。グラフレポートのコンポーネントを指定することはできません。グラフの見出しと脚注、およびグラフが表示されるページの背景色と背景画像にフォーマットを設定するには、WebFOCUS スタイルシート宣言は使用せずに、BODY 要素のルールを使用します。

それぞれのレポートコンポーネントは、1 つのクラスでフォーマットを設定することができます。レポートコンポーネントに複数のクラスを指定すると、以下が実行されます。

1. 条件付きフォーマットを含むスタイルシート宣言内のクラスが最初に評価されます。レポートコンポーネントの各セルでは、セルの行により条件が満たされた 1 つ目のクラスがセルに割り当てられます。
2. 条件が満たされない場合、または条件付き宣言がない場合、条件なしの最初のスタイルシート宣言内のクラスがレポートコンポーネントに適用されます。このコンポーネントでは、次の宣言がすべて無視されます。

subtype

一部のレポートコンポーネントを完全に指定するために必要なオプションの属性および値です。たとえば、特定のレポートフィールドの指定には、COLUMN およびフィールド識別子が必要です。

classname

レポートコンポーネントに適用するフォーマットが含まれたカスケードスタイルシートのクラス名です。1 つのクラスを複数のレポートコンポーネントに割り当てることができます。

クラス名の最大長は 511 バイトで、大文字と小文字が区別されます。そのため、外部カスケードスタイルシートのクラスルールで指定されたとおりにクラス名を指定する必要があります。ただし、Web ブラウザによっては、大文字と小文字を区別する必要がない場合もあります。

when

オプションの WHEN 属性および値です。これは、フォーマットを条件付きで適用する場合に使用します。

link

オプションのプロシジャ、URL、JAVASCRIPT 属性のいずれかと、値を指定します。これは、ドリルダウン可能なレポートなど、レポートコンポーネントを別のリソースにリンクする場合に使用します。

外部カスケードスタイルシート (CSS) とその他のフォーマット設定方法の組み合わせ

レポートのフォーマット設定に外部カスケードスタイルシート (CSS) を使用する際は、別のフォーマット設定方法を同時に使用することができますが、これらの方法には制限が適用される場合があります。外部カスケードスタイルシート (CSS) とともに使用可能な方法には、次のものがあります。

- **WebFOCUS スタイルシート**　外部カスケードスタイルシート (CSS) と WebFOCUS スタイルシートを効果的に組み合わせるには、外部カスケードスタイルシート (CSS) へのリンクを作成してデフォルトフォーマットを提供し、WebFOCUS スタイルシートを使用して個別のレポートコンポーネントのデフォルト値を上書きします。WebFOCUS

  スタイルシートと外部カスケードスタイルシートを組み合わせる場合、外部カスケードスタイルシート (CSS) に関連したパフォーマンスの低下を回避するためには、内部カスケードスタイルシートを生成します。

  外部カスケードスタイルシート (CSS) クラス (CLASS 属性) と WebFOCUS スタイルシート属性のフォーマットは競合する可能性があるため、1 つのレポートコンポーネントの同一プロパティのフォーマット設定にはこれら 2 種類の属性を使用しないでください。

- **TABLE 言語の指示**　HEADING CENTER、PAGE-BREAK、スポットマーカー (例、</3) などの TABLE 言語 (または GRAPH 言語) の指示を使用して、フォーマットを設定することができます。ただし、TABLE (または GRAPH) 言語の指示と外部カスケードスタイルシートルールは競合する場合があるため、1 つのレポートコンポーネントの同一プロパティのフォーマット設定にはこれらを同時に適用しないでください。

  たとえば、同一レポート内で次の両方の指示を同時に指定しないでください。

  - リクエスト内の HEADING CENTER
  - レポートのページ見出しに適用する、外部カスケードスタイルシート (CSS) でのテキストの位置揃え

  これらはいずれも、レポートのページ見出しの整列を行います。

外部カスケードスタイルシートと WebFOCUS スタイルシートの組み合わせ

レポートにフォーマットを設定する際は、外部カスケードスタイルシート (CSS) と WebFOCUS スタイルシートを併用することができます。内部カスケードスタイルシートを生成するかどうかに関係なく、この方法を使用することができます。

効果的な作業方法　外部カスケードスタイルシート (CSS) へのリンクを作成してデフォルトのフォーマットを指定します。WebFOCUS スタイルシートを使用して個別のレポートコンポーネントのデフォルト値を上書きします。カスケードスタイルシートの BODY または TD ルールは、レポートのデフォルトフォーマットを提供します。必要に応じて、ネイティブ WebFOCUS スタイルシート属性を使用して個別のレポートコンポーネントのデフォルト値を上書きすることができます。これにより、WebFOCUS スタイルシート属性を使用する WebFOCUS レポートのフォーマット標準をカスタマイズできると同時に、CSS を使用する組織のフォーマット標準に準拠することができます。

パフォーマンスの考慮事項　WebFOCUS スタイルシートから内部カスケードスタイルシートを生成しない限り、外部カスケードスタイルシート (CSS) と WebFOCUS スタイルシートの組み合わせは、外部カスケードスタイルシート (CSS) に関連したパフォーマンスに影響する場合があります。これは、外部カスケードスタイルシート (CSS) とネイティブ WebFOCUS スタイルシート属性の両方を使用したレポートでは、生成される HTML コード量が、ネイティブ WebFOCUS スタイルシート属性のみを使用した場合よりも少ないが、外部カスケードスタイルシート (CSS) のみを使用した場合よりも多いためです。生成される HTML コード量を減らすことで、ネットワークの負荷を低減し、ブラウザの表示にかかる時間を短縮することができます。

二重フォーマットの回避　外部カスケードスタイルシート (CSS) クラス (CLASS 属性) と WebFOCUS スタイルシート属性のフォーマットは競合する可能性があるため、1 つのレポートコンポーネントの同一プロパティのフォーマット設定に、これら 2 種類の属性を併用することは避けてください。

たとえば、次の 2 つの宣言を同一スタイルシートには含めないでください。これらの宣言の両方が、Country フィールドに色を割り当てようとするためです。

```
TYPE=Data, COLUMN=Country, COLOR=Orange, $
TYPE=Data, CLASS=TextColor, $
```

1 つのレポートコンポーネントの異なるプロパティにフォーマットを設定したり、異なるレポートコンポーネントにフォーマットを設定したりする場合は、クラスと WebFOCUS スタイルシート属性を使い分けてフォーマットを指定することができます。たとえば、次の宣言は、同一スタイルシートで受け入れ可能です。

```
1.TYPE=Heading, COLOR=Green, $
1.TYPE=Heading, CLASS=HeadingFontSize, $
2.TYPE=Data, Column=Country, BACKCOLOR=Yellow, $
2.TYPE=Data, Column=Car, CLASS=DataBackgroundColor, $  3.TYPE=Data, Column=Model,
FOCEXEC=NewSales(CarGroup=Car), $
```

1. これらの 2 つの宣言は、異なるプロパティ (色とフォントサイズ) にフォーマットを設定するため競合しません。
2. これらの 2 つの宣言は、異なるレポートコンポーネント (Country フィールドと Car フィールド) にフォーマットを設定するため競合しません。
3. この宣言は、レポートコンポーネントにフォーマットを設定することなくハイパーリンクを定義するため、すべての CSS クラスと競合しません。

外部カスケードスタイルシートへのリンク

外部カスケードスタイルシート (CSS) を使用してレポートにフォーマットを設定するには、次のいずれかの方法で、カスケードスタイルシートをレポートにリンクする必要があります。

- **レポートプロシジャの大部分**　CSS ファイルの URL を CSSURL 属性またはパラメータに割り当てます。
- **埋め込まれるレポート出力**　-HTMLFORM コマンドにより既存の HTML ページにレポート出力を埋め込む場合、既存の HTML ページ内に外部カスケードスタイルシート (CSS) ファイルを指定する LINK 要素を記述します。たとえば、HTMTABLE フォーマットで生成された出力 (HOLD) ファイルからのレポートを埋め込んでいる場合は、LINK を使用します。詳細は、使用する CSS のマニュアルを参照してください。

CSSURL 属性およびパラメータの使用

CSSURL WebFOCUS スタイルシート属性または CSSURL SET パラメータを使用して、外部カスケードスタイルシート (CSS) をレポートにリンクすることができます。スタイルシート属性とSET パラメータの選択では、次の利点が得られます。

- **属性**　CSSURL をスタイルシート属性として使用することにより、以下を指定することができます。

    - **より長い URL**　URL の長さの最大値は、属性では 255 バイトで、パラメータでは 69 バイトです。
    - **フォーマット情報の集約**　外部カスケードスタイルシート (CSS) へのリンクおよび CSS クラスへの参照が WebFOCUS スタイルシート内に指定されます。これにより、フォーマットロジックの管理が容易になります。
- **SET パラメータ**　CSSURL を SET パラメータとして使用すると、複数のレポートで使用されているリンク先を一括変更することができます (特定の CSS から別の CSS へ)。これを行うには、SET CSSURL コマンドをそのプロシジャ内に記述し、各レポートプロシジャ内の -INCLUDE ダイアログマネージャステートメントを使用してレポートプロシジャに結合します。

CSSURL をいくつかの方法で指定すると、ローカル範囲が最も狭い指定が優先されます。優先順位 (最高が 1 で最低が 3) は、次のとおりです。

CSS パラメータについての詳細は、Use the CSSURL Parameter to Link to an External CSS を参照してください。

CSSURL 属性を使用した外部カスケードスタイルシート (CSS) へのリンク

WebFOCUS スタイルシート属性を使用して外部カスケードスタイルシート (CSS) をレポートにリンクさせるには、次の構文を使用します。

```
[TYPE=REPORT,] CSSURL={url|ibfs}, $
```

説明

TYPE=REPORT

この属性をレポート全体に適用します。この指定を省略した場合、スタイルシート宣言にはこのデフォルト値が使用されます。

url

外部カスケードスタイルシートの URL です。外部カスケードスタイルシート (CSS) が大文字と小文字を区別する Web サーバプラットフォーム上に存在する場合、大文字と小文字を正しく使用して指定する必要があります。

URL には 255 バイトまで指定することができます。外部カスケードスタイルシートの URL がこの制限を超過する場合、パスの一部を表すエイリアス (「仮想ディレクトリ」とも呼ばれる) を Web サーバで定義することにより URL を短縮することができます。

URL は、絶対パスまたは相対パスで指定することができます。相対パスを使用する場合、外部カスケードスタイルシート (CSS) は WebFOCUS で使用される Web サーバ上に存在する必要があります。

ibfs

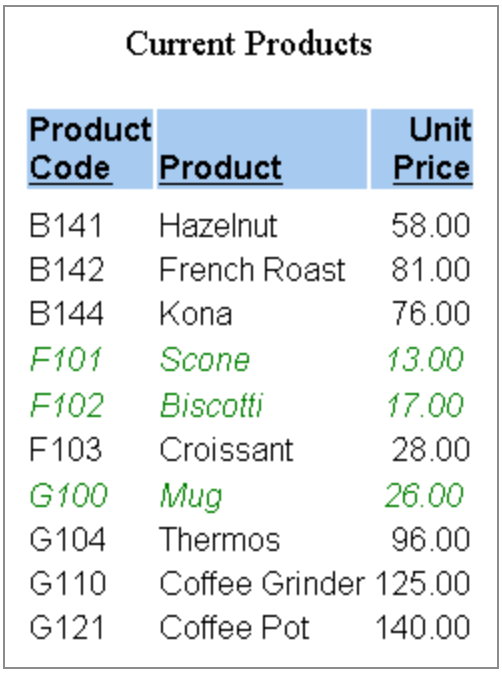
外部カスケードスタイルシートファイルの格納先 WebFOCUS リポジトリパスを指定します。このパスは、.css ファイルの格納先を指定する Web アクセスが可能な URL に、内部的に変換されます。

CSSURL 属性を使用した外部カスケードスタイルシートへのリンク

次のレポートは、Gotham Grinds により現在提供される製品を表示します。レポートには、外部カスケードスタイルシート (CSS) および WebFOCUS スタイルシート内の CSSURL 属性による CSS へのリンクによりフォーマットが設定されています。

```
TABLE FILE GGPRODS
HEADING
"</1 Current Products</1"
PRINT PRODUCT_DESCRIPTION UNIT_PRICE
BY PRODUCT_ID
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, CSSURL=http://websrv2/css/report01.css, $
TYPE=HEADING, CLASS=headText, $
TYPE=TITLE, CLASS=reportTitles, $
TYPE=DATA, CLASS=lowCost, WHEN=N3 LT 27, $
ENDSTYLE

END
```

このリクエストにより作成されるレポートは次のとおりです。



Current Products

Product Code	Product	Unit Price
B141	Hazelnut	58.00
B142	French Roast	81.00
B144	Kona	76.00
F101	Scone	13.00
F102	Biscotti	17.00
F103	Croissant	28.00
G100	Mug	26.00
G104	Thermos	96.00
G110	Coffee Grinder	125.00
G121	Coffee Pot	140.00

注意：report01.css ファイルの格納先 WebFOCUS リポジトリパスを指定するには、リクエストの TYPE=REPORT の CSSURL パラメータで、次の構文を使用します。

```
TYPE=REPORT, CSSURL=IBFS:/WFC/Repository/css/report01.css, $
```

ここで、「CSS」は WebFOCUS リポジトリで report01.css が格納されているフォルダを示します。

CSSURL パラメータを指定した外部カスケードスタイルシート (CSS) へのリンク

SET パラメータを使用して外部カスケードスタイルシート (CSS) をレポートにリンクさせるには、次の SET コマンドをプロシジャ内で発行します。

```
SET CSSURL = css_url
```

または、リクエスト内で次の ON TABLE SET コマンドを発行します。

```
ON TABLE SET CSSURL css_url
```

説明

css_url

外部カスケードスタイルシートの URL です。外部カスケードスタイルシート (CSS) が大文字と小文字を区別する Web サーバプラットフォーム上に存在する場合、大文字と小文字を正しく使用して指定する必要があります。

URL の長さの最大値は次のとおりです。

- SET コマンドでは 69 バイト
- ON TABLE SET コマンドでは 57 バイト

外部カスケードスタイルシートの URL がこの制限を超過する場合、パスの一部を表すエイリアス (「仮想ディレクトリ」とも呼ばれる) を Web サーバで定義することにより URL を短縮することができます。

URL は、絶対パスまたは相対パスで指定することができます。相対パスを使用する場合、外部カスケードスタイルシート (CSS) は WebFOCUS で使用される Web サーバ上に存在する必要があります。

CSSURL を複数回指定すると、ON TABLE SET を使用して指定された最後の値により、リクエスト内のその他すべての値が上書きされます。CSSURL がリクエスト内で指定されていない場合、SET を使用して指定された最後の値により、その他すべての値が上書きされます。

SET コマンドについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』の「環境のカスタマイズ」を参照してください。

継承および外部カスケードスタイルシート

外部カスケードスタイルシート (CSS) を使用してフォーマットが設定されたレポートでは、レポートコンポーネントは TD 要素、および TD 要素がネストされているすべての要素 (例、BODY) からフォーマットを継承します。継承は、すべての CSS の動作と同様に、各ユーザの Web ブラウザにより実装され、ブラウザに依存することに注意してください。

これは、WebFOCUS スタイルシートでフォーマットが設定されたレポートとは異なります。WebFOCUS スタイルシートの場合、レポートコンポーネントは上位のコンポーネントからフォーマットを継承します。外部カスケードスタイルシートのクラスを使用してレポートにフォーマットを設定する際は、レポートコンポーネントに割り当てられたクラスは、より上位のコンポーネントに割り当てられたクラスからフォーマットを継承するわけではありません。

TD 要素からフォーマットを継承するレポートフィールド

このレポートは、Gotham Grinds に製品を提供するベンダーのリストを表示します。フォーマットには、以下が指定されています。

- レポート全体のデフォルトの背景色にオレンジ色を指定する。これは、TD 要素のルールで指定されます。
- レポートのデータのフォントに斜体の Arial を指定する。レポートのデータは、TD 要素のルールからオレンジ色の背景色を継承します。
- レポートの PRODUCT_ID データの背景色に黄色を指定し、TD ルールに指定されたデフォルト値を上書きする。

  レポートのフォーマットが、外部カスケードスタイルシートではなく、WebFOCUS スタイルシートで指定されている場合、PRODUCT_ID はその親レポートコンポーネント (レポートのデータ) から斜体の Arial フォントを継承します。フォーマットが外部カスケードスタイルシート (CSS) に指定されているため、PRODUCT_ID はフォーマットを TD 要素のルールから継承します。上位のレポートコンポーネントからはフォーマットを継承しないため、斜体の Arial フォントは継承されません。

リクエストおよびインライン WebFOCUS スタイルシートは、prodvend.fex プロシジャでは次のように表されます。プロシジャの後には、「report02.css」という名前の外部カスケードスタイルシートが示されています。

prodvend.fex

```
   TABLE FILE GGPRODS
   PRINT PRODUCT_DESCRIPTION VENDOR_NAME
   BY PRODUCT_ID
   ON TABLE SET PAGE-NUM OFF
```

```
   ON TABLE SET STYLE *
1.TYPE=REPORT, CSSURL = http://websrv2/css/report02.css, $
2.TYPE=DATA, CLASS=Data, $
3.TYPE=DATA, COLUMN=PRODUCT_ID, CLASS=Sort, $
    ENDSTYLE
```

```
    END
```

report02.css

```
4.TD    {background:orange; border:0}
5.TABLE {border:0}
6. .Data {font-style:italic; font-family:Arial}
7. .Sort {background:yellow}
```

1. CSSURL を設定し、外部カスケードスタイルシート report02.css にリンクします。
2. Data クラスの CSS ルールを使用して、レポートのデータにフォーマットを設定します。
3. Sort クラスの CSS ルールを使用して、レポートの PRODUCT_ID データにフォーマットを設定します。これにより、2 行目のレポートデータの宣言が上書きされます。
4. この TD 要素の CSS ルールは、背景色にオレンジ色を指定します。これは TD 要素のルールであるため、レポート全体に適用されます。特定のレポートコンポーネントでこのルールを上書きするには、プロシジャの 7 行目の Sort クラスのルールのように、そのコンポーネントの汎用クラスのルールを適用します。
5. これらの TD および TABLE 要素の CSS ルールは、レポートのデフォルト罫線を削除します。
6. 汎用クラス Data の CSS ルールは、フォントファミリーに Arial を、フォントスタイルに斜体を指定します。WebFOCUS スタイルシートは、このルールをレポートのデータに適用します (2 行目を参照)。

   このルールは、背景色を TD 要素のルールから継承します (4 行目を参照)。
7. 汎用クラス Sort の CSS ルールは、背景色に黄色を指定します。WebFOCUS スタイルシートは、このルールを PRODUCT_ID のデータに適用します (3 行目を参照)。

   このルールは、4 行目で指定したデフォルトの背景色を上書きします。

プロシジャは、レポートに次のフォーマットを設定しています。

Product Code	Product	Vendor Name
B141	*Hazelnut*	*Coffee Connection*
B142	*French Roast*	*European Specialities,*
B144	*Kona*	*Evelina Imports, Ltd*
F101	*Scone*	*Ridgewood Bakeries*
F102	*Biscotti*	*Delancey Bakeries*
F103	*Croissant*	*West Side Bakers*
G100	*Mug*	*NY Ceramic Supply*
G104	*Thermos*	*ThermoTech, Inc*
G110	*Coffee Grinder*	*Appliance Craft*
G121	*Coffee Pot*	*Appliance Craft*

HTML 以外のレポートでの外部カスケードスタイルシートの使用

HTML として生成されたレポートのフォーマット設定に外部カスケードスタイルシート (CSS) を使用することはできますが、PDF などの異なる出力タイプとして生成されたレポートには使用できません。HTML 出力タイプ以外のレポートを作成する場合にもカスケードスタイルシートの利点を活用するには、次の方法を使用することをお勧めします。

- **共通フォーマット**　WebFOCUS スタイルシートマクロ内のすべての出力タイプに共有されるフォーマットを指定します。たとえば、フォントスタイルには HTML と PDF の両方の出力フォーマットに適用可能な斜体を設定することができるため、この設定をマクロに指定することができます。各マクロには、2 つのバージョンを定義します。
  - 1 つは HTML 出力用です。このバージョンでは、カスケードスタイルシートのクラスを使用してフォーマットを指定します。
  - 2 つ目は HTML 以外の出力用です。このバージョンでは、ネイティブ WebFOCUS スタイルシート属性を使用してフォーマットを指定します。
- **固有フォーマット**　HTML 出力のみ、または HTML 以外の出力のみに適用可能なフォーマットを WebFOCUS スタイルシートに指定します。出力タイプのマクロ定義を含む WebFOCUS スタイルシートセクションに、これらの宣言をそれぞれ配置します。たとえば、罫線表示の切り替えは HTML 出力には適用できますが、Excel 2000 には適用できません。このため、このフォーマットルールは HTML のマクロ定義に含めます。

- **分岐**　ダイアログマネージャを使用して、HTML の宣言と HTML 以外の宣言に分岐します。

複数の出力タイプでの外部カスケードスタイルシート (CSS) の使用

HTML 出力タイプ以外のレポートを作成する場合にもカスケードスタイルシート (CSS) の利点を活用するには、次の方法を使用することをお勧めします。

```
-DEFAULTS &FORMAT='output_type';
SET ONLINE-FMT = &FORMAT
    TABLE FILE datasource
     report_logic
```

```
    ON TABLE SET STYLE *
TYPE=REPORT, CSSURL = CascadingStyleSheetURL, $
-IF &FORMAT NE 'HTML' GOTO NONHTML;
DEFMACRO=macro1, CLASS=class1, $
    DEFMACRO=macro2, CLASS=class2, $
     .
     .
     .
TYPE=component3, CLASS=class3, $
     .
     .
     .
-GOTO SHARED
-NONHTML
DEFMACRO=macro1, attribute1=value1, $
    DEFMACRO=macro2, attribute2=value2, $
     .
     .
     .
TYPE=component4, attribute4=value4, $
     .
     .
     .
-SHARED
TYPE=component1, MACRO=macro1, $
    TYPE=component2, MACRO=macro2, $
     .
```

```
  .
  .
ENDSTYLE
END
```

1. レポート出力タイプ (例、HTML、PDF、EXL2K) をダイアログマネージャ変数 &FORMAT に割り当てます。この変数は、WebFOCUS スタイルシートを HTML 出力のフォーマット設定に使用するか、HTML 以外の出力のフォーマット設定に使用するかを切り替えます。また、SET ONLINE-FMT に値を入力するために使用します。

   フォームやその他のプレゼンテーションロジックを使用することにより、アプリケーションユーザがレポート出力タイプを選択できるようになります。

2. &FORMAT の値にレポートの出力タイプを設定します。このプロシジャでは、SET ONLINE-FMT はレポートの表示タイプを設定します。また、ON TABLE HOLD を使用して、レポートをファイルとして保存し、そのファイルタイプを設定することができます。

3. CSSURL を設定して、外部カスケードスタイルシートにリンクし、レポートの HTML 出力のフォーマット設定に使用することができます。レポートが HTML 以外のフォーマットで出力される場合は、このコマンドは無視されます。

4. レポートの現在の出力タイプ (&FORMAT で指定された出力タイプ) の WebFOCUS スタイルシート宣言に分岐します。

5. WebFOCUS スタイルシートマクロの HTML バージョンを定義します。これらのマクロで指定するフォーマットは、すべての出力タイプで共有されます。

   HTML バージョンのマクロは、外部カスケードスタイルシートクラスを使用して実装されます。

6. HTML 出力のみに適用可能なフォーマットがある場合、外部カスケードスタイルシートクラスを使用してそのフォーマットをここで指定します。

7. レポートコンポーネントにマクロを適用する WebFOCUS スタイルシート宣言に分岐します。

8. このラベルは、マクロ定義の開始位置および HTML 以外のレポート出力に一意のフォーマット宣言を示します。

9. WebFOCUS スタイルシートマクロの HTML 以外のバージョンを定義します。これらのマクロで指定するフォーマットは、すべての出力タイプで共有されます。

   HTML 以外のバージョンのマクロは、ネイティブ WebFOCUS スタイルシート属性を使用して実装されます。

10. HTML 以外の出力のみに適用可能なフォーマットがある場合、ネイティブ WebFOCUS スタイルシート属性を使用してそのフォーマットをここで指定します。
11. このラベルは、レポートにマクロを適用する宣言の開始位置を示します。
12. マクロには、定義済みのマクロとレポートに適用されているマクロがあります。

PDF および HTML 出力との外部カスケードスタイルシート (CSS) の使用

次のレポートプロシジャ (videorpt.fex) は、HTML および PDF フォーマットの出力を生成します。HTML 出力の生成には、レポートのフォーマット設定に外部カスケードスタイルシート (reports.css) が使用されます。PDF 出力の生成には、内部 WebFOCUS スタイルシートが使用されます。いずれの場合も、レポートの LASTNAME フィールドの背景色には水色が指定され、フィールドタイトルはすべて太字で表示されます。

次のプロシジャは、HTML で出力されます。

videorpt.fex

```
1.  -DEFAULTS &FORMAT='HTML';
2.SET CSSURL = http://websrv2/css/reports.css
3. SET ONLINE-FMT = &FORMAT
    TABLE FILE VIDEOTRK
    PRINT LASTNAME AS 'Last Name' FIRSTNAME AS 'First Name'
    BY LOWEST 5 CUSTID AS 'Cust ID'
    ON TABLE SET PAGE-NUM OFF
    ON TABLE SET STYLE *
4.  -IF &FORMAT NE 'HTML' GOTO NONHTML;
5.DEFMACRO=boldTitles, CLASS=bold, $
    DEFMACRO=blueColumn, CLASS=blueBack, $
6.  -GOTO SHARED
7.  -NONHTML
8.DEFMACRO=boldTitles, STYLE=bold, $
    DEFMACRO=blueColumn, BACKCOLOR=light blue, $
9.  -SHARED
10.TYPE=DATA, COLUMN=LastName, MACRO=blueColumn, $
    TYPE=TITLE, MACRO=boldTitles, $
    ENDSTYLE
    END
```

reports.css

```
11. .bold {font-weight: bolder}
12. .blueBack {background: lightblue}
13.TABLE {border:0}
13.TD    {border:0}
```

1. &FORMAT にデフォルト値を割り当てることで、WebFOCUS スタイルシートを HTML 出力のフォーマット設定に使用するか、PDF 出力のフォーマット設定に使用するかを切り替えます。ここでは、出力フォーマットは HTML に設定されています。
2. CSSURL を設定して、外部カスケードスタイルシート reports.css にリンクし、レポートの HTML 出力にフォーマットを設定します。
3. &FORMAT の値にレポートの表示タイプを設定します。
4. レポートの現在の出力タイプ (HTML) の WebFOCUS スタイルシート宣言に分岐します。
5. 外部カスケードスタイルシートクラスを使用して実装される WebFOCUS スタイルシートマクロの HTML バージョンを定義します。
6. レポートコンポーネントにマクロを適用する WebFOCUS スタイルシート宣言に分岐します。
7. このラベルは、PDF レポート出力のマクロ定義の開始位置を示します。
8. これらの宣言は、ネイティブ WebFOCUS スタイルシート属性を使用して実装される WebFOCUS スタイルシートマクロの PDF バージョンを定義します。これらのマクロ定義は、&FORMAT が HTML に設定されているために無視されます。
9. このラベルは、レポートにマクロを適用する宣言の開始位置を示します。
10. マクロには、定義済みのマクロとレポートに適用されているマクロがあります。
11. このカスケードスタイルシート宣言は、テキストの太さをより太くします。
12. このカスケードスタイルシート宣言は、背景色に水色を指定します。
13. TABLE 要素および TD 要素のこれらの CSS ルールは、レポートのデフォルト罫線を削除します。

プロシジャは、レポートに次のフォーマットを設定しています。

Cust ID	Last Name	First Name
0925	CRUZ	IVY
0944	HANDLER	EVAN
1118	WILSON	KELLY
1133	KRAMER	CHERYL
1237	GOODMAN	JOHN

外部カスケードスタイルシートの要件

レポートのフォーマット設定に外部カスケードスタイルシート (CSS) を使用する際は、次の要件に注意してください。

- **HTML レポート出力の生成**　HTML として生成するレポートはすべて、レポート出力をファイルに保存する場合でも、Web ブラウザに直接送信する場合でも、外部カスケードスタイルシートを使用してフォーマットを設定することができます。ただし、PDF や Excel などの HTML 以外のフォーマットで生成されたレポートには外部カスケードスタイルシート (CSS) は使用できません。
- 内部カスケードスタイルシートを生成しない場合、次の例外を除いて、外部カスケードスタイルシート (CSS) クラス (CLASS=) とネイティブ WebFOCUS スタイルシート属性は、同一 WebFOCUS スタイルシートには指定しないでください。両方のスタイルシートを指定すると、フォーマットの競合が発生する場合があります。

  例外　外部カスケードスタイルシート (CSS) クラスが指定されている場合にも、次の場合にはネイティブ WebFOCUS スタイルシートを使用してください。

  - ハイパーリンクの作成 (プロシジャ、JAVASCRIPT、URL 属性の使用)。ただし、ハイパーリンクのフォーマット設定には、カスケードスタイルシートを使用します。
  - WebFOCUS スタイルシート宣言への条件の指定 (WHEN 属性の使用)。
  - イメージの埋め込み (IMAGE 属性の使用)。ただし、イメージのフォーマット設定 (例、配置) には、カスケードスタイルシートを使用します。
- **同一フォーマットの指定に TABLE/GRAPH と CSS を使用しない**　HEADING CENTER、PAGE-BREAK、スポットマーカー (例、</3) などの TABLE 言語 (または GRAPH 言語) の指示を使用して、フォーマットを設定することができます。ただし、1 つのレポートコンポーネントの同一フォーマット設定には、TABLE (または GRAPH) 言語の指示と外部カスケードスタイルシートのルールの両方は適用しないでください。
- **SET STYLEMODE**　レポートの一般的なフォーマットにカスケードスタイルシートを

使用する場合、STYLEMODE を FULL (デフォルト) または PAGED に設定することができます。STYLEMODE を FIXED に設定し、外部カスケードスタイルにリンクすると、レポートは BODY 要素および PRE 要素からフォーマットを継承し、クラスおよび TD 要素を使用してレポートにフォーマットを設定することはできなくなります。

- **カスケードスタイルシートが有効な Web ブラウザの使用**　カスケードスタイルシートを使用してフォーマットが設定されたレポートを表示する場合、ユーザは CSS をサポートするブラウザを使用している必要があります。カスケードスタイルシートは、WebFOCUS で動作保証されている Microsoft Internet Explorer のすべてのバージョンでサポートされます。

  カスケードスタイルシートルールによるレポートのフォーマット設定方法は、Web ブラウザ (WebFOCUS ではなく) のカスケードスタイルシートのサポートと実装により決定されることに注意してください。Web ブラウザによっては、最新バージョンの CSS をサポートしない場合や、異なる方法で CSS 機能を適用する場合があります。

- **レポートに指定したカスケードスタイルシートを上書きしない**　ユーザのブラウザがカスケードスタイルシートを無視するようにカスタマイズされている場合、またはユーザ独自のカスケードスタイルシートを使用している場合、ユーザが指定したカスケードスタイルシートにより、意図したとおりにレポートを表示するには、使用しているブラウザの設定を変更し、各ドキュメントのカスケードスタイルシートを受け入れるように設定する必要があります。

  ブラウザの設定の確認および変更についての詳細は、ブラウザのヘルプドキュメントを参照してください。CSS ルール間 (例、ドキュメントの CSS 内に指定されたルールとユーザの Web ブラウザ CSS に指定されたルール) の競合の解決方法についての詳細は、使用する CSS のマニュアルを参照してください。

外部カスケードスタイルシート (SET HTMLCSS ON) 使用上の注意

- 要素内の複数のサブタイプに内部および外部スタイルが適用される場合、外部カスケードスタイルシートから参照される CLASS にのみスタイルが適用されます。要素に適用される内部スタイルは無視されます。HTMLCSS が OFF の場合、各サブタイプに適用される内部スタイルのみが反映されます。
- 外部カスケードスタイルシートで境界線が指定されていないレポートでは、HTMLCSS が OFF の場合、内部スタイルの境界線が適用され、罫線のみが適用されます。
- 外部カスケードスタイルシートで指定される境界スタイルは、内部スタイルシートの

指定を上書きします。HTML CSS が OFF の場合、外部カスケードスタイルシートで指定される罫線と境界線スタイルの両方が表示されます。

- 境界線の太さは、レポートのすべての要素で同一になります。HTMLCSS を OFF に設定した場合、レポートの異なる要素の境界線は、異なる太さで表示されます。
- レポート見出しに複数の線が含まれている場合、見出し全体が境界線で囲まれます。HTMLCSS を OFF に設定した場合、見出しの各行が境界線で囲まれます。

外部カスケードスタイルシートの使用に関する FAQ

ここでは、レポートのフォーマット設定での外部カスケードスタイルシート (CSS) の使用に関する FAQ を紹介します。

CSS を使用してレポートのデフォルトフォーマットを指定するにはどうすればいいですか。

レポート全体のデフォルトフォーマットは、外部カスケードスタイルシートルールの BODY 要素または TD 要素に指定することができます。

CLASS 属性を毎回使用する必要はありますか。

個々のレポートコンポーネントのフォーマットを指定する場合は、WebFOCUS スタイルシート内に CLASS 属性が必要です。CLASS 属性は、レポートコンポーネントへの汎用クラスのルールの割り当てに使用します。レポート全体のフォーマットは、クラスのルールではなく、BODY 要素または TD 要素のルールに指定します。このため、CLASS 属性は省略します。

スタイルシートの CSS クラスに参照を配置する場合、<TD> タグおよび <A> に適用されます。たとえば、「class1」クラスが外部カスケードスタイルシート (CSS) に格納されている場合、WebFOCUS はドリルダウンの値用に次の HTML を生成します。

```
<TD CLASS='class1'>
<A class='class1' HREF="...">ENGLAND</A>
</TD>
```

たとえば、下線なしの赤色のハイパーリンクを設定する場合、次を発行します。

```
SET CSSURL=http://myserver/mycss.css
TABLE FILE CAR
SUM SALES BY COUNTRY
ON TABLE PCHOLD FORMAT HTML
```

```
ON TABLE SET STYLE *
type=data, column=country, focexec=fex1, class=class1, $
END
```

mycss.css には、次のように記述されています。

```
.class1 { color:red; text-decoration:none }
```

出力結果は次のとおりです。

カスケードスタイルシートと WebFOCUS スタイルシートは同時に使用できますか。

外部カスケードスタイルシートをリンクする場合でも、WebFOCUS スタイルシートにネイティブ WebFOCUS スタイルシート属性を指定することができます。ただし、内部カスケードスタイルシートを生成しない場合、条件付きフォーマットへの条件の指定を除いて (例、別のリソースへのリンク作成やイメージの埋め込み)、同一 WebFOCUS スタイルシート内に CSS クラス (CLASS=) とネイティブ WebFOCUS スタイルシート属性を同時に指定しないでください。

WebFOCUS はどの CSS バージョンをサポートしますか。

カスケードスタイルシートの異なるバージョン (例、CSS2) のサポートは、Web ブラウザ (WebFOCUS ではなく) でのカスケードスタイルシートのサポートと実装により決定されます。Web ブラウザによっては、CSS の最新バージョンをサポートしない場合や、CSS 機能の実装方法が異なる場合があることに注意してください。

PDF または Excel 2000 として生成されるレポートのフォーマット設定に CSS を使用することはできますか。

できません。外部カスケードスタイルシートは、HTML として生成されたレポートのフォーマット設定にのみ使用できます。

外部カスケードスタイルシートを使用してフォーマットを設定できるレポートタイプを教えてください。

外部カスケードスタイルシート (CSS) は、すべてのレポートタイプのフォーマット設定に使用できます。

- **表形式レポート**　通常の (行と列で構成される) レポート、および FML (Financial Modeling Language、拡張マトリックスや行指向とも呼ばれます) レポートなどがあります。
- **グラフ**　グラフの見出しと脚注、およびグラフが表示されるページの背景色と背景イメージはフォーマットの設定が可能ですが、グラフ自体は Java で生成されるため、CSS を使用してフォーマットを設定することはできないことに注意してください。
- **自由形式レポート**　多くのユーザは、HTML 以外の出力タイプを使用して自由形式レポートを生成するため、CSS は自由形式のフォーマット設定にはほとんど使用されません。

外部カスケードスタイルシートのトラブルシューティング

次の説明は、外部カスケードスタイルシート (CSS) を使用してレポートにフォーマットを設定する際に発生する一般的な問題の解決方法を示しています。

問題 - レポートにカスケードスタイルシートで変更した内容が反映されない。

- **原因**　外部カスケードスタイルシートを参照するレポートを実行している場合、Web ブラウザのメモリまたはディスクキャッシュに CSS ファイルが格納されます。その後、CSS ファイルを編集し、レポートを再度実行すると、ブラウザは格納された CSS ファイルの以前のバージョンを継続して使用する場合があります。

  解決法　ブラウザの [更新] ボタン (Microsoft Internet Explorer) をクリックして、Web サーバから CSS ファイルを再ロードします。これにより、Web ブラウザでは、レポートのフォーマット設定にカスケードスタイルシートの最新バージョンが使用されます。

問題 - カスケードスタイルシートのフォーマット設定がレポートで使用されていない。

- **原因 1**　外部カスケードスタイルシートにリンクを作成したときに間違った URL を指定した可能性があります。

  解決法 1　CSSURL 属性、SET CSSURL コマンド、または LINK 要素 (レポートプロシジャが -HTMLFORM を使用する場合) でリンクを指定する URL を確認し、必要に応じて修正します。

- **原因 2**　Web ブラウザでカスケードスタイルシートがサポートされていない可能性があります。

  解決法 2　カスケードスタイルシートは、WebFOCUS で動作保証されている Microsoft Internet Explorer のすべてのバージョンでサポートされます。ブラウザが動作保証されていることを確認します。動作保証されていない場合は、Internet Explorer または Firefox の適切なバージョンをインストールします。

- **原因 3**　Web ブラウザでカスケードスタイルシートを無効にするよう設定されている可能性があります。

  解決法 3　ドキュメントのカスケードスタイルシートを受け入れるようにブラウザの設定を変更します。ブラウザの設定の確認および変更についての詳細は、ブラウザのヘルプドキュメントを参照してください。

- **原因 4**　Web ブラウザによっては、ルールに指定されたプロパティを 1 つでもサポートしない場合はルール全体が無視される可能性があります。このタイプの Web ブラウザでは、レポートのすべてのフォーマット設定が 1 つのルールで指定され (例、TD または BODY のルール)、そのルールで指定されたプロパティのいずれかがブラウザでサポートされない場合、フォーマット設定はいずれもレポートに適用されません。

  解決法 4　サポートされていないプロパティを削除するか、プロパティをサポートするバージョンにブラウザをアップグレードします。

- **原因 5**　Web ブラウザによっては、ネストされた要素の CSS 継承ルールの実装方法が CSS の標準に準拠しない場合があります。使用しているブラウザがこのタイプで、たとえば BODY のルールにすべてのフォーマットを指定している場合は、ブラウザは BODY 内にネストしているその他の要素にルールを適用しない場合があります。

  解決法 5　異なる要素のルールにレポートのフォーマットを指定するか (例、ブラウザが BODY からの継承を正しく実装しない場合は TD のルールを使用) 、継承を正しくサポートするバージョンにブラウザをアップグレードします。

- **原因 6**　CSS ファイルを参照するレポートの実行時に、サーバパス上で CSS ファイルが見つからなかった可能性があります。

  解決法 6　CSS ファイルが格納されているディレクトリがサーバ検索パス上に存在するようにします。

問題 - レポートに CSS フォーマットの一部のみが反映される。

- **原因 1**　カスケードスタイルシートによるレポートのフォーマット設定方法は、Web

ブラウザ (WebFOCUS ではなく) でのカスケードスタイルシートのサポートと実装により決定されます。この問題は、使用しているカスケードスタイルシートのレベルをブラウザがサポートしないため、実装されない CSS 機能が存在することにより、発生している可能性があります。

解決法 1　レポートのフォーマット設定に使用される CSS 機能のすべてをサポートするバージョンにブラウザをアップグレードするか、カスケードスタイルシートを編集して、レポートの表示に使用されるブラウザの一部でサポートされていない機能を削除します。

- **原因 2**　Web ブラウザがユーザ独自のカスケードスタイルシートを使用するよう設定されているため、そのカスケードスタイルシートで指定されたルールの一部が、レポートに割り当てられたカスケードスタイルシートで指定されたルールを上書きしている可能性があります。異なるカスケードスタイルシートのルール間の競合の解決方法についての詳細は、使用する CSS のマニュアルを参照してください。

  解決法 2　ブラウザの設定を変更して各ドキュメントのカスケードスタイルシートを受け入れるか、2 つのカスケードスタイルシート内のルールを編集して、競合を解消します。

- **原因 3**　Web ブラウザによっては、ルールに指定されたプロパティを 1 つでもサポートしない場合は、ルール全体を無視するものがあります。使用しているブラウザがこのタイプで、レポートに割り当てられたクラスのいずれかのルールに指定されたプロパティが 1 つでもサポートされない場合、ルールが割り当てられたレポートコンポーネントにはフォーマットが設定されません。

  解決法 3　サポートされていないプロパティを削除するか、プロパティをサポートするバージョンにブラウザをアップグレードします。

- **原因 4**　各レポートコンポーネントに割り当て可能なカスケードスタイルシートのクラスは 1 つのみです。複数のクラスを指定した場合、1 つ目に指定されたクラスのみがコンポーネントに割り当てられます。その他のクラスはすべて無視されます。

  クラスがレポートセルに割り当てられておらず、条件付きフォーマットが指定された場合、その行により条件が満たされた 1 つ目のクラスのみがセルに割り当てられます。その他のクラスはすべて無視されます。

  解決法 4　各レポートコンポーネントには、複数の CSS クラスを割り当てないでください。複数の属性を適用する必要がある場合は、これらの属性を 1 つのクラスにまとめます。

- **原因 5**　Web ブラウザによっては、ネストされた要素の CSS 継承ルールの実装方法が CSS の標準に準拠しない場合があります。使用しているブラウザがこのタイプで、たとえば BODY のルールに一部のフォーマットを指定している場合、ブラウザは BODY 内にネストしているその他の要素にルールを適用しない場合があります。

  解決法 5　異なる要素のルールにレポートのフォーマットを指定するか (例、ブラウザ

が BODY からの継承を正しく実装しない場合は TD のルールを使用) 、継承を正しくサポートするバージョンにブラウザをアップグレードします。

- **原因 6**　外部カスケードスタイルシートは、その他のフォーマット設定方法と使用される場合、制限の対象となります。たとえば、WebFOCUS スタイルシートレポートでは内部カスケードスタイルシートを生成しないが、そのレポートが外部カスケードスタイルシート (CSS) クラスを参照し、かつネイティブ WebFOCUS スタイルシート属性を指定している場合、フォーマットの競合が発生する可能性があります。

  解決法 6　解決方法は、フォーマットの競合の種類により異なります。上記の例では、内部カスケードスタイルシートを生成することにより問題が解決します。

問題 - レポートを対話形式で実行した場合は CSS スタイルが反映されるが、レポートを ReportCaster で配信した場合は CSS スタイルが反映されない。

- **原因**　ReportCaster がレポートを配信するメールサーバで、外部参照された CSS ファイルがサポートされない可能性があります。たとえば、Gmail では HTML Email から CSS が除外されるため、インライン CSS を使用する必要があります。

  解決法　プロシジャで WebFOCUS コマンドの SET HTMLCSS=ON を発行するか、リクエストで ON TABLE SET HTMLCSS ON コマンドを発行します。これにより、インライン CSS が含まれたレポートが作成されます。

レポートページのレイアウト

スタイルシート属性を使用して、ページレイアウトをカスタマイズすることができます。利用可能なページレイアウト属性は、他のすべての属性と同様に、表示フォーマットによって異なります。たとえば、ある属性は PDF などの印刷向け表示フォーマットでサポートされ、ブラウザ内で表示する HTML レポートには適用できません。

各レポートページには、デフォルトのレイアウト特性が設定されています。ただし、これらの特性を変更してレイアウトをカスタマイズすることができます。ページ内のフィールドの配置、ページ番号、改ページ、罫線、イメージの使用など、レポートの外観の制御が可能です。

ページレイアウトを定義するには、大部分のスタイルシート属性の代わりに SET パラメータを使用することができます。SET についての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

ページサイズ、用紙の方向、色の選択

レポートのページサイズ、用紙の方向 (縦長または横長)、およびページの色を選択することができます。デフォルトのページサイズは、レター (8.5 x 11 インチ) ですが、Legal、封筒 (Envelope) など、さまざまなサイズを選択することができます。

ページサイズ、用紙の方向、色の属性

属性	説明	適用先
`PAGESIZE`	ページサイズを設定します。	開発環境で設定 • PDF • PPT

属性	説明	適用先
		• PPTX
`ORIENTATION`	用紙の方向を設定します。	開発環境で設定 • PDF • EXL2K • PPT • PPTX
`PAGECOLOR`	ページの色を設定します。	• 内部カスケードスタイルシートを使用した HTML レポート • PPTX **注意：**レポートにテンプレートが含まれている場合、テンプレートのスタイルを保持するため、PAGECOLOR スタイルシート属性は無視されます。

ページサイズの設定

次の構文は、PDF、PPT、PPTX レポートに適用されます。

```
[TYPE=REPORT,] PAGESIZE={size|LETTER}, $
```

説明

TYPE=REPORT

ページサイズをレポート全体に適用します。デフォルト値であるため必須ではありません。

size

ページサイズです。レポートを印刷する場合、この値は用紙のサイズに一致する必要があります。サイズが一致していない場合、レポートの一部が欠落したり、余分なブランクが印刷されたりすることがあります。

有効な値には、次のものがあります。

値	説明
`SCREEN`	レポートが画面を満たすようにページサイズを設定します。印刷行の横方向の長さには制限がなく、ページの幅はレポートの幅により決定されます。1ページ内の行数は、画面を満たすのに必要な行数で決まり、フォントおよび画面の解像度により異なります。
`LETTER`	8.5 x 11 インチ (LETTER がデフォルト値)
`LEGAL`	8.5 x 14 インチ
`CUSTOM`	DHTML、PDF、PPTX レポートのカスタムページサイズ設定を有効にします。CUSTOM オプションを使用する場合は、リクエストの SET コマンドまたはスタイルシート属性で CUSTOM-PAGE-LENGTH=*number* および CUSTOM-PAGE-WIDTH=*number* を使用して、長さと幅を指定することができます。*number* の値は、UNITS パラメータ (インチ、センチメートル、ポイント) によって異なります。デフォルト値は インチです。

値	説明
TABLOID	11 x 17 インチ
LEDGER	17 x 11 インチ
WIDESCREEN	13.333 x 7.5 インチ
C	17 x 22 インチ
D	22 x 34 インチ
E	34 x 44 インチ
STATEMENT	5.5 x 8.5 インチ
EXECUTIVE	7.5 x 10.5 インチ
FOLIO	8.5 x 13 インチ
10x14	10 x 14 インチ
A3	297 x 420 ミリメートル

値	説明
A4	210 x 297 ミリメートル
A5	148 x 210 ミリメートル
B4	250 x 354 ミリメートル
B5	182 x 257 ミリメートル
QUARTO	215 x 275 ミリメートル
ENVELOPE-9	3.875 x 8.875 インチ
ENVELOPE-10	4.125 x 9.5 インチ
ENVELOPE-11	4.5 x 10.375 インチ
ENVELOPE-12	4.5 x 11 インチ
ENVELOPE-14	5 x 11.5 インチ
ENVELOPE-MONARCH	3.875 x 7.5 インチ

値	説明
ENVELOPE-PERSONAL	3.625 x 6.5 インチ
ENVELOPE-DL	110 x 220 ミリメートル
ENVELOPE-C3	324 x 458 ミリメートル
ENVELOPE-C4	229 x 324 ミリメートル
ENVELOPE-C5	162 x 229 ミリメートル
ENVELOPE-C6	114 x 162 ミリメートル
ENVELOPE-C65	114 x 229 ミリメートル
ENVELOPE-B4	250 x 353 ミリメートル
ENVELOPE-B5	176 x 250 ミリメートル
ENVELOPE-B6	176 x 125 ミリメートル
ENVELOPE-ITALY	110 x 230 ミリメートル

値	説明
`US-STANDARD-FANFOLD`	14.875 x 11 インチ
`GERMAN-STANDARD-FANFOLD`	8.5 x 12 インチ
`GERMAN-LEGAL-FANFOLD`	8.5 x 13 インチ

用紙方向の設定

次の構文は、PDF、EXL2K レポートに適用されます。

```
[TYPE=REPORT,] ORIENTATION={PORTRAIT|LANDSCAPE}, $
```

説明

TYPE=REPORT

用紙の方向をレポート全体に適用します。デフォルト値であるため必須ではありません。

PORTRAIT

縦方向が横方向より長いページを作成し、縦方向に長いページのレポートを表示します。デフォルト値は PORTRAIT です。

LANDSCAPE

横方向が縦方向より長いページを作成し、横方向に長いページのレポートを表示します。

用紙方向の設定

次のリクエストは、PDF レポートの用紙方向を横長に設定します。

```
SET ONLINE-FMT = PDF
TABLE FILE CENTQA
SUM CNT.PROBNUM AS 'Total Number, of Problems'
SUM CNT.PROBNUM AS 'Problems From, Each Plant' BY PLANT
SUM CNT.PROBNUM AS 'Problem by Product' BY PLANT BY PRODNAME
ON PLANT PAGE-BREAK
HEADING CENTER
"QA Report for Company, Plant, and Product"
" "
ON TABLE COLUMN-TOTAL
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, ORIENTATION=LANDSCAPE, $
ENDSTYLE
END
```

ページ色の設定

次の構文は、内部カスケードスタイルシートを使用する HTML レポートに適用されます。

```
[TYPE=REPORT,] ...PAGECOLOR=color, ... , $
```

説明

TYPE=REPORT

TYPE 指定は、この機能ではオプションです。この指定を省略した場合、デフォルト設定で TYPE は REPORT になります。

color

サポートされる色です。値のリストは、レポートデータのフォーマットを参照してください。

ページ色の設定

次のリクエストは、内部カスケードスタイルシートによる HTML レポートのページの色を銀色に設定します。

```
SET HTMLCSS = ON
TABLE FILE CENTORD
ON TABLE SUBHEAD
"SELECTED PRODUCT INVENTORY"
SUM QTY_IN_STOCK/D12 BY PROD_NUM BY SNAME BY STATE
WHERE PROD_NUM EQ '1004'
WHERE SNAME EQ 'eMart'
WHERE STATE EQ 'CA'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, PAGECOLOR=SILVER, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

SELECTED PRODUCT INVENTORY

Product Number#:	Store Name:	State:	Quantity In Stock:
1004	eMart	CA	689,088

ページマージンの設定

レポートのページマージンを設定できます。これには、上、下、左、右のマージンがあります。デフォルトの単位 (インチ) をセンチメートルまたはポイントのいずれかに変更することもできます。単位は、ページマージン、列幅、および列の配置に適用されます。

ページマージンの属性

属性	説明	適用先
`UNITS`	測定単位を設定します。	PDF

属性	説明	適用先
	マージンサイズやその他のページ特性を指定する際に使用します。定単位を変更した場合、新しい値は測定単位が使用されるすべての状況に適用されます。	内部カスケードスタイルシートを使用したHTML レポート

`TOPMARGIN`
`BOTTOMMARGIN`
`LEFTMARGIN`
`RIGHTMARGIN`
	上、下、左、右のマージンのサイズを設定します。	PDF
		内部カスケードスタイルシートを使用したHTML レポート

測定単位の設定

次の構文は、内部カスケードスタイルシートによる PDF または HTML レポートに適用されます。

スタイルシート内で、次の属性を追加します。

```
UNITS = units
```

リクエスト外で設定する場合、次の構文を使用します。

```
SET UNITS = units
```

リクエスト内で設定する場合、次の構文を使用します。

```
ON TABLE SET UNITS units
```

説明

units

測定単位です。選択できる値には次のものがあります。

- **INCHES** インチとして単位を指定します。これがデフォルト値です。

- **CM** センチメートルとして単位を指定します。
- **PTS** ポイントとして単位を指定します。ポイントは、活字表記では一般的な測定単位です。

マージンサイズの設定

次の構文は、内部カスケードスタイルシートによる PDF または HTML レポートに適用されます。

```
[TYPE=REPORT,] [TOPMARGIN={value|.25},] [BOTTOMMARGIN={value|.25},]
 [LEFTMARGIN={value|.25},] [RIGHTMARGIN={value|.25},] $
```

説明

TYPE=REPORT

マージンサイズをレポート全体に適用します。デフォルト値であるため必須ではありません。

TOPMARGIN

レポートコンテンツの上部の境界を設定します。

BOTTOMMARGIN

レポートコンテンツの下部の境界を設定します。

LEFTMARGIN

レポートコンテンツの左側の境界を設定します。

RIGHTMARGIN

レポートコンテンツの右側の境界を設定します。

value

指定したマージンのサイズです。レポートコンテンツは、このマージンの内側に表示されます。レポートを印刷する場合、プリンタの印刷範囲に対応する値を指定します。たとえ

ば、印刷範囲のマージンが上下左右 0.25 インチの場合、このマージンを 0.25 インチ以上に設定します。

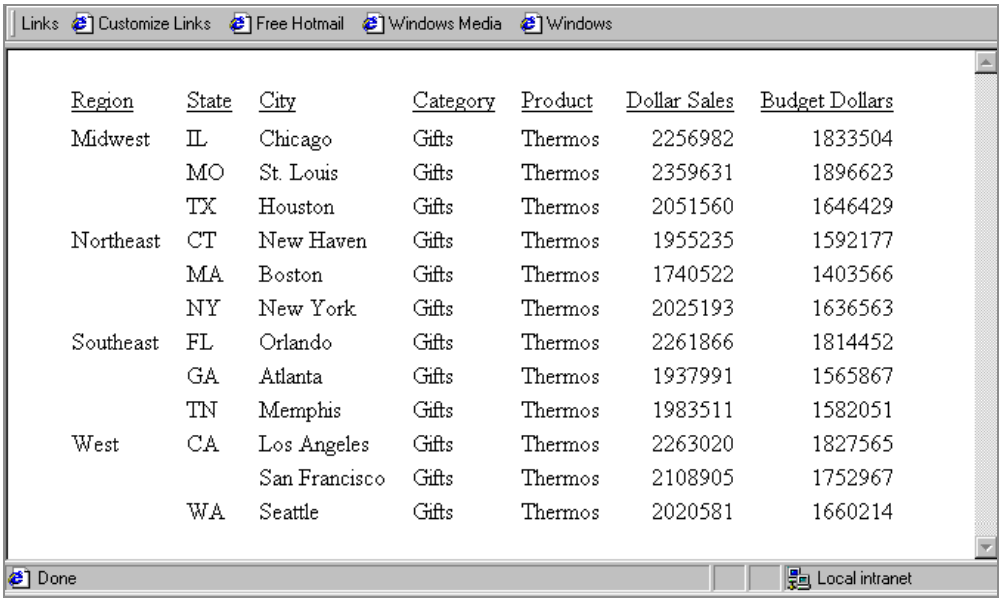
すべてのマージンのデフォルト値は 0.25 インチです。

左マージンの設定

次のリクエストは、内部カスケードスタイルシートによる HTML レポートの左マージンを 1 インチに設定します。

```
SET HTMLCSS = ON
TABLE FILE GGSALES
SUM CATEGORY PRODUCT DOLLARS BUDDOLLARS
BY REGION BY ST BY CITY
WHERE DOLLARS GT BUDDOLLARS
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
LEFTMARGIN = 1, $
ENDSTYLE
END
```

出力結果は次のとおりです。



Region	State	City	Category	Product	Dollar Sales	Budget Dollars
Midwest	IL	Chicago	Gifts	Thermos	2256982	1833504
	MO	St. Louis	Gifts	Thermos	2359631	1896623
	TX	Houston	Gifts	Thermos	2051560	1646429
Northeast	CT	New Haven	Gifts	Thermos	1955235	1592177
	MA	Boston	Gifts	Thermos	1740522	1403566
	NY	New York	Gifts	Thermos	2025193	1636563
Southeast	FL	Orlando	Gifts	Thermos	2261866	1814452
	GA	Atlanta	Gifts	Thermos	1937991	1565867
	TN	Memphis	Gifts	Thermos	1983511	1582051
West	CA	Los Angeles	Gifts	Thermos	2263020	1827565
		San Francisco	Gifts	Thermos	2108905	1752967
	WA	Seattle	Gifts	Thermos	2020581	1660214

レポートコンポーネントの配置

スタイルシートを使用して、列、見出し、脚注の開始位置および、見出しまたは脚注内にある要素の開始位置を、絶対位置または相対位置で指定することができます。レポートコンポーネント周囲にブランクを追加することもできます。

ここでは、スタイルシート属性 POSITION を使用して列を配置する方法について説明します。列配置コマンド IN についての詳細は、列のページ内整列を参照してください。

見出し、脚注の配置、または見出し、脚注の要素についての詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。

配置属性

属性	説明	適用先
`POSITION`	列の絶対または相対開始位置を設定します。 絶対位置は、印刷する用紙の左マージンからの距離です。 相対位置は、デフォルト位置からの距離です。最初の列の後では、デフォルト位置は前の列の終端です。	PDF
`TOPGAP` `BOTTOMGAP`	レポート行の先頭または末尾にブランクを追加します。	PDF
`LEFTGAP` `RIGHTGAP`	レポート列の左側または右側にブランクを追加します。	PDF

列開始位置の指定

この構文は、PDF レポートに適用されます。

```
TYPE=REPORT, COLUMN=identifier, POSITION={+|-}position, $
```

説明

identifier

1 つの列を選択し、使用可能なフィールドタイトル、データ、および合計を集合的に配置します。有効な値についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

+

デフォルト開始位置の右側に指定した距離で列を開始します。

デフォルト設定では、テキスト項目および文字フィールドは、列内で左側に揃えられ、数値フィールドは列内で右側に揃えられます。

-

デフォルト開始位置の左側に指定した距離で列を開始します。

レポートの列が重なり合う場合があります。これが発生した場合は、これらの値を調整します。

position

UNITS 属性で指定した測定単位で、距離を指定します。

列絶対開始位置の指定

以下は、印刷レポート内の列の配置方法について説明しています。PRODUCT_DESCRIPTION フィールドが、PDF レポートの左マージンから 3 インチの位置に表示することをリクエスト内で指定します。

```
SET ONLINE-FMT = PDF
TABLE FILE GGORDER
"PRODUCTS ORDERED ON 08/01/96"
SUM QUANTITY BY PRODUCT_DESCRIPTION
WHERE ORDER_DATE EQ '080196'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, COLUMN=PRODUCT_DESCRIPTION, POSITION=3, $
ENDSTYLE
END
```

出力結果は次のとおりです。

```
PRODUCTS ORDERED ON 08/01/96

                                                Ordered
                                Product         Units

                                Biscotti           6140
                                Coffee Grinder     2493
                                Coffee Pot         3100
                                Croissant          7465
                                French Roast      12965
                                Hazelnut           4186
                                Kona               2591
                                Mug                5332
                                Scone              5949
                                Thermos            2362
```

列相対開始位置の指定

次のリクエストは、QUANTITY フィールドのフィールドタイトルおよびデータをデフォルトの位置から 2 インチの位置に配置します。この場合は、前の列の終端から 2 インチの位置になります。

```
SET ONLINE-FMT = PDF
TABLE FILE GGORDER
"PRODUCTS ORDERED ON 08/01/96"
SUM QUANTITY BY PRODUCT_DESCRIPTION
WHERE ORDER_DATE EQ '080196'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, COLUMN=PRODUCT_DESCRIPTION, POSITION=3, $
TYPE=REPORT, COLUMN=QUANTITY, POSITION=+2, $
ENDSTYLE
END
```

レポート内で「Ordered Units」というタイトルの QUANTITY は、Product との相対位置で配置されます。

```
PRODUCTS ORDERED ON 08/01/96

                                  Product      Ordered Units

                                  Biscotti            6140
                                  Coffee Grinder      2493
                                  Coffee Pot          3100
                                  Croissant           7465
                                  French Roast       12965
                                  Hazelnut            4186
                                  Kona                2591
                                  Mug                 5332
                                  Scone               5949
                                  Thermos             2362
```

レポートコンポーネント周囲へのブランクの追加

この構文は、PDF レポートに適用されます。

```
TYPE=REPORT, {TOPGAP|BOTTOMGAP}=gap, $
TYPE=type, [COLUMN=identifier,|ACROSSCOLUMN=acrosscolumn,]
 {LEFTGAP|RIGHTGAP}=gap, $
```

```
TYPE=type, [COLUMN=identifier,|ACROSSCOLUMN=acrosscolumn,]
 {LEFTGAP|RIGHTGAP}=gap, $
```

説明

TOPGAP

レポート行の上側に追加するブランクの間隔を指定します。

BOTTOMGAP

レポート行の下側に追加するブランクの間隔を指定します。

gap

UNITS 属性で指定された測定単位でブランクの間隔を指定します。

罫線および背景色を使用しない場合、デフォルト値は 0 (ゼロ) です。RIGHTGAP では、デフォルト値はテキストフォントのサイズに比例します。

罫線または背景色を使用する場合、罫線とテキストの間にブランクを挿入するため、またはテキストより長い範囲に背景色を配置するためにデフォルト値が増加します。

1 列または 1行内の間隔は同一にする必要があります。つまり、同一列内の個別セルに異なる左間隔または右間隔、同一行内の個別セルに異なる上部間隔または下部間隔を指定することはできません。

type

レポートコンポーネントを指定します。有効な値についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

identifier

COLUMN 属性を使用して、1 つまたは複数の列を選択します。この属性についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

acrosscolumn

ACROSSCOLUMN 属性を使用して、すべての ACROSS フィールド列を選択します。この属性についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

LEFTGAP

レポートの列の左側に追加するブランクの間隔を指定します。

RIGHTGAP

レポートの列の右側に追加するブランクの間隔を指定します。

注意：TOPGAP、BOTTOMGAP、LEFTGAP、および RIGHT GAP では、最低 0.013889 (インチで表されたポイントの小数値) の値を指定する必要があります。これより小さい値を指定すると、その値が切り捨てられ、近似値の 0 (ゼロ) になります。

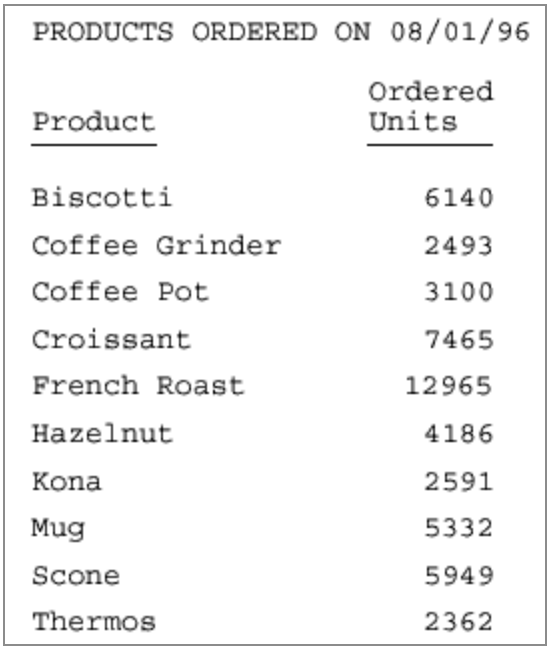
データ値の上へのブランクの追加

次のリクエストは、PDF レポート内の各データの上部に 0.1 インチのブランクを生成します。

```
SET ONLINE-FMT = PDF
SET PAGE-NUM = OFF
TABLE FILE GGORDER
"PRODUCTS ORDERED ON 08/01/96"
```

```
" "
SUM QUANTITY BY PRODUCT_DESCRIPTION
WHERE ORDER_DATE EQ '080196'
ON TABLE SET STYLE *
TYPE=DATA, TOPGAP = 0.1, $
ENDSTYLE
END
```

データを読みやすくするために間隔が取られています。



PRODUCTS ORDERED ON 08/01/96

Product	Ordered Units
Biscotti	6140
Coffee Grinder	2493
Coffee Pot	3100
Croissant	7465
French Roast	12965
Hazelnut	4186
Kona	2591
Mug	5332
Scone	5949
Thermos	2362

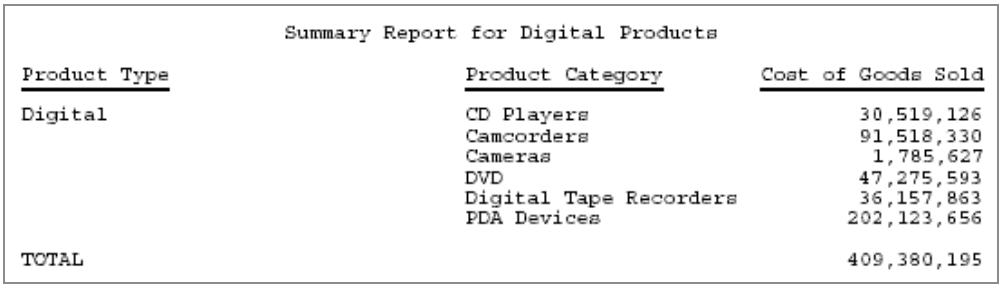
列左側へのブランクの追加

ここでは、レポートコンポーネントの左側にブランクを追加する方法について説明します。この例では、Product Category フィールドの左側に 1.5 インチのブランクが挿入されています。

```
SET ONLINE-FMT=PDF
TABLE FILE CENTORD
HEADING CENTER
"Summary Report for Digital Products"
" "
SUM    LINE_COGS/D12       AS 'Cost of Goods Sold'
BY     PRODTYPE            AS 'Product Type'
BY     PRODCAT             AS 'Product Category'
```

```
WHERE PRODTYPE EQ 'Digital';
ON TABLE COLUMN-TOTAL/D12
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, COLUMN=PRODCAT, LEFTGAP=1.5, $
ENDSTYLE
END
```

出力結果は次のとおりです。



Summary Report for Digital Products

Product Type	Product Category	Cost of Goods Sold
Digital	CD Players	30,519,126
	Camcorders	91,518,330
	Cameras	1,785,627
	DVD	47,275,593
	Digital Tape Recorders	36,157,863
	PDA Devices	202,123,656
TOTAL		409,380,195

列のページ内整列

ページ内の列を整列させることで、より簡単にデータを見つけることができるようになります。さまざまな設計オプションが用意されています。スタイルシート属性またはコマンドを使用して、次のことを行うことができます。

- 列幅の決定
- 列間隔の制御
- 列順序の変更
- 列を積み重ねてレポート幅を縮小、またはマトリックス作成による簡単な値の比較
- 列の絶対または相対開始位置の指定

列整列機能

機能	説明	適用先
`SQUEEZE`	列幅を設定します。	HTML PDF
`SET SPACES`	列間のブランク数を設定します。	HTML
`SEQUENCE`	列の順序を設定します。	PDF HTML EXL2K (**注意：**XLSX および EXL2K FORMULA では利用できません)
`FOLD-LINE`	列を積み重ねることでレポート幅を縮小します。	PDF
`OVER`	列を上部に積み重ねて配置します。	HTML PDF
`IN {n|+n}`	列の絶対または相対開始位置を設定します。	HTML PDF

列幅の決定

スタイルシート内の SQUEEZE 属性の値は、レポート内の列幅を決定します。SQUEEZE の値を設定するために、スタイルシートの代わりに SET パラメータを使用することができます。スタイルシートと SET の値が一致していない場合、スタイルシートが SET より優先されます。SET についての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

SQUEEZE が ON (デフォルト) に設定されている場合、スタイルシートの列幅の指定は、無視されます。列幅の決定には、ブラウザのデフォルト設定が使用されます。

SQUEEZE を使用する際は、レポート内の見出し、脚注、フィールドタイトルの表示方法に影響する場合があります。詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。

列幅の決定 (HTML)

この構文は、HTML レポートに適用されます。

標準 HTML レポートのテーブルおよび列の幅は、ブラウザの処理によって決定されます。これにはブラウザのデフォルト設定、およびブラウザウィンドウまたは iframe のサイズが含まれます。列幅は、最大データ値または列タイトルの大きい方が収まるように調整され、末尾のブランクは自動的に削除されます。

標準の HTML レポートでは、データは単一テーブルに表示されるため、列幅はすべてのデータ行で固定されます。

HFREEZE HTML レポートでは、列タイトルとデータ行の位置揃えを確実にするため、見出し、データ行、脚注の 3 つのテーブルの折り返しが無効になります。

```
[TYPE=REPORT,] SQUEEZE={ON|OFF}, $
```

説明

TYPE=REPORT

列幅をレポート全体に適用します。デフォルト値であるため必須ではありません。

ON

最も長いデータ値と列タイトルの長い方を基準に、列幅を決定します。デフォルト値は ON です。

HTML レポートでは、Webブラウザは最も短いフィールドタイトルまたはフィールド値に合わせて列幅が縮小されます。

OFF

最も長いデータ値と列タイトルの長い方を基準に、列幅を決定します。フィールドタイトルとフィールドフォーマットの長い方の長さになるまで、列幅にブランクを追加します。

注意

- HTML レポートに SQUEEZE 設定および列への条件付きスタイル設定が含まれている場合は、TYPE=REPORT, COLUMN=*n* ではなく TYPE=DATA, COLUMN=*n* を使用します。

- SQUEEZE は OVER 句が生成したフィールドではサポートされません。

デフォルト列幅の使用 (HTML)

次のリクエストは、HTML レポートの SQUEEZE=ON (デフォルト) を使用します。列幅は、データ値とフィールドタイトルの、いずれか長い方を基準に決定されます。

```
SET PAGE-NUM = OFF
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY CATEGORY BY PRODUCT
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT=COURIER, $
ENDSTYLE
END
```

Category、Unit Sales、Dollar Sales のフィールドタイトルは、対応するデータ値よりも長くなっています。Product の列幅は、長い方のデータ値で決定されます。HTML レポートは次のとおりです。

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

フィールドフォーマットに基づく列幅の使用 (HTML)

このリクエストは HTML レポートの SQUEEZE を OFF に設定します。最も長いデータ値と列タイトルの長い方を基準に、列幅を決定します。

```
SET PAGE-NUM = OFF
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY CATEGORY BY PRODUCT
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, SQUEEZE=OFF, FONT=COURIER, $
ENDSTYLE
END
```

Category (A11) および Product (A16) のフィールドフォーマットの長さになるまで、列幅にブランクを追加します。HTML レポートは次のとおりです。

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

列幅の決定 (PDF)

この構文は、PDF レポートに適用されます。

```
[TYPE=REPORT,] COLUMN=identifier, SQUEEZE={ON|OFF|width}, $
```

説明

TYPE=REPORT

列幅をレポート全体に適用します。デフォルト値であるため必須ではありません。

identifier

COLUMN 属性を使用して、列を選択します。この属性についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。列の識別子を省略する場合、SQUEEZE の値はレポート内のすべての列に適用されます。SET SQUEEZE を使用して、すべての列幅を設定することもできます。

ON

最も長いデータ値とフィールドタイトルの長い方を基準に、列幅を決定します。

OFF

最も長いデータ値と列タイトルの長い方を基準に、列幅を決定します。フィールドタイトルとフィールドフォーマットの長い方の長さになるまで、列幅にブランクを追加します。デフォルト値は OFF です。

width

UNITS 属性で指定した測定単位の列幅です。

最大長のデータ値がここで指定した列幅を超える場合、次のように表示されます。

フィールド	表示
文字	指定した幅に値を最大限入力し、省略を示すために感嘆符 (!) を続ける。
数値	フィールド値の代わりにアスタリスク (*) を使用する。

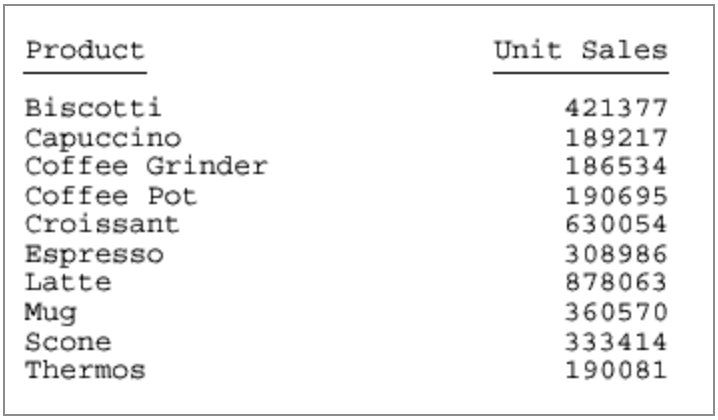
注意：SQUEEZE は OVER 句が生成したフィールドではサポートされません。

列幅の決定 (PDF)

次のリクエストは、SQUEEZE=2.5 を使用して、PDF レポート内の PRODUCT フィールドのデフォルト列幅を拡張します。この機能は、主として印刷レポートで使用します。画面の解像度によっては、列幅の表示は印刷とは異なる場合があります。

```
SET ONLINE-FMT = PDF
TABLE FILE GGSALES
SUM UNITS
BY PRODUCT
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, COLUMN=PRODUCT, SQUEEZE=2.5, $
ENDSTYLE
END
```

PDF レポートは次のとおりです。



Product	Unit Sales
Biscotti	421377
Capuccino	189217
Coffee Grinder	186534
Coffee Pot	190695
Croissant	630054
Espresso	308986
Latte	878063
Mug	360570
Scone	333414
Thermos	190081

列間隔の制御

デフォルト設定では、出力の幅に従ってレポートの各列は 1 つまたは 2 つのブランクで分割されます。SET SPACES または ON TABLE SET SPACES パラメータは、レポート内の列間のブランク数を制御します。

ACROSS 句の SPACES パラメータは、ACROSS フィールド間の距離を決定します。列の間隔は常にブランクが 1 つで、変更することはできません。

この機能は、HTML レポートに適用されます。STYLEMODE パラメータを FIXED に設定する必要があります。

列間隔の制御

この構文は、HTML レポートに適用されます。

プロシジャ内のすべてのリクエスト

```
SET SPACES = {n|AUTO}
```

単一リクエスト

```
ON TABLE SET SPACES {n|AUTO}
```

説明

n

レポートの列間ブランクの数を示す 1 から 8 の整数です。

AUTO

レポートフィールドを 1 つまたは 2 つのブランクで自動的に分割します。デフォルト値は AUTO です。

ACROSS フィールド列間隔の制御

次のリクエストは、ON TABLE SET SPACES で ACROSS を使用します。HTML の場合、ON TABLE SET STYLEMODE FIXED パラメータが必要です。

```
TABLE FILE CENTORD
SUM QUANTITY LINEPRICE ACROSS ORDER_NUM BY PLANT AS 'Plant'
WHERE ORDER_NUM EQ '28003' OR '28004'
ON TABLE SET SPACES 7
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLEMODE FIXED
END
```

ACROSS セットは、タイトルが Quantity および Line Total のフィールドで構成されます。各セットの間隔はブランク 7 つ分です。

	Order Number:			
	28003		28004	
Plant	Quantity:	Line Total	Quantity:	Line Total
BOS	.	.	6	$2,023.62
LA	5	$1,688.47	.	.

列順序の変更

BY フィールドのレポート内での表示順序を変更することができます。この機能は、ACROSS フィールドおよび OVER フィールドには適用されません。

列順序の変更

この構文は、PDF、HTML、XLSX、および EXL2K レポートに適用されます。XLSX FORMULA および EXL2K FORMULA フォーマットはサポートされません。

```
[TYPE=REPORT,] COLUMN=identifier, SEQUENCE=sequence, $
```

説明

TYPE=REPORT

列順序をレポート全体に適用します。デフォルト値であるため必須ではありません。

identifier

COLUMN 属性を使用して、列を選択します。この属性についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

sequence

選択した列の順序を表す数字です。

数字は、連続番号および 1 の増分値にする必要はありません。列の順序は、昇順です。NOPRINT フィールドは含まれません。

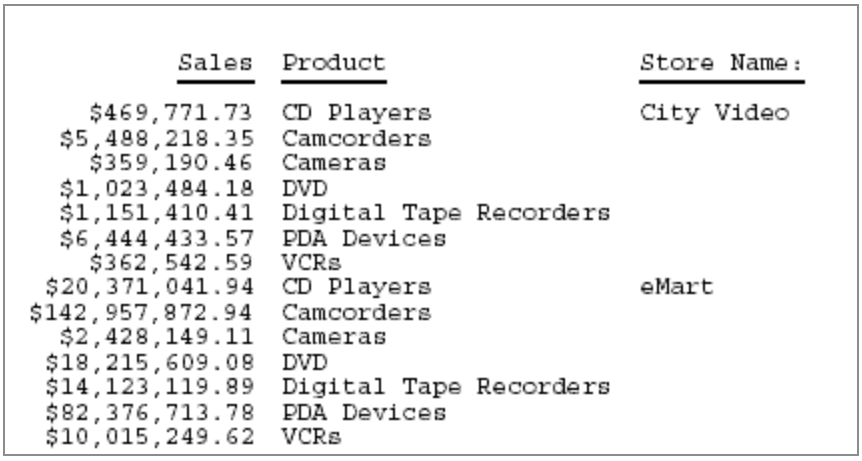
列順序の変更

次のリクエストは、SNAME が 1 番目、PRODCAT が 2 番目、LINEPRICE が 3 番目というように、通常レポート内に表示される順序に列を再整列します。

```
SET ONLINE-FMT = PDF
TABLE FILE CENTORD
SUM LINEPRICE AS 'Sales'
BY SNAME BY PRODCAT AS 'Product'
```

```
WHERE SNAME EQ 'eMart' OR 'City Video'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, COLUMN=SNAME, SEQUENCE=3, $
TYPE=REPORT, COLUMN=PRODCAT, SEQUENCE=2, $
TYPE=REPORT, COLUMN=LINEPRICE, SEQUENCE=1, $
ENDSTYLE
END
```

1 列目が LINEPRICE (Sales)、2 列目が PRODCAT (Product) (デフォルト設定と同一)、3 列目が SNAME (Store Name) となります。PDF レポートは次のとおりです。



Sales	Product	Store Name:
$469,771.73	CD Players	City Video
$5,488,218.35	Camcorders	
$359,190.46	Cameras	
$1,023,484.18	DVD	
$1,151,410.41	Digital Tape Recorders	
$6,444,433.57	PDA Devices	
$362,542.59	VCRs	
$20,371,041.94	CD Players	eMart
$142,957,872.94	Camcorders	
$2,428,149.11	Cameras	
$18,215,609.08	DVD	
$14,123,119.89	Digital Tape Recorders	
$82,376,713.78	PDA Devices	
$10,015,249.62	VCRs	

列の積み重ね

レポート幅の縮小や、マトリックスを作成して値の比較を簡単にする目的で、レポートの列を積み重ねることができます。列を積み重ねる場合、次のコマンドを使用することができます。

- FOLD-LINE
  - 変更頻度の低い BY フィールド列のブランクを減らします。リクエスト内に 16 個までの FOLD-LINE 句を含めることができます。
  - PDF および DHTML レポートで利用できますが、HTML、XLSX、EXL2K など、セルを基準とするフォーマットでは利用できません。
  - セルを基準とするスタイル機能 (例、BACKCOLOR、BORDERS) は、サポートされません。

    - 列の積み重ねでは、背景色を交互に表示する機能はサポートされません。
- OVER
    - 列を積み重ねます。ACROSS を使用し、レポートを横方向にソートしている場合は、特に読みやすくなります。OVER は、財務レポートを作成するときに役立ちます。詳細は、財務レポートの作成を参照してください。
    - 列の積み重ねでは、背景色を交互に表示する機能はサポートされません。

FOLD-LINE と OVER の相違は、FOLD-LINE は最初の行の下に (2 列目ではなく) 2 行目が、わずかにインデントが設定されて開始します。OVER は、指定した列の値を別の列の値に直接積み重ねます。FOLD-LINE および OVER は、同じリクエスト内で併用することができます。

FOLD-LINE による列の積み重ね

```
display_command fieldname ...FOLD-LINE fieldname ...
```

または

```
{ON|BY} fieldname FOLD-LINE
```

説明

display_command

表示コマンドです。表示フィールドの後で行が折り返される場合、補正は行われません。

fieldname

ON または BY フィールドの値が変わったところで、別の行に配置される表示フィールドまたはソートフィールドです。ソートフィールド上で折り返される場合は、行は前の行からブランク 2 つ分補正されます。

ON|BY

縦のソート句です。これらの句は同義です。

FOLD-LINE 使用による列の積み重ね

次の例は、FOLD-LINE を使用してレポートの幅を縮小する方法を示しています。この例では、ソートフィールド CATEGORY の値が変わったところで、列が積み重ねられます。

```
TABLE FILE GGSALES
SUM UNITS BUDUNITS
BY CATEGORY
ON CATEGORY FOLD-LINE
ON TABLE SET ONLINE-FMT PDF
ON TABLE SET PAGE-NUM OFF
END
```

レポートは次のとおりです。

```
Category
   Unit Sales  Budget Units

Coffee
      1376266       1385923
Food
      1384845       1377564
Gifts
       927880        931007
```

FOLD-LINE がない場合、レポートは次のようになります。

Category	Unit Sales	Budget Units
Coffee	1376266	1385923
Food	1384845	1377564
Gifts	927880	931007

OVER 使用による列の積み重ね

```
display_command fieldname1 OVER fieldname2 OVER fieldname3 ...
```

説明

display_command

表示コマンドです。

fieldname1, fieldname2, fieldname3

表示フィールドまたは一時項目 (COMPUTE) です。テキストフィールドは無効です。

OVER 使用による列の積み重ね

次のリクエストには、ACROSS 句が含まれており、HTML レポートを DEPARTMENT 別、横方向にソートします。列を積み重ねるために OVER 句を 2 つ使用します。

```
TABLE FILE EMPLOYEE
SUM GROSS OVER DED_AMT OVER
COMPUTE NET/D8.2M = GROSS - DED_AMT;
ACROSS DEPARTMENT
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

OVER を使用して、GROSS、DED_AMT、NET フィールド を積み重ねて見やすくします。

	DEPARTMENT	
	MIS	PRODUCTION
GROSS	$50,499.09	$50,922.38
DED_AMT	$28,187.11	$23,391.24
NET	$22,311.98	$27,531.14

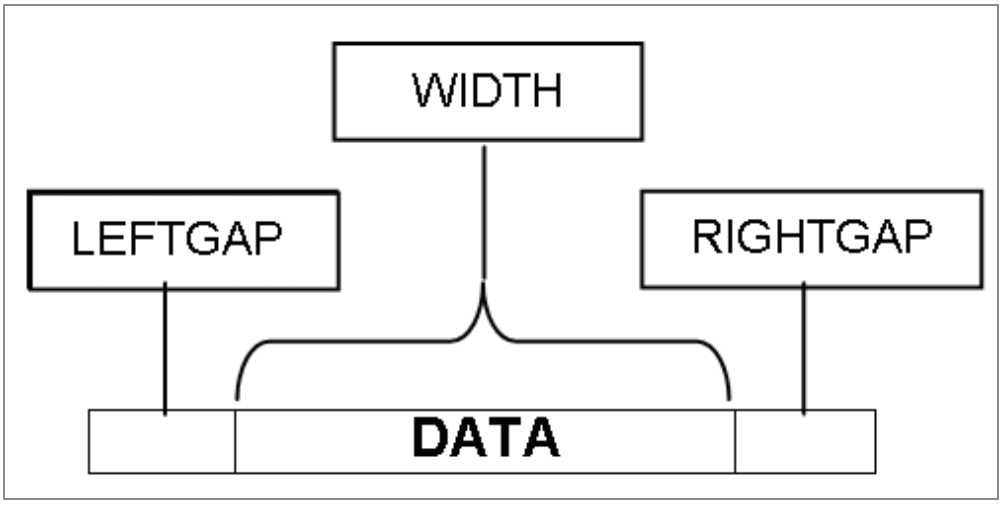
OVER を使用しない場合、HTML レポートは次のようになります。

DEPARTMENT					
MIS			PRODUCTION		
GROSS	DED_AMT	NET	GROSS	DED_AMT	NET
$50,499.09	$28,187.11	$22,311.98	$50,922.38	$23,391.24	$27,531.14

PDF レポートでの OVER による列の位置揃え

レポートに表示された複数列の間には、間隔が空けられます。スタイルシート属性の LEFTGAP および RIGHTGAP を使用して、列間の間隔のサイズを制御することができます。

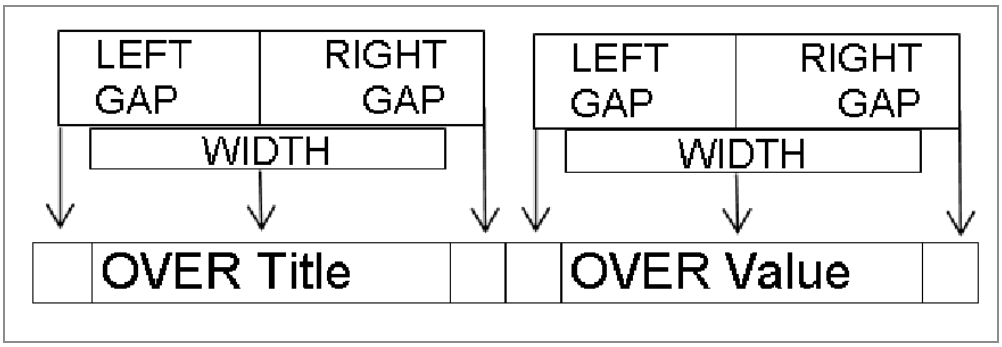
デフォルト設定では、列間の間隔は、レポートのフィールド用に確保された境界の外側に追加されます。このため、データ値のテキスト領域サイズは、フィールドの幅 (圧縮値) によって定義されます。列間の間隔幅は、計算されません。境界と背景色の定義に使用される境界領域は、データの幅、左の間隔、右の間隔の合計サイズに基づいて決定されます。



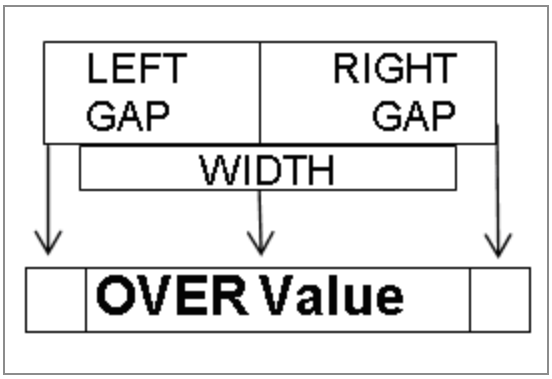
OVER 句を使用するレポートで、フィールドの位置を揃える場合、フィールドの境界の外側の間隔を考慮する必要があります。

この機能は、ブランクの AS 名 (フィールドタイトル) を使用する複数行レポートの作成をサポートすることを目的にしています。特に記述がない限り、これらの機能は、ブランク以外のタイトルで動作し、フィールドタイトルがブランクの場合の位置揃えはサポートされません。

デフォルト設定では、フィールドタイトルは、OVER を使用するレポートでフィールド値の左側に配置されます。下図のように、「OVER Title」、「OVER Value」のそれぞれは、LEFTGAP、WIDTH、RIGHTGAP という 3 つのパラメータの組み合わせで測定されます。



OVER の指定とブランクの AS 名によって、データ値のそれぞれはデータセルを形成し、レポートのデータ行内で、行と列の構成に使用することができます。データ値の位置を、上部に配置したフィールドの下に揃えるには、上部のデータの幅を記述する下部のフィールドの幅に加えて、その間に存在する左右の間隔すべてを計算に含める必要があります。



OVER を使用するレポートで、データと見出しの位置を揃える場合、各列のサイズ計算は複雑になります。各列サイズの計算には、それ以外に、外側の左右の間隔の計算も含める必要があります。さらに、使用する行の列数の増加とともに、これらの間隔も増加します。

スタイルシート属性の GAPINTERNAL=ON を使用して、PDF レポートの列の境界内に、間隔を空けることができます。OVER 句を使用して複数行を作成するレポートで、フィールドと見出しの位置を揃えるには、この機能を使用した方が簡単です。

注意：現在のバージョンでは、OVER を SQUEEZE とともに使用することができます。

レポートの列間隔の制御

```
TYPE=REPORT, GAPINTERNAL={OFF|ON}
```

説明

OFF

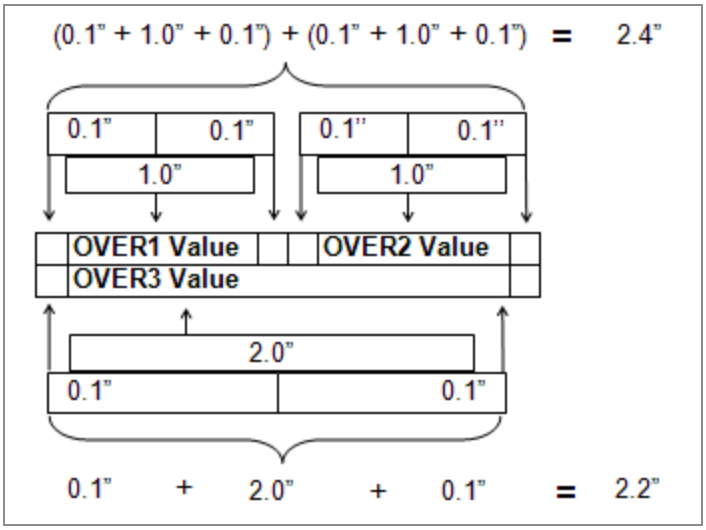
定義済みのフィールド幅の外側に、左右の間隔を指定します。デフォルト値は OFF です。

ON

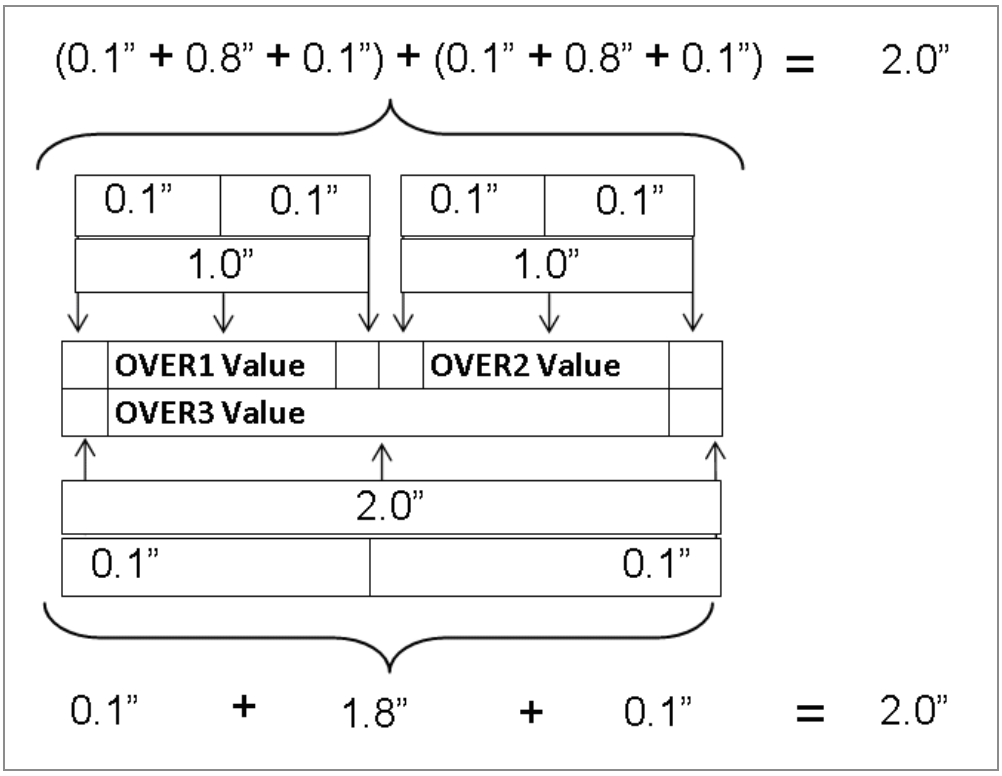
定義済みのフィールド幅の内側に、左右の間隔を指定します。

外側と内側の間隔の比較

GAPINTERNAL=OFF の場合、積み重なるフィールドの幅を定義する際は、フィールドの幅と同時に、左右の間隔の合計を計算に含める必要があります。



GAPINTERNAL=ON の場合、使用するデータセルまたはフィールドが占めるスペース全体は、定義済みの WIDTH で表されます。これにより、OVER 値が行幅を超える場合も、増加による影響を受けなくなります。



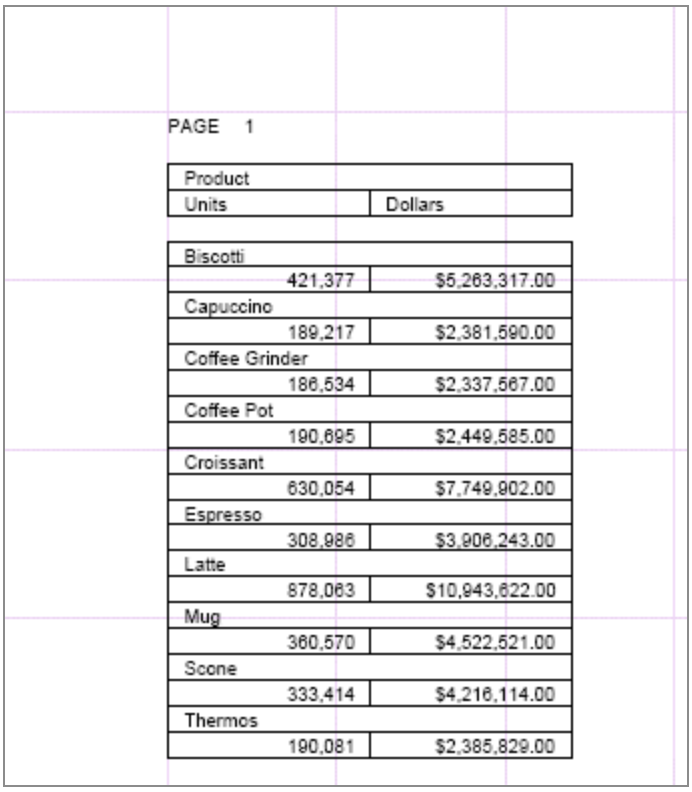
レポートでの GAPINTERNAL の使用

次のリクエストは、GGSALES データソースに対して実行され、PRODUCT を UNITS フィールドと DOLLARS の上部に配置し、GAPINTERNAL を OFF に設定します。

```
SET LAYOUTGRID=ON
TABLE FILE GGSALES
"Product<+0>"
"Units<+0>Dollars"
SUM
PRODUCT AS ''
OVER
UNITS/D8C AS '' DOLLARS/D12.2CM AS ''
BY PRODUCT NOPRINT
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, SQUEEZE=ON, FONT=ARIAL, SIZE=8, LEFTMARGIN=1, TOPMARGIN=1,
 LEFTGAP=.1, RIGHTGAP=.1, GAPINTERNAL=OFF, $
TYPE=REPORT, BORDER=ON, $
TYPE=HEADING, BORDERALL=ON, $
TYPE=HEADING, LINE=1, ITEM=1, POSITION = PRODUCT, $
TYPE=HEADING, LINE=2, ITEM=1, POSITION = UNITS, $
```

```
TYPE=HEADING, LINE=2, ITEM=2, POSITION = DOLLARS, $
TYPE=REPORT, COLUMN=PRODUCT(2),   SQUEEZE=2,  $
TYPE=REPORT, COLUMN=UNITS, SQUEEZE=1, $
TYPE=REPORT, COLUMN=DOLLARS, SQUEEZE=1, $
ENDSTYLE
END
```

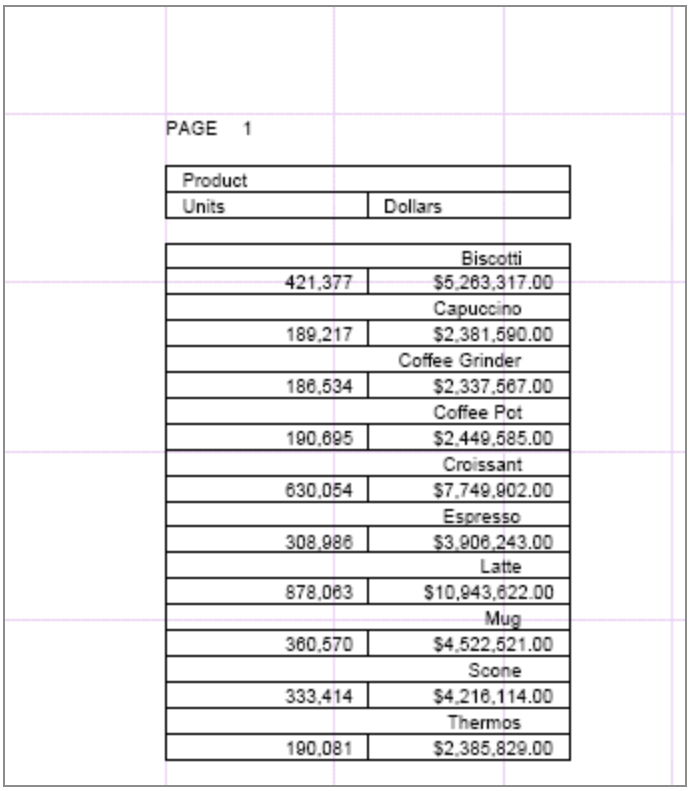
UNITS フィールドと DOLLARS フィールドの幅は、それぞれ 1 インチ、PRODUCT フィールドの幅は、2 インチに指定されています。GAPINTERNAL=OFF の場合、レイアウトグリッド (LAYOUTGRID) によって、フィールドの配置に使用する幅は、リクエストで指定された幅よりも大きいことが分かります。追加のスペースは、外側の左間隔 (LEFTGAP) と右間隔 (RIGHTGAP) によって指定されます。



PAGE 1

Product	
Units	Dollars
Biscotti	
421,377	$5,263,317.00
Capuccino	
189,217	$2,381,590.00
Coffee Grinder	
186,534	$2,337,567.00
Coffee Pot	
190,695	$2,449,585.00
Croissant	
630,054	$7,749,902.00
Espresso	
308,986	$3,906,243.00
Latte	
878,063	$10,943,622.00
Mug	
360,570	$4,522,521.00
Scone	
333,414	$4,216,114.00
Thermos	
190,081	$2,385,829.00

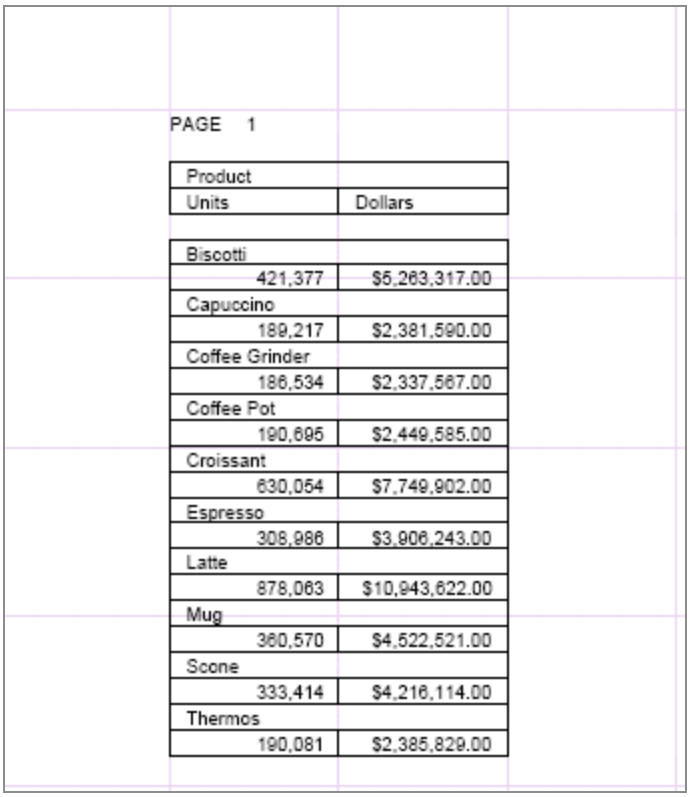
スタイルシートの SQUEEZE=ON 属性が指定されているため、見出しの境界線は、レポートの右側に揃えられます。見出しの位置を揃えるため、追加のブランクが挿入されています。PRODUCTS フィールドのスタイルシート指定を JUSTIFY=RIGHT に変更すると、追加のブランクにより、PRODUCTS フィールドの値は DOLLARS フィールドの値とは別の位置に配置されます。



PAGE 1

Product	
Units	Dollars
	Biscotti
421,377	$5,263,317.00
	Capuccino
189,217	$2,381,590.00
	Coffee Grinder
186,534	$2,337,567.00
	Coffee Pot
190,695	$2,449,585.00
	Croissant
630,054	$7,749,902.00
	Espresso
308,986	$3,906,243.00
	Latte
878,063	$10,943,622.00
	Mug
360,570	$4,522,521.00
	Scone
333,414	$4,216,114.00
	Thermos
190,081	$2,385,829.00

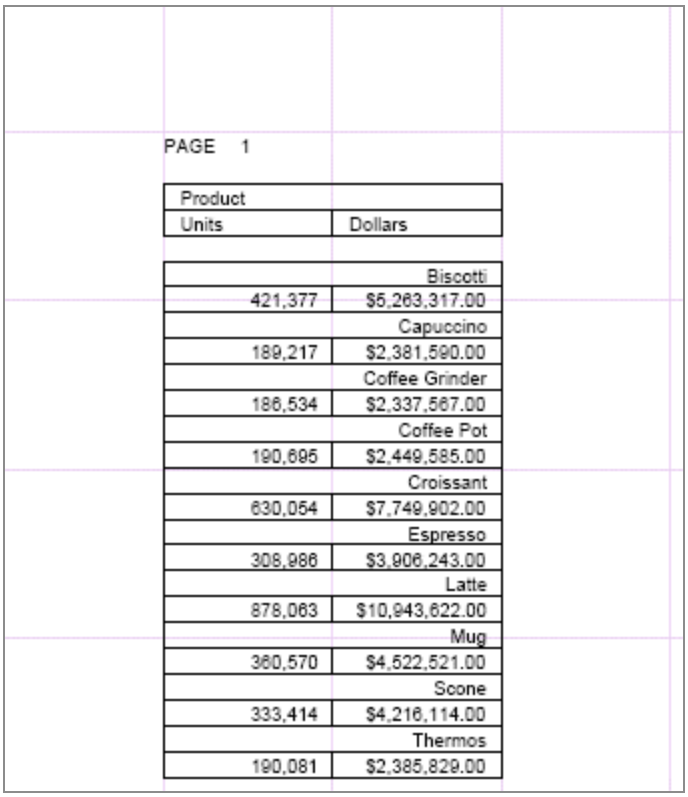
スタイルシートの指定を GAPINTERNAL=ON に変更すると、内側の間隔が設定され、指定した幅に含められることから、指定した幅が使用されます。



PAGE 1

Product	
Units	Dollars
Biscotti	
421,377	$5,263,317.00
Capuccino	
189,217	$2,381,590.00
Coffee Grinder	
186,534	$2,337,567.00
Coffee Pot	
190,695	$2,449,585.00
Croissant	
630,054	$7,749,902.00
Espresso	
308,986	$3,906,243.00
Latte	
878,063	$10,943,622.00
Mug	
360,570	$4,522,521.00
Scone	
333,414	$4,216,114.00
Thermos	
190,081	$2,385,829.00

次のレポートでは、PRODUCT の値は JUSTIFY=RIGHT として定義されていますが、値は適切に配置されています。



PAGE 1

Product	
Units	Dollars
	Biscotti
421,377	$5,263,317.00
	Capuccino
189,217	$2,381,590.00
	Coffee Grinder
186,534	$2,337,567.00
	Coffee Pot
190,695	$2,449,585.00
	Croissant
630,054	$7,749,902.00
	Espresso
308,986	$3,906,243.00
	Latte
878,063	$10,943,622.00
	Mug
360,570	$4,522,521.00
	Scone
333,414	$4,216,114.00
	Thermos
190,081	$2,385,829.00

列の配置

レポート内の列の絶対または相対開始位置を指定することができます。相対開始位置は、最終列の右のバイト数です。

この機能をHTML レポートで使用する場合は、STYLEMODE パラメータを FIXED に設定します。

列の配置

```
field IN {n|+n}
```

説明

field

配置する列です。

n

列の絶対位置を示す数字です。

ACROSS で使用される場合、*n* は ACROSS セットの開始位置を指定します。

FOLD-LINE または OVER で使用される場合、*n* は参照フィールドが出現する行に適用されます。

+n

フィールドの相対位置を示す数字です。n の値は、最終フィールドの右のバイト数です。

列の配置

次のリクエストは、レポート内の 3 列の絶対位置を指定します。HTML の場合、ON TABLE SET STYLEMODE FIXED パラメータが必要です。

```
TABLE FILE CENTQA
SUM CNT.PROBNUM IN 1 AS 'Total #,Problems'
SUM CNT.PROBNUM IN 45 AS '# Problems,by Product'
BY PLANT NOPRINT BY PRODNAME IN 15
WHERE PLANT EQ 'ORL'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLEMODE FIXED
END
```

読みやすくするために、列間に間隔が取られています。

Total # Problems	Product Name:	# Problems by Product
95	120 VHS-C Camcorder 40 X	7
	2 Hd VCR LCD Menu	6
	250 8MM Camcorder 40 X	10
	650DL Digital Camcorder 150 X	6
	AR2 35MM Camera 8 X	17
	AR3 35MM Camera 10 X	9
	Combo Player - 4 Hd VCR + DVD	12
	DVD Upgrade Unit for Cent. VCR	9
	QX Portable CD Player	9
	R5 Micro Digital Tape Recorder	3
	ZC Digital PDA - Standard	7

ACROSS フィールドの配置

次のリクエストでは、ACROSS フィールド PLANT に IN 句を使用して、列の開始位置を指定しています。PROBNUM 間にブランクを追加するために、相対配置も使用します。ON TABLE SET STYLEMODE FIXED パラメータは、HTML レポートで必要です。

```
TABLE FILE CENTQA
SUM PROBNUM IN +8
ACROSS PLANT IN 35
BY PROBLEM_CATEGORY
WHERE PLANT EQ 'BOS' OR 'ORL'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLEMODE FIXED
END
```

ACROSS セットは、35 桁目から始まり、ACROSS 内のデータフィールドの間に 8 つの追加ブランクがあります。

	Plant	
	BOS	ORL
Problem Category		
Incorrect Labeling	39921	19877
Mechanical Failure	19564	13792
Missing Component	38285	18709
Physical Damage	65497	16175
Power Failure	41472	14277
Remote Failure	59655	8635

OVER フィールドの配置

次のリクエストは、OVER を使用してフィールドを積み重ね、IN を使用して配置します。

```
TABLE FILE EMPLOYEE
SUM GROSS IN 40
OVER DED_AMT IN 40
BY DEPARTMENT BY LAST_NAME IN 20
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLEMODE FIXED
END
```

レポート内で、GROSS と DED_AMT は 40 桁目から始まり、積み重ねられます。LAST_NAME は、20 桁目から始まります。

DEPARTMENT	LAST_NAME		
MIS	BLACKWOOD	GROSS	$9,075.00
		DED_AMT	$6,307.00
	CROSS	GROSS	$22,013.75
		DED_AMT	$15,377.40
	GREENSPAN	GROSS	$2,970.84
		DED_AMT	$505.04
	JONES	GROSS	$6,099.50
		DED_AMT	$2,866.18
	MCCOY	GROSS	$1,540.00
		DED_AMT	$458.69
	SMITH	GROSS	$8,800.00
		DED_AMT	$2,672.80
PRODUCTION	BANNING	GROSS	$2,475.00
		DED_AMT	$1,427.24
	IRVING	GROSS	$17,094.00
		DED_AMT	$11,949.44
	MCKNIGHT	GROSS	$9,130.00
		DED_AMT	$3,593.92
	ROMANS	GROSS	$7,040.00
		DED_AMT	$3,507.88
	SMITH	GROSS	$6,183.36
		DED_AMT	$1,104.96
	STEVENS	GROSS	$9,000.02
		DED_AMT	$1,807.80

フィールドの非表示

リクエストには、特定の結果を作成するためのフィールドを含めることができます。たとえば、データを整列するために、ソートフィールドを指定することができます。ただし、タイトルまたはフィールドの値がレポートのほかの場所に表示される場合、タイトルまたはフィールドの値を表示したくない場合があります。NOPRINT 句 (SUP-PRINT と同義語) は、フィールドを非表示にします。

フィールド非表示コマンド

コマンド	説明	適用先
`NOPRINT または SUP-PRINT`	フィールドの非表示	HTML PDF

フィールドの非表示

```
display_command fieldname {NOPRINT|SUP-PRINT}
```

または

```
{ON|BY} fieldname {NOPRINT|SUP-PRINT}
```

説明

display_command

表示コマンドです。

fieldname

表示フィールドまたはソートフィールドです。フィールド値は使用されますが、これらは表示されません。HOLD ファイルには、非表示 BY フィールドの値は含められません。

NOPRINT 付きの一時項目 (COMPUTE) の場合、次の一時項目 (COMPUTE) の前に COMPUTE を繰り返します。

NOPRINT|SUP-PRINT

フィールドの非表示これらの句は同義です。

ON|BY

縦のソート句です。これらの句は同義です。

ソートフィールドの非表示

次のリクエストは、データを CITY でソートします。ページ見出しに CITY の値が含まれているため、ソートフィールドは非表示にします。

```
TABLE FILE SALES
HEADING
"Page <TABPAGENO"
"SALES REPORT FOR <CITY"
PRINT UNIT_SOLD AND DELIVER_AMT
BY CITY PAGE-BREAK NOPRINT
BY PROD_CODE
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

ページ見出しは、データが適用される 都市名を識別します。

Page 1
SALES REPORT FOR NEW YORK

PROD_CODE	UNIT_SOLD	DELIVER_AMT
B10	30	30
B17	20	40
B20	15	30
C17	12	10
D12	20	30
E1	30	25
E3	35	25

Page 2
SALES REPORT FOR NEWARK

PROD_CODE	UNIT_SOLD	DELIVER_AMT
B10	13	30
B12	29	30

NOPRINT がない場合、不要な都市名が繰り返されます。

Page 1
SALES REPORT FOR NEW YORK

CITY	PROD_CODE	UNIT_SOLD	DELIVER_AMT
NEW YORK	B10	30	30
	B17	20	40
	B20	15	30
	C17	12	10
	D12	20	30
	E1	30	25
	E3	35	25

Page 2
SALES REPORT FOR NEWARK

CITY	PROD_CODE	UNIT_SOLD	DELIVER_AMT
NEWARK	B10	13	30
	B12	29	30

中間合計によるソートフィールドの非表示

次のリクエストは、ソートフィールド CATEGORY の各値の中間合計を生成しますが、これらは表示されません。

```
TABLE FILE GGSALES
SUM UNITS BY CATEGORY
BY PRODUCT
ON CATEGORY SUB-TOTAL SUP-PRINT PAGE-BREAK
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

デフォルトの中間合計行には、CATEGORY の各値 (例、合計 Coffee) が表示されます。

Product	Unit Sales
Capuccino	189217
Espresso	308986
Latte	878063
*TOTAL Coffee	1376266

Product	Unit Sales
Biscotti	421377
Croissant	630054
Scone	333414
*TOTAL Food	1384845

SUP-PRINT を使用しない場合、レポートには不要な CATEGORY 値が繰り返し表示されます。

Category	Product	Unit Sales
Coffee	Capuccino	189217
	Espresso	308986
	Latte	878063
*TOTAL Coffee		1376266

Category	Product	Unit Sales
Food	Biscotti	421377
	Croissant	630054
	Scone	333414
*TOTAL Food		1384845

アルファベット順ソート

次のリクエストは、LAST_NAME をアルファベット順にソートしますが、LAST_NAME のソートフィールドを非表示にすることで、データの重複を防ぎます。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME
BY LAST_NAME NOPRINT
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

LAST_NAME はアルファベット順に整列されます。

改ページの挿入

PAGE-BREAK コマンドを使用すると、指定した BY フィールドの値が変わるたびに、新しいページが生成されます。これにより、関連する情報が複数ページにわたって表示されないようになります。改ページを使用すると、フィールドタイトルとすべてのページ見出しは、それぞれの新しいページの最上部に表示されます。リクエストで PAGE-BREAK を使用すると、GRANDTOTAL が自動的にページに表示されます。

レポート出力を HOLD、SAVE、または SAVB ファイルに保存する場合、PAGE-BREAK は適用されません。

HTML レポートで PAGE-BREAK を使用すると、同一の Web ページ上にレポートの新しいセクションが作成され、各セクションにフィールドタイトルおよび連続ページ番号が表示され

ます。PAGE-BREAK 自体が新しい Web ページを作成することはありません。HTML レポート内に複数の Web ページを作成するには、次の方法があります。

- ソートフィールド値ごとに、別のページが作成されるよう、レポートを分割 (バースト) する。詳細は、複数 HTML ファイルへのレポート分割を参照してください。
- SET STYLEMODE=PAGED を PAGE-BREAK と併用する。これは、ReportCaster でレポートを配信する際に役立ちます。
- Web ビューアにアクセスする SET WEBVIEWER=ON を PAGE-BREAK と併用する。詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

改ページコマンド

コマンド	説明	適用先
`PAGE-BREAK`	新しいページを生成します。	HTML PDF
`NOSPLIT`	改ページを行いません。	PDF
`SET LINES`	ブラウザページとレポートページの外観が一致します。	HTML

複数テーブル HTML レポートの使用

- リクエストで SET LINES または PAGE-BREAK を使用して、レポート内の区切り位置を制御することができます。
- ON *sortfield* PAGE-BREAK または BY *sortfield* PAGE-BREAK は、SET LINES コマンドを上書きし、ソートフィールドの値が変わるたびにレポートを複数の HTML テーブルに分割します。
- フィールドタイトルは、各 PAGE-BREAK または SET LINES パラメータに従い生成され

ます。

- レポートが複数の HTML テーブルに分割されるときは、ブラウザはそれ自体のアルゴリズムに従い各テーブルを表示します。SQUEEZE を OFF、または WRAP を OFF に設定し、HTML テーブルの位置をすべてのページで統一します。

改ページの挿入

```
{ON|BY} fieldname PAGE-BREAK [REPAGE] [WHEN expression;]
```

説明

ON|BY

縦のソート句です。これらの句は同義です。

fieldname

改ページを発生させるソートフィールドです。改ページを発生させる最下位のソートフィールドを指定します。上位のソートフィールドの値が変わるたびに改ページが自動的に挿入されます。

REPAGE

改ページごとにページ番号を 1 にリセットします。WHEN と組み合わせる場合は、ページ番号は、WHEN 条件が満たされるたびに 1 にリセットされます。

WHEN expression

レポートに条件付き改ページを挿入するための論理式を指定します。詳細は、レポートフォーマット設定の管理を参照してください。

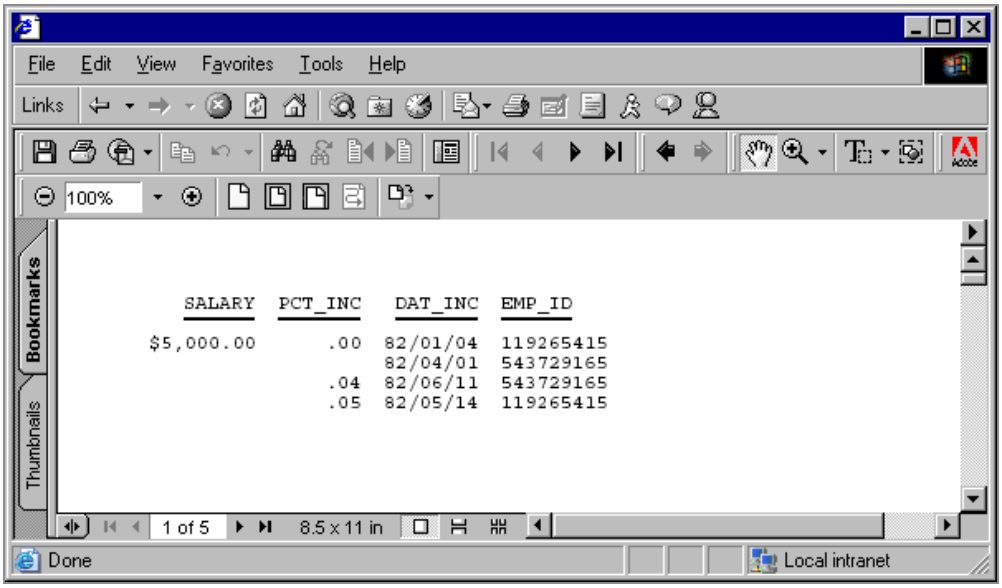
改ページの挿入

次のリクエストは、ソートフィールド SALARY の値が変わるたびに新しいページを生成します。

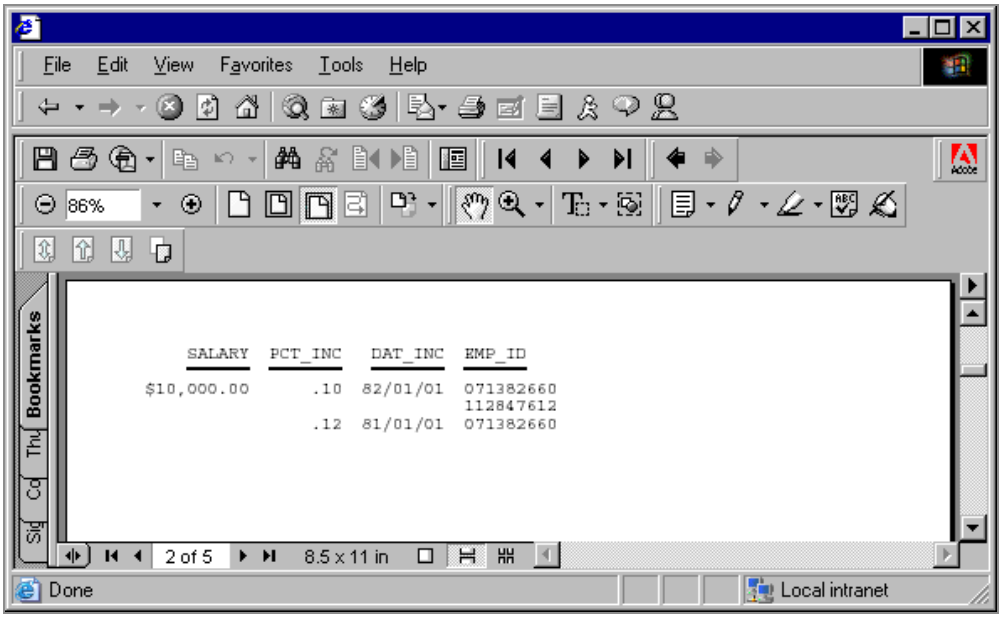
```
TABLE FILE EMPLOYEE
PRINT EMP_ID
BY SALARY IN-GROUPS-OF 5000
```

```
BY PCT_INC BY DAT_INC
ON SALARY PAGE-BREAK
ON TABLE SET ONLINE-FMT PDF
ON TABLE SET PAGE-NUM OFF
END
```

改ページが発生する位置を示すために、レポートの最初の 2 ページが表示されています。



2 ページ目は次のとおりです。



複数テーブル HTML レポートの表示

次のリクエストでは、各ページは別の HTML テーブルとしてブラウザに返されます。全ページのテーブルの位置揃えを統一するために、SQUEEZE は OFF に設定されています。

```
SET STYLEMODE = PAGED
SET LINES = 12
TABLE FILE CENTORD
HEADING
"SALES OVER $200,000"
PRINT LINEPRICE AS 'Sales'
BY SNAME BY ORDER_NUM
WHERE LINEPRICE GT 200000
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, SQUEEZE=OFF, $
ENDSTYLE
END
```

次の 2 ページのレポートは、位置揃えが統一されています。

PAGE 10

SALES OVER $200,000

Store Name:	Order Number:	Sales
Audio Expert	59079	$241,754.82
	59487	$299,272.98
	59515	$241,570.90
	74080	$200,724.66
	74549	$326,165.31
	74557	$267,960.68

PAGE 11

SALES OVER $200,000

Store Name:	Order Number:	Sales
Audio Expert	74610	$222,234.35
	76810	$286,025.60
	77790	$504,891.18
City Video	54691	$256,552.45
	74691	$250,675.75
Consumer Merchandise	37395	$283,885.51

SQUEEZE が ON に設定された場合、同じ 2 ページのレポートの位置揃えは統一されません。

PAGE 10

SALES OVER $200,000

Store Name:	Order Number:	Sales
Audio Expert	59079	$241,754.82
	59487	$299,272.98
	59515	$241,570.90
	74080	$200,724.66
	74549	$326,165.31
	74557	$267,960.68

PAGE 11

SALES OVER $200,000

Store Name:	Order Number:	Sales
Audio Expert	74610	$222,234.35
	76810	$286,025.60
	77790	$504,891.18
City Video	54691	$256,552.45
	74691	$250,675.75
Consumer Merchandise	37395	$283,885.51

不要な分割の防止

ソートフィールドによる論理グループの途中で改ページが発生し、グループに関連する行が次のページまたはウィンドウ内に独立して表示される場合があります。この種の改ページを防止するには、NOSPLIT オプションを使用します。これにより、ソートフィールドの値が変更されたところで、新しい値に関連する行の合計数全体が新しいページに表示されるようになります。ソート項目を指定している見出し、ソート項目を指定している脚注、中間合計が設定してあれば、これらも同時に新しいページに表示されます。

この機能は、PDF レポートに適用されます。

NOSPLIT を PAGE-BREAK と併用する場合、PAGE-BREAK は上位のソートフィールドに適用する必要があります。それ以外では、NOSPLIT は無視されます。レポート出力が HOLD、SAVE、または SAVB ファイルに保存される場合、NOSPLIT も無視されます。NOSPLIT は、TABLEF コマンドと互換性がないため、エラーメッセージが生成されます。

不要な分割の防止

この構文は、PDF レポートに適用されます。NOSPLIT は、各リクエストで 1 回だけ使用できます。

```
{ON|BY} fieldname NOSPLIT
```

説明

ON|BY

縦のソート句です。これらの句は同義です。

fieldname

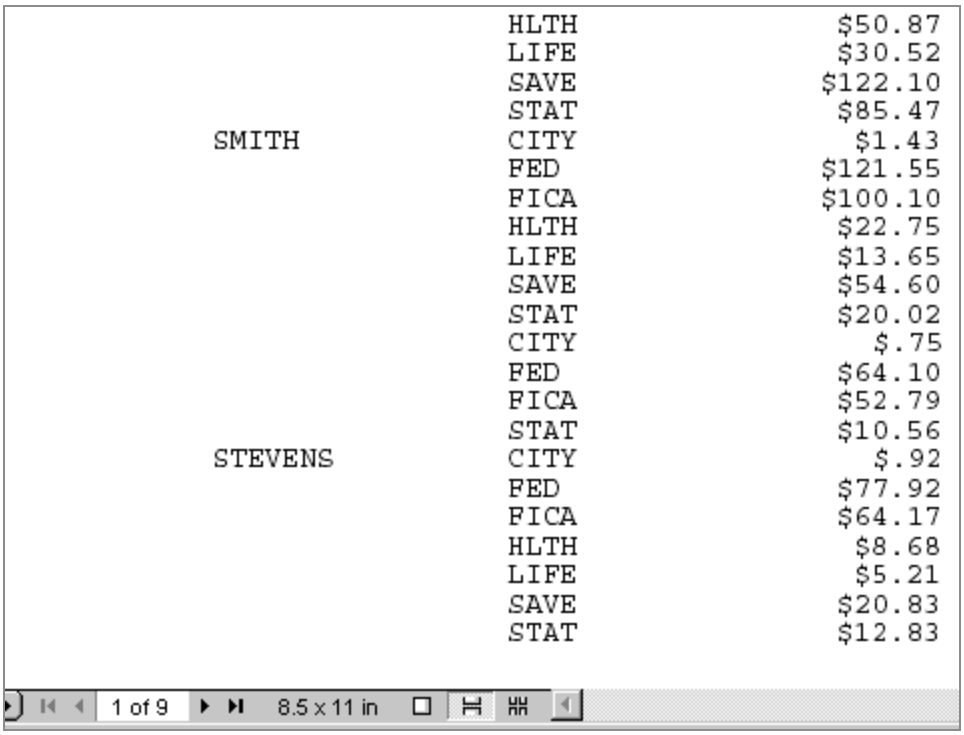
同一ページに表示するソートグループのソートフィールド名です。

不要な分割の防止

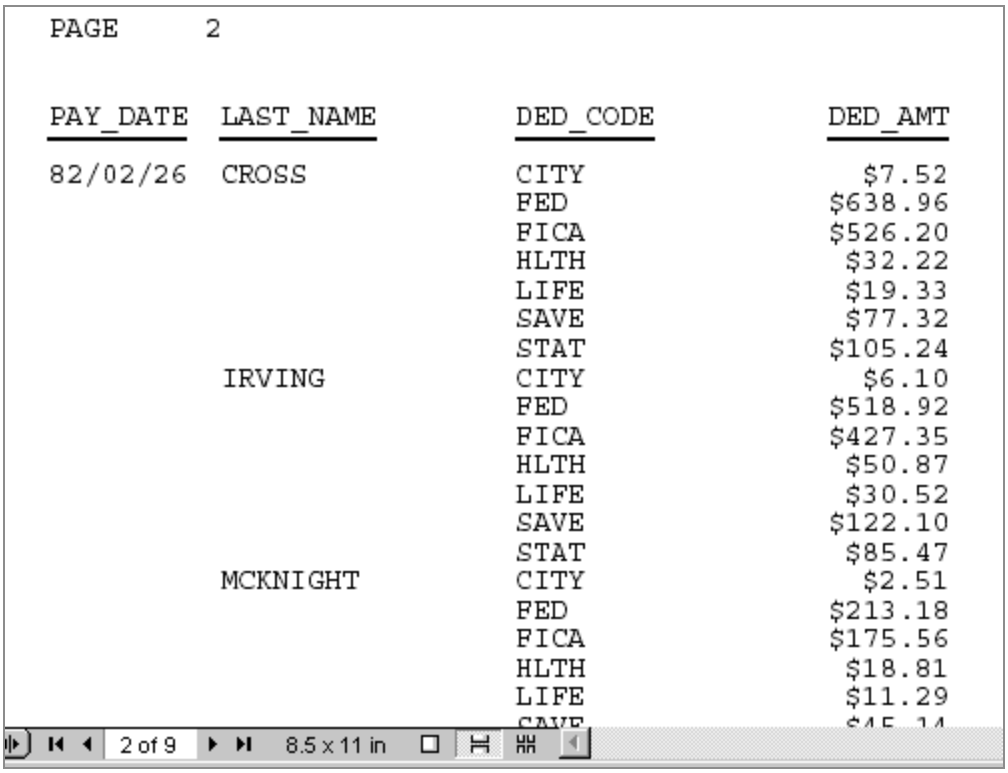
次のリクエストは、NOSPLIT を使用して、関連する情報を同一ページに表示します。

```
SET ONLINE-FMT = PDF
TABLE FILE EMPLOYEE
PRINT DED_CODE AND DED_AMT
BY PAY_DATE BY LAST_NAME
ON LAST_NAME NOSPLIT
END
```

LAST_NAME の値が STEVENS から CROSS に変わるところで、CROSS に関連する行を現行ページに収めることはできません。NOSPLIT を使用することで、CROSS に関連する行を次のページに表示します。



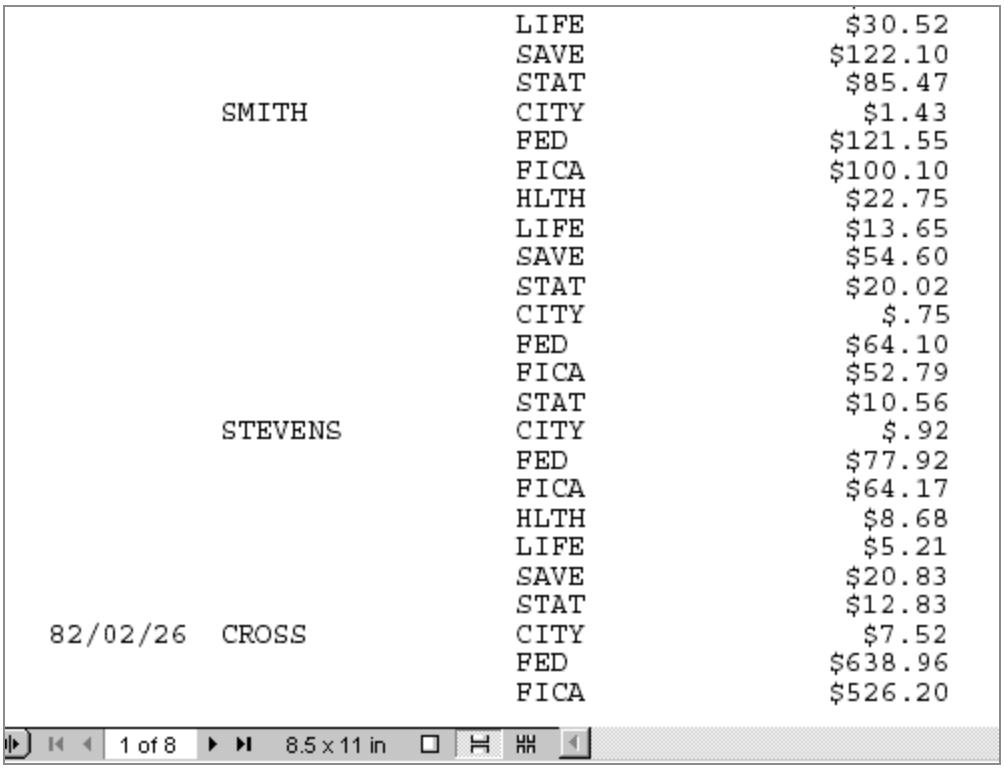
	HLTH	$50.87
	LIFE	$30.52
	SAVE	$122.10
	STAT	$85.47
SMITH	CITY	$1.43
	FED	$121.55
	FICA	$100.10
	HLTH	$22.75
	LIFE	$13.65
	SAVE	$54.60
	STAT	$20.02
	CITY	$.75
	FED	$64.10
	FICA	$52.79
	STAT	$10.56
STEVENS	CITY	$.92
	FED	$77.92
	FICA	$64.17
	HLTH	$8.68
	LIFE	$5.21
	SAVE	$20.83
	STAT	$12.83



PAGE 2

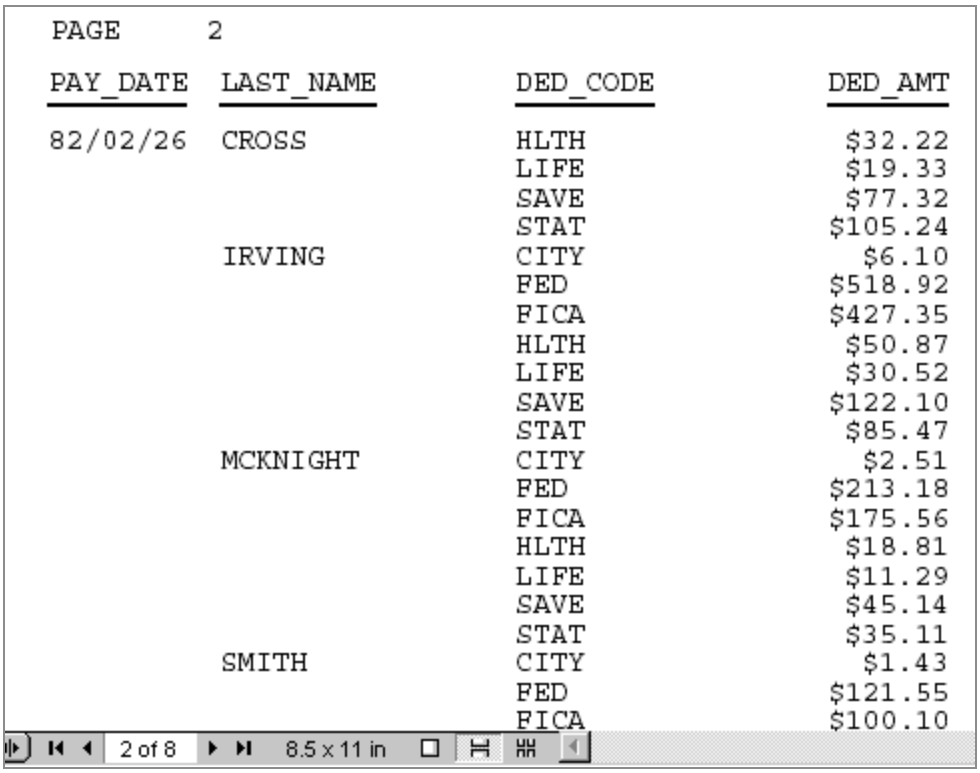
PAY_DATE	LAST_NAME	DED_CODE	DED_AMT
82/02/26	CROSS	CITY	$7.52
		FED	$638.96
		FICA	$526.20
		HLTH	$32.22
		LIFE	$19.33
		SAVE	$77.32
		STAT	$105.24
	IRVING	CITY	$6.10
		FED	$518.92
		FICA	$427.35
		HLTH	$50.87
		LIFE	$30.52
		SAVE	$122.10
		STAT	$85.47
	MCKNIGHT	CITY	$2.51
		FED	$213.18
		FICA	$175.56
		HLTH	$18.81
		LIFE	$11.29

NOSPLIT を使用しない場合、CROSS の情報は最初のページと 2 ページ目にまたがります。



		LIFE	$30.52
		SAVE	$122.10
		STAT	$85.47
	SMITH	CITY	$1.43
		FED	$121.55
		FICA	$100.10
		HLTH	$22.75
		LIFE	$13.65
		SAVE	$54.60
		STAT	$20.02
		CITY	$.75
		FED	$64.10
		FICA	$52.79
		STAT	$10.56
	STEVENS	CITY	$.92
		FED	$77.92
		FICA	$64.17
		HLTH	$8.68
		LIFE	$5.21
		SAVE	$20.83
		STAT	$12.83
82/02/26	CROSS	CITY	$7.52
		FED	$638.96
		FICA	$526.20

1 of 8 8.5 x 11 in



PAGE 2

PAY_DATE	LAST_NAME	DED_CODE	DED_AMT
82/02/26	CROSS	HLTH	$32.22
		LIFE	$19.33
		SAVE	$77.32
		STAT	$105.24
	IRVING	CITY	$6.10
		FED	$518.92
		FICA	$427.35
		HLTH	$50.87
		LIFE	$30.52
		SAVE	$122.10
		STAT	$85.47
	MCKNIGHT	CITY	$2.51
		FED	$213.18
		FICA	$175.56
		HLTH	$18.81
		LIFE	$11.29
		SAVE	$45.14
		STAT	$35.11
	SMITH	CITY	$1.43
		FED	$121.55
		FICA	$100.10

2 of 8 8.5 x 11 in

ページ番号の挿入

デフォルト設定では、レポートページの最初の 2 行は予約されています。最初の行はページ番号を左上の隅に表示し、2 行目をブランクにします。

注意：このセクションに記載されている機能は、複合レポートには適用されません。

次の処理を実行することができます。

- 現在のページ番号を含むシステム変数 TABPAGENO を使用して、ページ番号のデフォルト位置を変更できます。
- TABLASTPAGE システム変数を使用して、総ページ数を挿入します。
- FOCFIRSTPAGE パラメータを使用して、最初のページに任意の番号を割り当てます。
- デフォルトページ番号を非表示にします。

注意：条件付きスタイル設定 (WHEN) を使用してスタイルを定義する際に、TABPAGENO および TABLASTPAGE 変数を使用することはできません。

米国リハビリテーション法 (U. S. Rehabilitation Act) 第 508 条 (Section 508) アクセシビリティを有効にする場合、デフォルトページ番号は HTML テーブルには含められなくなります。

ページ番号コマンド

コマンド	説明	適用先
`<BYLASTPAGE`	REPAGE とともに使用します。REPAGE オプションが指定されているソートグループ内に総ページ数を挿入します。	HTML PDF PPTX
`REPAGE`	ページ番号を 1 にリセットします。	HTML PDF PPTX

コマンド	説明	適用先
`<TABPAGENO`	現在のページ番号を挿入します。TABPAGENO は、デフォルトページ番号を非表示にし、ページの最初の 2 行をブランクにします。	HTML PDF PPTX
`<TABLASTPAGE`	レポート内に総ページ数を挿入します。	HTML PDF PPTX
`SET FOCFIRSTPAGE`	指定したページ番号を先頭ページに割り当てます。	HTML PDF PPTX
`SET PAGE-NUM`	ページ番号の表示を制御します。	HTML PDF

現在のページ番号の挿入

現在のページ番号を追加するには、リクエストに次のコマンドを追加します。

```
<TABPAGENO
```

ソート項目を指定している脚注への現在のページ番号の挿入

次のリクエストは、ソートフィールド REGION の値が変わるたびに、新しいページを生成します。ソート項目を指定している脚注にページ番号を挿入するために、TABPAGENO を使用します。

```
TABLE FILE GGSALES
SUM BUDDOLLARS
BY REGION BY ST BY CITY
ON REGION PAGE-BREAK SUBFOOT
"Sales Quota for <REGION Cities"
"Page <TABPAGENO"
ON TABLE SET ONLINE-FMT PDF
END
```

次は出力結果の 1 ページ目です。

Region	State	City	Budget Dollars
Midwest	IL	Chicago	3866856
	MO	St. Louis	3646838
	TX	Houston	3680679

Sales Quota for Midwest Cities
Page 1

総ページ数の挿入

レポート内に総ページ数を挿入するには、<TABLASTPAGE システム変数を使用できます。たとえば、レポートに「Page 1 of 5」という脚注を追加する場合、<TABPAGENO システム変数とともに <TABLASTPAGE システム変数を使用できます。

総ページ数の挿入

総ページ数を挿入するには、リクエストに以下を追加します。

```
<TABLASTPAGE
```

TABLASTPAGE 使用上の注意

- TABLASTPAGE は FOCFIRSTPAGE または REPAGE コマンド内の変更を調整しません。たとえば、レポートが 10 ページあり、ユーザが FOCFIRSTPAGE を使用して最初

のページ番号を 1 ではなく 3 に設定した場合、TABLASTPAGE の値は 10 のままになります。

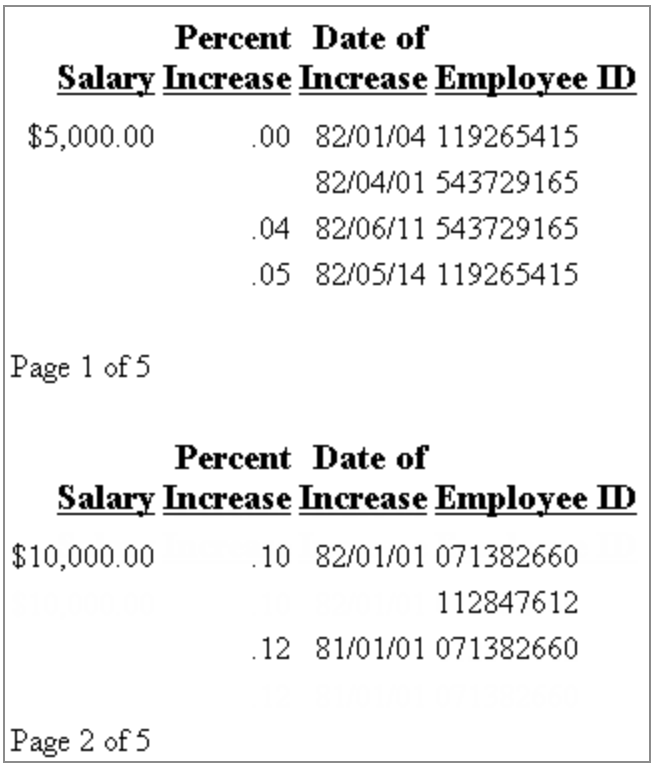
- TABLASTPAGE は、単一レポートのみでサポートされ、複合レポートではサポートされません。複合レポートでは、レポートごとに別のページ番号が生成されます。
- TABLASTPAGE は、HTML や PDF などのスタイルレポートのみでサポートされます。STYLE=OFF および STYLEMODE=FIXED を使用した EXL2K、WP、DOC、HTML ではサポートされません。
- TABLASTPAGE は、レポートに 2 回のアクセスを発生させます。最初に最終ページを計算し、その後、TABPAGENO (SQUEEZE=OFF の場合でも) を使用してそれを表示します。
- TABLASTPAGE はシステム (外部) ソートをサポートしません。
- GRAPH FILE は TABLASTPAGE をサポートしません。
- TABLEF は TABLASTPAGE ではサポートされません。

現在のページ番号および総ページ数の挿入

次の例は、レポートに現在のページ番号および総ページ数を追加する方法を示しています。関連する構文は、リクエスト内で太字で表示されています。

```
TABLE FILE EMPLOYEE
PRINT EMP_ID AS 'Employee ID'
BY SALARY IN-GROUPS-OF 5000 AS 'Salary'
BY PCT_INC AS 'Percent,Increase'
BY DAT_INC AS 'Date of,Increase'
ON SALARY PAGE-BREAK
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=TITLE, STYLE=BOLD, SIZE=11, $
ENDSTYLE
FOOTING
"Page <TABPAGENO of <TABLASTPAGE"
END
```

出力の最初の 2 ページは次のとおりです。



Salary	Percent Increase	Date of Increase	Employee ID
$5,000.00	.00	82/01/04	119265415
		82/04/01	543729165
	.04	82/06/11	543729165
	.05	82/05/14	119265415

Page 1 of 5

Salary	Percent Increase	Date of Increase	Employee ID
$10,000.00	.10	82/01/01	071382660
			112847612
	.12	81/01/01	071382660

Page 2 of 5

ソートグループ内での総ページ数の表示

レポートに REPAGE オプションを使用して各ソートグループのページ数を再設定する場合、見出しまたは脚注に <BYLASTPAGE 変数を使用すると、出力のページ数が各ソートグループ内に表示されます。この変数は、スタイル出力フォーマットでのみ使用可能です。

レポートで REPAGE オプションが使用されていない場合、<BYLASTPAGE では、レポートの総ページ数 (<TABLASTPAGE 変数) が使用されます。

各ソートグループ内の総ページ数の表示

リクエストに次の構文を含めて、スタイル出力フォーマットで出力を保持する必要があります。

```
BY sortfield REPAGE
```

「ページ x / y」を表示するには、見出しまたは脚注で次の構文を使用します。

```
{HEADING|FOOTING}
"Page <TABPAGENO of <BYLASTPAGE"
```

説明

sortfield

REPAGE オプションを指定するソートフィールドです。同一ソートフィールドまたは下位ソートフィールドで、PAGE-BREAK を指定する必要があります。PAGE-BREAK は、各ソート区切りで新しいページを開始します。REPAGE は、各ソート区切りでページ番号を 1 に再設定します。

<TABPAGENO

現在のページ番号を挿入します。

<BYLASTPAGE

ページ番号再設定前の最後のページ番号です。

ソートグループ内のページ設定

次のリクエストは、GGSALES データソースを使用し、PRODUCT、REGION、CATEGORY、CITY でソートします。PRODUCT の値が変わるたびに、ページ番号が再設定されます。見出しには、現在のページ番号および各製品 (PRODUCT) グループの総ページ数が表示されます。

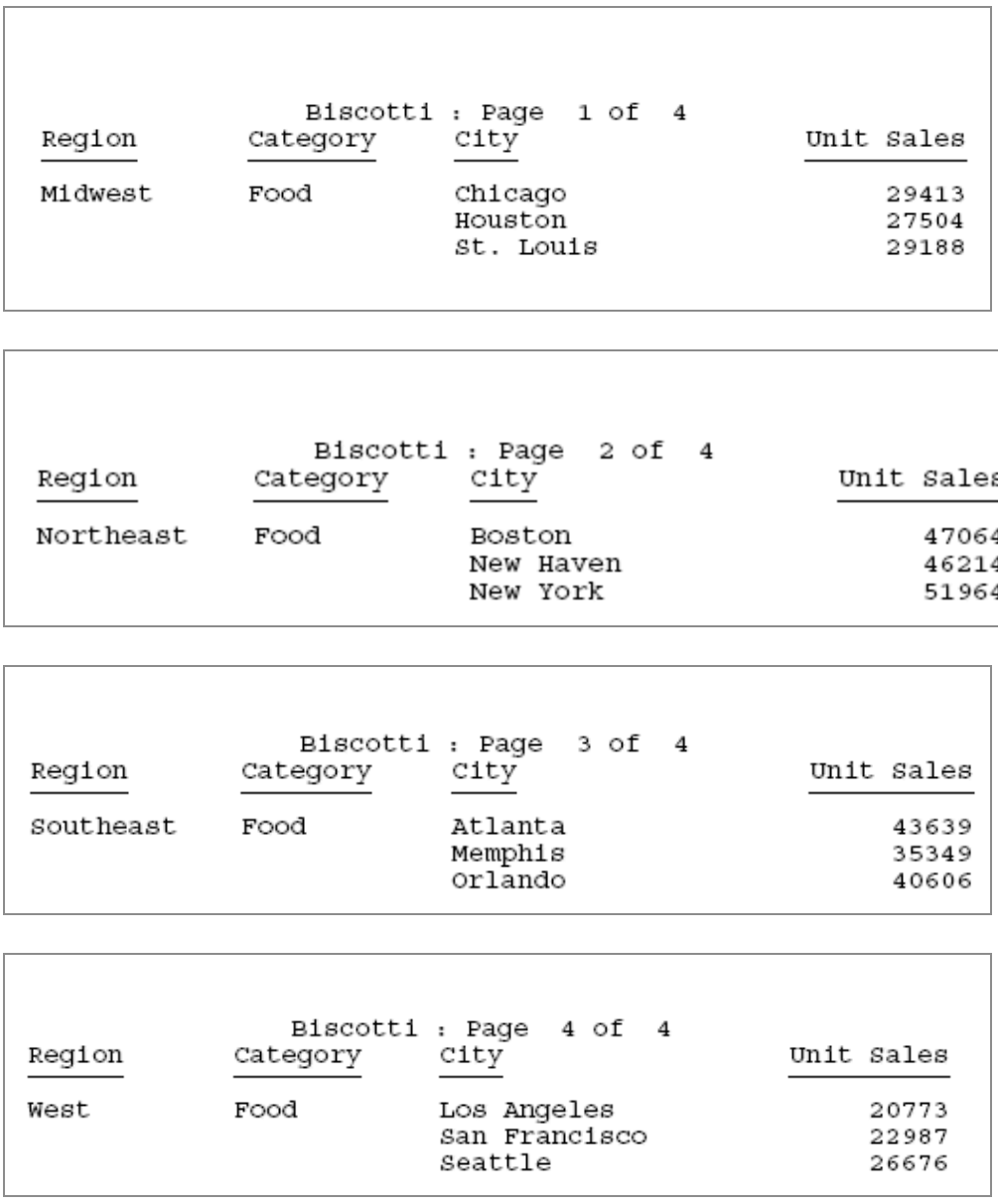
TABPAGENO および BYLASTPAGE 変数には、デフォルト設定で I5 フォーマットが指定されています。このフォーマットでは、ページ番号の前にブランクが多く残ります。スポットマーカーまたは COMPUTE コマンドを使用して、ページ番号を左側へ移動することができます。

次の例では、COMPUTE コマンドを使用して TABPAGENO 変数の値を I2 フォーマットとして格納するフィールド X を作成するとともに、スポットマーカーを使用して BYLASTPAGE ページ番号をブランク 4 個分だけ左へ移動します。フィールド X が認識されるようにするには、見出しコマンドを COMPUTE コマンドの後に配置する必要があります。

```
TABLE FILE GGSALES
SUM UNITS
COMPUTE X/I2 = TABPAGENO;
BY PRODUCT NOPRINT REPAGE
BY REGION PAGE-BREAK
BY CATEGORY
```

```
BY CITY
HEADING CENTER
"<PRODUCT :Page <X of <-4> <BYLASTPAGE "
ON TABLE PCHOLD FORMAT PDF
END
```

下図は、出力の一部で、PRODUCT の値が変わったところで、ページ番号が 1 に再設定され、BYLASTPAGE 変数によって、各 PRODUCT の総ページ数が表示されています。



Biscotti : Page 1 of 4

Region	Category	City	Unit Sales
Midwest	Food	Chicago	29413
		Houston	27504
		St. Louis	29188

Biscotti : Page 2 of 4

Region	Category	City	Unit Sales
Northeast	Food	Boston	47064
		New Haven	46214
		New York	51964

Biscotti : Page 3 of 4

Region	Category	City	Unit Sales
Southeast	Food	Atlanta	43639
		Memphis	35349
		Orlando	40606

Biscotti : Page 4 of 4

Region	Category	City	Unit Sales
West	Food	Los Angeles	20773
		San Francisco	22987
		Seattle	26676

```
                Capuccino : Page  1 of  3
Region       Category     City                    Unit Sales

Northeast    Coffee       Boston                       15358
                          New Haven                    12386
                          New York                     17041
```

開始ページ番号の指定

FOCFIRSTPAGE パラメータを使用して、レポートの最初のページにページ番号を割り当てることができます。この機能はレポートを印刷するときや、レポートを他のレポートの一部に組み込むときに役立ちます。

&FOCNEXTPAGE 変数を持つ FOCFIRSTPAGE パラメータを使用して、同じプロシジャ内で複数のレポートのページ番号を制御することもできます。

TABPAGENO が使用された場合、FOCFIRSTPAGE は無視されます。

先頭ページへのページ番号の割り当て

プロシジャ内のすべてのリクエスト

```
SET FOCFIRSTPAGE = {n|1|&FOCNEXTPAGE}
```

単一リクエスト

```
ON TABLE SET FOCFIRSTPAGE {n|1|&FOCNEXTPAGE}
```

説明

n

レポートの最初ページに割り当てる番号は、1 から 999999 までの整数です。

1

先頭ページにページ番号 1 を割り当てます。デフォルト値は 1 です。

&FOCNEXTPAGE

複数リクエスト内の前のレポートの最終ページ番号より 1 つ大きい値の変数です。値は実行時に計算されます。

先頭ページへのページ番号の割り当て

次のリクエストは、レポートの最初のページにページ番号 3 を割り当てます。

```
SET FOCFIRSTPAGE = 3
TABLE FILE CENTORD
HEADING
"Sales By Store"
SUM LINEPRICE AS 'Sales'
BY SNAME
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

レポートは次のとおりです。

PAGE 3

Sales By Store

Store Name:	Sales
AV VideoTown	$63,993,429.34
Audio Expert	$247,093,518.58
City Video	$15,299,051.29
Consumer Merchandise	$30,080,503.15
TV City	$83,041,182.81
Web Sales	$4,027,582.48
eMart	$290,487,756.36

連続レポート内のページ番号の制御

次のプロシジャには、2 つのリクエストが含まれています。2 つ目のリクエストは、FOCFIRSTPAGE を &FOCNEXTPAGE の値に設定します。

```
SET FOCFIRSTPAGE = 3
TABLE FILE CENTORD
HEADING
"Sales By Store"
SUM LINEPRICE AS 'Sales'
BY SNAME
WHERE SNAME EQ 'eMart'
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
-RUN
```

```
SET FOCFIRSTPAGE = &FOCNEXTPAGE
TABLE FILE CENTORD
HEADING
"Sales By Product"
SUM LINEPRICE AS 'Sales'
BY PRODCAT AS 'Product'
WHERE PRODCAT EQ 'VCRs'
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

2 番目のレポートの最初のページ番号には、前のレポートの最終ページよりも 1 つ大きい 4 が付けられます。

PAGE 3

Sales By Store

Store Name:	Sales
eMart	$290,487,756.36

PAGE 4

Sales By Product

Product	Sales
VCRs	$23,801,009.34

ページ番号表示の制御

デフォルト設定では、レポートページの最初の 2 行は予約されています。最初の行はページ番号を左上の隅に表示し、2 行目をブランクにします。デフォルトの表示を非表示にするには、PAGE-NUM パラメータを使用します。

ページ番号の表示制御

プロシジャ内のすべてのリクエスト

```
SET PAGE[-NUM] = num_display
```

単一リクエスト

```
ON TABLE SET PAGE[-NUM] num_display
```

説明

-NUM

オプションです。PAGE および PAGE-NUM は、同義語です。

num_display

次のいずれかです。

ON - 左上の隅にページ番号を表示し、その後に未使用のブランク行が続きます。デフォルト値は ON です。

OFF - デフォルトページ番号を非表示にします。

システム変数 TABPAGENO を使用できます。

NOPAGE - デフォルトページ番号を非表示にし、ページの最初の 2 行を使用できるようにします。

システム変数 TABPAGENO を使用できます。

TOP または NOLEAD - ページ番号のために確保された各ページの最上行、およびその後にあるブランク行を削除します。指定した場合、レポートの最初の行にはレポートまたはページ見出しが表示されます。見出しがない場合には、フィールドタイトルが表示されます。

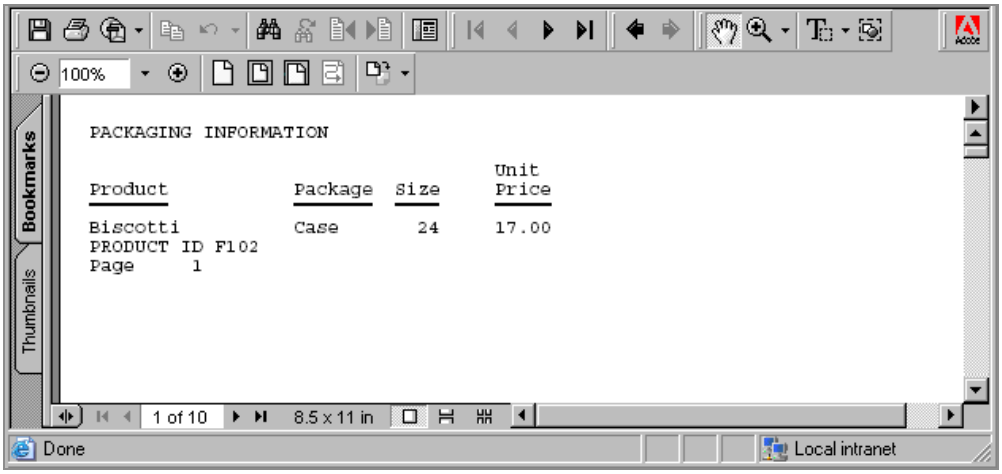
システム変数 TABPAGENO を使用すると、レポート内の任意の場所にページ番号を表示することができます。

デフォルトページ番号の非表示

次のリクエストは、SET PAGE-NUM = NOPAGE を使用して、デフォルトのページ番号を非表示にします。レポートの最初のページの先頭行を、レポートの見出しに使用します。

```
SET PAGE-NUM = NOPAGE
TABLE FILE GGPRODS
ON TABLE SUBHEAD
"PACKAGING INFORMATION"
" "
PRINT PACKAGE_TYPE AND SIZE AND UNIT_PRICE
BY PRODUCT_DESCRIPTION
ON PRODUCT_DESCRIPTION PAGE-BREAK SUBFOOT
"PRODUCT ID <PRODUCT_ID"
"Page <TABPAGENO "
ON TABLE SET ONLINE-FMT PDF
END
```

TABPAGENO は、ソート項目を指定している脚注内にページ番号を挿入します。レポートの最初のページは、次のとおりです。



AHTML リクエストでのレポート各ページのデータ行数の設定

AHTML リクエストで LINES-PER-PAGE スタイルシート属性を使用すると、レポート出力の各ページに表示するデータ行数を設定することができます。

AHTML リクエストでの 1 ページあたりのデータ行数の設定

レポート出力の各ページに表示するデータ行数を制御するには、次のスタイルシート構文を使用します。

```
TYPE=REPORT, LINES-PER-PAGE={n|UNLIMITED},$
```

説明

n

各ページに表示するデータ行数を指定します。デフォルト値は 57 行です。

UNLIMITED

出力結果をすべて 1 ページに表示します。

注意

- AHTML リクエストで次の SET コマンドを使用すると、LINES-PER-PAGE スタイルシートオプションと同一のデータ行数が表示されます。

  ```
  ON TABLE SET LINES {n|UNLIMITED}
  ```

- HTML リクエストで LINES-PER-PAGE スタイルシートオプションまたは SET LINE コマンドのいずれかを使用すると、データ行数ではなく、行数が表示されます。

レポート各ページのデータ行数の設定

次の例は、デフォルト WebFOCUS スタイルシートを使用し、レポート出力の各ページに 20 のデータ行を表示します。

```
TABLE FILE GGSALES
HEADING
"Sales Report"
SUM UNITS DOLLARS
BY CATEGORY BY PRODUCT BY DATE NOPRINT
WHERE DATE GE 19960101 AND DATE LE 19960401
ON TABLE PCHOLD FORMAT AHTML
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/ibi_themes/Warm.sty,$
TYPE=REPORT, GRID=OFF, LINES-PER-PAGE = 20, $
ENDSTYLE
END
```

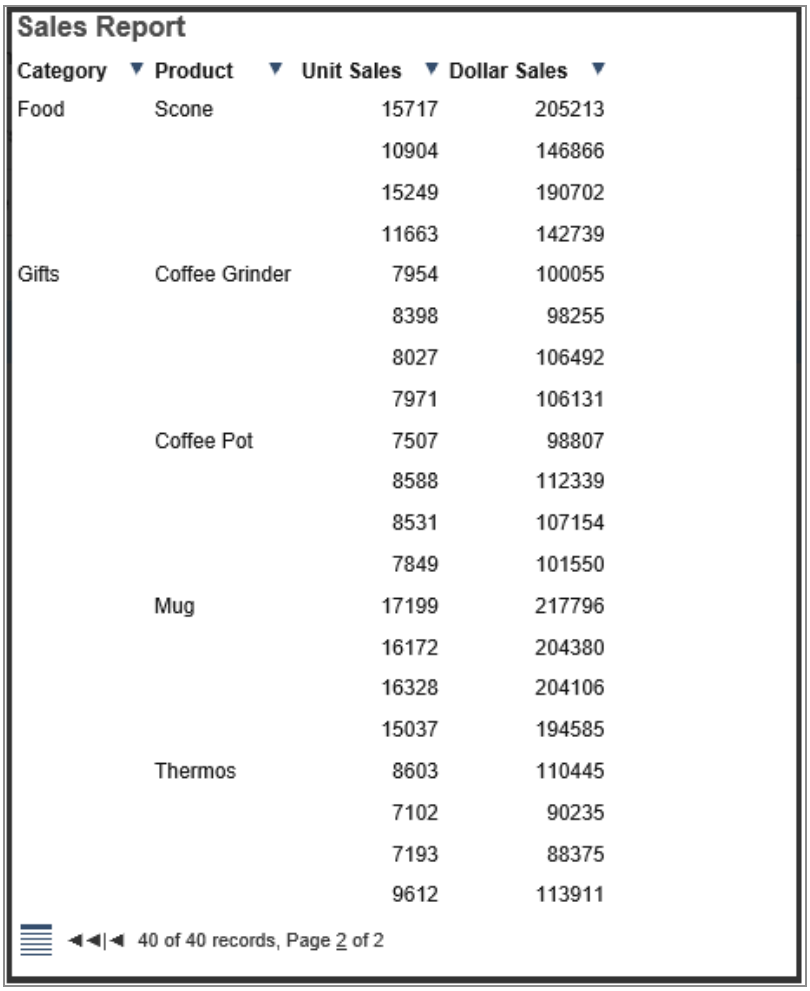
下図は、最初のページの出力を示しています。

Sales Report

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	5507	71728
		5219	67429
		6343	87290
		4170	58692
	Espresso	14499	182900
		14015	169023
		14871	184474
		11613	156396
	Latte	31469	401873
		38725	490645
		42717	520948
		36374	468137
Food	Biscotti	18993	236698
		18816	241975
		15602	197254
		17601	218437
	Croissant	22537	274926
		29691	361053
		30954	388868
		27412	344197

40 of 40 records, Page 1 of 2 ►|►►

下図は、2 ページ目の出力を示しています。



Sales Report

Category	Product	Unit Sales	Dollar Sales
Food	Scone	15717	205213
		10904	146866
		15249	190702
		11663	142739
Gifts	Coffee Grinder	7954	100055
		8398	98255
		8027	106492
		7971	106131
	Coffee Pot	7507	98807
		8588	112339
		8531	107154
		7849	101550
	Mug	17199	217796
		16172	204380
		16328	204106
		15037	194585
	Thermos	8603	110445
		7102	90235
		7193	88375
		9612	113911

◄◄|◄ 40 of 40 records, Page 2 of 2

罫線および境界線の追加

デフォルト設定では、HTML レポートには横方向と縦方向の罫線が含まれます。罫線を削除したり、BY ソートフィールドに表示する罫線を調整したりすることができます。罫線の特性はレポートの個別コンポーネントではなく、HTML レポート全体に適用されます。

レポート内の見出し、脚注、フィールドタイトル周囲に境界線および罫線を追加することで、それらを強調することができます。

境界線　PDF、HTML、DHTML、XLSX、EXL2K、PPTX、PPT レポートで、スタイルシートに BORDER 属性を使用して、境界線の太さ、スタイル、色を指定することができます。必要に応じて、境界線の上、下、左、右のフォーマットを指定することもできます。

BORDERALL スタイルシート属性は、PDF、DHTML、HTML、XLSX、PPTX レポートで、見出しまたは脚注の個々のセルの周囲への境界線の設定をサポートします。見出しまたは脚注

の個々のオブジェクトで、この属性を BORDER 属性とともに使用して、項目ごとに境界線を指定することができます。

現在のバージョンでは、SQUEEZE=ON に設定した場合、中間見出しおよび中間脚注の右マージンの境界線は、すべての見出し、脚注、中間見出し、中間脚注の最大幅に基づいて定義されます。中間見出し行と中間脚注行の長さは、レポートのデータ行のサイズではなく、ページ見出しとページ脚注の長さと連動します。スタイルシートで ALIGN-BORDERS=BODY 属性を使用して、中間見出しと中間脚注の位置を、その他の見出し要素ではなく、PDF レポートのレポート項目と揃えることができます。

罫線 HTML レポート内で、レポート全体の罫線をオンまたはオフに設定するために、スタイルシート内で GRID 属性を使用できます。内部カスケードスタイルシートと併用すると、GRID は太い二重線 (HTML デフォルト) ではなく細い罫線を生成します。PDF レポート内で横方向および縦方向の罫線を追加し、その太さを調整するために HGRID および VGRID 属性を使用できます。

注意：グラフに適用する SET GRID パラメータは、GRID スタイルシート属性とは異なります。

罫線表示属性

属性	説明	適用先
`GRID`	罫線表示を制御します。	HTML PDF DHTML PPTX PPT
`HGRID`	横方向の罫線表示および罫線の太さを制御します。	PDF DHTML PPTX PPT

属性	説明	適用先
VGRID	縦方向の罫線表示および罫線の太さを制御します。	PDF DHTML PPTX PPT

注意：Adobe Reader を使用して PDF レポートを表示する際に、表示が 100 パーセントに設定されていない場合は、GRID 線の表示は指定したものよりも細くなる場合があります。たとえば、ドキュメントの表示を 50 パーセントに設定した場合、GRID 線がほかよりも細くなることがあります。

HTML レポート内の罫線表示の制御

```
[TYPE=REPORT,] GRID= option, $
```

説明

TYPE=REPORT

罫線をレポート全体に適用します。デフォルト値であるため必須ではありません。

option

次のいずれかです。

ON - 罫線をレポートに適用します。BY フィールドの値が変わるまで、現在の値より下のセルに罫線を適用しません。フィールドタイトルに下線は付けられません。デフォルト値は ON です。

OFF - デフォルト罫線を無効にします。フィールドタイトルには、下線が付きます。ブランク行および下線を含めることができます。セルデータのデータ折り返しはできません。この設定では、レポートは読みにくくなる場合があります。

FILL - 罫線をレポートのセル全体に適用します。フィールドタイトルに下線は付けられません。

境界線の追加およびフォーマット

境界線をすべて同一にするには、次の構文を使用します。

```
TYPE=type, BORDER=option, [BORDER-STYLE=line_style,]
    [BORDER-COLOR={color|RGB(r g b)},] $
```

境界線の上下左右に異なる属性を指定するには、次の構文を使用します。

```
TYPE=type, BORDER-position=option,
   [BORDER[-position]-STYLE=line_style,]
   [BORDER[-position]-COLOR={color|RGB(r g b)},] $
```

説明

type

境界線が適用されるレポートコンポーネントを識別します。有効な値についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

option

次のいずれかの値です。

ON - 境界線をオンにします。ON を指定した場合、MEDIUM と同一の線が生成されます。

注意：MEDIUM 線の設定は、GRID 属性で作成された線の太さと同一です。

OFF - 境界線をオフにします。デフォルト値は OFF です。

LIGHT - 細い線を指定します。

MEDIUM - 標準の太さの線を指定します。ON は、線を MEDIUM に設定します。

HEAVY - 太い線を指定します。

width - 単位にポイントを使用して線の太さを指定します。72 ポイントは、1 インチです。Excel 2003 には、線の太さをポイント数で正確に指定するオプションが用意されていないため、このオプションは、Excel 2003 ではサポートされません。

ヒント：ポイントを使用して指定する線幅の表示は、HTML と PDF 出力で異なります。表示フォーマットに関係なく、外観を統一するには、LIGHT、MEDIUM、または HEAVY を使用します。

position

フォーマットを設定する境界線を指定します。有効な値は、TOP、BOTTOM、LEFT、RIGHT です。

任意の BORDER 属性の位置識別子を指定することができます。これにより、任意の境界線の幅、スタイル、色にフォーマットを個別に設定することができます。

line_style

境界線のスタイルを設定します。WebFOCUS スタイルシートでは、すべての標準カスケードスタイルシートの線スタイルがサポートされます。いくつかの 3 次元スタイルは、HTML 内でのみ利用可能です（アスタリスク (*) の付いたスタイルです）。有効な値には、次のものがあります。

スタイル	説明
NONE	境界線なし
SOLID	実線
DOTTED	点線
DASHED	破線
DOUBLE	二重線
GROOVE*	3D 溝線 (この境界線オプションは用意されていないため、Excel 2003 ではサポートされません)
RIDGE*	3D 稜線 (この境界線オプションは用意されていないため、Excel 2003 ではサポートされません)

スタイル	説明
INSET*	3D 凹線
OUTSET*	3D 凸線

color

定義済みの色です。デフォルト値は、BLACK です。

ディスプレイデバイスまたは出力デバイスが指定した色をサポートしない場合、灰色に置き換えられます。利用できる色の値のリストは、レポートデータのフォーマットを参照してください。

RGB

赤、緑、青の組み合わせを使用して、色を指定します。

(r g b)

赤、緑、青のそれぞれの最適な輝度です。0 (ゼロ) から 255 までの値が使用されます。0 (ゼロ) は最も輝度が低く、255 は最も輝度が高いことを示します。3 色の構成要素を同じ輝度にすると、結果は灰色になります。

注意：EXL2K フォーマットでは、GRID=ON パラメータはサポートされません。

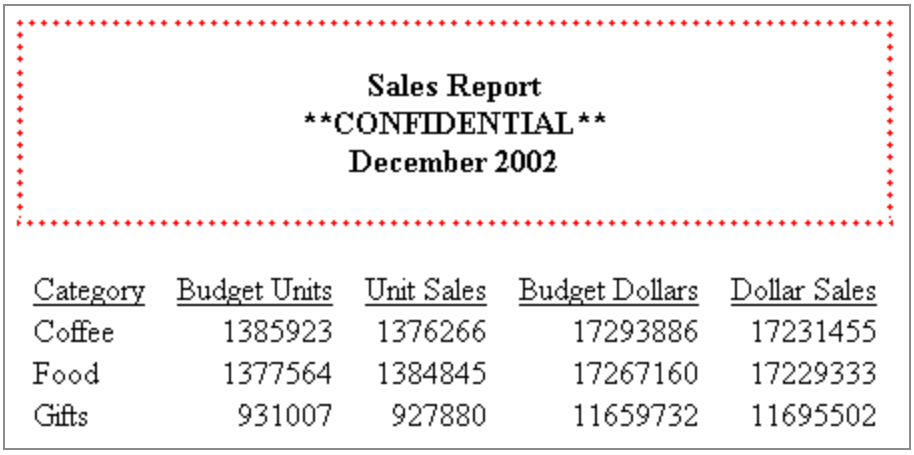
境界線の挿入およびフォーマットの設定

次のリクエストは、レポートの見出し全体を太い赤の点線で囲んだ HTML レポートを生成します。

```
TABLE FILE GGSALES
SUM BUDUNITS UNITS BUDDOLLARS DOLLARS
BY CATEGORY
ON TABLE SUBHEAD
"</1 Sales Report"
"**CONFIDENTIAL**"
"December 2002 </1"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
```

```
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=TABHEADING, STYLE=BOLD, JUSTIFY=CENTER, BORDER=HEAVY,
     BORDER-COLOR=RED, BORDER-STYLE=DOTTED, $
ENDSTYLE
END
```

出力結果は次のとおりです。



Sales Report
CONFIDENTIAL
December 2002

Category	Budget Units	Unit Sales	Budget Dollars	Dollar Sales
Coffee	1385923	1376266	17293886	17231455
Food	1377564	1384845	17267160	17229333
Gifts	931007	927880	11659732	11695502

ヒント：PDF レポート内でこの出力を生成するために、同じ BORDER 構文を使用できます。

HTML レポート内のデフォルト罫線の表示

次のリクエストは、デフォルト設定 GRID=ON を使用します。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY CATEGORY BY PRODUCT
ON TABLE SET PAGE-NUM OFF
END
```

ソートフィールド CATEGORY の下のセルは、値が変わるまで (例、Coffee から Food に) 罫線は付きません。

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

罫線の HTML レポートのセル全体への適用

次のリクエストは、GRID=FILL を使用して、ソートフィールド CATEGORY の下のセルも含め、罫線をセル全体に適用します。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY CATEGORY BY PRODUCT
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=FILL, $
ENDSTYLE
END
```

すべてのセルには、罫線が付いています。

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

HTML レポートからの罫線の削除

次のリクエストは、レポートからデフォルト罫線を削除するために GRID=OFF を使用します。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY CATEGORY BY PRODUCT
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

フィールドタイトルには、下線が付きます。

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

Excel レポート出力への境界線の追加

Excel レポート出力に境界線を追加すると、ブランクの行が追加される場合があります。

BORDER 構文 (BORDERS=OFF を含む) を使用する場合、インラインプロシジャかスタイルシート参照かに関わらず、プロシジャは BORDER モードに設定されてスタイルおよび間隔の設定が調整されるため、余分なブランク行が追加される原因となります。

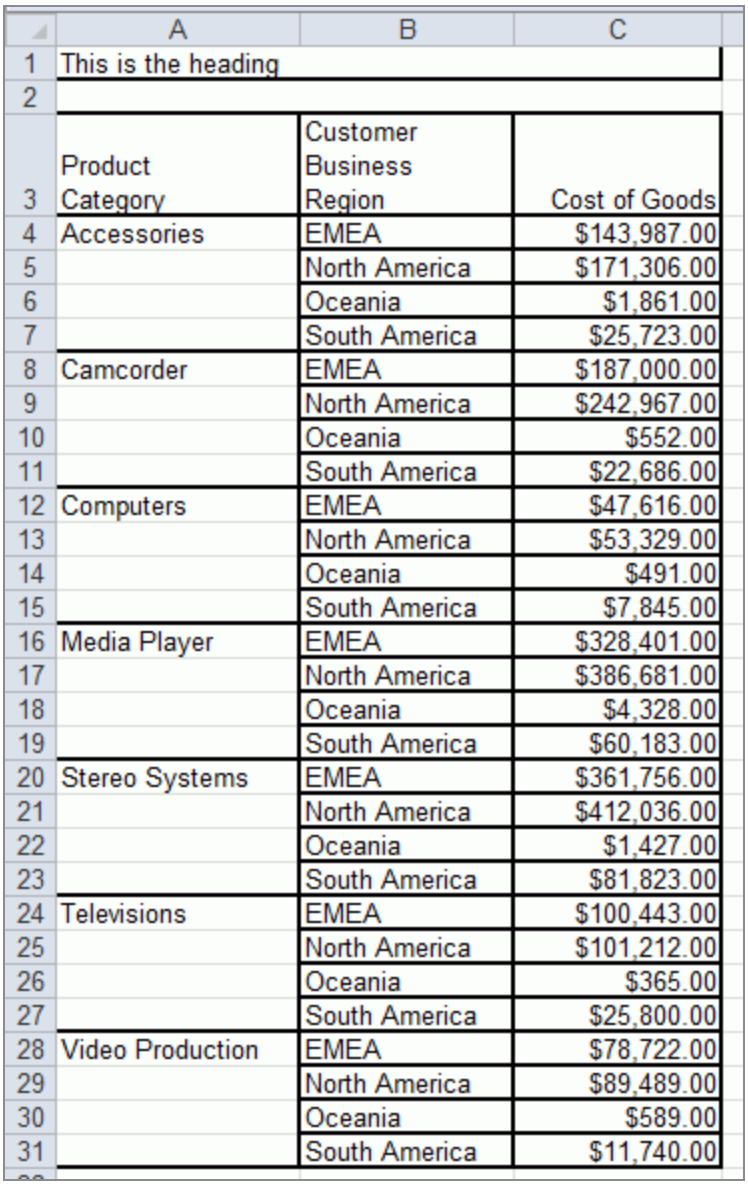
XLSX フォーマットレポート出力への境界線の追加

次のリクエストは、境界線を有効にした XLSX フォーマットのレポート出力を生成します。

```
TABLE FILE WF_RETAIL_LITE
SUM COGS_US
BY PRODUCT_CATEGORY
BY BUSINESS_REGION
HEADING
"This is the heading"
ON TABLE PCHOLD FORMAT XLSX
ON TABLE NOTOTAL
ON TABLE SET STYLE *
TYPE=REPORT,BORDER=ON,$
```

```
ENDSTYLE
END
```

出力結果は次のとおりです。見出しの下にブランク行が追加されています。



	A	B	C
1	This is the heading		
2			
3	Product Category	Customer Business Region	Cost of Goods
4	Accessories	EMEA	$143,987.00
5		North America	$171,306.00
6		Oceania	$1,861.00
7		South America	$25,723.00
8	Camcorder	EMEA	$187,000.00
9		North America	$242,967.00
10		Oceania	$552.00
11		South America	$22,686.00
12	Computers	EMEA	$47,616.00
13		North America	$53,329.00
14		Oceania	$491.00
15		South America	$7,845.00
16	Media Player	EMEA	$328,401.00
17		North America	$386,681.00
18		Oceania	$4,328.00
19		South America	$60,183.00
20	Stereo Systems	EMEA	$361,756.00
21		North America	$412,036.00
22		Oceania	$1,427.00
23		South America	$81,823.00
24	Televisions	EMEA	$100,443.00
25		North America	$101,212.00
26		Oceania	$365.00
27		South America	$25,800.00
28	Video Production	EMEA	$78,722.00
29		North America	$89,489.00
30		Oceania	$589.00
31		South America	$11,740.00

次のリクエストでは、境界線属性は指定されていません。

```
TABLE FILE WF_RETAIL_LITE
SUM COGS_US
BY PRODUCT_CATEGORY
BY BUSINESS_REGION
```

```
HEADING
"This is the heading"
ON TABLE PCHOLD FORMAT XLSX
ON TABLE NOTOTAL
END
```

出力結果は次のとおりです。レポート見出しとレポート本文の間に、ブランク行はありません。

	A	B	C
1	This is the heading		
2	Product Category	Customer Business Region	Cost of Goods
3	Accessories	EMEA	$143,987.00
4		North America	$171,306.00
5		Oceania	$1,861.00
6		South America	$25,723.00
7	Camcorder	EMEA	$187,000.00
8		North America	$242,967.00
9		Oceania	$552.00
10		South America	$22,686.00
11	Computers	EMEA	$47,616.00
12		North America	$53,329.00
13		Oceania	$491.00
14		South America	$7,845.00
15	Media Player	EMEA	$328,401.00
16		North America	$386,681.00
17		Oceania	$4,328.00
18		South America	$60,183.00
19	Stereo Systems	EMEA	$361,756.00
20		North America	$412,036.00
21		Oceania	$1,427.00
22		South America	$81,823.00
23	Televisions	EMEA	$100,443.00
24		North America	$101,212.00
25		Oceania	$365.00
26		South America	$25,800.00
27	Video Production	EMEA	$78,722.00
28		North America	$89,489.00
29		Oceania	$589.00
30		South America	$11,740.00

見出しまたは脚注への内側と外側の境界線の挿入

見出しグリッド全体に境界線を追加する場合、最も簡単な方法は、BORDERALL を使用することです。この機能は、PDF、DHTML、HTML、XLSX、PPTX フォーマットでサポートされます。境界線を個別に削除するには、BORDER、BORDER-LEFT、BORDER-RIGHT、BORDER-TOP、BORDER-BOTTOM を使用して、個々の項目の境界線の設定を明示的にオフにします。BORDERALL で境界線をオンにした項目に BORDER-LEFT=OFF を設定すると、その項目の左境界線は非表示になりますが、上、下、右の境界線では、定義済みの境界線スタイルが保持されます。

見出しと脚注の境界線として、3 つのレベルがサポートされます。

1. 個々のセルの境界線

   見出し、および選択した項目またはセルの外側の境界線のコンポーネントを個別に設定する場合、BORDER-LEFT、BORDER-RIGHT、BORDER-TOP、BORDER-BOTTOM を使用することができます。

2. 外側の境界線すべて

   見出しまたは脚注内で外側の境界線を設定する場合、BORDER= を使用します。

3. 外側と内側の境界線すべて

   選択した見出しまたは脚注の内側と外側の両方に境界線の属性を適用する場合、BORDERALL を使用します。

内側と外側の境界線をすべてオンにするには、次の構文を使用します。

```
TYPE=headfoot, BORDERALL=option, [BORDER-STYLE=line_style,] [BORDER-COLOR=
{color|RGB(rgb)},] $
```

説明

headfoot

見出しまたは脚注のタイプです。有効な値は、TABHEADING、TABFOOTING、HEADING、FOOTING、SUBHEAD、SUBFOOT です。

option

次のいずれかの値です。

- **ON** - 境界線をオンにします。ON を指定した場合、MEDIUM と同一の線が生成されます。

  注意：MEDIUM 線の設定は、GRID 属性で作成された線の太さと同一です。

- **OFF** - 境界線をオフにします。デフォルト値は OFF です。
- **LIGHT** - 細い線を指定します。
- **MEDIUM** - 標準の太さの線を指定します。ON は、線を MEDIUM に設定します。
- **HEAVY** - 太い線を指定します。
- 数値を入力すると、線の太さをポイント数で指定できます (72 ポイント = 1 インチ)。Excel 2003 には、線の太さをポイント数で正確に指定するオプションが用意されていないため、このオプションは、Excel 2003 ではサポートされません。

  ヒント：ポイントを使用して指定する線幅の表示は、HTML と PDF 出力で異なります。表示フォーマットに関係なく、外観を統一するには、LIGHT、MEDIUM、または HEAVY を使用します。

境界線をすべて同一にするには、次の構文を使用します。

```
TYPE=headfoot, BORDER=option
```

境界線の上下左右に異なる属性を指定するには、次の構文を使用します。

```
TYPE=headfoot, BORDER-position=option,
   [BORDER[-position]-STYLE=line_style,]
   [BORDER[-position]-COLOR={color|RGB(rgb)},] $
```

説明

headfoot

境界線の適用先として、見出し、脚注、中間見出し、中間脚注のいずれかを指定します。

position

フォーマットを設定する境界線を指定します。有効な値には、次のものがあります。有効な値は、TOP、BOTTOM、LEFT、RIGHT です。

任意の BORDER 属性の位置識別子を指定することができます。これにより、任意の境界線の幅、スタイル、色にフォーマットを個別に設定することができます。

option

次のいずれかの値です。

- **ON** - 境界線をオンにします。ON を指定した場合、MEDIUM と同一の線が生成されます。

  注意：MEDIUM 線の設定は、GRID 属性で作成された線の太さと同一です。

- **OFF** - 境界線をオフにします。デフォルト値は OFF です。
- **LIGHT** - 細い線を指定します。
- **MEDIUM** - 標準の太さの線を指定します。ON は、線を MEDIUM に設定します。
- **HEAVY** - 太い線を指定します。
- 数値を入力すると、線の太さをポイント数で指定できます (72 ポイント = 1 インチ)。Excel 2003 には、線の太さをポイント数で正確に指定するオプションが用意されていないため、このオプションは、Excel 2003 ではサポートされません。

  ヒント：ポイントを使用して指定する線幅の表示は、HTML と PDF 出力で異なります。表示フォーマットに関係なく、外観を統一するには、LIGHT、MEDIUM、または HEAVY を使用します。

line_style

境界線のスタイルを設定します。WebFOCUS スタイルシートでは、すべての標準カスケードスタイルシートの線スタイルがサポートされます。3 次元スタイルには、HTML のみで利用可能なものがあります （アスタリスク (*) の付いたスタイル）。有効な値には、次のものがあります。

スタイル	説明
NONE	境界線なし
SOLID	実線
DOTTED	点線
DASHED	破線
DOUBLE	二重線

スタイル	説明
GROOVE*	3D 溝線 (この境界線オプションは用意されていないため、Excel 2003 ではサポートされません)
RIDGE*	3D 稜線 (この境界線オプションは用意されていないため、Excel 2003 ではサポートされません)
INSET*	3D 凹線
OUTSET*	3D 凸線

注意：PDF、DHTML、PPTX でサポートされる線タイプはすべて、HEADALIGN=BODY とともに内側の個々の境界線に使用することができます。

color

定義済みの色です。デフォルト値は、BLACK です。

ディスプレイデバイスまたは出力デバイスが指定した色をサポートしない場合、灰色に置き換えられます。利用可能な色の値の完全なリストは、レポートのフォントフォーマット指定を参照してください。

RGB

赤、緑、青の組み合わせを使用して、色を指定します。

(r g b)

赤、緑、青、それぞれの輝度をブランクで区切って指定します。0 (ゼロ) から 255 までの値が使用されます。0 (ゼロ) は最も輝度が低く、255 は最も輝度が高いことを示します。3 色の構成要素を同じ輝度にすると、結果は灰色になります。

PDF レポートでの見出しと脚注に表示される要素の境界線制御

次のリクエストは、EMPLOYEE データソースに対して実行され、ページ見出し、中間見出し、中間脚注、レポート脚注を設定します。

```
TABLE FILE EMPLOYEE
HEADING
" Department Report Page <TABPAGENO "
PRINT LAST_NAME AS ''
FIRST_NAME AS ''
CURR_SAL AS ''
CURR_JOBCODE AS ''
BY DEPARTMENT AS ''
WHERE CURR_SAL NE 0.0
ON TABLE PCHOLD FORMAT PDF
ON DEPARTMENT SUBFOOT
" "
"Subtotal:<ST.CURR_SAL"
" "
ON DEPARTMENT SUBHEAD
"Department <+0>Last Name <+0>First Name <+0>Salary<+0>Jobcode <+0>"
ON TABLE SUBFOOT
"Grand Total:<ST.CURR_SAL"
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
TYPE=REPORT, FONT=ARIAL, BORDER=ON, SQUEEZE=ON, $
TYPE=REPORT, COLUMN=CURR_JOBCODE,SQUEEZE=.75, $
TYPE = SUBHEAD, HEADALIGN=BODY, BORDERALL=ON,$
TYPE = SUBFOOT, HEADALIGN=BODY,$
TYPE = SUBFOOT, LINE=2, ITEM=1, COLSPAN=3, JUSTIFY=RIGHT,$
TYPE = SUBFOOT, LINE=2, ITEM=2, JUSTIFY=RIGHT,$
TYPE = TABFOOTING, HEADALIGN=BODY,$
TYPE = TABFOOTING, ITEM=1, COLSPAN=3, JUSTIFY=RIGHT,$
TYPE = TABFOOTING, ITEM=2, JUSTIFY=RIGHT,$
ENDSTYLE
END
```

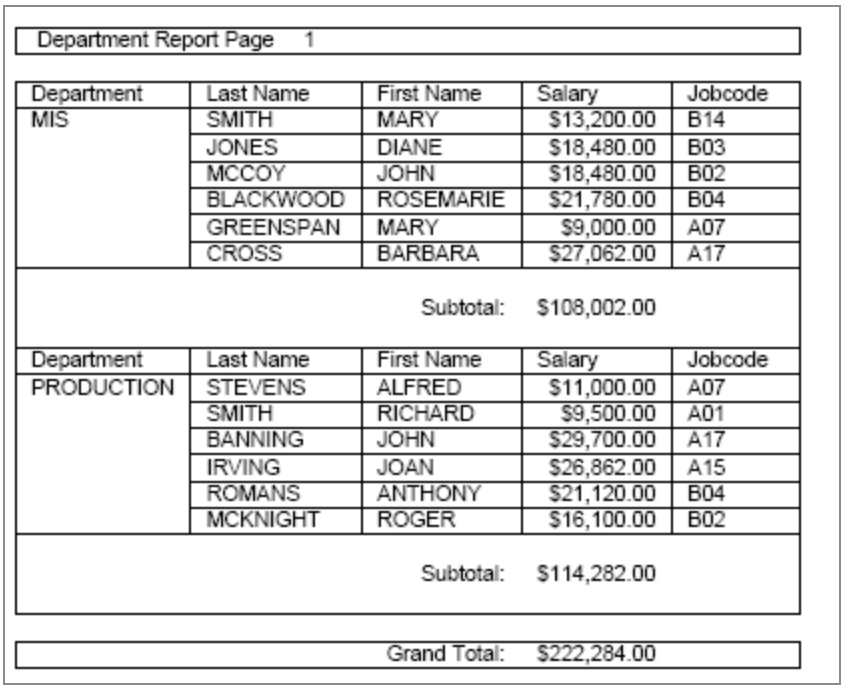
REPORT コンポーネントに BORDER=ON が指定されているため、このページ見出しには、外側の境界線が表示されます。

中間見出しには BORDERALL=ON および HEADALIGN=BODY が指定されているため、中間見出しのグリッドの位置は、レポートのグリッドと揃えられ、さらに中間見出しの各項目は、すべての境界線が表示される個々のセルとして表示されます。

スタイルシートによって、中間脚注要素の位置はレポートと揃えられます。また、2 列目の給与の中間合計の位置は Salary (CURR_SAL) 列と揃えられます。

REPORT コンポーネントで BORDER=ON が指定されているため、テーブル脚注では、脚注全体の周囲に境界線が表示されます。総合計 (Grand Total) の位置は、Salary (CURR_SAL) 列と揃えることができます。

出力結果は次のとおりです。



Department Report Page 1

Department	Last Name	First Name	Salary	Jobcode
MIS	SMITH	MARY	$13,200.00	B14
	JONES	DIANE	$18,480.00	B03
	MCCOY	JOHN	$18,480.00	B02
	BLACKWOOD	ROSEMARIE	$21,780.00	B04
	GREENSPAN	MARY	$9,000.00	A07
	CROSS	BARBARA	$27,062.00	A17
		Subtotal:	$108,002.00	
Department	Last Name	First Name	Salary	Jobcode
PRODUCTION	STEVENS	ALFRED	$11,000.00	A07
	SMITH	RICHARD	$9,500.00	A01
	BANNING	JOHN	$29,700.00	A17
	IRVING	JOAN	$26,862.00	A15
	ROMANS	ANTHONY	$21,120.00	B04
	MCKNIGHT	ROGER	$16,100.00	B02
		Subtotal:	$114,282.00	

Grand Total: $222,284.00

中間見出しと中間脚注マージンのレポートとの位置揃え

現在のバージョンでは、SQUEEZE=ON に設定した場合、中間見出しおよび中間脚注の右マージンの境界線は、すべての見出し、脚注、中間見出し、中間脚注の最大幅に基づいて定義されます。中間見出し行と中間脚注行の長さは、レポートのデータ行のサイズではなく、ページ見出しとページ脚注の長さと連動します。

スタイルシートで ALIGN-BORDERS=BODY 属性を使用して、中間見出しと中間脚注の位置を、その他の見出し要素ではなく、PDF レポートのレポート項目と揃えることができます。

スタイルシートで REPORT コンポーネントに ALIGN-BORDERS=BODY 属性を追加するか、TYPE 属性を含めずに、この属性を個別に指定することで、中間見出しまたは中間脚注のマージンの位置をレポートと揃えることができます。

```
[TYPE=REPORT,] ALIGN-BORDERS={OFF|BODY} ,$
```

説明

OFF

中間見出しと中間脚注の右マージンの位置を、レポートと揃えません。

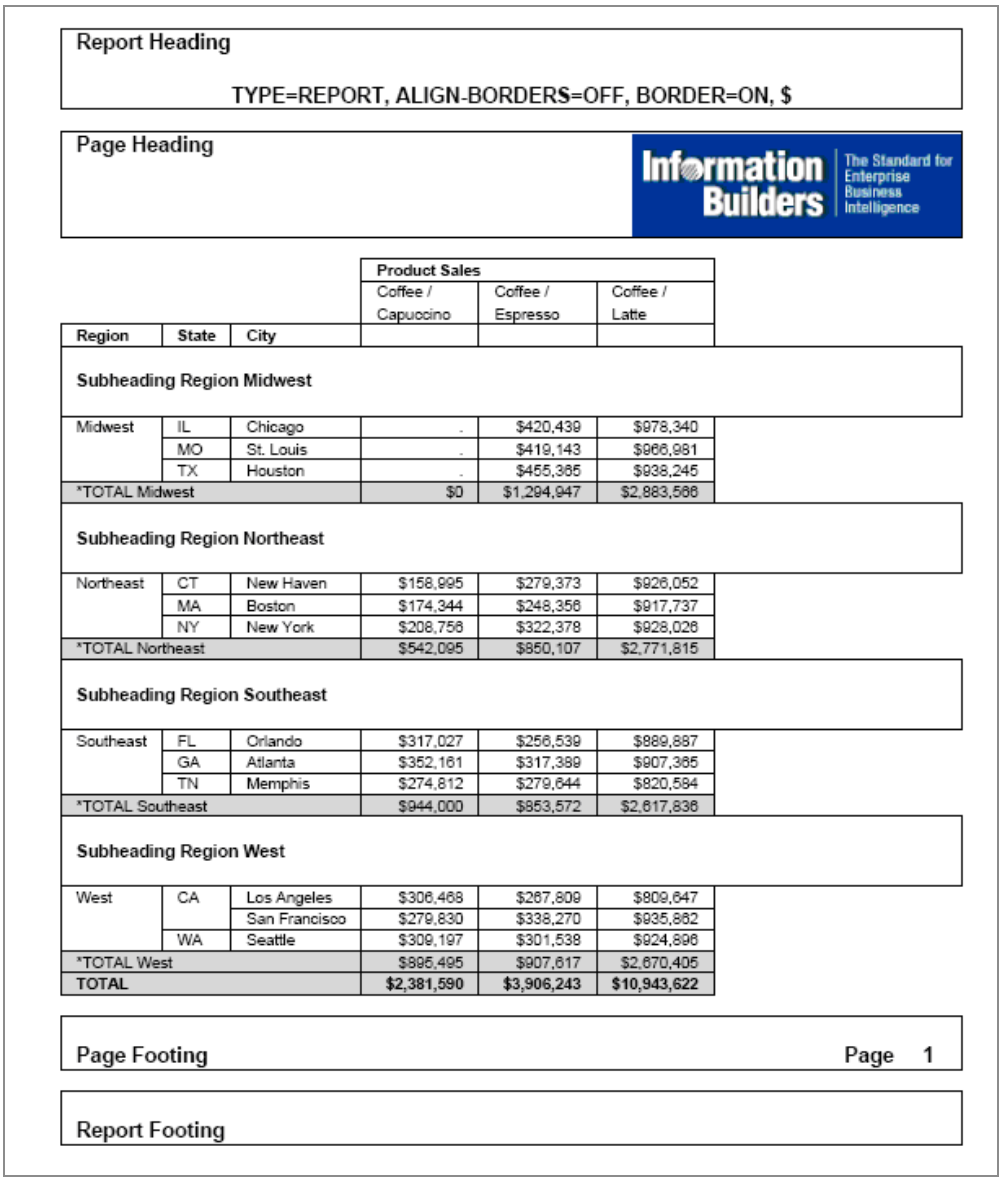
BODY

中間見出し行と中間脚注行の幅を、見出し行、脚注行、タブ見出し行、タブ脚注行とは別に指定します。さらに、レポートの右境界線の位置を、中間見出し行と中間脚注行の拡張 (データ行より狭い場合) またはデータ行の拡張 (データ行が中間見出しまたは中間脚注の最大幅より狭い場合) によって揃えます。

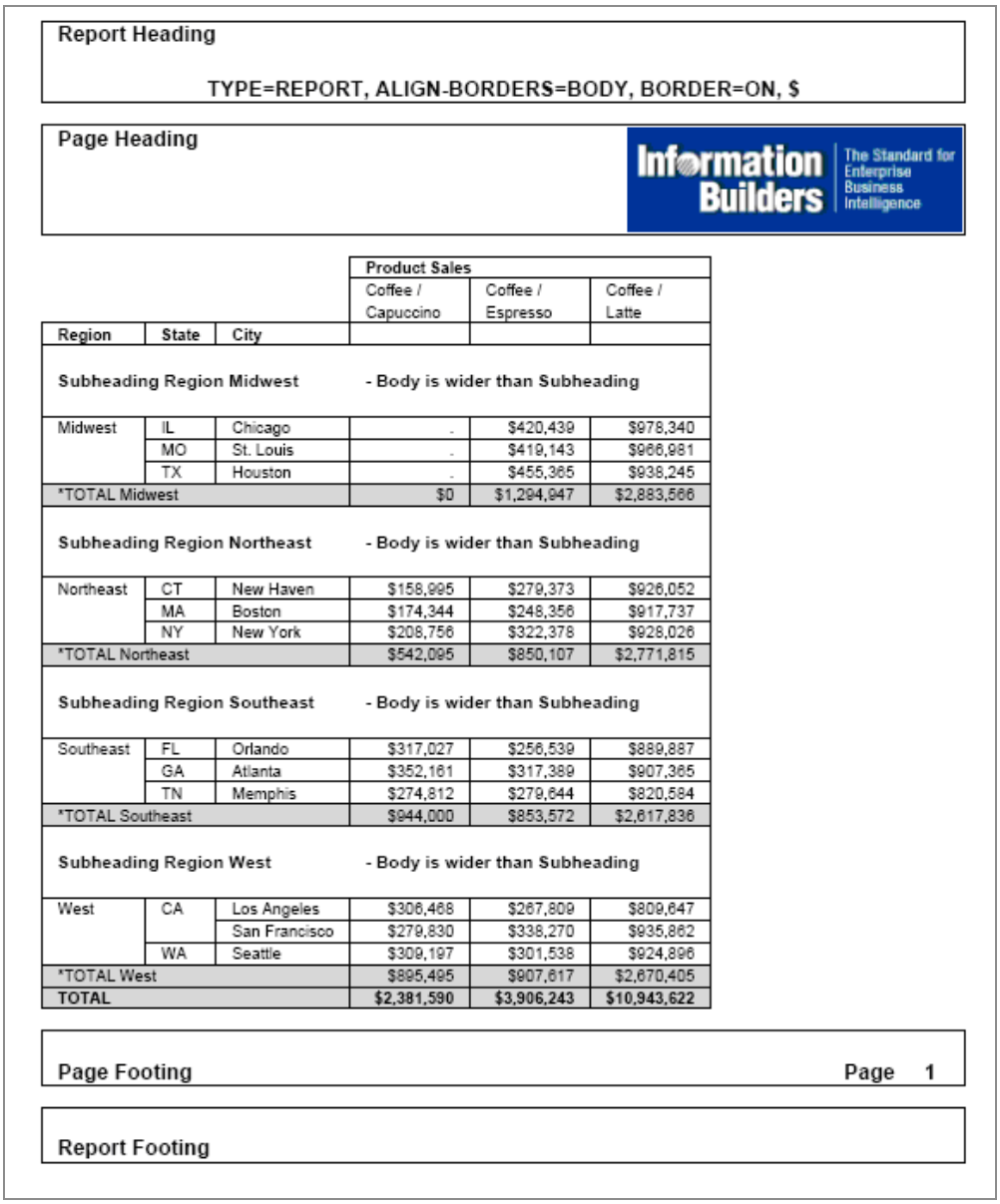
中間見出しと中間脚注マージンをレポートと位置揃えする際の注意

ALIGN-BORDERS=BODY 属性を指定しない場合、中間見出し行と中間脚注行の幅は、すべての見出しと脚注 (レポート見出し、ページ見出し、中間見出し、中間脚注) の最大幅で決定されます。

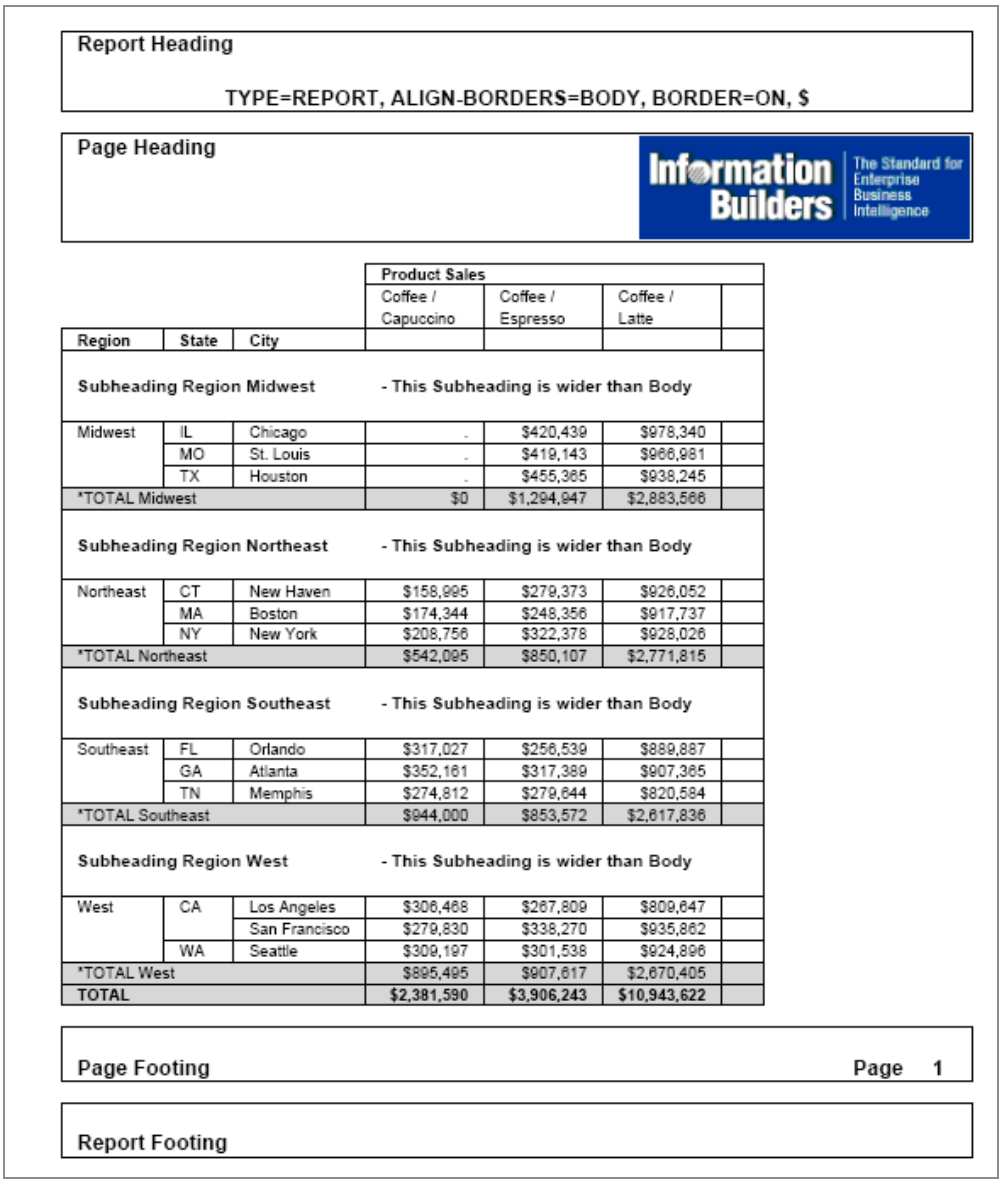
下図は、ALIGN-BORDERS=BODY 属性を指定しない場合のレポートの例を示しています。



下図のように、レポートの幅が中間見出し行および中間脚注行よりも広い場合、中間見出し行と中間脚注行の境界線と背景色が拡張され、データ行の幅と一致されます。



下図のように、中間見出し行と中間脚注行がレポート行よりも長い場合は、定義済みの境界線と背景色が、中間見出し行と中間脚注行で定義される幅と同一になるよう、各データ行に補間セルが追加されます。



ALIGN-BORDERS=BODY は、次のレポートに適用することができます。

- 単一パネルのレポート (横方向のパネル化を伴わないレポート)。
- レポートで定義済みのすべての中間見出しと中間脚注で HEADPANEL がオンに設定された、パネルレポート。

  HEADPANEL=ON を設定すると、パネル化されたレポートの最初のページの見出しおよび脚注が、後続するパネルに繰り返し表示されます。HEADPANEL を使用しない場合、スタイルシートの POSITION 属性を使用して、項目を見出し内に明示的に配置することで、パネル化された見出しにコンテンツを配置することができます。この場合、ALIGN-BORDERS=BODY は無視されます。

そのため、HEADPANEL を REPORT レベルでオンに設定し、さらに個々の中間見出しや中間脚注で明示的にオフに設定しない場合や、すべての中間見出しと中間脚注で明示的にオンにした場合は、ALIGN-BORDERS=BODY は、すべての中間見出しと中間脚注の位置を、データと揃えます。それ以外の場合は、境界線のデフォルトの動作が保持され、ページ見出しおよびページ脚注で位置揃えされます。

単一パネル PDF レポートの中間見出しと中間脚注マージンの位置揃え

次のリクエストは、GGSALES データソースに対して実行され、レポート見出し、レポート脚注、ページ見出し、ページ脚注、各地区の中間見出しを表示します。中間見出しと中間脚注のマージンは、位置揃えされません (ALIGN-BORDERS=OFF ,$)。

```
DEFINE FILE GGSALES
SHOWCATPROD/A30 = CATEGORY || (' / ' | PRODUCT);
END
TABLE FILE GGSALES
SUM
     DOLLARS/I8M AS ''
BY REGION
BY ST
BY CITY
ACROSS SHOWCATPROD AS 'Product Sales'

ON REGION SUBHEAD
" "
"Subheading <+0>Region <REGION<+0> "
" "
ON REGION SUBTOTAL AS '*TOTAL'
ON TABLE SUBHEAD
"Report Heading"
" "
"TYPE=REPORT, ALIGN-BORDERS=OFF, BORDER=ON, $"
HEADING
"Page Heading "
" "
" "
" "
FOOTING
" "
"Page Footing<+0>Page <TABPAGENO "
ON TABLE SUBFOOT
```

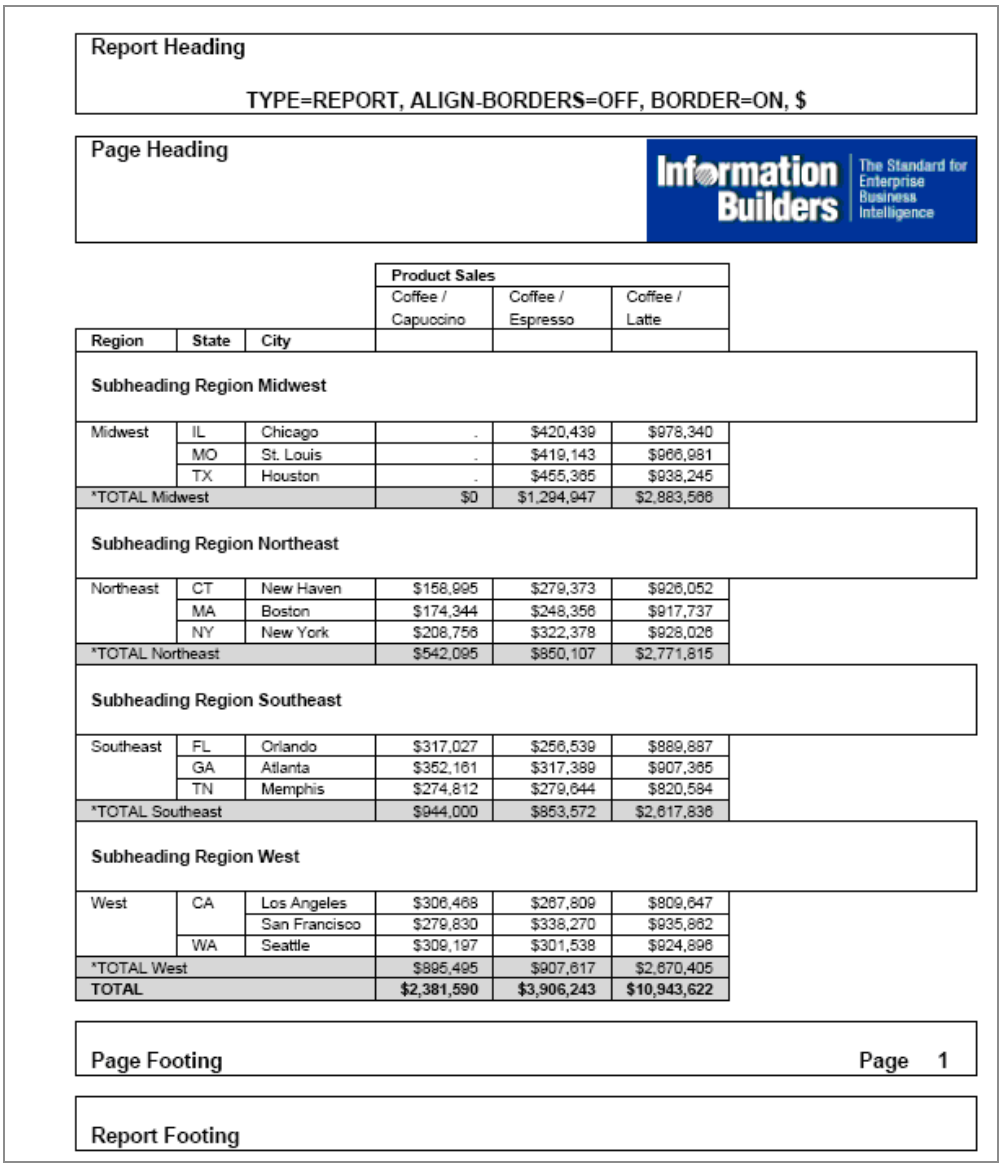
```
" "
"Report Footing"
```

```
WHERE CATEGORY EQ 'Coffee';
ON TABLE SET PAGE-NUM OFF
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
    TYPE=REPORT,
     FONT='ARIAL',
     SIZE=9,
     LEFTMARGIN=.75,
     RIGHTMARGIN=.5,
     TOPMARGIN=.1,
     BOTTOMMARGIN=.1,
     ALIGN-BORDERS=OFF,
     BORDER=ON,
     SQUEEZE=ON,$
$
TYPE=TITLE,
     STYLE=BOLD,
$
TYPE=TABHEADING,
     SIZE=12,
     STYLE=BOLD,
$
TYPE=TABHEADING,
     LINE=3,
     JUSTIFY=CENTER,
$
TYPE=TABFOOTING,
     SIZE=12,
     STYLE=BOLD,
$
TYPE=HEADING,
     SIZE=12,
     STYLE=BOLD,
$
TYPE=HEADING,
     IMAGE=smplogo1.gif,
     POSITION=(+4.6000000 +0.03000000),
     JUSTIFY=RIGHT,
$
TYPE=FOOTING,
     SIZE=12,
     STYLE=BOLD,
```

```
$
TYPE=FOOTING,
   LINE=2,
   ITEM=2,
   OBJECT=TEXT,
   POSITION=6.3,
     SIZE=12,
     STYLE=BOLD,
```

```
$
TYPE=SUBHEAD,
     SIZE=10,
     STYLE=BOLD,
$
TYPE=SUBHEAD,
  LINE=2,
  ITEM=3,
  OBJECT=TEXT,
  POSITION=2.5,
$
TYPE=SUBFOOT,
     SIZE=10,
     STYLE=BOLD,
$
TYPE=SUBTOTAL,
     BACKCOLOR=RGB(210 210 210),
$
TYPE=ACROSSVALUE,
     SIZE=9,
     WRAP=ON,
$
TYPE=ACROSSTITLE,
     STYLE=BOLD,
$
TYPE=GRANDTOTAL,
     BACKCOLOR=RGB(210 210 210),
     STYLE=BOLD,
$
ENDSTYLE
END
```

出力で、中間見出しマージンの位置は、レポートではなく、見出しと揃えられます。



Report Heading

TYPE=REPORT, ALIGN-BORDERS=OFF, BORDER=ON, $

Page Heading

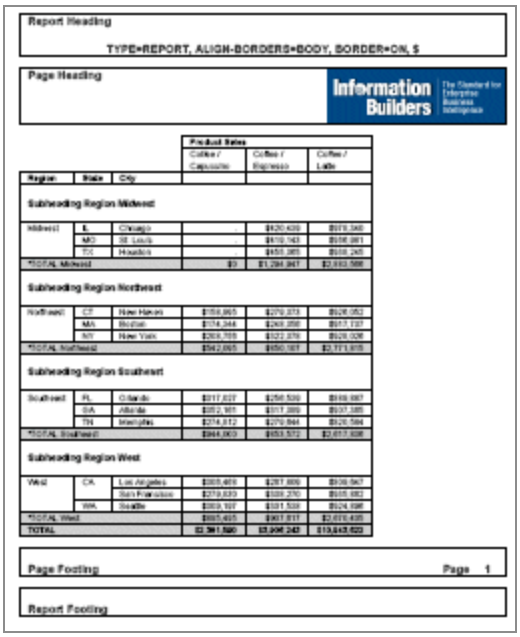
Information Builders — The Standard for Enterprise Business Intelligence

			Product Sales		
			Coffee / Capuccino	Coffee / Espresso	Coffee / Latte
Region	**State**	**City**			
Subheading Region Midwest					
Midwest	IL	Chicago	.	$420,439	$978,340
	MO	St. Louis	.	$419,143	$966,981
	TX	Houston	.	$455,365	$938,245
*TOTAL Midwest			$0	$1,294,947	$2,883,566
Subheading Region Northeast					
Northeast	CT	New Haven	$158,995	$279,373	$926,052
	MA	Boston	$174,344	$248,358	$917,737
	NY	New York	$208,756	$322,378	$928,026
*TOTAL Northeast			$542,095	$850,107	$2,771,815
Subheading Region Southeast					
Southeast	FL	Orlando	$317,027	$256,539	$889,887
	GA	Atlanta	$352,161	$317,389	$907,365
	TN	Memphis	$274,812	$279,644	$820,584
*TOTAL Southeast			$944,000	$853,572	$2,617,836
Subheading Region West					
West	CA	Los Angeles	$306,468	$267,809	$809,647
		San Francisco	$279,830	$338,270	$935,862
	WA	Seattle	$309,197	$301,538	$924,896
*TOTAL West			$895,495	$907,617	$2,670,405
TOTAL			**$2,381,590**	**$3,906,243**	**$10,943,622**

Page Footing Page 1

Report Footing

ALIGN-BORDERS 属性を ALIGN-BORDERS=BODY に変更し、リクエストを再び実行します。これにより、下図のように、中間見出しの位置は、レポートに揃えられます。



複数パネルレポートの中間見出しと中間脚注マージンの位置揃え

次のリクエストでは、すべての見出しおよび脚注に HEADPANEL=ON が設定されています。また、ALIGN-BORDERS=BODY 属性も設定されています。

```
SET BYPANEL=ON
DEFINE FILE GGSALES
SHOWCATPROD/A30 = CATEGORY || (' / ' | PRODUCT);
END
TABLE FILE GGSALES
SUM
     DOLLARS/I8M AS ''
BY REGION
BY ST
BY CITY
ACROSS SHOWCATPROD AS 'Product Sales'

ON REGION SUBHEAD
" "
"Subheading <+0>Region <REGION<+0> "
" "
ON REGION SUBTOTAL AS '*TOTAL'
ON TABLE SUBHEAD
```

```
"Report Heading"
" "
"TYPE=REPORT, ALIGN-BORDERS=BODY, HEADPANEL=ON, BORDER=ON, $"
HEADING
"Page Heading "
" "
" "
" "
FOOTING
" "
"Page Footing<+0>Page <TABPAGENO "
ON TABLE SUBFOOT
" "
"Report Footing"
```

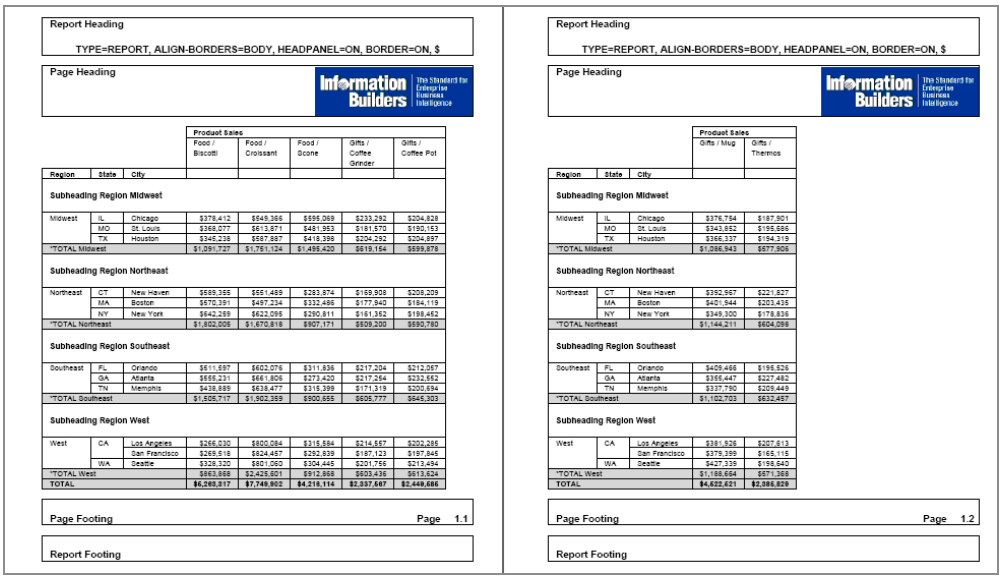
```
WHERE CATEGORY NE 'Coffee';
ON TABLE SET PAGE-NUM OFF
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
     UNITS=IN,
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,
$
TYPE=REPORT,
     FONT='ARIAL',
     SIZE=9,
     LEFTMARGIN=.75,
     RIGHTMARGIN=.5,
     TOPMARGIN=.1,
     BOTTOMMARGIN=.1,
     HEADPANEL=ON,
     ALIGN-BORDERS=BODY,
     BORDER=ON,
$
TYPE=TITLE,
     STYLE=BOLD,
$
TYPE=TABHEADING,
     SIZE=12,
     STYLE=BOLD,
$
TYPE=TABHEADING,
     LINE=3,
     JUSTIFY=CENTER,
$
```

```
TYPE=TABFOOTING,
     SIZE=12,
     STYLE=BOLD,
$
TYPE=HEADING,
     SIZE=12,
     STYLE=BOLD,
$
TYPE=HEADING,
     IMAGE=smplogo1.gif,
     POSITION=(+4.6000000 +0.03000000),
     JUSTIFY=RIGHT,
```

```
$
TYPE=FOOTING,
     SIZE=12,
     STYLE=BOLD,
$
TYPE=FOOTING,
   LINE=2,
   ITEM=2,
   OBJECT=TEXT,
   POSITION=6.3,
     SIZE=12,
     STYLE=BOLD,
$
TYPE=SUBHEAD,
     SIZE=10,
     STYLE=BOLD,
$
TYPE=SUBHEAD,
   LINE=2,
   ITEM=3,
   OBJECT=TEXT,
   POSITION=2.5,
$
TYPE=SUBFOOT,
     SIZE=10,
     STYLE=BOLD,
$
TYPE=SUBTOTAL,
     BACKCOLOR=RGB(210 210 210),
$
TYPE=ACROSSVALUE,
     SIZE=9,
     WRAP=ON,
```

```
$
TYPE=ACROSSTITLE,
     STYLE=BOLD,
$
TYPE=GRANDTOTAL,
     BACKCOLOR=RGB(210 210 210),
     STYLE=BOLD,
$
ENDSTYLE
END
```

出力で、中間見出しと中間脚注の位置は、各パネルのデータに揃えられます。



罫線の追加および調整 (PDF)

この構文は、PDF レポートに適用されます。

```
TYPE=type, {HGRID|VGRID}={ON|OFF|HEAVY}, $
```

説明

type

罫線が適用されるレポートコンポーネントを識別します。有効な値についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

HGRID

横方向の罫線を指定します。

VGRID

縦方向の罫線を指定します。

ON

細い罫線を適用します。

OFF

罫線を非表示にします。デフォルト値は OFF です。

HEAVY

太い罫線を適用します。

レポートデータへの罫線の適用 (PDF)

次のリクエストは、レポートデータに細い横方向罫線を適用します。

```
SET ONLINE-FMT = PDF
TABLE FILE GGDEMOG
HEADING
"State Statistics"
" "
SUM HH AS 'Number of,Households' AVGHHSZ98 AS 'Avg.,Size'
MEDHHI98 AS 'Avg.,Income'
BY ST
WHERE ST EQ 'CA' OR 'FL' OR 'NY'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=DATA, HGRID=ON, $
ENDSTYLE
END
```

このPDF レポートでは、罫線によって州ごとのデータが区別しやすくなります。

State Statistics

State	Number of Households	Avg. Size	Avg. Income
CA	11478067	3	44925
FL	5968392	2	34264
NY	6799434	3	42023

PPTX および PDF フォーマットでのボックス境界線の定義

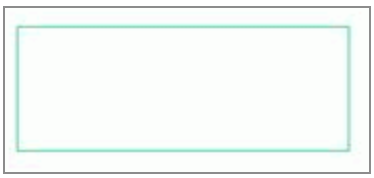
PPTX および PDF フォーマットでは、ボックスの背景色を、ボックスの境界色から独立して定義することができます。

背景色が存在しない場合の境界線

次の例のように、背景色が存在せず、境界色が定義されている場合、ボックスの境界線 (枠線) は、指定された境界色で表示されます。

```
TABLE FILE GGSALES
BY REGION NOPRINT
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
TYPE=REPORT, OBJECT=BOX, POSITION=(1 1), DIMENSION=(2 1), BORDER-
COLOR=GREEN,$
ENDSTYLE
END
```

出力結果は次のとおりです。



境界線が存在しない場合の背景色

PPTX フォーマットでは、境界線が存在せず (BORDER-STYLE=NONE)、背景色が定義されている場合、ボックスには背景色として定義された色がそのまま表示されます。

```
TABLE FILE GGSALES
BY REGION NOPRINT
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
TYPE=REPORT, OBJECT=BOX, POSITION=(1 1), DIMENSION=(2 1), BACKCOLOR=GREEN,
BORDER-STYLE=NONE, $
ENDSTYLE
END
```

出力結果は次のとおりです。

PDF フォーマットでは、下図のように、背景色の周囲に灰色の枠線が表示されます。

サポートされている境界線スタイル

PPTX および PDF フォーマットでは、3D 境界線スタイル (例、稜線、溝線、凹線、凸線) 以外の境界線スタイルがサポートされます。

```
TABLE FILE GGSALES
BY REGION NOPRINT
ON TABLE PCHOLD FORMAT PPTX
ON TABLE SET STYLE *
TYPE=REPORT, OBJECT=BOX, POSITION=(1 1), DIMENSION=(2 1), BORDER-COLOR=GREEN,
BORDER-STYLE=DASHED, $
ENDSTYLE
END
```

出力結果は次のとおりです。

注意：スタイルシート構文で FORMAT PPTX および FORMAT PDF とともに OBJECT=BOX および BORDER-STYLE=DOUBLE を使用すると、二重線の境界の代わりに、実線の境界が生成されます。

データ行、見出し行、脚注行での上付き文字の表示

HTML マークアップタグを使用することにより、テキストオブジェクトのテキストスタイルとして上付き文字がサポートされます。上付き文字マークアップタグのサポート対象は、HTML、PDF、PPTX 出力フォーマットのデータ列、見出し、脚注です。上付き文字は、データ内で定義できるほか、一時項目 (DEFINE) に追加したり、見出しおよび脚注内のテキストに追加したりできます。

HTML マークアップタグの変換を有効にするには、スタイルシートで、上付き文字を表示するレポートコンポーネントに対して MARKUP=ON を設定します。この属性を設定しない場合、マークアップタグは、タグではなく、テキストとして処理されます。

注意：XLSX 出力フォーマットでは、STYLE=SUPERSCRIPT 構文を使用して、上付き文字を指定することができます。その他すべての出力フォーマットでは、STYLE=SUPERSCRIPT 構文は無視されます。

レポートのデータ行、見出し行、脚注行での上付き文字の表示

タグがデータ自体に含まれていない場合、上付き文字として使用するテキストを格納するフィールドを作成します。また、上付き文字を表示するコンポーネントに対して、マークアップタグをオンに設定します。

- DEFINE および COMPUTE コマンドでは、上付き文字として使用するテキストを格納するフィールドを定義します。

  DEFINE FILE コマンドの場合、構文は次のとおりです。

  ```
  DEFINE FILE ...
  field/An = <sup>text</sup>;
  END
  ```

  COMPUTE コマンドまたはマスターファイル内の DEFINE の場合、構文は次のとおりです。

```
{COMPUTE|DEFINE} field/An = <sup>text</sup>;
```

説明

n

上付き文字を定義する文字列の長さです。この長さには、上付き文字として使用するテキストと、開始および終了マークアップタグ (^{および}) が含まれます。

text

上付き文字として使用するテキストです。

- スタイルシートで、上付き文字を表示するレポートコンポーネントに対して MARKUP=ON を設定します。

```
TYPE=component,MARKUP=ON ... ,$
```

説明

component

レポートコンポーネントのいずれかを指定します。これらのコンポーネントには、DATA、HEADING、FOOTING、SUBHEAD、SUBFOOT、TABHEADING、TABFOOTING があります。

PDF 出力のデータ行および脚注行での上付き文字の表示

次のリクエストは、GGSALES データソースを使用し、上付き文字として表示する 3 つのフィールドを定義します。SUP1 には「1」、SUP2 には「2」という数字を格納します。SUPCOPY には、著作権記号を格納します。これらのフィールドの構文で、一方ではテキスト値が定義されているのに対し、他方では HEX 値が定義されている点に注意してください。

COMPUTE コマンドは、売上額と販売目標額を比較します。計算値が、定義された最小値に満たない場合、カテゴリ名の末尾に上付き文字 SUP1 が連結されます。最小値を超える場合、SUP2 が連結されます。

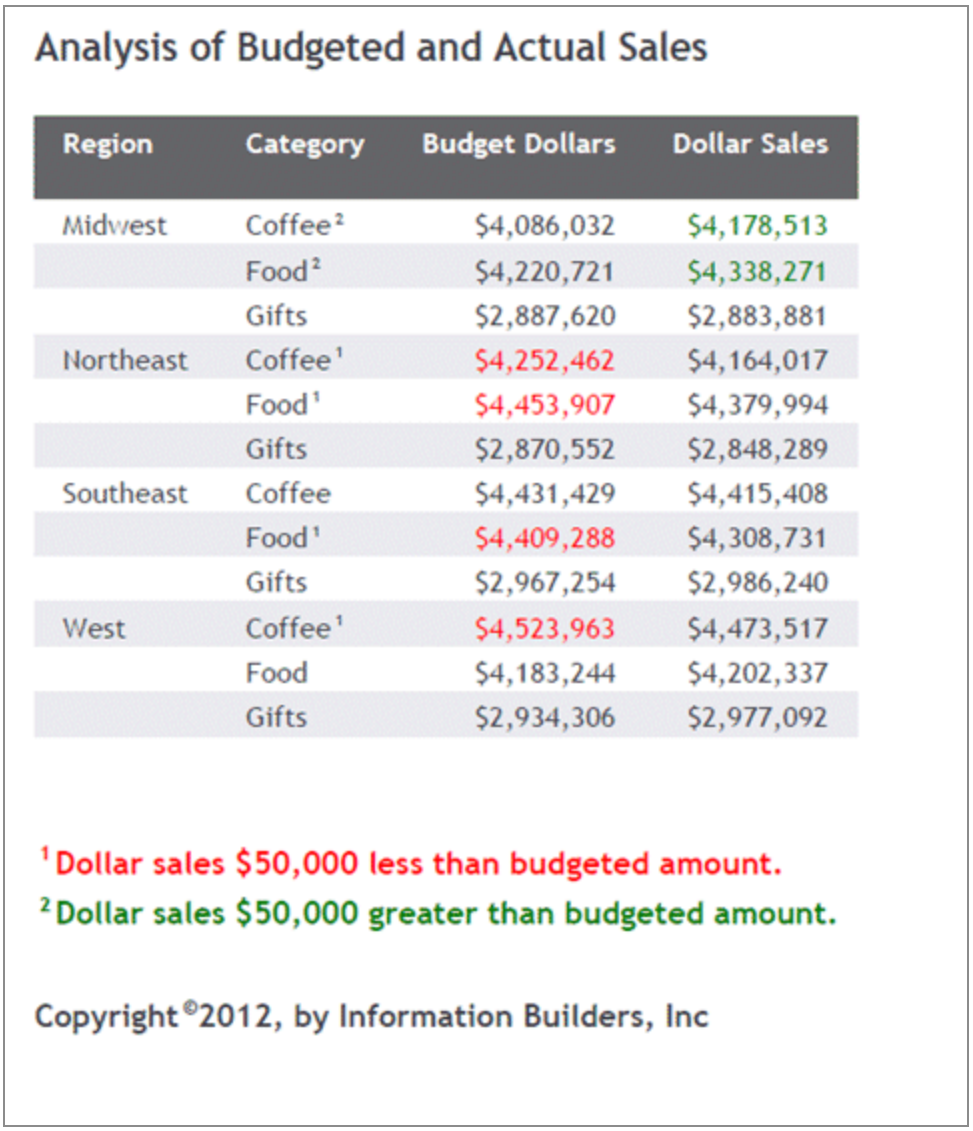
上付き文字の SUPCOPY は、レポートの脚注に著作権記号を表示するために使用されます。

脚注では、上付き文字フィールドが説明文の前に連結されます。

スタイルシートでは、上付き文字を表示する各コンポーネントで MARKUP=ON 属性が指定されています。

```
DEFINE FILE GGSALES
SUP1/A12= '<SUP>1</SUP>';
SUP2/A15= '<SUP>2</SUP>';
SUPCOPY/A20= '<SUP>'||HEXBYT(169,'A2')||'</SUP>';
END
TABLE FILE GGSALES
SUM
COMPUTE PROFIT/D12CM=DOLLARS-BUDDOLLARS; NOPRINT
COMPUTE SHOWCAT/A100=IF PROFIT LE -50000 THEN CATEGORY || SUP1
       ELSE IF PROFIT GT 50000 THEN CATEGORY || SUP2
           ELSE CATEGORY; AS Category
BUDDOLLARS/D12CM
DOLLARS/D12CM
BY REGION
BY CATEGORY NOPRINT
HEADING
"Analysis of Budgeted and Actual Sales"
FOOTING
""
"<SUP1 Dollar sales $50,000 less than budgeted amount."
"<SUP2 Dollar sales $50,000 greater than budgeted amount."
""
"Copyright<SUPCOPY 2012, by Information Builders, Inc "
ON TABLE SET HTMLCSS ON
ON TABLE SET SQUEEZE ON
ON TABLE SET PAGE-NUM OFF
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/javaassist/intl/EN/ENIADefault_combine.sty,$
TYPE=DATA,MARKUP=ON,$
TYPE=DATA,COLUMN=N5, COLOR=RED, WHEN=PROFIT LT -50000,$
TYPE=DATA,COLUMN=N6, COLOR=GREEN, WHEN=PROFIT GT 50000,$
TYPE=HEADING, JUSTIFY=LEFT,$
TYPE=FOOTING, MARKUP=ON, JUSTIFY=LEFT,$
TYPE=FOOTING, LINE=2,JUSTIFY=LEFT, COLOR=RED,$
TYPE=FOOTING, LINE=3,JUSTIFY=LEFT, COLOR=GREEN,$
ENDSTYLE
END
```

出力結果は次のとおりです。



Analysis of Budgeted and Actual Sales

Region	Category	Budget Dollars	Dollar Sales
Midwest	Coffee[2]	$4,086,032	$4,178,513
	Food[2]	$4,220,721	$4,338,271
	Gifts	$2,887,620	$2,883,881
Northeast	Coffee[1]	$4,252,462	$4,164,017
	Food[1]	$4,453,907	$4,379,994
	Gifts	$2,870,552	$2,848,289
Southeast	Coffee	$4,431,429	$4,415,408
	Food[1]	$4,409,288	$4,308,731
	Gifts	$2,967,254	$2,986,240
West	Coffee[1]	$4,523,963	$4,473,517
	Food	$4,183,244	$4,202,337
	Gifts	$2,934,306	$2,977,092

[1] Dollar sales $50,000 less than budgeted amount.

[2] Dollar sales $50,000 greater than budgeted amount.

Copyright ©2012, by Information Builders, Inc

XLSX 出力の見出しおよび脚注行での上付き文字の表示

次のリクエストは、GGSALES データソースを使用し、見出しおよび脚注の商標記号と著作権記号を上付き文字に設定します。COPYRIGHT には、著作権記号を格納します。見出し行と脚注行に、STYLE = SUPERSCRIPT 構文が含まれています。

```
DEFINE FILE GGSALES
COPYRIGHT/A1= HEXBYT(169, 'A1');
```

```
END
TABLE FILE GGSALES
SUM UNITS BY CATEGORY
HEADING
"Company-Trademark<+0>TM"
" "
"Company-Copyright<+0><COPYRIGHT"
" "
FOOTING
"Company-Trademark<+0>TM"
" "
"Company-Copyright<+0><COPYRIGHT"
" "
ON TABLE PCHOLD FORMAT XLSX
ON TABLE SET STYLE *
TYPE=HEADING, WRAP=OFF, $
TYPE=HEADING, LINE=1, ITEM=2, OBJECT=TEXT, STYLE='SUPERSCRIPT+BOLD+ITALIC',
COLOR=RED, $
TYPE=HEADING, LINE=3, ITEM=1, OBJECT=FIELD, STYLE='SUPERSCRIPT+BOLD',
COLOR=GREEN, $
TYPE=FOOTING, LINE=1, ITEM=2, OBJECT=TEXT, STYLE='SUPERSCRIPT+BOLD+ITALIC',
COLOR=RED, $
TYPE=FOOTING, LINE=3, ITEM=1, OBJECT=FIELD, STYLE='SUPERSCRIPT+BOLD',
COLOR=GREEN, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	A	B
1	Company-Trademark™	
2		
3	Company-Copyright©	
4		
5	Category	Unit Sales
6	Coffee	1376266
7	Food	1384845
8	Gifts	927880
9	Company-Trademark™	
10		
11	Company-Copyright©	

下線およびブランク行の追加

ブランク行または下線を使用してセクションに分割することで、詳細な表レポートを読みやすくします。

HTML レポートに罫線を表示する場合、ブランク行や下線を追加することはできません。GRID 属性を OFF にすると、ブランク行または下線を追加できます。

ブランク行を挿入するときは、ページにデータを表示後にスペースを空けるために、LINES パラメータの設定は PAPER パラメータの設定より少なくとも 1 つ小さくする必要があります。

列番号付きの FML (Financial Modeling Language) レポートには、デフォルト設定で、RECAP 演算の前に読みやすくするための下線が追加されます。このタイプのレポート内で、デフォルト下線を細い線から太い線 (PDF レポート内では一重から二重に) に変更できます。

セクション分割機能

機能	説明	適用先
`SKIP-LINE*`	ブランク行を追加します。	HTML (GRID=OFF が必要) DHTML PDF XLSX EXL2K
`TYPE=SKIPLINE`	ブランク行にフォーマットを設定します。	DHTML PDF
`UNDER-LINE*`	ソートグループに下線を付けます。	HTML (GRID=OFF が必要) DHTML PDF

機能	説明	適用先
`TYPE=UNDERLINE`	下線にフォーマットを設定します。	DHTML PDF
`STYLE={+|-}UNDERLINE*`	フィールドタイトル以外のレポートコンポーネントに下線を追加、またはレポートコンポーネントから下線を削除します。	HTML DHTML PDF XLSX EXL2K
`STYLE={+|-}EXTUNDERLINE*`	スタイルレポートのレポート列幅全体にまで下線を拡張、またはレポート列幅全体に拡張された下線を削除します。	DHTML PDF PPT PPTX
`BAR AS '{-|=}'*`	FML レポート内で、一重下線または二重下線を選択します。 HTML の場合は、細い下線または太い下線を選択します。	HTML DHTML PDF XLSX EXL2K

* 境界線ではサポートされません。

ブランク行の追加

1 つのリクエストに対しては SKIP-LINE は 1 つのみ使用できます。

```
display_command fieldname SKIP-LINE
```

または

```
{ON|BY} fieldname SKIP-LINE [WHEN expression;]
```

説明

display_command

表示コマンドです。

fieldname

ブランク行が挿入された後の表示またはソートフィールドです。

表示フィールドとともに使用した SKIP-LINE は、それぞれの表示行の後にブランク行を追加します。その結果、レポートに 2 行分のブランクが追加されます (ダブルスペース)。ダブルスペースは、レポート作成者が個々の行の下にコメントを書きやすくすることから、レポートのレビューに役立ちます。

ソートフィールドとともに使用した SKIP-LINE は、そのフィールド値が変更される前にブランク行を追加します。これは、唯一 BY フィールドを参照する必要がない コマンドの 1 つです。

ON|BY

縦のソート句です。これらの句は同義です。

WHEN expression

論理式で決定される、レポート表示内の条件付きブランク行を指定します。詳細は、式の使用を参照してください。

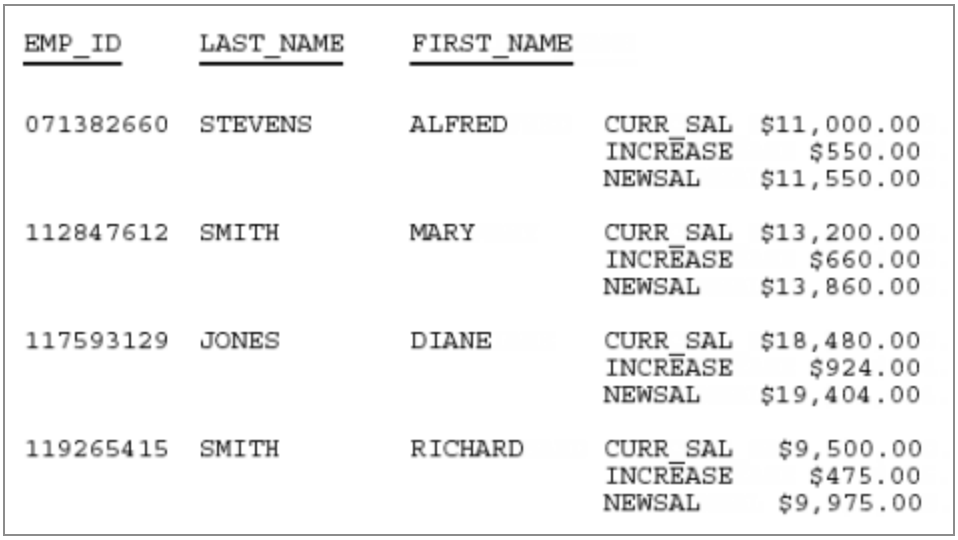
ソートグループ間へのブランク行の追加

次のリクエストは、ソートフィールド EMP_ID の値が変更される前にブランク行を挿入します。

```
DEFINE FILE EMPLOYEE
INCREASE/D8.2M = .05*CURR_SAL;
CURR_SAL/D8.2M=CURR_SAL;
NEWSAL/D8.2M=CURR_SAL + INCREASE;
END

TABLE FILE EMPLOYEE
PRINT CURR_SAL OVER INCREASE OVER NEWSAL
BY LOWEST 4 EMP_ID BY LAST_NAME BY FIRST_NAME
```

```
ON EMP_ID SKIP-LINE
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT PDF
END
```

各従業員のデータは強調され、読みやすくなります。



EMP_ID	LAST_NAME	FIRST_NAME		
071382660	STEVENS	ALFRED	CURR_SAL	$11,000.00
			INCREASE	$550.00
			NEWSAL	$11,550.00
112847612	SMITH	MARY	CURR_SAL	$13,200.00
			INCREASE	$660.00
			NEWSAL	$13,860.00
117593129	JONES	DIANE	CURR_SAL	$18,480.00
			INCREASE	$924.00
			NEWSAL	$19,404.00
119265415	SMITH	RICHARD	CURR_SAL	$9,500.00
			INCREASE	$475.00
			NEWSAL	$9,975.00

ブランク行のフォーマット

```
TYPE=SKIPLINE, attribute=value, $
```

説明

attribute

有効なスタイルシート属性です。

value

属性の値です。

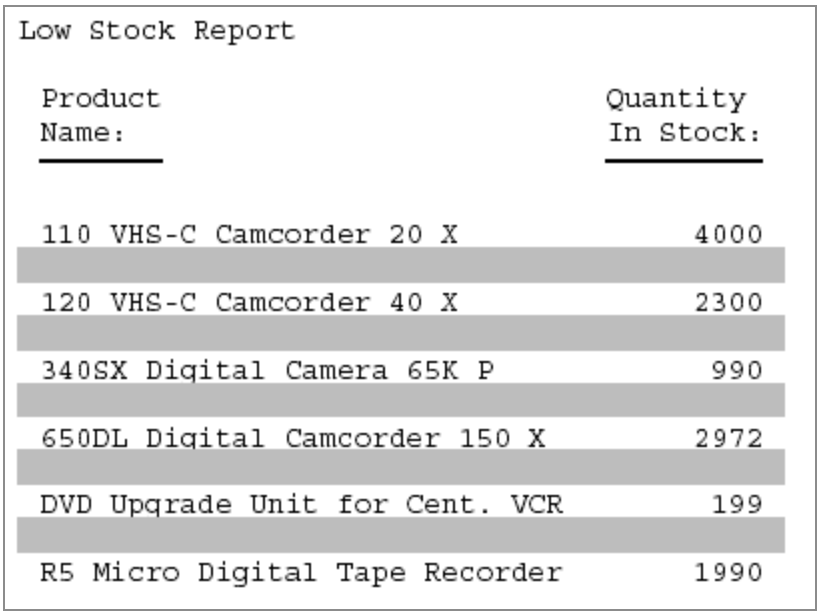
注意：このオプションは、内部カスケードスタイルシートと併用された場合に、PDF および HTML レポートでサポートされます。

ブランク行への色の追加

次のリクエストでは、ブランク行にフォーマットが設定され、出力内で銀色で表示されます。リクエストの太字の部分が、この例に関連するスタイルシートです。

```
SET ONLINE-FMT=PDF
TABLE FILE CENTINV
HEADING
"Low Stock Report"
" "
SUM QTY_IN_STOCK
WHERE QTY_IN_STOCK LT 5000
BY PRODNAME
ON PRODNAME SKIP-LINE
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=SKIPLINE, BACKCOLOR=SILVER, $
ENDSTYLE
END
```

レポートは次のとおりです。



Low Stock Report

Product Name:	Quantity In Stock:
110 VHS-C Camcorder 20 X	4000
120 VHS-C Camcorder 40 X	2300
340SX Digital Camera 65K P	990
650DL Digital Camcorder 150 X	2972
DVD Upgrade Unit for Cent. VCR	199
R5 Micro Digital Tape Recorder	1990

ソートグループの下線

```
{ON|BY} fieldname UNDER-LINE [WHEN expression;]
```

説明

ON|BY

縦のソート句です。これらの句は同義です。

fieldname

下線が適用されるソートフィールドです。UNDER-LINE は、ソートフィールドの値が変わるところで下線を追加します。下線は、RECAP や SUB-TOTAL などのオプションの後に、自動的に表示されますが、改ページの前に表示されます。

WHEN expression

論理式により決定される、レポート表示内の条件付き下線を指定します。詳細は、式の使用を参照してください。

ソートグループの下線

次のリクエストは、ソートフィールドBANK_NAME の値が変更されるところで下線が追加されます。GRID 属性は OFF に設定されていますが、これは、HTML レポートで必須です。

```
TABLE FILE EMPLOYEE
PRINT EMP_ID AND BANK_ACCT AND LAST_NAME
BY BANK_NAME
ON BANK_NAME UNDER-LINE
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

各銀行のデータは強調され、見やすくなります。

BANK NAME	EMP ID	BANK ACCT	LAST NAME
	071382660		STEVENS
	112847612		SMITH
	119265415		SMITH
	126724188		ROMANS
	219984371		MCCOY
	543729165		GREENSPAN
ASSOCIATED	123764317	819000702	IRVING
	326179357	122850108	BLACKWOOD
	451123478	136500120	MCKNIGHT
BANK ASSOCIATION	818692173	163800144	CROSS
BEST BANK	119329144	160633	BANNING
STATE	117593129	40950036	JONES

下線のフォーマット

```
TYPE=UNDERLINE ...COLOR={color|RGB} (r g b), $
```

説明

UNDERLINE

ON *fieldname* UNDER-LINE により生成された下線を表示します。

COLOR

下線の色を指定します。ディスプレイデバイスまたは出力デバイスが指定した色をサポートしない場合、灰色に置き換えられます。デフォルト値は黒です。

color

サポートされる色の値です。サポートされている値のリストについては、レポートのフォントフォーマット指定を参照してください。

RGB

赤、緑、青の組み合わせを使用してフォントの色を指定します。

(r g b)

赤、緑、青のそれぞれの最適な輝度です。0 (ゼロ) から 255 までの値が使用されます。0 (ゼロ) は最も輝度が低く、255 は最も輝度が高いことを示します。

色の構成要素を同じ輝度にすると、結果は灰色になります。

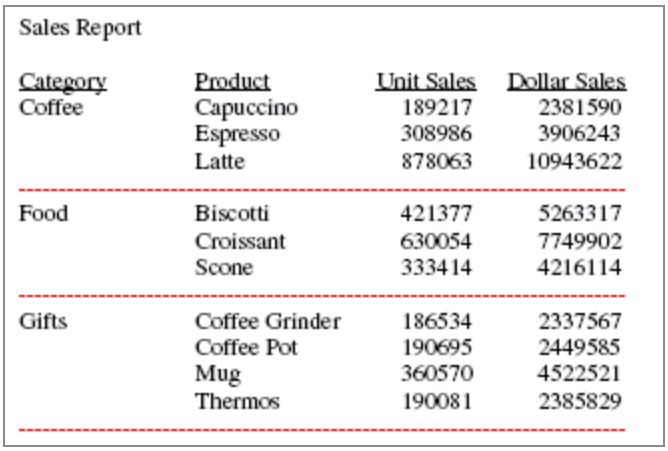
注意：このオプションは、内部カスケードスタイルシートと併用された場合に、PDF および HTML レポートでサポートされます。

ソートグループの下線フォーマット

次のリクエストは、UNDERLINE を使用して下線の色をデフォルトの黒から赤に変更します。

```
SET ONLINE-FMT = PDF
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY CATEGORY BY PRODUCT
HEADING
"Sales Report"
" "
ON CATEGORY UNDER-LINE
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=UNDERLINE, COLOR=RED, $
ENDSTYLE
END
```

結果のソートグループ値間で、相違が強調されます。PDF レポートは次のとおりです。



Sales Report

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

レポートコンポーネント下線の追加または削除

```
TYPE=type, [subtype,] STYLE=[+|-]UNDERLINE, $
```

説明

type

レポートコンポーネントです。有効な値についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

subtype

COLUMN、ACROSS、または ITEM などの追加属性で、レポートコンポーネントを識別するために必要です。有効な値についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

+

継承したテキストスタイルに下線を追加、またはテキストスタイル (例、STYLE=BOLD+UNDERLINE) の組み合わせを指定します。これがデフォルト値です。

-

継承したテキストスタイルから下線を削除します。

フィールドタイトルからの下線の削除

次の構文は、内部カスケードスタイルシートを使用する HTML レポートに適用されます。

```
TYPE=TITLE, [COLUMN=column,] STYLE=-UNDERLINE, $
```

説明

COLUMN=column

フィールドを指定します。有効な値についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

フィールド下線の追加およびフィールドタイトル下線の削除

次のリクエストは、内部カスケードスタイルシートを使用した HTML レポートで、CATEGORY フィールド値に下線を追加し、フィールドタイトルからデフォルト下線を削除します。

```
SET HTMLCSS = ON
TABLE FILE MOVIES
PRINT TITLE DIRECTOR
BY CATEGORY
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=TITLE, STYLE=-UNDERLINE, $
TYPE=REPORT, COLUMN=CATEGORY, STYLE=UNDERLINE, $
ENDSTYLE
END
```

以下は、レポートの一部を示しています。

CATEGORY	TITLE	DIRECTOR
ACTION	JAWS	SPIELBERG S.
	ROBOCOP	VERHOVEN P.
	TOTAL RECALL	VERHOVEN P.
	TOP GUN	SCOTT T.
	RAMBO III	MCDONALD P.
CHILDREN	SMURFS, THE	
	SHAGGY DOG, THE	BARTON C.
	SCOOBY-DOO-A DOG IN THE RUFF	
	ALICE IN WONDERLAND	GEROMINI
	SESAME STREET-BEDTIME STORIES AND SONGS	
	ROMPER ROOM-ASK MISS MOLLY	
	SLEEPING BEAUTY	DISNEY W.
	BAMBI	DISNEY W.
CLASSIC	EAST OF EDEN	KAZAN E.
	CITIZEN KANE	WELLES O.
	CYRANO DE BERGERAC	GORDON M.
	MARTY	MANN D.
	MALTESE FALCON, THE	HUSTON J.
	GONE WITH THE WIND	FLEMING V.

レポート列幅全体に下線を拡張

デフォルト設定では、レポート列タイトルの下線は、列タイトルテキストの開始位置から終了位置を超えて表示することはできません。WebFOCUS スタイルシートで EXTUNDERLINE オプションを使用して、スタイル設定された出力のレポート列幅全体に、下線を引く位置を拡張することができます。EXTUNDERLINE は、TITLE レポートコンポーネントの STYLE 属性オプションです。このオプションは、DHTML、PDF、PPT、PPTX の各フォーマットでサポートされます。

```
TYPE = TITLE, [COLUMN = colspec,] STYLE = [+|-]EXTUNDERLINE  ,$
```

説明

colspec

有効な列の指定です。

+EXTUNDERLINE

継承したテキストスタイルに下線を追加、またはテキストスタイル (たとえば STYLE=BOLD+UNDERLINE) の組み合わせを指定します。

-EXTUNDERLINE

継承したテキストスタイルの EXTUNDERLINE オプションを解除します。

EXTUNDERLINE 属性使用上の注意

- 列幅の計算とレポートの表示は、ブラウザが行うため、HTML フォーマットはサポートされていません。
- GRID で指定する線は列幅全体に拡張されるため、GRID=ON および EXTUNDERLINE は、相互に排他的です。列タイトル下線に指定されたスタイル設定は、すべて GRID の設定で上書きされます。

下線をレポート列幅全体に拡張

次のリクエストは、GGSALES データソースに対して実行され、売上 (DOLLARS) を都市 (CITY) と日付 (DATE) で集計します。

```
DEFINE FILE GGSALES
YEAR/YY = DATE;
MONTH/M = DATE;
END
TABLE FILE GGSALES
SUM DOLLARS AS 'Sales'
BY DATE
BY CITY
WHERE YEAR EQ 1997
WHERE MONTH FROM 01 TO 05
WHERE CITY EQ 'Seattle' OR 'San Francisco' OR 'Los Angeles'
ON TABLE SET PAGE NOPAGE
ON TABLE PCHOLD FORMAT DHTML
END
```

出力では、列タイトルにのみ下線が引かれています。

Date	City	Sales
1997/01/01	Los Angeles	159935
	San Francisco	162712
	Seattle	159804
1997/02/01	Los Angeles	148915
	San Francisco	143987
	Seattle	132505
1997/03/01	Los Angeles	165902
	San Francisco	145129
	Seattle	165847
1997/04/01	Los Angeles	146106
	San Francisco	158799
	Seattle	144169
1997/05/01	Los Angeles	178336
	San Francisco	144534
	Seattle	190208

列幅全体に下線を引くには、スタイル設定可能なフォーマットで出力を生成し、TITLE コンポーネントの STYLE 属性で、EXTUNDERLINE オプションを使用します。次のリクエストは、DHTML 出力を作成し、列タイトルに太字、左揃え、下線を設定します。下線はレポートの列幅全体に拡張されます。

```
DEFINE FILE GGSALES
YEAR/YY = DATE;
MONTH/M = DATE;
END
TABLE FILE GGSALES
SUM DOLLARS AS 'Sales'
BY DATE
BY CITY
WHERE YEAR EQ 1997
WHERE MONTH FROM 01 TO 05
WHERE CITY EQ 'Seattle' OR 'San Francisco' OR 'Los Angeles'
ON TABLE SET PAGE NOPAGE
ON TABLE PCHOLD FORMAT DHTML
ON TABLE SET STYLE *
TYPE=TITLE, STYLE= BOLD +EXTUNDERLINE, JUSTIFY=LEFT, $
ENDSTYLE
END
```

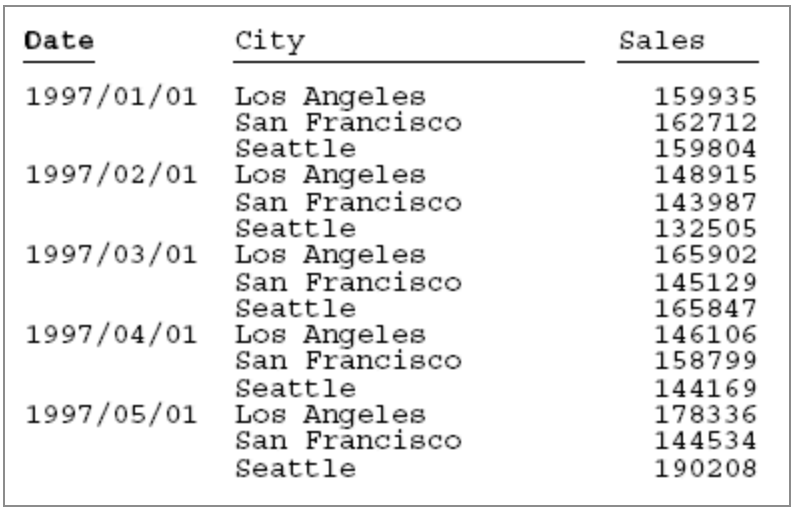
出力結果は次のとおりです。

Date	City	Sales
1997/01/01	Los Angeles	159935
	San Francisco	162712
	Seattle	159804
1997/02/01	Los Angeles	148915
	San Francisco	143987
	Seattle	132505
1997/03/01	Los Angeles	165902
	San Francisco	145129
	Seattle	165847
1997/04/01	Los Angeles	146106
	San Francisco	158799
	Seattle	144169
1997/05/01	Los Angeles	178336
	San Francisco	144534
	Seattle	190208

次のリクエストは、TITLE コンポーネントで、EXTUNDERLINE オプションおよび JUSTIFY=LEFT オプションをデフォルトとして設定し、さらに、DATE 列タイトルに太字を設定して、下線の拡張を解除します。

```
DEFINE FILE GGSALES
YEAR/YY = DATE;
MONTH/M = DATE;
END
TABLE FILE GGSALES
SUM DOLLARS AS 'Sales'
BY DATE
BY CITY
WHERE YEAR EQ 1997
WHERE MONTH FROM 01 TO 05
WHERE CITY EQ 'Seattle' OR 'San Francisco' OR 'Los Angeles'
ON TABLE SET PAGE NOPAGE
ON TABLE PCHOLD FORMAT DHTML
ON TABLE SET STYLE *
TYPE=TITLE,STYLE= EXTUNDERLINE, JUSTIFY=LEFT ,$
TYPE=TITLE,COLUMN= DATE, STYLE= -EXTUNDERLINE +BOLD ,$
ENDSTYLE
END
```

出力結果は次のとおりです。



Date	City	Sales
1997/01/01	Los Angeles	159935
	San Francisco	162712
	Seattle	159804
1997/02/01	Los Angeles	148915
	San Francisco	143987
	Seattle	132505
1997/03/01	Los Angeles	165902
	San Francisco	145129
	Seattle	165847
1997/04/01	Los Angeles	146106
	San Francisco	158799
	Seattle	144169
1997/05/01	Los Angeles	178336
	San Francisco	144534
	Seattle	190208

FML レポート内の下線の太さの変更

この構文は、HTML レポートに適用されます。

```
BAR [AS '{-|=}'] OVER
```

説明

-

細い下線を生成します。一重引用符 (') でハイフン (-) を囲みます。これがデフォルト値です。

=

太い下線を生成します。一重引用符 (') で等号 (=) を囲みます。

FML レポート (HTML) デフォルト下線の変更

次のリクエストは、FML レポートのデフォルトである細い下線を太い下線に変更します。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'       OVER
1020 AS 'DEMAND DEPOSITS'    OVER
```

```
1030 AS 'TIME DEPOSITS'      OVER
BAR AS '='                   OVER
RECAP TOTCASH = R1 + R2 + R3;
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

太い下線で TOTCASH を詳細データから分離し、強調します。

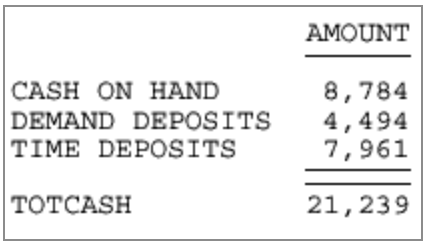
	AMOUNT
CASH ON HAND	8,784
DEMAND DEPOSITS	4,494
TIME DEPOSITS	7,961
TOTCASH	21,239

FML レポート (PDF) デフォルト下線の変更

次のリクエストは、PDF レポート内のデフォルトの一重下線を二重下線に変更します。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'       OVER
1020 AS 'DEMAND DEPOSITS'    OVER
1030 AS 'TIME DEPOSITS'      OVER
BAR AS '='                   OVER
RECAP TOTCASH = R1 + R2 + R3;
ON TABLE SET ONLINE-FMT PDF
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。



	AMOUNT
CASH ON HAND	8,784
DEMAND DEPOSITS	4,494
TIME DEPOSITS	7,961
TOTCASH	21,239

レポートからのブランク行の削除

DROPBLNKLINE パラメータは、WebFOCUS レポートにブランク行を表示するかどうかを制御します。このオプションを使用することで、レポート内のさまざまな位置に自動生成されるブランク行を制御することができます。中間合計、中間見出し、中間脚注の周囲のブランク行を削除したり、値が存在しないためにレポート出力でブランク行になる特定のデータ行を削除したりすることができます。また、境界線を使用する際に見出しおよび脚注の周囲に挿入されるブランク行を削除することもできます。これらのブランク行をレポート出力から削除するには、SET DROPBLNKLINE オプションを使用します。

レポート出力の自動ブランク行の制御

```
SET DROPBLNKLINE={OFF|ON|BODY|HEADING|ALL}
```

または

```
ON TABLE SET DROPBLNKLINE {OFF|ON|BODY|HEADING|ALL}
```

説明

OFF

システムが生成するブランク行および空のデータ行を挿入します。デフォルト値は OFF です。

ON|BODY

レポート本文内のシステム生成のブランク行を削除します (例、中間見出し前後のブランク行)。また、値が存在しないためにレポート出力でブランク行になる特定のデータ行を出力結果から削除します。BODY は ON の同義語です。

HEADING

見出しとタイトル間のブランク行、およびレポート本文と脚注間のブランク行を削除します。このオプションは、リクエストでレポート内の任意の位置に境界線または背景色スタイルシート属性が使用されている、固定レイアウトのフォーマット (PDF、DHTML、PPT、PPTX) で動作します。

ALL

ON と HEADING の両方の動作が適用されます。

SET DROPBLNKLINE=HEADING 使用時の注意

- 固定レイアウトのレポートフォーマット (PDF、DHTML、PPT、PPTX) で境界線または背景色を使用した場合、見出しの下および脚注の上にブランク行が自動的に生成されます。この自動生成は、各境界線が正しく表示されるように設計された動作です。一般に、行の境界線の設定は、各行の上境界線と左境界線で制御されます。そのため、見出しの下境界線は、見出しの下にある行の上境界線で設定されます。ブランク行が自動的に挿入されることで、見出しの下境界線と列タイトルの上境界線との競合が回避されます。自動的に挿入されたブランク行の上境界線には、見出しの下境界線として定義された境界線が配置されます。この動作は、データ行の下境界線と脚注の上境界線の場合にも適用されます。

  DROPBLNKLINE=HEADING を使用すると、自動的に挿入されるブランク行の高さを 0 (ゼロ) に設定することでブランク行が削除されます。その結果、見出しの下境界線が列タイトルの上境界線になります。境界線を配置せず、背景色のみを使用する場合は、この方法で色の隙間を埋めることができます。ただし、WebFOCUS 処理では種類の異なる線スタイルには対処できないため、各レポート要素で異なる境界線スタイルを使用すると、境界線スタイル定義で競合が発生する場合があります。各レポート要素の境界線スタイルを統一するには、レポートに表示する各要素に同一の線スタイルを使用します。

- 次の場合、DROPBLNKLINE=HEADING はサポートされません。
  - 見出しと列タイトル間、またはデータと脚注間に異なる境界線スタイルが使用されている場合。
  - レポートに ACROSS ソート句が使用されている場合。
- 使用上の注意

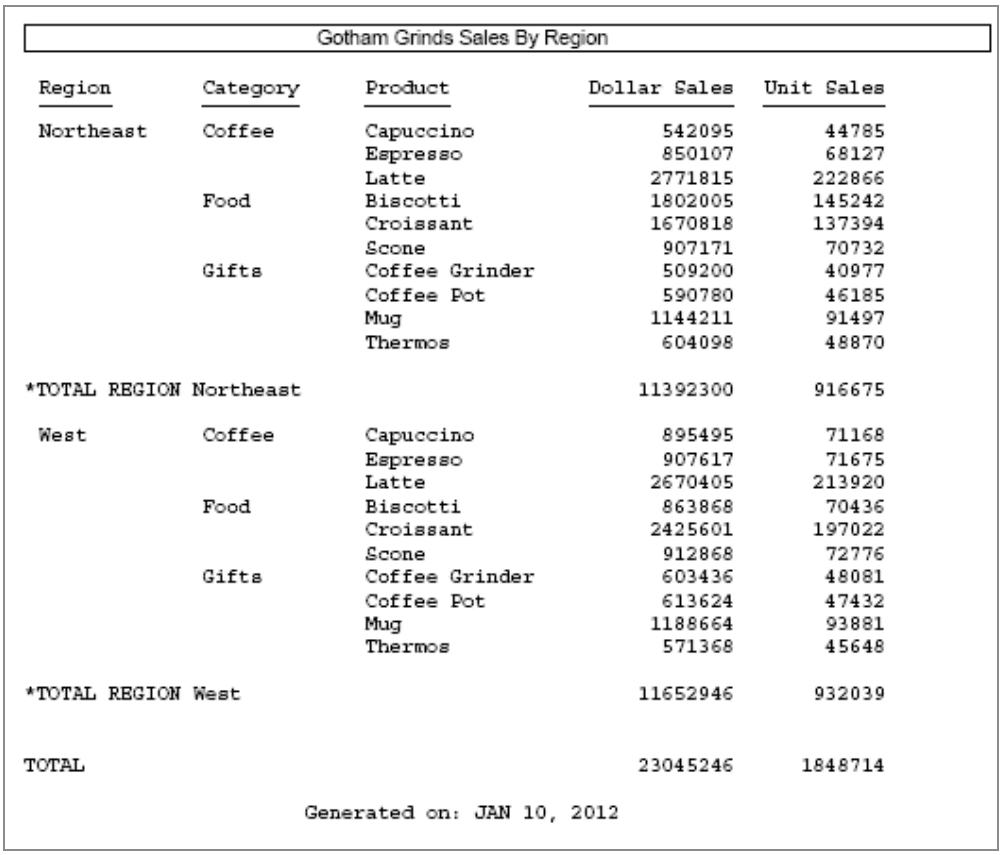
- 一部のレポートでは、FOOTING BOTTOM の要件により、データ行と脚注間にシステム生成のブランク行が追加され、セクション間の距離を適正に保つためのブランク領域が確保されます。この場合、FOOTING BOTTOM の上部とデータグリッドの下部が若干重なり合うことがあります。この問題を解決するには、脚注の上部にブランク行を追加します。
- 境界線はレポート全体に適用することをお勧めします (TYPE=REPORT)。これにより、各レポート要素に境界線を個別に設定する際の既知の問題を回避することができます。選択した要素のみに背景色および境界線を定義したレポートでは、見出しに適用された背景色の表示幅が、列タイトルに適用された背景色の表示幅と一致しない場合があります。この不一致により、見出しと列タイトルの右端で段差が発生します。また、背景色が設定された要素の境界線に背景色と同一の色を定義した場合 (BORDER-COLOR)、境界線の色が背景色に混じり合い、境界線が識別できなくなります。

DROPBLNKLINE パラメータ設定の比較

次のリクエストは、GGSALES データソースに対して実行され、見出し、脚注、中間合計を表示します。DROPBLNKLINE は OFF に設定されています。

```
TABLE FILE GGSALES
HEADING CENTER
"Gotham Grinds Sales By Region"
FOOTING CENTER
"Generated on: &DATETMDYY"
SUM DOLLARS UNITS
BY REGION SUBTOTAL
BY CATEGORY
BY PRODUCT
WHERE REGION EQ 'Northeast' OR 'West'
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET DROPBLNKLINE OFF
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
SQUEEZE = ON,
FONT = ARIAL,
TYPE=HEADING, BORDER=LIGHT,
$
ENDSTYLE
END
```

出力結果では、ブランク行が見出しの下、脚注の上、中間合計行の上下、総合計行の上下に挿入されています。

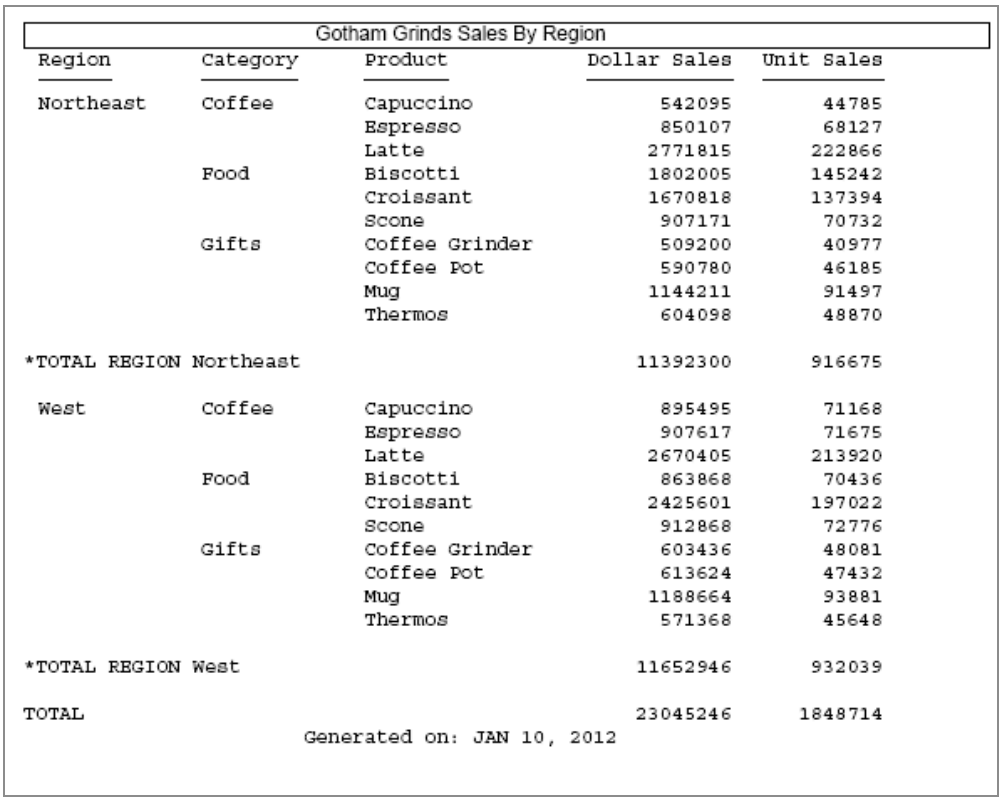


Gotham Grinds Sales By Region

Region	Category	Product	Dollar Sales	Unit Sales
Northeast	Coffee	Capuccino	542095	44785
		Espresso	850107	68127
		Latte	2771815	222866
	Food	Biscotti	1802005	145242
		Croissant	1670818	137394
		Scone	907171	70732
	Gifts	Coffee Grinder	509200	40977
		Coffee Pot	590780	46185
		Mug	1144211	91497
		Thermos	604098	48870
*TOTAL REGION Northeast			11392300	916675
West	Coffee	Capuccino	895495	71168
		Espresso	907617	71675
		Latte	2670405	213920
	Food	Biscotti	863868	70436
		Croissant	2425601	197022
		Scone	912868	72776
	Gifts	Coffee Grinder	603436	48081
		Coffee Pot	613624	47432
		Mug	1188664	93881
		Thermos	571368	45648
*TOTAL REGION West			11652946	932039
TOTAL			23045246	1848714

Generated on: JAN 10, 2012

DROPBLNKLINE 設定を HEADING に変更すると、次の出力結果が生成されます。見出しの下および脚注の上のブランク行が削除されています。中間合計行および総合計行の上下のブランク行は挿入されています。



Gotham Grinds Sales By Region

Region	Category	Product	Dollar Sales	Unit Sales
Northeast	Coffee	Capuccino	542095	44785
		Espresso	850107	68127
		Latte	2771815	222866
	Food	Biscotti	1802005	145242
		Croissant	1670818	137394
		Scone	907171	70732
	Gifts	Coffee Grinder	509200	40977
		Coffee Pot	590780	46185
		Mug	1144211	91497
		Thermos	604098	48870
*TOTAL REGION Northeast			11392300	916675
West	Coffee	Capuccino	895495	71168
		Espresso	907617	71675
		Latte	2670405	213920
	Food	Biscotti	863868	70436
		Croissant	2425601	197022
		Scone	912868	72776
	Gifts	Coffee Grinder	603436	48081
		Coffee Pot	613624	47432
		Mug	1188664	93881
		Thermos	571368	45648
*TOTAL REGION West			11652946	932039
TOTAL			23045246	1848714

Generated on: JAN 10, 2012

DROPBLNKLINE 設定を ON (または BODY) に変更すると、次の出力結果が生成されます。ここでは、中間合計行および総合計行の上下のブランク行は削除されていますが、見出しの下および脚注の上のブランク行は挿入されています。

Gotham Grinds Sales By Region

Region	Category	Product	Dollar Sales	Unit Sales
Northeast	Coffee	Capuccino	542095	44785
		Espresso	850107	68127
		Latte	2771815	222866
	Food	Biscotti	1802005	145242
		Croissant	1670818	137394
		Scone	907171	70732
	Gifts	Coffee Grinder	509200	40977
		Coffee Pot	590780	46185
		Mug	1144211	91497
		Thermos	604098	48870
*TOTAL REGION Northeast			11392300	916675
West	Coffee	Capuccino	895495	71168
		Espresso	907617	71675
		Latte	2670405	213920
	Food	Biscotti	863868	70436
		Croissant	2425601	197022
		Scone	912868	72776
	Gifts	Coffee Grinder	603436	48081
		Coffee Pot	613624	47432
		Mug	1188664	93881
		Thermos	571368	45648
*TOTAL REGION West			11652946	932039
TOTAL			23045246	1848714

Generated on: JAN 10, 2012

DROPBLNKLINE 設定を ALL に変更すると、次の出力結果が生成されます。ここでは、中間合計行および総合計行の上下のブランク行だけでなく、見出しの下および脚注の上のブランク行も削除されています。

Gotham Grinds Sales By Region

Region	Category	Product	Dollar Sales	Unit Sales
Northeast	Coffee	Capuccino	542095	44785
		Espresso	850107	68127
		Latte	2771815	222866
	Food	Biscotti	1802005	145242
		Croissant	1670818	137394
		Scone	907171	70732
	Gifts	Coffee Grinder	509200	40977
		Coffee Pot	590780	46185
		Mug	1144211	91497
		Thermos	604098	48870
*TOTAL REGION Northeast			11392300	916675
West	Coffee	Capuccino	895495	71168
		Espresso	907617	71675
		Latte	2670405	213920
	Food	Biscotti	863868	70436
		Croissant	2425601	197022
		Scone	912868	72776
	Gifts	Coffee Grinder	603436	48081
		Coffee Pot	613624	47432
		Mug	1188664	93881
		Thermos	571368	45648
*TOTAL REGION West			11652946	932039
TOTAL			23045246	1848714

Generated on: JAN 10, 2012

レポートへのイメージの追加

スタイルシートを使用することで、レポート内にイメージを追加し、配置することができます。ロゴなどのイメージは、レポートに企業イメージの統一感や視覚的訴求力を与えます。複数の宣言を作成することで、レポートに 1 つ以上のイメージを追加できます。

レポートの背景として、イメージを追加することもできます。背景イメージは、レポートが表示領域全体に表示されるように並べて表示されるか、繰り返し表示されます。レポート全体に添付したイメージ、見出しや脚注のイメージは、背景イメージとともに表示できます。

イメージは、GIF (Graphic Interchange Format) や JPEG (Joint Photographic Experts Group、.jpg 拡張子) など、ブラウザがサポートするファイルフォーマットにする必要があります。

WebFOCUS 標準レポートフォーマットでのイメージサポート

- GIF および JPG イメージは、DHTML、HTML、PDF、PPTX、XLSX、PPT 標準レポートフォーマットでサポートされます。JPEG イメージタイプは、HTML 標準レポートフォーマットでのみサポートされます。それ以外のレポートフォーマットでは、イメージ名の拡張子を .jpeg から .jpg に変更することで、イメージをレポート出力に表示することができます。
- PNG イメージタイプは、DHTML、HTML、PPTX、PDF 標準レポートフォーマットでサポートされ、WebFOCUS で生成される SVG グラフは、HTML、PDF レポートフォーマットに挿入された場合に表示されます。
- SVG イメージは、HTML レポートでのみサポートされます。
- PDF レポートでのイメージの中央揃えおよび右揃えは、JUSTIFY=CENTER または JUSTIFY=RIGHT スタイルシート属性が、イメージ宣言属性で明示的に設定された場合にのみ反映されます。イメージ宣言属性で JUSTIFY が指定されていない場合のデフォルト設定では、イメージは左揃えになります。
- イメージは、EXL2K 標準レポートフォーマットではサポートされません。

複合レポート構文でのイメージサポート

- GIF および JPG イメージは、DHTML、PDF、PPTX、PPT 複合ドキュメント構文でサポートされます。JPEG イメージタイプは、すべてのレポートフォーマットでサポートの対象外ですが、拡張子を .jpeg から .jpg に変更した場合は、これらの複合レポートフォーマットでイメージが正常に機能します。
- PNG イメージは、DHTML、 PPT、PPTX、PDF 複合ドキュメントでサポートされます。
- SVG イメージは、WebFOCUS レポートフォーマットの複合ドキュメントすべてでサ

ポートの対象外ですが、WebFOCUS で生成される SVG グラフは、PDF 複合ドキュメントでのみサポートされます。

- イメージは、EXL2K 複合ドキュメントではサポートされません。

注意：複合ドキュメント内のレポートコンポーネントに挿入されたイメージは、前述の「WebFOCUS 標準レポートフォーマットでのイメージサポート」セクションに記載されているとおりにサポートされます。「複合レポート構文でのイメージサポート」セクションに記載されている説明は、複合ドキュメントの PAGELAYOUT セクションに挿入されたイメージに適用されます。

PDF、HTML、DHTML 出力に使用するデータソースがバイナリラージオブジェクト (BLOB) データタイプをサポートする場合 (Microsoft SQL Server、DB2、Oracle、Informix、BYTEA データタイプを使用する PostgreSQL)、イメージをデータソースの BLOB フィールドに格納しておくことができます。

イメージは、WebFOCUS Reporting Server 上の EDAPATH または APPPATH ディレクトリに存在する必要があります。ファイルが検索パスにない場合、完全なパス名を指定します。

注意：JPEG ファイルの場合、現時点では .jpg 拡張子のみがサポートされます。.jpeg 拡張子はサポートされません。

HTML ドキュメントのイメージをサポートするブラウザおよびデバイス

HTML および DHTML フォーマットの標準レポートと複合ドキュメントでのイメージおよびグラフの表示サポートは、64 ビットイメージを生成済み .htm ファイルにエンコードするイメージ埋め込み機能によって提供されます。

SET HTMLEMBEDIMG は、すべての WebFOCUS レポートをあらゆる種類のブラウザまたはデバイスからアクセスできるようにするコマンドです。このコマンドのデフォルト設定は ON であり、イメージが .htm ファイルに埋め込められます。

```
SET HTMLEMBEDIMG={ON|OFF|AUTO}
```

説明

ON

.htm ファイル内でイメージをエンコードします。デフォルト値は ON で、この設定は HTMLARCHIVE 設定を上書きします。

OFF

イメージを埋め込みません。HTMLARCHIVE が ON に設定されている場合、.mht ファイルが生成されます。

AUTO

リクエストを送信するクライアントマシンのブラウザに基づいて、使用するエンコードアルゴリズムを決定します。特定されたブラウザが Internet Explorer の場合、またはブラウザが不明の場合 (例、ReportCaster から配信されたレポート)、WebFOCUS は従来どおり Web アーカイブファイル (.mht) を生成します。その他のすべてのブラウザでは、WebFOCUS がイメージをエンコードして HTML ファイル (.htm) に埋め込みます。

HTMLEMBEDIMG 使用上の注意

- HTMLEMBEDIMG が ON (デフォルト設定)、かつ HTMLARCHIVE が ON に設定されている場合、HTMLEMBEDIMG ON 設定は HTMLARCHIVE ON 設定を上書きし、イメージが埋め込まれた .htm ファイルが生成されます。
- HTMLEMBEDIMG が ON (デフォルト設定)、かつ HTMLARCHIVE が OFF に設定されている場合、.htm ファイルが生成され、イメージが埋め込まれます。
- HTMLEMBEDIMG を OFF に設定し、HTMLARCHIVE を有効にした場合、エンコードされたイメージが埋め込まれた .mht ファイルが生成されます。
- HTMLEMBEDIMG および HTMLARCHIVE を OFF に設定した場合、.html ファイルは生成されますが、イメージは埋め込まれません。
- Internet Explorer 9 以降を使用する場合、または Internet Explorer 以外のブラウザを使用する場合、任意のサイズのイメージで上記の新しいアルゴリズムがサポートされます。64 ビットエンコードのサポートについての詳細は、ブラウザのベンダー情報を参照してください。

イメージ属性

属性	説明
IMAGE	イメージを追加します。
IMAGEALIGN	イメージを配置します。これは、HTML レポートのみに適用されます。
POSITION	イメージを配置します。
IMAGEBREAK	イメージの後の改行を制御します。これは、内部カスケードスタイルシートを使用しない HTML レポートのみに適用されます。
SIZE	イメージのサイズを設定します。
ALT	米国リハビリテーション法 (U. S. Rehabilitation Act) 第 508 条 (Section 508) アクセシビリティに準拠するイメージの説明を指定します。ALT は HTML レポートのみに適用されます。
PRESERVERATIO	ON に設定すると、SIZE 属性で指定されたサイズにイメージが拡大縮小された場合に縦横比 (高さと幅の比率) が保持されます。これにより、イメージ表示の歪みを防止します。イメージは、SIZE で指定した範囲内で、縦横比の保持が可能な最大のサイズに拡大縮小されます。PDF レポート出力のイメージでサポートされます。
BACKIMAGE	背景イメージを追加します。

HTML レポートへのイメージの追加

この構文は、HTML レポートに適用されます。

```
TYPE={REPORT|heading}, IMAGE={url|(column)} [,IMAGEALIGN=position]
     [,IMAGEBREAK={ON|OFF}] [,ALT='description'], $
```

説明

REPORT

レポート本文にイメージを埋め込みます。REPORT がデフォルト値です。

注意：IMAGE=(column) オプションは TYPE=REPORT ではサポートされません。

heading

見出しまたは脚注にイメージを埋め込みます。有効な値は、TABHEADING、TABFOOTING、HEADING、FOOTING、SUBHEAD、SUBFOOT です。

url

イメージファイルの URL です。イメージは、GIF や JPEG など、ブラウザがサポートするフォーマットの個別ファイルに存在する必要があります。イメージファイルは、ローカルの Web サーバに配置することも、ネットワーク上でアクセス可能な任意のサーバに配置することもできます。詳細は、レポートへのイメージの追加を参照してください。

column

イメージファイルを示す URL が値であるリクエスト内の文字フィールド (たとえば表示フィールドまたは BY フィールド) です。値を WebFOCUS スタイルシートを使用したレポートコンポーネントの選択内で説明する COLUMN 属性を使用して指定します。*column* は括弧で囲みます。

このオプションにより、異なるイメージをフィールド値に応じて見出しまたは脚注に追加することができます。

IMAGEALIGN = position

イメージの位置です。

注意：HTMLCSS=ON に設定されている場合、IMAGEALIGN はサポートされません。HTMLCSS=ON の場合、POSITION 属性を使用して見出しまたは脚注の左上を基準とした相対位置を指定することで、見出しまたは脚注内にイメージを配置することができます。

有効な値には、次のものがあります。

TOP - イメージの右上端が見出しまたは脚注テキストと位置揃えされます。イメージがレポート全体に添付されている場合、レポートの一番上に表示されます。

MIDDLE - イメージは、見出しまたは脚注テキストの中央に表示されます。イメージがレポート全体に添付されている場合、レポートの中央に表示されます。

BOTTOM - イメージの右下端が見出しまたは脚注テキストと位置揃えされます。イメージがレポート全体に添付されている場合、レポートの一番下に表示されます。

LEFT - イメージは、見出しまたは脚注テキストの左側に表示されます。イメージがレポート全体に添付されている場合、レポートの左側に表示されます。

RIGHT - イメージは、見出しまたは脚注テキストの右側に表示されます。イメージがレポート全体に添付されている場合、レポートの右側に表示されます。

IMAGEBREAK

イメージの後の改行を制御します。有効な値には、次のものがあります。

ON - イメージの後に改行を生成します。レポート見出しのテキストなどに続く要素 は、次の行に表示されます。

OFF - イメージの後の改行を行わず、それに続く要素が同じ行になります。デフォルト値は OFF です。

description

セクション 508 (米国リハビリテーション法 508 条) アクセシビリティに準拠する説明テキストを指定します。説明は、一重引用符 (') で囲みます。

URL の指定

次のガイドラインは、IMAGE=*URL* と IMAGE=(*COLUMN*) 構文では同じです。後者は、データソースフィールドに保存した URL に適用されます。

次のように URL を指定します。

- イメージファイルへの絶対または相対アドレスを指定します。以下はその例です。

```
TYPE=TABHEADING,IMAGE=http://www.tibco.com/images/logo_wf3.gif,$
TYPE=TABHEADING, IMAGE=/ibi_apps/ibi_html/ggdemo/gotham.gif,$
```

- SET BASEURL パラメータを使用して、リクエスト内のすべての相対 URL の前に論理的に追加する URL を設定します。この機能を使用すると、IMAGE 属性にイメージのファイル名のみを指定することでイメージを追加することができます。以下はその例

です。

```
SET BASEURL=http://host:port/
.
.
.
TYPE=REPORT, IMAGE=gotham.gif,$
```

以下のように適用します。

- ベース URL は、スラッシュ (/) で終了する必要があります。
- 絶対 URL (http:// で始まる) は、ベース URL を上書きします。
- UNIX サーバを参照する場合は、URL の大文字と小文字が区別されます。
- イメージファイル名に拡張子が含まれていない場合は、.GIF が使用されます。

HTML レポート見出しへの GIF イメージの追加

次のリクエストは、レポートの見出しに Century_Corp ロゴを追加します。このロゴは、IMAGE 属性内の相対 URL により識別される、独立したイメージファイルです。

```
TABLE FILE GGORDER
ON TABLE SUBHEAD
"PRODUCTS ORDERED ON 08/01/96"
SUM QUANTITY AS 'Ordered Units' BY PRODUCT
WHERE PRODUCT EQ 'Coffee Grinder' OR 'Coffee Pot'
WHERE ORDER_DATE EQ '08/01/96'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=TABHEADING, IMAGE=/IBI_APPS/IBI_HTML/GGDEMO/GOTHAM.GIF, IMAGEBREAK=ON, $
ENDSTYLE
END
```

IMAGEBREAK を ON に設定すると、ロゴと見出しテキストの間に改行が挿入されます。

PRODUCTS ORDERED ON 08/01/96

Product	Ordered Units
Coffee Grinder	2493
Coffee Pot	3100

埋め込み JPEG イメージ付きレポート見出しの作成

```
TABLE FILE EMPLOYEE
ON TABLE SUBHEAD
"Employee Salary Information and Courses"
" "
" "
" "
" "
" "
" "
" "
" "
" "
PRINT CURR_SAL BY COURSE_NAME
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=TABHEADING, IMAGE=C:¥IBI¥APPS¥IMAGES¥Pencils.jpg,
POSITION=(.5 .5), SIZE=(.5 .5), $
ENDSTYLE
END
```

注意：このリクエストで使用されているイメージは、WebFOCUS に同梱されているイメージではありません。

出力結果は次のとおりです。

EMPLOYEE SALARY INFORMATION AND SKILLS

COURSE NAME	CURR SAL
ADVANCED TECHNIQUES	$18,480.00
BASIC REPORT PREP DP MGRS	$27,062.00
BASIC REPORT PREP FOR PROG	$13,200.00
	$18,480.00
BASIC REPORT PREP NON-PROG	$21,780.00
BASIC RPT NON-DP MGRS	$9,500.00
DECISION SUPPORT WORKSHOP	$21,780.00
FILE DESC & MAINT NON-PROG	$21,780.00
FILE DESCRPT & MAINT	$11,000.00
	$13,200.00
	$18,480.00
	$16,100.00
FOCUS INTERNALS	$18,480.00
HOST LANGUAGE INTERFACE	$27,062.00
TIMESHARING WORKSHOP	$21,780.00
WHAT'S NEW IN FOCUS	$21,780.00

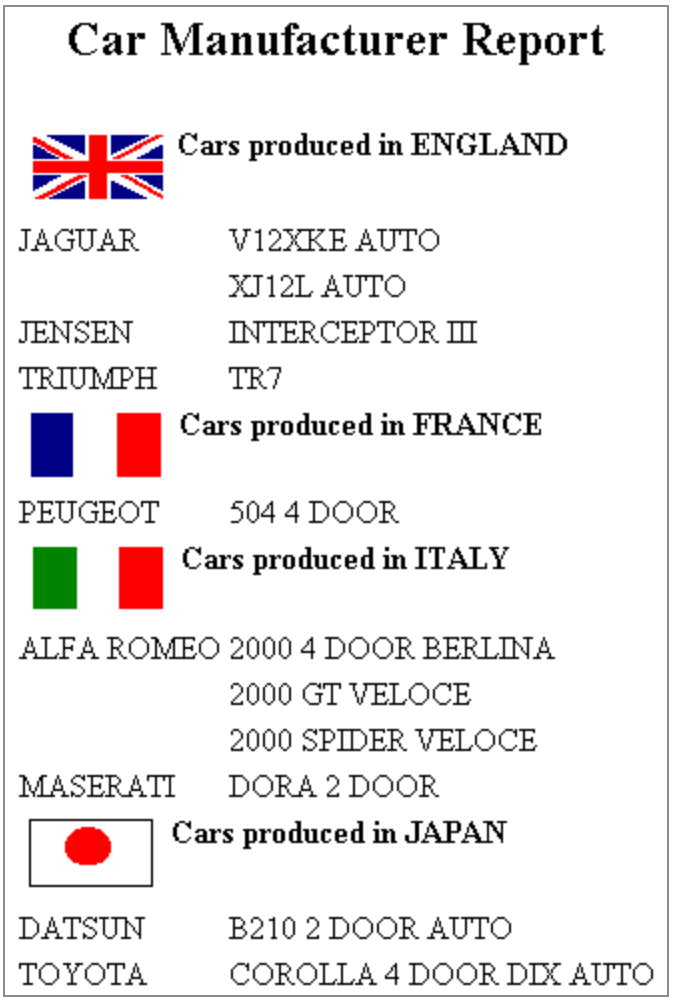
HTML レポートでのデータソースフィールドファイル名の使用

次の例は、SUBHEAD 内にイメージを埋め込む方法を示しています。SUBHEAD が表示される BY フィールドの各値に異なるイメージを使用します。

```
DEFINE FILE CAR
FLAG/A12=
DECODE COUNTRY ( 'ENGLAND' 'uk' 'ITALY' 'italy'
   'FRANCE' 'france' 'JAPAN' 'japan' );
END
```

```
TABLE FILE CAR
PRINT FLAG NOPRINT AND MODEL AS '' BY COUNTRY NOPRINT AS '' BY CAR AS ''
WHERE COUNTRY EQ 'ENGLAND' OR 'FRANCE' OR 'ITALY' OR 'JAPAN'
ON COUNTRY SUBHEAD
"                     <+0>Cars produced in <ST.COUNTRY"
HEADING CENTER
"Car Manufacturer Report"
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=SUBHEAD, IMAGE=(FLAG), $
TYPE=REPORT, GRID=OFF, $
TYPE=HEADING, SIZE=12, STYLE=BOLD, $
TYPE=SUBHEAD, STYLE=BOLD, $
ENDSTYLE
END
```

出力結果は次のとおりです。



ALT 属性を使用したイメージの説明

次のリクエストは、レポートの脚注に企業のロゴを追加します。ここでは、WebFOCUS スタイルシートの ALT 属性を使用して、イメージを特定する説明テキストを追加します。

```
TABLE FILE GGORDER
SUM QUANTITY AS 'Ordered Units'
BY PRODUCT
ON TABLE SUBHEAD
"PRODUCTS ORDERED"
FOOTING
" "
ON TABLE SET ACCESSHTML 508
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
```

```
TYPE=REPORT, GRID=OFF,$
TYPE=FOOTING, IMAGE=/ibi_html/iblogo.gif, ALT='Information Builders
logo',$
ENDSTYLE
END
```

注意：リクエストが WebFOCUS リポジトリに格納され、WebFOCUS 管理コンソールの [アプリケーションの設定] 下の [その他] 設定で [レポート埋め込みイメージのアップロード] を選択した場合、次のいずれかを実行する必要があります。

- WebFOCUS リポジトリに存在しないイメージの参照行 (IMAGE) の先頭に、-MRNOEDIT コマンドを含める。
- IMAGE パラメータに完全修飾パスを含める。

WebFOCUS 管理コンソールの [レポート埋め込みイメージのアップロード] アプリケーション設定は、リポジトリのイメージを Reporting Server にアップロードして、レポートおよび HTML ページに埋め込むかどうかを指定します。デフォルト設定では、リポジトリイメージがアップロードされます。[レポート埋め込みイメージのアップロード] アプリケーション設定についての詳細は、『ibi™ WebFOCUS® セキュリティ管理ガイド』を参照してください。

-MRNOEDIT コマンドは、WebFOCUS Client にコード行の処理を行わないことを指示します。

このリクエストを実行すると、下図のように、イメージがレポートデータの下部に表示されます。

PRODUCTS ORDERED

Product	Ordered Units
Biscotti	136045
Coffee Grinder	58703
Coffee Pot	61982
Croissant	204776
French Roast	285689
Hazelnut	100427
Kona	61498
Mug	117186
Scone	108359
Thermos	61778

ブラウザのイメージローダがオフに設定されている場合、またはブラウザにイメージが表示されない場合は、イメージ上にマウスポインタを置くと、四角い枠内に説明テキストが表示されます。

WebFOCUS によって、このイメージついての次の HTML コードが生成されます。

```
<IMG SRC="/ibi_html/ibilogo.gif"
ALT="Information Builders logo">
```

背景イメージの追加

この構文は、HTML レポートに適用されます。

```
[TYPE=REPORT,] BACKIMAGE=url, $
```

説明

TYPE=REPORT

イメージをレポート全体に適用します。デフォルト値であるため必須ではありません。

url

GIF または JPEG (.jpg) ファイルの URL です。ローカル Webサーバ、またはネットワークからアクセス可能なサーバ上にあるファイルを指定します。

URL は、絶対アドレスまたは相対アドレスです。詳細は、レポートへのイメージの追加を参照してください。

GIF を指定する場合は、ファイルの拡張子を省略できます。

背景イメージの追加

次のリクエストは、レポートに背景イメージを追加します。イメージファイル CALM_BKG.GIF は、指定された相対アドレスに存在します。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY CATEGORY BY PRODUCT
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, STYLE=BOLD, GRID=OFF, $
TYPE=REPORT, BACKIMAGE=/IBI_APPS/IBI_HTML/TEMPLATE/CALM_BKG.GIF, $
ENDSTYLE
END
```

背景はレポート領域全体に並べて表示されます。

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

内部カスケードスタイルシートを使用した PDF および HTML レポートへのイメージの追加

次の構文は、内部カスケードスタイルシートを使用した PDF または HTML レポートに適用されます。別ファイルとして保存されているイメージを使用することができます。

内部カスケードスタイルシートを使用したレポートは、HTML ドキュメント内のスタイルタグ間に HTML カスケードスタイルシート (CSS) が格納された HTML ページです。

```
TYPE={REPORT|heading}, IMAGE={url|file|(column)} [,BY=byfield]
[,POSITION=([+|-]x [+|-]y )] [,SIZE=(w h)] ,$
```

説明

REPORT

レポート本文にイメージを埋め込みます。イメージはレポートの背景に表示されます。REPORT がデフォルト値です。

heading

見出しまたは脚注にイメージを埋め込みます。有効な値は、TABHEADING、TABFOOTING、FOOTING、HEADING、SUBHEAD、および SUBFOOT です。

見出しまたは脚注内に十分な余白をとり、見出しや脚注のテキストと重ならないようにします。また、スタイルシートにスポットマーカーまたは POSITION 属性を使用して、イメージの右側に見出しまたは脚注テキストを配置することもできます。

url

内部カスケードスタイルシートを使用した HTML レポート

イメージファイルの絶対または相対アドレスです。イメージは、GIF や JPEG など、ブラウザがサポートするフォーマットの個別ファイルに存在する必要があります。イメージファイルは、ローカルの Web サーバに配置することも、ネットワーク上でアクセス可能な任意のサーバに配置することもできます。詳細は、レポートへのイメージの追加を参照してください。

file

PDF レポート

イメージファイル名です。このファイルは、EDAPATH または APPPATH で指定された、WebFOCUS Reporting Server 上のディレクトリに存在する必要があります。ファイルが検索パスにない場合、完全なパス名を指定します。

GIF を指定する場合は、ファイルの拡張子を省略できます。

column

イメージファイル名を含むデータソース内の文字フィールドです。WebFOCUS スタイルシートを使用したレポートコンポーネントの選択で説明した COLUMN 属性を使用します。*column* は括弧で囲みます。

フィールドにファイル名またはイメージが含まれている場合、そのフィールドはリクエスト内で参照されている表示フィールドまたは BY フィールドにする必要があります。

フィールドの値は、スタイルシート内のイメージの URL として入力されたときと同様に解釈されます。接尾語を省略した場合、デフォルト設定で「.GIF」 が指定されます。SET BASEURL は、イメージのベース URL を指定する際に役立ちます。ベース URL を指定した場合、フィールドの値に完全な URL を含める必要はありません。

たとえば、SUBHEAD にイメージを埋め込み、BY フィールドの値が変わるたびに SUBHEAD に異なるイメージを表示する場合にこの構文が役立ちます。

byfield

中間見出しまたは中間脚注を生成したソートフィールドです。

POSITION

イメージの開始位置です。

+|-

イメージを埋め込むレポートコンポーネントの、左上の隅から測定した横方向または縦方向の距離を指定します。

x

UNITS パラメータで指定された測定単位で、物理的なレポートページの左上隅から測定したイメージの横方向の開始位置を表します。

x および *y* の値は括弧で囲みます。その間にカンマ (,) は挿入しません。

y

UNITS パラメータで指定された測定単位で、物理的なレポートページの左上隅から測定したイメージの縦方向の開始位置を表します。

SIZE

イメージのサイズです。デフォルト設定では、イメージは元のサイズで追加されます。

w

UNITS パラメータで指定された測定単位でイメージの幅を表します。

w および *h* の値は括弧で囲みます。その間にカンマ (,) は挿入しません。

h

UNITS パラメータで指定された測定単位でイメージの高さを表します。

GIF イメージの内部カスケードスタイルシート付き HTML レポートへの追加

URL は、「WEBSRVR1」という名前のサーバ上のイメージファイル GOTHAM.GIF を特定します。TYPE 属性は、レポートの見出しにそのイメージを追加します。POSITION は、レポートページの左上の隅から横方向に 1/4 インチ、縦方向に 1/10 インチの位置にイメージを配置します。イメージは、SIZE 属性で幅が 1 インチ、高さが 1 インチに指定されています。

```
SET HTMLCSS = ON
TABLE FILE GGSALES
SUM UNITS BY PRODUCT
ON TABLE SUBHEAD
"REPORT ON UNITS SOLD"
" "
" "
" "
" "
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=TABHEADING, IMAGE=HTTP://WEBSRVR1/IBI_APPS/IBI_HTML/GGDEMO/GOTHAM.GIF,
     POSITION=(.25 .10), SIZE=(1 1), $
ENDSTYLE
END
```

会社のロゴが、指定された位置およびサイズでレポートの見出しに配置されます。

REPORT ON UNITS SOLD

Product	Unit Sales
Biscotti	421377
Capuccino	189217
Coffee Grinder	186534
Coffee Pot	190695
Croissant	630054
Espresso	308986
Latte	878063
Mug	360570
Scone	333414
Thermos	190081

PDF レポートへの GIF イメージの追加

次の例のイメージファイルは、GOTHAM.GIF です。POSITION 属性は、レポートページの左上の隅から横方向に 1/4 インチ、縦方向に 1/4 インチの位置にイメージを配置します。イメージは、SIZE 属性で幅が 1/2 インチ、高さが 1/2 インチに指定されています。

```
SET ONLINE-FMT = PDF
TABLE FILE GGSALES
SUM UNITS BY PRODUCT
ON TABLE SUBHEAD
"Report on Units Sold"
" "
" "
" "
" "
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=TABHEADING, IMAGE=GOTHAM.GIF, POSITION=(.25 .25), SIZE=(.5 .5), $
ENDSTYLE
END
```

レポートは次のとおりです。

Report on Units Sold

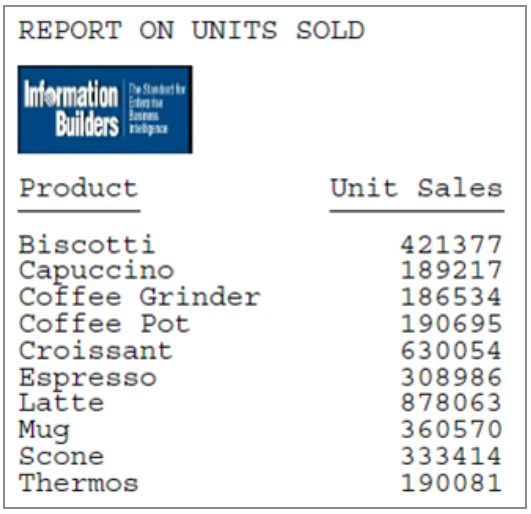
Product	Unit Sales
Biscotti	421377
Capuccino	189217
Coffee Grinder	186534
Coffee Pot	190695
Croissant	630054
Espresso	308986
Latte	878063
Mug	360570
Scone	333414
Thermos	190081

PDF レポートへの PNG イメージの追加

このサンプルのイメージファイルは Ibi_logo.png です。POSITION 属性で、イメージの配置先をレポートページの左上に設定します。イメージは、SIZE 属性で幅が 1 インチ、高さが 0.5 インチに指定されています。

```
SET HTMLCSS = ON
TABLE FILE GGSALES
SUM UNITS BY PRODUCT
ON TABLE SUBHEAD
"REPORT ON UNITS SOLD"
" "
" "
" "
" "
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=TABHEADING, IMAGE=Ibi_logo.png, POSITION=(0 .30), SIZE=(1 0.5), $
ENDSTYLE
END
```

レポートは次のとおりです。


REPORT ON UNITS SOLD

Product	Unit Sales
Biscotti	421377
Capuccino	189217
Coffee Grinder	186534
Coffee Pot	190695
Croissant	630054
Espresso	308986
Latte	878063
Mug	360570
Scone	333414
Thermos	190081

BLOB フィールドに格納されたイメージの PDF、DHTML、HTML レポートへの追加

PDF、HTML、DHTML 出力に使用するデータソースがバイナリラージオブジェクト (BLOB) データタイプをサポートする場合 (Microsoft SQL Server、DB2、Oracle、Informix、BYTEA データタイプを使用する PostgreSQL)、イメージをデータソースの BLOB フィールドに格納しておくことができます。

PDF、HTML、DHTML フォーマットのレポート出力生成に使用する WebFOCUS スタイルシートは、BLOB フィールドインスタンスに GIF イメージまたは JPG イメージの正確なバイナリコピーが格納されている場合、イメージソースとして BLOB フィールドにアクセスすることができます。HTML および DHTML レポートでは、PNG イメージもサポートされます。同一 BLOB フィールドに、異なるフォーマット (GIF、JPG、PNG) のイメージを混在させることができます。フォーマットは、イメージのヘッダに基づいて、WebFOCUS によって特定されます。イメージは、レポートの列、見出し、脚注、中間見出し、中間脚注に挿入することができます。

その BLOB フィールドは、リクエストの PRINT コマンドまたは LIST コマンドで参照する必要があります。集計はサポートされていません。BLOB イメージが格納されたレポートは、複合レイアウトレポートのコンポーネントとしてサポートされます。

次の SET コマンドを使用する場合、BLOB イメージがすべてのブラウザの HTML と DHTML の両方で機能します。

- SET HTMLEMBEDIMG=AUTO

- SET HTMLARCHIVE=ON (Internet Explorer で 32 キロバイトを超えるイメージをサポートするために必要)
- SET BASEURL='' (WebFOCUS Client から送信されたデフォルト設定が上書きされるため、埋め込みイメージを正しく機能させるために必要)
- SET HTMLCSS=ON (HTML レポートの中間見出し内のイメージの配置に必要)

  HTMLCSS=ON に設定すると、内部カスケードスタイルシートを使用した HTML レポートが作成されます。内部カスケードスタイルシートを使用したレポートは、HTML ドキュメント内のスタイルタグ間に HTML カスケードスタイルシート (CSS) が格納された HTML ページです。

```
TYPE={REPORT|heading}, IMAGE={url|file|(column)} [,BY=byfield]
[,POSITION=([+|-]x [+|-]y )] [,SIZE=(w h)] [,PRESERVERATIO={ON|OFF}],$
```

```
TYPE=DATA, COLUMN=imagefield, IMAGE=(imagefield), SIZE=(wh)
[,PRESERVERATIO={ON|OFF}] ,$
```

説明

REPORT

レポート本文にイメージを埋め込みます。イメージはレポートの背景に表示されます。デフォルト値は REPORT です。ただし、PDF 出力でサポートされる BLOB フィールドにイメージが格納されている場合は使用できません。

heading

見出しまたは脚注にイメージを埋め込みます。有効な値は、FOOTING、HEADING、SUBHEAD、および SUBFOOT です。

イメージの埋め込み先が、列ではなく、見出し、中間見出し、脚注、中間脚注の場合は、スタイルシート宣言を使用してイメージを見出し、中間見出し、脚注、中間脚注に配置する必要があります。スタイルシートから BLOB にアクセス可能にするには、NOPRINT オプション付きの PRINT コマンドまたは LIST コマンドで、BLOB フィールドを参照する必要があります。見出しまたは脚注内で BLOB フィールド名を直接参照することはできません。

見出しまたは脚注内に十分な余白をとり、見出しや脚注のテキストと重ならないようにします。また、スタイルシートにスポットマーカーまたは POSITION 属性を使用して、イメージの右側に見出しまたは脚注テキストを配置することもできます。

file

イメージファイル名です。このファイルは、EDAPATH または APPPATH で指定された、WebFOCUS Reporting Server 上のディレクトリに存在する必要があります。ファイルが検索パスにない場合、完全なパス名を指定します。

GIF を指定する場合は、ファイルの拡張子を省略できます。

column

GIF または JPG イメージのバイナリコピーが格納された、データソース内の BLOB フィールドです。HTML および DHTML フォーマットでは、PNG フォーマットのイメージもサポートされます。同一 BLOB フィールドに、異なるフォーマット (GIF、JPG、PNG) のイメージを混在させることができます。フォーマットは、イメージのヘッダに基づいて、WebFOCUS によって特定されます。イメージは、レポートの列、見出し、脚注、中間見出し、中間脚注に挿入することができます。WebFOCUS スタイルシートを使用したレポートコンポーネントの選択で説明した COLUMN 属性を使用します。*column* は括弧で囲みます。

フィールドにファイル名またはイメージが含まれている場合、そのフィールドはリクエスト内で参照されている表示フィールドまたは BY フィールドにする必要があります。

byfield

中間見出しまたは中間脚注を生成したソートフィールドです。

imagefield

イメージが格納された BLOB フィールドの、任意の有効な参照フィールドです。リクエストで、BLOB フィールドは、PRINT コマンドまたは LIST コマンドで参照される必要があります。

この指定を省略した場合、デフォルトサイズは縦横 1 インチになります。列幅および行間隔は、イメージに合わせて自動的に調整されます。

POSITION

イメージの開始位置です。

+|-

イメージを埋め込むレポートコンポーネントの、左上の隅から測定した横方向または縦方向の距離を指定します。

x

UNITS パラメータで指定された測定単位で、物理的なレポートページの左上隅から測定したイメージの横方向の開始位置を表します。

x および *y* の値は括弧で囲みます。その間にカンマ (,) は挿入しません。

y

UNITS パラメータで指定された測定単位で、物理的なレポートページの左上隅から測定したイメージの縦方向の開始位置を表します。

SIZE

イメージのサイズです。デフォルト設定では、イメージは元のサイズで追加されます。BLOB フィールドに格納されているイメージは、PDF、HTML、DHTML 出力でのみサポートされます。

w

UNITS パラメータで指定された測定単位でイメージの幅を表します。

w および *h* の値は括弧で囲みます。その間にカンマ (,) は挿入しません。

h

UNITS パラメータで指定された測定単位でイメージの高さを表します。

SIZE を省略した場合、イメージの元のサイズが使用されます。すべての GIF および JPG イメージには、ビットマップのサイズに基づく、拡大縮小されていない元のサイズがあります。

[PRESERVERATIO={ON|OFF}]

PNG フォーマットのイメージではサポートされません。PRESERVERATIO=ON に設定すると、SIZE 属性で指定されたサイズにイメージが拡大縮小された場合に、縦横比 (高さと幅の比率) が保持されます。これにより、イメージ表示の歪みを防止します。イメージは、SIZE で指定した範囲内で、縦横比の保持が可能な最大のサイズに拡大縮小されます。PDF 出力でサポートされます。OFF を指定すると、縦横比は保持されません。デフォルト値は OFF です。

BLOB フィールドに格納されているイメージの実際のサイズはイメージごとに異なるため、イメージを指定サイズに拡大縮小することで、表形式レポートに適合させることができます。

注意：BLOB フィールドに格納されているイメージは、PDF、HTML、DHTML 出力でのみサポートされます。

BLOB フィールドのイメージのレポートフィールドへの挿入

Microsoft SQL Server データソースの RETAILDETAIL には、スポーツウェアとシューズの小売店が販売する製品情報が格納されています。また、Microsoft SQL Server データソースの RETAILIMAGE には、RETAILDETAIL データソースと同一の製品 ID フィールドと、各製品のイメージが格納された BLOB データタイプの PRODIMAGE フィールドが含まれています。

次のマスターファイルは、「retaildetail」という名前の Microsoft SQL Server データソースの記述です。これにはスポーツウェアおよびスポーツシューズの小売製品情報が格納されています。

```
FILENAME=RETAILDETAIL, SUFFIX=SQLMSS  , $
  SEGMENT=SEG01, SEGTYPE=S0, $
    FIELDNAME=FOCLIST, ALIAS=FOCLIST, USAGE=I5, ACTUAL=I4, $
    FIELDNAME=PRODUCTID, ALIAS=ProductId, USAGE=A5, ACTUAL=A5,
      MISSING=ON, $
    FIELDNAME=DEPARTMENT, ALIAS=Department, USAGE=A10, ACTUAL=A10,
      MISSING=ON, $
    FIELDNAME=CATEGORY, ALIAS=Category, USAGE=A30, ACTUAL=A30,
      MISSING=ON, $
    FIELDNAME=SPORTS, ALIAS=Sports, USAGE=A30, ACTUAL=A30,
      MISSING=ON, $
    FIELDNAME=GENDER, ALIAS=Gender, USAGE=A10, ACTUAL=A10,
      MISSING=ON, $
    FIELDNAME=BRAND, ALIAS=Brand, USAGE=A25, ACTUAL=A25,
      MISSING=ON, $
    FIELDNAME=STYLE, ALIAS=Style, USAGE=A25, ACTUAL=A25,
      MISSING=ON, $
    FIELDNAME=COLOR, ALIAS=Color, USAGE=A25, ACTUAL=A25,
      MISSING=ON, $
    FIELDNAME=NAME, ALIAS=Name, USAGE=A80, ACTUAL=A80,
      MISSING=ON, $
    FIELDNAME=DESCRIPTION, ALIAS=Description, USAGE=A1000, ACTUAL=A1000,
      MISSING=ON, $
    FIELDNAME=PRICE, ALIAS=Price, USAGE=D7.2, ACTUAL=D8,
      MISSING=ON, $
```

次のマスターファイルは、「retailimage」という名前の Microsoft SQL Server データソースの記述です。retaildetail と同一の product ID フィールドがあり、各製品のイメージが「prodimage」というフィールドに格納されています。このフィールドのデータタイプは BLOB です。

```
FILENAME=RETAILIMAGE, SUFFIX=SQLMSS   , $
  SEGMENT=RETAILIMAGE, SEGTYPE=S0, $
    FIELDNAME=PRODUCTID, ALIAS=PRODUCTID, USAGE=A5, ACTUAL=A5, $
    FIELDNAME=PRODIMAGE, ALIAS=F02BLOB50000, USAGE=BLOB, ACTUAL=BLOB,
      MISSING=ON, $
```

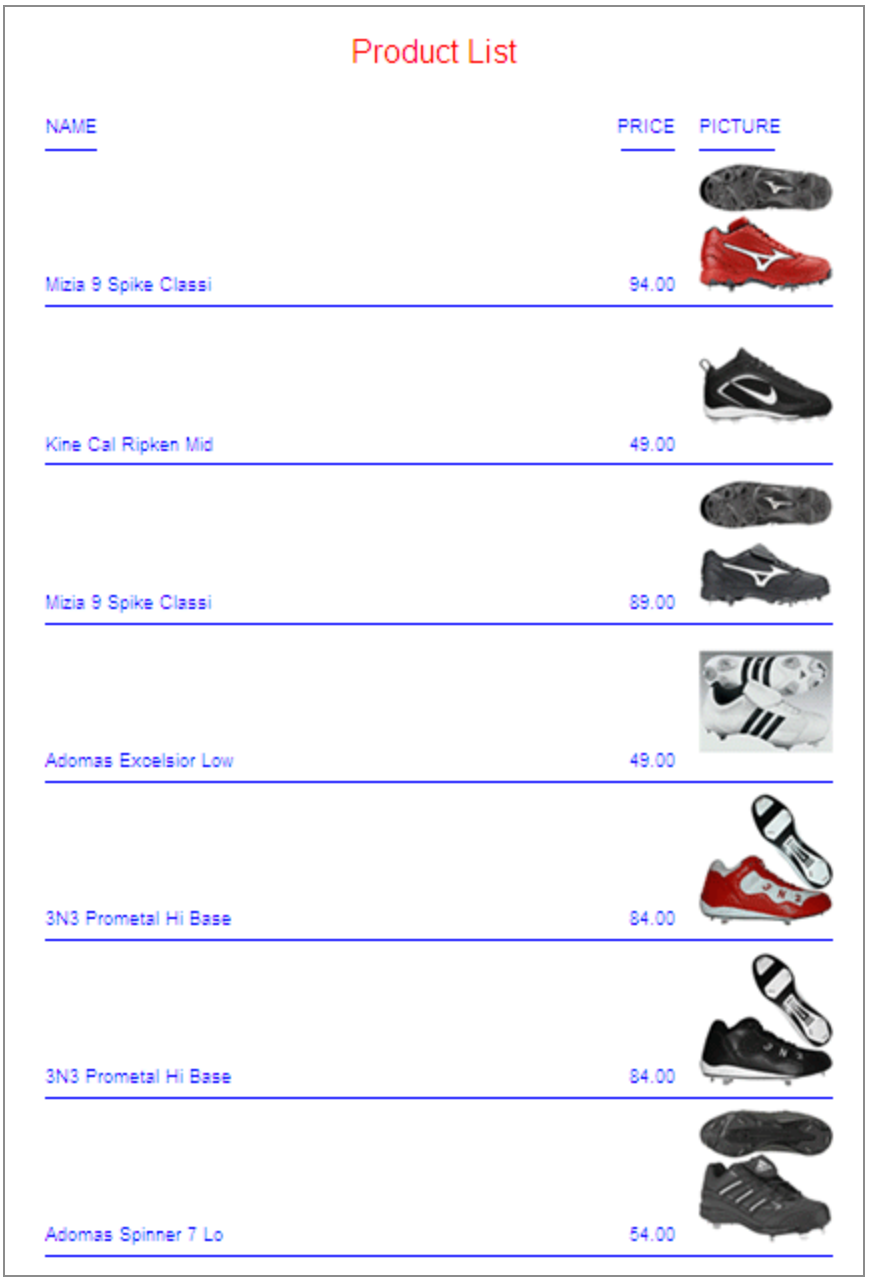
次のリクエストは、2 つのデータソースを結合し、製品名と価格を対応するイメージとともに表示します。出力は、DHTML フォーマットで生成されます。

```
-* Rel 7705 DHTML and HTML supports including Image stored in
-* BLOB field in report column, heading, footing, subhead, or
-* subfoot
-* Rel 769 supports PDF format
JOIN PRODUCTID IN RETAILDETAIL TO PRODUCTID IN RETAILIMAGE
TABLE FILE RETAILDETAIL
HEADING CENTER
"Product List"
" "
PRINT NAME/A20 PRICE PRODIMAGE AS 'PICTURE'
BY PRODUCTID NOPRINT
BY NAME NOPRINT
ON NAME UNDER-LINE
ON TABLE SET PAGE NOPAGE
-*************************
-* Lines between asterisk lines required for BLOB image support
-* for HTML and DHTML formats.
ON TABLE SET HTMLEMBEDIMG AUTO
-* Required to support IE8 with images larger than 32K
ON TABLE SET HTMLARCHIVE ON
-*Required for image positioning in subheads in HTML reports
ON TABLE SET HTMLCSS ON
-*************************
ON TABLE PCHOLD FORMAT DHTML
ON TABLE SET STYLE *
TYPE=REPORT,COLOR=BLUE,FONT=ARIAL, GRID=OFF,$
TYPE=HEADING, SIZE = 18, COLOR=RED,$
TYPE=DATA,COLUMN=PRODIMAGE,IMAGE=(PRODIMAGE),SIZE=(1 1),$
ENDSTYLE
END
```

イメージは、次のスタイルシート属性を使用して、レポート列に配置されます。このスタイルシートはイメージフィールド名を指定して、イメージ列のサイズと位置を確定します。

```
TYPE=DATA,COLUMN=PRODIMAGE,IMAGE=(PRODIMAGE),SIZE=(1 1),$
```

下図は、出力の一部を示しています。DHTML フォーマットでは、指定した間隔が保持されています。

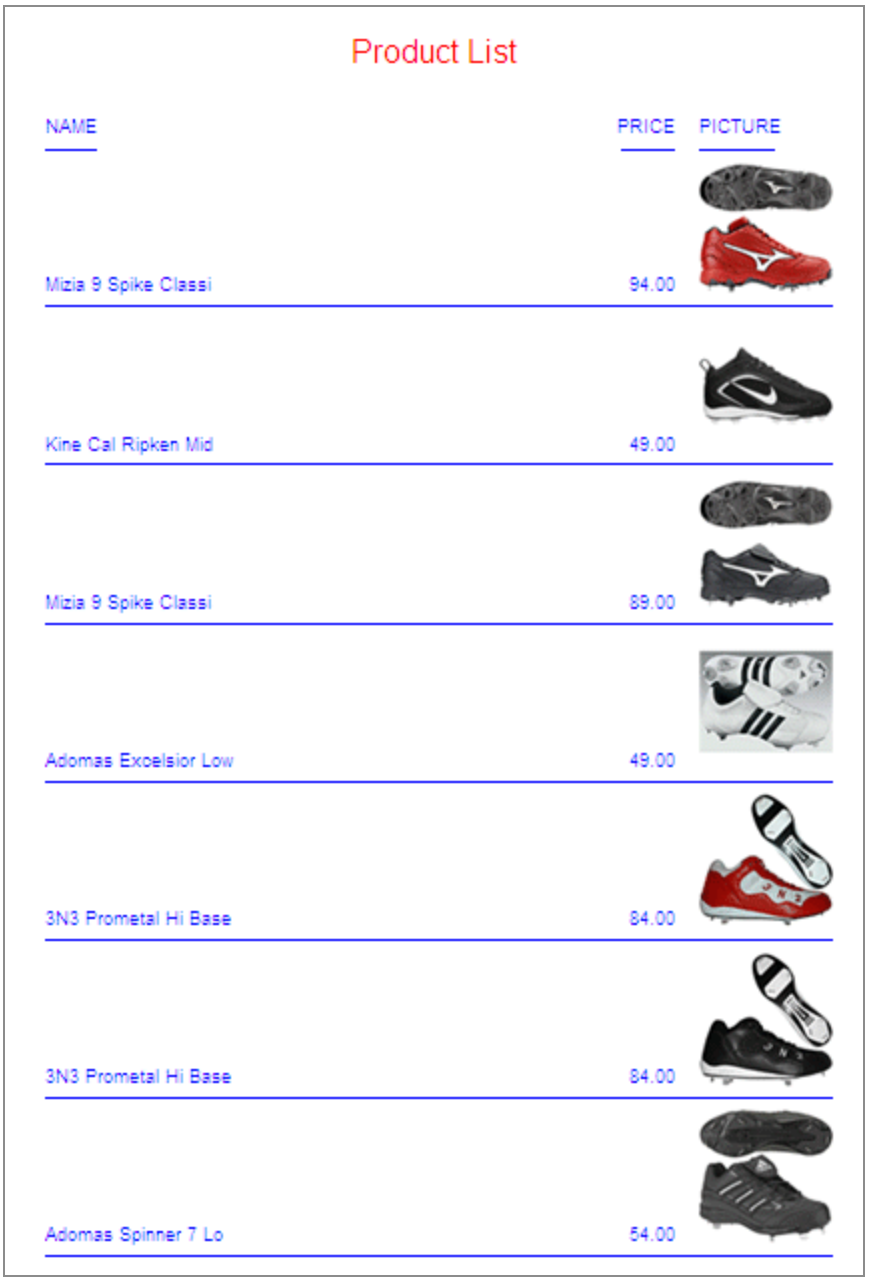


次のリクエストは、出力を HTML フォーマットで生成します。

```
-* Rel 7705 DHTML and HTML supports including Image stored in
-* BLOB field in report column, heading, footing, subhead, or
-* subfoot
-* Rel 769 supports PDF format
```

```
JOIN PRODUCTID IN RETAILDETAIL TO PRODUCTID IN RETAILIMAGE
TABLE FILE RETAILDETAIL
HEADING CENTER
"Product List"
" "
PRINT NAME/A20 PRICE PRODIMAGE AS 'PICTURE'
BY PRODUCTID NOPRINT
BY NAME NOPRINT
ON NAME UNDER-LINE
ON TABLE SET PAGE NOPAGE
-*************************
-* Lines between asterisk lines required for BLOB image support
-* for HTML and DHTML formats.
ON TABLE SET HTMLEMBEDIMG AUTO
-* Required to support IE8 with images larger than 32K
ON TABLE SET HTMLARCHIVE ON
-*Required for image positioning in subheads in HTML reports
ON TABLE SET HTMLCSS ON
-*************************
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET STYLE *
TYPE=REPORT,COLOR=BLUE, GRID=OFF, FONT=ARIAL,$
TYPE=HEADING, SIZE = 18, COLOR=RED,$
TYPE=DATA,COLUMN=PRODIMAGE,IMAGE=(PRODIMAGE),SIZE=(1 1),$
ENDSTYLE
END
```

下図は、出力の一部を示しています。HTML レポート出力では、ブラウザによりブランク領域が除外されるため、間隔が異なります。

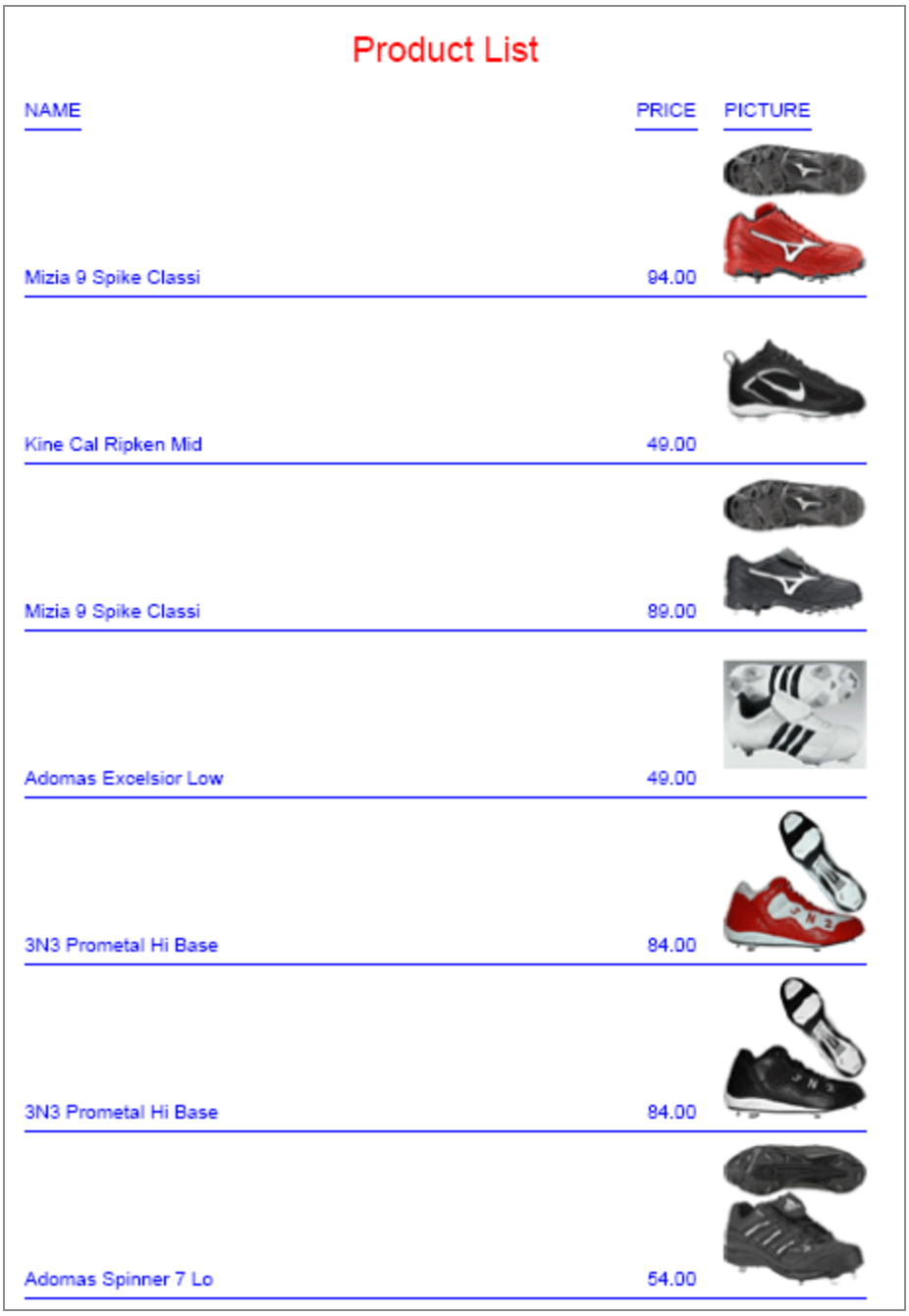


次のリクエストは、レポートを PDF フォーマットで生成します。

```
-* Rel 7705 DHTML and HTML supports including Image stored in
-* BLOB field in report column, heading, footing, subhead, or
-* subfoot
-* Rel 769 supports PDF format
JOIN PRODUCTID IN RETAILDETAIL TO PRODUCTID IN RETAILIMAGE
TABLE FILE RETAILDETAIL
HEADING CENTER
```

```
"Product List"
" "
PRINT NAME/A20 PRICE PRODIMAGE AS 'PICTURE'
BY PRODUCTID NOPRINT
BY NAME NOPRINT
ON NAME UNDER-LINE
ON TABLE SET PAGE NOPAGE
-*************************
-* Lines between asterisk lines required for BLOB image support
-* for HTML and DHTML formats.
ON TABLE SET HTMLEMBEDIMG AUTO
-* Required to support IE8 with images larger than 32K
ON TABLE SET HTMLARCHIVE ON
-*Required for image positioning in subheads in HTML reports
ON TABLE SET HTMLCSS ON
-*************************
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT,COLOR=BLUE, GRID=OFF,$
TYPE=HEADING, SIZE = 18, FONT = ARIAL, COLOR=RED,$
TYPE=DATA,COLUMN=PRODIMAGE,IMAGE=(PRODIMAGE),SIZE=(1 1),$
ENDSTYLE
END
```

下図は、出力の一部を示しています。PDF 出力では、指定した間隔が保持されるため、DHTML 出力と同様の結果になります。



BLOB フィールドのイメージの中間見出しへの挿入

Microsoft SQL Server データソースの RETAILDETAIL には、スポーツウェアとシューズの小売店が販売する製品情報が格納されています。また、Microsoft SQL Server データソースの

RETAILIMAGE には、RETAILDETAIL データソースと同一の製品 ID フィールドと、各製品のイメージが格納された BLOB データタイプの PRODIMAGE フィールドが含まれています。

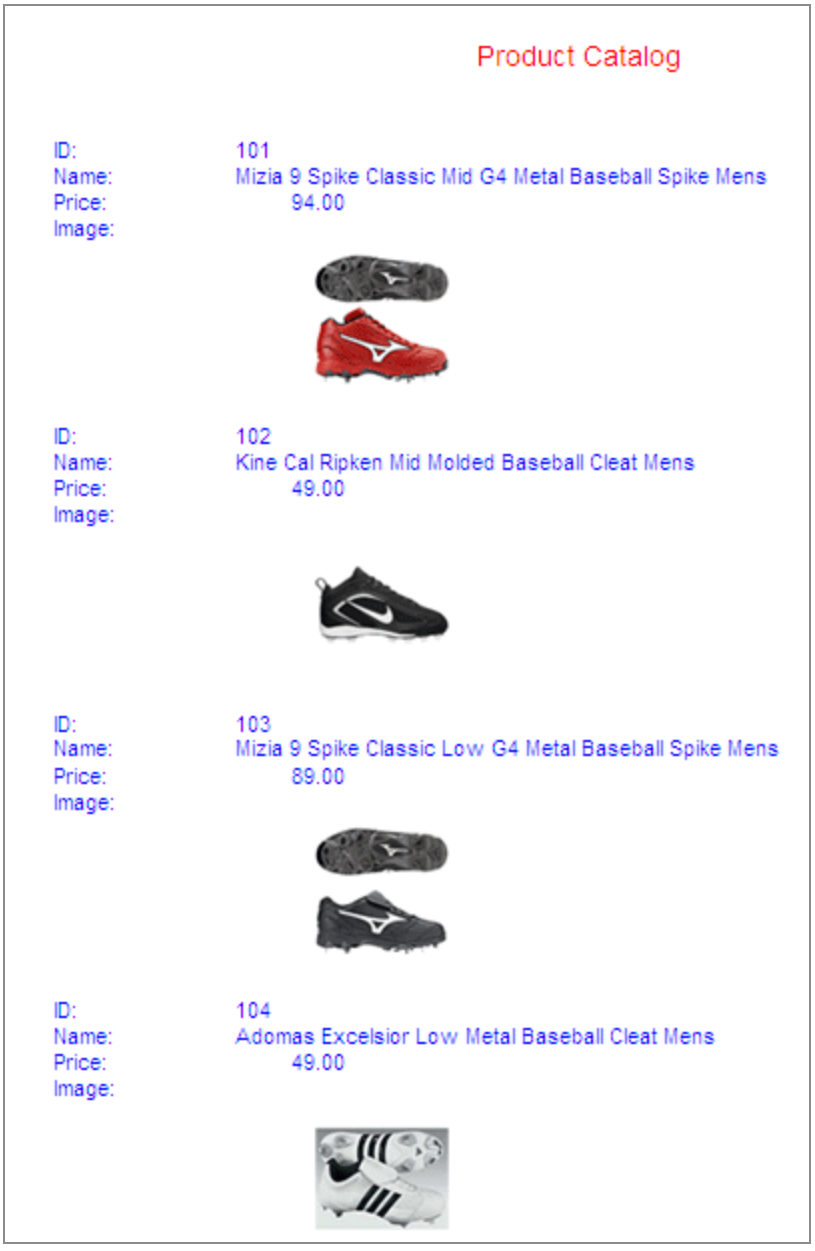
次のリクエストは、2 つのデータソースを結合し、製品イメージを中間見出しに表示します。出力は、DHTML フォーマットで生成されます。出力を HTML または PDF フォーマットで生成することもできます。

```
-* Rel 7705 DHTML and HTML supports including Image stored in
-* BLOB field in HTML report column, heading, footing, subhead,
-* or subfoot
-* Rel 769 supports PDF format
-*SET BASEURL='' - Required for embedded images to work.Overrides default
-* setting from WF Client.
SET BASEURL=''
JOIN PRODUCTID IN RETAILDETAIL TO PRODUCTID IN RETAILIMAGE
TABLE FILE RETAILDETAIL
HEADING CENTER
"Product Catalog"
" "
PRINT NAME NOPRINT PRODIMAGE NOPRINT
BY PRODUCTID NOPRINT
ON PRODUCTID SUBHEAD
""
" ID: <10<PRODUCTID "
" Name: <10<NAME "
" Price: <7<PRICE "
" Image: "
""
""
""
""
""
ON TABLE SET PAGE NOPAGE
```

```
-*************************
-* Lines between asterisk lines required for BLOB image support
-* for HTML and DHTML formats.
ON TABLE SET HTMLEMBEDIMG AUTO
-* Required to support IE8 with images larger than 32K
ON TABLE SET HTMLARCHIVE ON
-*Required for image positioning in subheads in HTML reports
ON TABLE SET HTMLCSS ON
-*************************
ON TABLE PCHOLD FORMAT DHTML
ON TABLE SET STYLE *
TYPE=REPORT,COLOR=BLUE,FONT = ARIAL,$
```

```
TYPE=HEADING, COLOR = RED, SIZE = 16, JUSTIFY=CENTER,$
TYPE=SUBHEAD,BY=PRODUCTID,IMAGE=(PRODIMAGE),SIZE=(1 1), POSITION=(+2 +1),$
ENDSTYLE
END
```

以下は、出力結果の一部を示しています。



BLOB フィールドのイメージのサイズ変更

Microsoft SQL Server データソースの RETAILDETAIL には、スポーツウェアとシューズの小売店が販売する製品情報が格納されています。また、Microsoft SQL Server データソースの RETAILIMAGE には、RETAILDETAIL データソースと同一の製品 ID フィールドと、各製品のイメージが格納された BLOB データタイプの PRODIMAGE フィールドが含まれています。

次のリクエストは、2 つのデータソースを結合し、同一のイメージを、それぞれ異なるサイズおよび異なる PRESERVERATIO 設定で、出力の 3 つの列に表示します。PNG フォーマットのイメージでは、PRESERVERATIO=ON はサポートされません。

出力は、DHTML フォーマットで生成されます。出力を HTML または PDF フォーマットで生成することもできます。

```
-* Rel 7705 DHTML and HTML supports including Image stored in
-* BLOB field in report column, heading, footing, subhead, or
-* subfoot
-* Rel 769 supports PDF format
JOIN PRODUCTID IN RETAILDETAIL TO PRODUCTID IN RETAILIMAGE
TABLE FILE RETAILDETAIL
PRINT PRODIMAGE AS '' PRODIMAGE AS '' PRODIMAGE AS ''
BY STYLE NOPRINT
WHERE NAME CONTAINS 'Pant' OR 'Tank'
ON STYLE UNDER-LINE
ON TABLE SET PAGE NOPAGE
-*************************
-* Lines between asterisk lines required for BLOB image support
-* for HTML and DHTML formats.
ON TABLE SET HTMLEMBEDIMG AUTO
-* Required to support IE8 with images larger than 32K
ON TABLE SET HTMLARCHIVE ON
-*Required for image positioning in subheads in HTML reports
ON TABLE SET HTMLCSS ON
-*************************
ON TABLE PCHOLD FORMAT DHTML
ON TABLE SET STYLE *
TYPE=REPORT,COLOR=BLUE,FONT = ARIAL,$
TYPE=DATA,COLUMN=P1,IMAGE=(PRODIMAGE),SIZE=(.75 .75),$
TYPE=DATA,COLUMN=P2,IMAGE=(PRODIMAGE),SIZE=(.75 1),PRESERVERATIO=ON,$
TYPE=DATA,COLUMN=P3,IMAGE=(PRODIMAGE),SIZE=(.75 1),PRESERVERATIO=OFF,$
ENDSTYLE
END
```

2 列目の SIZE 値 (高さ) には 1 列目のイメージスタイルとは異なる値が指定されていますが、2 列目には PRESERVERATIO=ON が指定されているため、この列のイメージの縦横比は保持されます。また、3 列目には PRESERVERATIO=OFF が指定されているため、この列ではイメージの縦横比が保持されず、リクエストの SIZE 値 (高さと幅) で指定されたサイズでイメージが表示されます。

以下は、出力の一部を示しています。

概要レポートでの BLOB フィールドのイメージの挿入

概要データを表示するレポートに、BLOB フィールドのイメージを挿入するには、リクエストに 2 つの表示コマンドを含める必要があります。概要情報の場合は SUM コマンド、イメージおよびそれ以外の詳細データを表示する場合は PRINT コマンドまたは LIST コマンドを使用します。

Microsoft SQL Server データソースの RETAILDETAIL には、スポーツウェアとシューズの小売店が販売する製品情報が格納されています。また、Microsoft SQL Server データソースの RETAILIMAGE には、RETAILDETAIL データソースと同一の製品 ID フィールドと、各製品のイメージが格納された BLOB データタイプの PRODIMAGE フィールドが含まれています。

次のリクエストは、2 つのデータソースを結合します。このリクエストには、2 つの表示コマンド (SUM コマンドおよび PRINT コマンド) が含まれています。SUM コマンドは、各カテゴリの総額を集計し、そのカテゴリ名および総額を中間見出しに表示します。PRINT コマンドは、カテゴリに属する各製品のイメージ、製品番号、価格を中間脚注に表示します。

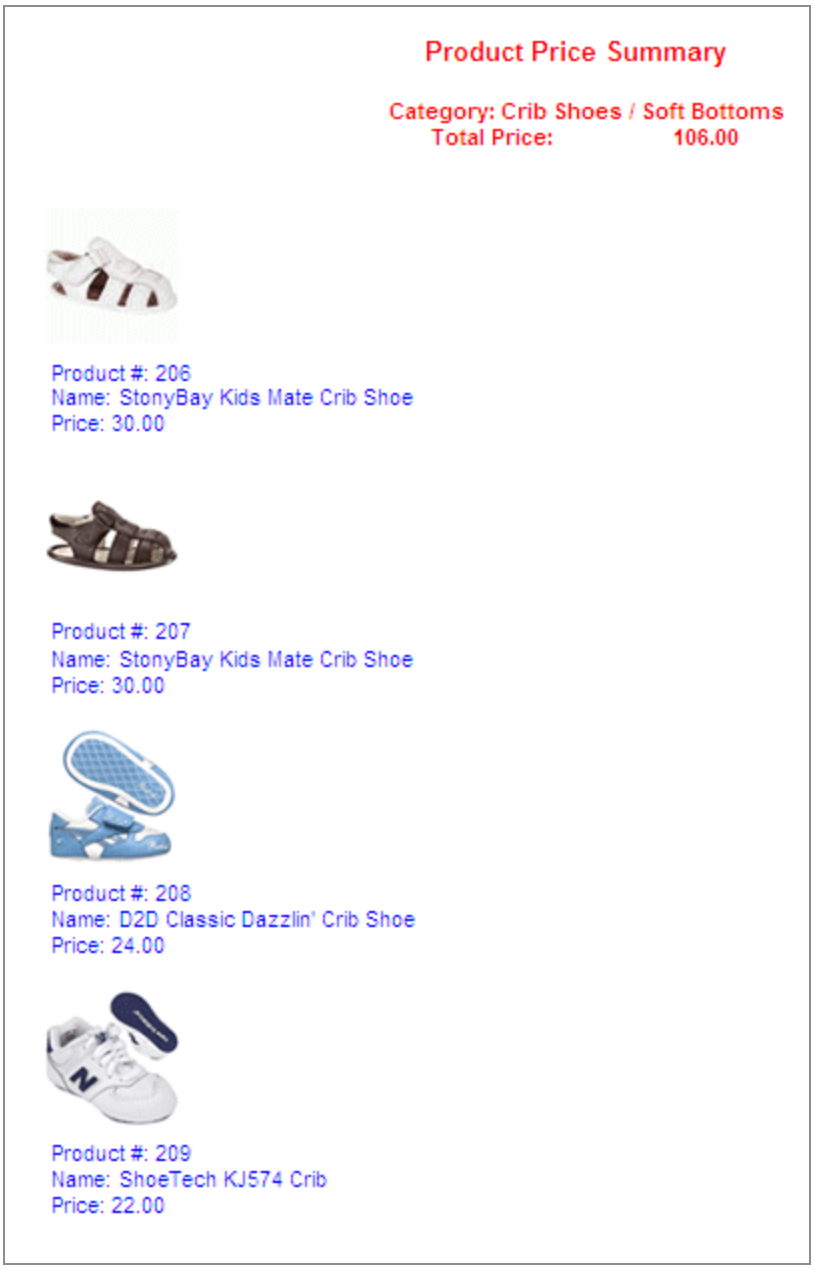
出力は、DHTML フォーマットで生成されます。出力を HTML または PDF フォーマットで生成することもできます。

```
-* Rel 7705 DHTML and HTML supports including images stored in
-* BLOB field in report column, heading, footing, subhead, or
-* subfoot
-* Rel 769 supports PDF format
SET PRINTPLUS=ON
JOIN PRODUCTID IN RETAILDETAIL TO PRODUCTID IN RETAILIMAGE
TABLE FILE RETAILDETAIL
HEADING CENTER
"Product Price Summary"
" "
SUM PRICE NOPRINT
BY CATEGORY NOPRINT
ON CATEGORY SUBHEAD
" Category: <CATEGORY "
" Total Price: <PRICE "
" "
```

```
PRINT PRICE NOPRINT PRODIMAGE NOPRINT
BY CATEGORY NOPRINT
BY PRODUCTID NOPRINT
```

```
ON PRODUCTID SUBFOOT
" "
" "
" "
" "
" "
" "
" Product #: <PRODUCTID "
" Name: <NAME "
" Price: <FST.PRICE "
ON TABLE SET PAGE NOPAGE
-*************************
-* Lines between asterisk lines required for BLOB image support
-* for HTML and DHTML formats.
ON TABLE SET HTMLEMBEDIMG AUTO
-* Required to support IE8 with images larger than 32K
ON TABLE SET HTMLARCHIVE ON
-*Required for image positioning in subheads in HTML reports]
ON TABLE SET HTMLCSS ON
-*************************
ON TABLE PCHOLD FORMAT DHTML
ON TABLE SET STYLE *
TYPE=REPORT,COLOR=BLUE,FONT=ARIAL,$
TYPE=HEADING, COLOR=RED, SIZE=14, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=SUBHEAD, COLOR=RED, SIZE=12, STYLE=BOLD, JUSTIFY=CENTER,$
TYPE=SUBFOOT,BY=PRODUCTID,IMAGE=(PRODIMAGE),SIZE=(1 1), POSITION=(0 0),$
TYPE=SUBFOOT,BY=PRODUCTID,OBJECT=FIELD, ITEM=1, WRAP=5,$
ENDSTYLE
END
```

下図は、最初のカテゴリの出力を示しています。



BLOB フィールドのイメージに関するファイルサイズと圧縮の考慮事項

BLOB フィールドに格納されているイメージの実際のサイズは、イメージごとに異なるため、イメージを指定サイズに拡大縮小することで、表形式レポートに適合させることができ

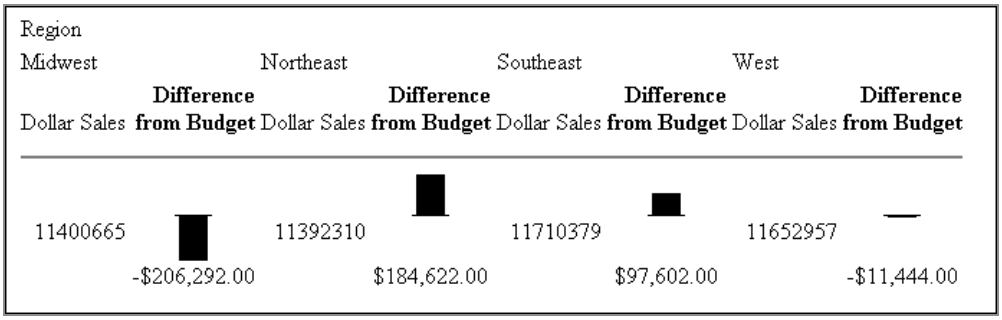
ます。

ファイルに多数のイメージを含めると、ファイルサイズが増大します。SIZE 属性でイメージサイズを縮小しても、ファイルサイズは減少しません。また、イメージはすでに圧縮形式で保存されているため、SET FILECOMPRESS=ON を使用して、PDF ファイルのイメージサイズを減少させることはできません。

レポートデータへのピアグラフの関連付け

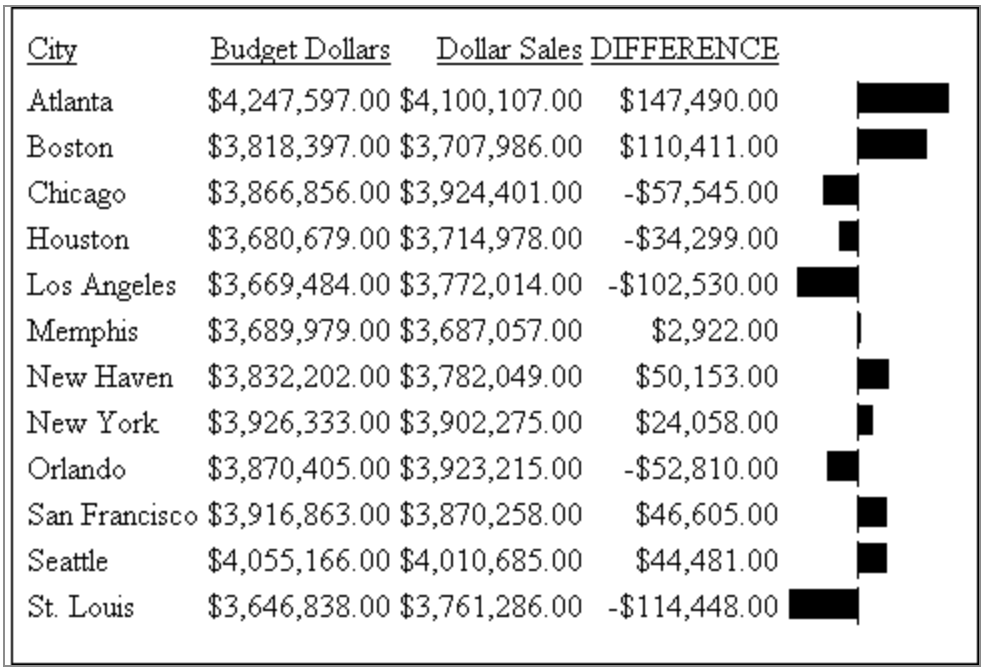
PDF、HTML、DHTML、PPTX、PPT の各レポートをより効果的にするために、選択したデータの視覚的表現を、レポート出力に直接挿入することができます。これらの視覚的な表現は、データ間の関係や動向をより明確にする縦または横のピアグラフ形式で表されます。次のグラフを追加できます。

- **縦のピアグラフ**　ACROSS フィールドに関連付けられたレポートフィールドに、縦のピアグラフを挿入することができます。レポート出力では、下図のように、関連データ値の上の新しい行に縦のピアグラフが表示されます。



  0 (ゼロ) の線より上に伸びたピアグラフは正の値、0 (ゼロ) の線より下に伸びたピアグラフは負の値を示します。

- **横のピアグラフ**　レポートフィールドに横のピアグラフを挿入することができます。レポート出力では、下図のように、関連データ値の右の新しい列に、横のピアグラフが表示されます。



City	Budget Dollars	Dollar Sales	DIFFERENCE
Atlanta	$4,247,597.00	$4,100,107.00	$147,490.00
Boston	$3,818,397.00	$3,707,986.00	$110,411.00
Chicago	$3,866,856.00	$3,924,401.00	-$57,545.00
Houston	$3,680,679.00	$3,714,978.00	-$34,299.00
Los Angeles	$3,669,484.00	$3,772,014.00	-$102,530.00
Memphis	$3,689,979.00	$3,687,057.00	$2,922.00
New Haven	$3,832,202.00	$3,782,049.00	$50,153.00
New York	$3,926,333.00	$3,902,275.00	$24,058.00
Orlando	$3,870,405.00	$3,923,215.00	-$52,810.00
San Francisco	$3,916,863.00	$3,870,258.00	$46,605.00
Seattle	$4,055,166.00	$4,010,685.00	$44,481.00
St. Louis	$3,646,838.00	$3,761,286.00	-$114,448.00

0 (ゼロ) の線より右に伸びたピアグラフは正の値、0 (ゼロ) の線より左に伸びたピアグラフは負の値を示します。

縦横それぞれのピアグラフの長さは、関連するデータ値の大きさに比例しています。最小値には最短のピアグラフ、最大値には最長のピアグラフが表示され、最小値と最大値の間に属する値にはそれぞれの長さを調整したピアグラフが表示されます。上図を見ると、147,480.00 の値の横棒が 50,153.00 の値の横棒より長くなっています。これにより、棒グラフは縦の棒グラフの行全体または横の棒グラフの列全体で形成されます。

ピアグラフは、数値レポートフィールド (整数、小数、単精度浮動点数、倍精度浮動点数、パック 10 進数) に対してのみ挿入することができます。文字、日付またはテキストのフィールドフォーマットに適用されたピアグラフは無視されます。

ピアグラフを列に挿入するには、GRAPHTYPE 属性で始まる宣言を WebFOCUS スタイルシートに追加します。この属性により、指定したデータに縦または横のピアグラフが挿入されます。

注意：OVER オプションを含むリクエストでは、ピアグラフはサポートされていません。

ピアグラフのフォーマットオプション

GRAPHTYPE 宣言には、必要に応じてピアグラフのフォーマット属性を指定することができます (例、グラフの色、長さ、幅)。下表は、フォーマット属性およびその説明の一覧です。

フォーマット属性	説明
GRAPHBASE	目盛りの開始点を 0 (ゼロ) または最低値に設定します。
GRAPHCOLOR	棒グラフの色を設定します。
GRAPHCOLORNEG	負の値を表す棒グラフの色を設定します。
GRAPHLENGTH	最も長い棒グラフの長さを設定します。GRAPHLENGTH の値は、縦棒または横棒グラフで最も長い棒グラフの長さを、使用する測定単位 (インチ、センチメートルなど) で設定します。 長さの値は、UNITS スタイルシート属性を使用して設定された現在の単位で表されます。その後、GRAPHLENGTH の値はピクセルに変換されます。
GRAPHSCALE	ピアグラフを挿入した共通の ACROSS フィールドの下にある複数のレポートフィールドに対して、相対的なピアグラフのスケールを指定します。GRAPHSCALE は、レポートレベルの設定です (TYPE=REPORT)。
GRAPHWIDTH	棒グラフの幅を設定します。幅の値は、単位で表されます。測定単位についての詳細情報は、前述の GRAPHLENGTH を参照してください。

ピアグラフフォーマット属性の組み合わせ

```
TYPE=REPORT, [GRAPHSCALE={UNIFORM|DISTINCT}]
TYPE=DATA, GRAPHTYPE=DATA, [{COLUMN|ACROSSCOLUMN|FIELD}=identifier],
 [GRAPHBASE={ZERO|MINIMUM},]
 [GRAPHCOLOR={color|RGB({r g b|#hexcolor}},]
 [GRAPHNEGCOLOR={color|RGB({r g b|#hexcolor}},]
 [GRAPHLENGTH=lengthvalue,]
 [GRAPHWIDTH=widthvalue,] $
```

注意：TYPE=DATA、GRAPHTYPE=DATA は、GRAPHTYPE=DATA と同義です。

説明

GRAPHBASE

目盛りを 0 (ゼロ) から開始するか、最小値から開始するかを指定します。GRAPHBASE=MINIMUM に設定すると、0 (ゼロ) に近接していない値の比較が容易になります。負の値が存在する場合、GRAPHBASE=MINIMUM は無視され、この設定は有効になりません。

GRAPHBASE=MINIMUM 設定は、グラフの棒の切れ目として視覚的に表現されます (切れ目の前の小さい棒が 0 (ゼロ) から最小値までの圧縮されたスペースを表す)。

注意：最小値は、高さが同一の二重の棒によって識別されます。最小値よりも大きいすべての値では、最上位の棒が、最小値が実際に 0 (ゼロ) の場合は、この二重の棒が表示されるために、最小値が 0 (ゼロ) であることが分かりにくくなる可能性があります。そのため、このフィールドに値 0 (ゼロ) が含まれることが予想される場合は、プロシジャでデフォルト設定 (GRAPHBASE=ZERO) を使用する必要があります。

GRAPHCOLOR

ピアグラフの色を指定します。構文からこの属性を省略する場合、デフォルト色の黒になります。

color

サポートされる色の値です。サポートされる色名のほかに、HEX または RGB 色値も有効なオプションです。サポートされている値のリストについては、レポートのフォントフォーマット指定を参照してください。

RGB(r g b)

赤、緑、青の組み合わせを使用して、色を指定します。(r, g, b) は、赤、緑、青の順序で指定する輝度です。0 (ゼロ) から 255 までの値が使用されます。0 (ゼロ) は最も輝度が低く、255 は最も輝度が高いことを示します。色の構成要素の輝度をすべて同じ値に設定すると、灰色になります。

RGB(#hexcolor)

色の 16 進数値です。たとえば、赤色の 16 進数値は FF0000 です。16 進数値では大文字と小文字は区別されません。また、先頭にシャープ記号 (#) を付ける必要があります。

GRAPHNEGCOLOR

負の値を表す棒グラフの色を定義します。

GRAPHLENGTH

最も長いピアグラフの長さを指定します。縦の棒グラフのデフォルト長は 60 ピクセルで、横の棒グラフのデフォルト長は 80 ピクセルです。

lengthvalue

関連するレポートフィールドの、最大データ値の縦または横の棒グラフを表示するために使用する値を設定します。この値は正の数にする必要があります。

初期状態では、この値は、UNITS 属性を使用した単位で表されます。その後、この値は対応するピクセル値に変換されます。

GRAPHSCALE

ピアグラフを挿入した共通の ACROSS フィールドの下にある複数のレポートフィールドに対して、相対的なピアグラフのスケールを指定します。GRAPHSCALE は、レポートレベルの設定です (TYPE=REPORT)。

UNIFORM

個別の ACROSS フィールドにピアグラフを挿入した場合、それぞれのフィールドから収集した値全体の最小値と最大値に基づいて、それぞれの縦のピアグラフのスケールを調整します。

DISTINCT

個別の ACROSS フィールドにピアグラフを挿入した場合、フィールドごとの最小値と最大値に基づいて、それぞれの縦のピアグラフのスケールが調整されます。

GRAPHWIDTH

レポート内のピアグラフの幅を指定します。

widthvalue

レポート内で棒グラフの幅を表示するために使用した値を設定します。この値は正の数にする必要があります。

初期状態では、この値はUNITS 属性を使用した現在の単位で表されます。その後、この値は対応するピクセル値に変換されます。

レポートフィールドへのピアグラフの関連付け

ピアグラフをレポート出力に挿入するには、WebFOCUS スタイルシートに次の構文を追加します。

```
GRAPHTYPE=DATA, {COLUMN|ACROSSCOLUMN}=identifier, $
```

説明

GRAPHTYPE=DATA

レポートのデータコンポーネントに対して、縦または横のピアグラフを作成します。現在、レポートコンポーネントとして指定できるのは DATA のみに限られます。

COLUMN

指定したレポートフィールドの右側に横のピアグラフを表示します。

ACROSSCOLUMN

ACROSS フィールドに関連付けられたそれぞれのデータ値の上側に縦のピアグラフを表示します。

identifier

有効な任意の識別子です。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

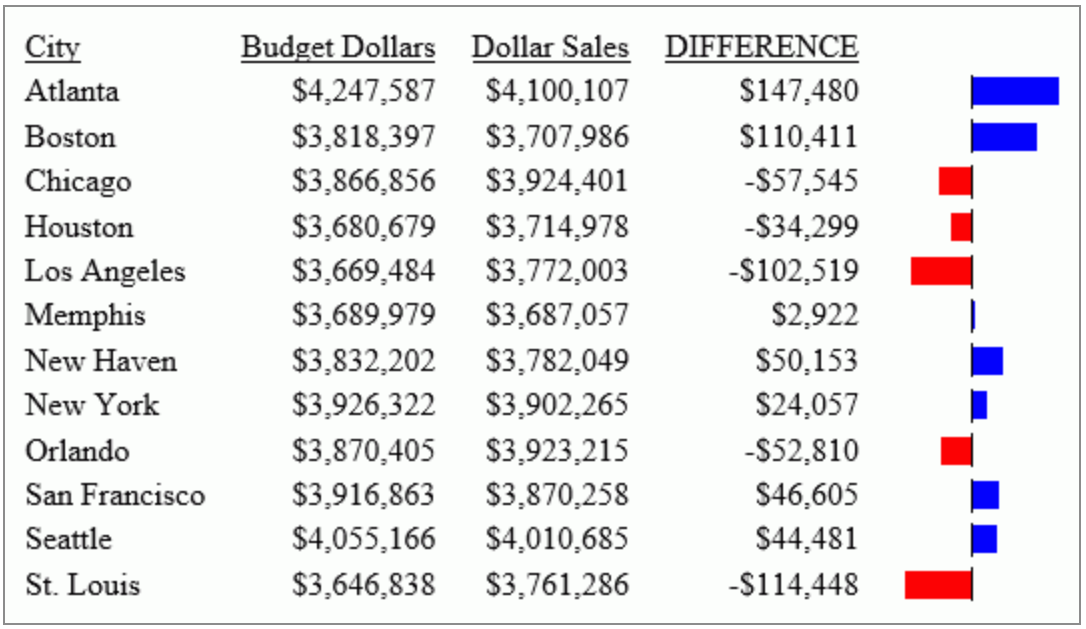
スタイルシート構文を使用して、WHEN 条件とこの条件に関連するグラフ機能を定義することができます。詳細は、レポートデータのフォーマットを参照してください。

レポートでのピアグラフの作成

次の例は、レポートのデータに対してピアグラフを作成する方法を示しています。このレポートは、BY フィールドの CITY でソートされているため、出力結果には横のピアグラフが表示されています。これを縦のピアグラフに変更するには、ソートフィールドを ACROSS CITY に変更し、スタイルシートを GRAPHTYPE=DATA, ACROSSCOLUMN=DIFFERENCE, $ に変更します。正の値を表す棒グラフは、青色で表示されます。負の値を表す棒グラフは、赤色で表示されます。

```
DEFINE FILE GGSALES
DIFFERENCE/D7M=BUDDOLLARS-DOLLARS;
END

TABLE FILE GGSALES
BY CITY
SUM BUDDOLLARS/D7M DOLLARS/D7M DIFFERENCE
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
GRID=OFF, $
GRAPHTYPE=DATA, COLUMN=N4, GRAPHCOLOR=BLUE, GRAPHNEGCOLOR=RED, $
ENDSTYLE
END
```

出力結果は次のとおりです。

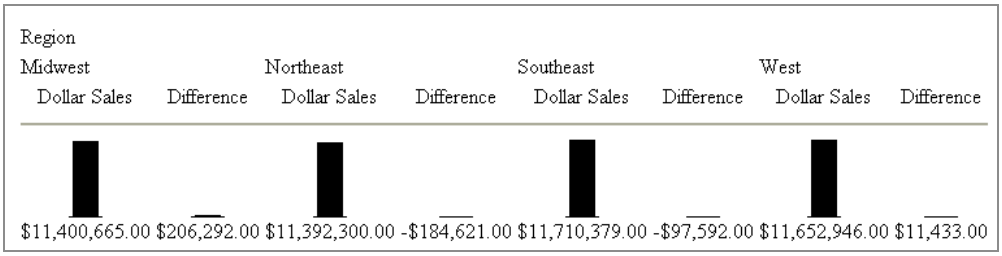


City	Budget Dollars	Dollar Sales	DIFFERENCE
Atlanta	$4,247,587	$4,100,107	$147,480
Boston	$3,818,397	$3,707,986	$110,411
Chicago	$3,866,856	$3,924,401	-$57,545
Houston	$3,680,679	$3,714,978	-$34,299
Los Angeles	$3,669,484	$3,772,003	-$102,519
Memphis	$3,689,979	$3,687,057	$2,922
New Haven	$3,832,202	$3,782,049	$50,153
New York	$3,926,322	$3,902,265	$24,057
Orlando	$3,870,405	$3,923,215	-$52,810
San Francisco	$3,916,863	$3,870,258	$46,605
Seattle	$4,055,166	$4,010,685	$44,481
St. Louis	$3,646,838	$3,761,286	-$114,448

ACROSS フィールドでのピアグラフスケール制御

共通の ACROSS フィールドごとに縦のピアグラフを挿入することができます。ACROSS フィールドの値全体がグループ化されて、ピアグラフとして挿入されます。

この方法は、種類の異なるフィールドで、それぞれの値の範囲が類似している場合に適しています。ただし、種類の異なるフィールドの値をまとめて取り扱う場合は、その差が大きくなりすぎることがよくあります。下図は、その相違を示しています。



上記の数字のように、Dollar Sales フィールドの値 (最小の $11,392,300.00 から最大の $11,710,379.00 まで) は、Difference フィールドの値 (最小の -$206,292.00 から最大の $184,621.00 まで) に比べてはるかに大きい数字になっています。その結果、値全体をグラフに表示した場合、Difference フィールド値に対する縦のピアグラフがほとんど見えなくなっています。

個々のフィールドの値セットに基づいて縦のピアグラフを表示するには、GRAPHSCALE スタイルシート属性を使用します。この属性により、共通の ACROSS フィールドの下にあるフィールドごとの最小値と最大値を使用するよう変更され、縦のピアグラフが別々に作成されます。

ピアグラフの方向の設定

VISBARORIENT パラメータを使用すると、ACROSS フィールドのピアグラフの方向を縦方向または横方向に設定することができます。

注意：このパラメータは、HTML レポート出力でのみサポートされます。

```
SET VISBARORIENT = {H|V}
```

説明

H

ピアグラフの方向に縦方向を指定します。

V

ピアグラフの方向に横方向を指定します。これがデフォルト値です。

ピアグラフの方向の設定

次のレポートは、ACROSS フィールド値 (DOLLARS、BUDDOLLARS、UNITS、BUDUNITS) の縦棒を作成します。

```
SET VISBARORIENT=V
TABLE FILE GGSALES
SUM DOLLARS BUDDOLLARS UNITS BUDUNITS
BY REGION
ACROSS CATEGORY
WHERE CATEGORY EQ 'Coffee' OR 'Food'
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
TYPE =REPORT , BORDER=light,$
GRAPHTYPE=DATA, ACROSSCOLUMN=DOLLARS, GRAPHCOLOR=GREEN,BACKCOLOR=RGB
(#ffff00), $
GRAPHTYPE=DATA, ACROSSCOLUMN=BUDDOLLARS, GRAPHCOLOR=RGB(255 0 0), $
GRAPHTYPE=DATA, ACROSSCOLUMN=UNITS, GRAPHCOLOR=blue, $
GRAPHTYPE=DATA, ACROSSCOLUMN=BUDUNITS, GRAPHCOLOR=thistle, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	Category							
	Coffee				Food			
Region	Dollar Sales	Budget Dollars	Unit Sales	Budget Units	Dollar Sales	Budget Dollars	Unit Sales	Budget Units
Midwest	4178513	4086032	332777	335526	4338271	4220721	341414	339263
Northeast	4164017	4252462	335778	335920	4379994	4453907	353368	351431
Southeast	4415408	4431429	350948	355693	4308731	4409288	349829	351509
West	4473517	4523963	356763	358784	4202337	4183244	340234	335361

次のレポートは、ACROSS フィールド値 (DOLLARS および BUDDOLLARS) の縦棒を作成します。

```
SET VISBARORIENT=H
TABLE FILE GGSALES
SUM DOLLARS BUDDOLLARS
```

```
BY REGION
ACROSS CATEGORY
WHERE CATEGORY EQ 'Coffee' OR 'Food'
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
TYPE =REPORT ,BORDER=light,$
GRAPHTYPE=DATA, ACROSSCOLUMN=DOLLARS, GRAPHCOLOR=GREEN,BACKCOLOR=RGB
(#ffff00), $
GRAPHTYPE=DATA, ACROSSCOLUMN=BUDDOLLARS, GRAPHCOLOR=RGB(255 0 0), $
ENDSTYLE
END
```

出力結果は次のとおりです。

	Category			
	Coffee		Food	
Region	Dollar Sales	Budget Dollars	Dollar Sales	Budget Dollars
Midwest	4178513	4086032	4338271	4220721
Northeast	4164017	4252462	4379994	4453907
Southeast	4415408	4431429	4308731	4409288
West	4473517	4523963	4202337	4183244

ピアグラフへのスケールの適用

GRAPHSCALE パラメータは、ACROSS ソートフィールド下でピアグラフを適用した複数のレポートフィールドに対して、相対的な棒グラフのスケールを指定します。GRAPHSCALE は、レポート全体 (TYPE=REPORT) のみに設定できます。

ピアグラフへのスケールの適用

```
TYPE=REPORT, GRAPHSCALE={UNIFORM|DISTINCT}
```

説明

TYPE=REPORT

構文が、レポート内の個別のピアグラフではなく、レポート全体に適用されることを指定します。

GRAPHSCALE

ピアグラフを挿入した共通の ACROSS フィールドの下にある複数のレポートフィールドに対して、相対的なピアグラフのスケールを指定します。GRAPHSCALE は、レポートレベルの設定です (TYPE=REPORT)。

UNIFORM

個別の ACROSS フィールドにピアグラフを挿入した場合、それぞれのフィールドから収集した値全体の最小値と最大値に基づいて、それぞれの縦のピアグラフのスケールを調整します。

DISTINCT

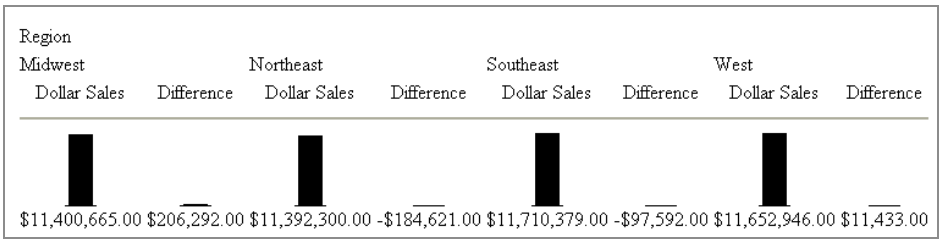
個別の ACROSS フィールドにピアグラフを挿入した場合、フィールドごとの最小値と最大値に基づいて、それぞれの縦のピアグラフのスケールが調整されます。

GRAPHSCALE による縦のピアグラフの個別表示

次のリクエストは、共通の ACROSS フィールド (REGION) に関連する 2 つのフィールド (DOLLARS と DIFFERENCE) に対して縦のピアグラフを表示します。

```
DEFINE FILE GGSALES
Difference/D12.2M=DOLLARS-BUDDOLLARS;
END

TABLE FILE GGSALES
SUM DOLLARS/D12.2M Difference
ACROSS REGION
ON TABLE SET STYLE *
TYPE=REPORT,GRID=OFF,$
GRAPHTYPE=DATA, ACROSSCOLUMN=N1,$
GRAPHTYPE=DATA,ACROSSCOLUMN=N2,$
ENDSTYLE
END
```

このリクエストによるレポートの出力結果は次のとおりです。

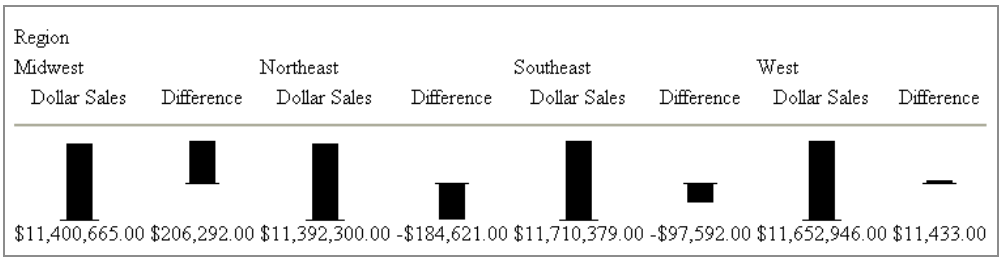


GRAPHSCALE 属性が指定されていないため、デフォルト設定の UNIFORM がレポートに適用されています。この設定では、値全体 (Dollar Sales と Difference の両方の値) を使用して、両方のフィールドに対するピアグラフが作成されています。

次のリクエストは上記のリクエストと同じですが、スタイルシートに GRAPHSCALE=DISTINCT 属性を指定しています。

```
DEFINE FILE GGSALES
Difference/D12.2M=DOLLARS-BUDDOLLARS;
END

TABLE FILE GGSALES
SUM DOLLARS/D12.2M Difference
ACROSS REGION
ON TABLE SET STYLE *
GRAPHTYPE=DATA, ACROSSCOLUMN=N1,$
GRAPHTYPE=DATA, ACROSSCOLUMN=N2,$
TYPE=REPORT, GRAPHSCALE=DISTINCT,$
ENDSTYLE
END
```

出力結果の相違に注目してください。



それぞれのピアグラフが、各フィールドの値に基づいて別々に作成されています。

宛名ラベルと複数枠ページの使用

1 つのラベルがそれぞれに対応するよう、各ページをサブページのマトリックスに分割することで、レーザプリンタ用宛名ラベル用紙を印刷することができます。レポート内の各改

ページは、次のラベルの開始位置にプリンタを設定します。

複数枠が含まれる場合は、印刷時にすべてのレポートを 1 ページに収まるようにします。テキストがページからはみ出したときに、次のページではなく、同じページの次の列または行に表示されるよう、行または列を作成します。

これらの機能は、PDF レポートに適用されます。

宛名ラベルおよび複数枠印刷用属性

テーブル内の属性に加え、標準マージン属性 (例、LEFTMARGIN、TOPMARGIN) を使用することにより、ラベル用紙すべての位置を一括指定し、各用紙のマージンを同じにすることができます。

属性	説明	適用先
`PAGEMATRIX`	1 ページ当たりのラベルの行数と列数を設定します。	PDF
`ELEMENT`	UNITS 属性で指定された測定単位で、各ラベルの幅および高さを設定します。	PDF
`GUTTER`	UNITS 属性で指定された測定単位で、各ラベル間の縦と横の距離を設定します。	PDF
`MATRIXORDER`	ラベルを印刷する順序を設定します。	PDF
`LABELPROMPT`	宛名ラベル用紙の最初のラベルの位置を設定します。	PDF

レポートの宛名ラベル印刷を設定するには

手順

1. ラベルをページ見出しとして作成します。
2. ソートフィールドを非表示にするために NOPRINT を使用して、ラベルのソートを行います。ページ見出しに埋め込まれたフィールドのみを印刷します。
3. 個別ラベルにそれぞれの新しいフィールドを配置するために、ソートフィールドに改ページを挿入します。
4. それぞれのページ (SET PAGE-NUM=NOPAGE) の開始からデフォルトページ番号および関連したブランク行を非表示にします。

宛名ラベルまたは複数枠レポートの印刷

```
[TYPE=REPORT,] PAGEMATRIX=(c r), ELEMENT=(w h), [GUTTER=(x y),]
  [MATRIXORDER={VERTICAL|HORIZONTAL},]  [LABELPROMPT={OFF|ON},] $
```

説明

TYPE=REPORT

設定をレポート全体に適用します。デフォルト値であるため必須ではありません。

c

ページ横方向のラベル列数です。

括弧で *c* および *r* の値を囲みます。その間にカンマ (,) を入れません。

r

ページ縦方向のラベル行数です。

w

各ラベルの幅です。

括弧で *w* および *h* の値を囲みます。その間にカンマ (,) を入れません。

h

各ラベルの高さです。

GUTTER

それぞれのラベル間の距離です。

x

それぞれのラベル間の横方向の距離です。

括弧で *x* および *y* の値を囲みます。その間にカンマ (,) を入れません。

y

それぞれのラベル間の縦方向の距離です。

MATRIXORDER

ラベルが印刷される順序です。

VERTICAL

ラベルをページの縦方向に印刷します。

HORIZONTAL

ラベルをページの横方向に印刷します。

LABELPROMPT

宛名ラベル用紙の最初のラベルの位置です。

OFF

用紙の最初のラベルでレポートを開始します。デフォルト値は OFF です。

ON

印刷を開始する行番号と列番号の指定を実行時に要求します。すべての残りのラベルは、連続的に実行されます。この機能により、途中まで使用したラベル用紙の再利用が可能になります。

宛名ラベルの印刷

次のレポートは、8.5 x 11 インチの宛名ラベル用紙を印刷します。

```
SET ONLINE-FMT = PDF
TABLE FILE EMPLOYEE
BY LAST_NAME NOPRINT BY FIRST_NAME NOPRINT
ON FIRST_NAME PAGE-BREAK
HEADING
"<FIRST_NAME <LAST_NAME"
"<ADDRESS_LN1"
```

```
"<ADDRESS_LN2"
"<ADDRESS_LN3"
ON TABLE SET PAGE-NUM NOPAGE
ON TABLE SET STYLE LABEMP
END
```

このラベルは、スタイルシート LABEMP 内で定義した、次の範囲になります。

```
UNITS=IN, PAGESIZE=LETTER, LEFTMARGIN=0.256, TOPMARGIN=0.5,
PAGEMATRIX=(2 5), ELEMENT=(4 1), GUTTER=(0.188 0), $
```

ラベルの最初のページは、次のように印刷されます。

```
JOHN BANNING                    DIANE JONES
160 LOMBARDO AVE.
APT 4C                          235 MURRAY HIL PKWY
FREEPORT NY 11520               RUTHERFORD NJ 07073


ROSEMARIE BLACKWOOD             JOHN MCCOY
MRS. P. JONES                   ASSOCIATED
3704 FARRAGUT RD.               2 PENN PLAZA
BROOKLYN NY 11210               NEW YORK NY 10001


BARBARA CROSS                   ROGER MCKNIGHT
APT 2G                          APT 4D
147-15 NORTHERN BLD             117 HARRISON AVE.
FLUSHING NY 11354               ROSELAND NJ 07068


MARY GREENSPAN                  ANTHONY ROMANS

13 LINDEN AVE.                  271 PRESIDENT ST.
JERSEY CITY NJ 07300            FREEPORT NY 11520


JOAN IRVING                     MARY SMITH
APT 2J                          ASSOCIATED
123 E 32 ST.                    2 PENN PLAZA
NEW YORK NY 10001               NEW YORK NY 10001
```

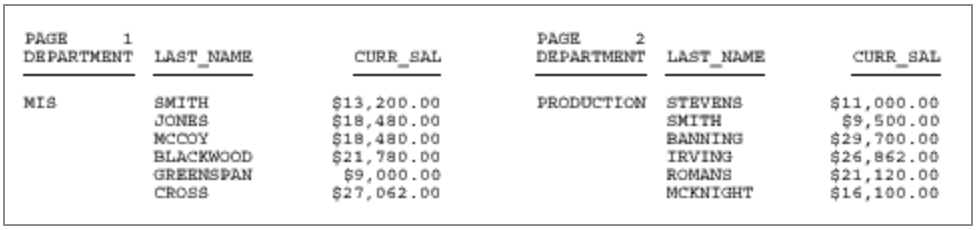
複数枠レポートの印刷

次のリクエストは、最初のレポートページを 2 列に分割します。2 番目のレポートページは、最初のページの 2 列目に表示されます。PAGE-BREAK は、この例の目的である複数ページレポートを作成します。

```
SET ONLINE-FMT = PDF
TABLE FILE EMPLOYEE
PRINT LAST_NAME AND CURR_SAL BY
```

```
DEPARTMENT
ON DEPARTMENT PAGE-BREAK
HEADING
"PAGE <TABPAGENO"
ON TABLE SET STYLE *
UNITS=IN, PAGESIZE=LETTER, PAGEMATRIX=(2 1), ELEMENT=(3.5 8.0),
MATRIXORDER=VERTICAL, $
TYPE=REPORT, SIZE=8, $
ENDSTYLE
END
```

レポートは次のように印刷されます。



PAGE 1			PAGE 2		
DEPARTMENT	LAST_NAME	CURR_SAL	DEPARTMENT	LAST_NAME	CURR_SAL
MIS	SMITH	$13,200.00	PRODUCTION	STEVENS	$11,000.00
	JONES	$18,480.00		SMITH	$9,500.00
	MCCOY	$18,480.00		BANNING	$29,700.00
	BLACKWOOD	$21,780.00		IRVING	$26,862.00
	GREENSPAN	$9,000.00		ROMANS	$21,120.00
	CROSS	$27,062.00		MCKNIGHT	$16,100.00

見出し、脚注、タイトル、ラベルの使用

レポートに表示するデータを選択した後、見出し、脚注、タイトル、ラベルを追加することにより、分かりやすいレポートにすることができます。見出しおよび脚注は、レポートの目的や対象者などの重要な情報を提供します。また、これらのコンポーネントでレポートの構造が明確になり、ユーザが必要とする詳細情報を簡単に特定することができます。タイトルおよびラベルは、データが正しく解釈されるよう、個別データを識別します。これらのコンポーネントにより、データの意味が明確になり、レポートの外観も向上します。

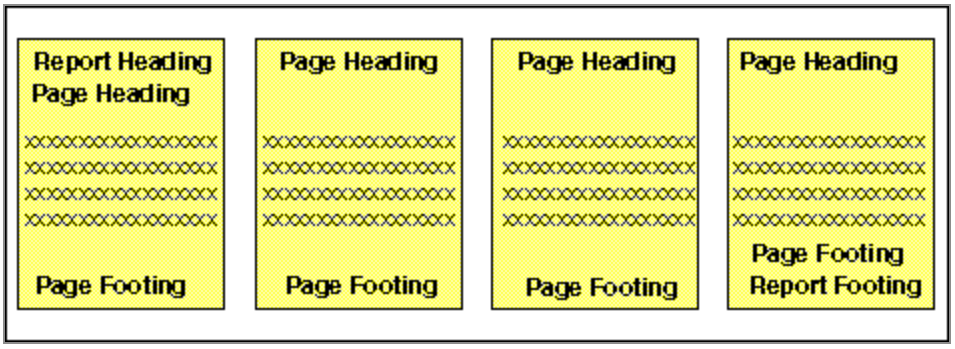
特別なタイプのレポートにコンテキスト情報を追加する方法についての詳細は、グラフの作成を参照してください。

見出しと脚注の作成

見出しと脚注にはいくつかのタイプがあります。

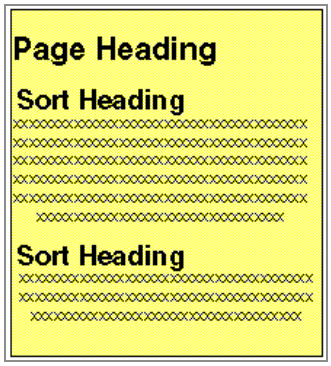
- レポートタイトル。定義するタイトルには、レポートを HTML として実行するときに、ブラウザのタイトルバーに表示されるものがあります。また、EXL2K レポートのワークシートのタブとして定義するタイトルもあります。詳細は、カスタムレポートタイトルまたはワークシートタイトルの作成を参照してください。
- レポート見出しとレポート脚注。レポート見出しはレポートの最初のページの上部に、レポート脚注は、レポートの最終ページに表示されます。レポート見出しとレポート脚注についての詳細は、レポート見出しとレポート脚注の作成を参照してください。
- ページ見出しとページ脚注。ページ見出しはレポート各ページの上部に表示され、ページ脚注はレポートの各ページの下部に表示されます。ページ見出しとページ脚注についての詳細は、ページ見出しまたはページ脚注の作成を参照してください。
- ソート項目を指定している見出しとソート項目を指定している脚注。ソート項目を指定している見出しは、関連データのグループの開始を識別するために、レポート本文中に表示されますソート項目を指定している脚注は、関連データのグループの終了を識別するために、レポート本文中に表示されます。ソート項目を指定している見出しおよび脚注についての詳細は、見出しと脚注の作成を参照してください。

次のサンプルレポートには、レポートのはじめにレポート見出し、レポートの最後にレポート脚注がそれぞれ含まれています。さらに、レポート各ページには、ページ見出しとページ脚注が含まれています。

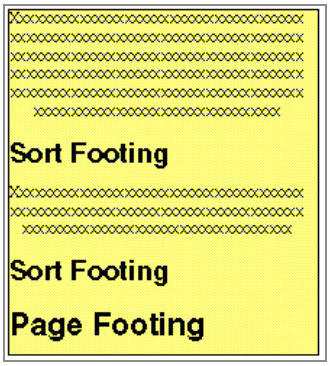


次のサンプルレポートには、参照用に、ページ見出しおよびページ脚注のほかに、ソート項目を指定している見出しおよびソート項目を指定している脚注が含まれてます。

ソート項目を指定している見出しは、次のようになります。



ソート項目を指定している脚注は、次のようになります。



見出しと脚注の制限事項

次の制限は、レポート見出しと脚注、ページ見出しと脚注、ソート項目を指定している見出しと脚注に適用されます。

- 見出し、脚注、中間見出し、中間脚注のスペースは、動的に割り当てられます。WebFOCUS では、使用するスペースに制限はありません。
- 1 つのリクエストで使用できるソート項目を指定している見出しと脚注の合計は、最大

で 64 バイトです。

- ネストした見出しの最大制限数は 64 です。
- リクエストで、見出しまたは脚注の 1 行のコードが複数行に分割された場合、<0X スポットマーカーを使用することで、それらすべてを 1 行の見出しにすることができます。
- PDF レポートで見出しと脚注を適切に表示するには、レポートの最大幅に収める必要があります。また、レポート本文を表示するために、見出し、脚注の行数に、少なくとも詳細行 1 行分のブランク (フィールドタイトルを含む) を追加する必要があります。

見出し脚注リクエストコードの複数行での指定

二重引用符 (") で囲まれた 1 行の見出しまたは脚注コードは、最大 32 キロバイトです。ただし、エディタには、プロシジャの 1 行のコードの最大長が、80 バイトのものがあります。このような場合には、<0X スポットマーカーを使用することで、見出しを次の行に継続することが可能です。見出しまたは脚注の内容および間隔は、1 行に入力したときと同様に表示されます。

1 行につき 80 バイトの制限を超えるコードの拡張を行う必要がない場合でも、短い行は画面上で読みやすく、プリンタで簡単に印刷できることから、この方法は役立ちます。

リクエストの見出し、脚注コードを複数行で指定するには

1 行の見出しまたは脚注の長さを 81 バイト以上に拡張するには、次の手順を実行します。

手順

1. 二重引用符 (") で見出しまたは脚注を指定します。
2. 見出しまたは脚注を 76 バイト以内の複数行に分割します。見出しを次の行に継続するために、76 バイト以内の任意の位置に <0X スポットマーカーを使用します。残りの 4 バイトは、スポットマーカー自体と、その前の 1 つのブランクとして必要です。

3. 見出しまたは脚注行には、各行の終わりの <0X スポットマーカーを含めて、最大で 410 バイトを指定することができます。
4. 見出しまたは脚注コードの最終行の終わりに二重引用符 (") を指定します。

結果

この方法で、最大で 410 バイトのレポート見出しとレポート脚注、ページ見出しとページ脚注、ソート項目を指定している見出しとソート見出しを指定している脚注を作成することができます。

見出し脚注リクエストコードの複数行での指定

次のリクエストは、ソート項目を指定している見出しを 2 行に記述して作成します。<0X スポットマーカーは、継続行の最初の文字を、前の行の最後の文字のすぐ右側に配置します。スポットマーカーと継続行の開始点の間には、ブランクは挿入されません。

```
SET ONLINE-FMT = HTML
SET PAGE-NUM = OFF
JOIN STORE_CODE IN CENTCOMP TO STORE_CODE IN CENTORD
```

```
TABLE FILE CENTCOMP
HEADING
"Century Corporation Orders Report"
PRINT PROD_NUM QUANTITY LINEPRICE
BY STORE_CODE NOPRINT
BY ORDER_NUM
ON STORE_CODE SUBHEAD
"Century Corporation orders for store <STORENAME <0X
    (store # <STORE_CODE|) in <STATE|."
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=HEADING, FONT='ARIAL', STYLE=BOLD, $
TYPE=SUBHEAD, OBJECT=FIELD, ITEM=2, STYLE=ITALIC, $
TYPE=SUBHEAD, OBJECT=FIELD, ITEM=3, STYLE=BOLD, $
ENDSTYLE
END
```

以下は、出力結果の一部を示しています。

Century Corporation Orders Report

Order Number:	Product Number#:	Quantity:	Line Total
Century Corporation orders for store Audio Expert (store # *1003CA*) in **CA**.			
74610	1012	268	$222,234.35
	1028	323	$29,215.62
	1032	339	$27,208.12
	1034	339	$154,943.38
	1036	339	$93,266.39
Century Corporation orders for store Audio Expert (store # *1003CO*) in **CO**.			
39274	1006	179	$54,201.50
	1008	179	$29,034.16
	1020	179	$45,457.59
	1032	400	$30,861.24
	1034	400	$143,744.49
Century Corporation orders for store Audio Expert (store # *1003CT*) in **CT**.			
35995	1008	13	$2,132.45
	1020	13	$3,895.38
	1022	146	$52,159.13
	1024	146	$45,306.84
	1030	146	$18,525.49

ヒント：この方法で、最大 410 バイトの見出しを作成することができます。ここではソート項目を指定している見出しを示していますが、この方法は、任意の見出しまたは脚注の行に使用することができます。

カスタムレポートタイトルまたはワークシートタイトルの作成

次のレポートタイトルを作成することができます。

- HTML (グラフ) レポートのブラウザタイトルに表示するデフォルトレポートタイトル (FOCUS レポート) を上書きする。
- EXL2K レポートのデフォルトワークシートタブ名を、指定した名前で置き換える。

Excel 目次レポートのワークシートのタブ名は、現在のワークシート上のデータに対応する BY フィールドの値になります。ユーザがスタイルシートに TITLETEXT キーワードを指定した場合は、その名前は無視されます。

- Excel では、ワークシートのタイトルの最大長は 31 バイトに限定されています。特殊文字の「:」、「?」、「*」、「/」を使用することはできません。
- データフィールドを分割用 BY フィールドとして使用する場合は、「/」の代わりに「-」を使用することができます。「-」は、Excel のタブ名で有効になります。ただし、「/」を使用すると、WebFOCUS により「-」に変更されます。

カスタムレポートタイトルの作成

WebFOCUS スタイルシートに、次の宣言を追加します。

```
TYPE=REPORT, TITLETEXT='title', $
```

説明

title

タイトルテキストです。

最大バイト数は、次のとおりです。

- EXL2K レポートのワークシートタブ名の制限は、31 バイトです。31 バイトを超えるテキストは、切り捨てられます。
- HTML (グラフ) レポートのブラウザタイトルは、95 バイトです。これはブラウザによる制限です。

  タイトルで指定されたテキストは、そのままファイル内に配置され、暗号化はされません。<、>、& などの特殊文字は、HTML 内で特別な意味を持ち、予期しない結果を生じる場合があるため、使用は推奨されません。

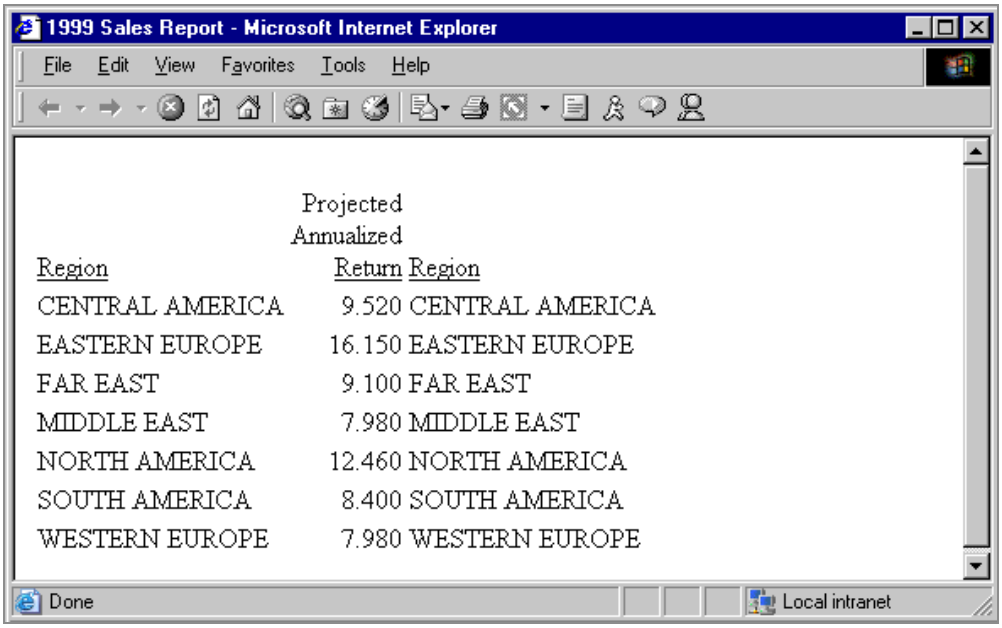
  注意：HTML レポートタイトルには、「Internet Explorer」 が常に追加されます。

HTML カスタムレポートタイトルの作成

次の例は、スタイルシートで TITLETEXT 属性を使用して、HTML レポートのデフォルトレポートタイトルを置き換える方法を示しています。

```
TABLE FILE SHORT
SUM PROJECTED REGION
BY REGION
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT='1999 Sales Report', $
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。



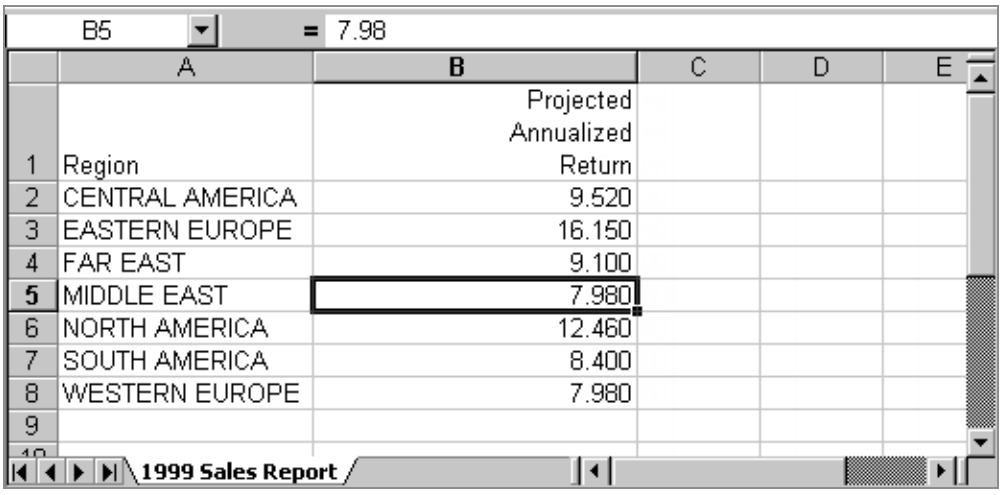
EXL2K カスタムレポートタイトルの作成

次の例は、スタイルシートで TITLETEXT 属性を使用して、EXL2K レポートのデフォルトワークシートタブ名を置き換える方法を示しています。

```
TABLE FILE SHORT
SUM PROJECTED_RETURN
BY REGION
ON TABLE PCHOLD FORMAT EXL2K
ON TABLE SET STYLE *
TYPE=REPORT, TITLETEXT='1999 Sales Report', $
```

```
ENDSTYLE
END
```

出力結果は次のとおりです。



	A	B	C	D	E
1	Region	Projected Annualized Return			
2	CENTRAL AMERICA	9.520			
3	EASTERN EUROPE	16.150			
4	FAR EAST	9.100			
5	MIDDLE EAST	7.980			
6	NORTH AMERICA	12.460			
7	SOUTH AMERICA	8.400			
8	WESTERN EUROPE	7.980			
9					

B5 = 7.98

1999 Sales Report

レポート見出しとレポート脚注の作成

レポート見出しは、最初のページの先頭に表示され、レポートの最も重要なコンポーネントの 1 つです。レポートに一意の名前を与え、その目的または内容を特定します。短い 1 行のレポート見出しで十分な場合もありますが、必要に応じて、情報を複数行含めることもできます。

レポート脚注は、レポートの最終ページの後に表示されます。レポートの最後であることをユーザに知らせるために、データの最後を示すレポート脚注を追加することができます。レポート脚注には、レポートの作成者などの情報を指定することもできます。

レポート見出しおよびレポート脚注には、テキスト、フィールド、ダイアログマネージャ変数、イメージ、スポットマーカーを含めることができます。

レポート見出しの作成

リクエストに次の構文を含めます。見出しおよび脚注の各行は、二重引用符 (") で開始、終了する必要があります。

```
ON TABLE [PAGE-BREAK AND] SUBHEAD
 "content ... "
["content ... "]
 を使用します。
 を使用します。
 を使用します。
["content ... "]
```

説明

PAGE-BREAK

レポート見出しのみを含む先頭ページを作成する、オプションのコマンドです。その後に 1 ページまたは複数ページのデータが続きます。PAGE-BREAK を使用しない場合は、レポート見出しはレポートの最初のページに、ページ見出し (存在する場合)、フィールドタイトルを伴って表示されます。詳細は、レポート見出しまたは脚注の単独ページ配置を参照してください。

SUBHEAD

レポート見出しを指定するために必要なコマンドです。

content

見出しまたは脚注の内容には、次の要素を二重引用符 (") で囲んで含めることができます。終了用引用符が省略された場合、リクエストの後に続くすべての行は、レポート見出しの一部分として扱われます。

text

レポートの最初のページに表示するテキストです。複数行のテキストを含めることができます。

テキストは、SUBHEAD コマンドに続けて、テキストのみの行で開始する必要があります。

テキストは、変数およびスポットマーカーと組み合わせることができます。

variable

次の 1 つまたは組み合わせです。

`Fields` - 実データソースフィールド、マスターファイルまたはリクエスト内で DEFINE コマンドを使用して作成した一時項目 (DEFINE) 、リクエスト内で COMPUTE コマンドを使用して作成した一時項目 (COMPUTE) 、または TABPAGENO などのシステム

フィールドです。データソースフィールドは、特定の演算接頭語で修飾することができます。

```
Dialogue Manager variables.
```

Images - 見出しまたは脚注にイメージを含めることができます。

詳細は、見出しまたは脚注内への要素の追加を参照してください。

スポットマーカー

項目の配置、フォーマット設定項目の識別が可能です。

<+0> - フォーマット設定を目的として、見出しまたは脚注を項目に分割します。

</n - ブランク行を指定します。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。

<-n - 行に次の文字を配置します。詳細は、スポットマーカーを使用した配置調整を参照してください。

<0X - 見出しまたは脚注指定を、リクエストの次の行に継続します。

注意：終了スポットマーカーの直後に開始スポットマーカーを追加すると (><)、この 2 つのスポットマーカー間にブランクが 1 つ挿入されます (> <)。フォーマット設定を適用する際は、このことを考慮する必要があります。

ブランク行

すべてのテキスト、変数、スポットマーカーを省略する場合、見出しや脚注がブランクになります (例、" ")。これを使用して、ブランク行を挿入することができます。ブランク行を挿入するには、</1 などの縦のスポットマーカーを使用することもできます。

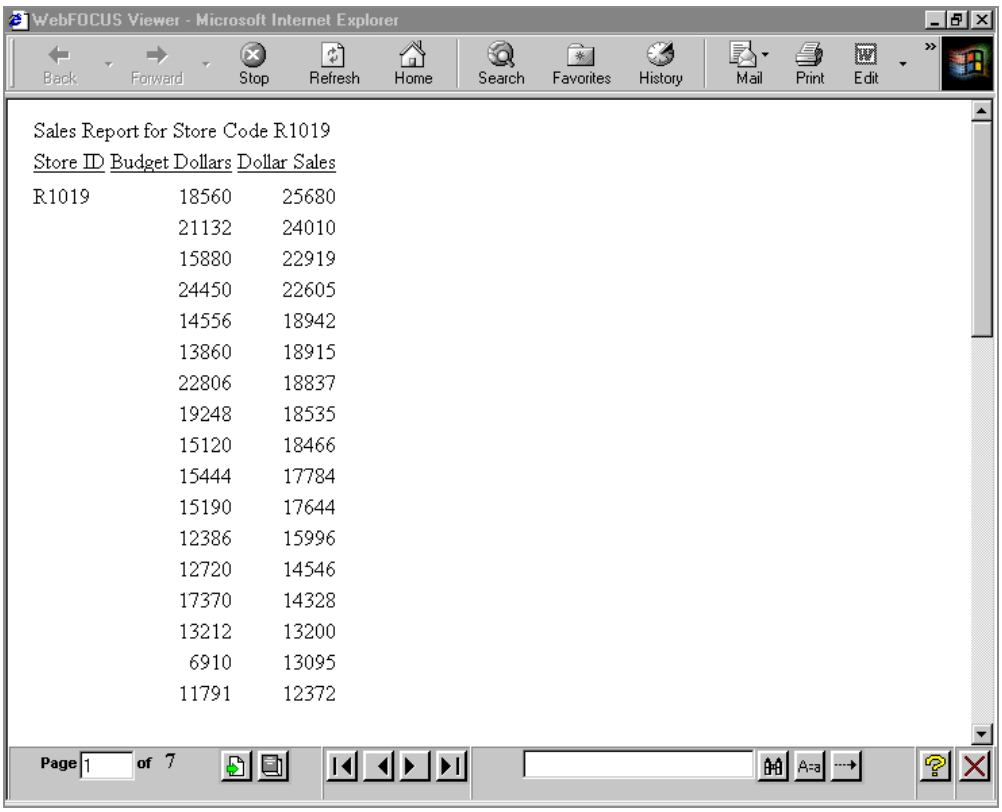
単数行レポート見出しの作成

次のリクエストは、レポートの内容を識別する単数行のレポート見出しを作成します。

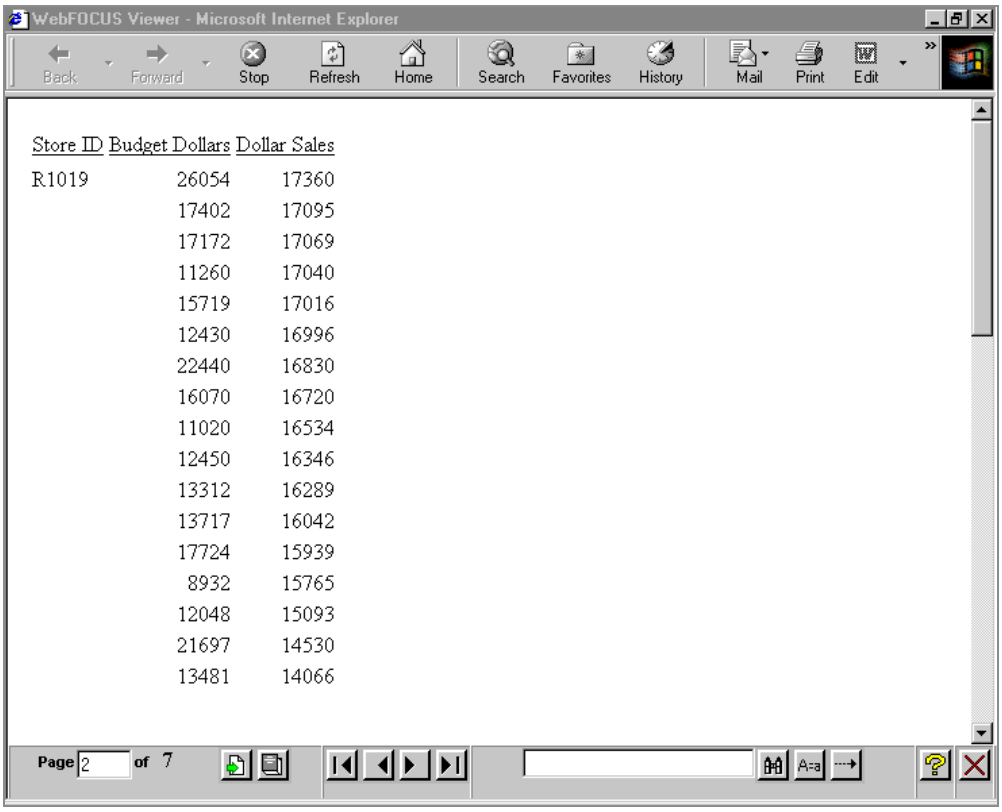
```
TABLE FILE GGSALES
PRINT BUDDOLLARS DOLLARS
BY STCD
WHERE STCD EQ 'R1019'
ON TABLE SUBHEAD
"Sales Report for Store Code R1019"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
```

```
ON TABLE SET WEBVIEWER ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

出力には、Web ビューアレポートにレポート見出しが表示されています。レポート見出しは、最初のページの最上部になります。



後に続くページには、レポート見出しは表示されません。



ヒント：ナビゲーションの矢印が表示されない場合、最大化ボタンをクリックします。

レポート脚注の作成

リクエストに次の構文を含めます。見出しおよび脚注の各行は、二重引用符 (") で開始、終了する必要があります。

```
ON TABLE [PAGE-BREAK AND] SUBFOOT
 "content ... "
["content ... "]
 を使用します。
 を使用します。
 を使用します。
["content ... "]
```

説明

PAGE-BREAK

最終ページの後にレポート脚注のみのページを作成する、オプションのコマンドです。PAGE-BREAK を使用しない場合、レポート脚注はレポートの最終行に表示されます。詳細は、レポート見出しまたは脚注の単独ページ配置を参照してください。

SUBFOOT

レポート脚注を指定するために必要なコマンドです。

content

見出しまたは脚注の内容には、次の要素を二重引用符 (") で囲んで含めることができます。終了用引用符が省略された場合、<0X スポットマーカーを使用しない限り、リクエストの後に続くすべての行は、レポート脚注の一部分として扱われます。

text

レポートの最後のページに表示するテキストです。複数行のテキストを含めることができます。

テキストは、SUBFOOT コマンドに続いてテキストのみの行で開始する必要があります。

テキストは、変数およびスポットマーカーと組み合わせることができます。

variable

次の 1 つまたは組み合わせです。

`Fields` - 実データソースフィールド、マスターファイルまたはリクエスト内で DEFINE コマンドを使用して作成した一時項目 (DEFINE) 、リクエスト内で COMPUTE コマンドを使用して作成した一時項目 (COMPUTE) 、または TABPAGENO などのシステムフィールドです。データソースフィールドは、特定の演算接頭語で修飾することができます。

```
Dialogue Manager variables.
```

`Images` - 見出しまたは脚注にイメージを含めることができます。

詳細は、見出しまたは脚注内への要素の追加を参照してください。

スポットマーカー

項目の配置、フォーマット設定項目の識別が可能です。

`<+0>` - フォーマット設定を目的として、見出しまたは脚注を項目に分割します。

</n - ブランク行を指定します。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。

<-n - 行に次の文字を配置します。詳細は、スポットマーカーを使用した配置調整を参照してください。

<0X - 見出しまたは脚注指定を、リクエストの次の行に継続します。

注意：終了スポットマーカーの直後に開始スポットマーカーを追加すると (><)、この 2 つのスポットマーカー間にブランクが 1 つ挿入されます (> <)。フォーマット設定を適用する際は、このことを考慮する必要があります。

ブランク行

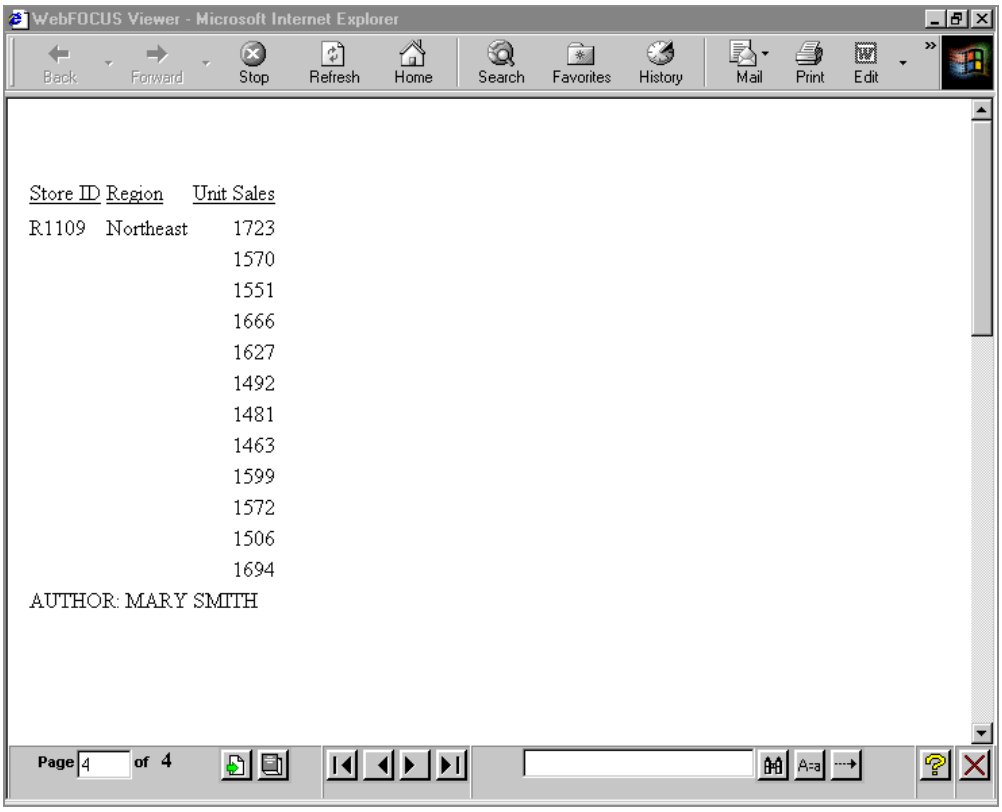
すべてのテキスト、変数、スポットマーカーを省略する場合、見出しや脚注がブランクになります (例、" ")。これを使用して、ブランク行を挿入することができます。ブランク行を挿入するには、</1 などの縦のスポットマーカーを使用することもできます。

単数行レポート脚注の作成

次のリクエストは、レポートの作成者を識別する単数行のレポート脚注を作成します。

```
TABLE FILE GGSALES
PRINT UNITS
WHERE UNITS GE 1400
BY STCD BY REGION
WHERE REGION EQ 'Northeast'
ON TABLE SUBFOOT
"AUTHOR:MARY SMITH"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET WEBVIEWER ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

出力には、Web ビューアレポートレポートにページ脚注が表示されています。レポート脚注は、最終ページのデータの後に表示されます。



ヒント：ナビゲーションの矢印が表示されない場合、最大化ボタンをクリックします。

ページ見出しまたはページ脚注の作成

ページ見出しは、レポートの各ページの最上部に表示され、ページ脚注は、レポートの各ページの最下部に表示されます。

ページ見出しを追加すると、ページごとのレポートの内容と目的を識別して補足することができます。また、ページ見出しには、各ページの見出しをカスタマイズするための変数を追加することもできます。たとえば、部門別に分類された従業員の銀行口座情報のレポートについて考察します。各部門の情報をそれぞれ別のページに表示します。このレポートのページ見出しで、各ページに表示される部門名を識別します。たとえば、「ACCOUNT REPORT FOR PRODUCTION DEPARTMENT」というページ見出しを作成します。

ページ脚注を追加すると、レポートの日付や社外秘であることなど、各ページに必ず繰り返し表示される情報を入力することができます。ページ脚注を使用して、レポートに関する記述的情報を提供することもできます。

ページ見出しまたは脚注は、テキスト、フィールド、ダイアログマネージャ変数、イメージ、スポットマーカーを含めることができます。

さらに、独自にデザインしたレイアウトを使用し、ページにデータを配置する、自由形式 (表を使用しない) レポートを作成するためにページ見出しおよび脚注構文を使用できます。

TABLE リクエストでは、複数ページの見出しおよび脚注を指定することができます。見出しまたは脚注のそれぞれについて、取得されるデータに WHEN 句を使用して、レポート出力に見出しまたは脚注を表示するかどうかを指定することができます。

見出しの場合、WHEN 句のデータと見出しに表示されるデータフィールド値は、ページの最初の行に表示されます。脚注の場合、WHEN 句のデータと脚注に表示されるデータフィールド値は、ページの最後の行に表示されます。

CONDITION スタイルシート属性を使用して、特定の WHEN 句を識別することができ、見出しと脚注のそれぞれに、個別にスタイルを設定することができます。詳細は、見出し、脚注、タイトル、FML テキストの選択を参照してください。

ページ見出しの作成

リクエストに次の構文を含めます。見出しおよび脚注の各行は、二重引用符 (") で開始、終了する必要があります。

```
[HEADING [CENTER]]
 "content ... "
["content ... "]
 を使用します。
 を使用します。
 を使用します。
["content ... "]
```

説明

HEADING

最初の表示コマンド (例、PRINT または SUM) の前にテキストを配置する場合、オプションです。それ以外は、ページ見出しとしてテキストを識別するために必要です。

CENTER

レポートデータの表示幅でページ見出しを中央に揃えるオプションのコマンドです。詳細は、レガシーフォーマットを使用したページ見出しまたは脚注の中央揃えを参照してください。

content

見出しまたは脚注の内容には、次の要素を二重引用符 (") で囲んで含めることができます。最終の引用符が省略された場合、リクエストの後に続くすべての行は、ページ見出しの一部分として扱われます。

text

ページ見出しのテキストです。複数行のテキストを含めることができます。

テキストは、テキストのみの行で開始する必要があり、HEADING コマンドが続きます。

テキストは、変数およびスポットマーカーと組み合わせることができます。

variable

次の 1 つまたは組み合わせです。

`Fields` - 実データソースフィールド、マスターファイルまたはリクエスト内で DEFINE コマンドを使用して作成した一時項目 (DEFINE) 、リクエスト内で COMPUTE コマンドを使用して作成した一時項目 (COMPUTE) 、または TABPAGENO などのシステムフィールドです。データソースフィールドは、特定の演算接頭語で修飾することができます。

```
Dialogue Manager variables.
```

`Images` - 見出しまたは脚注にイメージを含めることができます。

詳細は、見出しまたは脚注内への要素の追加を参照してください。

スポットマーカー

項目の配置、フォーマット設定項目の識別が可能です。

`<+0>` - フォーマット設定を目的として、見出しまたは脚注を項目に分割します。

`</n` - ブランク行を指定します。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。

`<-n` - 行に次の文字を配置します。詳細は、スポットマーカーを使用した配置調整を参照してください。

`<0X` - 見出しまたは脚注指定を、リクエストの次の行に継続します。

注意：終了スポットマーカーの直後に開始スポットマーカーを追加すると (><)、この 2 つのスポットマーカー間にブランクが 1 つ挿入されます (> <)。フォーマット設定を適用する際は、このことを考慮する必要があります。

ブランク行

すべてのテキスト、変数、スポットマーカーを省略する場合、見出しや脚注がブランクになります (例、" ")。これを使用して、ブランク行を挿入することができます。ブランク行を挿入するには、</1 などの縦のスポットマーカーを使用することもできます。

単一行のページ見出しの作成

次のリクエストでは、ページ見出しテキストが表示コマンド PRINT を優先することから、HEADING コマンドは省略されています。ページ見出しには、テキストおよび埋め込みフィールドが含まれます。

```
TABLE FILE EMPLOYEE
"ACCOUNT REPORT FOR DEPARTMENT <DEPARTMENT"
PRINT CURR_SAL BY DEPARTMENT BY HIGHEST BANK_ACCT
BY EMP_ID
ON DEPARTMENT PAGE-BREAK
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

出力には、複数ページの HTML レポートにページ見出しが表示されています。ページ見出しは、部門を識別するためにそれぞれのページに表示されます。埋め込みフィールド値についての詳細は、見出しまたは脚注へのフィールド値の追加を参照してください。データの最初のページは、MIS 部門に適用されます。

ACCOUNT REPORT FOR DEPARTMENT MIS

DEPARTMENT	BANK_ACCT	EMP_ID	CURR_SAL
MIS	163800144	818692173	$27,062.00
	122850108	326179357	$21,780.00
	40950036	117593129	$18,480.00
		112847612	$13,200.00
		219984371	$18,480.00
		543729165	$9,000.00

データの 2ページ目は、PRODUCTION 部門に適用されます。

ACCOUNT REPORT FOR DEPARTMENT PRODUCTION

DEPARTMENT	BANK_ACCT	EMP_ID	CURR_SAL
PRODUCTION	819000702	123764317	$26,862.00
	136500120	451123478	$16,100.00
	160633	119329144	$29,700.00
		071382660	$11,000.00
		119265415	$9,500.00
		126724188	$21,120.00

ページ脚注の作成

リクエストに次の構文を含めます。見出しおよび脚注の各行は、二重引用符 (") で開始、終了する必要があります。

```
FOOTING [CENTER] [BOTTOM]
  "content ... "
["content ... "]
 を使用します。
 を使用します。
 を使用します。
["content ... "]
```

説明

FOOTING

ページ脚注として内容を識別するために必要なコマンドです。

CENTER

レポートデータの表示幅でページ脚注を中央に揃えるオプションのコマンドです。CENTER についての詳細は、レガシーフォーマットを使用したページ見出しまたは脚注の中央揃えを参照してください。

BOTTOM

ページの最下部に脚注を配置するオプションのコマンドです。BOTTOM を省略した場合、ページ脚注はレポートデータの下から 2 行目に表示されます。

BOTTOM についての詳細は、ページ下部へのページ脚注の配置を参照してください。

content

見出しまたは脚注の内容には、次の要素を二重引用符 (") で囲んで含めることができます。終了用引用符が省略された場合、<0X スポットマーカーを使用しない限り、リクエストの後に続くすべての行は、レポート脚注の一部分として扱われます。

text

ページ脚注のテキストです。複数行のテキストを含めることができます。

テキストは、テキストのみの行で開始する必要があり、FOOTING コマンドが続きます。

テキストは、変数およびスポットマーカーと組み合わせることができます。

variable

次の 1 つまたは組み合わせです。

`Fields` - 実データソースフィールド、マスターファイルまたはリクエスト内で DEFINE コマンドを使用して作成した一時項目 (DEFINE) 、リクエスト内で COMPUTE コマンドを使用して作成した一時項目 (COMPUTE) 、または TABPAGENO などのシステムフィールドです。データソースフィールドは、特定の演算接頭語で修飾することができます。

```
Dialogue Manager variables.
```

`Images` - 見出しまたは脚注にイメージを含めることができます。

詳細は、見出しまたは脚注内への要素の追加を参照してください。

スポットマーカー

項目の配置、フォーマット設定項目の識別が可能です。

`<+0>` - フォーマット設定を目的として、見出しまたは脚注を項目に分割します。

`</n` - ブランク行を指定します。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。

`<-n` - 行に次の文字を配置します。詳細は、スポットマーカーを使用した配置調整を参照してください。

`<0X` - 見出しまたは脚注指定を、リクエストの次の行に継続します。

注意：終了スポットマーカーの直後に開始スポットマーカーを追加すると (><)、この 2 つのスポットマーカー間にブランクが 1 つ挿入されます (> <)。フォーマット設定を適用する際は、このことを考慮する必要があります。

ブランク行

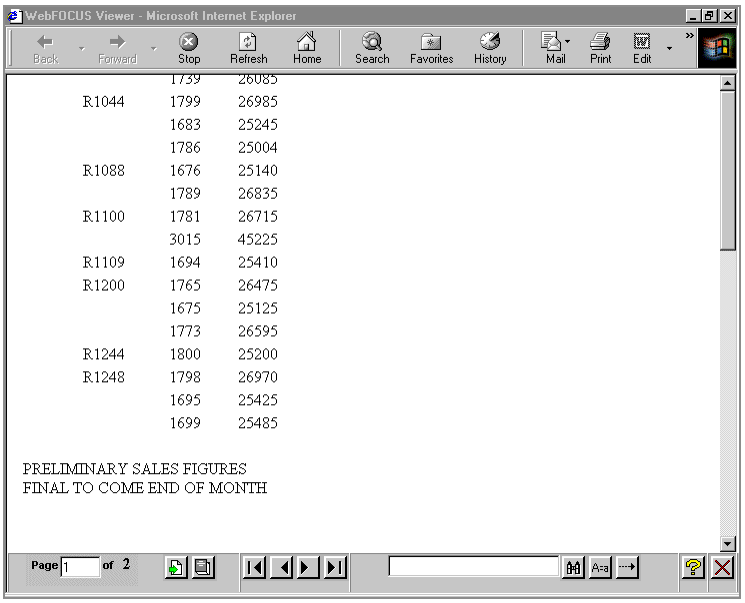
すべてのテキスト、変数、スポットマーカーを省略する場合、見出しや脚注がブランクになります (例、" ")。これを使用して、ブランク行を挿入することができます。ブランク行を挿入するには、</1 などの縦のスポットマーカーを使用することもできます。

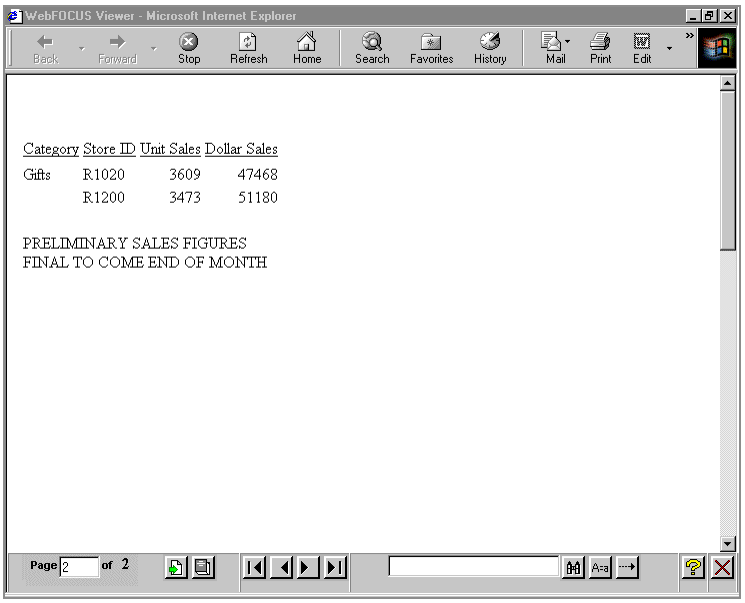
複数行のページ脚注の作成

次のリクエストは、2 行のページ脚注を作成します。

```
TABLE FILE GGSALES
PRINT UNITS DOLLARS
BY CATEGORY BY STCD
WHERE TOTAL DOLLARS GE 25000
FOOTING
"PRELIMINARY SALES FIGURES"
"FINAL TO COME END OF MONTH"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET WEBVIEWER ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

出力には、Web ビューアレポートにページ脚注が表示されています。ページ脚注は、レポートのそれぞれのページに表示されます。





ヒント：ナビゲーションの矢印が表示されない場合、最大化ボタンをクリックします。

WHEN 句による見出脚注の指定

```
{HEADING [CENTER]|FOOTING}
"text_and_data1"
   .
```

```
    .
    .
"text_and_datan"
WHEN expression
```

説明

text_and_data1, text_and_datan

各見出し脚注行のテキストおよびデータです。

expression

TRUE または FALSE (1 または 0) を設定する式です。値が TRUE に設定される場合、見出しまたは脚注が表示されます。値が FALSE に設定される場合、見出しまたは脚注は表示されません。

複数見出し使用上の注意

- HEADING CENTER および FOOTING CENTER は、指定された見出しおよび脚注にのみ適用されます。
- リクエストには、合計で 120 個の見出し、120 個の脚注、120 個の中間見出し、120 個の中間脚注を含めることができます。
- 脚注で BOTTOM オプションを使用すると、それに続く脚注もすべてページの下部に配置されます。

HTML、AHTML、XLSX での 見出し、脚注、列タイトルの固定

レポート出力のコンテンツをスクロールする際に、見出し、フィールドタイトル、脚注を画面に固定した状態でデータをスクロールしたい場合があります。

スタイルシート属性を使用すると、HTML、AHTML、XLSX レポート出力でスクロール可能な領域を画面に固定することができます。

HFREEZE で HTMLARCHIVE を使用する際の注意

WebFOCUS インタラクティブレポート機能が正常に動作するには、WebFOCUS Client に接続して、この機能に必要なコンポーネントにアクセスする必要があります。

WebFOCUS Client へのアクセスが利用できない場合は、HTMLARCHIVE を使用して、ユーザ定義イメージが含まれた自己完結型 HTML ページを作成することができます。

ユーザ定義イメージが含まれた HTML ページを生成し、インタラクティブ動作を可能にするには、次のコマンドのいずれかを使用します。

```
SET HTMLEMBEDIMG=ON
SET HTMLARCHIVE=ON
```

次の構文を使用して、これらのファイルへのアクセスが可能なホストマシンを指定した BASEURL を定義します。

```
SET BASEURL=http://{hostname:portnumber}
```

SET BASEURL についての詳細は、ベース URL の指定を参照してください。

HTML レポートでのスクロール可能領域の作成

```
TYPE=REPORT,HFREEZE={OFF|ON|TOP|BOTTOM},[SCROLLHEIGHT={AUTO|nn[.n]}], $
```

説明

HFREEZE=OFF

見出し、フィールドタイトル、総合計、脚注を固定しません。デフォルト値は OFF です。

HFREEZE=ON

見出し、フィールドタイトル、総合計、脚注を固定します。

HFREEZE=TOP

見出しおよび列タイトルを固定します。

HFREEZE=BOTTOM

総合計および脚注を固定します。

SCROLLHEIGHT=AUTO

HFREEZE レポートの出力をレスポンシブ対応にし、出力コンテナのサイズが変更された場合に、出力コンテナの高さに自動的に収まるようにします。

AUTOFIT=ON に設定すると、出力コンテナのサイズが変更された場合に、レポートが出力コンテナのサイズ (縦横両方) に自動的に収まるようになります。

注意：AUTOFIT=ON 設定は、SCROLLHEIGHT 設定を上書きます。

nn[.n]

スクロール可能な領域の高さ (インチ) です。モバイルデバイス以外でのデフォルト設定は、4 インチです。

AHTML レポートでのスクロール可能領域の作成

```
TYPE=REPORT,HFREEZE={OFF|ON}, $
```

説明

HFREEZE=OFF

見出し、フィールドタイトル、総合計、脚注を固定しません。デフォルト値は OFF です。

HFREEZE=ON

見出し、フィールドタイトル、総合計、脚注を固定します。

XLSX レポートでのスクロール可能領域の作成

```
TYPE=REPORT,HFREEZE={OFF|ON}, $
```

説明

HFREEZE=OFF

見出しおよび列タイトルを固定しません。デフォルト値は OFF です。

HFREEZE=ON

見出しおよび列タイトルを固定します。

XLSX レポート出力で、列を固定することができます。これにより、右側にスクロールしてワークシートのデータを表示する際に、指定した列が画面上に保持されるため、コンテキストが失われることがありません。詳細は、XLSX レポートの列の固定を参照してください。

注意：ソートフィールドに基づいて複数のワークシートを表示する BYTOC オプションを使用した場合、ブック内の複数のワークシートに固定を適用できます。

HFREEZE でのブランクフィールドタイトルのサポート

HTML HFREEZE レポート機能では、ブランクのフィールドタイトルがサポートされます。縦方向の HFREEZE スクロールバーは、レポートデータの先頭行に合わせて位置が揃えられます。

HTML レポート出力の固定領域を使用する際の注意

レポートの見出しと脚注を固定することができますが、データ行のスクロールは HTML、AHTML、XLSX レポートのみで可能です。その他すべてのフォーマットでは、スタイルシート属性の HFREEZE は無視されます。レポートに ON TABLE SET HTMLCSS ON (デフォルト) 設定を含める必要があります。

次の HTML 機能を HFREEZE とともに使用することはできません。

- Internet Explorer を使用する場合、HFREEZE の固定エリア内で、中間見出しおよび中間脚注にイメージを配置することはできません。Mozilla Firefox、Google Chrome、Microsoft Edge を使用する場合、HFREEZE の固定エリア内で、中間見出しおよび中間脚注にイメージを配置することはできません。
- アコーディオンレポート
- WRAP スタイルシート属性
- カスタム HTML タグまたは JavaScript
- 複合レポート

AHTML レポート出力の固定領域を使用する際の注意

レポートの見出しと脚注を固定することができますが、データ行のスクロールは HTML、AHTML、XLSX レポートのみで可能です。その他すべてのフォーマットでは、スタイルシート属性の HFREEZE は無視されます。レポートに ON TABLE SET HTMLCSS ON (デフォルト) 設定を含める必要があります。

次の AHTML 機能を HFREEZE とともに使用することはできません。

- SET SUBTOTALS=ABOVE および SET SUBTOTALS=BELOW

  標準の AHTML レポートでは、中間合計はデータ行の下部に表示され、総合計行は固定フレームの下部に固定されます。この現象は、SET SUBTOTALS コマンドの有無に関わらず発生します。
- HFREEZE の固定エリア内で、中間見出しおよび中間脚注にイメージを配置することは

できません。

- アコーディオンレポート

XLSX レポート出力の固定領域を使用する際の注意

レポートの見出しと脚注を固定することができますが、データ行のスクロールは HTML、AHTML、XLSX レポートのみで可能です。その他すべてのフォーマットでは、スタイルシート属性の HFREEZE は無視されます。レポートに ON TABLE SET HTMLCSS ON (デフォルト) 設定を含める必要があります。

次の XLSX 機能を HFREEZE とともに使用することはできません。

- SET SUBTOTALS=ABOVE および SET SUBTOTALS=BELOW

  標準の AHTML レポートでは、中間合計はデータ行の下部に表示され、総合計行は固定フレームの下部に固定されます。この現象は、SET SUBTOTALS コマンドの有無に関わらず発生します。

- HFREEZE の固定エリア内で、中間見出しおよび中間脚注にイメージを配置することはできません。

ソート項目を指定している見出しおよび脚注の作成

ソート項目を指定している見出しは、ソート項目値が変わると表示されるテキストで、関連データのグループの開始を識別します。ソート項目を指定している脚注は、関連データのグループの終了を識別し、ソート項目値の変更に続くテキストです。

レポートの本文に表示されるソート項目を指定している見出しまたは脚注は、レポートの詳細部分の異なる領域の特定に役立ちます。ソート項目を指定している見出しまたは脚注は、テキスト、項目、ダイアログマネージャ変数、イメージ、スポットマーカーを含めることができます。

リクエスト内に WHEN 句を含めることにより、定義した条件を満たすデータ用にソート項目を指定している見出しまたは脚注として実装したメッセージを生成できます。条件付き

フォーマット設定についての詳細は、レポートフォーマット設定の管理を参照してください。

演算で中間合計の値を出すために RECAP コマンドを使用している場合、SUBFOOT コマンドを含んだフィールドの後に RECAP コマンドを続けることにより、デフォルト RECAP ラベルをより意味のあるソート項目を指定している脚注に交換できます。RECAP コマンドについての詳細は、合計と中間合計の追加を参照してください。

ソート項目を指定している脚注内に 1 つ以上のデータフィールドが埋め込まれている場合、デフォルトでは見出しおよび脚注内のデータフィールドは合計されることから、リクエストから表示コマンドを省略できます。ただし、リクエストが明示的に SUM コマンドを含み、表示フィールドがソート項目を指定している脚注内で指定されている場合、ソート項目を指定している脚注内のフィールドは、合計されます。同様に、見出しおよび脚注のその他のタイプから表示コマンドを省略できます。見出しのデータは、最初のソートグループから得られ、脚注のデータは、最後のソートグループから得られます。

デフォルト設定では、中間見出しおよび中間脚注の前にブランク行が自動的に生成されます。これらの自動的なブランク行を削除するには、SET DROPBLNKLINE=ON コマンドを発行します。

中間見出しおよび中間脚注の位置揃え

デフォルト設定では、SQUEEZE=ON に設定した場合、中間見出しおよび中間脚注の境界線および背景色に使用される右マージンは、すべての見出し、脚注、中間見出し、中間脚注の行の最大幅に基づいて定義されます。中間見出し行と中間脚注行の長さは、レポートのデータ行のサイズではなく、ページ見出しとページ脚注の長さと連動します。スタイルシートで ALIGN-BORDERS=BODY 属性を使用すると、中間見出しと中間脚注の位置を、PDF レポート出力の他の見出し要素ではなく、データ (レポート本文) の表示幅に合わせて揃えることができます。

ソート項目を指定している見出しの作成

それぞれの見出しおよび脚注の行は、行の連続スポットマーカー (<0x) を使用しない限り、二重引用符 (") で開始し二重引用符 (") で終了する必要があります。

```
BY fieldname SUBHEAD [NEWPAGE]
 "content ... "
["content ... "]
```

```
.
.
.
["content ... "]
[WHEN expression;]
BY fieldname
ON fieldname SUBHEAD [NEWPAGE]
 "content ... "
["content ... "]
.
.
.
["content ... "]
[WHEN expression;]
```

または

```
BY fieldname
ON fieldname SUBHEAD [NEWPAGE]
 "content ... "
["content ... "]
.
.
.
["content ... "]
[WHEN expression;]
```

説明

fieldname

前に見出しのテキストを表示するソート項目です。

content

見出しまたは脚注の内容には、次の要素を二重引用符 (") で囲んで含めることができます。終了用引用符が省略された場合、リクエストの後に続くすべての行は、レポート見出しの一部分として扱われます。

text

ソート項目を指定している見出しのテキストです。複数行のテキストを含めることができます。

テキストは、SUBHEAD コマンドに続けて、テキストのみの行で開始する必要があります。

テキストは、変数およびスポットマーカーと組み合わせることができます。

variable

次の 1 つまたは組み合わせです。

`Fields` - 実データソースフィールド、マスターファイルまたはリクエスト内で DEFINE コマンドを使用して作成した一時項目 (DEFINE) 、リクエスト内で COMPUTE コマンドを使用して作成した一時項目 (COMPUTE) 、または TABPAGENO などのシステムフィールドです。データソースフィールドは、特定の演算接頭語で修飾することができます。

```
Dialogue Manager variables.
```

`Images` - 見出しまたは脚注にイメージを含めることができます。

詳細は、見出しまたは脚注内への要素の追加を参照してください。

スポットマーカー

項目の配置、フォーマット設定項目の識別が可能です。

`<+0>` - フォーマット設定を目的として、見出しまたは脚注を項目に分割します。

`</n` - ブランク行を指定します。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。

`<-n` - 行に次の文字を配置します。詳細は、スポットマーカーを使用した配置調整を参照してください。

`<0X` - 見出しまたは脚注指定を、リクエストの次の行に継続します。

注意：終了スポットマーカーの直後に開始スポットマーカーを追加すると (><)、この 2 つのスポットマーカー間にブランクが 1 つ挿入されます (> <)。フォーマット設定を適用する際は、このことを考慮する必要があります。

WHEN expression

論理式により決定するように、ソート項目を指定している見出しが表示される条件を指定します。テキストの後に続いて行の上に WHEN 句を配置する必要があります。

条件付きフォーマット設定についての詳細は、レポートフォーマット設定の管理を参照してください。詳細は、式の使用を参照してください。

ブランク行

すべてのテキスト、変数、スポットマーカーを省略する場合、見出しや脚注がブランクになります (例、" ")。これを使用して、ブランク行を挿入することができます。ブランク行を挿入するには、</1 などの縦のスポットマーカーを使用することもできます。

NEWPAGE

ソート項目を指定している見出しの後に新しいページを挿入します。フィールドタイトルは、各ページに表示されます。

NEWPAGE オプションは、PDF レポートで使用することができます。HTML レポートでは、新しいページではなく、ブランクが追加されます。

製品説明変更時のソート項目を指定している見出しの作成

次のリクエストは、製品の説明が変更されるたびに、ソート項目を指定している見出しを表示します。ソート項目を指定している見出しには、テキストおよび埋め込みフィールドが含まれます。

```
TABLE FILE GGPRODS
PRINT PACKAGE_TYPE AND UNIT_PRICE
WHERE UNIT_PRICE GT 50
BY PRODUCT_DESCRIPTION NOPRINT BY PRODUCT_ID
ON PRODUCT_DESCRIPTION SUBHEAD
"Summary for <PRODUCT_DESCRIPTION"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

ソート項目を指定している見出しは、次の行のデータが適用する製品を識別します。

Product Code	Package	Unit Price
Summary for Coffee Grinder		
G 110	Case	125.00
Summary for Coffee Pot		
G121	Case	140.00
Summary for French Roast		
B142	Pounds	81.00
Summary for Hazelnut		
B141	Pounds	58.00
Summary for Kona		
B144	Pounds	76.00
Summary for Thermos		
G104	Case	96.00

埋め込みフィールド値についての詳細は、見出しまたは脚注内への要素の追加を参照してください。

条件付きソート項目を指定している見出しの作成

次のリクエストは、売上が $17,000,000 を下回った場合にのみ、分類のソート項目を指定している見出しを表示します。

```
TABLE FILE GGSALES
SUM DOLLARS
BY CATEGORY SUBHEAD
"<CATEGORY ALERT:SALES FALL BELOW $17,000,000"
WHEN DOLLARS LT 17000000;
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

「Gifts」という分類の売上が指定した額を下回ると、ソート項目を指定している見出しに警告が表示されます。その他の分類は、ソート項目を指定している見出しを表示しません。

Category	Dollar Sales
Coffee	17231455
Food	17229333
Gifts ALERT: SALES FALL BELOW $17,000,000	
Gifts	11695502

埋め込みフィールド値についての詳細は、見出しまたは脚注へのフィールド値の追加を参照してください。

ソート項目を指定している脚注の作成

見出しおよび脚注の各行は、二重引用符 (") で開始、終了する必要があります。

単一ソートフィールド用に次の構文を使用します。

```
BY fieldname SUBFOOT [WITHIN] [MULTILINES] [NEWPAGE]
 "content ... "
["content ... "]
.
.
.
["content ... "]
[WHEN expression;]
BY fieldname
ON fieldname SUBFOOT [WITHIN] [MULTILINES] [NEWPAGE]
 "content ... "
["content ... "]
.
.
.
["content ... "]
[WHEN expression;]
```

複数ソートフィールド用に次の構文を使用します。

```
BY fieldname
ON fieldname SUBFOOT [MULTILINES] [NEWPAGE]
 "content ... "
["content ... "]
.
.
```

```
.
["content ... "]
[WHEN expression;]
```

説明

fieldname

後に脚注テキストを表示するソートフィールドです。

WITHIN

このオプションを使用した場合、中間脚注内のフィールドがそれぞれのフィールド値の中で計算されます。このオプションを使用しない場合、中間脚注内のフィールドの値は、中間脚注の上に表示されたレポート出力の最終行から取得されます。

MULTILINES

ソートフィールド値のデータが 1 行のみの場合、脚注を非表示にします。MULTI-LINES は MULTILINES の同義語です。

content

見出しまたは脚注の内容には、次の要素を二重引用符 (") で囲んで含めることができます。終わりの引用符が省略された場合、リクエストの後に続くすべての行は、ソート項目を指定している脚注の一部分として扱われます。

text

レポートの最初のページに表示するテキストです。複数行のテキストを含めることができます。

テキストは、SUBFOOT コマンドに続いてテキストのみの行で開始する必要があります。

テキストは、変数およびスポットマーカーと組み合わせることができます。

variable

次の 1 つまたは組み合わせです。

`Fields` - 実データソースフィールド、マスターファイルまたはリクエスト内で DEFINE コマンドを使用して作成した一時項目 (DEFINE)、リクエスト内で COMPUTE コマンドを使用して作成した一時項目 (COMPUTE)、または TABPAGENO などのシステムフィールドです。データソースフィールドは、特定の演算接頭語で修飾することができます。

```
Dialogue Manager variables.
```

Images - 見出しまたは脚注にイメージを含めることができます。

詳細は、見出しまたは脚注内への要素の追加を参照してください。

スポットマーカー

項目の配置、フォーマット設定項目の識別が可能です。

<+0> - フォーマット設定を目的として、見出しまたは脚注を項目に分割します。

</n - ブランク行を指定します。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。

<-n - 行に次の文字を配置します。詳細は、スポットマーカーを使用した配置調整を参照してください。

<0X - 見出しまたは脚注指定を、リクエストの次の行に継続します。

注意：終了スポットマーカーの直後に開始スポットマーカーを追加すると (><)、この 2 つのスポットマーカー間にブランクが 1 つ挿入されます (> <)。フォーマット設定を適用する際は、このことを考慮する必要があります。

WHEN expression

論理式により決定するようにソート項目を指定している脚注が表示される条件を指定します。テキストの後に続いて行の上に WHEN 句を配置する必要があります。

条件付きフォーマット設定についての詳細は、レポートフォーマット設定の管理を参照してください。詳細は、式の使用を参照してください。

ブランク行

すべてのテキスト、変数、スポットマーカーを省略する場合、見出しや脚注がブランクになります (例、" ")。これを使用して、ブランク行を挿入することができます。ブランク行を挿入するには、</1 などの縦のスポットマーカーを使用することもできます。

NEWPAGE

ソート項目を指定している脚注の前に新しいページを挿入します。

製品説明変更時のソート項目を指定している脚注の作成

次のリクエストは、製品の説明が変更されるたびに、ソート項目を指定している脚注を表示します。

```
TABLE FILE GGPRODS
PRINT PACKAGE_TYPE AND UNIT_PRICE
WHERE UNIT_PRICE GT 50
BY PRODUCT_DESCRIPTION NOPRINT BY PRODUCT_ID
ON PRODUCT_DESCRIPTION SUBFOOT
"Summary for <PRODUCT_DESCRIPTION"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

Product Code	Package	Unit Price
G110	Case	125.00
Summary for Coffee Grinder		
G121	Case	140.00
Summary for Coffee Pot		
B142	Pounds	81.00
Summary for French Roast		
B141	Pounds	58.00
Summary for Hazelnut		
B144	Pounds	76.00
Summary for Kona		
G104	Case	96.00
Summary for Thermos		

埋め込みフィールド値についての詳細は、見出しまたは脚注へのフィールド値の追加を参照してください。

複数ソートオプションを持つ条件付きソート項目を指定している脚注の作成

次のレポートは、Century Corporation の注文、注文日、注文合計を一覧表示します。合計が $200,000 より大きい注文と $200,000 より小さい注文を区別するために、条件付きソート項目を脚注に使用します。

ここでは、1 つのソート句 (ON ORDER_NUM) が、複数のソートオプション (2 つの異なる SUBFOOT 句) を指定し、それぞれのオプションに個別の WHEN 句が指定されています。

```
TABLE FILE CENTORD
HEADING
"Order Revenue"
" "
SUM ORDER_DATE LINEPRICE AS 'Order,Total:'
BY HIGHEST 5 ORDER_NUM
ON ORDER_NUM
   SUBFOOT
      "--- Order total is less than $200,000 ---"
      " "
      WHEN LINEPRICE LT 200000;
   SUBFOOT
      "+++ Order total is greater than or equal to $200,000 +++"
      " "
      WHEN LINEPRICE GE 200000;
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

次のレポートが表示されます。

Order Revenue

Order Number:	Date Of Order:	Order Total:
94710	2001/01/02	$406,964.24
+++ Order total is greater than or equal to $200,000 +++		
94680	2001/01/02	$421,916.60
+++ Order total is greater than or equal to $200,000 +++		
94670	2001/01/02	$513,868.76
+++ Order total is greater than or equal to $200,000 +++		
94550	2001/01/02	$496,323.64
+++ Order total is greater than or equal to $200,000 +++		
94530	2001/01/02	$3,472.41
--- Order total is less than $200,000 ---		

ソート項目を指定している脚注の非表示

次のリクエストは、1 行のデータのみを持つ製品のソート項目の脚注を非表示にします。これは 01/01/96 に一度だけ注文があった製品です。

```
TABLE FILE GGORDER
PRINT QUANTITY
BY PRODUCT_CODE NOPRINT BY PRODUCT_DESCRIPTION
WHERE ORDER_DATE EQ '01/01/96'
WHERE STORE_CODE EQ 'R1019'
WHERE PRODUCT_DESCRIPTION EQ 'Hazelnut' OR 'Biscotti' OR 'Croissant'
ON PRODUCT_CODE SUBFOOT MULTILINES
"<PRODUCT_DESCRIPTION has multiple orders."
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果内で Biscotti のソート項目を指定している脚注は非表示になります。

Product	Ordered Units
Hazelnut	300
	117
Hazelnut has multiple orders.	
Biscotti	399
Croissant	93
	433
Croissant has multiple orders.	

ソート項目を指定している脚注によるデフォルト RECAP ラベルの置換

次のリクエスト内で、フィールド DEPARTMENT の SUBFOOT コマンドは、そのフィールドの RECAP コマンドの後に続きます。RECAP コマンドにより、中間合計値が生成されます。

```
TABLE FILE SHORT
SUM BALANCE AS 'Dollars' ENGLAND_POUND AS 'Sterling'
BY REGION
WHERE REGION EQ 'FAR EAST' OR 'CENTRAL AMERICA' OR 'WESTERN EUROPE';
BY COUNTRY NOPRINT
RECAP EURO/D16=BALANCE * 1.03;
SUBFOOT
" "
"Balance of investments for <COUNTRY> in Euros is <EURO>."
" "
END
```

ソート項目を指定している脚注テキスト (例、「Balance of investments for FRANCE in Euros is 87,336,971.」) が、RECAP 値のデフォルトラベル (例、「** EURO 87,336,971」) の代わりに表示されます。

Region	Dollars	Sterling
CENTRAL AMERICA	727,810	443,964
Balance of investments for GUATEMALA in Euros is 749,644 .		
	39,472,857	24,078,443
Balance of investments for HONDURAS in Euros is 40,657,043 .		
FAR EAST	88,068,897	53,722,027
Balance of investments for HONG KONG in Euros is 90,710,964 .		
	114,591,945	69,901,086
Balance of investments for JAPAN in Euros is 118,029,703 .		
WESTERN EUROPE	125,757,198	76,711,891
Balance of investments for ENGLAND in Euros is 129,529,914 .		
	84,793,176	51,723,837
Balance of investments for FRANCE in Euros is 87,336,971 .		

ソート項目を指定している脚注内の表示コマンドの省略

次のリクエストは、レポートをソート項目の脚注として作成します。ソート項目を指定している脚注にデータフィールド DEPARTMENT および SALARY が含まれるため、表示コマンドは必要ありません。デフォルト設定では、フィールド SALARY は、ソート項目を指定している脚注内で合計されます。

```
TABLE FILE EMPLOYEE
BY DEPARTMENT NOPRINT SUBFOOT
"<DEPARTMENT DEPARTMENT TOTAL SALARY IS <SALARY"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
```

```
ENDSTYLE
END
```

出力結果は次のとおりです。

```
MIS DEPARTMENT TOTAL SALARY IS $160,177.00

PRODUCTION DEPARTMENT TOTAL SALARY IS $172,752.00
```

中間脚注の使用上の注意

- SUBFOOT WITHIN は、ソート区切り内で接頭語の付いたフィールドが単一値を生成する場合に使用すると便利です (例、AVE、MIN、MAX)。PCT. または APCT. を使用すると、ソートグループの最後の値のみが表示されます。
- SUBFOOT WITHIN "<prefix.fieldname" を使用した場合と SUBTOTAL 接頭語を使用した場合とでは結果が異なります。SUBFOOT WITHIN を使用した場合は、表示フィールドを作成して、元の入力レコードで演算を実行します。SUBTOTAL に接頭語を使用した場合は、内部マトリックスで演算を実行します。そのため、AVE. は SUM の平均になる場合もあれば、表示フィールドに AVE. 接頭語が付いていれば平均の平均になる場合もあります。SUBFOOT WITHIN "<AVE.field" は、全体の平均を生成します。
- 演算接頭語は、WITHIN 句の文字フィールドではサポートされません。
- ST. 演算接頭語は、SUBFOOT WITHIN 句ではサポートされません。

見出しまたは脚注内への要素の追加

次の要素を追加して、見出しおよび脚注をカスタマイズすることができます。

- フィールド値。詳細は、見出しまたは脚注へのフィールド値の追加および 見出しまたは脚注へのテキストフィールドの追加を参照してください。
- ページ番号。
- ダイアログマネージャ変数。
- イメージ。

見出しまたは脚注へのフィールド値の追加

見出しまたは脚注テキスト内にフィールド名を含めることができます。リクエストを実行すると、出力結果にフィールド値が表示されます。結果には、カスタマイズされた見出しまたは脚注に、ユーザごとに異なるデータ識別子が表示されます。

この方法は任意のレポートで有効ですが、自由形式レポートを作成する場合に重要です。

複数表示とソートフィールドのセット、レポート見出しまたは脚注内のフィールド、ページ見出しまたは脚注のリクエストは、あたかも最初の表示コマンドの目的語であるかのように評価されます。ソート項目を指定している見出しまたは脚注内のフィールドは、参照される最初の表示コマンドの一部として評価されます。フィールドが参照されていない場合、そのフィールドは最後の表示コマンドの一部として評価されます。

見出しまたは脚注内のフィールド値を取得するために、演算接頭語を使用できます。ただし、DST.、MDE.、MDN. 演算接頭語は、ACROSS 句が指定されたリクエスト、および複数の表示コマンドが使用されたリクエストの見出しおよび脚注ではサポートされません。演算接頭語による演算のリストは、レポートデータの表示を参照してください。

2 つの演算子は、ソート項目を指定している脚注で使用できます。

- ST. - レポート内のソート区切りで数値フィールドの中間合計値を作成します。
- CT. - 数値フィールドの累計を作成します。

見出しまたは脚注へのフィールド値の追加

```
<[prefix_operator]fieldname<fieldname[>]
```

または

```
<fieldname[>]
```

説明

<fieldname

見出しまたは脚注にフィールド値を設定し、SET STYLEMODE 値すべての文字フィールドの末尾のブランクを非表示にします。

<fieldname>

見出しまたは脚注にフィールド値を設定し、SET STYLEMODE = FIXED の場合には、文字フィールド内の末尾のブランクを保持します。SET STYLEMODE のその他すべての値の末尾のブランクを非表示にします。PDF 出力結果は、STYLEMODE の設定に関わらず末尾のブランクを保持します。

prefix_operator

フィールドの値で直接、計算を行います。演算接頭語は単一フィールドに適用され、そのフィールドのみに影響を与えます。

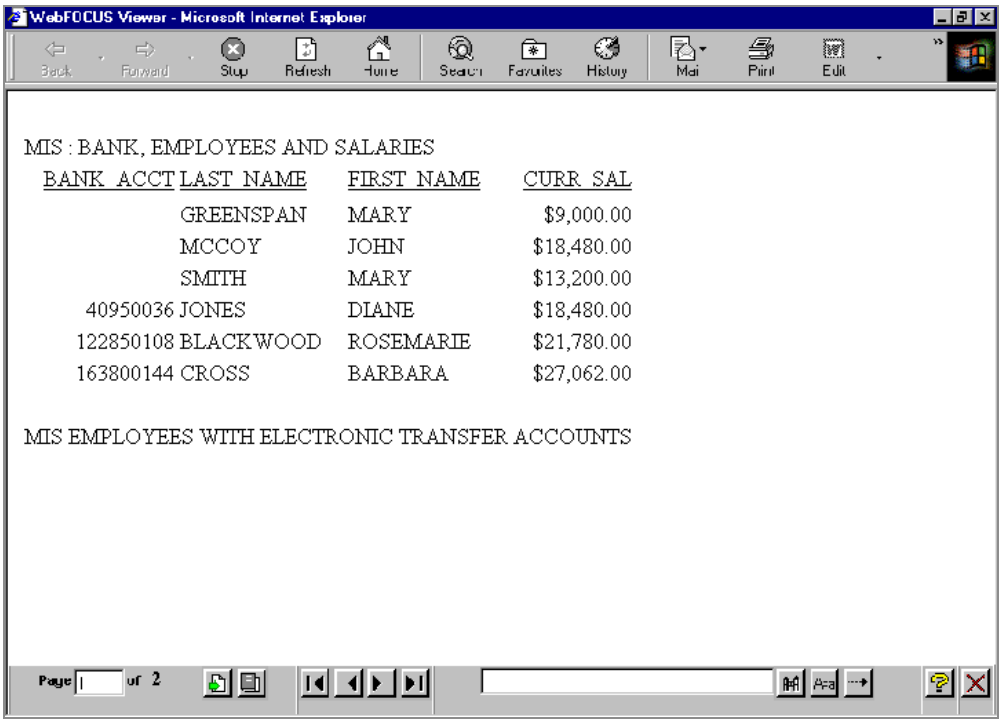
注意：見出しまたは脚注でキャレット記号 (<) をテキストとして表示するには、連続する 2 つのキャレット記号 (<<) を使用します。

ページ見出しおよび脚注内への部門名の追加

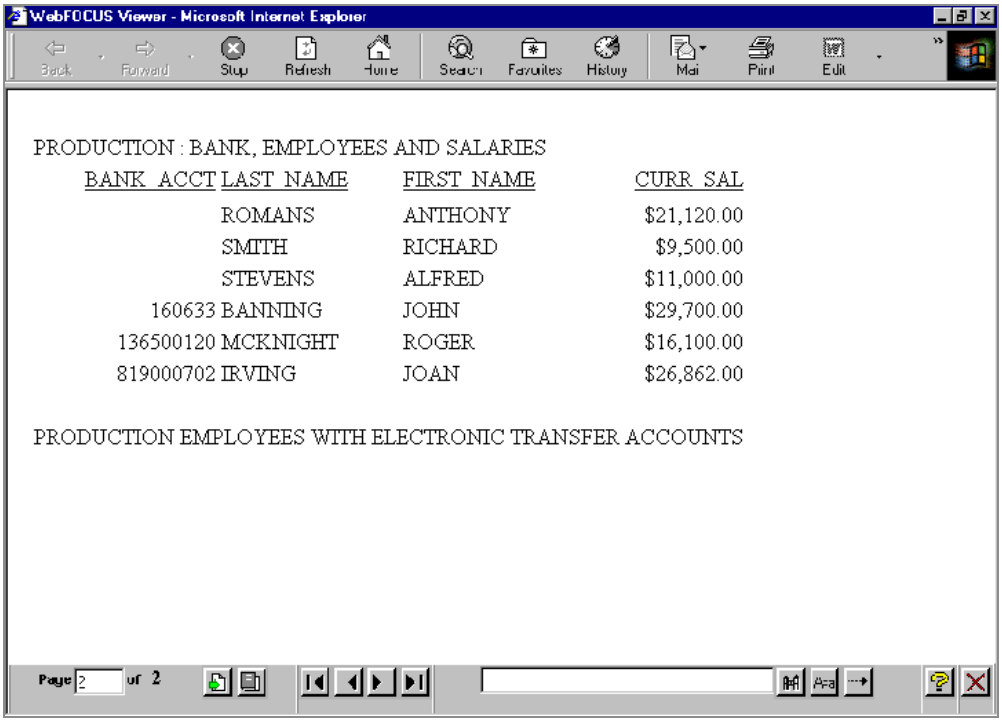
次のリクエストは、ページ見出しおよび脚注テキストの両方にフィールド名 DEPARTMENT を含めます。ページ見出しテキストは、PRINT コマンドの前に記述されるため、HEADING コマンドは、このリクエストでは必要ありません。

```
TABLE FILE EMPLOYEE
"<DEPARTMENT :BANK, EMPLOYEES AND SALARIES"
PRINT CURR_SAL
BY DEPARTMENT NOPRINT BY BANK_ACCT
BY LAST_NAME BY FIRST_NAME
ON DEPARTMENT PAGE-BREAK
FOOTING
"<DEPARTMENT EMPLOYEES WITH ELECTRONIC TRANSFER ACCOUNTS"
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET WEBVIEWER ON
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力には、Web ビューアレポートが表示されています。出力結果の最初のページの見出し、および脚注内の DEPARTMENT の値は、MIS です。



出力結果の 2 ページ目の、DEPARTMENT の値は、PRODUCTION です。



注意：ナビゲーションの矢印が表示されない場合、最大化ボタンをクリックします。

見出しでのキャレット記号 (<) の表示

次のリクエストは、REVENUE_US と COGS_US の差を計算し、その差が 100,000 より小さい行のみを表示します。

```
TABLE FILE WF_RETAIL_LITE
HEADING CENTER
" Difference <<  100,000"
" "
SUM COGS_US REVENUE_US
COMPUTE Difference/D20.2 = REVENUE_US - COGS_US;
BY PRODUCT_CATEGORY
WHERE TOTAL Difference LT 100000
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

下図のように、見出しに「Difference < 100,000」が表示されます。

Difference < 100,000

Product Category	Cost of Goods	Revenue	Difference
Computers	$109,281.00	$179,761.46	70,480.46
Televisions	$227,820.00	$286,160.68	58,340.68
Video Production	$180,540.00	$259,179.84	78,639.84

文字フィールド内の末尾のブランクの保持

末尾のブランクは、標準 HTML 出力結果内で保持されません。出力タイプが HTML の場合、デフォルト設定により STYLEMODE は FULL に設定されます。文字フィールド DEPARTMENT 内で末尾のブランクを保持するには、STYLEMODE の設定は、リクエスト内で FIXED に変更されており、区切り文字の < および > は、ソート項目を指定している脚注テキスト内のフィールド名の周囲に含まれています。

```
SET STYLEMODE = FIXED
TABLE FILE EMPLOYEE
SUM SALARY
```

```
BY DEPARTMENT SUBFOOT
"<DEPARTMENT> DEPARTMENT TOTAL SALARY IS <SALARY"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

DEPARTMENT の値は、MIS および PRODUCTION としてソート項目を指定している脚注内に表示されます。

```
DEPARTMENT            SALARY
----------            ------
MIS                $160,177.00
MIS          DEPARTMENT TOTAL SALARY IS     $160,177.00
PRODUCTION         $172,752.00
PRODUCTION   DEPARTMENT TOTAL SALARY IS     $172,752.00
```

注意：SET STYLEMODE=FIXED を使用すると、ブラウザでそのレポートの HTML フォーマット設定が無効になります。レポート結果は、色やその他の Web 機能が設定されない固定フォントで表示されます。

ページ見出し内の演算接頭語 TOT の使用

次のリクエストは、3 つのフィールドの総合計を生成するために、演算接頭語 TOT を使用します。

```
DEFINE FILE SALES
ACTUAL_SALES/D8.2 = UNIT_SOLD - RETURNS;
SALES/F5.1 = 100 * ACTUAL_SALES / UNIT_SOLD;
END
TABLE FILE SALES
"SUMMARY OF ACTUAL SALES"
"UNITS SOLD <TOT.UNIT_SOLD"
"RETURNS <TOT.RETURNS"
"TOTAL SOLD <TOT.ACTUAL_SALES"
" "
"BREAKDOWN BY PRODUCT"
PRINT UNIT_SOLD AND RETURNS AND ACTUAL_SALES
BY PROD_CODE
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT PDF
```

```
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

合計は、ページ見出し内に表示されます。

```
SUMMARY OF ACTUAL SALES
UNITS SOLD   645
RETURNS  58
TOTAL SOLD    587.00

BREAKDOWN BY PRODUCT
PROD_CODE  UNIT_SOLD  RETURNS  ACTUAL_SALES

B10                60       10         50.00
                   30        2         28.00
                   13        1         12.00
B12                40        3         37.00
                   29        1         28.00
B17                29        2         27.00
                   20        2         18.00
B20                15        0         15.00
                   25        1         24.00
C13                25        3         22.00
C17                12        0         12.00
C7                 45        5         40.00
                   40        0         40.00
D12                27        0         27.00
                   20        3         17.00
E1                 30        4         26.00
E2                 80        9         71.00
E3                 70        8         62.00
                   35        4         31.00
```

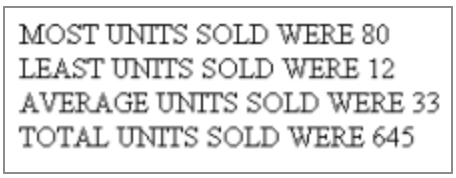
ページ見出し内の複数演算接頭語の使用

次のリクエストは、演算接頭語 MAX、MIN、AVE、TOT を使用します。ページ見出しテキストはデータフィールドを含むことから、表示コマンドは必要ありません。

```
TABLE FILE SALES
"MOST UNITS SOLD WERE <MAX.UNIT_SOLD"
"LEAST UNITS SOLD WERE <MIN.UNIT_SOLD"
"AVERAGE UNITS SOLD WERE <AVE.UNIT_SOLD"
"TOTAL UNITS SOLD WERE <TOT.UNIT_SOLD"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
```

```
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

演算接頭語は、ページ見出し内で概要データを生成します。



ソート項目を指定している脚注内の複数演算接頭語の使用

次のリクエストは、ソート項目を指定している脚注内で、演算接頭語 CNT および AVE を使用します。出力結果には、データフィールドは含まれません。すべてのデータは、ソート項目を指定している脚注自体に含まれます。

```
TABLE FILE EMPLOYEE
BY DEPARTMENT NOPRINT SUBFOOT
"NUMBER OF EMPLOYEES IN DEPARTMENT <DEPARTMENT = <CNT.LAST_NAME"
"WITH AVERAGE SALARY OF <AVE.CURR_SAL"
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

ソート項目を指定している脚注は、部門内の従業員数および平均給与の概要レポートです。

演算接頭語は、ページ見出し内で概要データを生成します。

```
NUMBER OF EMPLOYEES IN DEPARTMENT MIS = 6
WITH AVERAGE SALARY OF $18,000.33
NUMBER OF EMPLOYEES IN DEPARTMENT PRODUCTION = 6
WITH AVERAGE SALARY OF $19,047.00
```

見出しまたは脚注へのテキストフィールドの追加

見出しまたは脚注内に 1 つ以上のテキストフィールドを含めることができます。テキストフィールドのマスターファイルには、FORMAT=TX*n* 属性が含まれています。

見出しまたは脚注内のテキストフィールドの制限事項

- FML レポートにテキストフィールドを埋め込むことはできません。
- テキストフィールドは、マスターファイル内の FORMAT 属性の文字数で複数行に分割されるため、テキストフィールドにスタイルシート属性 WRAP を適用することはできません。

見出しまたは脚注へのテキストフィールドの追加

```
<TEXTFLD
```

ソート項目を指定している脚注へのテキストフィールドの追加

1. 次のリクエストを作成し、保存します。

   ```
   TABLE FILE TXTFLD
   ```

```
BY CATALOG SUBFOOT
"<TEXTFLD"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

2. リクエストを実行します。

   出力結果は次のとおりです。

```
CATALOGCOURSE 100
This course provides the junior programmer with
the skills needed to code simple reports
COURSE 200
This course provides the advanced programmer with
techniques helpful in developing complex
applications.
```

ヒント：この例の見出しは、1 つの埋め込みテキストフィールドを含むため、デフォルトの位置揃えで十分です。ただし、テキストを含めて埋め込みフィールドを作成または追加する際は、出力結果の項目に位置揃えを設定し、読みやすくすることもできます。

見出しまたは脚注へのページ番号の追加

見出しまたは脚注内に、システムが生成したページ番号を含めることができます。詳細は、レポートページのレイアウトを参照してください。

見出しまたは脚注へのダイアログマネージャ変数の追加

プロシジャの実行時に値が決定される変数を見出しまたは脚注に含めることができます。この方法は、プロシジャを実行するたびに異なる値を与えることにより、見出しまたは脚注をカスタマイズできるようにします。

変数は次の 2 つに分類されます。

- システム変数と統計変数は定義済みであり、これらの値はプロシジャから参照されるときにシステムが自動的に入力します。システム変数名および統計変数名は、アンパサンド (&) で始まります。たとえば、&DATE は、レポートの出力結果に現在の日付を生成します。
- ユーザ定義した値のローカル (&) およびグローバル (&&) 変数は、実行時に指定される必要があります。
    - ローカル変数の値は、プロシジャの実行中は保持されますが、そのプロシジャの処理が終了した時点で失われます。値は、同一変数名を含むその他の呼び出しプロシジャには渡されません。

      ローカル変数は、変数名の前に 1 つのアンパサンド (&) を付けて識別します。
    - グローバル変数は WebFOCUS Reporting Server に接続している間その値を保持し、1 つの呼び出しプロシジャの実行から次の呼び出しプロシジャに渡されます。

      リクエストが WebFOCUS Reporting Server に送信されるたびに新しいセッションが作成されるため、グローバル変数の値はリクエスト間で保持されません。そのため、1 つのリクエストで複数のプロシジャを呼び出す場合は、同一のグローバル変数の値を使用することができます。

      グローバル変数は、変数名の前に 2 つのアンパサンド (&&) を付けて識別します。

  注意：競合を回避するために、ローカルまたはグローバル変数に日付、IBI、WF で開始する名前を付けることはできません。これらの値で始まる変数名は、予約語です。

ダイアログマネージャ変数についての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

見出しまたは脚注内へのダイアログマネージャ変数の追加

```
&[&]variable
```

説明

&

ローカルダイアログマネージャ変数を開始します。

&&

グローバルダイアログマネージャ変数を開始します。

variable

実行時にシステムまたはユーザにより値が入力される変数です。

レポート見出しへの現在日付の追加

次のリクエストには、レポート見出しの 2 行目に青色で強調した今日の日付が含まれています。

```
TABLE FILE GGSALES
PRINT BUDDOLLARS DOLLARS
BY STCD
WHERE STCD EQ 'R1019'
ON TABLE SUBHEAD
"Sales Report for Store Code R1019"
"&DATE"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF,$
TYPE=TABHEADING, LINE=1, FONT='TIMES', SIZE=10, STYLE=BOLD, $
TYPE=TABHEADING, LINE=2, COLOR=BLUE, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Sales Report for Store Code R1019
06/07/02

Store ID	Budget Dollars	Dollar Sales
R1019	18560	25680
	21132	24010
	15880	22919
	24450	22605
	14556	18942
	13860	18915
	22806	18837
	19248	18535
	15120	18466
	15444	17784

注意：日付のフォーマットは、変更することができます。次はフォーマットの例です。

変数	表示フォーマット
&DATEtrYYMD	2002, 12月 11
&DATEMDYY	12/11/2002
&DATEtrMDYY	12月 11, 2002
&DATEQYY	Q4 2002

見出しまたは脚注内へのイメージの追加

スタイルシートにより、見出しまたは脚注にイメージを含めることができます。ロゴなどのイメージは、レポートに企業イメージの統一感や視覚的訴求力を与えます。

イメージの追加および配置についての詳細は、レポートページのレイアウトを参照してください。

見出しおよび脚注オブジェクトでの構文コンポーネントの表示

見出しおよび脚注オブジェクトに次の 1 つまたは複数の属性を追加することで、レポートまたはグラフリクエストで使用されている構文コンポーネントを見出しオブジェクトに自動的に表示することができます。

- **<REQUEST.FILTERS** リクエスト内の WHERE および IF 条件をリスト表示します。
- **<REQUEST.VERB_OBJECTS** リクエストで参照されている表示フィールドをリスト表示します。
- **<REQUEST.SORT_KEYS** リクエスト内のすべてのソートフィールドをリスト表示します。
- **<REQUEST.BYKEYS** リクエスト内のすべての BY ソートフィールドをリスト表示します。
- **<REQUEST.ACROSSKEYS** リクエスト内のすべての ACROSS ソートフィールドをリスト表示します。
- **<REQUEST.VERB_OBJECTS_CONTEXT** 各表示フィールドに使用されている表示コマンド構文をリスト表示します。
- **<REQUEST.SORT_KEYS_CONTEXT** リクエスト内のすべてのソート句をリスト表示します。

注意：見出し行がレポートまたはグラフコンテナの幅より長い場合は、構文コンポーネントが複数の行に分割されます。

レポート構文コンポーネントの表示

次のリクエストは、レポートリクエストの見出しに使用されている構文コンポーネントをすべて表示します。見出し項目と構文コンポーネントの間にスポットマーカー (<+0>) が使用され、スタイルシートでそれぞれに別のスタイルが設定されています。

```
TABLE FILE WF_RETAIL_LITE
HEADING
"Display Objects: <+0> <REQUEST.VERB_OBJECTS"
"Sort Fields: <+0> <REQUEST.SORT_KEYS"
"BY Fields: <+0> <REQUEST.BYKEYS"
```

```
"ACROSS Fields: <+0> <REQUEST.ACROSSKEYS"
"Filters: <+0> <REQUEST.FILTERS"
" "
"Display Commands: <+0> <REQUEST.VERB_OBJECTS_CONTEXT"
"Sort Phrases: <+0> <REQUEST.SORT_KEYS_CONTEXT
" "
" "
SUM COGS_US REVENUE_US
COMPUTE RATIO/D12.2=REVENUE_US/COGS_US;
BY PRODUCT_CATEGORY SUBTOTAL COGS_US
ACROSS BUSINESS_REGION ACROSS-TOTAL
WHERE BUSINESS_REGION NE 'Oceania'
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
TYPE=REPORT, COLOR = BLUE, SIZE=10, GRID=OFF,$
TYPE=HEADING, ITEM=1, OBJECT=TEXT, FONT=Courier, COLOR=BLUE, STYLE=BOLD,$
TYPE=HEADING, ITEM=2, FONT=Courier, COLOR=TEAL, STYLE=ITALIC,$
TYPE=TITLE, FONT=ARIAL, STYLE=BOLD, COLOR=NAVY,$
TYPE=ACROSSTITLE, FONT=ARIAL, STYLE=BOLD, COLOR=NAVY,$
TYPE=ACROSSVALUE, FONT=ARIAL, STYLE=ITALIC, COLOR=NAVY, SIZE=10,$
ENDSTYLE
END
```

出力結果は次のとおりです。

```
Display Objects: 'Cost of Goods' Revenue
Sort Fields: 'Product,Category' 'Customer,Business,Region'
BY Fields: 'Product,Category'
ACROSS Fields: 'Customer,Business,Region'
Filters: WHERE 'Customer,Business,Region' NE 'Oceania';

Display Commands: WRITE 'Cost of Goods' Revenue
Sort Phrases: BY 'Product,Category' SUBTOTAL COGS_US ACROSS 'Customer,Business,Region' ACROSS-TOTAL
```

	Customer,Business,Region											
	EMEA			*North America*			*South America*			*TOTAL*		
Product Category	Cost of Goods	Revenue	RATIO	Cost of Goods	Revenue	RATIO	Cost of Goods	Revenue	RATIO	Cost of Goods	Revenue	RATIO
Accessories	$143,987.00	$211,477.99	1.47	$171,306.00	$249,935.10	1.46	$25,723.00	$35,445.49	1.38	$341,016.00	$496,858.58	4.31
*TOTAL Accessories	$143,987.00			$171,306.00			$25,723.00			$341,016.00		
Camcorder	$187,000.00	$280,235.28	1.50	$242,967.00	$349,755.15	1.44	$22,686.00	$36,276.20	1.60	$452,653.00	$666,266.63	4.54
*TOTAL Camcorder	$187,000.00			$242,967.00			$22,686.00			$452,653.00		
Computers	$47,616.00	$78,240.81	1.64	$53,329.00	$87,843.34	1.65	$7,845.00	$12,869.35	1.64	$108,790.00	$178,953.50	4.93
*TOTAL Computers	$47,616.00			$53,329.00			$7,845.00			$108,790.00		
Media Player	$328,401.00	$424,907.47	1.29	$386,681.00	$501,551.53	1.30	$60,183.00	$77,727.74	1.29	$775,265.00	$1,004,186.74	3.88
*TOTAL Media Player	$328,401.00			$386,681.00			$60,183.00			$775,265.00		
Stereo Systems	$361,756.00	$514,689.93	1.42	$412,036.00	$585,129.52	1.42	$81,823.00	$114,226.95	1.40	$855,615.00	$1,214,046.40	4.24
*TOTAL Stereo Systems	$361,756.00			$412,036.00			$81,823.00			$855,615.00		
Televisions	$100,443.00	$123,921.37	1.23	$101,212.00	$128,882.05	1.27	$25,800.00	$33,057.86	1.28	$227,455.00	$285,861.28	3.79
*TOTAL Televisions	$100,443.00			$101,212.00			$25,800.00			$227,455.00		
Video Production	$78,722.00	$112,141.72	1.42	$89,489.00	$129,247.23	1.44	$11,740.00	$16,872.90	1.44	$179,951.00	$258,261.85	4.31
*TOTAL Video Production	$78,722.00			$89,489.00			$11,740.00			$179,951.00		
TOTAL	$1,247,925.00			$1,457,020.00			$235,800.00			$2,940,745.00		

PDF レポートパネル単位の見出しと脚注の繰り返し

PDF レポートの列のすべてを 1 ページに表示できない場合、これらのページは自動的にパネル化されます。パネル化が有効な場合、同一ページの後続フィールドは、「オーバーフローページ」に配置されます。これらのオーバーフローページは、レポートの幅全体が表示されるまで生成され続け、幅全体が表示された後、次の縦方向ページが生成され、新しいページ番号と関連する横方向のパネルが表示されます。

最初のページに続くパネルを読みやすくするため、HEADPANEL=ON スタイルシート属性を使用して、最初のページの見出し要素が後続するパネルに繰り返し表示されるように指定する必要があります。

パネル化が有効で、デフォルトのページ番号設定を使用した場合、表示されるページ番号には、パネル番号が追加されます (例、1.1、1.2、1.3)。また、HEADPANEL をオンにすると、パネルの指定に TABPAGENO が含められます。

HEADPANEL をレポート全体に指定した場合、すべての見出しと脚注が、パネル化されたページに繰り返し表示されます。また、この設定は、見出し、脚注、中間見出し、中間脚注で、個別に有効にすることもできます。

HEADPANEL により、最初のページの境界線は、パネル化されたページに繰り返し表示されます。ALIGN-BORDERS を使用して、中間見出しおよび中間脚注に追加のコントロールを設定することができます。これにより、サブ項目の境界線の位置を、ページ見出しやレポート見出しではなく、データと揃えることができます。

パネルでの見出し要素の繰り返し

```
TYPE={REPORT|headfoot [BY=sortcolumn]}, HEADPANEL={ON|OFF}, $
```

説明

REPORT

レポート見出し、レポート脚注、ページ見出し、ページ脚注、中間見出し、中間脚注のすべてを繰り返します。

headfoot

見出しまたは脚注を指定します。次のオプションから選択します。

- **TABHEADING** レポート見出しです。レポートの先頭に一度だけ表示され、ON TABLE SUBHEAD で生成されます。
- **TABFOOTING** レポート脚注です。レポートの末尾に一度だけ表示され、ON TABLE SUBFOOT で生成されます。
- **HEADING** ページ見出しです。各レポートページの上部に表示され、HEADING で生成されます。
- **FOOTING** ページ脚注です。各レポートページの下部に表示され、FOOTING で生成されます。
- **SUBHEAD** ソート項目を指定している見出しです。BY フィールドグループの先頭に表示され、ON sortfield SUBHEAD で生成されます。
- **SUBFOOT** ソート項目を指定している脚注です。BY ソートグループの末尾に表示され、ON sortfield SUBFOOT で生成されます。

BY

ソート項目を指定している見出しまたは脚注が複数存在し、それぞれ異なる BY フィールド列に関連付けられている場合に、フォーマットの設定対象を指定することができます。

ソート項目を指定している見出しまたは脚注が複数存在し、それぞれ異なる BY フィールド列に関連付けられている場合、この属性を指定しないと、これらすべてにフォーマットが設定されます。

sortcolumn

レポートでソート項目を指定している見出しまたは脚注の 1 つに関連付ける BY フィールド列を指定します。

ON

指定した見出し要素または脚注要素を、各パネルに繰り返し表示します。

OFF

見出し要素または脚注要素を、最初のパネルにのみ表示します。デフォルト値は OFF です。

HEADPANEL=ON 属性は、見出し全体、または脚注全体にのみ適用可能です。見出しまたは脚注の行や項目に、個別に適用することはできません。

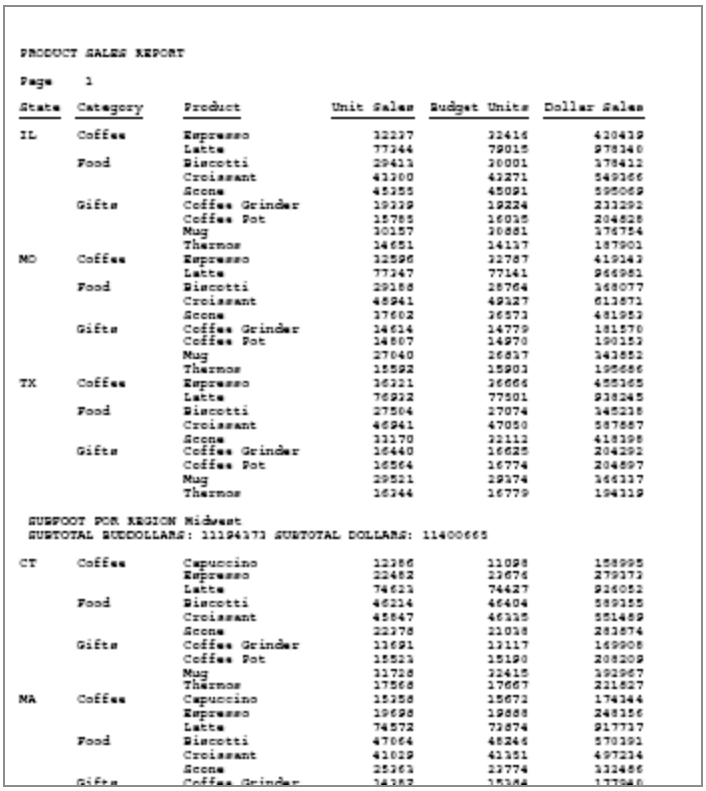
レポートパネルでの見出しと脚注すべての繰り返し表示

次のリクエストは、GGSALES データソースに対して実行され、地区、州、都市、カテゴリ、製品ごとの売上数量、売上数量目標、売上、売上目標の合計を計算します。このレポートにはページ見出し、および地区ごとの中間見出しが設定されます。

```
TABLE FILE GGSALES
HEADING
"PRODUCT SALES REPORT"
""
"Page<TABPAGENO"
""
SUM UNITS BUDUNITS DOLLARS BUDDOLLARS
BY REGION NOPRINT
BY ST BY CATEGORY BY PRODUCT
ON REGION SUBFOOT
" "
" SUBFOOT FOR REGION <REGION "
" SUBTOTAL BUDDOLLARS: <ST.BUDDOLLARS SUBTOTAL DOLLARS: <ST.DOLLARS "
" "
ON TABLE SET BYPANEL ON
ON TABLE SET PAGE ON
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE = REPORT, HEADPANEL=OFF,$
ENDSTYLE
END
```

このリクエストは、BYPANEL ON を設定し、各パネルにソートフィールドの値が表示されます。ただし、レポート全体に HEADPANEL=OFF が設定されているため、1 ページ (Page 1) の最初のパネルには見出しと中間見出しが表示されますが、2 つ目のパネルには表示されません。

下図のように、1 ページのパネル 1 の出力には、見出しと中間脚注が表示されます。HEADPANEL=OFF の場合、TABPAGENO にパネル番号は含められません。



PRODUCT SALES REPORT

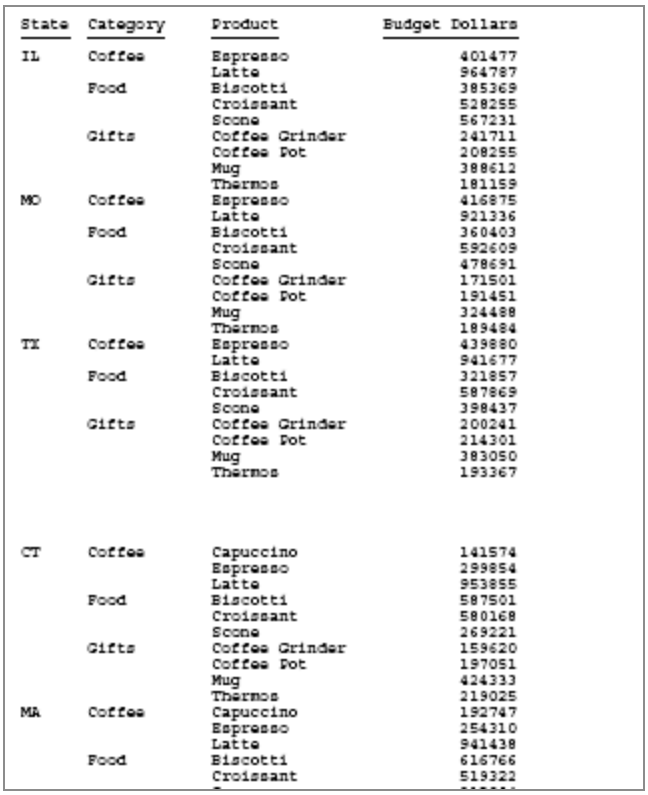
Page 1

State	Category	Product	Unit Sales	Budget Units	Dollar Sales
IL	Coffee	Espresso	32237	32416	420419
		Latte	77244	79015	978340
	Food	Biscotti	29413	30001	378412
		Croissant	41300	41271	549366
		Scone	45355	45091	595069
	Gifts	Coffee Grinder	19339	19224	233292
		Coffee Pot	15785	16035	204828
		Mug	10157	10881	176754
		Thermos	14651	14117	187901
MO	Coffee	Espresso	32596	32767	419143
		Latte	77347	77141	966961
	Food	Biscotti	29188	28764	368077
		Croissant	48941	49327	613871
		Scone	37602	36573	481953
	Gifts	Coffee Grinder	14614	14779	181570
		Coffee Pot	14807	14970	190153
		Mug	27040	26817	343852
		Thermos	15592	15903	195686
TX	Coffee	Espresso	36321	36666	455365
		Latte	76932	77501	938245
	Food	Biscotti	27504	27074	345238
		Croissant	46941	47050	587887
		Scone	33170	32111	418398
	Gifts	Coffee Grinder	16440	16625	204292
		Coffee Pot	16564	16774	204897
		Mug	29521	29374	366317
		Thermos	16344	16779	194319

SUBFOOT FOR REGION Midwest
SUBTOTAL BUDDOLLARS: 11194171 SUBTOTAL DOLLARS: 11400665

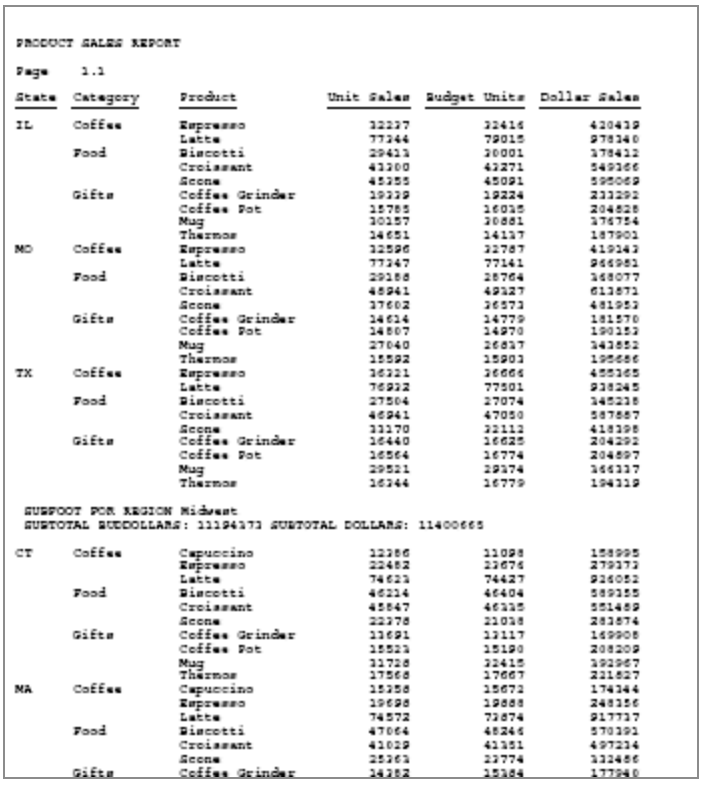
CT	Coffee	Capuccino	12386	11098	158995
		Espresso	22482	21676	279173
		Latte	74621	74427	926052
	Food	Biscotti	46214	46404	589155
		Croissant	45847	46115	551489
		Scone	22378	21018	283874
	Gifts	Coffee Grinder	11691	13117	169908
		Coffee Pot	15521	15190	208209
		Mug	31728	32415	392967
		Thermos	17568	17667	221827
MA	Coffee	Capuccino	15158	15672	174344
		Espresso	19698	19888	248156
		Latte	74572	73874	917717
	Food	Biscotti	47064	48246	570191
		Croissant	41029	41151	497214
		Scone	25361	23774	332486
	Gifts	Coffee Grinder	14367	15364	177840

下図のように、1 ページのパネル 2 の出力には、見出しや中間脚注は表示されません。



State	Category	Product	Budget Dollars
IL	Coffee	Espresso	401477
		Latte	964787
	Food	Biscotti	385369
		Croissant	528255
		Scone	567231
	Gifts	Coffee Grinder	241711
		Coffee Pot	208255
		Mug	388612
		Thermos	181159
MO	Coffee	Espresso	416875
		Latte	921336
	Food	Biscotti	360403
		Croissant	592609
		Scone	478691
	Gifts	Coffee Grinder	171501
		Coffee Pot	191451
		Mug	324488
		Thermos	189484
TX	Coffee	Espresso	439880
		Latte	941677
	Food	Biscotti	321857
		Croissant	587869
		Scone	398437
	Gifts	Coffee Grinder	200241
		Coffee Pot	214301
		Mug	383050
		Thermos	193367
CT	Coffee	Capuccino	141574
		Espresso	299854
		Latte	953855
	Food	Biscotti	587501
		Croissant	580168
		Scone	269221
	Gifts	Coffee Grinder	159620
		Coffee Pot	197051
		Mug	424333
		Thermos	219025
MA	Coffee	Capuccino	192747
		Espresso	254310
		Latte	941438
	Food	Biscotti	616766
		Croissant	519322

スタイルシートの指定を変更し、レポート全体に HEADPANEL=ON を設定した場合 (TYPE=REPORT, HEADPANEL=ON ,$)、パネル 1 とパネル 2 は、下図のようになります。見出しと脚注は、各パネルに繰り返し表示されます。HEADPANEL=ON の場合、TABPAGENO にパネル番号が含められます。



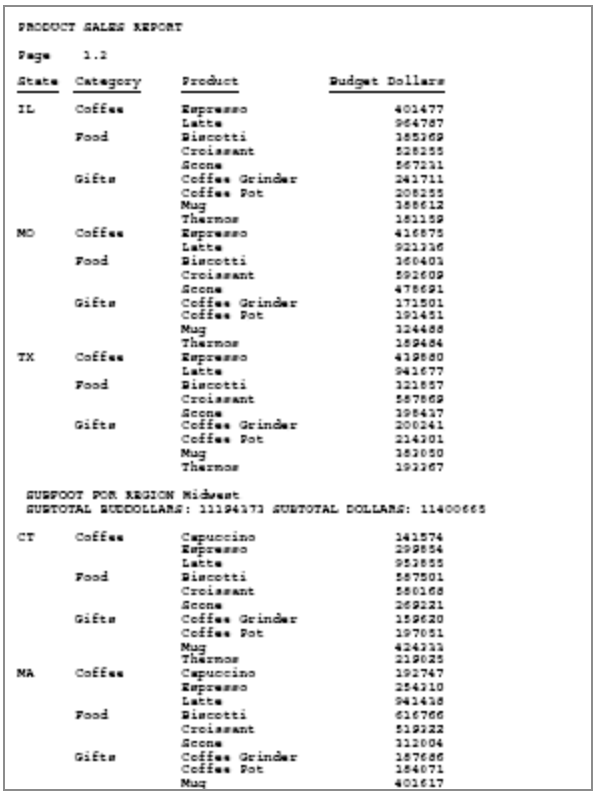
PRODUCT SALES REPORT

Page 1.1

State	Category	Product	Unit Sales	Budget Units	Dollar Sales
IL	Coffee	Espresso	32237	32416	420439
		Latte	77344	79015	978140
	Food	Biscotti	29413	30001	178412
		Croissant	41300	41271	549166
		Scone	45355	45091	596069
	Gifts	Coffee Grinder	19339	19224	233292
		Coffee Pot	15785	16035	204525
		Mug	30157	30881	176754
		Thermos	14651	14117	187901
MO	Coffee	Espresso	32596	32787	419143
		Latte	77347	77141	966951
	Food	Biscotti	29188	28764	168077
		Croissant	48941	49127	613871
		Scone	37602	36573	481953
	Gifts	Coffee Grinder	14614	14779	181570
		Coffee Pot	14807	14970	190153
		Mug	27040	26817	143852
		Thermos	15592	15903	195686
TX	Coffee	Espresso	36321	36666	455365
		Latte	76922	77501	938245
	Food	Biscotti	27504	27074	145238
		Croissant	46941	47050	587867
		Scone	33170	32112	418398
	Gifts	Coffee Grinder	16440	16625	204292
		Coffee Pot	16564	16774	204897
		Mug	29521	29174	166137
		Thermos	16344	16779	194119

SUBFOOT FOR REGION Midwest
SUBTOTAL BUDDOLLARS: 11194173 SUBTOTAL DOLLARS: 11400665

CT	Coffee	Capuccino	12386	11098	158995
		Espresso	22482	23676	279373
		Latte	74621	74427	926052
	Food	Biscotti	46214	46404	589155
		Croissant	45847	46135	551469
		Scone	22378	21038	283874
	Gifts	Coffee Grinder	13691	13117	169908
		Coffee Pot	15521	15190	208209
		Mug	31728	32415	192967
		Thermos	17568	17667	221827
MA	Coffee	Capuccino	15356	15672	174344
		Espresso	19698	19868	248156
		Latte	74572	72874	917737
	Food	Biscotti	47064	48246	570191
		Croissant	41029	41151	497214
		Scone	25361	23774	332456
	Gifts	Coffee Grinder	14282	15384	177940



PRODUCT SALES REPORT

Page 1.2

State	Category	Product	Budget Dollars
IL	Coffee	Espresso	401477
		Latte	964787
	Food	Biscotti	185369
		Croissant	528255
		Scone	567231
	Gifts	Coffee Grinder	241711
		Coffee Pot	208255
		Mug	188612
		Thermos	181159
MO	Coffee	Espresso	416875
		Latte	921336
	Food	Biscotti	160401
		Croissant	592609
		Scone	478691
	Gifts	Coffee Grinder	171501
		Coffee Pot	191451
		Mug	124488
		Thermos	189484
TX	Coffee	Espresso	419880
		Latte	941677
	Food	Biscotti	121857
		Croissant	587869
		Scone	195417
	Gifts	Coffee Grinder	200241
		Coffee Pot	214201
		Mug	183050
		Thermos	192267

SUBFOOT FOR REGION Midwest
SUBTOTAL BUDDOLLARS: 11194173 SUBTOTAL DOLLARS: 11400665

CT	Coffee	Capuccino	141574
		Espresso	299854
		Latte	953855
	Food	Biscotti	587501
		Croissant	580168
		Scone	269221
	Gifts	Coffee Grinder	159620
		Coffee Pot	197051
		Mug	424211
		Thermos	219025
MA	Coffee	Capuccino	192747
		Espresso	254210
		Latte	941418
	Food	Biscotti	616766
		Croissant	519122
		Scone	112004
	Gifts	Coffee Grinder	187686
		Coffee Pot	184071
		Mug	401617

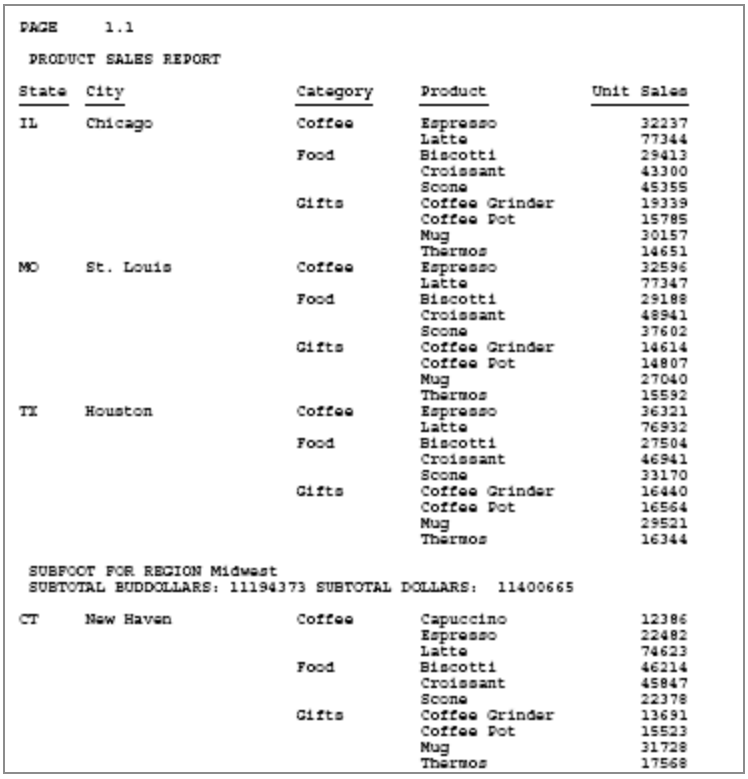
PDF レポートパネルでの中間脚注の繰り返し表示

次のリクエストは、GGSALES データソースに対して実行され、レポート全体ではなく、中間見出しのみに HEADPANEL=ON を指定します。このリクエストでは、TABPAGENO ではなく、デフォルトのページ番号設定 (ON TABLE SET PAGE ON) を使用して、各ページにページ番号を表示します。

```
TABLE FILE GGSALES
HEADING
" PRODUCT SALES REPORT"
" "
SUM UNITS BUDUNITS DOLLARS BUDDOLLARS
BY REGION NOPRINT
BY ST BY CITY  BY CATEGORY BY PRODUCT
ON REGION SUBFOOT
" "
" SUBFOOT FOR REGION <REGION "
" SUBTOTAL BUDDOLLARS: <ST.BUDDOLLARS SUBTOTAL DOLLARS:  <ST.DOLLARS "
```

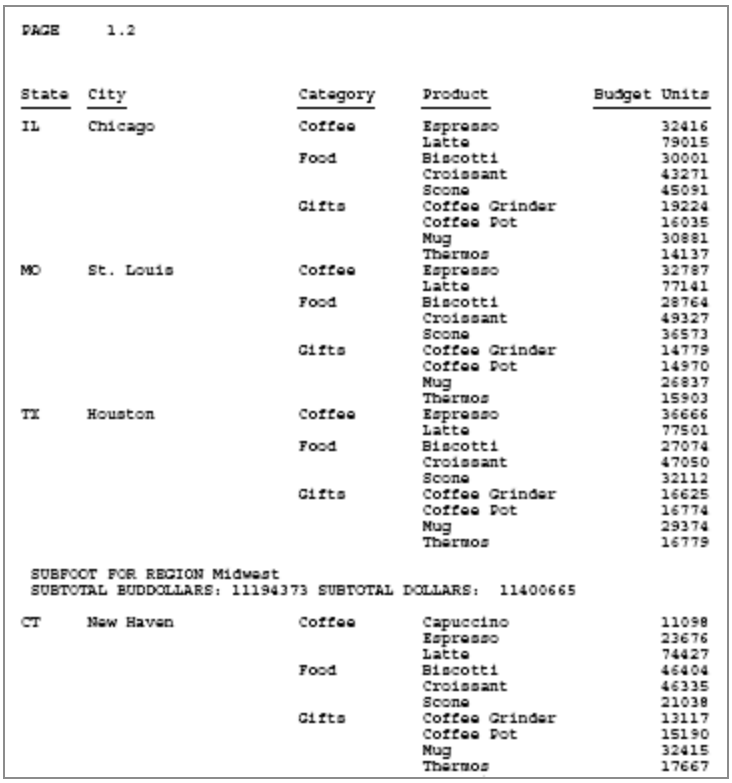
```
" "
ON TABLE SET BYPANEL ON
ON TABLE SET PAGE ON
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE = SUBFOOT, HEADPANEL=ON,$

ENDSTYLE
END
```

下図のように、パネル 1 には、見出しと中間脚注の両方が表示されます。



```
PAGE     1.1

 PRODUCT SALES REPORT

State  City                 Category     Product           Unit Sales
IL     Chicago              Coffee       Espresso               32237
                                         Latte                  77344
                            Food         Biscotti               29413
                                         Croissant              43300
                                         Scone                  45355
                            Gifts        Coffee Grinder         19339
                                         Coffee Pot             15785
                                         Mug                    30157
                                         Thermos                14651
MO     St. Louis            Coffee       Espresso               32596
                                         Latte                  77347
                            Food         Biscotti               29188
                                         Croissant              48941
                                         Scone                  37602
                            Gifts        Coffee Grinder         14614
                                         Coffee Pot             14807
                                         Mug                    27040
                                         Thermos                15592
TX     Houston              Coffee       Espresso               36321
                                         Latte                  76932
                            Food         Biscotti               27504
                                         Croissant              46941
                                         Scone                  33170
                            Gifts        Coffee Grinder         16440
                                         Coffee Pot             16564
                                         Mug                    29521
                                         Thermos                16344

 SUBFOOT FOR REGION Midwest
 SUBTOTAL BUDDOLLARS: 11194373 SUBTOTAL DOLLARS:  11400665

CT     New Haven            Coffee       Capuccino              12386
                                         Espresso               22482
                                         Latte                  74623
                            Food         Biscotti               46214
                                         Croissant              45847
                                         Scone                  22378
                            Gifts        Coffee Grinder         13691
                                         Coffee Pot             15523
                                         Mug                    31728
                                         Thermos                17568
```

下図のように、パネル 2 には、見出しではなく、中間脚注のみが表示されます。



PAGE 1.2

State	City	Category	Product	Budget Units
IL	Chicago	Coffee	Espresso	32416
			Latte	79015
		Food	Biscotti	30001
			Croissant	43271
			Scone	45091
		Gifts	Coffee Grinder	19224
			Coffee Pot	16035
			Mug	30881
			Thermos	14137
MO	St. Louis	Coffee	Espresso	32787
			Latte	77141
		Food	Biscotti	28764
			Croissant	49327
			Scone	36573
		Gifts	Coffee Grinder	14779
			Coffee Pot	14970
			Mug	26837
			Thermos	15903
TX	Houston	Coffee	Espresso	36666
			Latte	77501
		Food	Biscotti	27074
			Croissant	47050
			Scone	32112
		Gifts	Coffee Grinder	16625
			Coffee Pot	16774
			Mug	29374
			Thermos	16779

SUBFOOT FOR REGION Midwest
SUBTOTAL BUDDOLLARS: 11194373 SUBTOTAL DOLLARS: 11400665

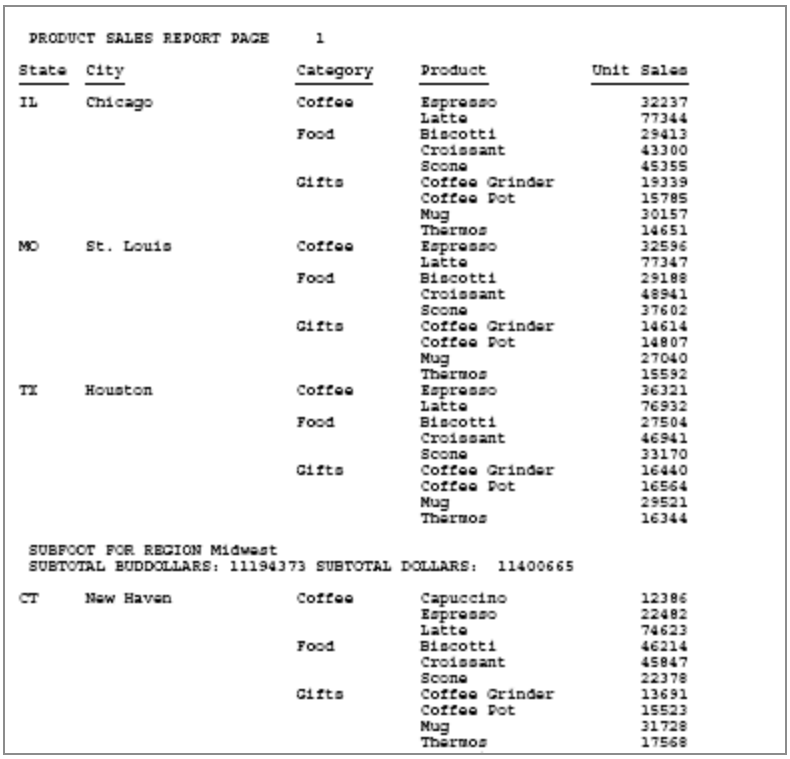
CT	New Haven	Coffee	Capuccino	11098
			Espresso	23676
			Latte	74427
		Food	Biscotti	46404
			Croissant	46335
			Scone	21038
		Gifts	Coffee Grinder	13117
			Coffee Pot	15190
			Mug	32415
			Thermos	17667

ページ見出しは繰り返し表示されないため、システム変数 <TABPAGENO を使用して見出しにページ番号を配置した場合、パネル番号は表示されず、2 つ目のパネルにページ番号は表示されません。

```
TABLE FILE GGSALES
HEADING
" PRODUCT SALES REPORT PAGE <TABPAGENO"
" "
SUM UNITS BUDUNITS DOLLARS BUDDOLLARS
BY REGION NOPRINT
BY ST BY CITY  BY CATEGORY BY PRODUCT
ON REGION SUBFOOT
" "
" SUBFOOT FOR REGION <REGION "
" SUBTOTAL BUDDOLLARS: <ST.BUDDOLLARS SUBTOTAL DOLLARS:  <ST.DOLLARS "
" "
ON TABLE SET BYPANEL ON
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE = SUBFOOT, HEADPANEL=ON,$
```

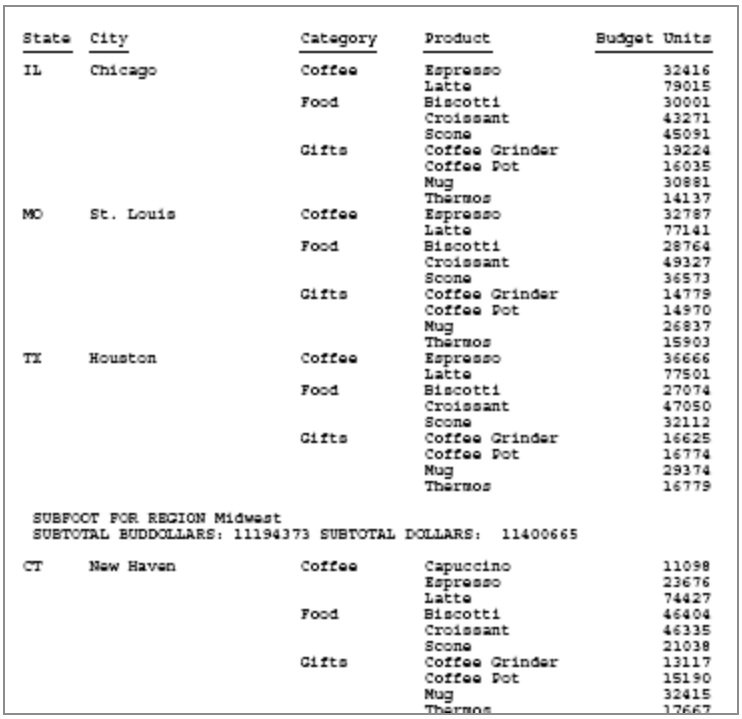
```
ENDSTYLE
END
```

下図のように、1 つ目のパネルには、見出しにページ番号が表示され、パネル番号は表示されません。



```
 PRODUCT SALES REPORT PAGE     1

State  City                 Category     Product           Unit Sales

IL     Chicago              Coffee       Espresso               32237
                                         Latte                  77344
                            Food         Biscotti               29413
                                         Croissant              43300
                                         Scone                  45355
                            Gifts        Coffee Grinder         19339
                                         Coffee Pot             15785
                                         Mug                    30157
                                         Thermos                14651
MO     St. Louis            Coffee       Espresso               32596
                                         Latte                  77347
                            Food         Biscotti               29188
                                         Croissant              48941
                                         Scone                  37602
                            Gifts        Coffee Grinder         14614
                                         Coffee Pot             14807
                                         Mug                    27040
                                         Thermos                15592
TX     Houston              Coffee       Espresso               36321
                                         Latte                  76932
                            Food         Biscotti               27504
                                         Croissant              46941
                                         Scone                  33170
                            Gifts        Coffee Grinder         16440
                                         Coffee Pot             16564
                                         Mug                    29521
                                         Thermos                16344

 SUBFOOT FOR REGION Midwest
 SUBTOTAL BUDDOLLARS: 11194373 SUBTOTAL DOLLARS:  11400665

CT     New Haven            Coffee       Capuccino              12386
                                         Espresso               22482
                                         Latte                  74623
                            Food         Biscotti               46214
                                         Croissant              45847
                                         Scone                  22378
                            Gifts        Coffee Grinder         13691
                                         Coffee Pot             15523
                                         Mug                    31728
                                         Thermos                17568
```

下図のように、2 つ目のパネルには、見出しは表示されないことから、埋め込みのページ番号は表示されません。



State	City	Category	Product	Budget Units
IL	Chicago	Coffee	Espresso	32416
			Latte	79015
		Food	Biscotti	30001
			Croissant	43271
			Scone	45091
		Gifts	Coffee Grinder	19224
			Coffee Pot	16035
			Mug	30881
			Thermos	14137
MO	St. Louis	Coffee	Espresso	32787
			Latte	77141
		Food	Biscotti	28764
			Croissant	49327
			Scone	36573
		Gifts	Coffee Grinder	14779
			Coffee Pot	14970
			Mug	26837
			Thermos	15903
TX	Houston	Coffee	Espresso	36666
			Latte	77501
		Food	Biscotti	27074
			Croissant	47050
			Scone	32112
		Gifts	Coffee Grinder	16625
			Coffee Pot	16774
			Mug	29374
			Thermos	16779

SUBFOOT FOR REGION Midwest
SUBTOTAL BUDDOLLARS: 11194373 SUBTOTAL DOLLARS: 11400665

CT	New Haven	Coffee	Capuccino	11098
			Espresso	23676
			Latte	74427
		Food	Biscotti	46404
			Croissant	46335
			Scone	21038
		Gifts	Coffee Grinder	13117
			Coffee Pot	15190
			Mug	32415
			Thermos	17667

パネルページでのスタイル設定された見出しと脚注の繰り返し表示

次のリクエストは、GGSALES データソースに対して実行され、レポート見出し、イメージ付きのページ見出し、脚注、中間見出し、中間脚注、中間合計が含まれています。

```
SET BYPANEL=ON
DEFINE FILE GGSALES
SHOWCATPROD/A30 = CATEGORY || ' / ' || PRODUCT;
END
TABLE FILE GGSALES
SUM
     DOLLARS/I8M AS ''
BY REGION
BY ST
BY CITY
ACROSS SHOWCATPROD AS 'Product Sales'

ON REGION SUBHEAD
" "
"Subheading Region <REGION"
" "
```

```
ON REGION SUBTOTAL AS '*TOTAL'
ON REGION SUBFOOT WITHIN
" "
"Subfooting Region <REGION"
" "
ON TABLE SUBHEAD
"Report Heading"
HEADING
"Page <TABPAGENO  "
" "
" "
" "
FOOTING
" "
"PAGE FOOTING "
ON TABLE SUBFOOT
" "
"Report Footing"
ON TABLE SET PAGE-NUM OFF
-*ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
     UNITS=IN,
     SQUEEZE=ON,
     ORIENTATION=PORTRAIT,
$
```

```
TYPE=REPORT,
     FONT='ARIAL',
     SIZE=9,
     HEADPANEL=ON,
     BORDER=ON,
$
TYPE=TITLE,
     STYLE=BOLD,
$
TYPE=TABHEADING,
     SIZE=20,
     STYLE=BOLD,
$
TYPE=TABFOOTING,
     SIZE=20,
     STYLE=BOLD,
$
TYPE=HEADING,
```

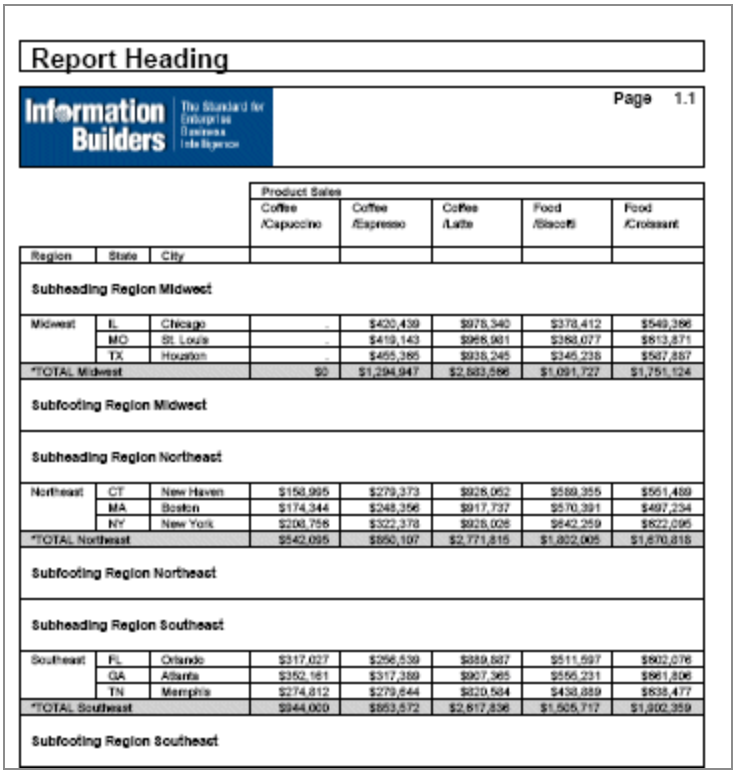
```
     SIZE=12,
     STYLE=BOLD,
$
TYPE=HEADING,
     LINE=1,
     JUSTIFY=RIGHT,
$
TYPE=HEADING,
     LINE=2,
     JUSTIFY=RIGHT,
$
TYPE=HEADING,
     LINE=3,
     JUSTIFY=RIGHT,
$
TYPE=HEADING,
     LINE=4,
     JUSTIFY=RIGHT,
$
TYPE=HEADING,
     LINE=5,
     JUSTIFY=RIGHT,
$
TYPE=HEADING,
     IMAGE=smplogo1.gif,
     POSITION=(+0.000000 +0.000000),
$
TYPE=FOOTING,
     SIZE=12,
     STYLE=BOLD,
     JUSTIFY=RIGHT,
$
TYPE=SUBHEAD,
     SIZE=10,
     STYLE=BOLD,
$
```

```
TYPE=SUBFOOT,
     SIZE=10,
     STYLE=BOLD,
$
TYPE=SUBTOTAL,
     BACKCOLOR=RGB(210 210 210),
$
TYPE=ACROSSVALUE,
     SIZE=9,
```

```
     WRAP=ON,
$
TYPE=ACROSSTITLE,
     STYLE=BOLD,
$
TYPE=GRANDTOTAL,
     BACKCOLOR=RGB(210 210 210),
     STYLE=BOLD,
$
ENDSTYLE
END
```

レポート全体に HEADPANEL=ON が設定されているため、両方のパネルに、見出し要素と脚注要素がすべて表示されます。

下図は、1 ページのパネル 1 を示しています。



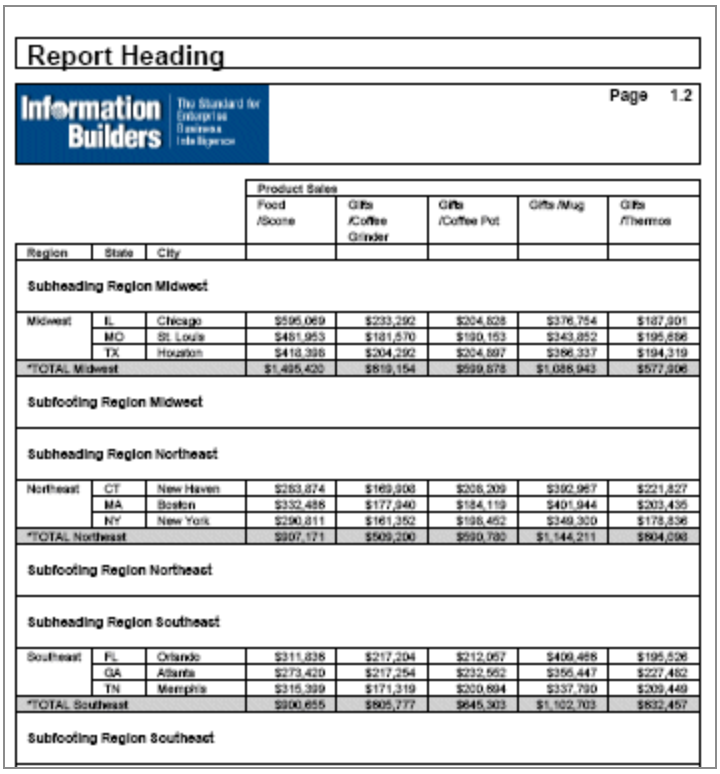
Report Heading

Information Builders — The Standard for Enterprise Business Intelligence

Page 1.1

			Product Sales				
			Coffee /Cappuccino	Coffee /Espresso	Coffee /Latte	Food /Biscotti	Food /Croissant
Region	State	City					
Subheading Region Midwest							
Midwest	IL	Chicago	..	$420,439	$978,340	$378,412	$549,386
	MO	St. Louis	..	$419,143	$966,981	$368,077	$513,871
	TX	Houston	..	$455,365	$938,245	$345,238	$587,867
*TOTAL Midwest			$0	$1,294,947	$2,883,566	$1,091,727	$1,751,124
Subfooting Region Midwest							
Subheading Region Northeast							
Northeast	CT	New Haven	$158,995	$279,373	$926,052	$589,355	$561,489
	MA	Boston	$174,344	$248,356	$917,737	$570,391	$497,234
	NY	New York	$208,756	$322,378	$926,026	$642,259	$822,095
*TOTAL Northeast			$542,095	$850,107	$2,771,815	$1,802,005	$1,870,818
Subfooting Region Northeast							
Subheading Region Southeast							
Southeast	FL	Orlando	$317,027	$256,539	$889,887	$511,597	$902,076
	GA	Atlanta	$352,161	$317,389	$907,365	$555,231	$861,806
	TN	Memphis	$274,812	$279,644	$820,584	$438,889	$838,477
*TOTAL Southeast			$944,000	$853,572	$2,617,836	$1,505,717	$1,902,359
Subfooting Region Southeast							

下図は、1 ページのパネル 2 を示しています。



Report Heading

Information Builders The Standard for Enterprise Business Intelligence

Page 1.2

			Product Sales				
			Food /Scone	Gifts /Coffee Grinder	Gifts /Coffee Pot	Gifts /Mug	Gifts /Thermos
Region	State	City					
Subheading Region Midwest							
Midwest	IL	Chicago	$595,069	$233,292	$204,828	$376,754	$187,801
	MO	St. Louis	$481,953	$181,570	$190,153	$343,852	$195,686
	TX	Houston	$418,398	$204,292	$204,897	$366,337	$194,319
*TOTAL Midwest			$1,495,420	$619,154	$599,878	$1,086,943	$577,806
Subfooting Region Midwest							
Subheading Region Northeast							
Northeast	CT	New Haven	$283,874	$169,908	$208,209	$392,967	$221,827
	MA	Boston	$332,488	$177,940	$184,119	$401,944	$203,435
	NY	New York	$290,811	$161,352	$198,462	$349,300	$178,836
*TOTAL Northeast			$907,171	$509,200	$590,790	$1,144,211	$804,098
Subfooting Region Northeast							
Subheading Region Southeast							
Southeast	FL	Orlando	$311,838	$217,204	$212,067	$409,466	$195,526
	GA	Atlanta	$273,420	$217,254	$232,562	$356,447	$227,482
	TN	Memphis	$315,399	$171,319	$200,684	$337,790	$209,449
*TOTAL Southeast			$900,655	$805,777	$645,303	$1,102,703	$832,457
Subfooting Region Southeast							

フィールドタイトルのカスタマイズ

フィールドタイトルは、レポート内のデータを識別します。デフォルトフィールドタイトルを変更してカスタマイズしたデータ ID を使用したり、より適切なフォーマットを適用するには、AS 句を使用します。フィールドタイトルは、次の位置で変更することができます。

- リクエスト
- マスターファイル

フィールドタイトルのデフォルト値は、マスターファイルのフィールド名です。一時項目 (COMPUTE) では、タイトルのデフォルト値はリクエスト内のフィールド名です。

デフォルトフィールドタイトルの使用

次のリクエストについて考察します。

```
TABLE FILE EMPDATA
SUM SALARY
BY DEPT
BY LASTNAME
WHERE DEPT IS 'SALES' OR 'CONSULTING' OR 'ACCOUNTING'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

このレポートの出力結果は、デフォルトフィールドタイトルを表示します。

- 表示コマンド (SUM) で指定されるフィールドタイトルは、SALARY です。
- BY 句で指定されるフィールドタイトルは、DEPT および LASTNAME です。

出力結果は次のとおりです。

DEPT	LASTNAME	SALARY
ACCOUNTING	ANDERSON	$32,400.00
	LOPEZ	$26,400.00
	SANCHEZ	$83,000.00
	SOPENA	$79,000.00
	WANG	$62,500.00
CONSULTING	ELLNER	$35,900.00
	WANG	$49,500.00
	WHITE	$40,900.00
SALES	ADDAMS	$54,100.00
	CASSANOVA	$70,000.00
	CHISOLM	$43,000.00
	DUBOIS	$43,600.00
	LASTRA	$115,000.00
	MEDINA	$39,000.00
	OLSON	$30,500.00

フィールドタイトルの制限事項

- 利用可能なメモリの制限を除いて、フィールドタイトルに割り当てる幅に制限はありません。
- スタイルフォーマットの出力では、フィールドタイトルに 16 行以内のテキストを含めることができます。
- ACROSS 句で指定されるフィールドのタイトルは、1 行のテキストのみで置き換えることができます。

リクエストでのフィールドタイトルのカスタマイズ

```
fieldname AS 'title_line_1 [,title_line_2,...]'
```

説明

fieldname

表示コマンド (PRINT や SUM など)、ACROSS 句、または BY 句で指定されるフィールド名です。

title_line_1,title_line_2

カスタマイズするフィールドタイトルを一重引用符 (') で囲んで指定します。

複数行のフィールドタイトルを指定するには、それぞれの行をカンマ (,) で区切ります。

一時項目 (COMPUTE) のフィールドタイトルをカスタマイズするには、次の構文を使用します。

```
COMPUTE fieldname[/format] = expression AS 'title'
```

詳細は、一時フィールドの作成を参照してください。

ヒント：

- フィールドタイトルを非表示にするには、ブランクを入れずに一重引用符 (') を続けて

2 回入力します。以下はその例です。

```
PRINT LAST_NAME AS ''
```

- アンダスコア (_) を表示するには、ブランクを一重引用符 (') で囲みます。
- 一時項目 (COMPUTE) に AS 句を使用する場合、次の一時項目 (COMPUTE) を参照する前に、COMPUTE コマンドを繰り返します。
- ACROSS フィールドでは、AS 句を使用して作成された複数行のフィールドタイトルはサポートされません。

リクエストでのフィールドタイトルのカスタマイズ

次のリクエストは、SUM コマンド (SALARY) で指定されるフィールドのタイトル、および BY 句 (DIV および DEPT) で指定されるフィールドのタイトルをカスタマイズします。

```
TABLE FILE EMPDATA
SUM SALARY AS 'Total,Salary'
BY DIV AS 'Division'
BY DEPT AS 'Department'
WHERE DIV EQ 'NE' OR 'SE' OR 'CORP'
HEADING
"Current Salary Report"
ON TABLE SET STYLE *
TYPE=REPORT,GRID=OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。

Current Salary Report

Division	Department	Total Salary
CORP	ACCOUNTING	$283,300.00
	MARKETING	$153,200.00
NE	CUSTOMER SUPPORT	$81,800.00
	MARKETING	$139,800.00
	SALES	$82,600.00
SE	CONSULTING	$85,400.00
	CUSTOMER SUPPORT	$62,500.00
	MARKETING	$113,000.00
	PERSONNEL	$80,500.00
	PROGRAMMING & DVLPMT	$49,000.00

フィールドタイトルの非表示

次のリクエストは、LAST_NAME のフィールドタイトルを非表示にします。さらに、EMP_ID データに複数行にわたるフィールドタイトル (EMPLOYEE NUMBER) を指定します。

```
TABLE FILE EMPLOYEE
PRINT FIRST_NAME AS 'NAME' AND LAST_NAME AS ''
BY DEPARTMENT
BY EMP_ID AS 'EMPLOYEE,NUMBER'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

DEPARTMENT	EMPLOYEE NUMBER	NAME	
MIS	112847612	MARY	SMITH
	117593129	DIANE	JONES
	219984371	JOHN	MCCOY
	326179357	ROSEMARIE	BLACKWOOD
	543729165	MARY	GREENSPAN
	818692173	BARBARA	CROSS
PRODUCTION	071382660	ALFRED	STEVENS
	119265415	RICHARD	SMITH
	119329144	JOHN	BANNING
	123764317	JOAN	IRVING
	126724188	ANTHONY	ROMANS
	451123478	ROGER	MCKNIGHT

一時項目 (COMPUTE) のフィールドタイトルのカスタマイズ

次のリクエストは、一時項目 (COMPUTE) REV のフィールドタイトルをカスタマイズします。

```
TABLE FILE SALES
SUM UNIT_SOLD RETAIL_PRICE
COMPUTE REV/D12.2M = UNIT_SOLD * RETAIL_PRICE;AS 'GENERATED REVENUE'
BY PROD_CODE
WHERE CITY EQ 'NEW YORK'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

PROD_CODE	UNIT_SOLD	RETAIL_PRICE	GENERATED REVENUE
B10	30	$.85	$25.50
B17	20	$1.89	$37.80
B20	15	$1.99	$29.85
C17	12	$2.09	$25.08
D12	20	$2.09	$41.80
E1	30	$.89	$26.70
E3	35	$1.09	$38.15

マスターファイルでのフィールドタイトルのカスタマイズ

フィールドの TITLE 属性オプションを使用すると、デフォルトフィールドタイトルを変更できます。フィールドに適用するフォーマットは、カスタマイズしたタイトルにも適用されます。

TITLE 属性についての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

重複フィールド名の区別

SET QUALTITLES コマンド は、レポートの出力内の重複フィールド名に、修飾フィールドタイトルを表示するかどうかを決定します。修飾フィールドタイトルは、セグメントを含めることにより、同一フィールド名を区別します。

AS 句内で指定されるフィールドタイトルは、重複フィールド名が MATCH コマンドで参照されるとき、または HOLD ファイル内に重複フィールド名が存在するときに使用されます。

重複フィールド名の区別

```
SET QUALTITLES = {ON|OFF}
```

説明

ON

重複フィールド名が存在し、SET FIELDNAME が NEW (デフォルト) に設定されるときに、修飾フィールドタイトルの使用を有効にします。SET コマンドについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

OFF

修飾フィールドタイトルの使用を無効にします。デフォルト値は OFF です。

SET コマンドによるフィールドタイトルの下線の制御

SET TITLELINE コマンドを使用して、レポート出力のフィールドタイトルに下線を挿入するかどうかを制御することができます。

SET コマンドによるフィールドタイトルの下線の制御

```
SET TITLELINE =  (ON|OFF|SKIP)
```

```
ON TABLE SET TITLELINE  (ON|OFF|SKIP)
```

説明

ON

フィールドタイトルに下線を挿入します。デフォルト値は ON です。

OFF

下線の代わりにブランク行を挿入します。

SKIP

下線が表示される位置の下線および行を除外します。

注意：ACROSSLINE は、TITLELINE と同義です。

SET コマンドによるフィールドタイトルの下線の制御

次のリクエストでは、BY および ACROSS フィールドが指定されています。

```
SET TITLELINE=ON
TABLE FILE GGSALES
SUM UNITS BY PRODUCT
ACROSS REGION
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/javaassist/intl/EN/ENIADefault_combine.sty,$
ENDSTYLE
END
```

デフォルト設定の SET TITLELINE=ON を使用すると、フィールドタイトルに下線が挿入されます。

	Region			
	Midwest	Northeast	Southeast	West
Product				
Biscotti	86105	145242	119594	70436
Capuccino	.	44785	73264	71168
Coffee Grinder	50393	40977	47083	48081
Coffee Pot	47156	46185	49922	47432
Croissant	139182	137394	156456	197022
Espresso	101154	68127	68030	71675
Latte	231623	222866	209654	213920
Mug	86718	91497	88474	93881
Scone	116127	70732	73779	72776
Thermos	46587	48870	48976	45648

SET TITLELINE=OFF を使用すると、フィールドタイトルに下線は挿入されませんが、下線が表示される位置にブランク行が残っています。

	Region			
	Midwest	Northeast	Southeast	West
Product				
Biscotti	86105	145242	119594	70436
Capuccino	.	44785	73264	71168
Coffee Grinder	50393	40977	47083	48081
Coffee Pot	47156	46185	49922	47432
Croissant	139182	137394	156456	197022
Espresso	101154	68127	68030	71675
Latte	231623	222866	209654	213920
Mug	86718	91497	88474	93881
Scone	116127	70732	73779	72776
Thermos	46587	48870	48976	45648

SET TITLELINE=SKIP を使用すると、下線とブランク行の両方が除外されます。

	Region			
	Midwest	Northeast	Southeast	West
Product				
Biscotti	86105	145242	119594	70436
Capuccino	.	44785	73264	71168
Coffee Grinder	50393	40977	47083	48081
Coffee Pot	47156	46185	49922	47432
Croissant	139182	137394	156456	197022
Espresso	101154	68127	68030	71675
Latte	231623	222866	209654	213920
Mug	86718	91497	88474	93881
Scone	116127	70732	73779	72776
Thermos	46587	48870	48976	45648

スタイルシート属性によるフィールドタイトルの下線の制御

TITLELINE 属性を使用して、レポート出力のフィールドタイトルに下線を挿入するかどうかを制御することができます。

スタイルシート属性によるフィールドタイトルの下線の制御

```
TYPE={REPORT|TITLE}, TITLELINE =  (ON|OFF|SKIP)
```

説明

ON

フィールドタイトルに下線を挿入します。デフォルト値は ON です。

OFF

下線の代わりにブランク行を挿入します。

SKIP

下線が表示される位置の下線および行を除外します。

スタイルシート属性によるフィールドタイトルの下線の制御

次のリクエストでは、BY および ACROSS フィールドが指定されています。

```
TABLE FILE GGSALES
SUM UNITS BY PRODUCT
ACROSS REGION
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, TITLELINE=ON, GRID=OFF, FONT=ARIAL,$
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/javaassist/intl/EN/ENIADefault_combine.sty,$
ENDSTYLE
END
```

デフォルト設定の TITLELINE=ON を使用すると、フィールドタイトルに下線が挿入されます。

	Region			
	Midwest	Northeast	Southeast	West
Product				
Biscotti	86105	145242	119594	70436
Capuccino	.	44785	73264	71168
Coffee Grinder	50393	40977	47083	48081
Coffee Pot	47156	46185	49922	47432
Croissant	139182	137394	156456	197022
Espresso	101154	68127	68030	71675
Latte	231623	222866	209654	213920
Mug	86718	91497	88474	93881
Scone	116127	70732	73779	72776
Thermos	46587	48870	48976	45648

TITLELINE=OFF を使用すると、フィールドタイトルに下線は挿入されず、下線の代わりにブランク行が挿入されます。

	Region			
	Midwest	Northeast	Southeast	West
Product				
Biscotti	86105	145242	119594	70436
Capuccino	.	44785	73264	71168
Coffee Grinder	50393	40977	47083	48081
Coffee Pot	47156	46185	49922	47432
Croissant	139182	137394	156456	197022
Espresso	101154	68127	68030	71675
Latte	231623	222866	209654	213920
Mug	86718	91497	88474	93881
Scone	116127	70732	73779	72776
Thermos	46587	48870	48976	45648

TITLELINE=SKIP を使用すると、下線とブランク行の両方が除外されます。

	Region			
	Midwest	Northeast	Southeast	West
Product				
Biscotti	86105	145242	119594	70436
Capuccino	.	44785	73264	71168
Coffee Grinder	50393	40977	47083	48081
Coffee Pot	47156	46185	49922	47432
Croissant	139182	137394	156456	197022
Espresso	101154	68127	68030	71675
Latte	231623	222866	209654	213920
Mug	86718	91497	88474	93881
Scone	116127	70732	73779	72776
Thermos	46587	48870	48976	45648

データ識別ラベルの作成

ラベルを使用すると、汎用ラベルにより識別される次のレポート要素に、意味のある固有名を付けることができます。

- 行および列の合計
- ソートグループの中間合計
- 財務レポート内の行

行または列合計ラベルの作成

行または列合計のラベルは、2 つ以上のフィールドの値の合計を識別します。ラベルにより、合計が強調されます。1 つのレポート内に行および列合計がある場合、行または列合計のラベルを作成することが特に重要となります。

詳細は、合計と中間合計の追加を参照してください。

行または列合計のラベルの作成

```
fieldname [AND] ROW-TOTAL[/justification][/format] [AS 'label']
fieldname [AND] COLUMN-TOTAL[/justification] [AS 'label']
```

または

```
fieldname [AND] COLUMN-TOTAL[/justification] [AS 'label']
```

説明

fieldname

表示コマンド内で指定されたフィールドです。

justification

ラベルの位置揃えです。有効な値には、次のものがあります。

L - ラベルを左揃えにします。

R - ラベルを右揃えにします。

C - ラベルを中央揃えにします。

詳細は、見出し、脚注、タイトル、ラベルの位置揃えを参照してください。

format

行または列合計のフォーマットです。同一フォーマットのフィールドを合計する場合、合計のフォーマットはフィールドのフォーマットと同じになります。異なるフォーマットのフィールドを合計する場合、デフォルトの D12.2 が、行または列合計のいずれかに使用されます。

label

行または列合計のカスタマイズしたラベルです。デフォルトラベルは「TOTAL」です。

ON TABLE 句を使用して行合計または列合計を指定することもできます。この構文では、ROW-TOTAL を使用してフィールド名を含めることはできません。フィールド名は、COLUMN-TOTAL ではオプションです。

```
ON TABLE ROW-TOTAL[/justification][/format] [AS 'label']
ON TABLE COLUMN-TOTAL[/justification] [AS 'label']
 [fieldname fieldname fieldname]
```

リクエストの COMPUTE で作成したフィールドに対するクエリを発行する場合、フィールドの値は、行合計または列合計に含められます。合計を識別するラベルをカスタマイズする際は、このことに注意する必要があります。

行および列合計ラベルの作成

次のリクエストは、行合計ラベル「Total Population」、列合計ラベル「Total Population by Gender」を作成します。ROW-TOTAL の D12 フォーマットは、データをカンマ (,) 付きで表示します。

```
TABLE FILE GGDEMOG
PRINT MALEPOP98 FEMPOP98
ROW-TOTAL/D12 AS 'Total Population by State'
BY ST
WHERE (ST EQ 'WY' OR 'MT')
ON TABLE COLUMN-TOTAL AS 'Total Population by Gender'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

State	Male Population	Female Population	Total Population by State
MT	426357	438836	865,193
WY	245023	245897	490,920
Total Population by Gender	671380	684733	1,356,113

ACROSS による行合計ラベルの作成

次のリクエストは、2 つの州の人口の和を計算し、ACROSS 句を使用して情報をソート後、行合計に「Total by Gender」というラベルを追加します。「Total by Gender」フィールドには、「Male Population」および「Female Population」という 2 つの行合計があります。

```
TABLE FILE GGDEMOG
SUM MALEPOP98/D12 FEMPOP98/D12
ROW-TOTAL AS 'Total by Gender'
ACROSS ST
WHERE ST EQ 'WY' OR 'MT';
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

State					
MT		WY		Total by Gender	
Male Population	Female Population	Male Population	Female Population	Male Population	Female Population
426,357	438,836	245,023	245,897	671,380	684,733

中間合計ラベルおよび総合計ラベルの作成

レポートにソートグループの詳細情報が含まれており、グループの中間合計を表示し、すべてのグループの総合計をレポートの最後に追加すると便利な場合があります。

詳細は、合計と中間合計の追加を参照してください。

中間合計ラベルまたは総合計ラベルの作成

```
{BY|ON} fieldname {SUB-TOTAL|SUBTOTAL|COLUMN-TOTAL} [MULTILINES]
        [field1 [AND] field2...] [AS 'label'] [WHEN expression;]
```

説明

fieldname

BY または ON 句で指定されたソートフィールドです。

MULTILINES

MULTILINES が指定されると、ソート区切りの詳細行が 1 行のみの場合、中間合計を非表示にします。MULTI-LINES は MULTILINES と同義です。

field1 field2

中間合計を計算するフィールドを指定します。指定したフィールドは、すべての数値表示フィールドを含むデフォルトを上書きします。

AS 'label'

中間合計のカスタマイズしたラベルです。SUB-TOTALを使用している場合、それよりも上位のソートフィールドデフォルトラベルを変更することはできません。

WHEN expression

論理式を基にした条件付き中間合計を指定します。詳細は、式の使用を参照してください。

中間合計ラベルおよび総合計ラベルの作成

次のリクエストは、中間合計および総合計のカスタムラベルを作成します。中間合計は部門ごとに従業員の給与から市税を引いた合計額、総合計は両部門の合計額です。

```
TABLE FILE EMPLOYEE
SUM DED_AMT BY DED_CODE BY DEPARTMENT
BY BANK_ACCT
WHERE DED_CODE EQ 'CITY'
WHERE BANK_ACCT NE 0
ON DEPARTMENT SUBTOTAL AS 'Total City Deduction for'
ON TABLE COLUMN-TOTAL AS '**GRAND TOTAL**'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果で、カスタマイズした中間合計ラベルには、デフォルト設定で部門の値 MIS および PRODUCTION が含まれます。

DED_CODE	DEPARTMENT	BANK_ACCT	DED_AMT
CITY	MIS	40950036	$14.00
		122850108	$31.75
		163800144	$82.70
Total City Deduction for MIS			$128.45
	PRODUCTION	160633	$7.42
		136500120	$18.25
		819000702	$60.20
Total City Deduction for PRODUCTION			$85.87
GRAND TOTAL			$214.32

特定フィールド中間合計のラベルの作成

次のリクエストは、LINEPRICE の中間合計にカスタムラベルである「Order Total」を作成します。総合計にはデフォルトラベルの「縦計」が使用されます。

```
TABLE FILE CENTORD
PRINT PNUM QUANTITY LINEPRICE
BY ORDER_NUM SUBTOTAL LINEPRICE AS 'Order Total'
WHERE ORDER_NUM EQ '28003' OR '28004';
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Order Number:	Product Number#:	Quantity:	Line Total
28003	1006	1	$501.60
	1008	1	$239.18
	1020	1	$324.06
	1032	1	$103.23
	1034	1	$520.40
Order Total 28003			$1,688.47
28004	1006	1	$471.83
	1016	1	$282.19
	1018	1	$462.96
	1022	1	$401.30
	1028	1	$110.34
	1036	1	$295.00
Order Total 28004			$2,023.62
TOTAL			$3,712.09

一時項目 (COMPUTE) 中間合計のラベルの作成

```
{BY|ON} fieldname {SUMMARIZE|RECOMPUTE} [MULTILINES]
        [field1 [AND] field2...] [AS 'label'] [WHEN expression;]
ON TABLE {SUMMARIZE|RECOMPUTE}
```

説明

fieldname

BY または ON 句で指定されたソートフィールドです。

MULTILINES

MULTILINES が指定されると、ソート区切りの詳細行が 1 行のみの場合、中間合計を非表示にします。MULTI-LINES は MULTILINES と同義です。

field1 field2

中間合計を計算するフィールドを指定します。指定したフィールドは、すべての数値表示フィールドを含むデフォルトを上書きします。

AS 'label'

中間合計のカスタマイズしたラベルです。SUMMARIZE を使用している場合、それよりも上位のソートフィールドデフォルトラベルを変更することはできません。

WHEN expression

論理式を基にした条件付き中間合計を指定します。詳細は、式の使用を参照してください。

ON TABLE 句を使用して中間合計を生成することもできます。

```
ON TABLE {SUMMARIZE|RECOMPUTE}
```

一時項目 (COMPUTE) 中間合計のラベルの作成

次のリクエストは、COMPUTE を使用して作成したフィールド DG_RATIO の一時項目を含む、中間合計のカスタムラベルを作成します。

```
TABLE FILE EMPLOYEE
SUM GROSS DED_AMT AND COMPUTE
DG_RATIO/F4.2 = DED_AMT / GROSS;
BY DEPARTMENT BY BANK_ACCT
WHERE BANK_ACCT NE 0
ON DEPARTMENT SUMMARIZE AS 'SUBTOTAL FOR '
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果で、カスタマイズした中間合計ラベルには、デフォルト設定で部門の値 MIS および PRODUCTION が含まれます。デフォルトの総合計ラベルは、「縦計」です。

DEPARTMENT	BANK ACCT	GROSS	DED AMT	DG RATIO
MIS	40950036	$6,099.50	$2,866.18	.47
	122850108	$9,075.00	$6,307.00	.69
	163800144	$22,013.75	$15,377.40	.70
SUBTOTAL FOR MIS		$37,188.25	$24,550.58	.66
PRODUCTION	160633	$2,475.00	$1,427.24	.58
	136500120	$9,130.00	$3,593.92	.39
	819000702	$17,094.00	$11,949.44	.70
SUBTOTAL FOR PRODUCTION		$28,699.00	$16,970.60	.59
TOTAL		$65,887.25	$41,521.18	.63

財務レポートの行ラベル作成

FML (Financial Modeling Language) は、作成、演算、財務指向データの提供に関連する特別なニーズに対応します。FML レポートは、行ごとに構築されます。この構成により、レポートに組み込むデータおよびその外観が制御しやすくなります。

行はラベルで識別され、ラベルをカスタマイズすることで、データの正確な特定、フォーマット設定による外観の向上、データの明確化が可能になります。

見出し、脚注、タイトル、ラベルのフォーマット設定

レポートの構成要素を拡張するには、さまざまな方法があります。構成要素には、見出し、脚注、行合計と列合計、行列の合計や中間合計に割り当てるラベルなどがあります。これらを追加すると、レポートの視覚的訴求力を高め、完成度の高さを伝達することができます。フォントと色で強調と区別ができ、位置揃えで構造を明確にして理解しやすくできます。

次の処理を実行することができます。

- 行および列のタイトル、見出し、脚注、要素、中間合計、総合計、中間合計演算にフォント特性を適用する。

- 見出しおよび脚注の周囲に境界線を、見出し、脚注、フィールドタイトルの周囲に罫線を追加する。詳細は、レポートページのレイアウトを参照してください。

  見出しまたは脚注の内容と罫線の間にブランクを追加する。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。

- 見出しと脚注 (複数行見出しと脚注の各行)、フィールドタイトル、中間合計に、左、右、中央の位置揃えを設定する。詳細は、レガシーフォーマットを使用したページ見出しまたは脚注の中央揃えおよび 見出し、脚注、タイトル、ラベルの位置揃えを参照してください。

  位置揃えは、行合計、列合計のラベルに設定することもできます。詳細は、見出し、脚注、タイトル、ラベルの位置揃えを参照してください。

- 見出し、脚注、またはそれらの要素の位置を揃える。

  - 位置揃えは、レポートの主 HTML テーブルのデータ列を基準にするか、見出しまたは脚注用に作成された埋め込み HTML テーブル内の列を基準にします。これらの位置揃え方法は、HTML レポートでサポートされます。詳細は、HTML、XLSX、EXL2K、PDF、PPTX、DHTML レポートの見出しまたは脚注の位置揃えを参照してください。

  - 複数行の見出しおよび脚注の幅と配置を明示的に指定する。たとえば、測定単位 (例、インチ) を指定して、数値データや文字データ列を積み重ねて表示する際に小数点の位置を基準に位置揃えします。これらの位置揃え方法は、HTML および PDF レポートでサポートされます。

  - 単位 (インチなど) または桁数で、絶対あるいは相対開始位置を定義する。これらの位置揃え方法は、PDF レポートでサポートされます。内部カスケードスタイルシートとともに使用する場合は、機能のいくつかは、HTML でもサポートされます。

  ユーザのニーズに最適な方法の選定については、見出しおよび脚注の位置揃えを参照してください。

- 見出しまたは脚注の縦方向の配置を制御する。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。

- レポート、ソート項目を指定している見出しまたは脚注の配置を、ページごとに設定する。詳細は、レポート見出しまたは脚注の単独ページ配置を参照してください。

見出し、脚注、タイトル、ラベルのフォント属性適用

スタイルシート内で識別可能な任意のレポート要素に、フォントファミリー、サイズ、色、スタイルを指定することができます。

- 行および列のタイトル。スタイルは、デフォルトタイトルまたはカスタマイズしたタイトルのいずれか一方に適用されます。
- 複数行の見出しまたは脚注内の特定行、行内の項目、テキスト、埋め込みフィールドを含む見出しと脚注、および埋め込み要素。背景色は、要素ごとに指定することもできます。
- 中間合計、総合計、中間合計演算、行合計のラベル。スタイルは、デフォルトまたはカスタマイズした名前に適用されます。
- 見出しまたは脚注内のページ番号。
- 下線およびブランク行 (HTML レポートではサポートされません)。

構文についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。サイズ、色、およびスタイルなどのフォントオプションについての詳細は、レポートデータのフォーマットを参照してください。

レポート見出しおよびフィールドタイトルへのフォント特性の適用

次のリクエストは、スタイルシートを使用して、レポート見出し (Sales Report) に 12 ポイント、Arial、太字を、デフォルトフィールドタイトル (Category Product、Unit Sales、Dollar Sales) に 10 ポイント、Arial、斜体を選択します。HTML ポイントスケールを基準としており、標準ポイントサイズとは異なります。詳細は、レポートデータのフォーマットを参照してください。

HTML レポートでは、フォント名を一重引用符 (') で囲む必要があります。スタイルシートの TYPE = TABHEADING 属性はレポート見出し、TYPE = TITLE 属性はフィールドタイトルをそれぞれ識別します。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS
```

```
BY CATEGORY BY PRODUCT
ON TABLE SUBHEAD
"Sales Report"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID=OFF, $
TYPE = TABHEADING, FONT = 'ARIAL', SIZE = 12, STYLE = BOLD, $
TYPE = TITLE, FONT = 'ARIAL', SIZE = 10, STYLE = ITALIC, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Sales Report

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

内部カスケードスタイルシートを使用したレポート見出しフォントサイズの設定

内部カスケードスタイルシートでは、対応する HTML ポイントサイズスケールではなく、絶対ポイントサイズを指定できます。その結果、レポートのフォントの制御がより詳細に行えます。詳細は、レポートデータのフォーマットおよび レポートフォーマット設定の管理を参照してください。

このリクエストは、内部カスケードスタイルシートを生成し、レポート見出しのフォント特性を指定します。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS
BY CATEGORY BY PRODUCT
ON TABLE SUBHEAD
"Sales Report"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID=OFF, $
TYPE = TABHEADING, FONT = 'ARIAL', SIZE = 12, STYLE = BOLD, $
TYPE = TITLE, FONT = 'ARIAL', SIZE = 10, STYLE = ITALIC, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Sales Report

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

レポート見出しシステム変数へのフォントスタイルの適用

次のリクエストでは、見出しにシステム変数 &DATE が含まれています。太字が適用されるそれ以外の見出しテキストと区別するため、斜体スタイルが適用されます。スポットマーカー <+0> は、それぞれ個別にフォーマットを設定できるよう、見出し内に 2 つの項目を作成します。

```
TABLE FILE GGSALES
PRINT BUDDOLLARS DOLLARS
BY STCD
WHERE STCD EQ 'R1019'
ON TABLE SUBHEAD
"Sales Report for Store Code R1019 <+0>&DATE"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF,$
TYPE=TABHEADING, FONT='TIMES', SIZE=10, STYLE=BOLD,$
TYPE=TABHEADING, ITEM=2, STYLE=ITALIC,$
ENDSTYLE
END
```

以下は、出力結果の一部を示しています。

Sales Report for Store Code R1019 *06/13/02*

Store ID	Budget Dollars	Dollar Sales
R1019	18560	25680
	21132	24010
	15880	22919
	24450	22605
	14556	18942
	13860	18915
	22806	18837
	19248	18535
	15120	18466
	15444	17784
	15190	17644
	12386	15996
	12720	14546
	17370	14328

境界線および罫線の追加

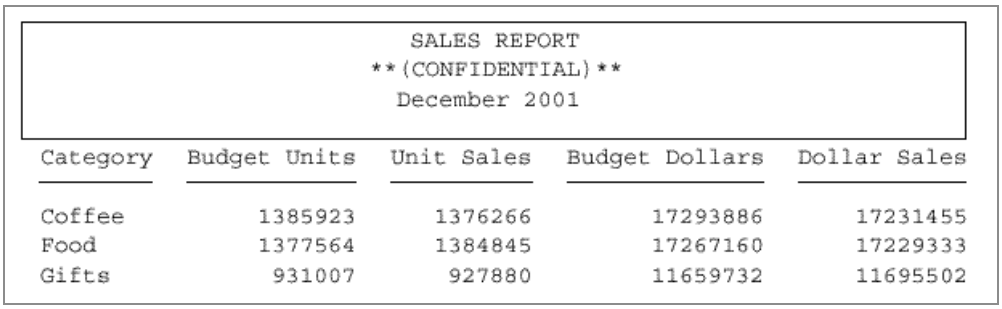
見出し、脚注、タイトル、ラベルに、境界線および罫線を追加することができます。構文についての詳細は、レポートページのレイアウトを参照してください。

PDF レポート見出し周囲への罫線の追加

次のリクエストは、見出しの周囲に罫線を表示する PDF レポートを生成します。罫線は、GRID 属性で作成され、レポートの見出しと本文を区切ります。

```
TABLE FILE GGSALES
SUM BUDUNITS UNITS BUDDOLLARS DOLLARS
BY CATEGORY
ON TABLE SUBHEAD
"SALES REPORT"
"**(CONFIDENTIAL)**"
"December 2001 </1"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT PDF
ON TABLE SET SQUEEZE ON
ON TABLE SET STYLESHEET *
TYPE = TABHEADING, JUSTIFY = CENTER, GRID=ON, $
ENDSTYLE
END
```

出力結果は次のとおりです。



SALES REPORT
(CONFIDENTIAL)
December 2001

Category	Budget Units	Unit Sales	Budget Dollars	Dollar Sales
Coffee	1385923	1376266	17293886	17231455
Food	1377564	1384845	17267160	17229333
Gifts	931007	927880	11659732	11695502

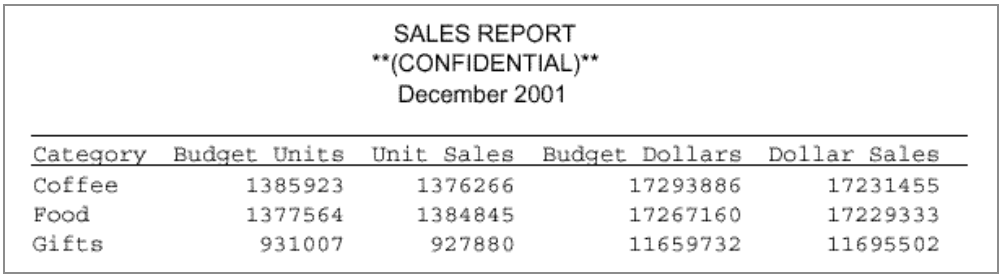
PDF レポートフィールドタイトルの横線強調

次のリクエストは、PDF レポートを作成し、HGRID 属性で作成した横線をフィールドタイトルの上下に追加します。

```
TABLE FILE GGSALES
SUM BUDUNITS UNITS BUDDOLLARS DOLLARS
BY CATEGORY
ON TABLE SUBHEAD
"SALES REPORT"
"**(CONFIDENTIAL)**"
```

```
"December 2001 </1"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT PDF
ON TABLE SET SQUEEZE ON
ON TABLE SET STYLESHEET *
TYPE = TABHEADING, JUSTIFY = CENTER, FONT=ARIAL, SIZE=12, $
TYPE = TITLE, HGRID=ON, $
END
```

出力結果は次のとおりです。



SALES REPORT
(CONFIDENTIAL)
December 2001

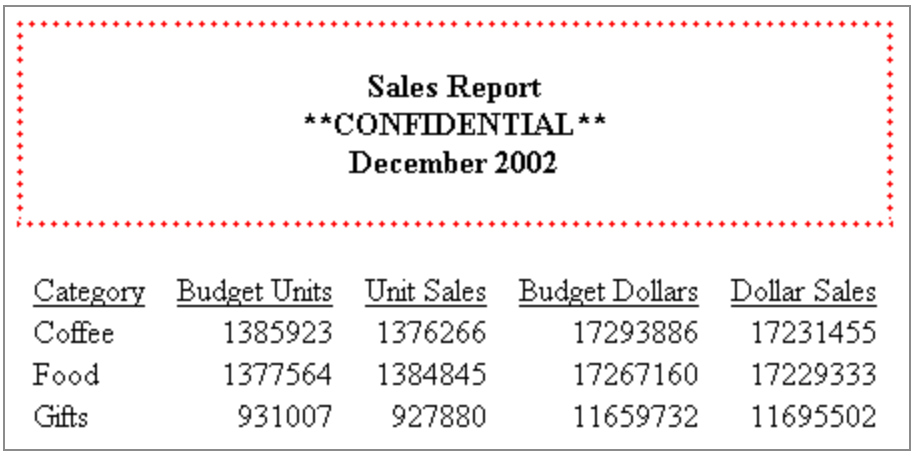
Category	Budget Units	Unit Sales	Budget Dollars	Dollar Sales
Coffee	1385923	1376266	17293886	17231455
Food	1377564	1384845	17267160	17229333
Gifts	931007	927880	11659732	11695502

レポート見出し周囲境界線のフォーマット設定

次のリクエストは、レポートの見出し全体を太い赤の点線で囲んだ HTML レポートを生成します。

```
TABLE FILE GGSALES
SUM BUDUNITS UNITS BUDDOLLARS DOLLARS
BY CATEGORY
ON TABLE SUBHEAD
"</1 Sales Report"
"**CONFIDENTIAL**"
"December 2002 </1"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=TABHEADING, STYLE=BOLD, JUSTIFY=CENTER, BORDER=HEAVY,
     BORDER-COLOR=RED, BORDER-STYLE=DOTTED, $
ENDSTYLE
END
```

出力結果は次のとおりです。



Sales Report
CONFIDENTIAL
December 2002

Category	Budget Units	Unit Sales	Budget Dollars	Dollar Sales
Coffee	1385923	1376266	17293886	17231455
Food	1377564	1384845	17267160	17229333
Gifts	931007	927880	11659732	11695502

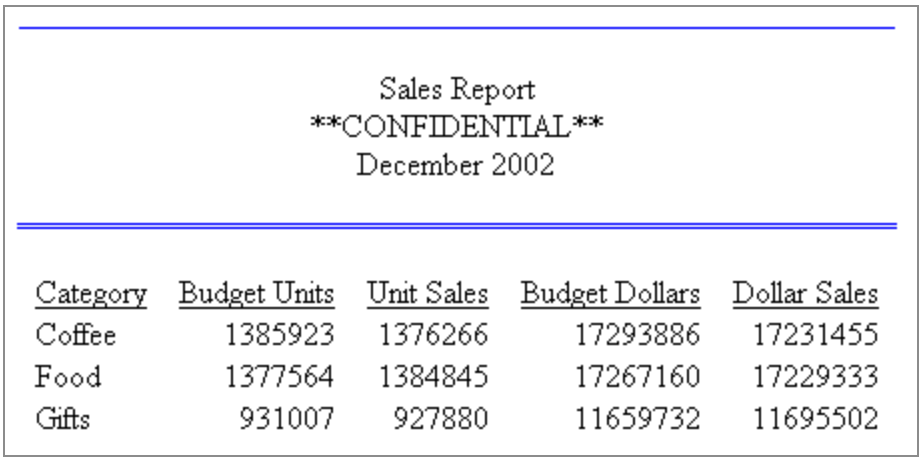
ヒント：PDF レポート内でこの出力を生成するために、同じ BORDER 構文を使用できます。

上下境界線によるレポート見出しフォーマット設定

次のリクエストは、見出しの上に水色の線、見出しの下に紺色の二重線を生成します。このリクエストでは、見出しの左右の境界線を指定しません。

```
TABLE FILE GGSALES
SUM BUDUNITS UNITS BUDDOLLARS DOLLARS
BY CATEGORY
ON TABLE SUBHEAD
"</1 Sales Report"
"**CONFIDENTIAL**"
"December 2002 </1"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=TABHEADING, JUSTIFY=CENTER, BORDER-TOP=LIGHT, BORDER-COLOR=BLUE,
       BORDER-BOTTOM=HEAVY, BORDER-BOTTOM-STYLE=DOUBLE,$
ENDSTYLE
END
```

出力結果は次のとおりです。



Sales Report
CONFIDENTIAL
December 2002

Category	Budget Units	Unit Sales	Budget Dollars	Dollar Sales
Coffee	1385923	1376266	17293886	17231455
Food	1377564	1384845	17267160	17229333
Gifts	931007	927880	11659732	11695502

ヒント：PDF レポート内でこの出力を生成するために、同じ BORDER 構文を使用できます。

見出し、脚注、タイトル、ラベルの位置揃え

次のレポート要素は、左、右、中央のいずれかに位置を揃えることができます。

- 見出しおよび脚注。
- フィールドタイトル。
- 行合計ラベルおよび列合計ラベル。
- 中間合計ラベルおよび総合計ラベル。

また、見出し、脚注の配置には、スタイルシート構文と位置揃え構文を組み合わせて使用することができます。その際、測定単位や他のレポート要素 (フィールドなど) との関係を基準にすることができます。これらのオプションの概要については、見出しおよび脚注の位置揃えを参照してください。

見出しおよび脚注の位置揃え

スタイルシートで見出しまたは脚注の位置を左揃え、右揃え、中央揃えにすることができます。デフォルト設定では、見出しおよび脚注は左揃えされます。さらに、複数行の見出しまたは脚注内では、個別の行、または複数行を同時に揃えることができます。

レポートデータのページ見出しまたは脚注を中央に揃えるには、スタイルシートが不要な従来のフォーマット設定方法を使用して、単純に、HEADING または FOOTING コマンドに CENTER コマンドを追加します。

HTML および PDF の位置揃え動作　HTML レポートでは、位置揃えはレポートの幅に基づいて実行されます。その結果、見出しはレポートコンテンツに対して中央揃えされます。これに対して、PDF レポートでは、デフォルトの位置揃え領域がレポート幅ではなくページ幅であるため、見出しや脚注はレポートの中央には配置されません。ほとんどの場合、リクエストに SET SQUEEZE=ON コマンドを追加することで、PDF レポートの位置揃え基準をレポート幅にすることができます。このコマンドは、フィールド間の余分なブランク部分を取り除き、レポートコンテンツに基づいた位置揃えを実行することにより、レポートの外観を整えます。ただし、見出しがレポート幅よりも広いときは、SQUEEZE=ON に設定されている場合でも、ページの中央に配置されます。

ヒント：位置揃え構文をスタイルシート構文と併用することで、見出し、脚注、見出し脚注項目を、他のレポート要素と位置揃えすることもできます。位置揃えの基準には、測定単位、または他のフィールドを使用することができます。これらのオプションの概要については、見出しおよび脚注の位置揃えを参照してください。

スタイルシートでの見出しまたは脚注の位置揃え

```
TYPE = headfoot, [LINE = line_#,] JUSTIFY = option, $
```

説明

headfoot

見出しまたは脚注のタイプです。有効な値は、TABHEADING、TABFOOTING、HEADING、FOOTING、SUBHEAD、SUBFOOT です。

line_#

必要に応じて見出しまたは脚注内の行の位置によりその行を識別します。これにより個別に位置揃えが可能です。見出しまたは脚注が複数行で、このオプションを省略する場合、JUSTIFY で与えた値はすべての行に適用されます。

option

位置揃えのタイプです。有効な値には、次のものがあります。

LEFT - 見出しまたは脚注を左揃えにします。デフォルト値は LEFT です。

RIGHT - 見出しまたは脚注を右揃えにします。

CENTER - 見出しまたは脚注を中央揃えにします。

スタイルシートを使用せずにページを中央揃えする別の方法についての詳細は、見出し、脚注、タイトル、ラベルの位置揃えを参照してください。

注意：WRAP とともに JUSTIFY を使用することはできません。

レポートの見出しの位置揃え

次のリクエストは、属性 JUSTIFY = CENTER を使用して、レポートの PRODUCT REPORT 見出しを中央に揃えます。

```
TABLE FILE GGPRODS
SUM UNITS BY PRODUCT_DESCRIPTION BY PRODUCT_ID BY VENDOR_NAME
ON TABLE SUBHEAD
"PRODUCT REPORT"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID=OFF, $
TYPE = REPORT, COLUMN = VENDOR_NAME, JUSTIFY = CENTER, $
TYPE = TABHEADING, JUSTIFY = CENTER, $
ENDSTYLE
END
```

出力結果は次のとおりです。

PRODUCT REPORT

Product	Product Code	Vendor Name	Unit Price
Biscotti	F102	Delancey Bakeries	17.00
Coffee Grinder	G110	Appliance Craft	125.00
Coffee Pot	G121	Appliance Craft	140.00
Croissant	F103	West Side Bakers	28.00
French Roast	B142	European Specialities,	81.00
Hazelnut	B141	Coffee Connection	58.00
Kona	B144	Evelina Imports, Ltd	76.00
Mug	G100	NY Ceramic Supply	26.00
Scone	F101	Ridgewood Bakeries	13.00
Thermos	G104	ThermoTech, Inc	96.00

ヒント：このレポートを PDF フォーマットで実行する場合は、ON TABLE SET SQUEEZE ON コードを追加することで、フィールド間の余分なブランク部分を取り除き、見出しをレポートの中央に揃えます。

フィールドタイトル位置揃えについての詳細は、見出し、脚注、タイトル、ラベルの位置揃えを参照してください。

複数行レポート見出しの個別位置揃え

次のリクエストでは、1 行目の見出し (SALES REPORT) と 2 行目の見出し (**CONFIDENTIAL**) は、ともに中央に、3 行目の見出し (December 2001) は右に配置されます。

```
TABLE FILE GGSALES
SUM BUDUNITS UNITS BUDDOLLARS DOLLARS
BY CATEGORY
ON TABLE SUBHEAD
"SALES REPORT"
"**(CONFIDENTIAL)**"
"December 2001"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID=OFF, $
TYPE = TABHEADING, LINE = 1, JUSTIFY = CENTER, $
TYPE = TABHEADING, LINE = 2, JUSTIFY = CENTER, $
```

```
TYPE = TABHEADING, LINE = 3, JUSTIFY = RIGHT, $
ENDSTYLE
END
```

出力結果は次のとおりです。

SALES REPORT
(CONFIDENTIAL)
December 2001

Category	Budget Units	Unit Sales	Budget Dollars	Dollar Sales
Coffee	1385923	1376266	17293886	17231455
Food	1377564	1384845	17267160	17229333
Gifts	931007	927880	11659732	11695502

ヒント：このレポートを PDF フォーマットで実行する場合は、ON TABLE SET SQUEEZE ON コードを追加することで、フィールド間の余分なブランク部分を取り除き、見出しをレポートの中央に揃えます。

複数行レポート見出し全行の中央揃え

次のリクエストは、すべての見出しに単一スタイルシート属性を使用して、複数行のレポートの見出し内ですべての行を中央に揃えます。

```
TABLE FILE GGSALES
SUM BUDUNITS UNITS BUDDOLLARS DOLLARS
BY CATEGORY
ON TABLE SUBHEAD
"SALES REPORT"
"**(CONFIDENTIAL)**"
"December 2001"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID=OFF, $
TYPE = TABHEADING, JUSTIFY = CENTER, $
ENDSTYLE
END
```

出力結果は次のとおりです。

SALES REPORT
(CONFIDENTIAL)
December 2001

Category	Budget Units	Unit Sales	Budget Dollars	Dollar Sales
Coffee	1385923	1376266	17293886	17231455
Food	1377564	1384845	17267160	17229333
Gifts	931007	927880	11659732	11695502

ヒント：このレポートを PDF フォーマットで実行する場合は、ON TABLE SET SQUEEZE ON コードを追加することで、フィールド間の余分なブランク部分を取り除き、見出しをレポートの中央に揃えます。

位置揃えの範囲および動作

テキストが位置揃えされる範囲は、レポート内のある要素のサイズの関係により決まります。

- SQUEEZE=ON に設定されている場合、レポート内に存在する見出しすべての最大幅が計算されます。この値は、「MaxHeadWidth」と呼ばれます。

  MaxHeadWidth がレポートフィールドの合計幅以下の場合、見出しはレポートフィールドのスペース内で位置揃えされます。

  MaxHeadWidth がレポートフィールド合計幅を超える場合、見出しはページ全体幅の中央に配置され、右揃えされます。

- SQUEEZE=OFF に設定されている場合、見出しすべての幅の事前計算は行われません。見出しはページ幅全体の中央に配置されます。

  スタイルの設定された複数パネルレポート (幅が 1 ページを超える) では、見出しは最初のパネルのみに表示されます。このため、事前計算の対象は、レポート内の全フィールド合計幅ではなく、最初のパネル内のフィールド合計幅になります。

レガシーフォーマットを使用したページ見出しまたは脚注の中央揃え

```
{HEADING|FOOTING} CENTER
 "content ... "
["content ... "]
.
.
.
["content ... "]
```

説明

HEADING

ページ見出しです。

FOOTING

ページ脚注です。

CENTER

レポートデータの上部のページ見出し、または下部のページ脚注を中央揃えにします。

content

見出しまたは脚注の内容には、次の要素を二重引用符 (") で囲んで含めることができます。終了用引用符が省略された場合、リクエストの後に続くすべての行は、見出しまたは脚注の一部分として扱われます。

text

見出しまたは脚注のテキストです。複数行のテキストを含めることができます。

テキストは、テキストのみの行で開始する必要があり、HEADING または FOOTING コマンドに続きます。

テキストは、変数およびスポットマーカーと組み合わせることができます。

variable

次の 1 つまたは組み合わせです。

`Fields` - 実データソースフィールド、マスターファイルまたはリクエスト内で DEFINE コマンドを使用して作成した一時項目 (DEFINE)、リクエスト内で COMPUTE コマンドを使用して作成した一時項目 (COMPUTE)、または TABPAGENO などのシステムフィールドです。データソースフィールドは、特定の演算接頭語で修飾することができます。

```
Dialogue Manager variables.
```

`Images` - 見出しまたは脚注にイメージを含めることができます。

詳細は、見出しまたは脚注内への要素の追加を参照してください。

スポットマーカー

項目の配置、フォーマット設定項目の識別が可能です。

`<+0>` - フォーマット設定を目的として、見出しまたは脚注を項目に分割します。

`</n` - ブランク行を指定します。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。

`<-n` - 行に次の文字を配置します。詳細は、見出し、脚注、その内部の項目の配置を参照してください。

`<0X` - 見出しまたは脚注指定を、リクエストの次の行に継続します。

注意：終了スポットマーカーの直後に開始スポットマーカーを追加すると (><)、この 2 つのスポットマーカー間にブランクが 1 つ挿入されます (> <)。フォーマット設定を適用する際は、このことを考慮する必要があります。

ブランク行

すべてのテキスト、変数、スポットマーカーを省略する場合、見出しや脚注がブランクになります (例、" ")。これを使用して、ブランク行を挿入することができます。ブランク行を挿入するには、</1 などの縦のスポットマーカーを使用することもできます。

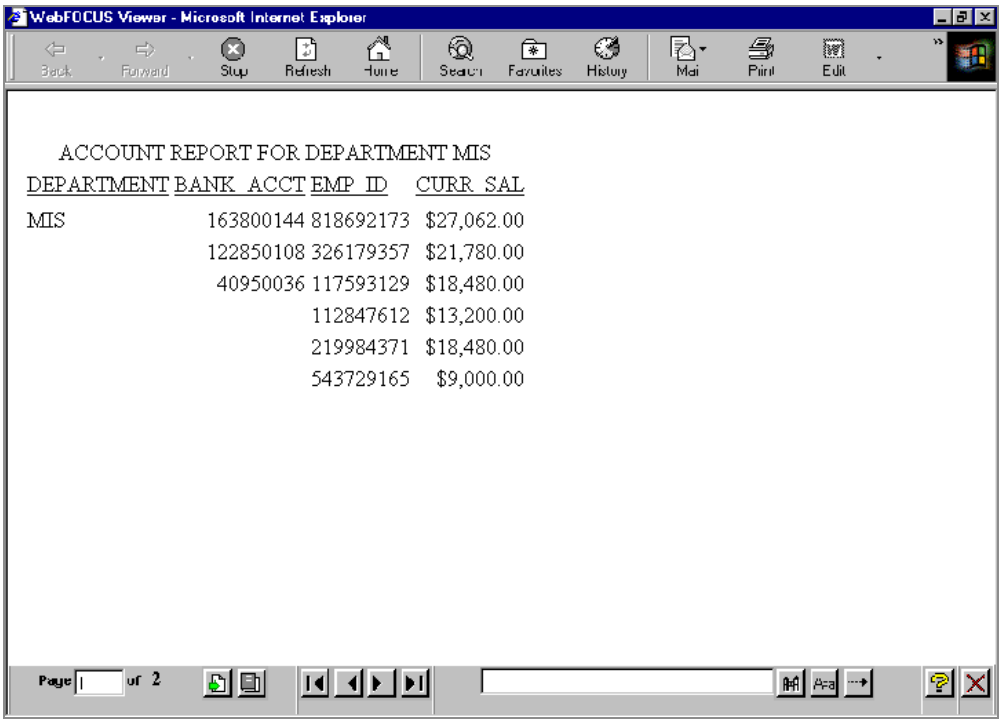
ヒント：スタイルシート属性 JUSTIFY = CENTER とともにコマンド CENTER を使用することはできません。1 つの方法で目的の結果が生成されます。

ページ見出しの中央揃え

次のリクエストは、ページ見出し構文で CENTER コマンドを使用します。

```
TABLE FILE EMPLOYEE
HEADING CENTER
"ACCOUNT REPORT FOR DEPARTMENT <DEPARTMENT"
PRINT CURR_SAL BY DEPARTMENT BY HIGHEST BANK_ACCT
BY EMP_ID
ON DEPARTMENT PAGE-BREAK
ON TABLE SET PAGE-NUM OFF
ON TABLE SET WEBVIEWER ON
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, SIZE=10, GRID=OFF,$
ENDSTYLE
END
```

出力結果の最初のページが示すように、ページ見出しは、レポートデータで中央揃えされます。



ヒント：ナビゲーションの矢印が表示されない場合、最大化ボタンをクリックします。

フィールドタイトルの位置揃え

スタイルシートを使用して、表示フィールド、BY フィールド、ACROSS フィールド、または一時項目 (COMPUTE) のフィールドタイトルを左揃え、右揃え、中央揃えにすることができ

ます。

リクエストの AS 句またはマスターファイルの TITLE 属性でタイトルが指定されている場合、そのタイトルはスタイルシート構文でフィールドに対して指定されたとおりに位置揃えされます (リクエストにこの構文が存在する場合)。詳細は、フィールドタイトルのカスタマイズを参照してください。

HTML および PDF の位置揃え動作　HTML レポートでは、位置揃えはレポートの幅に基づいて実行されます。このため、中央揃えが設定されたフィールドタイトルは、レポートフィールドの中央に配置されます。これに対して、PDF レポートでは、デフォルトの位置揃え領域がレポート幅ではなくページ幅であるため、フィールドタイトルはレポートフィールドの中央には配置されません。リクエストに SET SQUEEZE=ON コマンドを追加することで、PDF レポートの位置揃え基準に、レポート幅を使用することができるようになります。このコマンドは、フィールド間の余分なブランク部分を取り除き、レポートコンテンツに基づいた位置揃えを実行することにより、レポートの外観を整えます。

レガシーフォーマットを使用して、表示または BY フィールドのフィールドタイトルの位置を揃えることもできます。ただし、レガシーフォーマットを ACROSS フィールドに適用する場合は、位置揃えの対象は、フィールドタイトルではなく、データ値になります。詳細は、見出し、脚注、タイトル、ラベルの位置揃えを参照してください。

スタイルシートを使用したフィールドタイトルの位置揃え

BY フィールドのフィールドタイトル、表示フィールド (PRINT、LIST、SUM、COUNT) の位置を揃えるには、次のスタイルシート構文を使用します。

```
TYPE=TITLE, [COLUMN=column,] JUSTIFY=option, $
TYPE=ACROSSTITLE, [ACROSS=column,] JUSTIFY=option, $
TYPE=ACROSSVALUE, [COLUMN=column,] JUSTIFY=option, $
```

ACROSS フィールドの位置を揃えるには、次のスタイルシート構文を使用します。

```
TYPE=ACROSSTITLE, [ACROSS=column,] JUSTIFY=option, $
```

HTML レポート内で ACROSS または ROW-TOTAL フィールドの位置を揃えるには、次の構文を使用します。

```
TYPE=ACROSSVALUE, [COLUMN=column,] JUSTIFY=option, $
```

説明

TITLE

BY フィールドのタイトルまたは表示フィールドのタイトルを指定します。

column

位置揃えの対象となるフィールドを指定します。この属性および値を省略すると、フォーマットはレポートフィールドタイトルすべてに適用されます。フィールドの識別についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

ACROSSTITLE

ACROSS フィールドのタイトルを指定します。

ACROSSVALUE

ACROSS フィールドの値または ROW-TOTAL フィールドタイトルを指定します。

option

位置揃えのタイプです。有効な値には、次のものがあります。

`LEFT` フィールドタイトルを左揃えにします。この値は、文字フィールドのデフォルトです。

`RIGHT` フィールドタイトルを右揃えにします。この値は、数値または日付フィールドのデフォルトです。

`CENTER` フィールドタイトルを中央揃えにします。PDF レポート内で ACROSSTITLE を中央揃えにすることはできません。

注意：WRAP とともに JUSTIFY を使用することはできません。

表示フィールドおよび BY フィールドのタイトル位置揃え

次のリクエストは、STORE_NAME および ADDRESS1 のフィールドタイトルを中央に揃えます。STORE_NAME のデフォルトフィールドタイトルは、マスターファイル内で指定された 'Store Name' です。ADDRESS1 のデフォルトフィールドタイトルは、マスターファイルで指定された「Contact」です。このリクエストは、STATE フィールドを右揃えにします。この指定は、AS 句で St として指定されます。各フィールドはフィールド名により識別され、個別に位置揃えされます。

```
TABLE FILE GGSTORES
PRINT STORE_NAME STATE AS 'St' BY ADDRESS1
WHERE STATE EQ 'CA'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET SQUEEZE ON
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID=OFF, $
TYPE=TITLE, COLUMN=STORE_NAME, JUSTIFY=CENTER, $
TYPE=TITLE, COLUMN=STATE, JUSTIFY=RIGHT, $
TYPE=TITLE, COLUMN=ADDRESS1, JUSTIFY=CENTER, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Contact	Store Name	St
JEFF DARFELL	GOTHAM GRINDS #1244	CA
PAT SMILEY	GOTHAM GRINDS #1040	CA

スタイルシートを使用した ACROSS および ROW-TOTAL フィールドタイトルの位置揃え

次のリクエストは、2 つの値 (MT と WY) で構成される ACROSS フィールドのタイトル「State」、および 2 つの行合計 (「Male Population」、「Female Population」) のフィールドタイトル「Total by Gender」を中央に揃えます。レポートで、ACROSS 値はそれぞれ 1 つ以上のフィールドタイトルになります。

```
TABLE FILE GGDEMOG
SUM MALEPOP98 FEMPOP98
ROW-TOTAL/D12 AS 'Total by Gender'
ACROSS ST
WHERE ST EQ 'WY' OR 'MT';
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=ACROSSTITLE, JUSTIFY=CENTER, FONT='TIMES', SIZE=11, STYLE=BOLD, $
```

```
TYPE=ACROSSVALUE, COLUMN=N5, JUSTIFY=CENTER, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	State				
MT		WY		Total by Gender	
Male Population	Female Population	Male Population	Female Population	Male Population	Female Population
426357	438836	245023	245897	671380	684733

一時項目 (COMPUTE) フィールドタイトルの位置揃え

次のリクエストは、一時項目 (COMPUTE) のフィールドタイトルを識別し、データ内で左揃えします。

```
TABLE FILE SALES
SUM UNIT_SOLD RETAIL_PRICE
COMPUTE REV/D12.2M = UNIT_SOLD * RETAIL_PRICE;
BY PROD_CODE
WHERE CITY EQ 'NEW YORK'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=TITLE, COLUMN=REV, STYLE=BOLD, JUSTIFY=LEFT, $
ENDSTYLE
END
```

出力結果は次のとおりです。

PROD_CODE	UNIT_SOLD	RETAIL_PRICE	REV
B10	30	$.85	$25.50
B17	20	$1.89	$37.80
B20	15	$1.99	$29.85
C17	12	$2.09	$25.08
D12	20	$2.09	$41.80
E1	30	$.89	$26.70
E3	35	$1.09	$38.15

注意：このレポートを PDF フォーマットで実行する場合は、ON TABLE SET SQUEEZE ON コードを追加することで、フィールド間の余分なブランク部分を取り除き、フィールドタイトルをレポートの中央に揃えます。

レガシーフォーマットを使用した表示フィールドまたは BY フィールドタイトルの位置揃え

```
fieldname/justification [/format] [AS 'title']
```

説明

fieldname

フィールド名です。

justification

位置揃えのタイプです。有効な値には、次のものがあります。

L　フィールドタイトルを左揃えにします。この値は、文字フィールドのデフォルトです。

R　フィールドタイトルを右揃えにします。この値は、数値または日付フィールドのデフォルトです。

C　フィールドタイトルを中央揃えにします。

/format

フィールドフォーマットのオプション指定です。表示フィールドでは、フィールドデータの幅の調整または表示オプションを指定するために、位置揃えの値と形式の値 (どちらの順序でも可) を組み合わせることができます。

AS 'title'

オプションのカスタムフィールドタイトルです。

ヒント：ACROSS フィールドでは、この構文はフィールドタイトルではなく、データ値に対して有効となります。タイトルの位置揃え構文についての詳細は、見出し、脚注、タイトル、ラベルの位置揃えを参照してください。

レガシーフォーマットを使用した表示フィールドおよび BY フィールドタイトルの位置揃え

次のリクエストは、STORE_NAME および ADDRESS1 のフィールドタイトルを中央に揃えます。STORE_NAME のデフォルトフィールドタイトルは、マスターファイル内で指定された 'Store Name' です。ADDRESS1 のデフォルトフィールドタイトルは、マスターファイルで指定された「Contact」です。また、このリクエストでは、STATE のフィールドタイトル「St」 (AS で指定) を右揃えにします。

```
TABLE FILE GGSTORES
PRINT STORE_NAME/C STATE/R AS 'St' BY ADDRESS1/C
WHERE STATE EQ 'CA'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Contact	Store Name	St
JEFF DARFELL	GOTHAM GRINDS #1244	CA
PAT SMILEY	GOTHAM GRINDS #1040	CA

注意：PDF フォーマットを使用している場合は、ON TABLE SET SQUEEZE ON 構文をリクエストに追加します。

行合計ラベルまたは列合計ラベルの位置揃え

行合計ラベルまたは列合計ラベルを左揃え、右揃え、中央揃えにすることができます。詳細は、データ識別ラベルの作成を参照してください。

レガシーフォーマットを使用した行合計ラベルまたは列合計ラベルの位置揃え

```
ROW-TOTAL/justification [/format] [AS 'label']
COLUMN-TOTAL/justification [AS 'label']
```

または

```
COLUMN-TOTAL/justification [AS 'label']
```

説明

justification

位置揃えのタイプです。有効な値には、次のものがあります。

L - ラベルを左揃えにします。

R - ラベルを右揃えにします。

C - ラベルを中央揃えにします。

/format

行合計のオプションのフォーマット指定です。フィールドデータの幅を調整または表示オプションを指定するために、位置揃え値とフォーマット値 (どちらの順序でも可) を組み合わせることができます。

AS 'label'

オプションのカスタマイズしたラベルです。

行合計ラベルの作成

次のリクエストは、行合計ラベル「Total Population by State」を作成し、ラベルを中央に揃えます。ROW-TOTAL の D12 フォーマットは、デフォルト設定でカンマ (,) を表示します。

```
TABLE FILE GGDEMOG
PRINT MALEPOP98 FEMPOP98
ROW-TOTAL/C/D12 AS 'Total,Population,by State'
BY ST
WHERE (ST EQ 'WY' OR 'MT')
ON TABLE COLUMN-TOTAL AS 'Total by Gender'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

State	Male Population	Female Population	Total Population by State
MT	426357	438836	865,193
WY	245023	245897	490,920
Total by Gender	671380	684733	1,356,113

中間合計ラベルまたは総合計ラベルの位置揃え

カスタマイズされた中間合計ラベルを直接位置揃えすることはできません。ただし、HTML、EXL2K、XLSX レポート出力では、フィールドの合計または中間合計に使用する中間合計コマンドが 1 つのみで、スタイルシートでフィールドを指定しない場合は、すべてのフィールドの合計、中間合計、これらのラベルテキストにフォーマットを適用することができます。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

中間合計ラベルまたは総合計ラベルの位置揃え

次のリクエストは、レポートの数値フィールドの中間合計を計算し、中間合計値の前のラベルテキストを含めて、出力結果を右揃えします。数値出力結果はデフォルト設定で右揃えされるため、この例では、スタイルシートの位置揃え指定を使用して、ラベルを再配置しています。総合計 (GRANDTOTAL) のラベルは、自動的に右揃えされます。

```
TABLE FILE EMPLOYEE
SUM DED_AMT BY DED_CODE BY DEPARTMENT
BY BANK_ACCT
WHERE DED_CODE EQ 'CITY'
WHERE BANK_ACCT NE 0
ON DEPARTMENT SUBTOTAL AS 'Total City Deduction for'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE=SUBTOTAL, STYLE=BOLD, JUSTIFY=RIGHT,$
TYPE=GRANDTOTAL, STYLE=BOLD, JUSTIFY=RIGHT,$
ENDSTYLE
END
```

出力結果は次のとおりです。

DED_CODE	DEPARTMENT	BANK_ACCT	DED_AMT
CITY	MIS	40950036	$14.00
		122850108	$31.75
		163800144	$82.70
	Total City Deduction for MIS		**$128.45**
	PRODUCTION	160633	$7.42
		136500120	$18.25
		819000702	$60.20
	Total City Deduction for PRODUCTION		**$85.87**
		TOTAL	**$214.32**

見出しおよび脚注の位置揃え

左、右、中央揃え以外の要素で見出しおよび脚注内のテキストとデータを位置揃えするには、次の説明を参照し、最適な位置揃え方法を決定します。

位置揃え方法	適用先	用途	関連方法
1) スタイルシート属性 HEADALIGN COLSPAN JUSTIFY **詳細** HTML、XLSX、EXL2K、PDF、PPTX、DHTML レポートの見出しまたは脚注の位置揃えを参照してください。	HTML XLSX EXL2K PDF PPTX DHTML	**HTML とEXL2K レポートの見出しまたは脚注項目の位置揃え** HTML または EXL2K フォーマットでレポートを表示する場合、HEADALIGN オプションを使用して、見出しおよび脚注項目を位置を、レポート本文の HTML テーブル内のフィールド、または埋め込み HTML テーブル内のセルに揃えます。ブラウザは、指定に基づき位置揃えを行います。WIDTH および JUSTIFY とは異なり、ここでは単位を指定する必要はありません。 **PDF レポートの見出しまたは脚注の**	

位置揃え方法	適用先	用途	関連方法
		位置揃え　PDF フォーマットでレポートを表示する場合、HEADALIGN=BODY オプションを使用して、見出しおよび脚注項目の位置をレポート本文の列に揃えます。 **複数項目にわたる見出しまたは脚注項目の指定**　この結果を取得するには、HEADALIGN 構文と COLSPAN 属性を組み合わせることができます。詳細は、HTML または PDF レポートの見出しおよび脚注要素の複数列範囲での位置揃えを参照してください。	
2) スタイル属性 WIDTH JUSTIFY	HTML PDF	**HTML と PDF 間の互換**　コードを変更せずに、HTML (内部カスケードスタイルシート使用) および PDF を同時出力するには、スタイルシート内で WIDTH および JUSTIFY 属性を使用します。これらの設定は、レポート、ページ、ソート項目を指定している見出し、および脚注に適用できます。 **見出しまたは脚注項目の位置揃え**　両方を使用することで、WIDTH および JUSTIFY は、見出しまたは脚注の全体、または見出しまたは脚注の全体の行ではなく、明示的に指定されない位置揃え幅がレポートパネルの合計幅である見出し内の特定した項目を揃えるようにします。見出しまたは脚注内の項目を右揃えまたは中央揃えにするには、位置を揃える領域の幅を知る必要があります。その情報は、WIDTH 属性により提供されます。	HTML、PDF、EXL2K レポートでは、HEADALIGN オプションを使用して特定の項目の位置を揃えることができます。

位置揃え方法	適用先	用途	関連方法
		複数行見出しまたは脚注の小数点の位置揃え　小数桁数がさまざまなデータ内の小数点を位置揃えするために、この方法を使用します。小数項目の幅を定義し、次に、フィールドの何桁目に小数点を配置するかを決定します。これにより、右側に表示される小数点以下の桁数に関係なく、小数点はフィールド内の同じ位置に配置されます。	
3) スタイルシート属性 POSITION	PDF HTML (制限付き)	**見出し、脚注の開始位置、またはそれらの中の項目**　絶対および相対開始位置を指定するには、POSITION構文を使用します。 HTML では、内部カスケードスタイルシートで POSITION を使用することで、見出しまたは脚注の開始位置を特定することができます。見出しまたは脚注内にイメージを配置することもできます。 **フィールドの見出しまたは脚注項目の位置揃え**　見出し項目とフィールドの位置を揃えるには、POSITION構文を使用します。以下はその例です。 `TYPE=SUBHEAD, LINE=1, ITEM=3,POSITION=SALES, $` フィールドが SALES である横方向の位置に、ソート項目を指定している見出しの ITEM 3 を配置します。	PDF レポートでは、WIDTHおよびJUSTIFYを使用することで、ほとんどの位置揃えを実行することができます。 HTML または PDF レポートでは、HEADALIGN 属性を BODY に設定することで、見出し項目とフィールドの位置を揃えることができます。

HTML、XLSX、EXL2K、PDF、PPTX、DHTML レポートの見出しまたは脚注の位置揃え

HTML 出力および HTML 位置揃えを使用する Excel 2000 出力の場合、HEADALIGN オプションを使用して見出しおよび脚注内のテキストとフィールド項目の位置を揃えることができます。このオプションを使用することでフォーマットの柔軟な設定が可能になりますが、HTML およびブラウザ動作の制限に依存します。次に、HEADALIGN がどのように動作するかについて説明します。

PDF 出力では、HEADALIGN=BODY オプションを使用して、レポート本文に合わせて見出しおよび脚注を位置揃えすることができます。

HTML および Excel 2000 出力では、HEADALIGN を BODY または INTERNAL のいずれかに設定した場合、出力結果が HTML テーブルとして配置されます。この場合、列幅はブラウザによって決定されるため、各項目の正確な配置が制限されます。各項目 (テキストまたは埋め込みフィールド) は、次の HTML テーブルセル (<TD>) 内に順次配置されます。HEADALIGN がデフォルト設定の NONE に設定されるときは、見出しまたは脚注内のすべての項目は、単一セル内に連結されます。ブラウザは、テキスト長に適合するよう、見出しテーブルおよびレポートテーブルを拡張します。

BODY または INTERNAL のいずれかを選択することにより、デフォルトを無効にして、項目の配置を設定できます。

- HTML または EXL2K 出力では、HEADALIGN=BODY に設定すると、レポート本文と同一の HTML テーブルに見出し項目のセルが配置されます。これにより、見出し内の項目とレポート本文のデータの列幅が一致するため、必然的にこれらの位置が揃います。PDF 出力では、HEADALIGN=BODY に設定すると、見出し項目の位置がデータ列の位置に揃います。これは、見出し項目とデータの列幅を揃えるための簡単な方法です。たとえば、ソート項目を指定している脚注に、計算された中間合計値を表示する場合があります。HEADALIGN=BODY を使用することにより、合計されたデータと中間合計の値を同一の列内に揃えることができます。
- HEADALIGN=INTERNAL に設定すると、見出し項目自体を HTML テーブルに配置することができます。これにより、見出し項目とレポートの列幅 (またはその逆) が互いに影響されるため、データとは関係なく、見出し項目は相互に縦に揃えられます。
- EXL2K フォーマットのレポートで、見出しまたは脚注内の複数の項目が、ブランクを含めずにスポットマーカーのみで区切られている場合、そのスポットマーカーにより、セルのテキスト内の項目間に余分なブランクが追加されます。この問題を回避するには、XLSX フォーマットのレポートを使用します。

サンプル出力を比較するには、HEADALIGN オプションによる出力結果の比較を参照してください。

テキストを複数の部分に分割するには、リクエストで <+0> スポットマーカーを使用します。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

HEADALIGN オプションは、COLSPAN 属性と併用することができます。COLSPAN は、見出し項目を複数のテーブル列に拡張するため、見出しの設計が柔軟にできるようになります。詳細は、HTML または PDF レポートの見出しおよび脚注要素の複数列範囲での位置揃えを参照してください。

レポートに複数タイプの見出しまたは脚注がある場合、この構文を使用して、それぞれの中にある要素を個別に位置揃えできます。

ヒント：その他の位置揃え方法の概要については、見出しおよび脚注の位置揃えを参照してください。

HTML または PDF レポートの見出しおよび脚注の位置揃え

```
TYPE = {REPORT|headfoot}, HEADALIGN = option, $
```

説明

REPORT

選択した位置揃えを、レポート内のすべての見出しおよび脚注に適用します。

headfoot

見出しまたは脚注のタイプです。有効な値は、TABHEADING、TABFOOTING、HEADING、FOOTING、SUBHEAD、SUBFOOT です。

option

位置揃えのタイプです。有効な値には、次のものがあります。

NONE - HTML レポートの見出し項目を主 (本文) テーブル内の埋め込み HTML テーブルに配置し、すべての見出し項目 (テキストおよびフィールド) を埋め込みテーブルの単一セル内で連結します。その結果、見出し項目がすべて 1 行に表示されます。PDF レポートでこの

値を指定すると、見出しのデフォルト位置揃えが使用されます。デフォルト値は NONE です。

INTERNAL - 見出し項目をそれ自体の HTML テーブルに配置します。各項目は個別のセルに配置されます。これにより、データフィールドとは関係なく、見出し項目を相互に縦方向に揃えることができます。見出し項目の幅は、レポートフィールドの幅 (またはその逆) に影響しません。

注意：PDF レポートでは、HEADALIGN=INTERNAL はサポートされません。

BODY - 見出し項目をデータ列に合わせて位置揃えします。HTML 出力でこの値を指定すると、見出し項目は、レポート本文と同一の HTML テーブルのセル内に配置されます。これらの列幅は同じであるため、見出し項目と、レポート本文のデータは自動的に位置揃えされます。PDF 出力でこの値を指定すると、見出しおよび脚注の項目はデータ列に合わせて位置が揃えられます。

注意：HEADALIGN=BODY は、パネル化をサポートしません。

見出し項目が複数 HTML テーブルフィールドに拡張するよう、HEADALIGN オプションを COLSPAN 属性と組み合わせることができます。詳細は、HTML または PDF レポートの見出しおよび脚注の位置揃えを参照してください。

PDF レポートでの中間脚注項目に表示されるレポート列との位置揃え

次のリクエストは、GGORDER データソースに対して実行され、中間見出しにテキストオブジェクト (Total) およびフィールドオブジェクト (ST.QUANTITY) が指定されています。TYPE=SUBFOOT, HEADALIGN=BODY ,$ を使用して、中間見出しの項目の位置を、レポート列と揃えます。スポットマーカー「<+0>」を使用して、テキストオブジェクトをレポートの 2 列目内に、スポットマーカー「<+0>」を使用して、フィールドオブジェクトをレポートの 3 列目内に、それぞれ配置します。さらに、テキスト項目を、レポート列で左揃え (デフォルト設定) にします。フィールドオブジェクトは、レポート列で右揃えにします。

```
TABLE FILE GGORDER
PRINT QUANTITY
ORDER_NUMBER ORDER_DATE STORE_CODE
BY PRODUCT_CODE BY PRODUCT_DESCRIPTION
WHERE ORDER_DATE EQ '01/01/96'
WHERE STORE_CODE EQ 'R1019'
ON PRODUCT_CODE SUBFOOT
" <+0 Total: <+0 <ST.QUANTITY"
ON TABLE SET PAGE-NUM OFF
```

```
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLESHEET *
TYPE = SUBFOOT,HEADALIGN=BODY, $
TYPE = SUBFOOT,OBJECT=TEXT,STYLE = BOLD, $
TYPE = SUBFOOT,OBJECT=FIELD,JUSTIFY=RIGHT,STYLE = BOLD, $
ENDSTYLE
END
```

出力では、テキスト「Total」の位置は製品名と揃えられ、中間見出しフィールドオブジェクトは「Ordered Units」列で右揃えになります。

Product Code	Product	Ordered Units	Order Number	Order Date	Store Code
B141	Hazelnut	300	1	01/01/96	R1019
		117	2	01/01/96	R1019
	Total:	**417**			
B142	French Roast	126	3	01/01/96	R1019
		86	4	01/01/96	R1019
		49	5	01/01/96	R1019
	Total:	**261**			
F101	Scone	37	11	01/01/96	R1019
		325	12	01/01/96	R1019
	Total:	**362**			
F102	Biscotti	399	13	01/01/96	R1019
	Total:	**399**			
F103	Croissant	93	14	01/01/96	R1019
		433	15	01/01/96	R1019
	Total:	**526**			
G100	Mug	289	6	01/01/96	R1019
		231	7	01/01/96	R1019
	Total:	**520**			
G104	Thermos	347	8	01/01/96	R1019
	Total:	**347**			
G110	Coffee Grinder	265	9	01/01/96	R1019
	Total:	**265**			
G121	Coffee Pot	309	10	01/01/96	R1019
	Total:	**309**			

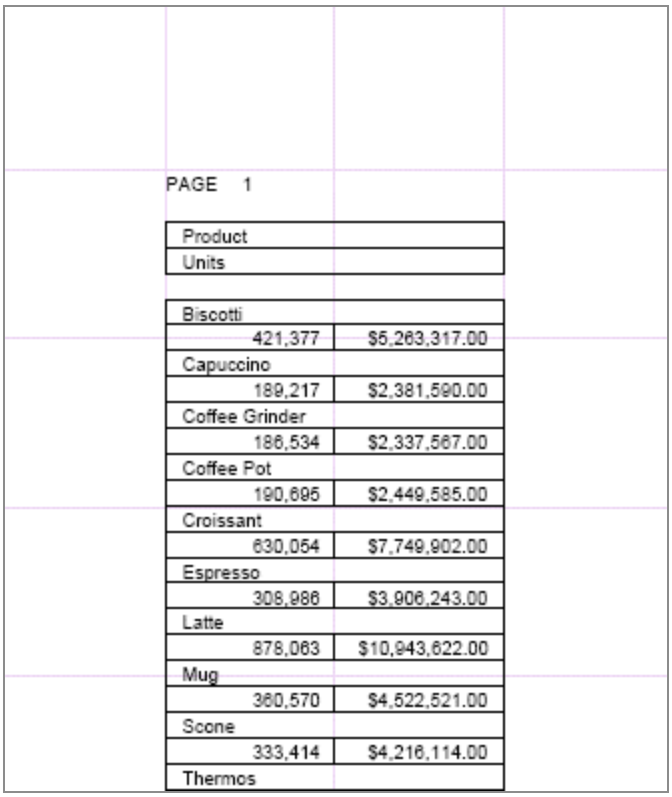
PDF レポートでの HEADALIGN=BODY と OVER の使用

HEADALIGN=BODY を使用して見出し要素の位置をデータ行と揃える場合、フィールドの最初の行は、アンカーデータ行になります。各見出し行には、アンカーデータ行で指定された数の行が含められます。その他のデータ行に表示されるそれ以外の列は、表示されません。最初のデータ行に含まれるデータ値のセル数が、その他のデータ行よりも少ない場合、最初のデータ行に含まれていない列の位置揃え列を、見出しに追加することはできません。

次の例では、最初の行 (アンカーデータ行) に単一の値が含まれています。見出しに配置された項目で、後続行に表示される列 2 に対応するものは、表示されません。

```
SET LAYOUTGRID=ON
TABLE FILE GGSALES
"Product<+0>"
"Units<+0>Dollars"
SUM
PRODUCT AS ''
OVER
UNITS/D8C AS '' DOLLARS/D12.2CM AS ''
BY PRODUCT NOPRINT
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, SQUEEZE=ON,FONT=ARIAL, SIZE=8, LEFTGAP=.1, RIGHTGAP=.1,
GAPINTERNAL=ON, LEFTMARGIN=1,$
TYPE=REPORT, BORDER=ON, $
TYPE=HEADING, BORDERALL=ON, HEADALIGN=BODY, $
TYPE=HEADING, LINE=1, ITEM=1, COLSPAN=2, WIDTH=2, JUSTIFY=LEFT, $
TYPE=HEADING, LINE=2, ITEM=1, WIDTH=1, JUSTIFY=LEFT, $
TYPE=HEADING, LINE=2, ITEM=2, WIDTH=1, JUSTIFY=LEFT, $
TYPE=REPORT, COLUMN=PRODUCT(2),   SQUEEZE=2,  $
TYPE=REPORT, COLUMN=UNITS, SQUEEZE=1, $
TYPE=REPORT, COLUMN=DOLLARS, SQUEEZE=1, $
END
```

出力では、見出し行はそれぞれ 1 列になり、データ行は交互に 1 列と 2 列になります。



PAGE 1

Product	
Units	
Biscotti	
421,377	$5,263,317.00
Capuccino	
189,217	$2,381,590.00
Coffee Grinder	
186,534	$2,337,567.00
Coffee Pot	
190,695	$2,449,585.00
Croissant	
630,054	$7,749,902.00
Espresso	
308,986	$3,906,243.00
Latte	
878,063	$10,943,622.00
Mug	
360,570	$4,522,521.00
Scone	
333,414	$4,216,114.00
Thermos	

HEADALIGN オプションによる出力結果の比較

次のリクエストは、各 HEADALIGN 設定を使用した位置揃えの相違を示しています。レポート本文の HTML テーブルと、見出しのバリエーションの埋め込み HTML テーブルとを区別するため、出力内には罫線が表示されています。

すべての HEADALIGN 設定は COLSPAN 構文と互換性があるため、見出しを複数フィールドに拡張することができます。

```
TABLE FILE CAR
SUM SALES BY COUNTRY BY CAR BY MODEL
ON COUNTRY SUBHEAD
"This is my subhead"
" "
"Country is:<COUNTRY Car is:<CAR"
"Model is:<MODEL"
IF COUNTRY EQ 'ENGLAND'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLESHEET *
TYPE=SUBHEAD, HEADALIGN=OPTION, $
TYPE=SUBHEAD, LINE=1, ITEM=1, COLSPAN=4, JUSTIFY=CENTER, $
```

```
ENDSTYLE
END
```

HEADALIGN=NONE　上記リクエスト構文の太字で示された 2 つ目の TYPE=SUBHEAD 宣言を使用しない場合、デフォルトの左揃えで別のテーブルが作成されます。各見出しの行の中のテキストおよびフィールドは、単一 HTML テーブルセル内で連結しています。

```
TYPE=SUBHEAD, HEADALIGN=NONE, $
```

COUNTRY	CAR	MODEL	SALES
This is my subhead Country is:ENGLAND Car is:JAGUAR Model is:V12XKE AUTO			
ENGLAND	JAGUAR	V12XKE AUTO	0
		XJ12L AUTO	12000
	JENSEN	INTERCEPTOR III	0
	TRIUMPH	TR7	0

HEADALIGN=NONE と COLSPAN

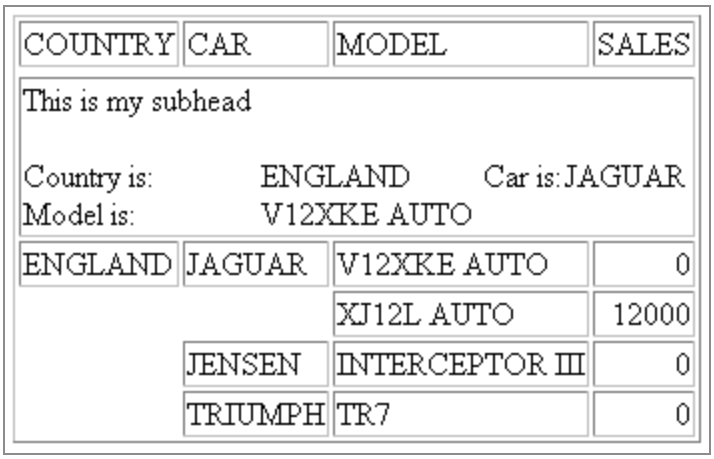
```
TYPE=SUBHEAD, HEADALIGN=NONE, $
TYPE=SUBHEAD, LINE=1, ITEM=1, COLSPAN=4, JUSTIFY=CENTER, $
```

COUNTRY	CAR	MODEL	SALES
This is my subhead Country is:ENGLAND Car is:JAGUAR Model is:V12XKE AUTO			
ENGLAND	JAGUAR	V12XKE AUTO	0
		XJ12L AUTO	12000
	JENSEN	INTERCEPTOR III	0
	TRIUMPH	TR7	0

見出しの 1 行目は、COLSPAN=4 設定に基づいて、内部テーブルの 4 列すべてを範囲として中央揃えで配置されています。

HEADALIGN=INTERNAL　別の HTML テーブルを作成します。見出し項目数に基づいて、列が生成されます。それぞれの項目は、別のセルに配置されます。これらの列は、レポート本文の HTML テーブル列とは対応しません。

```
TYPE=SUBHEAD, HEADALIGN=INTERNAL, $
```

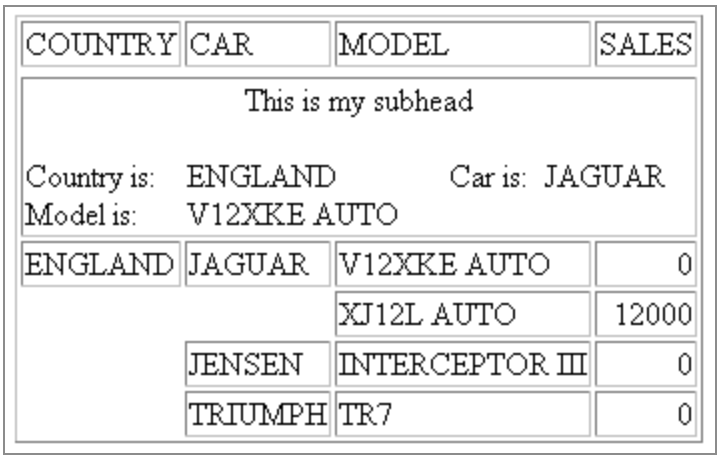


COUNTRY	CAR	MODEL	SALES
This is my subhead Country is: ENGLAND Car is: JAGUAR Model is: V12XKE AUTO			
ENGLAND	JAGUAR	V12XKE AUTO	0
		XJ12L AUTO	12000
	JENSEN	INTERCEPTOR III	0
	TRIUMPH	TR7	0

COUNTRY は、内部テーブルの最初の列内で MODEL と位置揃えされます。<COUNTRY の値は、第 2 列内で <MODEL の値と位置揃えされます。

HEADALIGN=INTERNAL と COLSPAN

```
TYPE=SUBHEAD, HEADALIGN=INTERNAL, $
TYPE=SUBHEAD, LINE=1, ITEM=1, COLSPAN=4, JUSTIFY=CENTER, $
```



COUNTRY	CAR	MODEL	SALES
This is my subhead Country is: ENGLAND Car is: JAGUAR Model is: V12XKE AUTO			
ENGLAND	JAGUAR	V12XKE AUTO	0
		XJ12L AUTO	12000
	JENSEN	INTERCEPTOR III	0
	TRIUMPH	TR7	0

最初の行は、COLSPAN=4 の設定を基に内部テーブルの全 4 列を中央に揃えます。

HEADALIGN=BODY　主 HTML テーブルのセルの中に見出し行を配置します。その結果、見出しの列が主テーブルの列に一致します。

```
TYPE=SUBHEAD, HEADALIGN=BODY, $
```

COUNTRY	CAR	MODEL	SALES
This is my subhead			
Country is:	ENGLAND	Car is:	JAGUAR
Model is:	V12XKE AUTO		
ENGLAND	JAGUAR	V12XKE AUTO	0
		XJ12L AUTO	12000
	JENSEN	INTERCEPTOR III	0
	TRIUMPH	TR7	0

COUNTRY は、主 (本文) HTML テーブルの最初の列内で MODEL と位置揃えされます。
<COUNTRY の値は、第 2 列内で <MODEL の値と位置揃えされます。

HEADALIGN=BODY と COLSPAN

```
TYPE=SUBHEAD, HEADALIGN=BODY, $
TYPE=SUBHEAD, LINE=1, ITEM=1, COLSPAN=4, JUSTIFY=CENTER, $
```

COLSPAN は、見出しの先頭行を複数の列範囲で位置揃えします。

COUNTRY	CAR	MODEL	SALES
This is my subhead			
Country is:	ENGLAND	Car is:	JAGUAR
Model is:	V12XKE AUTO		
ENGLAND	JAGUAR	V12XKE AUTO	0
		XJ12L AUTO	12000
	JENSEN	INTERCEPTOR III	0
	TRIUMPH	TR7	0

ソート項目が指定された脚注要素とデータ列の位置揃え

次のリクエストは、2 つのデータフィールド (Product および Ordered Unites) のそれぞれを含んだソート項目を指定している脚注 (TOTAL IS およびその値) の 2 つの要素を位置揃えするために、HEADALIGN = BODY を使用して HTML レポートを作成します。JUSTIFY = RIGHT は、ソート項目を指定している脚注の全体に適用され、データ列に表示される脚注の各要素を右揃えにします。

```
TABLE FILE GGORDER
PRINT QUANTITY
BY PRODUCT_CODE NOPRINT BY PRODUCT_DESCRIPTION
WHERE ORDER_DATE EQ '01/01/96'
WHERE STORE_CODE EQ 'R1019'
ON PRODUCT_CODE SUBFOOT
"TOTAL IS: <ST.QUANTITY"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = SUBFOOT, HEADALIGN = BODY, JUSTIFY = RIGHT, $
TYPE = SUBFOOT, OBJECT = FIELD, STYLE = BOLD, $
ENDSTYLE
END
```

以下は、出力結果の一部を示しています。

Product	Ordered Units
Hazelnut	300
	117
TOTAL IS:	**417**
French Roast	126
	86
	49
TOTAL IS:	**261**
Scone	37
	325
TOTAL IS:	**362**

別の HTML テーブルを使用したページ見出し要素の位置揃え

次のリクエストは、レポートの本文中で位置揃えを管理する HTML テーブル内で、ページ見出しの埋め込み HTML テーブルを作成します。このテーブルは、すべての見出し要素を表示するため、3 行、3 列あります。

見出しの最初の行内で、スポットマーカー (<+0>) は、2 つのテキスト要素を作成します。最初の要素はブランクで、2 つ目の要素は、「Gotham Grinds, Inc」です。出力結果内で、埋め込みテーブルの最初の行の 2 つ目のセル内に表示されます。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

見出しの 2 行目と 4 行目はブランクです。

見出しの 3 行目内にあるスポットマーカーは、この行を 3 つのテキスト要素「Orders Report」、ブランク、「Run on: &DATE」に分割します。出力結果で、各要素は、リクエストで指定した順序で、埋め込み HTML テーブルの 3 行目のセル内に表示されます。

```
TABLE FILE GGORDER
HEADING
" <+0>Gotham Grinds, Inc."
" "
"Orders Report <+0> <+0> Run on: &DATE"
" "
PRINT ORDER_NUMBER ORDER_DATE STORE_CODE QUANTITY
```

```
BY PRODUCT_CODE BY PRODUCT_DESCRIPTION
IF RECORDLIMIT EQ 10
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = ON, $
TYPE = HEADING, HEADALIGN = INTERNAL, STYLE = BOLD, $
ENDSTYLE
END
```

リクエストの GRID=ON は、見出しの埋め込み HTML テーブルおよびレポート本文の主 HTML テーブルを表示できるようにします。

出力結果は次のとおりです。

Gotham Grinds, Inc.

Orders Report **Run on: 06/19/02**

Product Code	Product	Order Number	Order Date	Store Code	Ordered Units
B141	Hazelnut	1	01/01/96	R1019	300
		2	01/01/96	R1019	117
B142	French Roast	3	01/01/96	R1019	126
		4	01/01/96	R1019	86
		5	01/01/96	R1019	49
G100	Mug	6	01/01/96	R1019	289
		7	01/01/96	R1019	231
G104	Thermos	8	01/01/96	R1019	347
G110	Coffee Grinder	9	01/01/96	R1019	265
G121	Coffee Pot	10	01/01/96	R1019	309

罫線が非表示の (オフ) 場合でも、配置は保持されます。

Gotham Grinds, Inc.

Orders Report **Run on: 04/20/01**

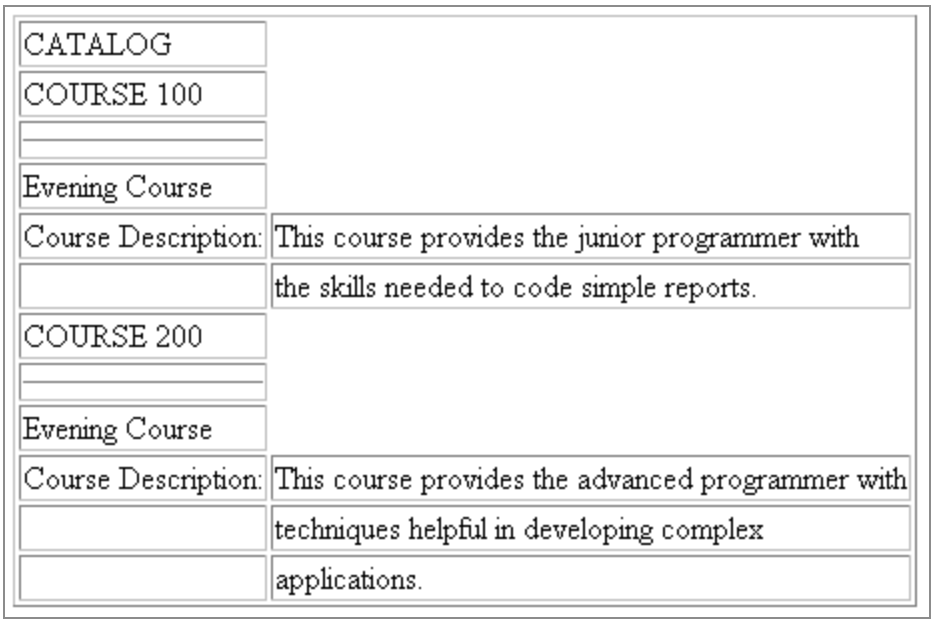
Product Code	Product	Order Number	Order Date	Store Code	Ordered Units
B141	Hazelnut	1	01/01/96	R1019	300
		2	01/01/96	R1019	117
B142	French Roast	3	01/01/96	R1019	126
		4	01/01/96	R1019	86
		5	01/01/96	R1019	49
G100	Mug	6	01/01/96	R1019	289
		7	01/01/96	R1019	231
G104	Thermos	8	01/01/96	R1019	347
G110	Coffee Grinder	9	01/01/96	R1019	265
G121	Coffee Pot	10	01/01/96	R1019	309

ソート項目を指定している脚注内のフィールドを使用したテキストフィールドの位置揃え

次のリクエストは、ソート項目を指定している脚注内のテキストフィールド行を位置揃えするために、HEADALIGN=BODY を使用します。この設定で、脚注内の各要素は、レポートに生成される主 HTML テーブル内の列と位置揃えされます。最初の要素 (Course Description テキスト) は、最初のデータ列 (CATALOG) と位置揃えされます。埋め込みフィールドは 2 つ目の列と位置揃えされます。この例では、位置揃えを見やすくするために、罫線をオンにします。

```
TABLE FILE TXTFLD
BY CATALOG SUBFOOT
"Course Description: <TEXTFLD"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = ON, $
TYPE = SUBFOOT, HEADALIGN = BODY, $
ENDSTYLE
END
```

出力結果では、CATALOG の値が変わるたびにテキストフィールドの新しい値が表示されています。



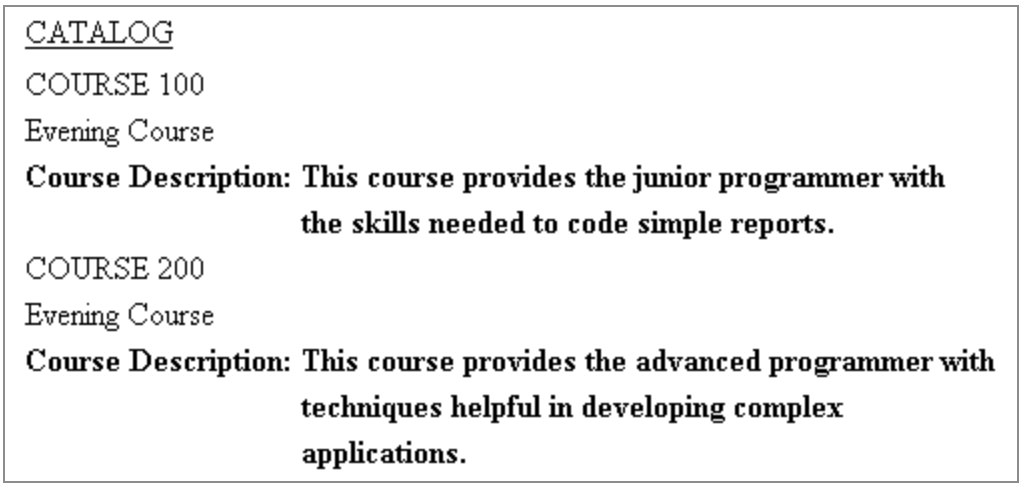
CATALOG	
COURSE 100	
Evening Course	
Course Description:	This course provides the junior programmer with
	the skills needed to code simple reports.
COURSE 200	
Evening Course	
Course Description:	This course provides the advanced programmer with
	techniques helpful in developing complex
	applications.

ソート項目を指定している脚注内のテキストフィールド位置揃えおよびスタイル

次のリクエストは、「Course Description:」 と <TEXTFLD の値を含む脚注の 2 行目に太字を適用します。

```
TABLE FILE TXTFLD
BY DESCRIPTION AS 'CATALOG' SUBFOOT
"Evening Course"
"Course Description: <TEXTFLD"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = SUBFOOT, HEADALIGN = BODY, $
TYPE = SUBFOOT, LINE = 2, STYLE = BOLD, $
ENDSTYLE
END
```

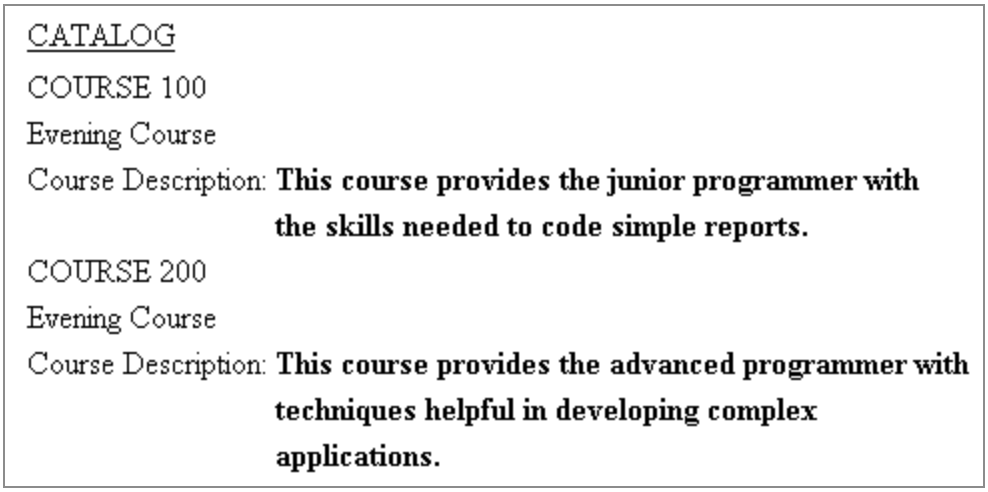
出力結果は次のとおりです。



次のように、スタイルシートで、スタイル設定のオブジェクトとしてテキストフィールドを指定する場合、<TEXTFLD の値のみが太字になります。

```
TYPE = SUBFOOT, HEADALIGN = BODY, $
TYPE = SUBFOOT, LINE = 2, OBJECT = FIELD, STYLE = BOLD, $
```

出力結果では、<TEXTFLD の値のみが太字になっています。



HTML または PDF レポートの見出しおよび脚注要素の複数列範囲での位置揃え

HTML または PDF レポートで HEADALIGN=BODY を使用すると、見出しまたは脚注の各要素はデータ列に合わせて位置が揃えられます。HEADALIGN=INTERNAL を使用すると、各要素

は、レポートの見出しまたは脚注用として作成、配置された HTML テーブルの単一列内に表示されます。デフォルト設定では、見出しまたは脚注の各要素 (ITEM) は、最初の列に配置されます。ただし、COLSPAN 属性を使用して、項目を複数列にわたり、配置することができます。HEADALIGN オプションについての詳細は、HTML、XLSX、EXL2K、PDF、PPTX、DHTML レポートの見出しまたは脚注の位置揃えを参照してください。

HEADALIGN 属性が見出しおよび脚注の全体に適用されるのに対して、COLSPAN 属性は、見出しおよび脚注内の特定の項目に適用されます。そのため、HEADALIGN 属性および COLSPAN 属性は、2 つのスタイルシートで個別に指定する必要があります。

HTML または PDF レポートの見出しおよび脚注要素の複数列範囲での位置揃え

```
TYPE = headfoot, [subtype,] COLSPAN = n, $
```

説明

headfoot

見出しまたは脚注のタイプです。有効な値は、TABHEADING、TABFOOTING、HEADING、FOOTING、SUBHEAD、SUBFOOT です。

subtype

レポートコンポーネントを識別する追加の属性です。これらのオプションは、要素識別の必要度に応じて単独または組み合わせて使用します。有効な値には、次のものがあります。

- **LINE** 見出しまたは脚注内の行の位置 (n 行目) に基づいて行を指定します。個別の行を指定すると、それぞれの行に異なるフォーマットを設定することができます。

  見出しまたは脚注が複数行あり、スタイルシートで LINE を指定しない場合、設定はすべての行に適用されます。ブランク行は、LINE の値を解析するときに計算されます。

- **OBJECT** 見出しまたは脚注内の要素を、テキストまたは項目値として指定します。有効な値は TEXT または FIELD です。TEXT として、テキストまたはダイアログマネージャ (&) 変数のテキストを指定することができます。

  同一の見出しまたは脚注内で、テキストおよび埋め込みフィールドの両方を同時に

スタイル設定しない場合は、OBJECT=TEXT を指定する必要はありません。

- **ITEM**　行内の項目の位置に基づいて項目を指定します。<+0> スポットマーカーを使用して、見出しまたは脚注の行を複数の項目に分割することができます。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

OBJECT の ITEM を指定する際は、次のガイドラインに従います。

- OBJECT=TEXT を使用する場合は、テキストのみが左から右に数えられます。
- OBJECT=FIELD を使用する場合は、値のみが左から右に数えられます。
- OBJECT を使用しない場合は、テキストおよび項目の値が左から右に数えられます。

スタイルシートで ITEM を指定する場合、最初の行の開始位置のみではなく、見出しまたは脚注内の各行の開始位置から数えられます。

COLSPAN

この属性を使用して、項目の位置を複数列の幅に揃えます。

n

指定した項目の位置を揃える列幅です。

3 列にわたるページ見出しの中央揃え

次のリクエストで、HEADALIGN=INTERNAL は 3 列の見出し埋め込み HTML テーブルを作成します。さらに、COLSPAN 属性は、HTML テーブルの 3 列にわたる見出し 1 行目「Gotham Grinds, Inc.」を、レポート内で中央に揃えます。

```
TABLE FILE GGORDER
HEADING
"Gotham Grinds, Inc."
" "
"Orders Report <+0> <+0> Run on: &DATE"
" "
PRINT ORDER_NUMBER ORDER_DATE STORE_CODE QUANTITY
BY PRODUCT_CODE BY PRODUCT_DESCRIPTION
IF RECORDLIMIT EQ 10
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
```

```
TYPE = HEADING, HEADALIGN = INTERNAL, $
TYPE = HEADING, LINE=1, COLSPAN=3, STYLE = BOLD, JUSTIFY=CENTER, $
TYPE = HEADING, LINE=3, ITEM=3, JUSTIFY=RIGHT, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Gotham Grinds, Inc.

Orders Report **Run on: 06/19/02**

Product Code	Product	Order Number	Order Date	Store Code	Ordered Units
B141	Hazelnut	1	01/01/96	R1019	300
		2	01/01/96	R1019	117
B142	French Roast	3	01/01/96	R1019	126
		4	01/01/96	R1019	86
		5	01/01/96	R1019	49
G100	Mug	6	01/01/96	R1019	289
		7	01/01/96	R1019	231
G104	Thermos	8	01/01/96	R1019	347
G110	Coffee Grinder	9	01/01/96	R1019	265
G121	Coffee Pot	10	01/01/96	R1019	309

複数フィールドにわたるフィールド値の位置揃え

次のリクエストでは、HEADALIGN=BODY はレポートの本文と同じ HTML テーブル内の、ソート項目を指定している脚注を位置揃えします。COLSPAN = 5 は、HTML テーブルの 5 つ目の列内のソート項目を指定している脚注 (「Total」というテキスト) 内の最初の項目を配置します。ソート項目を指定している脚注 (フィールド <ST.QUANTITY) 内の 2 つ目の項目は、次の列に配置されます。

COLSPAN が単一項目合計に適用されるのに対して、HEADALIGN 属性はソート項目を指定している脚注の全体 (結果として項目の両方) に適用されるため、COLSPAN 属性とは別の行になります。

```
TABLE FILE GGORDER
PRINT ORDER_NUMBER ORDER_DATE STORE_CODE QUANTITY
BY PRODUCT_CODE BY PRODUCT_DESCRIPTION
WHERE ORDER_DATE EQ '01/01/96'
WHERE STORE_CODE EQ 'R1019'
ON PRODUCT_CODE SUBFOOT
"Total: <ST.QUANTITY"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = SUBFOOT, HEADALIGN = BODY, JUSTIFY = RIGHT, STYLE = BOLD, $
TYPE = SUBFOOT, OBJECT = TEXT, COLSPAN = 5, $
ENDSTYLE
END
```

以下は、出力結果の一部を示しています。

Product Code	Product	Order Number	Order Date	Store Code	Ordered Units
B141	Hazelnut	1	01/01/96	R1019	300
		2	01/01/96	R1019	117
				Total:	**417**
B142	French Roast	3	01/01/96	R1019	126
		4	01/01/96	R1019	86
		5	01/01/96	R1019	49
				Total:	**261**
F101	Scone	11	01/01/96	R1019	37
		12	01/01/96	R1019	325
				Total:	**362**

PDF レポートでのフィールド値の複数列範囲の位置揃え

次のリクエストでは、HEADALIGN=BODY で、ソート項目を指定している脚注の位置を、レポートの列と揃えます。COLSPAN = 5 は、ソート項目を指定している脚注 (「Total」というテキスト) 内の最初の項目を、レポートの 5 列目に配置します。ソート項目を指定している

脚注 (フィールド <ST.QUANTITY) 内の 2 つ目の項目は、次の列に配置されます。中間見出しの項目は、右揃えになります。

COLSPAN はテキスト項目「Total」のみに適用されますが、HEADALIGN 属性は、ソート項目を指定している脚注全体 (結果として両方の項目) に適用されるため、COLSPAN 属性とは別の行になります。

```
TABLE FILE GGORDER
PRINT ORDER_NUMBER ORDER_DATE STORE_CODE QUANTITY
BY PRODUCT_CODE BY PRODUCT_DESCRIPTION
WHERE ORDER_DATE EQ '01/01/96'
WHERE STORE_CODE EQ 'R1019'
ON PRODUCT_CODE SUBFOOT
"Total:<ST.QUANTITY"
""
ON TABLE SET PAGE-NUM OFF
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLESHEET *
TYPE=REPORT, FONT=ARIAL, SQUEEZE=ON,$
TYPE = SUBFOOT, HEADALIGN = BODY, JUSTIFY = RIGHT, STYLE = BOLD, $
TYPE = SUBFOOT, ITEM=1, COLSPAN = 5, $
ENDSTYLE
END
```

出力で、ソート項目を指定している見出しの最初の項目 (「Total」というテキスト) は、レポートで 5 つ目の列に表示されます。ソート項目を指定している脚注 (フィールド <ST.QUANTITY) 内の 2 つ目の項目は、次の列に配置されます。

Product Code	Product	Order Number	Order Date	Store Code	Ordered Units
B141	Hazelnut	1	01/01/96	R1019	300
		2	01/01/96	R1019	117
				Total:	**417**
B142	French Roast	3	01/01/96	R1019	126
		4	01/01/96	R1019	86
		5	01/01/96	R1019	49
				Total:	**261**
F101	Scone	11	01/01/96	R1019	37
		12	01/01/96	R1019	325
				Total:	**362**
F102	Biscotti	13	01/01/96	R1019	399
				Total:	**399**
F103	Croissant	14	01/01/96	R1019	93
		15	01/01/96	R1019	433
				Total:	**526**
G100	Mug	6	01/01/96	R1019	289
		7	01/01/96	R1019	231
				Total:	**520**
G104	Thermos	8	01/01/96	R1019	347
				Total:	**347**
G110	Coffee Grinder	9	01/01/96	R1019	265
				Total:	**265**
G121	Coffee Pot	10	01/01/96	R1019	309
				Total:	**309**

複数行見出しまたは脚注内容の位置揃え

HTML、XLSX、EXL2K、PDF、PPTX、DHTML レポートの見出しまたは脚注の位置揃え内で説明する HEADALIGN および COLSPAN 構文は、HTML レポート用です。ここでは、印刷可能な HTML および PDF フォーマットのレポートを設計する方法について説明します。スタイルシート内で WIDTH および JUSTIFY 構文を使用することで、次のことができます。

- テキストまたはデータの列単位左右位置揃え。
- 列と行のフォーマットの組み合わせ。
- 小数点以下の桁数が異なる場合の小数点の位置揃え。

見出しと脚注、ページ見出しとページ脚注、ソート項目を指定している見出しと脚注には、等幅フォントまたはプロポーショナルフォントのいずれかを使用して、WIDTH 属性および JUSTIFY 属性を適用することができます。

これらの方法は、以前のバージョンの HTML レポートでは利用できなかった WebFOCUS スタイルシート属性をサポートする内部カスケードスタイルシートを基にしています。これらの方法に関連する構文は、HTML レポート (HEADALIGN および COLSPAN を使用) と、PDF レポート (POSITION と スポットマーカーを使用) で異なるフォーマットが設定される問題を解決します。

WIDTH および JUSTIFY 属性は、複数行で構成される見出し、脚注のフォーマット設定や、縦方向の小数点の位置揃えが必要なときに、特に役立ちます。この構文は、個別の見出し、脚注行内の項目を配置するために使用することもできます。

ヒント：その他の位置揃え方法の概要については、見出しおよび脚注の位置揃えを参照してください。

列内の見出しテキストおよびデータの位置揃え

複数行のレポート見出し、ページ見出し、脚注では、次の構文を使用します。

```
TYPE=headfoot, WRAP=OFF, $
TYPE=headfoot, [LINE=line_#,] ITEM=item_#, [OBJECT={TEXT|FIELD}],
 WIDTH=width,   [JUSTIFY=option,] $
```

複数行のソート項目を指定している見出し、または脚注では、次の構文を使用します。

```
TYPE=headfoot, WRAP=OFF, $
TYPE={SUBHEAD|SUBFOOT}, [BY=sortfield] [LINE=line_#,] ITEM=item_#,
 [OBJECT={TEXT|FIELD}], WIDTH=width, [JUSTIFY=option,] $
```

説明

headfoot

見出しまたは脚注のタイプです。有効な値は、TABHEADING、TABFOOTING、HEADING、FOOTING、SUBHEAD、SUBFOOT です。

sortfield

TYPE=SUBHEAD または SUBFOOT を指定するときは、特定のソート項目を指定している見出しまたは脚注の位置揃えを指定することができます。ソート項目が指定されない場合、フォーマットは、すべてのソート項目に関連する見出しまたは脚注に適用されます。

LINE

見出しまたは脚注内の行を識別するオプションの項目です。行を個別に指定すると、それぞれの行に異なるフォーマットを設定することができます。

見出しまたは脚注が複数行あり、スタイルシートで LINE を指定しない場合、設定はすべての行に適用されます。ブランク行は、LINE の値を解析するときに計算されます。

LINE は ITEM と併用することができます。

ITEM

位置揃えを制御するために WIDTH を使用するときに必要な項目です。項目は次のいずれかです。

- 列単位で位置揃えする、縦方向のテキストまたはデータ。
- 行内の項目の配置。それぞれの行要素を項目として識別する必要があります。

単一の見出しまたは脚注には、この方法のいずれか、または両方を使用することができます。

<+0> スポットマーカーを使用して、見出しまたは脚注の行を複数の項目に分割することができます。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。識別できる項目数は、レポートページの物理的な幅の制限内で、見出しまたは脚注内の項目の累積的な幅により制限されます。

定義する幅の要素識別を調整するために OBJECT とともに ITEM を使用することができます。OBJECT の ITEM を指定する際は、次のガイドラインに従います。

- OBJECT=TEXT を使用する場合は、テキストのみが左から右に数えられます。
- OBJECT=FIELD を使用する場合は、値のみが左から右に数えられます。
- OBJECT を使用しない場合は、テキストおよび項目の値が左から右に数えられます。

スタイルシートで ITEM を指定する場合、最初の行の開始位置のみではなく、見出しまたは脚注内の各行の開始位置から数えられます。

OBJECT

見出しまたは脚注内の要素をテキストまたはフィールド値として識別するオプションの属性です。有効な値は TEXT または FIELD です。TEXT として、テキストまたはダイアログマネージャ (&) 変数のテキストを指定することができます。

同一の見出しまたは脚注内で、テキストおよび埋め込みフィールドの両方を同時にスタイル設定しない場合は、OBJECT=TEXT を指定する必要はありません。

width

番号付き項目に関連付けられた最長のテキストまたはフィールド値を格納するために必要な長さです (デフォルト単位はインチ)。詳細は、複数行見出しまたは脚注内容の位置揃えを参照してください。

option

位置揃えのタイプです。有効な値には、次のものがあります。

LEFT - 見出しまたは脚注を左揃えにします。デフォルト値は LEFT です。

RIGHT - 見出しまたは脚注を右揃えにします。

CENTER - 見出しまたは脚注を中央揃えにします。

DECIMAL (n)

フィールドの右側から数えた、小数点を配置する位置を指定する長さ (デフォルト単位はインチ) です。この指定により、小数点以下の桁数に関係なくフィールド内の同じ位置に小数点の位置を決めることができます。

この長さは、項目に対して指定された幅の一部になります。詳細は、列幅を決定するにはを参照してください。

注意：WRAP とともに JUSTIFY を使用することはできません。

複数行の見出しまたは脚注内の行と項目のフォーマット

行のフォーマットは、それぞれの行で識別する項目を制御できます。

- 同一幅を持つ同じ数の項目を位置揃えして、縦方向に並べることができます。以下はその例です。

行 1	項目 1	項目 2	項目 3
行 2	項目 1	項目 2	項目 3

- 項目数が異なる場合でも、各行の項目の開始位置と累積幅が一致している限り、これらの項目の位置を揃えることができます。

行 1			
行 2			

HEADALIGN および COLSPAN 構文を使用することはできません。これらの構文は HTML レポート用で、WIDTH および JUSTIFY の設定と競合する場合があります。

HTML レポートでは、WIDTH および JUSTIFY の適切な処理をするために WRAP OFF (デフォルトは ON) を設定します。

複数行の見出しまたは脚注内のデータおよびテキストの位置揃え

次の自由形式レポートで、内容はソート項目を指定している見出し内ですべて定義され、テキストとデータを積み重ねることで、Country 間が比較しやすくなっています。各データのセットは、縦方向に整列されています。この位置揃えを適用するために、最初のテキスト列を「項目 1」、次のデータ列を「項目 2」のように、縦方向の各セットを「項目」として識別します。

特に、最後の列に注目してください。桁数が異なる小数データが小数点で整列されているため、読みやすく、比較しやすくなっています。

Country:	ARGENTINA	Exchange Rate:	35.00
Type:	ST. NOTES	Projected Return:	4.200
Holder:	COMM	Balance:	42,167,880.00
Country:	BRAZIL	Exchange Rate:	39.55
Type:	ST. NOTES	Projected Return:	4.200
Holder:	COMM	Balance:	45,493,501.00
Country:	CANADA	Exchange Rate:	49.00
Type:	ST. NOTES	Projected Return:	3.990
Holder:	COMM	Balance:	56,212,634.00
Country:	CZECH. REP	Exchange Rate:	1,192.45
Type:	ST. NOTES	Projected Return:	12.110
Holder:	COMM	Balance:	51,121,201.00

下表は、前述のレポート構造の抜粋です。

項目 1 **テキスト**	**項目 2** **データ値**	**項目 3** **テキスト**	**項目 4** **小数値**
Country	ARGENTINA、BRAZIL と続きます。	Exchange Rate	*nn.dd*
Type	ST.NOTES	Projected Return	*n.ddd*
Holder	COMM	Balance	*nn,nnn,nnn.dd*

次のコードに示すように、各項目の列幅とその内容の配置を指定します。

```
DEFINE FILE SHORT
BALANCE/D14.2=BALANCE;
END
TABLE FILE SHORT
BY COUNTRY NOPRINT SUBHEAD
"Country:<COUNTRY Exchange Rate:<EXCHANGE_RATE"
"Type:<TYPE Projected Return:<PROJECTED_RETURN"
"Holder:<HOLDER Balance:<BALANCE"
ON TABLE SET PAGE-NUM OFF
```

```
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, FONT='TIMES', $
TYPE=REPORT, GRID=OFF, $
TYPE=SUBHEAD, ITEM=1, WIDTH=1.00, JUSTIFY=RIGHT, $
TYPE=SUBHEAD, ITEM=2, WIDTH=1.25, JUSTIFY=RIGHT, $
TYPE=SUBHEAD, ITEM=3, WIDTH=1.25, JUSTIFY=RIGHT,$
TYPE=SUBHEAD, ITEM=4, WIDTH=1.5,  JUSTIFY=DECIMAL(.6),$
ENDSTYLE
END
```

このプロシジャにより、ソート項目を指定している見出しが 3 行で生成され、それぞれが 4 つの項目に分割されて、指定された幅および定義された位置揃えで配置されます。項目 4 の小数値には、小数点の位置で整列するための標準位置揃えが使用されています。詳細は、複数行見出しまたは脚注内容の位置揃えおよび 複数行見出し、脚注の小数点位置揃えを参照してください。

注意：HTML レポートでこの機能を利用するには、内部カスケードスタイルシートをオン (SET HTMLCSS=ON) にする必要があります。このコマンドにより、以前のバージョンの HTML レポートでは利用できなかった WebFOCUS スタイルシート属性が使用可能になります。このコード行は、PDF レポートでは無視されます。

複数行見出し、脚注の小数点位置揃え

幅と配置の値を基に複数行の見出し内の位置を揃えることは、異なる小数点位置を持つデータを含むレポートの中で特に役立ちます。たとえば、数値がドルの場合、小数点以下が 2 桁になるようにフォーマットが設定されます。数値がスイスフランの場合、小数点以下が 4 桁になるようにフォーマットが設定されます。数値が円の場合には、小数点は付きますが、小数点以下は 0 (ゼロ) 桁です。通貨または単位は同じで、小数精度の桁数が変化する場合もあります。

次の出力結果が示すように、縦方向の並びの中で小数点を位置揃えすることにより、これらの数字をより読みやすく、比較しやすくできます。

浮動小数点		位置揃え済み小数点	
Bond -----	Face Value ---	Bond -----	Face Value

浮動小数点		位置揃え済み小数点	
-------	---------	-------	----------
Galosh Ltd.	22375.5784596	Galosh Ltd.	22375.5784596
Mukluk Inc.	1212345.457	Mukluk Inc.	1212345.457
Overshoe Inc.	232.45484	Overshoe Inc.	232.45484

この方法は、小数点の位置を適切に揃えるために、小数を含んだ項目の幅の指定と左、右、中央揃えの組み合わせを使用します。この結果を生成する構文については、列内の見出しテキストおよびデータの位置揃えを参照してください。

列幅を決定するには

見出しまたは脚注項目の幅を決定するには、次の 3 つの手順を実行します。

手順

1. テキストまたは項目内の文字の最大長を確認します。
2. 英数文字テキストの場合は、文字数を計算します。フィールドの場合は、マスターファイルまたは DEFINE などのコマンドのフォーマット指定を参照します。
3. 使用するフォントのサイズを基に、手順 1 で確認したバイト数を表示するために必要な物理スペースのサイズを、使用する測定単位 (例、インチ) で計算します。たとえば、COUNTRY フィールドの次の値は、以下のようになります。

フォント	フォントサイズ	比較	インチ
Helvetica	10	England	.5
Times New Roman	10	England	.44
Courier	10	England	.56

ヒント：測定結果を再利用する場合は、レポート内で同一フォントセットを使用することをお勧めします。

小数点の位置を決定するには

項目の幅を決定した後に、関連する計測を行い、小数点の右側にさまざまな桁数の小数データを表示するために必要な物理スペースを決定することができます。

手順

1. 小数点の右側の桁数、および小数点自体を格納するために必要な最大桁数を決定します。
2. 使用するフォントのサイズを基に、手順 1 で確認したバイト数を表示するために必要な物理スペースのサイズを、使用する測定単位 (例、インチ) で計算します。

見出しおよび脚注の行列フォーマット組み合わせ

行と列のフォーマットを組み合わせることにより、複雑なレポートを作成することができます。同一の見出しや脚注に表示する行の範囲を変更し、さらにそれらに表示する列の数や表示位置、位置揃えを変更することができます。

列と行のフォーマットの組み合わせを使用した項目の位置揃え

次のリクエストは、7 行のソート項目を指定している見出しで内容が定義されるレポートを作成します。テキストおよびデータは、次の 2 つのグループに分類されます。

- 最初のグループは、Country および Region (大陸) を識別します。
- 2 番目のグループは、各 Country と Region の財務情報を提供します。

これは単一のソート項目を指定している見出しですが、ここでの目的は、レポートをグループごとに分けて相違を強調し、比較しやすくすることです。このリクエストは、レポートの設計者がフォーマットの変更を簡単にできるよう、コーディング方法も示しています。詳細は、コードに続く注釈を参照してください。

サンプルリクエストに示すように、見出しには、テキストと埋め込みフィールドの 2 種類の項目を含めることができます。埋め込みフィールド間やスポットマーカー間のテキスト項目には、1 つのブランクを含む任意のテキストを使用することができます。特に、1 つのテキストを 2 つの項目として扱う場合、<+0> スポットマーカーを使用して 2 つの項目に分けることができます。以下は、見出し行の例です。

```
" <+0>Country:<COUNTRY"
```

最初の項目は、1 バイトのブランクです。

2 番目の項目は、<+0> スポットマーカーにより分割されたテキスト (Country) です。

3 番目の項目は、埋め込みフィールド <COUNTRY です。

<+0> スポットマーカーについての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

リクエストと注釈

```
    DEFINE FILE SHORT
    BALANCE/D14.2=BALANCE;
    END
    TABLE FILE SHORT
    BY COUNTRY NOPRINT SUBHEAD
1.  " <+0>Country:<COUNTRY"
2.  " <+0>Region:<REGION"
    " "
```

```
3."Type:<TYPE <+0>Exchange Rate:<EXCHANGE_RATE"
4."Holder:<HOLDER <+0>Projected Return:<PROJECTED_RETURN"
5."Risk class:<RISK_CLASS <+0>Balance:<BALANCE"
    " "
    ON TABLE SET PAGE-NUM OFF
6.ON TABLE SET HTMLCSS ON
    ON TABLE SET STYLESHEET *
    TYPE=REPORT, FONT='TIMES', $
    TYPE=REPORT, GRID=OFF, $
    -* Bottom section of subhead:
7.TYPE=SUBHEAD, ITEM=1, WIDTH=1.00, JUSTIFY=RIGHT, $
8.TYPE=SUBHEAD, ITEM=2, WIDTH=1.25, JUSTIFY=RIGHT, $
9.TYPE=SUBHEAD, ITEM=3, WIDTH=.5, $
10.TYPE=SUBHEAD, ITEM=4, WIDTH=1.25, JUSTIFY=RIGHT,$
11.TYPE=SUBHEAD, ITEM=5, WIDTH=1.5, JUSTIFY=DECIMAL(.6),$
    -* Top section of subhead (overrides above ITEM defaults
    -*   for lines 1 and 2):
12. -SET &INDENT=1.5;
13.TYPE=SUBHEAD, LINE=1, ITEM=1, WIDTH=&INDENT, $
14.TYPE=SUBHEAD, LINE=1, ITEM=2, WIDTH=1, JUSTIFY=LEFT, $
15.TYPE=SUBHEAD, LINE=1, ITEM=3, SIZE=14, WIDTH=2, JUSTIFY=LEFT, $
16.TYPE=SUBHEAD, LINE=2, ITEM=1, WIDTH=&INDENT, $
17.TYPE=SUBHEAD, LINE=2, ITEM=2, WIDTH=1, JUSTIFY=LEFT, $
18.TYPE=SUBHEAD, LINE=2, ITEM=3, WIDTH=2, JUSTIFY=LEFT, $
    ENDSTYLE
    END
```

出力では、テキストおよびデータの位置を揃えることにより重要な情報およびその関係が強調され、比較がしやすいよう、小数データは小数点で位置揃えされます。

	Country:	ARGENTINA	
	Region:	SOUTH AMERICA	
Type:	ST. NOTES	Exchange Rate:	35.00
Holder:	COMM	Projected Return:	4.200
Risk class:	Medium	Balance:	42,167,880.00
	Country:	BRAZIL	
	Region:	SOUTH AMERICA	
Type:	ST. NOTES	Exchange Rate:	39.55
Holder:	COMM	Projected Return:	4.200
Risk class:	Medium	Balance:	45,493,501.00
	Country:	CZECH. REP	
	Region:	EASTERN EUROPE	
Type:	ST. NOTES	Exchange Rate:	1,192.45
Holder:	COMM	Projected Return:	12.110
Risk class:	High	Balance:	51,121,201.00
	Country:	ENGLAND	
	Region:	WESTERN EUROPE	
Type:	ST. NOTES	Exchange Rate:	21.35
Holder:	COMM	Projected Return:	4.060
Risk class:	Medium	Balance:	125,757,198.00

行 #	説明
1-2	ソート項目を指定している見出しの最初の 2 行の部分の内容を定義します。各行には、3 つの項目が含まれています。最初の項目はブランクで、<+0> スポットマーカーにより次の項目から分割されています。2 つ目の項目にはテキスト、3 つ目の項目にはテキストに関連するデータ値がそれぞれ含まれています。
3-5	ソート項目を指定している見出しの末尾 3 行の部分の内容を定義します。それぞれの行には、テキスト、そのテキストに関連するデータ値、ブランク列 (NULL スポットマーカーにより次の項目から分割)、テキスト、そのテキストに関連するデータ値の 5 つの項

行 #	説明
	目が含まれています。
6	内部カスケードスタイルシートをオンにします。これらのフォーマットの必要条件です。このコマンドにより、以前のバージョンの HTML レポートでは利用できなかった WebFOCUS スタイルシート属性が使用可能になります。このコード行は、PDF レポートでは無視されます。
7-11	内容を 5 つの項目に分けることにより、ソート項目を指定している見出しの基本的なフォーマット特性を指定します。それぞれの項目には、定義された幅および配置情報 (ブランク列を除く) が指定されます。 **重要：**リクエストに含まれていない追加フォーマットコード (12-17 として表示) があります。7-11 として表示した指定は、ソート項目を指定している見出しの全体に適用されます。見出しの上部 3 つの列のフォーマットは、次で説明する最初の 3 つの列の指定を基にしています。ただし、これは目的とする効果ではないため、スタイルシートコードの 2 番目を定義することで、ソート項目を指定している見出しの 1 行目と 2 行目のフォーマットを無効にします。12-18の注釈を参照してください。 見出しの末尾 3 行のフォーマットは、次の指定により制御されます。 項目 1 は、テキスト (Type、Holder、Risk Class) を含む列単位で識別します。幅が 1 インチに定義され、テキストは右揃えです。 項目 2 は、項目 1 のテキストに関連するデータ値を含む列単位で識別します。幅が 1.25 インチに定義され、データは右揃えです。 項目 3 は、ブランクです。幅が 0.5 インチです。配置は関係ありません。 項目 4 は、テキスト (例として、Exchange Rate、Projected Return、Balance) を含む列単位で識別します。幅が 1.25 インチで定義され、テキストは右揃えです。 項目 5 は、小数値を含む列単位で識別します。値を持つ列の幅は 1.5 インチで、小数点は、列の右側から 0.6 インチに固定されています。

行 #	説明
	共通な幅および配置の定義は、各項目の適切な位置揃えを行います。
12	幅を 1.5 インチに設定する「&INDENT」という変数を定義します。この変数は、ソート項目を指定している見出しの 1 行目と 2 行目の先頭のブランク領域 (第 1 項目) の幅を定義します。 変数として幅を定義するのは、1 箇所で値を変更することにより簡単に異なる幅をテストすることができるためです。複雑なレポートにおいて、この方法は開発時間を大幅に節約できる場合があります。詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』の「ダイアログマネージャ」の項目を参照してください。
13-18	見出しの最初の 2 行の行ごとのフォーマットを指定します。このコードは、行の番号を指定することからソート項目を指定している見出しの 1 行目と 2 行目の以前のフォーマットを上書きします。 各行の第 1 項目は、ブランク領域を参照します。幅は、変数として定義され、&INDENT の現在の値を基に実装されます。 各行の第 2 項目は、テキスト領域を参照します。1 インチの幅で定義され、テキストは左揃えです。 各行の第 3 項目は、データ値を参照します。2 インチの幅で定義され、データは左揃えです。 共通な幅および配置の定義は、各項目の適切な位置揃えを行います。 15 行目の第 1 項目は、COUNTRY フィールドに関連するデータ値のフォントサイズを定義します。両方の行にあるその他すべての項目は、デフォルトフォントを使用します。行ごとに異なるフォーマットを設定することで、各項目に固有の特性を定義することができます。

見出し、脚注、その内部の項目の配置

PDF または HTML レポートでは、スタイルシートの POSITION 属性を使用して、見出しまたは脚注の開始位置を指定することができます。この場合、所定の測定単位で距離を指定します。HTML では、この機能を使用するには内部カスケードスタイルが必要です。位置揃え方法の選択についての詳細は、見出しおよび脚注の位置揃えを参照してください。

さらに、PDF レポートでは、POSITION 属性を使用して、見出しまたは脚注内の要素の絶対または相対開始位置を指定したり、レポートフィールド内の見出しまたは脚注内の項目の位置を揃えることができます。絶対開始位置は、レポートの左マージンからの距離です。相対開始位置は、前のオブジェクトからの距離です。見出し行の最初の項目では、これはレポートの左マージンです。

HTML レポートでは、関連構文およびスタイルシートを使用して、見出しまたは脚注にイメージを配置することができます。イメージについての詳細は、レポートページのレイアウトを参照してください。

見出しまたは脚注開始位置の設定

見出しまたは脚注全体の開始位置をレポートの左マージンとの相対関係で指定するには、次の構文を使用します。

```
TYPE = headfoot, POSITION = position, $
```

説明

headfoot

見出しまたは脚注のタイプです。有効な値は、TABHEADING、TABFOOTING、HEADING、FOOTING、SUBHEAD、SUBFOOT です。

position

UNITS 属性 (デフォルト値は INCHES) により指定される、左からの距離です。

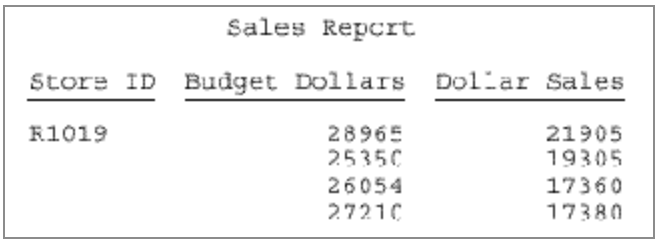
注意：HTML レポートでは、この構文は、内部カスケードスタイルシートとともに使用する必要があります。

PDF レポート見出し開始位置の設定

次のリクエストは、レポート見出しを左マージンから 1.25 インチの位置に配置します。

```
SET ONLINE-FMT=PDF
TABLE FILE GGSALES
PRINT BUDDOLLARS DOLLARS
BY STCD
WHERE BUDDOLLARS GE 25000
WHERE STCD EQ 'R1019'
ON TABLE SUBHEAD
"Sales Report"
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE SET SQUEEZE ON
ON TABLE SET STYLESHEET *
TYPE = TABHEADING, POSITION = 1.25, $
ENDSTYLE
END
```

出力結果は次のとおりです。



Sales Report

Store ID	Budget Dollars	Dollar Sales
R1019	28965	21905
	25350	19305
	26054	17360
	27210	17380

HTML レポート見出し開始位置の設定

次のリクエストは、HTML コードの一部として内部カスケードシートを生成し、POSITION 属性を使用して見出し (Sales Report) 開始位置を左マージンから 1.5 インチに指定します。

```
SET ONLINE-FMT = HTML
TABLE FILE GGSALES
PRINT BUDDOLLARS DOLLARS
BY STCD
WHERE BUDDOLLARS GE 25000
WHERE STCD EQ 'R1019'
ON TABLE SUBHEAD
"Sales Report"
```

```
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
TYPE = TABHEADING, POSITION = 1.5, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Century Corporation Orders Report

Order Number:	Product Number#:	Quantity:	Line Total
Century Corporation orders for store Audio Expert (Store # 1003CA) in CA.			
74610	1012	268	$222,234.35
	1028	323	$29,215.62
	1032	339	$27,208.12
	1034	339	$154,943.38
	1036	339	$93,266.39
Century Corporation orders for store Audio Expert (Store # 1003CO) in CO.			
39274	1006	179	$54,201.50
	1008	179	$29,034.16
	1020	179	$45,457.59
	1032	400	$30,861.24
	1034	400	$143,744.49

見出しまたは脚注要素開始位置の設定

PDF レポートで、見出しまたは脚注項目の開始位置を前の項目との関係で指定するには、次の構文を使用します。

```
TYPE = headfoot, [subtype,] POSITION = {+|-}option, $
```

説明

headfoot

見出しまたは脚注のタイプです。有効な値は、TABHEADING、TABFOOTING、HEADING、FOOTING、SUBHEAD、SUBFOOT です。

subtype

レポートコンポーネントを識別する追加の属性です。これらのオプションは、要素の識別の詳細度により、個別または組み合わせて使用されます。有効な値には、次のものがあります。

- **LINE** 見出しまたは脚注内の行の位置 (n 行目) に基づいて行を指定します。個別の行を指定すると、それぞれの行に異なるフォーマットを設定することができます。

  見出しまたは脚注が複数行あり、スタイルシートで LINE を指定しない場合、設定はすべての行に適用されます。ブランク行は、LINE の値を解析するときに計算されます。

- **ITEM** 行内の項目の位置に基づいて項目を指定します。<+0> スポットマーカーを使用して、見出しまたは脚注の行を複数の項目に分割することができます。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

  OBJECT の ITEM 番号を決定するには、次のガイドラインに従います。

  - OBJECT=TEXT を使用する場合は、テキストのみが左から右に数えられます。
  - OBJECT=FIELD を使用する場合は、値のみが左から右に数えられます。
  - OBJECT を使用しない場合は、テキストおよび項目の値が左から右に数えられます。

  スタイルシートで ITEM を指定する場合、最初の行の開始位置のみではなく、見出しまたは脚注内の各行の開始位置から数えられます。

- **OBJECT** 見出しまたは脚注内の要素を、テキストまたは項目値として指定します。有効な値は TEXT または FIELD です。TEXT として、テキストまたはダイアログマネージャ (&) 変数のテキストを指定することができます。

  同一の見出しまたは脚注内で、テキストおよび埋め込みフィールドの両方を同時にスタイル設定しない場合は、OBJECT=TEXT を指定する必要はありません。

option

位置揃えの方法です。有効な値には、次のものがあります。

- ***position*** UNITS 属性で設定された測定単位 (デフォルトはインチ) に基づいて指定する絶対位置の距離です。

- **+** 見出しまたは脚注の要素の開始位置を、1 つ前の項目から右側の距離で指定します。見出しまたは脚注内の最初の項目では、前の項目はレポートの左マージンになります。
- **-** 見出しまたは脚注の要素の開始位置を、1 つ前の項目から左側の距離で指定します。これは、見出し内でイメージを重ねる場合に役立ちます。
- ***column_title*** 見出しまたは脚注の要素を、指定するフィールドの最初の文字で位置揃えします。

見出し項目の絶対開始位置設定

次のリクエストは、スポットマーカー <+0> を使用して、レポート見出しを 3 つのテキストに分割します。左のレポートマージンから 3 インチの位置で 3 番目のテキスト「1st Qtr 2001」を開始します。この方法は、PDF レポートで使用できます。

```
SET ONLINE-FMT = PDF
TABLE FILE GGSALES
SUM UNITS DOLLARS BY CATEGORY BY PRODUCT
ON TABLE SUBHEAD
"Sales Report - <+0>All Products<+0> 1st Qtr 2001"
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE SET SQUEEZE ON
ON TABLE SET STYLESHEET *
TYPE = TABHEADING, OBJECT = TEXT, ITEM=1, SIZE = 12, STYLE = BOLD, $
TYPE = TABHEADING, OBJECT = TEXT, ITEM=2, STYLE = BOLD, $
TYPE = TABHEADING, OBJECT = TEXT, ITEM=3, POSITION = 3, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Sales Report - All Products 1st Qtr 2001

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

見出し項目の相対開始位置設定

次のリクエストは、スポットマーカー <+0> を使用して、レポート見出しを 3 つのテキストに分割します。見出し行にある前の項目から右に 1 インチの位置で、3 番目のテキスト「1st Qtr 2001」を開始します。インチは、デフォルト単位です。この方法は、PDF レポートで使用できます。

```
SET ONLINE-FMT = PDF
TABLE FILE GGSALES
SUM UNITS DOLLARS BY CATEGORY BY PRODUCT
ON TABLE SUBHEAD
"Sales Report - <+0>All Products<+0> 1st Qtr 2001"
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE SET SQUEEZE ON
ON TABLE SET STYLESHEET *
TYPE = TABHEADING, OBJECT = TEXT, ITEM=1, SIZE = 12, STYLE = BOLD, $
TYPE = TABHEADING, OBJECT = TEXT, ITEM=2, STYLE = BOLD, $
TYPE = TABHEADING, OBJECT = TEXT, ITEM=3, POSITION = +1, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Sales Report - All Products 1st Qtr 2001

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

フィールド見出し項目の位置揃え

次のリクエストは、スポットマーカー <+0> を使用して、レポート見出しを 3 つのテキストに分割します。フィールド UNITS (売上数量) の横に、2 番目のテキストを開始します。この方法は、PDF レポートで使用できます。

```
SET ONLINE-FMT = PDF
TABLE FILE GGSALES
SUM UNITS DOLLARS BY CATEGORY BY PRODUCT
ON TABLE SUBHEAD
"Sales Report - <+0>All Products<+0> 1st Qtr 2001"
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE SET SQUEEZE ON
ON TABLE SET STYLESHEET *
TYPE = TABHEADING, LINE=1, ITEM=2, POSITION=UNITS, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Sales Report - All Products 1st Qtr 2001

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

ヒント：このリクエストでは、フィールド (UNITS) は名前で識別されます。ただし、フォーマットを設定するフィールドを識別するには、別の方法もあります。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

PRINTPLUS の使用

PRINTPLUS を使用することで、レポートを柔軟に作成することができます。たとえば、レポートで SUBFOOT の後に FOOTING を配置することができます。PRINTPLUS は、必要なレポートを正確に作成する柔軟性を提供します。

次のTABLE 機能を使用するには、PRINTPLUS パラメータを ON に設定する必要があります。

- PAGE-BREAK は、ページのブランクを適正に表示するために内部処理されます。たとえば、新しいレポートページが始まり、このページの先頭にブランク行がある場合、WebFOCUS はブランク行を表示せずに、レポートはページの最初から表示されます。
- NOSPLIT は、内部的に処理されます。指定した位置で改ページを実行するには、NOSPLIT を使用します。
- 指定済み条件が一致するところで、RECAP を実行できます。
- レポート SUBFOOT は、脚注の下部ではなく、上部に表示されます。
- IF/WHERE TOTAL または BY HIGHEST が使用されるときに、中間脚注内データが正しく表示されます。
- BY 項目の動作は BY 項目オプションに関係しているため、同じページに表示されます。脚注が 2 ページに分割されることはなくなります。
- 脚注および中間脚注は、常に最低 1 つのデータ項目とともにページに表示され、2 ページに分割されることはありません。
- ページの長さを超えて印刷されることはありません。
- OVER でリンクした項目が、別のページに分割されることはありません。
- 条件付き出力結果で予約済みの余白はありません。出力結果のページは、完全に使用されます。
- ソートフィールドの順序は意味がありません。

注意：PRINTPLUS は、スタイルシートではサポートされません。この場合、警告メッセージが生成されます。

PRINTPLUS の使用

次のコマンドを発行します。

```
SET PRINTPLUS = {ON|OFF}
```

SUBFOOT および FOOTING への PRINTPLUS の使用

PRINTPLUS が ON の場合、SUBFOOT が最初に表示され、続いて FOOTING が表示されます。

```
SET PRINTPLUS = ON
 TABLE FILE CAR
  PRINT CAR MODEL
  BY SEATS BY COUNTRY
  IF COUNTRY EQ ENGLAND OR FRANCE OR ITALY
  ON TABLE SUBFOOT
  " "
  " SUMMARY OF CARS IN COUNTRY BY SEATING CAPACITY"
  FOOTING
  " RELPMEK CAR SURVEY "
ON TABLE SET STYLE *
TYPE=REPORT,GRID=OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。

SEATS	COUNTRY	CAR	MODEL
-----	-------	_	-----

2	ENGLAND	TRIUMPH	TR7
	ITALY	ALFA ROMEO	2000 GT VELOCE
		ALFA ROMEO	2000 SPIDER VELOCE
		MASERATI	DORA 2 DOOR
4	ENGLAND	JAGUAR	V12XKE AUTO
		JENSEN	INTERCEPTOR III
	ITALY	ALFA ROMEO	2000 4 DOOR BERLINA
5	ENGLAND	JAGUAR	XJ12L AUTO
	FRANCE	PEUGEOT	504 4 DOOR

SUMMARY OF CARS IN COUNTRY BY SEATING CAPACITY

RELPMEK CAR SURVEY

スポットマーカーを使用した配置調整

数種類のスポットマーカーを使用して、プロポーショナルフォントを使用する HTML と PDF レポートの見出しおよび脚注、その中の要素の位置を調整することができます。スポットマーカーをその他の位置揃え方法と組み合わせることで、制御を最大限に活用することができます。詳細は、見出しおよび脚注の位置揃えを参照してください。

次のスポットマーカーは、項目を配置し、フォーマットを設定する項目を識別できるようにします。

- **<+0>** フォーマット設定を目的として、見出しまたは脚注を項目に分割します。

  見出しまたは脚注を後で個別にフォーマット設定できる項目に分割するには、指定するテキストまたはフィールドの後に、<+0> スポットマーカーを配置します。これにより、見出しまたは脚注にブランクが追加されることはありません。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

- **</n** ブランク行を指定します。

  見出しまたは脚注にブランク行を指定するには、リクエスト内のテキストと同じ行に </*n* スポットマーカーを配置します。これを単一行として配置した場合、WebFOCUS はスポットマーカーのある行に指定したブランク行を加えて計算します。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。

- **<-n** フィールドの直後に続く文字の位置を制御します。

注意：終了スポットマーカーの直後に開始スポットマーカーを追加すると (><)、この 2 つのスポットマーカー間にブランクが 1 つ挿入されます (> <)。フォーマット設定を適用する際は、このことを考慮する必要があります。

スポットマーカーを使用して、見出しおよび脚注の要素を固定または相対位置に配置することもできます。スポットマーカーには、定義済みの等幅フォントの文字幅を基準に、位置を制御するものがあります。これは、レガシーフォーマット設定方法であり、プロポーショナルフォントではサポートされません。

HTML レポートフィールド直後への文字配置

次のリクエストで生成される HTML レポートは、ソート項目を指定している見出しにおいて、STORE_CODE 項目の後ろに右括弧が表示され、STATE 項目の後ろにピリオド (.) が表示

されます。この動作は、前のオブジェクトからの相対開始位置を示す <-1 スポットマーカーで制御されます。前のオブジェクトの後ろに句読点が続くことを示すスポットマーカーが指定されていない場合、画面ではこれらの各位置に余分なスペースが表示されます。

```
SET ONLINE-FMT = HTML
SET PAGE-NUM = OFF
JOIN STORE_CODE IN CENTCOMP TO STORE_CODE IN CENTORD
```

```
TABLE FILE CENTCOMP
HEADING
"Century Corporation Orders Report </1"
PRINT PROD_NUM QUANTITY LINEPRICE
BY STORE_CODE NOPRINT
BY ORDER_NUM
ON STORE_CODE SUBHEAD
"Century Corporation orders for store <STORENAME <0X
(store # <STORE_CODE<-1 ) in <STATE|.</1"
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

以下は、出力結果の一部を示しています。

Century Corporation Orders Report

Order Number:	Product Number#:	Quantity:	Line Total
Century Corporation orders for store Audio Expert (Store # 1003CA) in CA.			
74610	1012	268	$222,234.35
	1028	323	$29,215.62
	1032	339	$27,208.12
	1034	339	$154,943.38
	1036	339	$93,266.39
Century Corporation orders for store Audio Expert (Store # 1003CO) in CO.			
39274	1006	179	$54,201.50
	1008	179	$29,034.16
	1020	179	$45,457.59
	1032	400	$30,861.24
	1034	400	$143,744.49

PDF レポートフィールド直後への文字配置

PDF レポートでは、リクエストの見出しまたは脚注の埋め込みフィールドの後ろに 1 つのブランクを挿入することで、処理において認識されるようにする必要があります。ただし、ブランクは出力結果内で必要がない場合があります。次の例では、PDF レポート内のフィールドの直後に句読点を配置をする 2 つの方法を示しています。

最初の方法は、スタイルシートで POSITION 属性を使用し、STORE_CODE 値の直後に、右括弧を配置します。2 つ目の方法は、<-1 スポットマーカーを使用し、STATE 値の直後にピリオド (.) を配置します。POSITION の測定単位には、UNITS で指定された POINTS が使用されます。テストの結果、デフォルトのプロポーショナルフォントとサイズを使用した場合、マイナス 7 ポイントの位置へ移動することにより、フィールドの後の適切な位置に右括弧が配置されることが分かりました。

```
SET ONLINE-FMT = PDF
SET PAGE-NUM = OFF
JOIN STORE_CODE IN CENTCOMP TO STORE_CODE IN CENTORD

TABLE FILE CENTCOMP
HEADING
"Century Corporation Orders Report"
PRINT PROD_NUM QUANTITY LINEPRICE
BY STORE_CODE NOPRINT
BY ORDER_NUM
ON STORE_CODE SUBHEAD
"Century Corporation orders for store <STORENAME (store # <STORE_CODE )
in <0X <STATE <-1 . </1"
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, UNITS=POINTS, $
TYPE=SUBHEAD, OBJECT=TEXT, ITEM=3, POSITION= -7, $
ENDSTYLE
END
```

出力結果は次のとおりです。

```
Century Corporation Orders Report
Order Number:  Product Number#:   Quantity:       Line Total

Century Corporation orders for store Audio Expert (store # 1003CA) in    CA.

74610          1012                    268      $222,234.35
               1028                    323       $29,215.62
               1032                    339       $27,208.12
               1034                    339      $154,943.38
               1036                    339       $93,266.39
Century Corporation orders for store Audio Expert (store # 1003CO) in    CO.

39274          1006                    179       $54,201.50
               1008                    179       $29,034.16
               1020                    179       $45,457.59
               1032                    400       $30,861.24
               1034                    400      $143,744.49
Century Corporation orders for store Audio Expert (store # 1003CT) in    CT.
```

スポットマーカーおよび位置指定がない場合、出力結果は次のようになります。

```
Century Corporation Orders Report
Order Number:  Product Number#:   Quantity:       Line Total

Century Corporation orders for store Audio Expert (store # 1003CA) in    CA .

74610          1012                    268      $222,234.35
               1028                    323       $29,215.62
               1032                    339       $27,208.12
               1034                    339      $154,943.38
               1036                    339       $93,266.39
Century Corporation orders for store Audio Expert (store # 1003CO) in    CO .

39274          1006                    179       $54,201.50
               1008                    179       $29,034.16
               1020                    179       $45,457.59
               1032                    400       $30,861.24
               1034                    400      $143,744.49
Century Corporation orders for store Audio Expert (store # 1003CT) in    CT .
```

フォント属性位置指定のカスタマイズ

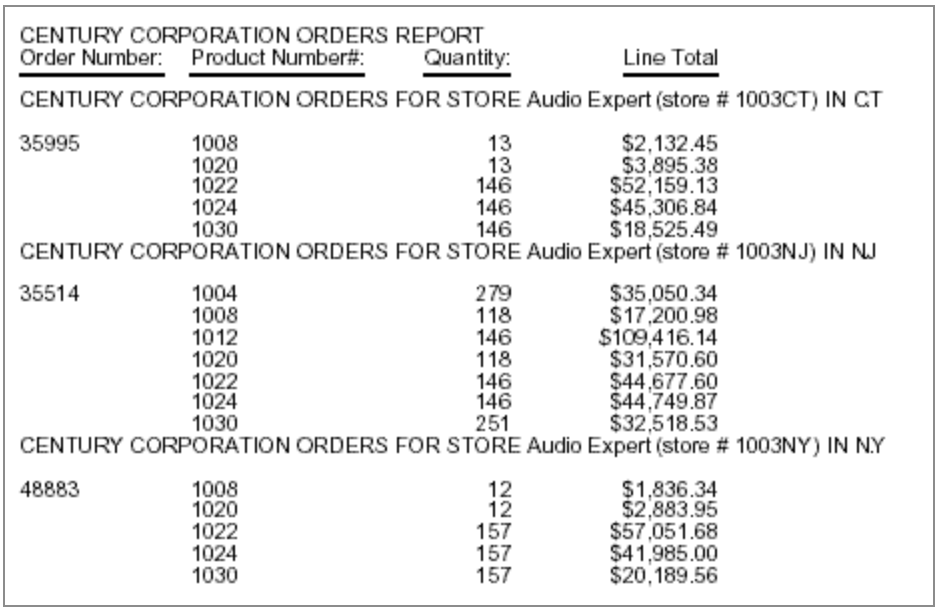
次のリクエストは、Helvetica フォントの 12 ポイントを使用しています。実験の結果、POSITION に -2 を指定すると、このフォントサイズのテキストが適切な位置へ移動することが分かりました。

```
SET ONLINE-FMT = PDF
SET PAGE-NUM = OFF
JOIN STORE_CODE IN CENTCOMP TO STORE_CODE IN CENTORD

TABLE FILE CENTCOMP
HEADING
"CENTURY CORPORATION ORDERS REPORT"
PRINT PROD_NUM QUANTITY LINEPRICE
BY STORE_CODE NOPRINT WHERE STORE_CODE EQ '1003NY' OR '1003CT' OR
    '1003NJ'
BY ORDER_NUM
ON STORE_CODE SUBHEAD
```

```
"CENTURY CORPORATION ORDERS FOR STORE <STORENAME (store # <STORE_CODE)<0X
 IN <STATE <-1 . </1"
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, UNITS=POINTS, FONT='Helvetica', SIZE=12, $
TYPE=SUBHEAD, OBJECT=TEXT, ITEM=3, POSITION= -2, $
ENDSTYLE
END
```

ヒント：HTML レポートで、POSITION 属性を使用して見出し要素を配置することはできません。ただし、STORE_CODE および STATE フィールド直後に横方向スポットマーカー <-1 を配置することにより、同じ結果を得ることができます。フィールドとその後に続く文字の間にブランクを追加することはできません。

出力結果は次のとおりです。



CENTURY CORPORATION ORDERS REPORT

Order Number:	Product Number#:	Quantity:	Line Total
CENTURY CORPORATION ORDERS FOR STORE Audio Expert (store # 1003CT) IN CT			
35995	1008	13	$2,132.45
	1020	13	$3,895.38
	1022	146	$52,159.13
	1024	146	$45,306.84
	1030	146	$18,525.49
CENTURY CORPORATION ORDERS FOR STORE Audio Expert (store # 1003NJ) IN NJ			
35514	1004	279	$35,050.34
	1008	118	$17,200.98
	1012	146	$109,416.14
	1020	118	$31,570.60
	1022	146	$44,677.60
	1024	146	$44,749.87
	1030	251	$32,518.53
CENTURY CORPORATION ORDERS FOR STORE Audio Expert (store # 1003NY) IN NY			
48883	1008	12	$1,836.34
	1020	12	$2,883.95
	1022	157	$57,051.68
	1024	157	$41,985.00
	1030	157	$20,189.56

見出しまたは脚注の縦方向配置制御

レポートの外観と読みやすさを向上させるために、縦方向の配置方法をいくつか組み合わせて使用することができます。

- HTML、PDF、またはその他のレポート形式で生成されたレポートでは、スポットマーカーおよびテキスト行を使用して、見出しまたは脚注テキストの上下に 1 行またはそれ以上を追加することができます。詳細は、見出しまたは脚注へのブランク行の追加を参照してください。

- PDF レポートでは、スタイルシートで TOPGAP および BOTTOMGAP 属性を使用することにより、見出し行または脚注行の行間 (上側または下側)、見出し行または脚注行のテキストと罫線との間隔 (上側または下側) を制御することができます。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。TOPGAP および BOTTOMGAP についての詳細は、レポートページのレイアウトを参照してください。
- PDF レポートでは、レポートデータの直後ではなく、ページの下部にページ脚注を配置することができます。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。

見出しまたは脚注へのブランク行の追加

見出しまたは脚注の上部、下部、またはその内部にブランク行を追加するには、次の構文オプションを使用します。

</n

追加するブランク行の数を指定するスポットマーカーです。スポットマーカーは、リクエストのテキストと同じ行に配置することをお勧めします。スポットマーカー </*n* のみを1行に配置した場合、指定したブランク行に加えてその行も追加されます。

" "

見出しまたは脚注内に挿入するブランク行を表します。

これらの方法は、単独または組み合わせて使用することができます。

レポート見出し上部または下部へのブランク行の追加

次のリクエストは、ページ見出し各行の間に 1 行のブランク、ページ見出しと実際のレポートの間に 2 行のブランクがある HTML レポートを作成します。最初のブランク行は、ブランクのテキスト行として、次のブランク行はブランク行スポットマーカーによって追加されます。

```
TABLE FILE GGSALES
SUM BUDUNITS UNITS BUDDOLLARS DOLLARS
BY CATEGORY
ON TABLE SUBHEAD
```

```
"SALES REPORT"
" "
"**(CONFIDENTIAL)**</1"
"December 2002 </2"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, TOPMARGIN=0, $
TYPE = TABHEADING, JUSTIFY = CENTER, $
ENDSTYLE
END
```

出力結果は次のとおりです。

SALES REPORT

(CONFIDENTIAL)

DECEMBER 2002

Category	Budget Units	Unit Sales	Budget Dollars	Dollar Sales
Coffee	1385923	1376266	17293907	17231465
Food	1377564	1384845	17267160	17229344
Gifts	931007	927880	11659732	11695502

見出しまたは脚注内の縦間隔の制御

PDF レポートでは、TOPGAP および BOTTOMGAP 属性を使用して、見出し行または脚注行の行間 (上側または下側)、見出しテキストまたは脚注テキストと罫線との間隔 (上側または下側) を制御することができます。

注意：TOPGAP および BOTTOMGAP は、複数行の見出しで 使用することができます。見出行の間に上部と下部のブランクが挿入されるため、行間隔は、見出しの上部および下部の間隔よりも大きくなります。

```
TYPE=headfoot, {TOPGAP|BOTTOMGAP}=gap, $
```

説明

headfoot

見出しまたは脚注のタイプです。有効な値は、TABHEADING、TABFOOTING、HEADING、FOOTING、SUBHEAD、SUBFOOT です。

TOPGAP

レポートコンポーネントの上側に追加するブランクの間隔を指定します。

BOTTOMGAP

レポートコンポーネントの下側に追加するブランクの間隔を指定します。

gap

UNITS パラメータで指定した単位でブランク数を指定します (デフォルト単位はインチ)。

罫線を使用しない場合、デフォルト値は 0 (ゼロ) です。

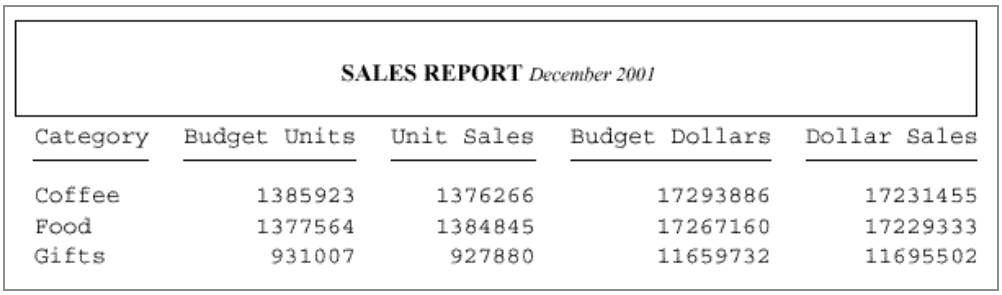
罫線を使用する場合、デフォルト値は、罫線とテキストの間の間隔を追加するために増加します。

PDF レポートで罫線行と見出しテキストを分離するブランク

次のリクエストは、上下の罫線からテキストを分離するために、レポートの見出しの上下にブランクが挿入された PDF レポートを生成します。TOPGAP 属性により上部、BOTTOMGAP 属性により下部にブランクが、それぞれ挿入されます。

```
TABLE FILE GGSALES
SUM BUDUNITS UNITS BUDDOLLARS DOLLARS
BY CATEGORY
ON TABLE SUBHEAD
"SALES REPORT <+0>December 2001"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT PDF
ON TABLE SET SQUEEZE ON
ON TABLE SET STYLESHEET *
TYPE = TABHEADING, GRID=ON, JUSTIFY=CENTER, TOPGAP=.25, BOTTOMGAP=.25, $
TYPE = TABHEADING,  FONT='TIMES', SIZE=12, STYLE=BOLD, $
TYPE = TABHEADING, ITEM=2, SIZE=10, STYLE=ITALIC, $
ENDSTYLE
END
```

出力結果は次のとおりです。



SALES REPORT *December 2001*

Category	Budget Units	Unit Sales	Budget Dollars	Dollar Sales
Coffee	1385923	1376266	17293886	17231455
Food	1377564	1384845	17267160	17229333
Gifts	931007	927880	11659732	11695502

ソート項目を指定している脚注下部縦間隔の調整

次のリクエストは、PDF レポートを生成します。ソート項目を指定している脚注を強調するために太字が設定され、各脚注テキストの下にブランクが追加されてデータと視覚的に結び付けられます。

```
TABLE FILE GGPRODS
PRINT PACKAGE_TYPE AND UNIT_PRICE
WHERE UNIT_PRICE GT 50
BY PRODUCT_DESCRIPTION NOPRINT BY PRODUCT_ID
ON PRODUCT_DESCRIPTION SUBFOOT
"Summary for <PRODUCT_DESCRIPTION"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT PDF
ON TABLE SET SQUEEZE ON
ON TABLE SET STYLESHEET *
TYPE=SUBFOOT, STYLE=BOLD, BOTTOMGAP=.25, $
ENDSTYLE
END
```

出力結果は次のとおりです。

```
Product              Unit
Code      Package    Price

G110      Case      125.00
Summary for Coffee Grinder

G121      Case      140.00
Summary for Coffee Pot

B142      Pounds     81.00
Summary for French Roast

B141      Pounds     58.00
Summary for Hazelnut

B144      Pounds     76.00
Summary for Kona
```

ページ下部へのページ脚注の配置

ページの下部にページ脚注を配置することができます。デフォルト設定では、ページ脚注はレポートデータの 2 行下に表示されます。

```
FOOTING [BOTTOM]
  "content ... "
["content ... "]
.
.
.
["content ... "]
```

説明

FOOTING

ページ脚注として内容を識別するために必要なコマンドです。

BOTTOM

ページの最下部に脚注を配置するオプションのコマンドです。BOTTOM を省略した場合、ページ脚注はレポートデータの下から 2 行目に表示されます。

注意：FOOTING BOTTOM は、HTML レポートおよび Web ビューアではサポートされません。

content

脚注内容には、二重引用符 (") の間に次の要素を含めることができます。終了用引用符が省略された場合、リクエストの後に続くすべての行は、脚注の一部として扱われます。

text

脚注のテキストです。複数行のテキストを含めることができます。

テキストは、テキストのみの行で開始する必要があり、FOOTING コマンドが続きます。

テキストは、変数およびスポットマーカーと組み合わせることができます。

variable

次の 1 つまたは組み合わせです。

`Fields` - 実データソースフィールド、マスターファイルまたはリクエスト内で DEFINE コマンドを使用して作成した一時項目、リクエスト内で COMPUTE コマンドを使用して作成した一時項目、または TABPAGENO などのシステムフィールドです。データソースフィールドは、特定の演算接頭語で修飾することができます。

```
Dialogue Manager variables.
```

`Images` - 見出しまたは脚注にイメージを含めることができます。

詳細は、見出しまたは脚注内への要素の追加を参照してください。

スポットマーカー

項目の配置、フォーマット設定項目の識別が可能です。

`<+0>` - フォーマット設定を目的として、見出しまたは脚注を項目に分割します。

`</n` - ブランク行を指定します。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。

`<-n` - 行に次の文字を配置します。詳細は、見出し、脚注、その内部の項目の配置を参照してください。

`<0X` - 見出しまたは脚注指定を、リクエストの次の行に継続します。

注意：終了スポットマーカーの直後に開始スポットマーカーを追加すると (><)、この 2 つのスポットマーカー間にブランクが 1 つ挿入されます (> <)。フォーマット設定を適用する際は、このことを考慮する必要があります。

ブランク行

すべてのテキスト、変数、スポットマーカーを省略する場合、見出しや脚注がブランクになります (例、" ")。これを使用して、ブランク行を挿入することができます。ブランク行を挿入するには、</1 などの縦のスポットマーカーを使用することもできます。

注意：1 つのリクエストで使用可能なソート項目を指定している見出しとソート項目を指定している脚注の合計の最大数は、33 バイトです。

ページ末尾へのページ脚注の配置

次のリクエストは、レポートデータの 2 行下のデフォルトの位置ではなく、ページの一番下にページ脚注を配置する PDF レポートを作成します。

```
TABLE FILE GGSALES
PRINT UNITS DOLLARS
BY CATEGORY BY STCD
WHERE TOTAL DOLLARS GE 25000
FOOTING BOTTOM
"PRELIMINARY SALES FIGURES"
ON TABLE SET ONLINE-FMT PDF
ON TABLE SET PAGE-NUM OFF
END
```

次の出力結果は、レポートの下部に脚注を表示します。

```
               R1088            1676          25140
                                1789          26835
               R1100            1781          26715
                                3015          45225
               R1109            1694          25410
               R1200            1765          26475
                                1675          25125
                                1773          26595
               R1244            1800          25200
               R1248            1798          26970
                                1695          25425
                                1699          25485
Gifts          R1020            3609          47468
               R1200            3473          51180


 PRELIMINARY SALES FIGURES
```

レポート見出しまたは脚注の単独ページ配置

Web ビューアで表示する PDF または HTML レポートで、レポート見出しまたは脚注を単独ページに配置して、重要な情報を強調するリクエストを作成することができます。たとえば、給与情報に「Confidential」という印が付けられた表紙を作成し、実際のデータから分離することができます。

レポート見出しまたは脚注の単独ページ配置

見出しおよび脚注の各行は、二重引用符 (") で開始、終了する必要があります。

```
ON TABLE PAGE-BREAK AND {SUBHEAD|SUBFOOT}
"content ... "
["content ... "]
.
.
.
["content ... "]
```

説明

PAGE-BREAK

新しいページの開始を指定します。SET LINES コマンドとともに使用することで、印刷するページの長さを制御します。

SUBHEAD

レポートの見出しを生成します。

SUBFOOT

レポート脚注を生成します。

content

見出しまたは脚注の内容には、次の要素を二重引用符 (") で囲んで含めることができます。終了用引用符が省略された場合、リクエストの後に続くすべての行は、レポート見出しの一部分として扱われます。

text

見出しまたは脚注のテキストです。複数行のテキストを含めることができます。

テキストは、SUBHEAD または SUBFOOT コマンドに続いてテキストのみの行で開始する必要があります。

テキストは、変数およびスポットマーカーと組み合わせることができます。

variable

次の 1 つまたは組み合わせです。

`Fields` - 実データソースフィールド、マスターファイルまたはリクエスト内で DEFINE コマンドを使用して作成した一時項目、リクエスト内で COMPUTE コマンドを使用して作成した一時項目、または TABPAGENO などのシステムフィールドです。データソースフィールドは、特定の演算接頭語で修飾することができます。

```
Dialogue Manager variables.
```

`Images` - 見出しまたは脚注にイメージを含めることができます。

詳細は、見出しまたは脚注内への要素の追加を参照してください。

スポットマーカー

項目の配置、フォーマット設定項目の識別が可能です。

`<+0>` - フォーマット設定を目的として、見出しまたは脚注を項目に分割します。

`</n` - ブランク行を指定します。詳細は、見出しまたは脚注の縦方向配置制御を参照してください。

`<-n` - 行に次の文字を配置します。詳細は、見出し、脚注、その内部の項目の配置を参照してください。

`<0X` - 見出しまたは脚注指定を、リクエストの次の行に継続します。

注意：終了スポットマーカーの直後に開始スポットマーカーを追加すると (><)、この 2 つのスポットマーカー間にブランクが 1 つ挿入されます (> <)。フォーマット設定を適用する際は、このことを考慮する必要があります。

ブランク行

すべてのテキスト、変数、スポットマーカーを省略する場合、見出しや脚注がブランクになります (例、" ")。これを使用して、ブランク行を挿入することができます。ブランク行を挿入するには、</1 などの縦のスポットマーカーを使用することもできます。

別ページへのレポート見出しの配置

次のリクエストでは、PAGE-BREAK を使用して 2 ページのレポートを生成し、最初のページに重要な情報のみを表示します。

```
TABLE FILE CENTORD
SUM ORDER_DATE LINEPRICE AS 'Order,Total:'
BY HIGHEST 5 ORDER_NUM
ON TABLE PAGE-BREAK AND SUBHEAD
"CONFIDENTIAL COMPANY INFORMATION"
"March 2003"
HEADING
"Order Revenue"
" "
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT PDF
END
```

出力結果の最初のページは、レポートおよびデータが社外秘であることを示しています。

```
CONFIDENTIAL COMPANY INFORMATION
March 2003
```

出力結果の 2 ページ目には、フィールドタイトルおよびデータが含まれています。

```
Order Revenue

                                        Order
Order Number:  Date Of Order:           Total:
94710          2001/01/02          $406,964.24
94680          2001/01/02          $421,916.60
94670          2001/01/02          $513,868.76
94550          2001/01/02          $496,323.64
94530          2001/01/02            $3,472.41
```

ヒント：これに相当する結果を HTML 形式で生成するには、Web ビューアをオンにし、リクエストに次のコードを含めます。

```
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET WEBVIEWER ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

最初のページには、レポートの見出しが表示されます。レポートデータの 2 ページ目へ移動することができます。

レポート脚注の別ページへの配置

次のリクエストは、PAGE-BREAK を使用して、2 ページのレポートを作成します。レポートの終わりを示す脚注は、単独ページとして作成されます。

```
TABLE FILE CENTORD
HEADING
"Order Revenue"
" "
SUM ORDER_DATE LINEPRICE AS 'Order,Total:'
BY HIGHEST 5 ORDER_NUM
ON TABLE PAGE-BREAK AND SUBFOOT
"END OF REPORT"
ON TABLE SET PAGE-NUM OFF
ON TABLE SET ONLINE-FMT PDF
END
```

出力結果の最初のページには、フィールドタイトルおよびデータが含まれています。

```
Order Revenue

Order Number:  Date Of Order:          Order
                                       Total:
94710          2001/01/02          $406,964.24
94680          2001/01/02          $421,916.60
94670          2001/01/02          $513,868.76
94550          2001/01/02          $496,323.64
94530          2001/01/02            $3,472.41
```

出力結果の最後のページは、レポートの終了を知らせます。

```
END OF REPORT
```

注意：これに相当する結果を HTML 形式で生成するには、Web ビューアをオンにし、リクエストに次のコードを含めます。

```
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET WEBVIEWER ON
ON TABLE SET STYLESHEET *
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果の最初のページには、フィールドタイトルおよびデータが含まれています。最後のページへ移動すると、「END OF REPORT」と表示されています。

レポートデータのフォーマット

この章では、レポートテキストのフォーマット設定および配置についての情報を提供します。レポートテキストの位置を調整できることに加えて、レポートテキストで使用するサイズ、色、スタイルを選択できます。

レポートのフォントフォーマット指定

スタイルシート属性を使用して、フォント、サイズ、およびフォントの色を指定することにより、レポートの外観を向上させることができます。フォントのフォーマットは、レポート全体に対して指定することも、見出し、脚注、列のそれぞれに対して個別に指定することもできます。

レポートのフォントサイズ指定

フォントサイズを指定するには、スタイルシートで次の構文を使用します。

```
TYPE = type, [subtype,] SIZE=pts, $
```

説明

type

REPORT、HEADING、TITLE など、対象となるレポートコンポーネントです。

subtype

フォーマットを設定するレポートコンポーネントを識別するための任意の追加属性です(例、COLUMN、ACROSS、ITEM)。別のレポートコンポーネントの指定方法についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

pts

ポイントで示すフォントサイズです。デフォルト値は 10 です。これは HTML デフォルトフォントサイズ 3 に対応します。ポイントサイズと HTML フォントサイズの相関関係についての詳細は、レポートのフォントフォーマット指定を参照してください。

レポートのフォントサイズ指定

次のリクエストでは、フィールドタイトルのポイントサイズが 12 に設定されます。

```
TABLE FILE GGSALES
ON TABLE SET PAGE-NUM OFF
SUM UNITS DOLLARS BY CATEGORY
ON TABLE SET STYLE *
TYPE=TITLE, SIZE=12, $
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Category	Unit Sales	Dollar Sales
Coffee	1376266	17231465
Food	1384845	17229344
Gifts	927880	11695502

フォントサイズ変更時の注意

HTML レポート以外は、ポイントサイズは固定です。相対ポイントサイズは、HTML フォントサイズとは異なるサイズを使用します。下表は、ポイントサイズおよび対応する HTML フォントサイズの一覧です。

ポイントサイズ	対応する HTML フォントサイズ
8 以下	1
9	2

ポイントサイズ	対応する HTML フォントサイズ
10	3
11	4
12	5
13	6
14 以上	7

レポートの太字、斜体フォント指定

フォントスタイルを指定するには、スタイルシートで次の構文を使用します。

```
TYPE=type, [subtype,] STYLE=[+|-]txtsty[{+|-}txtsty], $
```

説明

type

REPORT、HEADING、TITLE など、対象となるレポートコンポーネントです。

subtype

フォーマットを設定するレポートコンポーネントを識別するための任意の追加属性です(例、COLUMN、ACROSS、ITEM)。別のレポートコンポーネントの指定方法についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

txtsty

NORMAL、BOLD、ITALIC の値のいずれかです。デフォルト値は NORMAL です。

指定したスタイルが、使用するフォントでサポートされていない場合、そのフォントはスタイルなしで表示されます。

+

フォントスタイルの組み合わせを指定できるようにします。複数のフォントスタイルを適用した属性に、さらにフォントスタイルを追加することができます。

-

属性からフォントスタイルを削除できるようにします。

レポートのフォントスタイルの指定

次のレポートでは、フィールドタイトルは太字、斜体フォントスタイルが指定されます。

```
TABLE FILE GGSALES
SUM UNITS DOLLARS BY CATEGORY
ON TABLE SET STYLE *
TYPE=TITLE, STYLE=BOLD+ITALIC, $
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

Category	*Unit Sales*	*Dollar Sales*
Coffee	1376266	17231465
Food	1384845	17229344
Gifts	927880	11695502

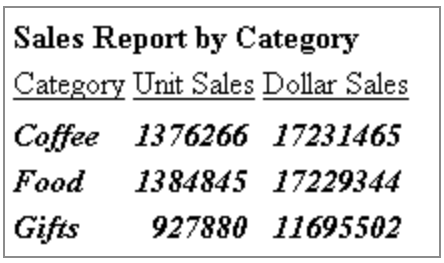
レポートへの継承フォントスタイルの追加および削除

次のリクエストでは、レポート全体に太字、斜体フォントスタイルが指定されます。継承されたスタイルのうち、見出しからは斜体が、フィールドタイトルからはその両方が削除されます。

```
TABLE FILE GGSALES
HEADING
"Sales Report by Category"
SUM UNITS DOLLARS BY CATEGORY
ON TABLE SET STYLE *
TYPE=REPORT, STYLE=BOLD+ITALIC, $
TYPE=HEADING, STYLE=-ITALIC, $
```

```
TYPE=TITLE, STYLE=-BOLD-ITALIC, $
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。



Sales Report by Category

Category	Unit Sales	Dollar Sales
Coffee	*1376266*	*17231465*
Food	*1384845*	*17229344*
Gifts	*927880*	*11695502*

レポートのフォント色指定

レポートまたはレポートコンポーネントのフォントの色を指定するには、スタイルシートで次の構文を使用します。

```
TYPE=type, [subtype,] COLOR={color|RGB({r g b|#hexcolor})},$
```

説明

type

REPORT、HEADING、TITLE など、対象となるレポートコンポーネントです。

subtype

フォーマットを設定するレポートコンポーネントを識別するための任意の追加属性です(例、COLUMN、ACROSS、ITEM)。別のレポートコンポーネントの指定方法についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

color

GREY や GOLD など、定義済みの色の値です。ディスプレイデバイスまたは出力デバイスが指定した色をサポートしない場合、灰色に置き換えられます。デフォルト値は、BLACK です。利用可能な色の値の完全なリストは、レポートのフォントフォーマット指定を参照してください。

RGB(r g b)

赤、緑、青の組み合わせを使用して、色を指定します。

(r, g, b) は、赤、緑、青の順序で指定する輝度です。0 (ゼロ) から 255 までの値が使用されます。0 (ゼロ) は最も輝度が低く、255 は最も輝度が高いことを示します。色の構成要素の輝度をすべて同じ値に設定すると、灰色になります。

RGB(#hexcolor)

色の 16 進数値です。たとえば、赤色の 16 進数値は FF0000 です。16 進数値では大文字と小文字は区別されません。また、先頭にシャープ記号 (#) を付ける必要があります。

レポートの色値

下表は、構文で利用可能なすべての色の値を一覧表示しています。

```
COLOR=color, or BACKCOLOR=color,
```

color は次の値の 1 つです。

AQUA (CYAN)	MEDIUM FOREST GREEN (OLIVE)
AQUAMARINE	MEDIUM GOLDENROD
BLACK	MEDIUM ORCHID
BLUE VIOLET	MEDIUM SLATE BLUE
CADET BLUE	MEDIUM SPRING GREEN

CORAL	MEDIUM TURQUOISE
CORNFLOWER BLUE	MEDIUM VIOLET RED
CYAN (AQUA)	MIDNIGHT BLUE
DARK GREEN	NAVY (NAVY BLUE)
DARK OLIVE GREEN	OLIVE (MEDIUM FOREST GREEN)
DARK ORCHID	ORANGE
DARK SLATE BLUE (PURPLE)	ORANGE RED
DARK SLATE GREY	ORCHID
DARK TURQUOISE	PALE GREEN
DIM GREY (GRAY, GREY)	PINK
FIREBRICK	PLUM

FOREST GREEN (GREEN)	PURPLE (DARK SLATE BLUE)
FUCHSIA (MAGENTA)	RED
GOLD	SALMON
GOLDENROD	SEA GREEN
GRAY (DIM GREY, GREY)	SIENNA
GREEN (FOREST GREEN)	SILVER
GREEN YELLOW	SKY BLUE
GREY (DIM GREY, GRAY)	SLATE BLUE
INDIAN RED	STEEL BLUE (TEAL)
KHAKI	TAN
LIGHT BLUE	TEAL (STEEL BLUE)

LIGHT GREY	THISTLE
LIGHT STEEL BLUE	TURQUOISE
LIME	VIOLET
LIME GREEN	VIOLET RED
MAGENTA (FUCHSIA)	WHEAT
MAROON	WHITE
MEDIUM AQUAMARINE	YELLOW
MEDIUM BLUE	YELLOW GREEN

レポートのフォント指定

スタイルシートで FONT 属性を使用することにより、レポートに独自のフォントを指定できます。HTML レポートのフォントを指定する場合、そのフォントが Web ブラウザでサポートされている必要があります。Web ブラウザがそのフォントをサポートしない場合、デフォルトプロポーショナルフォントを使用する、デフォルト表示に戻ります。

レポートフォントの指定

レポートのフォントを指定するには、スタイルシートで次の構文を使用します。

```
TYPE=type, [subtype,] FONT='font[,font]',$
```

説明

type

REPORT、HEADING、TITLE など、対象となるレポートコンポーネントです。

subtype

フォーマットを設定するレポートコンポーネントを識別するための任意の追加属性です (例、COLUMN、ACROSS、ITEM)。別のレポートコンポーネントの指定方法についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

font

フォントの名前です。一重引用符 (') で値を囲む必要があります。HTML レポートを作成する場合、複数のブラウザに対応するため、一重引用符 (') 内に複数のフォントを指定することができます。

注意：HTML レポートで、異なるレポートコンポーネントに別のフォントを複数指定すると、ソースコードのサイズが大幅に増加します。

HTML レポートでの複数フォントの指定

複数プラットホームでのレポートの外観を制御するため、リクエストに共通 Windows フォント、および共通 UNIX フォントを両方指定することができます。Web ブラウザは、リストで最初のフォントを検索します。ブラウザが最初のフォントを見つけられない場合、リストの次のフォントを検索します。フォントを識別することができない場合、ブラウザはデフォルトのプロポーショナルフォントを使用します。

次の例では、Web ブラウザは Arial フォントを最初に検索します。ブラウザが Arial フォントを見つけられない場合、Helvetica フォントを検索します。両方のフォントを識別できない場合、ブラウザはデフォルトのプロポーショナルフォントを使用します。

```
TYPE=REPORT, FONT='ARIAL,HELVETICA',$
```

HTML レポートのデフォルトブラウザフォント指定

ブラウザは、デフォルトのプロポーショナルフォントおよび等幅フォントとして、特定のフォントを割り当てます。HTML レポートのデフォルトブラウザフォントを指定するには、レポートのスタイルシートで予約名 DEFAULT-PROPORTIONAL および DEFAULT-FIXED を使用します。ブラウザは、それに従ってレポートを表示します。

ブラウザのデフォルト固定フォントおよびプロポーショナルフォントを選択するには、次の構文を使用します。対象となるレポートコンポーネントを示すために、TYPE を指定する必要があります。

```
FONT={DEFAULT-PROPORTIONAL|DEFAULT-FIXED},$
```

説明

DEFAULT-PROPORTIONAL

Web ブラウザのデフォルトプロポーショナルフォントを指定します。

DEFAULT-FIXED

Web ブラウザのデフォルト等幅フォントを指定します。

デフォルトブラウザフォントの指定

次の例では、Web ブラウザは、レポートの見出しおよびフィールドタイトルを除いて、レポート全体にデフォルト等幅フォントを使用します。レポートの見出しおよびフィールドタイトルには、デフォルトプロポーショナルフォントを使用します。

```
TABLE FILE GGSALES
HEADING
"Sales Report"
SUM UNITS DOLLARS BY CATEGORY BY PRODUCT
ON TABLE SET STYLE *
TYPE=REPORT,FONT=DEFAULT-FIXED,$
TYPE=TITLE,FONT=DEFAULT-PROPORTIONAL,$
TYPE=HEADING,FONT=DEFAULT-PROPORTIONAL,$
TYPE=REPORT, GRID=OFF, $
```

```
ENDSTYLE
END
```

出力結果は次のとおりです。

Sales Report

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381600
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263328
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

レポート背景色の指定

スタイルシート属性で背景色を指定することにより、レポートの外観を向上させることができます。背景色は、レポート全体に対して指定することも、見出し、脚注、列のそれぞれに対して個別に指定することもできます。

また、レポートのデータ行の背景色を交互に表示する機能 (色バンド) を使用することもできます。背景色を交互に表示する機能は、列の積み重ね (OVER、FOLD-LINE) ではサポートされません。

レポート背景色の指定

レポートの背景色を指定するには、スタイルシートで次の構文を使用します。

PDF レポートで BACKCOLOR を使用すると、レポートデータの各セル上部、下部、右側、および左側にブランクが追加されます。これは読みやすくし、データが一部非表示になることを防ぐためです。

```
TYPE=type, [subtype,] BACKCOLOR={color|RGB({r g b|#hexcolor})}, $
```

説明

type

REPORT、HEADING、TITLE など、対象となるレポートコンポーネントです。HEADING、FOOTING、SUBHEADING、SUBFOOTING では、個々の要素に背景色を指定することができます。

subtype

フォーマットを設定するレポートコンポーネントを識別するための任意の追加属性です (例、COLUMN、ACROSS、ITEM)。HEADING、FOOTING、SUBHEADING、SUBFOOTING では、個々の要素に背景色を指定することができます。異なるタイプのレポートコンポーネントを指定する方法についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

color

指定したレポートコンポーネントの背景色です。デフォルト値は NONE です。HTML フォーマットでレポートを作成した場合、背景色はそれをサポートする Web ブラウザ内でのみ表示されます。

RGB (r g b)

赤、緑、青の組み合わせを使用して、色を指定します。

(r, g, b) は、赤、緑、青の順序で指定する輝度です。0 (ゼロ) から 255 までの値が使用されます。0 (ゼロ) は最も輝度が低く、255 は最も輝度が高いことを示します。色の構成要素の輝度をすべて同じ値に設定すると、灰色になります。

RGB (#hexcolor)

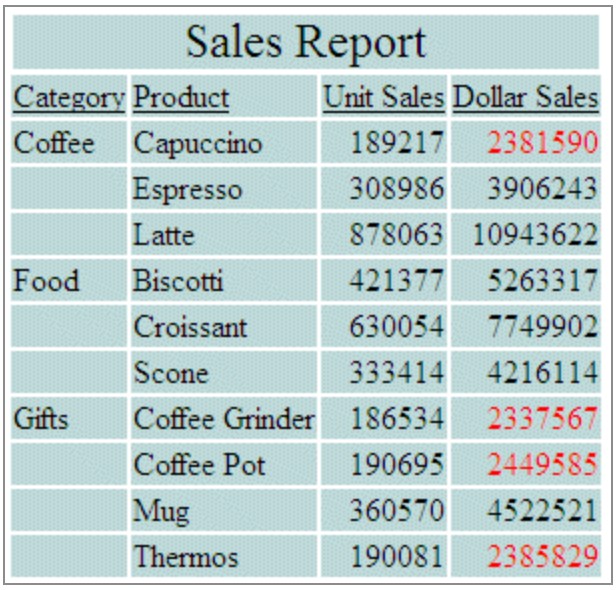
色の 16 進数値です。たとえば、赤色の 16 進数値は FF0000 です。16 進数値では大文字と小文字は区別されません。また、先頭にシャープ記号 (#) を付ける必要があります。

レポート背景色およびフォント色の指定

レポートに色を使用して、レポート内の重要な情報を強調することができます。次のリクエストでは、Dollar Sales フィールドのデータ値が 2,500,000 ドルより小さい場合に RED が指定されます。背景色は LIGHT BLUE に設定されています。

```
TABLE FILE GGSALES
ON TABLE SET PAGE-NUM OFF
HEADING
"Sales Report"
SUM UNITS DOLLARS BY CATEGORY BY PRODUCT
ON TABLE SET STYLE *
TYPE=REPORT, BACKCOLOR=LIGHT BLUE, $
TYPE=DATA, COLUMN=DOLLARS, COLOR=RED, WHEN=DOLLARS LT 2500000, $
TYPE=REPORT, GRID=OFF, $
TYPE=HEADING, JUSTIFY=CENTER, SIZE=12,$
ENDSTYLE
END
```

出力結果は次のとおりです。



Sales Report

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

レポートデータ行の代替背景色の指定

レポートのデータ行に代替背景色を指定するには、スタイルシートで次の構文を使用します。

PDF レポートで BACKCOLOR を使用すると、レポートデータの各セル上部、下部、右側、および左側にブランクが追加されます。これは読みやすくし、データが一部非表示になることを防ぐためです。

```
TYPE=DATA,BACKCOLOR=({c1|RGB({r1 g1 b1|#hc1})} {c2|RGB({r2 g2 b2|#hc2})}),$
```

説明

c1, c2

レポートのデータ行に使用する背景色です。デフォルト値は NONE です。HTML フォーマットでレポートを作成した場合、背景色はそれをサポートする Web ブラウザ内でのみ表示されます。色の名前にブランクが含まれている場合、そのブランクは代替色値の区切り文字と見なされるため、色の名前を一重引用符 (') で囲む必要があります。

RGB(r1 g1 b1), RGB(r2 g2 b2)

赤、緑、青の組み合わせを使用して、背景色を指定します。

(r, g, b) は、赤、緑、青の順序で指定する輝度です。0 (ゼロ) から 255 までの値が使用されます。0 (ゼロ) は最も輝度が低く、255 は最も輝度が高いことを示します。色の構成要素の輝度をすべて同じ値に設定すると、灰色になります。

RGB(#hc1), RGB(#hc2)

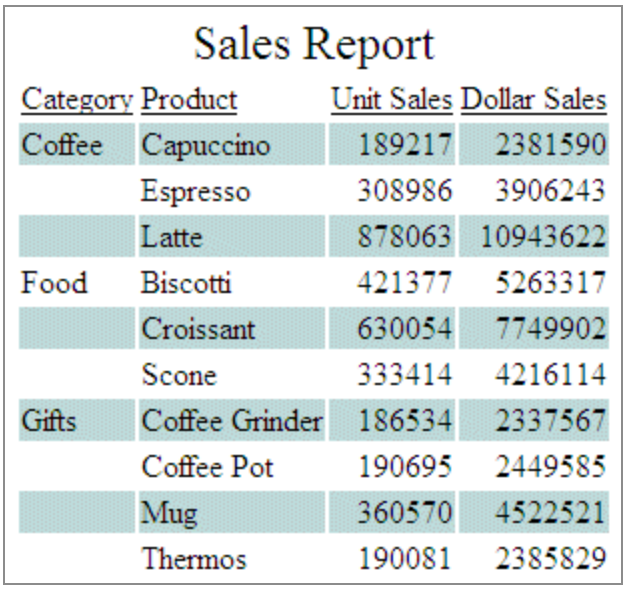
色の 16 進数値です。たとえば、赤色の 16 進数値は FF0000 です。16 進数値では大文字と小文字は区別されません。また、先頭にシャープ記号 (#) を付ける必要があります。

レポートデータ行の代替背景色の指定

次のリクエストは、GGSALES データソースを使用し、レポート出力のデータ行に水色と白色の代替色を適用します。

```
TABLE FILE GGSALES
ON TABLE SET PAGE-NUM OFF
HEADING
"Sales Report"
SUM UNITS DOLLARS
BY CATEGORY
BY PRODUCT
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=DATA, BACKCOLOR=('LIGHT BLUE' WHITE),$
TYPE=HEADING, JUSTIFY=CENTER, SIZE=12,$
ENDSTYLE
END
```

出力結果は次のとおりです。



Sales Report

Category	Product	Unit Sales	Dollar Sales
Coffee	Capuccino	189217	2381590
	Espresso	308986	3906243
	Latte	878063	10943622
Food	Biscotti	421377	5263317
	Croissant	630054	7749902
	Scone	333414	4216114
Gifts	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Mug	360570	4522521
	Thermos	190081	2385829

折り返し行への交互に異なる背景色の適用

ALTBACKPERLINE 属性は、位置決めドライバを使用するレポート (例、PDF、DHTML、PPT、PPTX) で、1 行おきに異なる背景色を表示します。これにより、長いフィールド値を折り返した場合に、境界とは関係なく、各値内の折り返し行ごとに交互に異なる背景色を表示することができます。1 行おきに異なる背景色を適用するには、WRAP を使用するプロシジャに SET ALTBACKPERLINE=ON コマンドを明示的に追加する必要があります。

折り返し行への交互に異なる背景色の適用

```
SET ALTBACKPERLINE = {ON|OFF}
```

説明

ON

各値内の折り返し行ごとに交互に異なる背景色を表示します。

OFF

フィールド値ごとに交互に異なる背景色を表示します。これがデフォルト値です。

折り返し行への交互に異なる背景色の適用

次のリクエストは、WRAP を使用して長い行の一時項目 (COMPUTE) テキストを表示します。SET ALTBACKPERLINE コマンドは使用しません。デフォルト設定の交互に異なる背景色は、折り返しを含めた行単位で適用されます。

```
TABLE FILE WF_RETAIL PRINT
COMPUTE LONG_LINE/A1000 = 'This is a very long line of data, set up so that'
                          |'it wraps onto the next line within a report'
                          |'and to test the alternate color syntax'
                          |'in style sheets.';
BY PRODUCT_CATEGORY BY PRODUCT_SUBCATEG BY MODEL
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET PAGE OFF
ON TABLE SET STYLE *
UNITS=IN, PAGESIZE='A4', LEFTMARGIN=0.19, TOPMARGIN=0.00, BOTTOMMARGIN=0.00,
SQUEEZE=ON, ORIENTATION=LANDSCAPE, $
TYPE=REPORT, FONT='ARIAL', SIZE=8, COLOR=BLACK, BACKCOLOR='NONE',
STYLE=NORMAL, $
TYPE=REPORT, COLUMN=N4, WRAP=1, $
TYPE=DATA, COLUMN=N4, BACKCOLOR=(RGB(235 240 178) RGB(255 255 255)), $
ENDSTYLE
END
```

出力結果は次のとおりです。

Product Category	Product Subcategory	Model	LONG_LINE
Accessories	Charger	B00D7MOHDO	This is a very long line of data, set up so thatit wraps onto the next line within a reportand to test the alternate color syntaxin style sheets.
		BCG34HRE4KN	This is a very long line of data, set up so thatit wraps onto the next line within a reportand to test the alternate color syntaxin style sheets.
	Headphones	Audio Technica ATHW5000	This is a very long line of data, set up so thatit wraps onto the next line within a reportand to test the alternate color syntaxin style sheets.
		Denon AHD5000	This is a very long line of data, set up so thatit wraps onto the next line within a reportand to test the alternate color syntaxin style sheets.

リクエストに SET ALTBACKPERLINE=ON を追加すると、下図のように、交互に異なる背景色は、折り返しを含めない行単位で適用されます。

Product Category	Product Subcategory	Model	LONG_LINE
Accessories	Charger	B00D7MOHDO	This is a very long line of data, set up so thatit wraps onto the next line within a reportand to test the alternate color syntaxin style sheets.
		BCG34HRE4KN	This is a very long line of data, set up so thatit wraps onto the next line within a reportand to test the alternate color syntaxin style sheets.
	Headphones	Audio Technica ATHW5000	This is a very long line of data, set up so thatit wraps onto the next line within a reportand to test the alternate color syntaxin style sheets.
		Denon AHD5000	This is a very long line of data, set up so thatit wraps onto the next line within a reportand to test the alternate color syntaxin style sheets.

レポートのデータフォーマットの指定

レポート内のデータ表示は、いくつかの方法で制御することができます。

- フィールドに表示する数値のフォーマットを変更することができます。
- コンチネンタル式 10 進表記 (CDN) を使用することができます。
- データを持たないフィールドを表すために、特定の文字または文字セットを設定することができます。

レポートフィールド値のフォーマット変更

フィールドのフォーマットは、マスターファイル内で定義されます。ただし、レポートフィールドのフォーマットは、変更することができます。

フィールド値のフォーマット変更

```
fieldname [alignment] [/format]
```

説明

fieldname

表示フィールドです。PRINT、LIST、SUM、COUNT コマンド、行合計、または列合計により表示されます。

alignment

フィールドタイトルの位置を指定します。

/R - フィールドタイトルを右揃えにします。

/L - フィールドタイトルを左揃えにします。

/C - フィールドタイトルを中央揃えにします。

format

有効なフィールドのフォーマットで、スラッシュ (/) の後に指定します。フィールドフォーマットは、フィールド合計には使用できません。

フィールド値のフォーマット変更

UNIT_PRICE フィールドのフォーマットは、GGPRODS マスターファイル D7.2 に定義されています。表示に浮動ドルサインを追加するには、フィールドフォーマットを次のように再定義します。

```
TABLE FILE GGPRODS
PRINT UNIT_PRICE/D7.2M
END
```

出力結果は次のとおりです。

```
    Unit
   Price  $58.00
  $81.00
  $76.00
  $13.00
  $17.00
  $28.00
  $26.00
  $96.00
 $125.00
 $140.00
```

フィールドの複数フォーマット指定

次のリクエストは、フォーマット指定よるにフィードタイトルの位置揃え、BY フィールド指定、AS 句指定の例を示しています。

```
TABLE FILE CAR
PRINT MODEL/A10 STANDARD/A15/R AS 'RJUST,STANDARD'
BY CAR/C
WHERE CAR EQ 'JAGUAR' OR 'TOYOTA'
END
```

出力結果は次のとおりです。

```
                                    RJUST
       CAR        MODEL             STANDARD
----------------  -----       ---------------
JAGUAR            V12XKE AUT  POWER STEERING
                  XJ12L AUTO  RECLINING BUCKE
                              WHITEWALL RADIA
                              WRAP AROUND BUM
                              4 WHEEL DISC BR
TOYOTA            COROLLA 4   BODY SIDE MOLDI
                              MACPHERSON STRU
```

フィールドフォーマット変更時の注意

- フィールドフォーマットを再設定するたびに、1 つのレポート内の表示フィールドの制限に対して、フィールドは 2 回計算されます。詳細は、レポートフォーマット設定の管理を参照してください。
- レポートから HOLD、PCHOLD、SAVE、SAVB などの抽出ファイルを作成する場合、HOLDLIST=PRINTONLY を指定しない限り、抽出ファイルのフィールドには元のフォーマットと再定義したフォーマットの両方が格納されます。抽出ファイルについては、レポート出力の保存と再利用を参照してください。
- テキストフィールドインスタンス内の語句のサイズがマスターファイルのテキストフィールドフォーマットを超える場合は、その語句は2 行目に折り返され、その次の語句は同じ行で開始されます。
- 表示フィールド、BY フィールド、および ACROSS フィールドは、位置揃えを指定することができます。ACROSS フィールドの場合、フィールドタイトルではないデータ値は、指定どおりに位置揃えされます。
- 表示コマンドに限り、位置揃えパラメータをフォーマット指定と併用することができます。フォーマット指定は位置揃えパラメータの前、または後に配置することができます。
- タイトルが AS 句またはマスターファイルで指定される場合、そのタイトルの位置揃えは、FORMAT による指定に従います。
- リクエストに複数の ACROSS フィールドが指定される場合は、最下位 の ACROSS フィールドのみで位置揃えされます。ACROSS フィールドのその他の位置揃えパラメータは、すべて無視されます。

フォーマットが再設定されたフィールドのミッシング値の制御

リクエスト内でフィールドにフォーマットを再設定 (例、*field*/*format*) すると、内部的に一時項目 (COMPUTE) が作成されてフォーマットの再設定されたフィールド値が格納されます。レポート出力へはこのフィールド値が表示されます。元のフィールドにミッシング値が含まれている場合は、COMPMISS パラメータを ON に設定することで、そのミッシング値を内部フィールドに継承することができるようになります。ミッシング値が内部フィールドに継承されない場合は、ゼロ (数字フィールド) またはブランク (文字フィールド) が表示されま

す。ミッシング値が内部フィールドに継承される場合は、レポート出力にミッシングデータシンボルが表示されます。

フォーマット再設定フィールドのミッシング値制御

```
SET COMPMISS = {ON|OFF}
```

説明

ON

ミッシング値をフォーマットが再設定されたフィールドに継承させます。デフォルト値は ON です。

OFF

フォーマットが再設定されたフィールドにブランクまたは 0 (ゼロ) を表示します。

フォーマット再設定フィールドのミッシング値制御

次のプロシジャは、SALES データソースの店舗 14Z からの RETURNTS フィールドを表示します。COMPMISS を OFF に設定した場合、フォーマットが再設定されたフィールド値のミッシング値は 0 (ゼロ) として列に表示されます。

注意：この例を実行する前に、SALES データソースにミッシング値を追加する SALEMISS プロシジャを実行しておく必要があります。

```
SET COMPMISS = OFF
TABLE FILE SALES
PRINT RETURNS RETURNS/D12.2 AS 'REFORMATTED,RETURNS'
BY STORE_CODE
WHERE STORE_CODE EQ '14Z'
END
```

出力結果は次のとおりです。

```
                         REFORMATTED
STORE_CODE  RETURNS      RETURNS
----------  -------      -----------
14Z                2             2.00
                   2             2.00
                   0              .00
                   .              .00
                   4             4.00
                   0              .00
                   3             3.00
                   4             4.00
                   .              .00
                   4             4.00
```

COMPMISS を ON に設定した場合、フォーマットが再設定された RETURNS 列の値が欠落している箇所には、ミッシングデータ記号が表示されます。

```
SET COMPMISS = ON
TABLE FILE SALES
PRINT RETURNS RETURNS/D12.2 AS 'REFORMATTED,RETURNS'
BY STORE_CODE
WHERE STORE_CODE EQ '14Z'
END
```

出力結果は次のとおりです。

```
                         REFORMATTED
STORE_CODE  RETURNS      RETURNS
----------  -------      -----------
14Z                2             2.00
                   2             2.00
                   0              .00
                   .                .
                   4             4.00
                   0              .00
                   3             3.00
                   4             4.00
                   .                .
                   4             4.00
```

SET COMPMISS の使用上の注意

- COMPMISS を ON に設定して HOLD ファイルを作成する場合、HOLD マスターファイルのフォーマットの再設定されたフィールドには、元のフィールドと同様に MISSING = ON が記述されます。COMPMISS を OFF に設定する場合、作成したマスターファイルのフォーマットを再設定したフィールドには、MISSING = ON は記述されません。
- COMPMISS パラメータを ON TABLE コマンドで設定することはできません。

カンマ表記とコンチネンタル 10 進表記の相違

CDN パラメータは、数値内で使用する区切り文字を指定します。CDN パラメータについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

ミッシング値を表す文字の設定

フィールドにデータが存在しない場合に表示する、独自の文字列を指定することで、レポート出力の外観を変更できます。

ミッシング値を表す文字の設定

NODATA フィールドの文字列を指定するには、次の構文を使用します。

```
ON TABLE SET NODATA character string
```

説明

NODATA

NODATA 文字が設定されることを示します。

character string

フィールドにデータが存在しないときに表示する文字列です。デフォルト値は、ピリオド (.) です。

SET コマンドによる NODATA 文字の設定

NODATA フィールドの文字を指定するには、次の構文を使用します。

```
SET NODATA=character
```

説明

character

フィールドにデータが存在しないときに表示する文字または文字列です。最大文字数は 11 バイトです。デフォルト値は、ピリオド (.) です。

リクエストでの NODATA 文字の設定

次のリクエストは、ミッシングデータの NODATA 文字をピリオド (.) から文字列 NONE に変更します。

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL
BY LAST_NAME BY FIRST_NAME
ACROSS DEPARTMENT
ON TABLE SET NODATA NONE
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, SQUEEZE=OFF,$
ENDSTYLE
END
```

リクエストの結果は次のとおりです。

		DEPARTMENT	

LAST_NAME	FIRST_NAME	MIS	PRODUCTION
BANNING	JOHN	NONE	$29,700.00
BLACKWOOD	ROSEMARIE	$21,780.00	NONE
CROSS	BARBARA	$27,062.00	NONE
GREENSPAN	MARY	$9,000.00	NONE
IRVING	JOAN	NONE	$26,862.00
JONES	DIANE	$18,480.00	NONE
MCCOY	JOHN	$18,480.00	NONE
MCKNIGHT	ROGER	NONE	$16,100.00
ROMANS	ANTHONY	NONE	$21,120.00
SMITH	MARY	$13,200.00	NONE
	RICHARD	NONE	$9,500.00
STEVENS	ALFRED	NONE	$11,000.00

レポートデータの配置

フィールドの位置揃え (右、左、中央揃え) やセル内でデータを折り返すかどうかを指定することにより、レポートにデータを配置することができます。ページ内でのフィールドの配置についての詳細は、レポートページのレイアウトを参照してください。

レポートデータの折り返し制御

レポートデータの折り返しを制御して、セル内で改行が発生しないようにすることができます。HTML 出力を使用する場合、ほとんどの Web ブラウザのデフォルト設定では、レポートの文字データがセル内の 1 行に収まらない場合にデータが折り返されます。

これにより、セルの内容は次行に送られます。Web ブラウザは、それ自体のアルゴリズム設定に基づいて、データを折り返します。Web ブラウザのデータ折り返し設定を抑制する場合は、WRAP 属性を使用します。

HTML 出力の場合、デフォルト設定で WRAP が ON に指定されているため、各ブラウザでレポート内の列幅を定義することができます。PDF、DHTML、PPT、PPTX 出力の場合、デフォルト設定で WRAP は OFF に指定されています。レポート内の各項目の位置が明示的に定義された位置固定の出力フォーマットでは、WRAP = ON は有効値ではありません。ただし、ACROSSVALUE に対して指定された場合は有効です。レポートのその他の要素では (例、見出し、脚注、タイトル、データ)、数値を WRAP = n として指定することで、折り返し行の幅を定義することができます。

PDF レポート出力では、WRAPGAP 属性を使用して折り返し行の行間を制御することができます。

OVER 句を使用する PDF レポートでのデータの折り返し

OVER を使用して、単一データレコードが、レポート内の複数行に表示されることを防止することができます。デフォルト設定では、レポート内で OVER が定義されている場合、レポートの列表示は、行レベルの表示に変更されます。フィールドタイトルは、各フィールドの先頭ではなく、各値の左側に表示されます。このレイアウトは、特定の方法で位置揃えを行うことを目的としたものではなく、複数のデータ要素を狭い領域に表示することを可能にします。多くの場合、複数の列を積み重ねたレポートでは、これらの列の位置を正しく揃えるためにブランクの AS 名を使用します。OVER 句を使用する PDF レポートでは、WRAP 属性を使用してデータを折り返すことができます。この方法は、ブランクの AS 名との併用も可能です。

ACROSS 句を使用する PDF レポートでのデータの折り返し

ACROSS を使用するリクエストの出力で、ACROSS フィールドの各値は、ACROSS 値に適用されるデータフィールドセットの上部に表示されます。

ACROSS の値がフィールドの列幅よりも長い場合、対応する列幅に収まるよう、ACROSS の値を折り返すことができます。

デフォルト設定では、ACROSS 値のグループ (ACROSS 値とそのデータフィールド) それぞれの幅は、データフィールドの幅の合計、またはそのグループの ACROSS の最大値のいずれかとして定義されます。折り返しを指定する場合、ACROSS の折り返しそれぞれのサイズは、すべてのフィールドおよび折り返しを伴わない ACROSS フィールドを含め、この規則に基づく幅で定義されます。

ACROSS グループ内の各 ACROSS フィールドの幅は、グループで最大の値の長さに定義されます。各グループでは、単一の幅が使用されるため、最大値よりも値が短いグループでは、セル内の右側の間隔が大きくなります。

レポートに複数の ACROSS フィールドが含まれている場合、ACROSS フィールドを個別に折り返すことや、すべての ACROSS フィールドを折り返すことができます。指定された値のそれぞれは、定義済みの ACROSS グループ内で折り返されます。

レポートデータの折り返し制御

レポート内のテキストの折り返しを制御するには、スタイルシートで次の構文を使用します。

```
TYPE=type, [subtype,] WRAP=value, $
```

説明

type

REPORT、HEADING、TITLE など、対象となるレポートコンポーネントです。

subtype

フォーマットを設定するレポートコンポーネントを識別するための任意の追加属性です (例、COLUMN、ACROSS、ITEM)。別のレポートコンポーネントの指定方法についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

value

次のいずれかです。

- **ON** データの折り返しをオンにします。HTML レポート出力のデフォルト値は ON です。PDF、DHTML、PPT、PPTX レポート出力の場合、デフォルト設定で WRAP

は OFF に指定されています。レポート内の各項目の位置が明示的に定義された位置固定の出力フォーマットでは、WRAP = ON は有効値ではありません。ただし、ACROSSVALUE に対して指定された場合は有効です。レポートのその他の要素では(例、見出し、脚注、タイトル、データ)、数値を WRAP = n として指定することで、折り返し行の幅を定義することができます。HTML レポートでは、WRAP はすべてのフィールドでサポートされます。PDF レポートでは、WRAP はテキストフィールドではなく、埋め込みフィールドのみでサポートされます。

注意：この設定は、PDF レポート出力で OVER とともに WRAP を使用する場合はサポートされません。

- **OFF** データの折り返しをオフにします。PDF、DHTML、PPT、PPTX レポート出力の場合、これがデフォルト値です。
- ***n*** 列幅を設定する特定の数値を表します。値は、UNITS パラメータで指定した測定単位で指定されます。この設定は、OVER 句を使用する PDF レポートでデータを折り返す場合にサポートされます。

注意：WRAP=ON および WRAP=*n* は、JUSTIFY とともに使用することはできません。

Web ブラウザデータ折り返し機能の使用

次の例では、WRAP=ON を使用して、Web ブラウザの機能を基にレポートデータを折り返します。この値はデフォルトのため、リクエスト構文で WRAP=ON に指定する必要はありません。

```
TABLE FILE GGPRODS
PRINT SIZE UNIT_PRICE PACKAGE_TYPE
VENDOR_CODE VENDOR_NAME
BY PRODUCT_ID BY PRODUCT_DESCRIPTION
ON TABLE SET STYLE *
TYPE=REPORT, GRID=ON, $
ENDSTYLE
END
```

注意：データの折り返しはブラウザウィンドウのサイズに基づいて決定されるため、下図のようにウィンドウを縮小して、データが折り返された状態を確認する必要があります。

PAGE 1						
Product Code	Product	Size	Unit Price	Package	Vendor ID	Vendor Name
B141	Hazelnut	16	58.00	Pounds	V082	Coffee Connection
B142	French Roast	12	81.00	Pounds	V083	European Specialities,
B144	Kona	12	76.00	Pounds	V081	Evelina Imports, Ltd

Vendor Name フィールドのレコードは、次の行に分割されます。

レポートデータ折り返しの禁止

次のリクエストでは、WRAP=OFF を使用して、Web ブラウザのデータ折り返しを禁止します。

```
TABLE FILE GGPRODS
PRINT SIZE UNIT_PRICE PACKAGE_TYPE
VENDOR_CODE VENDOR_NAME
BY PRODUCT_ID BY PRODUCT_DESCRIPTION
ON TABLE SET STYLE *
TYPE=REPORT, WRAP=OFF, $
TYPE=REPORT, GRID=ON, $
ENDSTYLE
END
```

出力結果は次のとおりです。

PAGE 1						
Product Code	Product	Size	Unit Price	Package	Vendor ID	Vendor Name
B141	Hazelnut	16	58.00	Pounds	V082	Coffee Connection
B142	French Roast	12	81.00	Pounds	V083	European Specialities,
B144	Kona	12	76.00	Pounds	V081	Evelina Imports, Ltd

OVER による列の積み重ね

次のリクエストは、GGPRODS データソースに対して実行され、OVER 句により、VENDOR_NAME 列を新しい行に配置します。折り返しのデフォルト設定はオフで、スタイルシートで明示的に定義する必要があります。

```
TABLE FILE GGPRODS
PRINT SIZE UNIT_PRICE PACKAGE_TYPE OVER
VENDOR_NAME
BY PRODUCT_ID BY PRODUCT_DESCRIPTION
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, SQUEEZE=ON, $
ENDSTYLE
END
```

下図は、出力結果の一部を示しています。

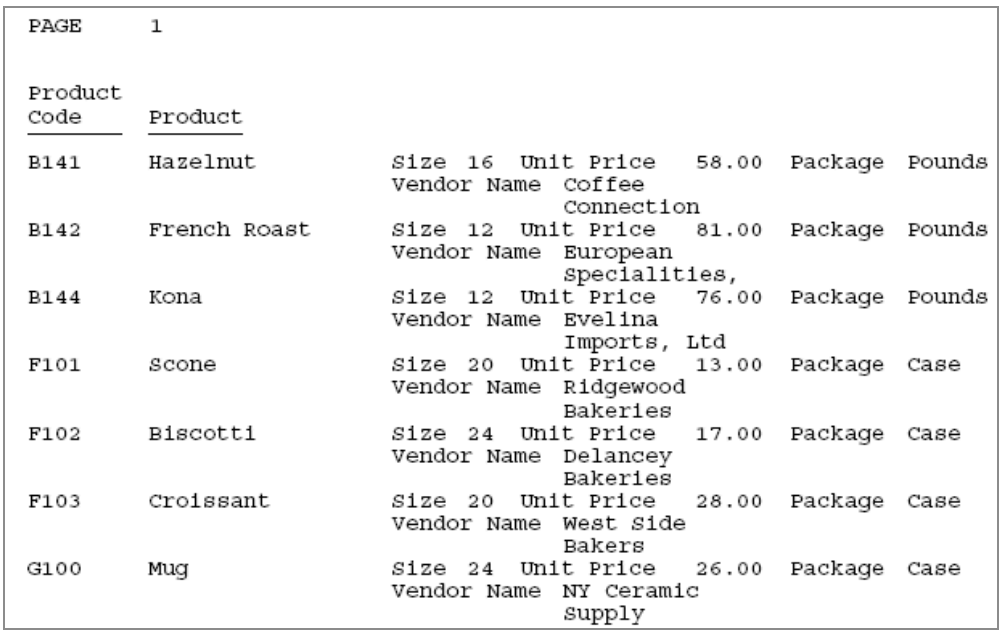
```
PAGE     1


Product
Code     Product
-------  ---------------
B141     Hazelnut          Size  16  Unit Price   58.00  Package  Pounds
                           Vendor Name  Coffee Connection
B142     French Roast      Size  12  Unit Price   81.00  Package  Pounds
                           Vendor Name  European Specialities,
B144     Kona              Size  12  Unit Price   76.00  Package  Pounds
                           Vendor Name  Evelina Imports, Ltd
F101     Scone             Size  20  Unit Price   13.00  Package  Case
                           Vendor Name  Ridgewood Bakeries
F102     Biscotti          Size  24  Unit Price   17.00  Package  Case
                           Vendor Name  Delancey Bakeries
F103     Croissant         Size  20  Unit Price   28.00  Package  Case
                           Vendor Name  West Side Bakers
G100     Mug               Size  24  Unit Price   26.00  Package  Case
                           Vendor Name  NY Ceramic Supply
G104     Thermos           Size  16  Unit Price   96.00  Package  Case
                           Vendor Name  ThermoTech, Inc
G110     Coffee Grinder    Size  12  Unit Price  125.00  Package  Case
                           Vendor Name  Appliance Craft
G121     Coffee Pot        Size   8  Unit Price  140.00  Package  Case
                           Vendor Name  Appliance Craft
```

リクエストを次のように変更し、OVER 句により新しい行に表示した VENDOR_NAME フィールドの折り返しをオンに設定し、さらに列幅を 1.5 に設定します。

```
TABLE FILE GGPRODS
PRINT SIZE UNIT_PRICE PACKAGE_TYPE OVER
VENDOR_NAME
BY PRODUCT_ID BY PRODUCT_DESCRIPTION
```

```
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, COLUMN=VENDOR_NAME, WRAP=1.5,$
ENDSTYLE
END
```

下図は、出力結果の一部を示しています。VENDOR_NAME 列が折り返されています。WRAP ON の設定は、OVER TITLE ではなく OVER の値を折り返します。



```
PAGE     1


Product
Code     Product

B141     Hazelnut          Size 16  Unit Price   58.00  Package  Pounds
                           Vendor Name  Coffee
                                        Connection
B142     French Roast      Size 12  Unit Price   81.00  Package  Pounds
                           Vendor Name  European
                                        Specialities,
B144     Kona              Size 12  Unit Price   76.00  Package  Pounds
                           Vendor Name  Evelina
                                        Imports, Ltd
F101     Scone             Size 20  Unit Price   13.00  Package  Case
                           Vendor Name  Ridgewood
                                        Bakeries
F102     Biscotti          Size 24  Unit Price   17.00  Package  Case
                           Vendor Name  Delancey
                                        Bakeries
F103     Croissant         Size 20  Unit Price   28.00  Package  Case
                           Vendor Name  West Side
                                        Bakers
G100     Mug               Size 24  Unit Price   26.00  Package  Case
                           Vendor Name  NY Ceramic
                                        Supply
```

ACROSS 値の折り返し

HTML および PDF 出力フォーマットでは、ACROSS 値の折り返しがサポートされます。

```
TYPE=ACROSSVALUE, [ACROSS={fieldname|Nn|An}] WRAP={OFF|ON} ,$
```

説明

ACROSS

リクエストで複数の ACROSS フィールドを使用している場合は、ACROSS 識別子を使用して特定のフィールドを選択することができます。リクエストに複数の ACROSS フィールドが使用されている場合に限り、ACROSS 識別子を使用する必要があります。

fieldname

ACROSS フィールドのフィールド名を指定します。

Nn

レポート内の位置に基づいて列を選択します。n の値を特定するには、BY フィールド、表示フィールド、ROW-TOTAL フィールドの数を左から右に数えます。この場合、NOPRINT フィールドもその数に含められます。

An

横方向ソート行の位置に基づいて、ACROSS フィールドを指定します。この値を特定するには、ACROSS フィールドの数を計算します。同一スタイルシート内の fieldname 指定と組み合わせることはできません。

OFF

ACROSS 値の折り返しをオフに設定します。デフォルト値は OFF です。

ON

ACROSS 値の折り返しをオンに設定します。

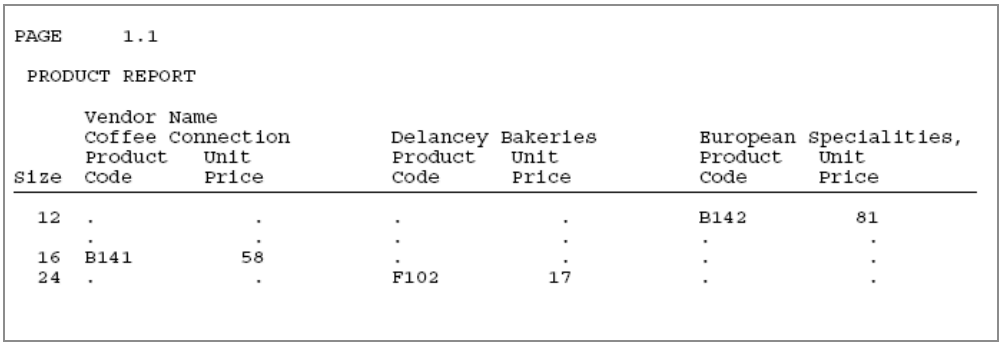
注意：WRAP=ON は、JUSTIFY とともに使用することはできません。

PDF レポートでの ACROSS 値の折り返し

次のリクエストは、GGPRODS データソースに対して実行され、VENDOR_NAME は ACROSS フィールドです。

```
TABLE FILE GGPRODS
HEADING
" PRODUCT REPORT"
" "
PRINT PRODUCT_ID UNIT_PRICE/D5
ACROSS VENDOR_NAME
BY SIZE
WHERE VENDOR_NAME GT 'B' AND VENDOR_NAME LT 'F'
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=REPORT, COLUMN=PRODUCT_ID, WIDTH=.25, $
TYPE=REPORT, COLUMN=UNIT_PRICE, WIDTH=.25, $
ENDSTYLE
END
```

下図のように、ACROSS フィールドの VENDOR_NAME の長さが、その下の PRODUCT_ID フィールドと UNIT_PRICE フィールドの合計サイズを超えるため、出力幅は、1 つのパネルに収まりません。



```
PAGE     1.1

 PRODUCT REPORT

      Vendor Name
      Coffee Connection         Delancey Bakeries         European Specialities,
      Product   Unit            Product   Unit            Product   Unit
Size  Code      Price           Code      Price           Code      Price
------------------------------------------------------------------------
  12  .             .           .             .           B142         81
      .             .           .             .           .             .
  16  B141         58           .             .           .             .
  24  .             .           F102         17           .             .
```

リクエストを次のように変更すると、ACROSS 値の折り返しが有効になります (TYPE=ACROSSVALUE, WRAP=ON ,$)。

```
TABLE FILE GGPRODS
HEADING
" PRODUCT REPORT"
" "
PRINT PRODUCT_ID UNIT_PRICE/D5
ACROSS VENDOR_NAME
BY SIZE
WHERE VENDOR_NAME GT 'B' AND VENDOR_NAME LT 'F'
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, $
TYPE=REPORT, COLUMN=PRODUCT_ID, WIDTH=.25, $
TYPE=REPORT, COLUMN=UNIT_PRICE, WIDTH=.25, $
TYPE = ACROSSVALUE, WRAP=ON,$
ENDSTYLE
END
```

下図のように、このリクエストの出力は、1 つのパネルに収まります。

```
PAGE     1

 PRODUCT REPORT

      Vendor Name
      Coffee            Delancey          European          Evelina
      Connection        Bakeries          Specialities,     Imports, Ltd
      Product    Unit   Product    Unit   Product    Unit   Product    Unit
Size  Code       Price  Code       Price  Code       Price  Code       Price
----------------------------------------------------------------------------

  12  .              .  .              .  B142          81  .              .
      .              .  .              .  .               .  B144          76
  16  B141          58  .              .  .               .  .              .
  24  .              .  F102          17  .               .  .              .
```

ブランクフィールドタイトルの OVER

OVER フィールドをブランクの AS 名とともに (フィールドタイトルを '' として) 定義する場合、これらを使用して、複数のデータ行を表形式で表示するレポートを作成することができます。

このタイプのレポートでは、通常、フィールドタイトルは、列フィールドの左側に表示されるデフォルトタイトルを使用する代わりに、ページ見出しに対応する複数の行を追加することで指定されます。これらの新しい位置揃え機能を活用して OVER フィールドに一意のタイトルを表示するには、個々のフィールドにフィールドタイトルを指定し、これらをリクエスト内でフィールドとして含めます。

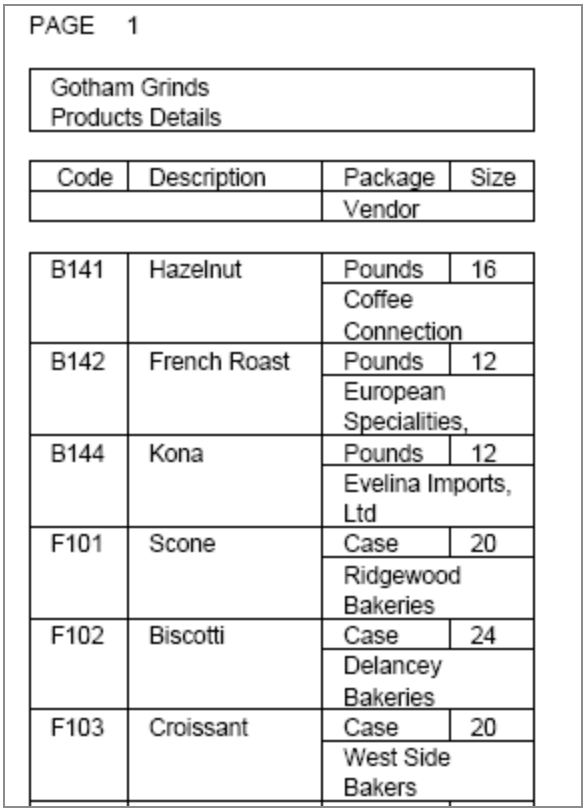
ブランク AS 名の OVER と WRAP

次の例は、ブランクの AS 名および WRAP とともに OVER を使用し、複数データ行のレポートを作成します。

```
TABLE FILE GGPRODS
PRINT PACKAGE_TYPE AS '' SIZE  AS '' OVER
VENDOR_NAME AS ''
BY PRODUCT_ID  AS ''
BY PRODUCT_DESCRIPTION  AS ''
ON TABLE SUBHEAD
"Gotham Grinds"
"Products Details"
HEADING
" Code <+0>Description<+0>Size  <+0>Package"
```

```
-*" <+0> <+0>Vendor"
" <+0>Vendor"
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT=ARIAL, SIZE=10, SQUEEZE=ON,$
TYPE=REPORT, COLUMN=PACKAGE_TYPE, SQUEEZE=.5 ,$
TYPE=REPORT, COLUMN=VENDOR_NAME, WRAP=1 ,$
TYPE=REPORT, BORDER=ON, $
TYPE=HEADING, LINE=1, ITEM=1, BORDER=ON, $
TYPE=HEADING, LINE=1, ITEM=2, BORDER=ON, POSITION=PRODUCT_DESCRIPTION,$
TYPE=HEADING, LINE=1, ITEM=3, BORDER=ON, POSITION=SIZE ,$
TYPE=HEADING, LINE=1, ITEM=4, BORDER=ON, POSITION=PACKAGE_TYPE, $
TYPE=HEADING, LINE=2, ITEM=1, BORDER=ON, $
TYPE=HEADING, LINE=2, ITEM=2, BORDER=ON, POSITION=PACKAGE_TYPE,$
ENDSTYLE
END
```

下のレポートでは、パッケージタイプ (Package) とサイズ (Size) は、ベンダー名 (Vendor) の上部に表示されています。ページ見出しには、対応するタイトルが表示されています。また、見出しには、「Package」、「Size」というタイトルが、ベンダー名の上部に表示されています。ベンダー名は折り返されて、位置揃えが保持されています。



PAGE 1

Gotham Grinds
Products Details

Code	Description	Package	Size
		Vendor	
B141	Hazelnut	Pounds	16
		Coffee Connection	
B142	French Roast	Pounds	12
		European Specialities,	
B144	Kona	Pounds	12
		Evelina Imports, Ltd	
F101	Scone	Case	20
		Ridgewood Bakeries	
F102	Biscotti	Case	24
		Delancey Bakeries	
F103	Croissant	Case	20
		West Side Bakers	

ブランク以外のフィールドタイトルの OVER と WRAP

OVER の値それぞれのフィールドタイトルとフィールドデータの幅は、どちらも単一の SQUEEZE または WRAP で指定されます。タイトルのサイズは自動的に設定され、折り返されるデータフィールドの幅と同一になります。フィールドタイトルの幅がフィールドの折り返し幅として定義されている値よりも広い場合、タイトルの幅を狭くするか、タイトルを OVER フィールドとして追加して個別に指定します。

次の例は、レポートのフィールドタイトルが、WRAP で指定したサイズよりも大きい場合に、OVER と WRAP を使用してレポートを作成する方法を示しています。

フィールドタイトルの OVER と WRAP

次のリクエストでは、2 つの一時項目 (DEFINE) を定義し、製品名フィールドとベンダー名フィールドのタイトルを格納します。さらに、各一時項目を関連するデータフィールドの横に表示し、それぞれにブランクの AS 名を指定します。最初の一時項目 (DEFINE) およびデータフィールドを、2 つ目の一時項目 (DEFINE) とデータフィールドの上部に表示します。

```
DEFINE FILE GGPRODS
TITLE_PROD/A20 = 'Product Description';
TITLE_VEND/A20 = 'Vendor Name';
END
TABLE FILE GGPRODS
PRINT TITLE_PROD AS '' PRODUCT_DESCRIPTION  AS '' OVER
TITLE_VEND AS '' VENDOR_NAME  AS ''
BY PRODUCT_ID  AS ''
ON TABLE SUBHEAD
"Gotham Grinds"
"Products Details"
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, FONT=ARIAL, SIZE=10, SQUEEZE=ON,$
TYPE=REPORT, COLUMN=TITLE_PROD , SQUEEZE=1.25 ,$
TYPE=REPORT, COLUMN=TITLE_VEND , SQUEEZE=1.25 ,$
TYPE=REPORT, COLUMN=PRODUCT_DESCRIPTION, WRAP=.75 ,$
TYPE=REPORT, COLUMN=VENDOR_NAME, WRAP=.75 ,$
TYPE=REPORT, BORDER=ON, $
ENDSTYLE
END
```

出力結果のタイトルとデータは、適切に位置揃えされます。

PAGE 1

Gotham Grinds
Products Details

B141	Product Description	Hazelnut
	Vendor Name	Coffee Connection
B142	Product Description	French Roast
	Vendor Name	European Specialities,
B144	Product Description	Kona
	Vendor Name	Evelina Imports, Ltd
F101	Product Description	Scone
	Vendor Name	Ridgewood Bakeries
F102	Product Description	Biscotti
	Vendor Name	Delancey Bakeries
F103	Product Description	Croissant
	Vendor Name	West Side Bakers
G100	Product Description	Mug
	Vendor Name	NY Ceramic Supply
G104	Product Description	Thermos

折り返し行の行間の制御

スタイルシートの WRAPGAP 属性を使用して、PDF レポート出力のデータ要素の折り返し行の行間を制御することができます。

```
TYPE=DATA, WRAPGAP={ON|OFF|n}
```

説明

ON

折り返し行の行間にブランクを挿入しません。ON は、n の値に 0.0 を指定した場合と同様に動作します。

OFF

折り返し行を次の行に配置します。デフォルト値は OFF です。

n

UNITS 属性で指定された測定単位を使用して、折り返し行の行間に配置するブランクを指定する 0 (ゼロ) 以上の数値です。n の値に 0 (ゼロ) を指定すると、折り返し行の行間にブランクは配置されません。これは、WRAPGAP=ON を指定した場合と同義です。

折り返し行の行間指定

次のリクエストでは、レポートの ADDRESS_LN3 列に対して行の折り返しが有効になっています。

```
TABLE FILE EMPLOYEE
PRINT ADDRESS_LN3
BY LAST_NAME BY FIRST_NAME
WHERE LAST_NAME LE 'CROSS'
  ON TABLE PCHOLD FORMAT PDF
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
type=report, grid=on, $
type=data, topgap=0.2, bottomgap=0.2,  $
type=data, wrapgap=off, $
type=REPORT, column=ADDRESS_LN3, wrap=1.0 ,$
ENDSTYLE
END
```

WRAPGAP=OFF の場合、折り返し行はレポートの次の行に配置されます。

LAST NAME	FIRST NAME	ADDRESS L!
BANNING	JOHN	FREEPORT NY 11520
BLACKWOOD	ROSEMARIE	NEW YORK NY 10001
		EDISON NJ 08817
		BROOKLYN NY 11210
CROSS	BARBARA	NEW YORK NY 10001
		FLUSHING NY 11354

WRAPGAP=ON の場合、折り返し行は 1 つ前の行の直下に配置されます。

LAST_NAME	FIRST_NAME	ADDRESS_L!
BANNING	JOHN	FREEPORT NY 11520
BLACKWOOD	ROSEMARIE	NEW YORK NY 10001
		EDISON NJ 08817
		BROOKLYN NY 11210
CROSS	BARBARA	NEW YORK NY 10001
		FLUSHING NY 11354

WRAPGAP 使用上の注意

WRAPGAP は、行の折り返しが有効になっている列に対してのみ指定することができます (WRAP 属性またはパラメータが ON または特定の数値に設定されている列)。TOPGAP および BOTTOMGAP は、レポートコンポーネントの上部および下部に配置するブランクの高さを指定する属性です。これらの属性の値を大きくすると、折り返し行の間隔がより一層狭くなります。

レポートフィールドの位置揃え

レポートフィールドに左揃え、右揃え、または中央揃えを指定することで、フィールド内のテキストを調整できます。デフォルト設定では、文字フィールドは左揃え、数値フィールドは右揃え、見出しおよび脚注要素は左揃えされます。ただし、JUSTIFY 属性を使用することで、デフォルト設定を変更することができます。/R、/L、/C を使用したフィールドタイトルの位置揃えについての詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。

レポートフィールドの位置揃え

フィールドを左揃え、右揃え、または中央揃えにするには、スタイルシートで次の構文を使用します。

```
TYPE=type, [subtype,] [COLUMN=column,] JUSTIFY=option, $
```

説明

type

REPORT、HEADING、TITLE など、対象となるレポートコンポーネントです。

subtype

フォーマットを設定するレポートコンポーネントを識別するための任意の追加属性です (例、COLUMN、ACROSS、ITEM)。異なるタイプのレポートコンポーネントを指定する方法についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

column

位置揃え対象のフィールドまたはフィールドグループです。この属性は、特定のフィールド (1 つまたは複数) を揃えるときのみ必要です。この属性を省略すると、レポート全体が位置揃えの対象になります。

option

選択する位置揃えです。

- **LEFT** フィールドの左揃えを指定します。
- **RIGHT** フィールドの右揃えを指定します。
- **CENTER** フィールドの中央揃えを指定します。

注意：JUSTIFY は、WRAP=ON または WRAP=*n* を使用する場合はサポートされません。

レポートフィールドデータの位置揃え

次の例は、Vendor Name フィールドのデータを中央に揃えるためのスタイルシート構文を示しています。見出しも中央揃えになります。

```
TABLE FILE GGPRODS
HEADING
"PRODUCT REPORT"
SUM UNITS BY PRODUCT_DESCRIPTION BY PRODUCT_ID BY VENDOR_NAME
ON TABLE SET STYLE *
TYPE=REPORT, COLUMN=VENDOR_NAME, JUSTIFY=CENTER,  $
TYPE=HEADING, JUSTIFY=CENTER, $
TYPE=REPORT, GRID=OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

PRODUCT REPORT

Product	Product Code	Vendor Name	Unit Price
Biscotti	F102	Delancey Bakeries	17.00
Coffee Grinder	G110	Appliance Craft	125.00
Coffee Pot	G121	Appliance Craft	140.00
Croissant	F103	West Side Bakers	28.00
French Roast	B142	European Specialities,	81.00
Hazelnut	B141	Coffee Connection	58.00
Kona	B144	Evelina Imports, Ltd	76.00
Mug	G100	NY Ceramic Supply	26.00
Scone	F101	Ridgewood Bakeries	13.00
Thermos	G104	ThermoTech, Inc	96.00

フィールド基準のフォーマット再設定

フィールドを基準とするフォーマットの再設定は、各行に適用するフォーマットの識別にフィールドを使用することにより、1 つのレポートフィールドの各行に、異なるフォーマットを適用することを可能にします。たとえば、この方法を使用すると、各行が異なる国を表す場合に、それぞれに適切な 10 進数通貨フォーマットを適用することができるようになります。

フォーマット指定を含めることのできるフィールドは次のとおりです。

- データソースの実フィールド。
- DEFINE コマンドで指定した一時項目。

- マスターファイルで指定した一時項目 (DEFINE)。
- COMPUTE コマンド。一時項目を COMPUTE コマンドで作成する場合は、リクエストで、フォーマットの再設定の前に、コマンドを指定する必要があります。

フォーマットを含むフィールドは文字で、長さは少なくとも 8 バイトの必要があります。フォーマット設定には、最初の 8 バイトのみが使用されます。

フィールドを基準とするフォーマットには、元のフィールドより長い値を指定をすることもできます。ただし、新しい長さが元の長さよりも 3 分の 1 以上長くなる場合は、レポートの列幅が値の表示に十分ではなくなることがあります。その場合、フィールドの値はアスタリスク (*) で表示されます。

フィールドを基準とするフォーマットは、あらゆるタイプのフィールドに適用することができます。ただし、新しいフォーマットは、元のフォーマットに一致させておく必要があります。

- 数値フィールドのフォーマットを再設定し、任意の編集フォーマットオプションを持つ別の数値フォーマットにすることができます。
- 文字フィールドのフォーマットを再設定し、異なる長さにすることができます。
- 日付フィールドのフォーマットは、他のタイプの日付フォーマットに再設定することができます。
- 日付時間フィールドのフォーマットは、他の日付時間フォーマットに再設定することができます。

フィールドを基準とするフォーマットが無効、または使用不可能なタイプ変更を指定すると、レポート出力のフィールドは、プラス記号 (++++) とともに表示されます。フォーマットフィールドがブランクまたは欠落している場合、値のフォーマットは再設定されずに表示されます。

フォーマットフィールドの定義および適用

- DEFINE コマンドでは、次のように定義します。

```
DEFINE FILE filename
format_field/A8 = expression;
END
```

- マスターファイルでは、次のように定義します。

```
DEFINE format_field/A8 = expression; $
```

- リクエストでは、次の構文を使用します。

```
COMPUTE format_field/A8 = expression;
```

説明

format_field

行のフォーマットを設定するフィールド名です。

expression

フォーマット値をフォーマットフィールドに割り当てる式です。

フォーマットフィールドの定義後、それをリクエストに適用することができます。

```
TABLE FILE filename
displayfieldname/format_field[/just]
END
```

説明

display

任意の有効な表示コマンドです。

fieldname

フォーマットを再設定するリクエストのフィールドです。

format_field

フォーマットを設定するフィールド名です。フォーマットフィールド名が明示的フォーマットと同一の場合、明示的フォーマットが使用されます。たとえば、「I8」 という名前のフィールドは明示的フォーマット I8 として解釈されるため、フィールドを基準とするフォーマットの再設定には使用できません。

just

位置揃えオプションの L、R または C です。位置揃えオプションはフォーマットフィールドの前または後ろにスラッシュ (/) でフォーマットと区別して配置することができます。

通貨の値の異なる小数位置の表示

```
DEFINE FILE CAR
CFORMAT/A8 = DECODE COUNTRY('ENGLAND' 'D10.1' 'JAPAN' 'D10' ELSE
'D10.2');
END

TABLE FILE CAR
SUM SALES/CFORMAT/C DEALER_COST/CFORMAT
BY COUNTRY
END
```

出力結果は次のとおりです。

```
COUNTRY       SALES    DEALER_COST
-------     ---------  -----------
ENGLAND      12,000.0     37,853.0
FRANCE            .00     4,631.00
ITALY       30,200.00    41,235.00
JAPAN          78,030        5,512
W GERMANY   88,190.00    54,563.00
```

複数行 An および AnV フィールド表示

スタイルシート属性を使用すると、PDF レポートで、複数行に改行文字が含まれる An (文字列) および AnV (可変長文字列) フィールドを表示することができます。改行は、ラインフィード、キャリッジリターン、またはそれらの組み合わせを基準にすることができます。これらのスタイルシート属性を An および AnV フィールドに追加しない場合、フィールド内にあるすべての改行 (キャリッジリターン) フォーマットは無視され、すべての文字はレポートの幅で折り返されて、1 行で表示されます。

複数行に改行を含む An および AnV フィールドの表示

```
TYPE=REPORT,LINEBREAK='type',$
```

説明

REPORT

レポートコンポーネントのタイプです。TYPE は必ず REPORT です。それ以外はエラーとなります。

'type'

以下を基準として、レポート内に挿入される改行を指定します。

LF - すべての An および AnV フィールドの各ラインフィード文字の後に改行を挿入します。

CR - すべての An および AnV フィールドの各キャリッジリターン文字の後に改行を挿入します。

LFCR - すべての An および AnV フィールドのラインフィード文字と、それに続くキャリッジリターン文字の組み合わせそれぞれの後に改行を挿入します。

CRLF - すべての An および AnV フィールドのラインフィード文字と、それに続くキャリッジリターン文字の組み合わせそれぞれの後に改行を挿入します。

注意：この機能は、PDF および PPTX フォーマットでサポートされます。

グラフの作成

データの意味を伝える場合、そのデータを表形式で表現するよりも、グラフを使用した方が分かりやすい場合がよくあります。GRAPH コマンドを使用することにより、ほぼすべてのデータタイプをグラフに簡単に変換することができます。グラフは、必要に応じてカスタマイズすることができます。

グラフを別のリソースにリンクできます。また、条件付きスタイルを使用してグラフの特定のデータを強調することもできます。また、複数のグラフスタイルを使用することもできます。グラフスタイルには、標準グラフ形式 (棒グラフ、折れ線グラフ、円グラフ、散布図、およびこれらのタイプのバリエーション) が含まれます。

さらに、ピアグラフを使用してデータをグラフィカルに表現することができます。詳細は、レポートデータの表示を参照してください。

内容分析 - グラフ化目的の決定

WebFOCUS は、さまざまなレポートツールを備えており、これによって、ユーザに重要な情報を通知するレポートを作成できます。これらのツールの中から特定のニーズに合う最適なものを選択することにより、ユーザに伝達する情報を自由に設計することが可能です。ほぼすべてのデータタイプを効果的に表現できるオプションにはグラフがあります。

グラフを使用することにより、複雑なデータを効果的かつ正確に表し、直感的に把握することができます。グラフは、アイデアを視覚的に表現し、データの重大な変化を印象的な方法でユーザに伝達するという点で、効果的なプレゼンテーションツールです。ユーザは、グラフを見るだけで、作成者が注意を促したい変化に気づき、その変化を追跡することができます。

意味のあるグラフの作成とは、単にデータの見栄えをよくすることではありません。グラフを使用して外観を設計することにより、ユーザはデータに内包されている本質的な情報を把握することができるようになります。

優れたグラフを作成する第一歩として、グラフ化の目的を決定することが必要です。この作業は、次のような手順に分割されます。

1. データを評価する。

   データの中で重要な意味をなすパターンや変化点を特定します。たとえば、データの

最も劇的な変化が、時間の経過によるものか、別の値との関係によるものかを判断します。また、相互に比較する 2 セットのデータがあるかどうかを判断します。

強調する動向または変化を決定します。たとえば、ユーザに印象付けるデータパターンを決定します。

2. 論点およびデータの全体的な形態に最も適したグラフタイプを選択する。ユーザが正しく認識し、全体の関連性を把握できるようなグラフを決定します。
3. グラフの開発を開始する。
4. グラフを調整する。

   ラベルの必要性と利便性を判断します。

   データを意味のある編成にする方法や、グラフに使用するスケールのカスタマイズについても考慮します。

GRAPH コマンド

GRAPH リクエスト構文は、TABLE リクエスト構文に類似しています。表形式ではなく、グラフ形式でレポートを作成するには、リクエスト内の TABLE コマンドを GRAPH に変更します。このように、TABLE リクエストを GRAPH リクエストに変換するだけで、グラフを作成することができます。

ただし、すべての TABLE には対応する GRAPH が存在するわけではなく、グラフで効果的に表すことのできる情報量にも実質上の制限があります。TABLE リクエストがこの方法で変換されると、リクエストの本文を構成するさまざまな句により、グラフのフォーマットおよびレイアウトが決定されます。GRAPH リクエストにより作成されたグラフのタイプは、使用した表示コマンド (SUM または PRINT) およびソート句 (ACROSS または BY) により異なります。

GRAPH と TABLE の類似点

通常、GRAPH リクエスト要素は、対応する TABLE リクエスト要素と同一の規則に準拠します。

- 「FILE」およびファイル名は、事前に SET コマンドで指定されている場合を除いて、GRAPH コマンドの直後に続ける必要があります。

```
SET FILE=filename
```

JOIN 構造やクロスリファレンス構造を含め、WebFOCUS で利用可能な任意のファイルを指定することができます。

- MORE コマンドを使用することにより、GRAPH リクエスト内の異なるタイプのデータソースファイルを連結することができます。詳細は、ユニバーサル連結を参照してください。
- リクエスト内の句の順序は、グラフのフォーマットには影響しません。たとえば、選択句は、表示コマンドおよびソート句の前後いずれにも配置することができます。ソート句の順序は、TABLE リクエスト内のソート句の順序がレポートの表示に影響するように、グラフのフォーマットに影響します。
- リクエストを終了するには、「END」が 1 行に単独で記述されている必要があります。
- すべての日付は、文字フィールドに変更されない限り、MDY 形式で表示されます。

GRAPH と TABLE の相違点

GRAPH および TABLE には、いくつかの構文上の相違点があります。特に、次のような制限が適用されます。

- 意味のあるグラフを生成するには、GRAPH リクエストには、表示フィールドとソート句 (BY または ACROSS) がそれぞれ最低 1 つ含まれている必要があります。
- GRAPH リクエストでは、表示コマンドの目的語は、常に数値フィールドである必要があります。
- GRAPH リクエストには、表示フィールドが 5 つまで許可されています。標準のグラフ形式は、多くの変数を使用すると判読不能になります。
- 1 つのリクエストには BY 句を複数使用することができます。この場合、複数のグラフが作成されます。1 つの GRAPH リクエストで使用できる ACROSS 句は 1 つのみです。特定のグラフ形式のリクエストには、ACROSS 句と BY 句の両方を含めることができます。
- ACROSS 値の数は、64 以下でなければなりません。
- RUN オプションを END の代用として利用することはできません。

TABLE リクエストの GRAPH リクエストへの変換

次の例は、TABLE コマンドを GRAPH コマンドに変更することにより、TABLE リクエストを GRAPH リクエストに容易に変換できることを示しています。

```
TABLE FILE GGORDER
HEADING CENTER
"SAMPLE TABLE"
SUM QUANTITY
BY PRODUCT_DESC AS 'Coffee Types'
WHERE PRODUCT_DESC EQ 'French Roast' OR 'Hazelnut' OR 'Kona'
END
```

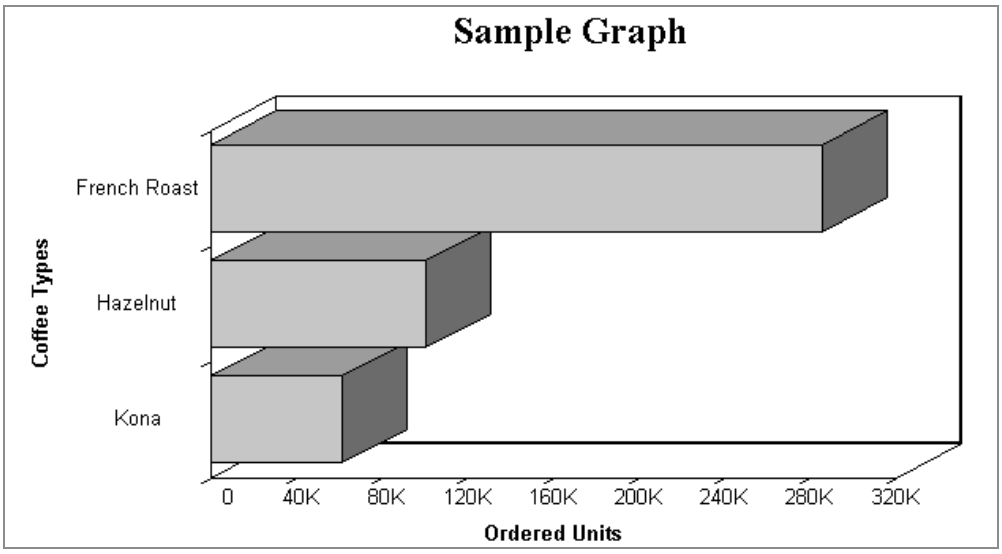
出力結果は次のとおりです。

```
        SAMPLE TABLE
Coffee Types      Ordered Units
------------      -------------
French Roast             285689
Hazelnut                 100427
Kona                      61498
```

以下は、同一リクエストで TABLE コマンドを GRAPH コマンドに変換したものです。

```
GRAPH FILE GGORDER
HEADING CENTER
"Sample Graph"
SUM QUANTITY
BY PRODUCT_DESC AS 'Coffee Types'
WHERE PRODUCT_DESC EQ 'French Roast' OR 'Hazelnut' OR 'Kona'
END
```

出力結果は次のとおりです。



HTML5 グラフの作成

WebFOCUS では、HTML5 標準を活用したグラフ出力フォーマットがサポートされます。このグラフは、ビルトイン JavaScript エンジンを使用して、高画質でインタラクティブなベクタグラフィックスとしてブラウザに表示されます。古いブラウザでは、HTML5 標準の一部の機能がサポートされません。

SET AUTOFIT コマンドを使用して、配置先のコンテナ内に収まるよう HTML5 グラフ出力のサイズを自動的に変更することができます。

HTML5 グラフ出力の作成

グラフリクエストに次のコマンドを追加します。

```
ON GRAPH PCHOLD FORMAT JSCHART
```

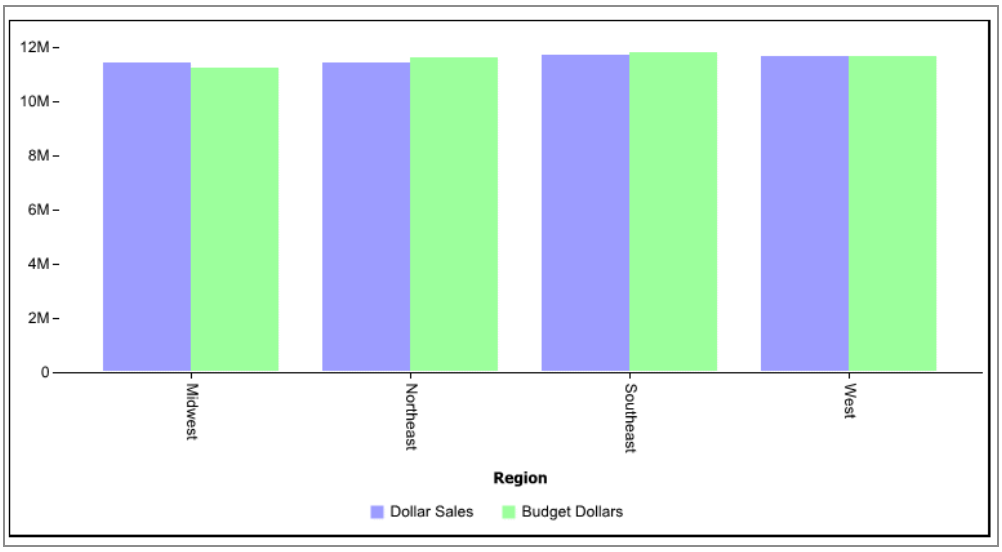
ON GRAPH PCHOLD FORMAT JSCHART が発行されていない場合は、サーバサイドグラフが生成されます。

HTML5 縦棒グラフの作成

次のリクエストは、GGSALES データソースに対して実行され、HTML5 縦棒グラフを作成します。

```
GRAPH FILE GGSALES
SUM DOLLARS BUDDOLLARS
BY REGION
ON GRAPH PCHOLD FORMAT JSCHART
ON GRAPH SET LOOKGRAPH VBAR
END
```

出力結果は次のとおりです。



コンテナに格納する HTML5 グラフ出力のサイズ変更

```
ON GRAPH SET AUTOFIT {OFF|ON|RESIZE}
```

説明

OFF

HAXIS および VAXIS パラメータで指定されたサイズに固定されます。

ON

コンテナ内に収まるよう HTML5 グラフ出力のサイズが常に変更されます。

RESIZE

最初は HAXIS および VAXIS パラメータで指定されたサイズに固定されますが、コンテナのサイズが変更された場合はグラフ出力のサイズも変更されます。

グラフタイプの選択

グラフを作成する際には、表示するデータに適したグラフタイプを選択することが重要です。多数の基本グラフタイプ、およびこれらのタイプのバリエーションから選択することができます。基本グラフタイプには、折れ線グラフ (点をつなげたグラフ)、棒グラフ、円グラフ、散布図があります。グラフタイプの説明を参照して、表示するデータセットおよび強調する変化に適したグラフタイプを選択します。データは表示する数値のセットであり、スケールはグラフの軸に沿って表示される数値または変数の尺度です。

注意：積み上げグラフを使用する場合は、どのタイプであっても、最低 2 組の数値列が必要です。

グラフタイプ

以下の記述は、作成できるグラフのタイプについて説明しています。

- **折れ線**　折れ線グラフは、時間の経過に沿った数値データの動向または傾向を強調する際に役立ちます。これは、過去の状態を観察したり補間したりすることによって、特定の点の動向を追跡できるためです。高低のレベル、動向の速度、安定に向けての傾向などは、すべて折れ線グラフに適した傾向のタイプです。

  また、折れ線グラフで複数のスケールを使用し、同じ値または値のセットの比較を異なる期間で提示することもできます。グラフに表示できるスケールの数は、選択するグラフのタイプによって異なります。詳細は、LOOKGRAPH によるグラフスタイルの決定を参照してください。

- **棒**　棒グラフは、スケールを基準にした長方形のブロックを表示することによって、数値データを表します。棒の長さは、各データの値または数量に相当します。棒の長さ

を対比することにより、データ間の比較結果を明確なイメージとして頭の中に描くことができます。数値データを表示してデータの分布状況を分析する場合は、棒グラフが適しています。棒グラフは、縦横いずれの方向にも作成することができます。

- **円**　円グラフは、特定のデータが全体の中でどの程度の位置を占めるかを強調します。円グラフは、データが複数の大規模なセットから構成される場合に適切です。変数が多すぎると、円が小さいセグメントに分割されるため、判読が困難になります。視覚的なコントラストを付けるには、個々のセグメントに色またはテクスチャを使用します。
- **散布図**　散布図は、基本的な折れ線グラフと共通の特性を多く持っています。縦軸と横軸の 2 変数のスケールを基準にしてデータを表示することができます。散布図を使用すると、データは基本的な折れ線パターンで表されます。散布図は、特定の点の周囲のデータ値の密度を表示、またはデータのパターンを明示する場合に使用します。デフォルト設定では、数値 X 軸またはソートフィールドからは常に散布図が生成されます。

  散布図および折れ線グラフは、X 軸の形式によってのみ識別できることに注意する必要があります。折れ線グラフは、接続線なしで散布図のように表示することも、接続線付きで折れ線グラフのように表示することもできます。
- **面**　面グラフは、折れ線グラフに類似していますが、データの線と 0 (ゼロ) の線 (または軸) の間の領域に通常色またはテクスチャが適用されている点で異なります。面グラフでは、相互のデータの上にデータを積み上げることができます。積み上げにより、あるデータ系列が 2 つ目のデータ系列にどのように近づくか、追い越すかを表示し、データ間の関係を強調することができます。
- **3D**　3D グラフは、グラフによるプレゼンテーションに次元を追加します。次元を使用することにより、複数のデータセットの傾向が認識しやすくなります。また、3D グラフは、プレゼンテーションをよりインパクトのあるものにします。
- **極**　極グラフは、横線に沿って分割されます。このタイプのグラフは、幅広いデータ値の異なる時点や他のソート値での比較傾向分析に役立ちます。
- **レーダー**　レーダーグラフは円形グラフタイプで、部類が周期的な場合に使用されます。スケールが循環していることを除けば、基本的に折れ線グラフに類似しています。レーダーグラフは、1 年の月など、周期的なデータの場合に効果的です。

スケールの選択

グラフタイプを選択した後、適切なスケールを選択する必要があります。スケールは、グラフの軸に適用するために選択する分類基準または一連の尺度です。スケールは、データがプロットされる枠組みを提供します。データ用に適切なスケールを選択すると、意味のあるパ

ターンを明らかにすることができます。スケールを変更すると、グラフの全体的な形状が変化します。

スケール間隔または単位は、グラフの軸に沿ったマークによって表現されます。選択したスケールのタイプによって、スケールに沿った分割の数が決定されます。グラフの Y 軸に適用できる目盛りの一般的なタイプが 2 種類あります。

- 線形スケール
- 対数スケール

線形スケールは、スケールに等差級数を使用します。スケールに沿った各尺度は、先行する尺度より 1 単位高くなっています。線形スケールは、プロットするデータが比較的狭い範囲に収まる場合に役立ちます。

対数スケールは、スケールに対数を使用します。スケールに沿った各尺度は、データ値の指数増加を表現しています。対数スケールは、値が広範囲にわたる場合に役立ちます。

スケールの選択

グラフに対数スケールを使用するには、GRAPH リクエストに次の構文を追加します。

```
ON GRAPH SET STYLE *
*GRAPH_SCRIPT
setY1LogScale(value);
*END
ENDSTYLE
END
```

説明

value

次のいずれかです。

`true` - 対数スケールをオンにします。

`false` - 対数スケールをオフにします。線形スケールが代わりに使用されます。

表示コマンドおよびソート句によるグラフスタイルの決定

各 GRAPH リクエストには、表示フィールドが最低 1 つ (最大 5 つ) とソート句が 1 つ含まれている必要があります。

グラフの主体であるフィールドは、実フィールドまたは一時項目 (DEFINE) です。これらのフィールドでの直接演算接頭語 (AVE.、MIN.、MAX. など) の使用はオプションです。これらはまた、一時項目 (COMPUTE) である場合もあります。

注意：計算のみに使用される表示フィールドはグラフに表示する必要はありません。このようなフィールドを非表示にするには、NOPRINT または SUP-PRINT 句を使用します。詳細は、表形式レポートのソートを参照してください。

デフォルト設定では、表示コマンドとソート句の特別な組み合わせにより、グラフのフォーマットが決定されます。組み合わせは次のとおりです。

グラフのタイプ	表示コマンドおよびソート句
折れ線グラフ	`PRINT A {ACROSS|BY} B` B は文字です。
縦棒グラフ	`SUM A ACROSS B` B は文字です。
横棒グラフ	`SUM A BY B`

円グラフ	`SET LOOKGRAPH=PIE`
	`SUM A {ACROSS|BY} B`
	または
	`SET PIE=ON`

グラフのタイプ	表示コマンドおよびソート句
	`SUM A {ACROSS|BY} B`
散布図 (線を接続しない)	`PRINT A ACROSS B` B は数値です。
散布図 (線を接続する)	`SUM A ACROSS B` B は数値です。

LOOKGRAPH パラメータを使用することにより、デフォルトのグラフフォーマットを上書きすることができます。詳細は、LOOKGRAPH によるグラフスタイルの決定を参照してください。

折れ線グラフの作成

折れ線グラフを作成するには、次の表示コマンドおよびソートフィールドを組み合わせた GRAPH リクエストを発行します。

```
PRINT fieldname1 [AND] fieldname2...
{ACROSS|BY} sortfield
```

説明

fieldname1...

グラフの Y 軸上に表示されるフィールド名です。GRAPH リクエストには、最大 5 つの表示フィールドを設定できます。

AND

読みやすさを向上させるオプション句です。任意の 2 つのフィールド名の間に使用することができます。グラフには影響しません。

sortfield

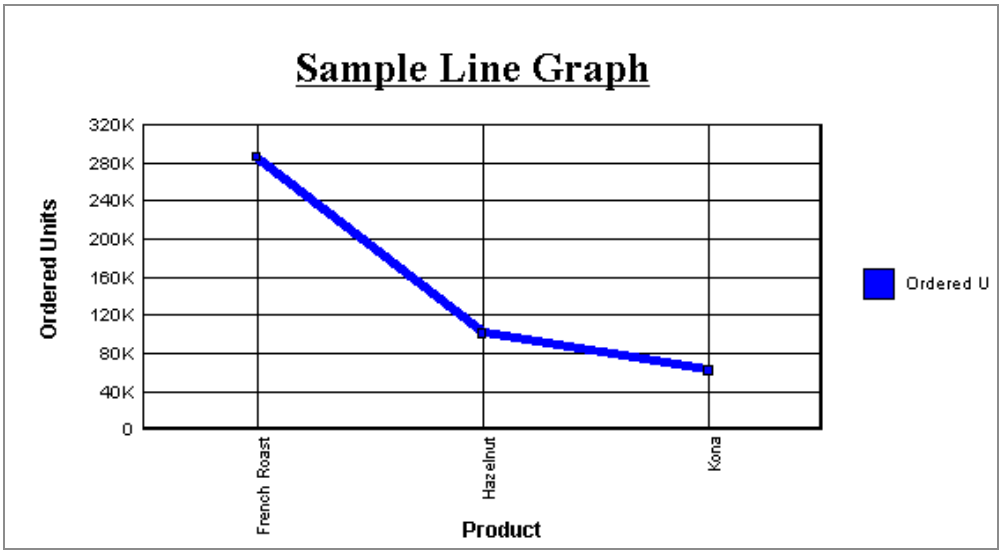
グラフの X 軸上に表示されるフィールド名です。折れ線グラフを生成するには、必ず文字フィールドを指定します。指定したフィールドが数値の場合、LOOKGRAPH=LINE パラメータを使用することで、折れ線グラフを作成することも可能です。詳細は、LOOKGRAPH によるグラフスタイルの決定を参照してください。

折れ線グラフの作成

次の例は、LOOKGRAPH コマンドを使用した折れ線グラフの作成方法を示しています。

```
SET LOOKGRAPH = LINE, GRID=ON
SET HAXIS=600, VAXIS=315
GRAPH FILE GGORDER
HEADING CENTER
"Sample Line Graph"
SUM QUANTITY
ACROSS PRODUCT_DESC
WHERE PRODUCT_DESC EQ 'French Roast' OR 'Hazelnut' OR 'Kona'
END
```

出力結果は次のとおりです。



横棒グラフの作成

横棒グラフを作成するには、次のように表示コマンドとソートフィールドを組み合わせた GRAPH リクエストを発行します。

```
SUM fieldname1 [AND] fieldname2...
BY sortfield
```

説明

fieldname1...

グラフの Y 軸上に表示されるフィールド名です。GRAPH リクエストには、最大 5 つの表示フィールドを設定できます。

AND

読みやすさを向上させるオプション句です。任意の 2 つのフィールド名の間に使用することができます。グラフには影響しません。

sortfield

グラフの X 軸上に表示されるフィールド名です。BY フィールド値ごとに棒グループが作成され、各棒グループには、表示コマンド (SUM) の目的語ごとに棒が含まれます。

縦棒グラフの作成

縦棒グラフを作成するには、次のように表示コマンドとソートフィールドを組み合わせた GRAPH リクエストを発行します。

```
SUM fieldname1 [AND] fieldname2...
ACROSS sortfield
```

説明

fieldname1...

グラフの Y 軸上に表示されるフィールド名です。GRAPH リクエストには、最大 5 つの表示フィールドを設定できます。

AND

読みやすさを向上させるオプション句です。任意の 2 つのフィールド名の間に使用することができます。グラフには影響しません。

sortfield

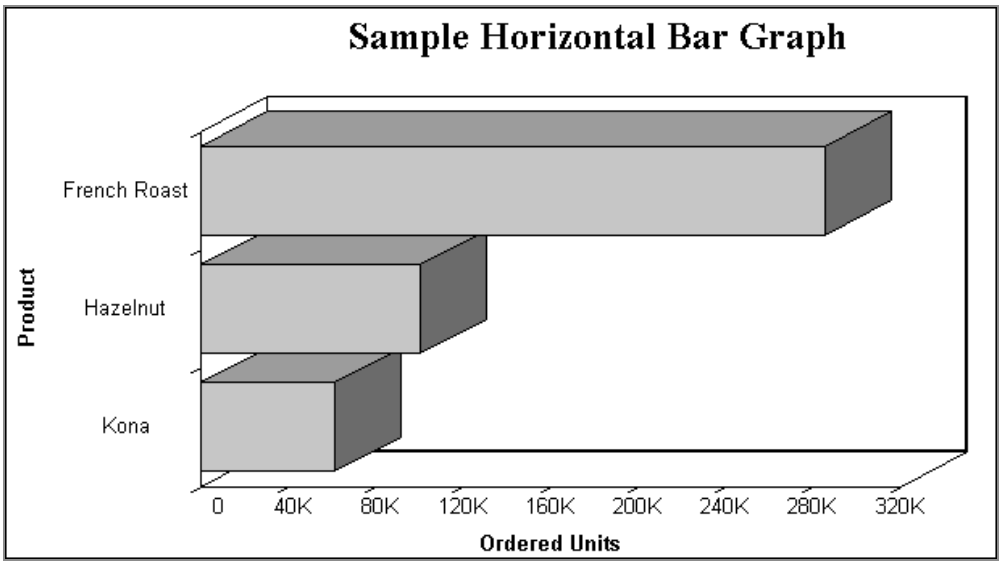
グラフの X 軸上に表示される文字フィールド名です。

横棒グラフの作成

次の例は、横棒グラフの作成方法を示しています。

```
GRAPH FILE GGORDER
HEADING CENTER
"Sample Horizontal Bar Graph"
SUM QUANTITY
BY PRODUCT_DESC
WHERE PRODUCT_DESC EQ 'French Roast' OR 'Hazelnut' OR 'Kona'
END
```

出力結果は次のとおりです。

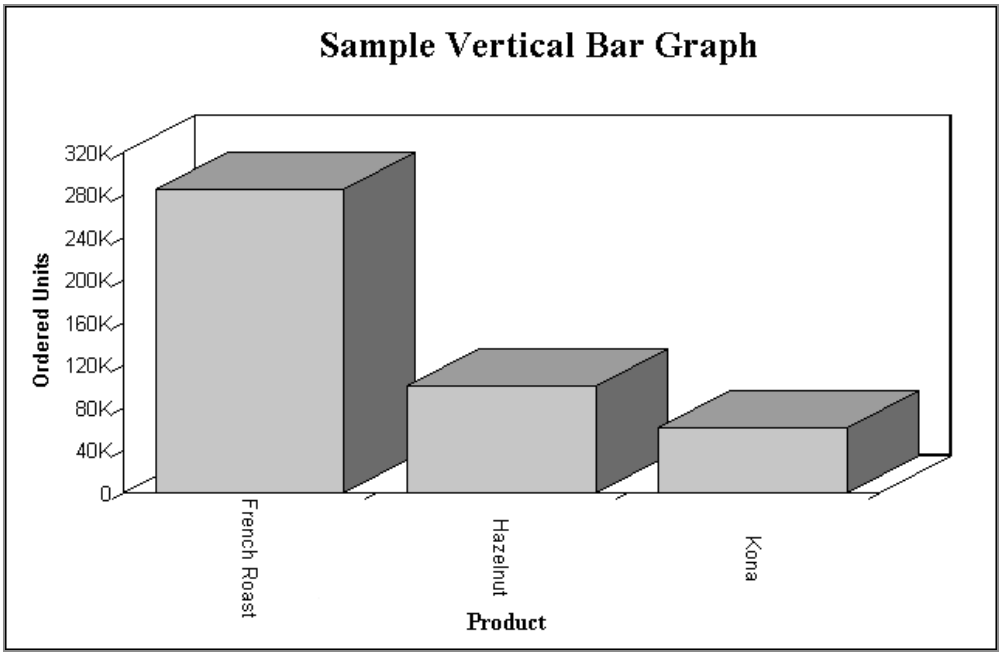


縦棒グラフの作成

次の例は、縦棒グラフの作成方法を示しています。

```
GRAPH FILE GGORDER
HEADING CENTER
"SAMPLE VERTICAL BAR GRAPH"
SUM QUANTITY
ACROSS PRODUCT_DESC
WHERE PRODUCT_DESC EQ 'French Roast' OR 'Hazelnut' OR 'Kona'
END
```

出力結果は次のとおりです。



円グラフの作成

円グラフを作成するには、次のように SET コマンドおよび表示コマンドとソートフィールドの組み合わせを使用した GRAPH リクエストを発行します。

```
SET LOOKGRAPH=PIE
SUM fieldname1 [AND] fieldname2...
{ACROSS|BY} sortfield
```

説明

fieldname1...

グラフに表示するフィールド名です。GRAPH リクエストには、最大 5 つの表示フィールドを設定できます。

AND

読みやすさを向上させるオプション句です。任意の 2 つのフィールド名の間に使用することができます。グラフには影響しません。

sortfield

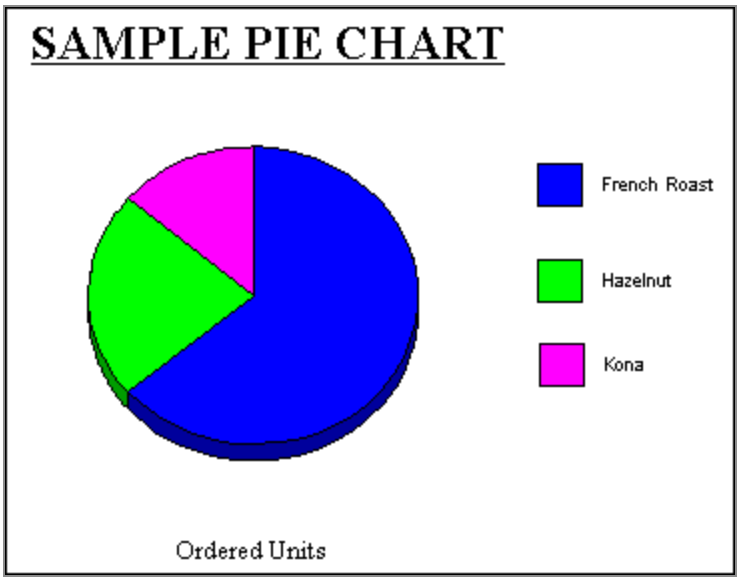
グラフに表示するフィールド名です。ソートフィールドの各値は、円グラフの一区分として表現されます。

円グラフの作成

次の例は、BY ソート句および LOOKGRAPH コマンドを使用した円グラフの作成方法を示しています。

```
SET LOOKGRAPH=PIE
GRAPH FILE GGORDER
HEADING CENTER
"SAMPLE PIE CHART"
SUM QUANTITY
BY PRODUCT_DESC AS COFFEES
WHERE PRODUCT_DESC EQ 'French Roast' OR 'Hazelnut' OR 'Kona'
END
```

出力結果は次のとおりです。



散布図の作成

散布図を作成するには、次のように表示コマンドおよびソートフィールドの組み合わせを使用した GRAPH リクエストを発行します。

```
{PRINT|SUM} fieldname1 [AND] fieldname2...
ACROSS sortfield
```

説明

fieldname

グラフに表示するフィールド名です。GRAPH リクエストには、最大 5 つの表示フィールドを設定できます。複数の表示フィールドを指定する場合、各フィールドは異なる記号で表されます。

AND

読みやすさを向上させるオプション句です。任意の 2 つのフィールド名の間に使用することができます。グラフには影響しません。

sortfield

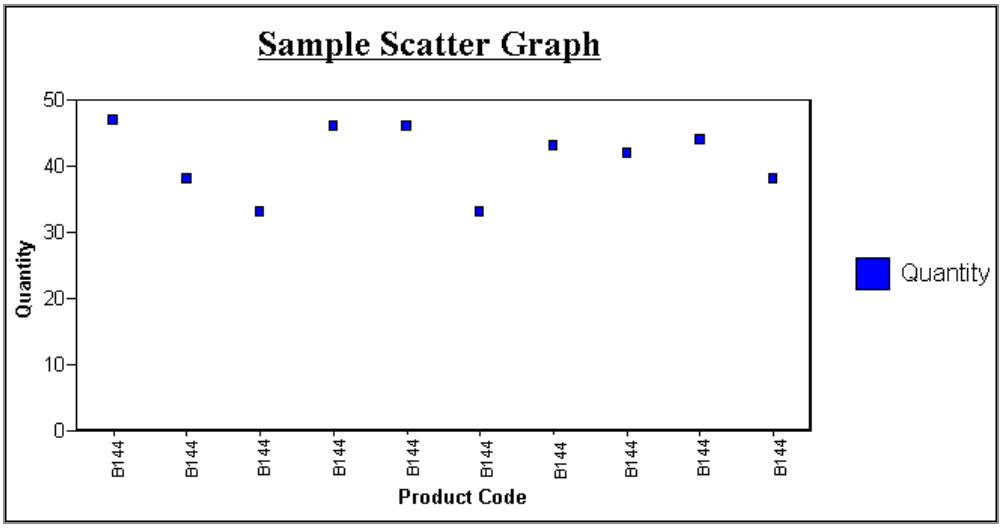
グラフの X 軸上に表示される数値フィールド名です。

散布図の作成

次の例は、散布図の作成方法を示しています。

```
GRAPH FILE GGORDER
HEADING CENTER
"Sample Scatter Graph"
PRINT QUANTITY AS 'Quantity'
ACROSS PRODUCT_CODE
WHERE PRODUCT_CODE EQ 'B144'
WHERE QUANTITY LT 51
END
```

出力結果は次のとおりです。



LOOKGRAPH によるグラフスタイルの決定

デフォルト設定では、表示コマンドとソート句の特別な組み合わせにより、グラフのフォーマットが決定されます。LOOKGRAPH パラメータを使用することにより、デフォルトのグラフフォーマットを上書きすることができます。

また、LOOKGRAPH パラメータにより、個々の制御パラメータの設定や GRAPH リクエストの再構築を行うことなく、グラフフォーマットを変更することができます。ただし、LOOKGRAPH パラメータを使用する場合でも、個々の制御パラメータを設定 (例、SET GRID=ON) することができます。

LOOKGRAPH によるグラフスタイルの指定

```
SET LOOKGRAPH= option
```

説明

option

グラフのスタイルを指定します。グラフスタイルについての詳細は、次の各ページを参照してください。

- 折れ線グラフのスタイルオプション
- 棒グラフのスタイルオプション
- 円グラフのスタイルオプション
- 散布図のスタイルオプション
- 3D グラフのスタイルオプション
- 面グラフのスタイルオプション
- 株価グラフのスタイルオプション
- 極グラフのスタイルオプション
- レーダーグラフのスタイルオプション
- バブルグラフのスタイルオプション
- ブロックグラフのスタイルオプション
- その他のグラフタイプ
- HTML5 固有のグラフタイプのオプション

折れ線グラフのスタイルオプション

折れ線グラフのスタイルを変更するには、次の LOOKGRAPH 値のいずれかを選択します。

SET LOOKGRAPH=	説明
LINE	一般的な 2D 折れ線グラフです。
HLINE	横絶対折れ線グラフです。
HLINE2	横 2 軸絶対折れ線グラフです。
HLINE2S	横 2 極絶対折れ線グラフです。
HLINSTK	横積み上げ折れ線グラフです。
HLINSTK2	横 2 軸積み上げ折れ線グラフです。
HLNSTK2S	横 2 極積み上げ折れ線グラフです。
HLNSTKPC	横パーセント折れ線グラフです。
VLINE	一般的な 2D 折れ線グラフです。
VLINE2	縦 2 軸絶対折れ線グラフです。
VLINE2S	横 2 極絶対折れ線グラフです。

SET LOOKGRAPH=	説明
VLINSTK	縦積み上げ折れ線グラフです。
VLINSTK2	縦 2 軸積み上げ折れ線グラフです。
VLNSTK2S	縦 2 極積み上げ折れ線グラフです。
VLNSTKPC	縦パーセント折れ線グラフです。

棒グラフのスタイルオプション

棒グラフのスタイルを変更するには、次の LOOKGRAPH 値のいずれかを選択します。

SET LOOKGRAPH=	説明
BAR	一般的な平面棒グラフです。
STACK	積み上げ棒グラフです。
VBAR	縦棒グラフです。
VBAR2AX	縦 2 軸棒グラフです。

SET LOOKGRAPH=	説明
VBAR2AXS	縦 2 極棒グラフです。
VBRSTK1	縦積み上げ棒グラフです。
VBRSTK2	縦 2 軸積み上げ棒グラフです。
VBRSTK2S	縦 2 極積み上げ棒グラフです。
VBRSTKPC	縦パーセント棒グラフです。
HBAR	横棒グラフです。
HBAR2AX	横 2 軸棒グラフです。
HBAR2AXS	横 2 極棒グラフです。
HBRSTK1	横積み上げ棒グラフです。
HBRSTK2	横 2 軸積み上げ棒グラフです。
HBRSTK2S	横 2 極積み上げ棒グラフです。

SET LOOKGRAPH=	説明
HBRSTKPC	横パーセント棒グラフです。

円グラフのスタイルオプション

円グラフのスタイルを変更するには、次の LOOKGRAPH 値のいずれかを選択します。

SET LOOKGRAPH=	説明
PIE	一般的な円グラフです。
PIESINGL	円グラフです。
PIEMULTI	マルチ円グラフです。
PIERING	リング型円グラフです。
PIEMULPR	マルチ比例リング型円グラフです。
PIEMULTP	マルチ比例円グラフです。
PIEMULTR	マルチリング型円グラフです。

散布図のスタイルオプション

散布図のスタイルを変更するには、次の LOOKGRAPH 値のいずれかを選択します。

SET LOOKGRAPH=	説明
SCATTER	一般的な散布図です。
SCATTERD	XY2 軸グラフ (2 Y 軸散布図) です。2 つ目のデータセットは、第 2 の値 (Y) 軸上に表示されます。
SCATTRLS	ラベル付き XY 軸散布図です。
SCATTRLD	XY 軸 2 Y 軸ラベル付き散布図です。

3D グラフのスタイルオプション

3D グラフのスタイルを変更するには、次の LOOKGRAPH 値のいずれかを選択します。

SET LOOKGRAPH=	説明
3DAREAG	3D 連続グループ面グラフです。
3DAREAS	3D 連続シリーズ面グラフです。
3DBAR	一般的な 3D タイプ棒を使用した 2D グラフです。

SET LOOKGRAPH=	説明
3D_BAR	一般的な 3D 棒グラフです。
3DCUBE	3D 浮遊キューブグラフです。すべてのデータ点が同一サイズの四角形で、データ値を示す位置に浮遊します。
3DGROUP	一般的な 3D 棒グラフです。
3DOCTAGN	3D 八角形グラフです。
3DPYRAMD	3D ピラミッドグラフです。
3DRIBBNG	連続グループ 3D リボングラフです。
3DRIBBNS	連続シリーズ 3D リボングラフです。
3DSPHERE	浮遊ピラミッド 3D グラフです。すべてのデータ点が同一サイズの球で、データ値を示す位置に浮遊します。
3DSTACK	3D 棒による 2D 積み上げグラフです。
3DSURFCE	3D 表面グラフです。すべてのデータ点がうねる波のような 3D 表面で表示されます。
3DSURFHC	3D 格子状グラフです。すべてのデータ点が蜂の巣のような 3D 表面で表示されます。

SET LOOKGRAPH=	説明
3DSURFSD	側面がある 3D 面グラフです。すべてのデータ点が側面をもつ 3D 表面で表示されます。

面グラフのスタイルオプション

面グラフのスタイルを変更するには、次の LOOKGRAPH 値のいずれかを選択します。

SET LOOKGRAPH=	説明
VAREA	縦絶対値面グラフです。
VAREASTK	縦積み上げ面グラフです。
VAREAR2	縦 2 極絶対値面グラフです。
VARESTK2	縦 2 極積み上げ面グラフです。
VARESTKP	縦パーセント面グラフです。
HAREA	横絶対値面グラフです。
HAREAR2	横 2 極絶対値面グラフです。

SET LOOKGRAPH=	説明
HAREASTK	横積み上げ面グラフです。
HARESTK2	横 2 極積み上げ面グラフです。
HARESTKP	横パーセント面グラフです。

株価グラフのスタイルオプション

株価グラフのスタイルを変更するには、次の LOOKGRAPH 値のいずれかを選択します。

SET LOOKGRAPH=	説明
STOCK	一般的な株価グラフです。
STOCKH	Hi-Lo 株価グラフです。高い数値を表す棒は、低い数値を表す棒の背後に表示されます。高い数値を表す棒の上部のみが表示されるため、値の差異が明示されます。 このグラフタイプは、一般的に株価を表現するために使用されます。各棒は、ある株の 1 日の高値と安値を表します。
STOCKHB	Hi-Lo 2 極株価グラフです。異なるデータセットの値は、それぞれ別の極上に表示されます。

SET LOOKGRAPH=	説明
STOCKHD	2 極 (Y 軸 Hi-Lo) 株価グラフです。2 つ目のデータセットは、第 2 の値 (Y) 軸上に表示されます。
STOCKHCL	Hi-Lo Close 株価グラフです。このグラフタイプは、一般的に株価を表現するために使用されます。各棒は、ある株の 1 日の高値、安値、終値を表します。
STOCKHCB	2 極 Hi-Lo-Close 株価グラフです。異なるデータセットの値は、それぞれ別の極上に表示されます。 このグラフタイプは、一般的に株価を表現するために使用されます。各棒は、ある株の 1 日の高値、安値、終値を表します。
STOCKHCD	2 極 Hi-Lo Close 株価グラフです。2 つ目のデータセットは、第 2 の値 (Y) 軸上に表示されます。
STOCKHOC	Hi-Lo Open-Close 株価グラフです。 このグラフタイプは、一般的に株価を表現するために使用されます。各棒は、ある株の 1 日の高値、安値、始値、終値を表します。
STOCKHOB	2 極 Hi-Lo Open-Close 株価グラフです。異なるデータセットの値は、それぞれ別の極上に表示されます。
STOCKHOD	2極 Hi-Lo Open-Close 株価グラフです。2 つ目のデータセットは、第 2 の値 (Y) 軸上に表示されます。
STOCKHV	出来高表示 Hi-Lo 株価グラフです。
STOCKHOV	出来高表示 Hi-Lo Open-Close 株価グラフです。

SET LOOKGRAPH=	説明
STOCKC	ロウソク型株価グラフです。
STOCKHC	ロウソク型 Hi-Lo 株価グラフです。
STOCKCV	出来高付きロウソク型株価グラフです。
STOCKHCV	出来高表示ロウソク型 Hi-Lo 株価グラフです。

極グラフのスタイルオプション

極グラフのスタイルを変更するには、次の LOOKGRAPH 値のいずれかを選択します。

SET LOOKGRAPH=	説明
POLAR	XY 極グラフ (分散極グラフ) です。
POLAR2	XY 極2軸グラフ (2 Y 極グラフ) です。2 つ目のデータセットは、第 2 の値 (Y) 軸上に表示されます。

レーダーグラフのスタイルオプション

レーダーグラフのスタイルを変更するには、次の LOOKGRAPH 値のいずれかを選択します。

SET LOOKGRAPH=	説明
RADARA	レーダー面グラフ (円形面グラフ) です。
RADARL	レーダー折れ線グラフ (円形折れ線グラフ) です。
RADARL2	レーダー 2 軸グラフです。2 つ目のデータセットは、第 2 の値 (Y) 軸上に表示されます。

バブルグラフのスタイルオプション

バブルグラフのスタイルを変更するには、次の LOOKGRAPH 値のいずれかを選択します。

SET LOOKGRAPH=	説明
BUBBLE	バブルグラフです。
BUBBLED	2 軸バブルグラフです。
BUBBLEDL	2 軸ラベル付きバブルグラフです。
BUBBLEL	ラベル付きバブルグラフです。

ブロックグラフのスタイルオプション

ブロックグラフのスタイルを変更するには、次の LOOKGRAPH 値のいずれかを選択します。

SET LOOKGRAPH=	説明
SPECTRAL	ブロック地図グラフです。行または列マーカーがデータ値によって色分けされて表示されます。

その他のグラフタイプ

SET LOOKGRAPH=	説明
GANTT	プロジェクト指向型で時間制限の厳しいイベントを視覚的に表現します。ガントチャートには、6 つの表示フィールドと 1 つのソートフィールドがこの順序で必要です。条件付きスタイルおよびドリルダウンは、ガントチャートではサポートされません。
POSITION	製品位置付けグラフです。市場占有率や収益成長率などをその指標 (現在、過去、未来) との対比で視覚的に表現します。製品位置付けグラフには、3 つの表示フィールドのセットが必要です。
VWATERFL	縦滝型グラフです。
HWATERFL	横滝型グラフです。
PARETO	パレートの 80:20 規則に従うデータを表示します。パレートグラフは、1 つの表示フィールドのみが必要で

SET LOOKGRAPH=	説明
	す。
MULTI3Y MULTI4Y MULTI5Y	グラフの判読、分析、管理を容易にするために複数のグラフを積み重ねます。

HTML5 固有のグラフタイプのオプション

次の LOOKGRAPH 値は、HTML5 グラフを生成する場合にのみ有効です。

SET LOOKGRAPH	説明
BUBBLEMAP	バブルマップは、値に比例したサイズのバブルが地図上の関連する領域に表示されるグラフです。
CHOROPLETH	コロプレスマップは、表示するメジャーの値に応じて地図上の領域が濃淡やパターンで区分されたグラフです。
MEKKO	マリメッコチャートは、積み上げ棒グラフの一種で、棒の幅がデータセットの値に比例して調整されます。
PARABOX	パラボックス (並行座標プロット) は、標準の折れ線グラフに類似していますが、折れ線グラフの各グループにそれぞれ固有のインタラクティブな数値軸が使用される点が異なります。各折れ線は、データのシリーズを表します。各縦棒は、数値軸を表します。それぞれの軸上をクリックし、ドラッグして値の範囲を指定することで、軸のその部分を通過する線を選択 (フィルタ) することができます。

SET LOOKGRAPH	説明
STREAM	ストリームグラフは、積み上げ面グラフを簡素化したグラフです。ストリームグラフでは、軸および罫線が省略されます。ベースラインが固定されていないため、データ全体で特定のレイヤの厚みを簡単に把握することができます。
TAGCLOUD	タグクラウドは、発生頻度をグラフ化したものです。このグラフには、グループのラベルのみが表示されます。データ値に比例して、各ラベルのサイズが変化します。
TREEMAP	ツリーマップには、階層データが、ネストされた長方形の集合体として表示されます。

X 軸値および Y 軸値の選択

X 軸および Y 軸用に選択した値によって、作成するグラフに含まれるデータの内容、およびその表示形式が決定されます。

X 軸の値は、GRAPH リクエスト内で使用したソート句 (BY または ACROSS) により決定されます。GRAPH リクエストには、それぞれ最低 1 つのソート句が必要です。複数の BY 句がある場合、または ACROSS および BY 句が同一リクエスト内に含まれる場合、リクエストで参照されたフィールドの値の組み合わせに対して 1 つずつグラフが作成されます。詳細は、複数グラフの作成を参照してください。

Y 軸の値は、GRAPH リクエスト内で発行した表示コマンド (SUM または PRINT) に関連した表示フィールドで決定されます。

次のことも可能です。

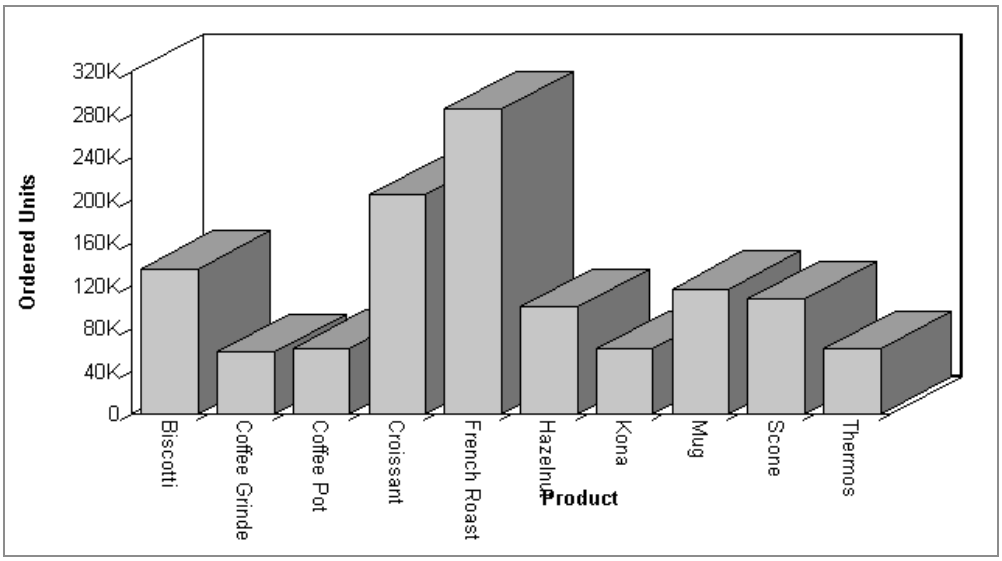
- 2 つ目の横方向カテゴリ (X 軸) を選択します。これにより、複数のグラフが作成されます。詳細は、複数グラフの作成を参照してください。
- Y 軸フィールドの表示を一時的に非表示にする。詳細は、Y 軸 フィールドの非表示を参照してください。
- 線形回帰を使用して X 軸と Y 軸の値を補間します。詳細は、線形回帰による X 軸値および Y 軸値の補間を参照してください。

X 軸値および Y 軸値の選択

次の例は、ACROSS 句を使用した X 軸 (PRODUCT_DESC) の設定方法、および表示コマンド SUM を使用した Y 軸 (QUANTITY) の設定方法を示しています。

```
GRAPH FILE GGORDER
SUM QUANTITY AS 'Ordered Units'
ACROSS PRODUCT_DESC
END
```

出力結果は次のとおりです。



Y 軸 フィールドの非表示

グラフの Y 軸フィールドを非表示にすることができます。このオプションは、元のグラフプロパティをすべて保持したまま、特定の Y 軸フィールドを一時的に除外する場合に役立ちます。

Y 軸フィールドを一時的に非表示にするには、フィールドに NOPRINT コマンドを追加します。NOPRINT コマンドは、TABLE リクエストでは動詞オブジェクトとソートフィールドの両方に適用されますが、GRAPH リクエストでは動詞オブジェクトにのみに適用されます。

線形回帰による X 軸値および Y 軸値の補間

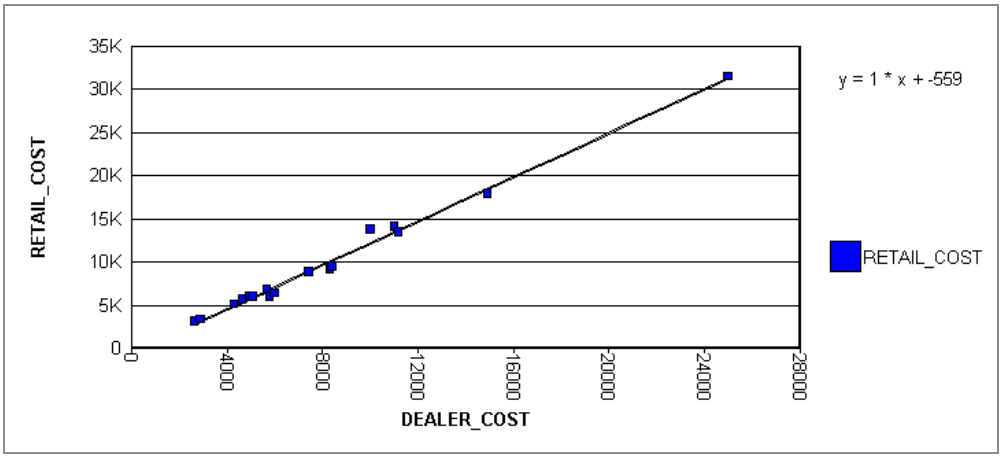
線形回帰を使用して X 軸と Y 軸の値を補間します。基本線形回帰は、X 軸および Y 軸の値の総和の平均を使用し、散布図の傾向を表す一次方程式を求めます。SET パラメータ GTREND を使用して、グラフ内の基本線形回帰をオンに設定します。

GTREND は、散布図でのみ利用可能です。

次の例は、散布図内での線形回帰の設定方法を示しています。

```
SET 3D=OFF
GRAPH FILE CAR
PRINT RC
ACROSS DC
ON GRAPH SET LOOKGRAPH SCATTER
ON GRAPH SET GTREND ON
END
```

出力結果は次のとおりです。



複数グラフの作成

2 つ目のソートディメンション (フィールド) を追加することで、複数のグラフを作成することができます。

デフォルト設定では、作成されるグラフ数は、ソート句 (BY、ACROSS) で指定したフィールドの値の個数によって決定されます。このデフォルト設定を変更するには、GRMERGE パラメータを使用します。

- GRMERGE OFF (デフォルト設定) では、リクエストに BY フィールドが 2 つ含まれる場合、最初の BY フィールドの値の個数と同数のグラフが生成されます。2 つ目の BY フィールドは、X 軸を決定します。たとえば、2 つの値を含む BY フィールドを選択した場合、2 つのグラフが生成されます。10 個の値を含むフィールドを選択した場合、10 個のグラフが生成されます。BY 句が 1 つで、ACROSS 句が 1 つの場合、BY フィールド内の値と同数のグラフが表示されます。ACROSS フィールドは、X 軸を決定します。2 つ目の横方向カテゴリは、同一リクエスト内に複数の BY 句または ACROSS 句と BY 句を 1 つずつ含めることにより選択することができます。
- GRMERGE ON では、統合された 1 つのグラフが作成されます。
- GRMERGE ADVANCED では、3 つのパラメータに基づいて次のことが決定されます。
  - 生成するグラフ数 (GRMULTIGRAPH)。
  - グラフ凡例に配置するソートフィールド数 (GRLEGEND)。
  - X 軸に配置するソートフィールド数 (GRXAXIS)。

複数のグラフを、統合された形式または列形式で表示することができます。詳細は、複数グラフの統合および 複数グラフの列表示を参照してください。

複数グラフの統合

デフォルト設定では、複数の BY フィールドが含まれるグラフ、または BY フィールドと ACROSS フィールドが含まれるグラフを作成すると、複数のグラフが生成されます。これらのグラフを統合して、1 つのグラフに統合することも、複数のグラフにすることもできます。

この作業には、SET コマンドの GRMERGE を使用します。

グラフの統合

```
SET GRMERGE={ON|OFF|ADVANCED}
```

説明

ON

グラフの統合オプションをオンにします。

OFF

グラフの統合オプションをオフにします。これがデフォルト値です。

ADVANCED

高度な統合オプションを有効にします。このオプションを選択すると、次の 3 つのパラメータに基づいて複数グラフの統合方法が決定されます。

- GRMULTIGRAPH - 複数グラフの作成に使用するソートフィールド数を指定します。
- GRLEGEND - グラフ凡例に配置するソートフィールド数を指定します。
- GRXAXIS - X 軸に表示するソートフィールド数を指定します。グラフを描画するには、GRXAXIS の値を少なくとも 1 に設定する必要があります。1 より大きい値を指定すると、ネストされた X 軸が作成されます。

注意：GRMULTIGRAPH、GRLEGEND、GRXAXIS で使用するソートフィールドの総数は、グラフリクエストで使用するソートフィールド数に一致させる必要があります。

GRMULTIGRAPH、GRLEGEND、GRXAXIS パラメータの構文は次のとおりです。

ON GRAPH SET GRMULTIGRAPH n

出力を複数のグラフに分割する際に使用するソートフィールド数 (0 から 2) を指定します。グラフの分割には、最も外側のソートフィールドが使用されます。n が 0 (ゼロ) より大きい場合、GRMERGE=OFF と同様に機能しますが、この設定ではソートフィールドを追加することが可能です。

ON GRAPH SET GRLEGEND n

GRMULTIGRAPH に使用する最も外側のソートフィールドを除いた残りのソートフィールドのうち、グラフ凡例に追加するソートフィールド数 (0 から 2) を指定します。n が 0 (ゼロ) より大きい場合、GRMERGE=ON と同様に機能しますが、この設定ではソートフィールドを追加することが可能です。

ON GRAPH SET GRXAXIS n

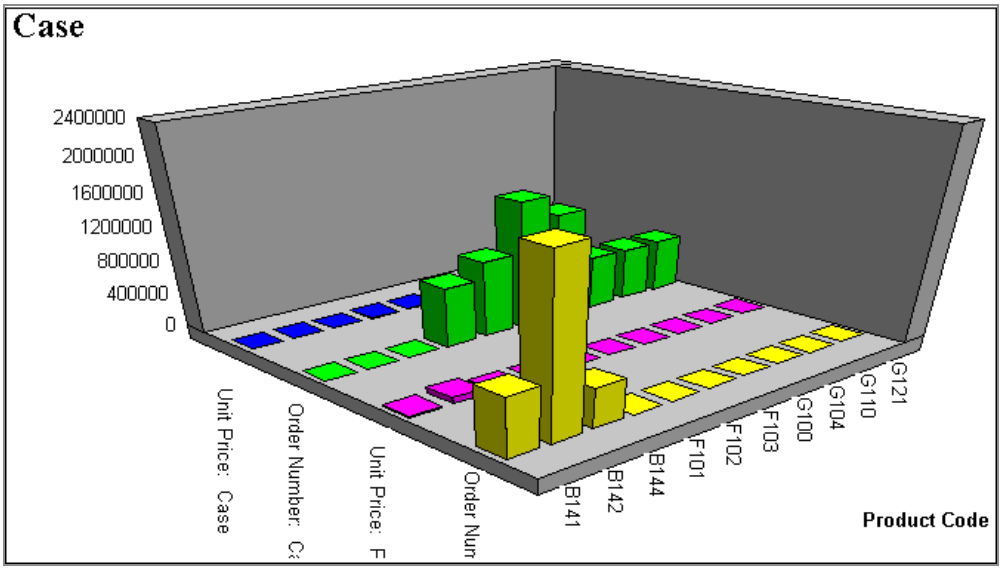
X 軸に表示する残りのソートフィールド数 (1 から 3) を指定します。n が 1 より大きい場合、ネストされた X 軸が作成されます。

GRMERGE ON による複数グラフの統合

次の例は、2 つの横方向 (X 軸) カテゴリ (PRODUCT_ID および PACKAGE_TYPE) が統合されたグラフを示しています。

```
SET GRMERGE=ON
GRAPH FILE GGORDER
SUM UNIT_PRICE ORDER_NUMBER
ACROSS PRODUCT_ID
BY PACKAGE_TYPE
END
```

出力結果は次のとおりです。



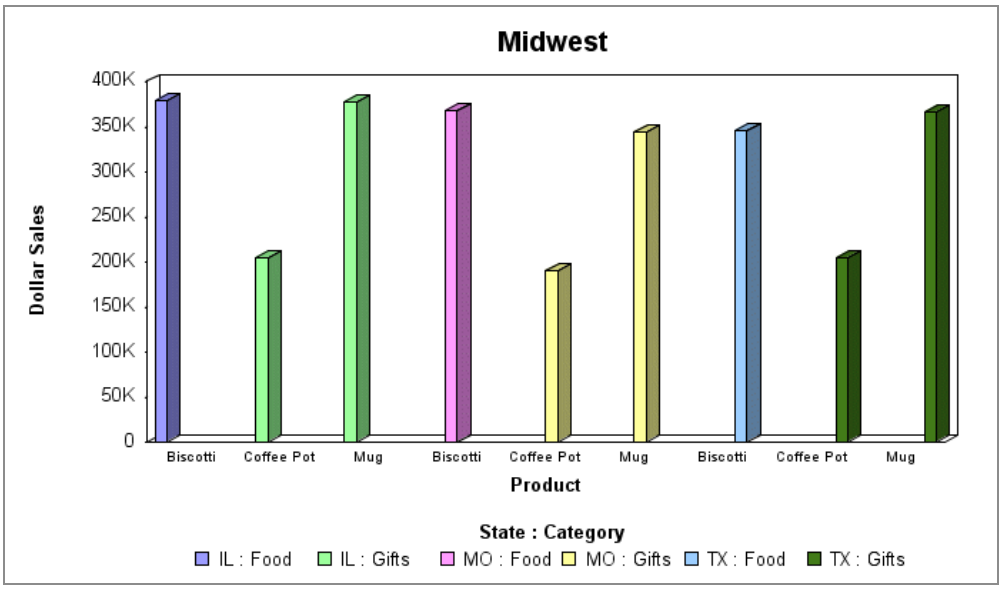
GRMERGE ADVANCED による複数グラフの統合

次のリクエストは縦棒グラフを生成し、最も外側のソートフィールド (REGION) の値ごとに、それぞれ個別のグラフを作成します。また、グラフ凡例に 2 つのソートフィールド (ST および CATEGORY) を統合することでこれらを識別し、X 軸上に CATEGORY ソートフィールドを配置します。

```
GRAPH FILE GGSALES
SUM DOLLARS
BY REGION BY ST BY CATEGORY BY PRODUCT
WHERE CATEGORY EQ 'Food' OR 'Gifts'
WHERE PRODUCT EQ 'Coffee Pot' OR 'Biscotti' OR 'Mug'
ON GRAPH SET GRMERGE ADVANCED
ON GRAPH SET GRMULTIGRAPH 1
```

```
ON GRAPH SET GRLEGEND 2
ON GRAPH SET GRXAXIS 1
ON GRAPH SET LOOKGRAPH VBAR
END
```

下図は、Midwest 地域のグラフを示しています。凡例では、州とカテゴリの組み合わせが色で区別されています。また、X 軸上では、州とカテゴリの組み合わせごとに PRODUCT ソートフィールドの値が繰り返されています。



複数 OLAP グラフの統合

複数の BY フィールド、または BY フィールドと ACROSS フィールドが含まれる OLAP グラフを作成すると、複数のグラフが生成されます。これらのグラフは、1 つのグラフに統合することができます。

これを行うには、SET コマンドの OLAPGRMERGE を使用します。

複数 OLAP グラフの統合

```
SET OLAPGRMERGE={ON|OFF}
```

説明

ON

グラフの統合オプションをオンにします。この設定では、グラフの AUTODRILL が無効になります。

OFF

グラフの統合オプションをオフにします。外部ソートフィールドの値ごとに、それぞれ別のグラフが作成されます。デフォルト値は OFF です。

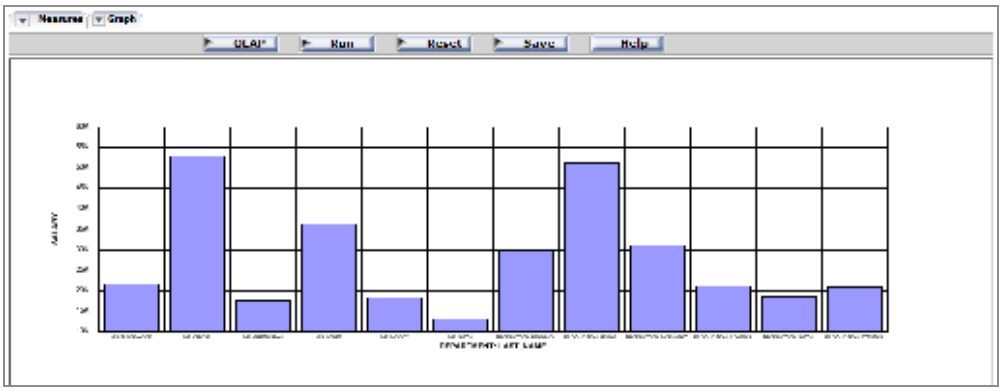
OLAP 対応グラフの統合

次の OLAP リクエストは EMPLOYEE データソースに対して実行され、2 つの BY フィールドが含まれています。グラフを統合するために、SET OLAPGRMERGE=ON コマンドが発行されています。

```
-OLAP ON
SET GRAPHEDIT=SERVER
SET OLAPGRMERGE=ON
TABLE FILE EMPLOYEE
SUM SALARY
BY DEP
BY LAST_NAME
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
ON GRAPH SET HAXIS 300
ON GRAPH SET VAXIS 100
ON TABLE SET AUTODRILL ALL
ON TABLE SET OLAPPANE TABBED
ON TABLE SET STYLE *
     INCLUDE = endeflt,
$
     LEFTMARGIN=0.500000,
     RIGHTMARGIN=0.500000,
     TOPMARGIN=0.500000,
     BOTTOMMARGIN=0.500000,
$
TYPE=REPORT,
     TOPGAP=0.000000,
     BOTTOMGAP=0.013889,
$
```

```
ENDSTYLE
END
```

出力結果は次のとおりです。



複数グラフの列表示

複数の BY フィールド、または BY フィールドと ACROSS フィールドでグラフを 1 つ作成すると、複数のグラフが生成されます。これらのグラフを複数の列に表示することができます。

これを行うには、SET コマンドの GRWIDTH を使用します。GRWIDTH には、0 (ゼロ) から 512 の任意の値を指定します。デフォルト値は 0 です。

複数グラフの列表示

```
SET GRWIDTH=nn
```

説明

nn

複数のグラフを表示する列数です。0 (ゼロ) から 512 までの任意の値を指定します。デフォルト値は 0 です。

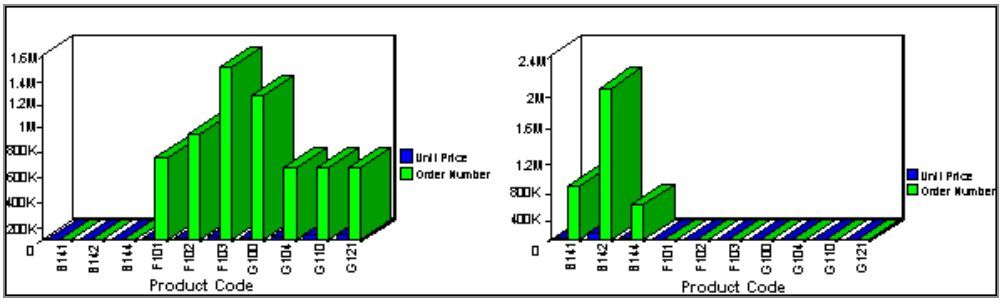
1 から 512 の値を指定すると、グラフは HTML テーブル内に表示されます。この HTML テーブルの列数は、指定した値に対応します。デフォルト値の 0 (ゼロ) では、グラフは Java アプレット内で上下に表示されます。

複数グラフの列表示

次の例は、複数のグラフを表示する場合の列数の設定方法を示しています。この例では、グラフが 2 列で表示されるように設定します。

```
SET GRWIDTH=2
GRAPH FILE GGORDER
SUM UNIT_PRICE ORDER_NUMBER
ACROSS PRODUCT_ID
BY PACKAGE_TYPE
END
```

出力結果は次のとおりです。



グラフの日付プロット

日付を含む数値フィールドは、マスターファイルに指定されたフィールドのフォーマットにより認識されます。このようなフィールドは、GRAPH リクエスト内の ACROSS 句または BY 句で使用することができます。

日付のプロットは、次のように処理されます。

- 指定した日付フィールドが月で開始する場合は、昇順にプロットされます。ファイルの日付を昇順に事前にソートする必要はありません。月と年を表す日付である 01/76、03/76、09/75 は、09/75、01/76、03/76 のようにソートされてプロットされます。
- 軸の目盛りは、1 か月あたりの日数と 1 年あたりの月数を基に指定されます。日付

フォーマットに日が含まれる場合、スケールは通常、軸目盛り 0 (ゼロ) を月の初日に開始します。

日付プロットグループを選択的に組み合わせることにより、横軸上の分割点数を減少させることができます。この作業には、IN-GROUPS-OF オプションを使用します。たとえば、日付フィールドのフォーマットが I6YMD の場合、データの増分値を 30 日に設定することにより、データを日単位ではなく、月単位で表示することができます。

```
ACROSS DATE IN-GROUPS-OF 30
```

これにより、日単位でのプロットが削除されます。日付フィールドのフォーマットが YMD の場合、フィールドの内容を 100 で除算するようにフォーマットを定義することにより、日単位の表示を削除することができます。

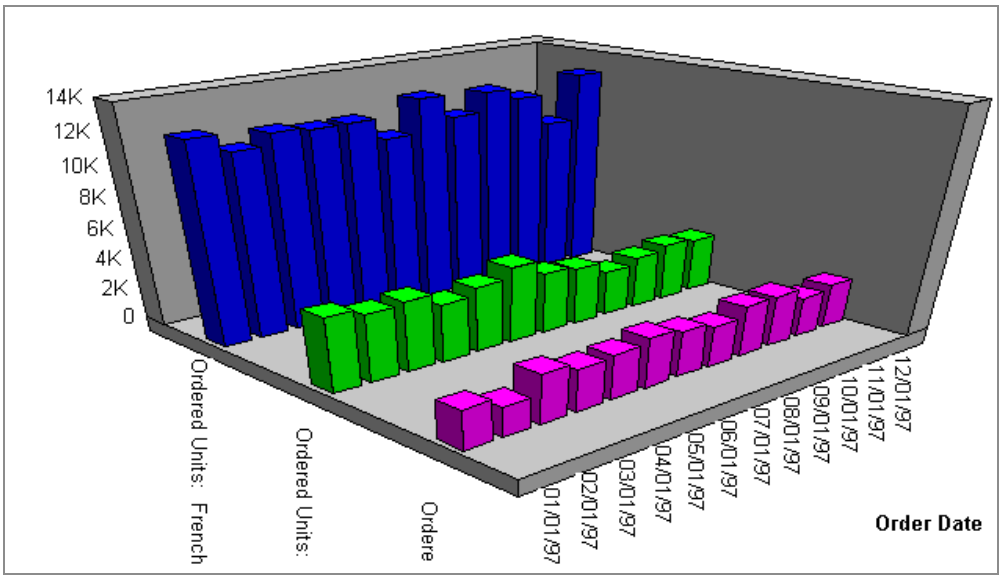
```
DATE/I4YM=DATE/100
```

グラフでの日付フィールドの表示

次の例は、月で開始するフォーマットの日付フィールドのグラフを示します。

```
SET GRMERGE = ON
GRAPH FILE GGORDER
SUM QUANTITY
ACROSS ORDER_DATE
BY PRODUCT_DESC
WHERE PRODUCT_DESC EQ 'French Roast' OR 'Hazelnut' OR 'Kona' AND
ORDER_DATE GE '010197'
END
```

出力結果は次のとおりです。



X 軸および Y 軸の基本日付サポート

OLDDATES は、YRTHRESH および DEFCENT SET パラメータとともに使用します。Y フォーマット (例、YMD、MDY、DMY、YM など) の日付に YRTHRESH および DEFCENT コマンドを指定しない場合、OLDDATES は 19 *XX* のフォーマットと見なされます。

日付サポートの制限事項

- 次の日付フォーマットが利用可能です。
  - SHORT (18) は、12/13/52 や 3:30pm などの完全数値です。
  - MEDIUM (19) では、Jan 12, 1952 などのように長くなります。
  - LONG (20) では、January 12, 1952 や 3:30:32pm などのように、さらに長くなります。
  - FULL (21) は、Tuesday, April 12, 1952 AD や 3:30:42pm PST などのように、ほぼ完全な形で表されます。

  X 軸および Y 軸のデフォルト日付フォーマットは MEDIUM です。デフォルトを上書きするには、次の API コールのいずれかを使用します。*xx* は、前述した数値の 1 つで

す。

```
setTextFormatPreset(getX1Label(),xx); // for X Axis
```

```
setTextFormatPreset(getY1Label(),xx); // for Y Axis
```

データテキストのデフォルト日付フォーマットは LONG です。これは Y 軸が日付であるグラフにのみ該当します。現在、このフォーマットは X 軸上ではサポートされません。デフォルトを上書きするには、次の API コールを使用します。

```
setDataTextFormat(xx);
```

詳細は、グラフへのカスタムスタイルの適用を参照してください。

- X 軸が日付で、Y 軸が数値フィールドのグラフでは、デフォルト設定で、ツールヒントにはグラフのデータフォーマットが表示されます。つまり、日付値には、その日付の GMT 値がミリ秒単位で表示されます。Y 軸が日付で、X 軸が文字列のグラフでは、データフォーマットがすでに日付フォーマットになっているため、この問題は発生しません。
- DATETIME は、完全にはサポートされていません。
- 特定の軸に開始日と終了日を設定する機能はサポートされていません。
- 日付に間隔を設定する機能はサポートされていません。

Y 軸値の日付フォーマット指定

日付フォーマットの指定された数値を Y 軸フィールドおよびツールヒントに表示することができます。次の日付フォーマットが利用可能です。

表示フォーマット	対応する WebFOCUS フォーマット
yy/mm/dd	YMD
yy/mm	YM
mm/dd/yyyy	MDYY

表示フォーマット	対応する WebFOCUS フォーマット
mm/dd/yy	MDY
mm/dd	MD

グラフデータセットの調整

X 軸および Y 軸のフィールド値を選択した後、グラフに表示するデータを制限することができます。これには、WHERE ステートメントを作成します。WHERE ステートメントを使用して、データセットに含めるデータの条件を定義するパラメータを作成することで、データを限定できます。

WHERE、WHERE TOTAL、および IF 句についての詳細は、レコードの選択を参照してください。

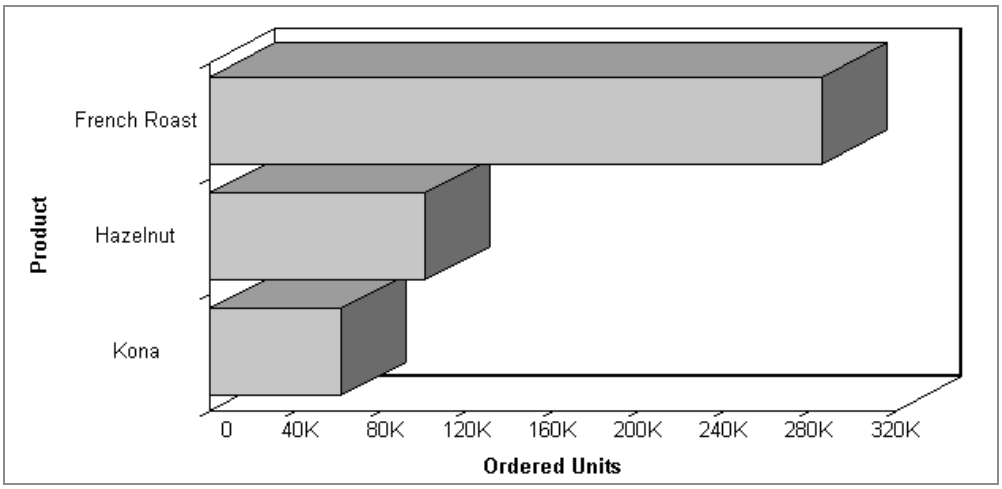
GRAPH リクエストでの WHERE 条件の指定

GRAPH リクエスト内の WHERE、WHERE TOTAL、IF 句の構文は、TABLE リクエストで使用される構文と同一です。

次の GRAPH リクエストは、特定の製品の説明 (French Roast、Hazelnut、Kona) のデータを表示します。

```
GRAPH FILE GGORDER
SUM QUANTITY
BY PRODUCT_DESC
WHERE PRODUCT_DESC EQ 'French Roast' OR 'Hazelnut' OR 'Kona'
END
```

出力結果は次のとおりです。



ミッシングデータ値のグラフ表示

次のフォーマットのいずれかを使用して、ミッシングデータ値を表示することができます。この場合、棒グラフ、折れ線グラフ、面グラフ、およびこれらのグラフタイプに属するあらゆる種類が対象になります。

- **ゼログラフ**　棒グラフでは、棒が 0 (ゼロ) の線上に表示されます。折れ線グラフでは、ミッシング値とそれに続く値が実線で接続されます。面グラフでは、領域が 0 (ゼロ) の線上に表示されます。
- **ギャップグラフ**　棒グラフ、折れ線グラフ、面グラフのすべてのグラフタイプで、ミッシング値はギャップとして表示されます。
- **ゼロへの点線**　折れ線グラフでは、ミッシング値とそれに続く値が点線で接続されます。3D 棒グラフでは、ミッシング値に相当する平面棒の枠線が実線で表されます。2D 棒グラフでは、グラフにギャップが表示されます。面グラフでは、透明な領域が 0 (ゼロ) の線まで拡張され、それに続く値に拡張されます。
- **ゼロへの補間点線**　折れ線グラフでは、ミッシング値はその点の直前と直後の 2 点を接続する補間点線として表示されます。棒グラフおよび面グラフでは、ミッシング値は補間された (透明な) 棒または面として表示されます。

  注意：ミッシング値を表すデフォルト値を指定することができます。このとき、デフォルト値には 0 (ゼロ) は使用しないでください。この作業には、DEFINE コマンドを使用します。詳細は、ミッシングフィールド値を含むレコード処理を参照してください。

グラフへのミッシング値の表示

```
GRAPH FILE filename
.
.
.
SET VZERO={ON|OFF}
ON GRAPH SET STYLE *
*GRAPH_SCRIPT
API call
*END
ENDSTYLE
END
```

説明

ON

ミッシング値を 0 (ゼロ) として表示します。API Call は、VZERO が ON に設定されている場合は必要ありません。SET VZERO = ON の代わりに、ON GRAPH SET VZERO ON を指定することができます。

OFF

追加される API コールに応じて、ミッシング値をギャップ、0 (ゼロ) への点線、または補間点線として表示します。SET VZERO = OFF の代わりに、ON GRAPH SET VZERO OFF を指定することができます。

API call

VZERO が OFF に設定されている場合のミッシング値のグラフでの表示方法を決定します。利用可能な値には、次のものがあります。

`setFillMissingData(0);` - ミッシング値をギャップとして表示します。

`setFillMissingData(1);` - ミッシング値を 0 (ゼロ) への点線として表示します。

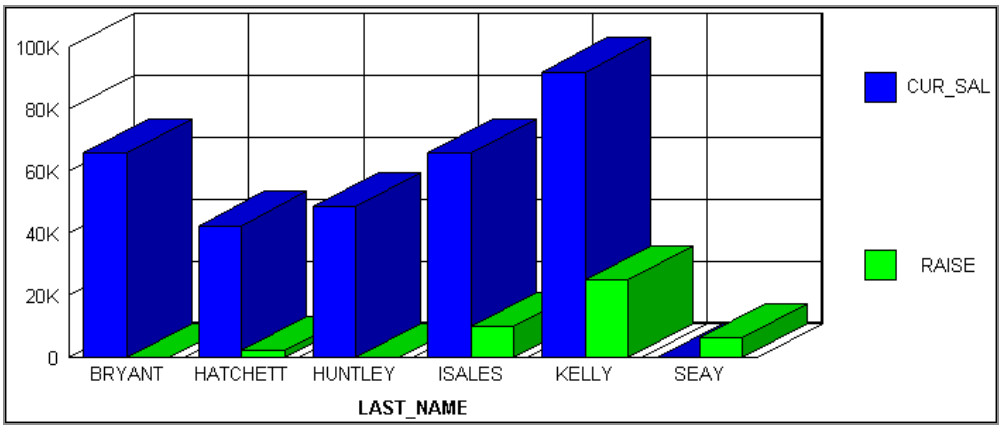
`setFillMissingData(2);` - ミッシング値を補間点線として表示します。

グラフでミッシング値を 0 (ゼロ) として表示

次の例は、0 (ゼロ) として表示するように指定したミッシング値の棒グラフでの表示方法を示しています。SEAY の CUR_SAL 値、および BRYANT と HUNTLEY の RAISE 値が欠落しています。

```
SET LOOKGRAPH=BAR
SET GRAPHEDIT=SERVER
SET GRID=ON
SET VZERO=ON
GRAPH FILE MSFATIA
SUM CUR_SAL
RAISE
ACROSS LAST_NAME
ON GRAPH SET STYLE *
*GRAPH_SCRIPT
setTextRotation(getO1Label(),0);
*END
ENDSTYLE
END
```

出力結果は次のとおりです。



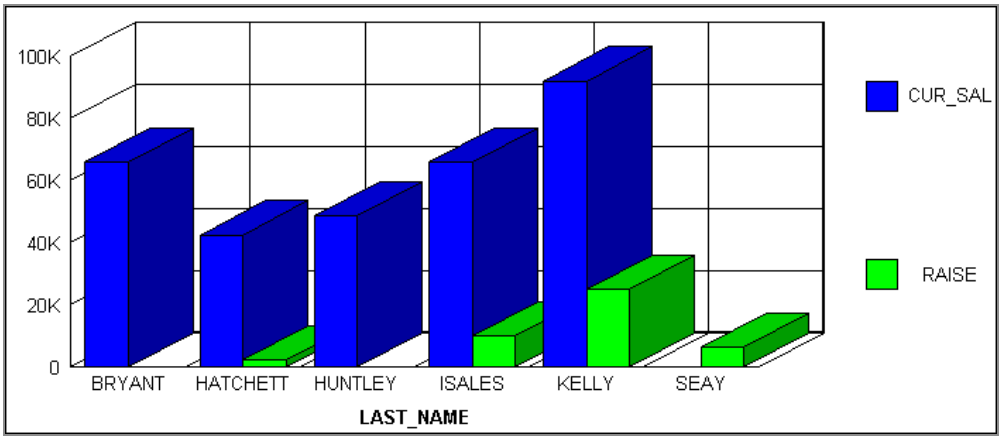
ミッシング値をギャップとして表示

次の例は、ギャップとして表示するように指定したミッシング値の棒グラフでの表示方法を示しています。SEAY の CUR_SAL 値、および BRYANT と HUNTLEY の RAISE 値が欠落しています。

```
SET LOOKGRAPH=BAR
SET GRAPHEDIT=SERVER
SET GRID=ON
SET VZERO=OFF
GRAPH FILE MSFATIA
SUM CUR_SAL
RAISE
```

```
ACROSS LAST_NAME
ON GRAPH SET STYLE *
*GRAPH_SCRIPT
setFillMissingData(0);
setTextRotation(getO1Label(),0);
*END
ENDSTYLE
END
```

出力結果は次のとおりです。



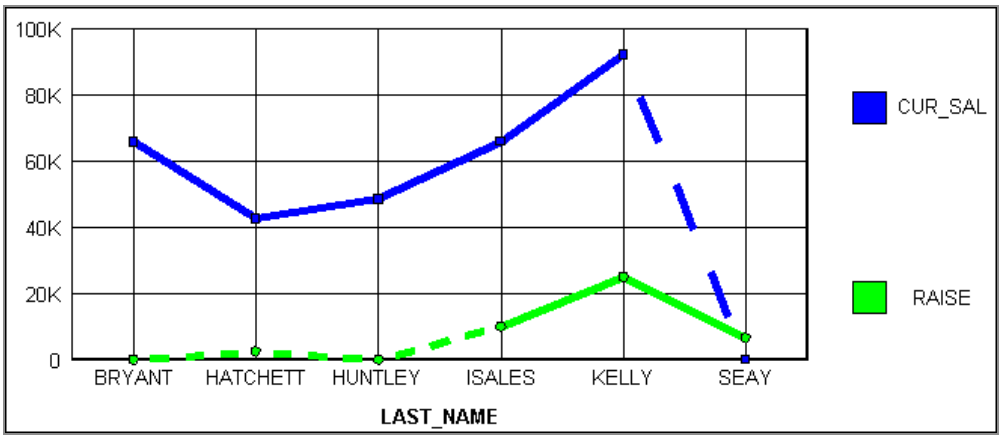
0 (ゼロ) への点線としてのミッシング値の表示

次の例は、0 (ゼロ) への点線として表示するように指定したミッシング値の折れ線グラフでの表示方法を示しています。SEAY の CUR_SAL 値、および BRYANT と HUNTLEY の RAISE 値が欠落しています。

```
SET LOOKGRAPH=LINE
SET GRAPHEDIT=SERVER
SET GRID=ON
SET VZERO=OFF
GRAPH FILE MSFATIA
SUM CUR_SAL
RAISE
ACROSS LAST_NAME
ON GRAPH SET STYLE *
*GRAPH_SCRIPT
setFillMissingData(1);
setTextRotation(getO1Label(),0);
*END
```

```
ENDSTYLE
END
```

出力結果は次のとおりです。

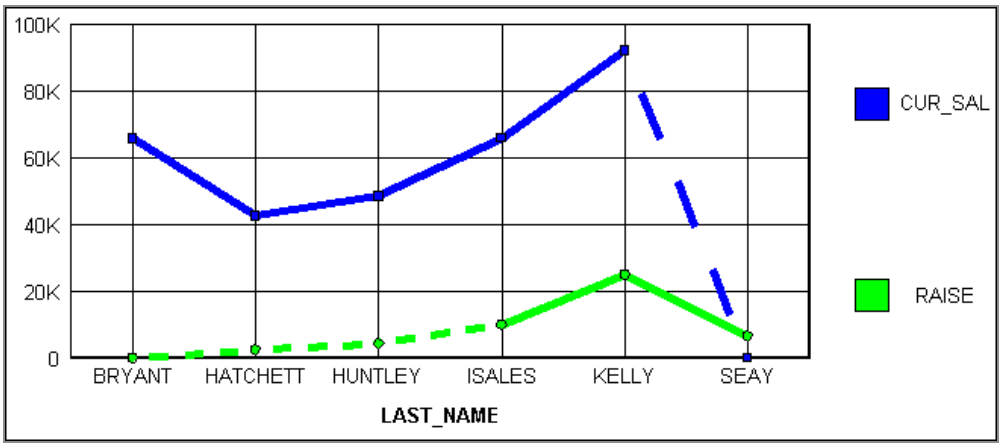


補間点線としてのミッシング値の表示

次の例は、補間点線として表示するように指定したミッシング値の折れ線グラフでの表示方法を示しています。SEAY の CUR_SAL 値、および BRYANT と HUNTLEY の RAISE 値が欠落しています。

```
SET LOOKGRAPH=LINE
SET GRAPHEDIT=SERVER
SET GRID=ON
SET VZERO=OFF
GRAPH FILE MSFATIA
SUM CUR_SAL
RAISE
ACROSS LAST_NAME
ON GRAPH SET STYLE *
*GRAPH_SCRIPT
setFillMissingData(2);
setTextRotation(getO1Label(),0);
*END
ENDSTYLE
END
```

出力結果は次のとおりです。



グラフへの条件付きスタイルの適用

条件付きスタイルを使用し、指定したスタイルと色で特定の定義済みデータを強調することによって、グラフをさらに見やすくできます。

たとえば、販売ノルマに到達していない部門に赤色を適用し、到達した全部門に黒色を適用することができます。この例では、ノルマを達成した部門を即座に把握することができます。ある部門の結果が別の部門の結果に及ぼす影響を検討するため、その可能性を検討するレポートへのドリルダウンの実行が必要な場合があります。

次のグラフタイプに色を適用することができます。

- 棒グラフ
- 連続性のない点を持つ 3D グラフ
- 円グラフ
- 積み上げグラフ

スタイルシートを使用して条件付きスタイルを適用することができます。

注意：条件付きスタイルは、Y 軸の値にのみサポートされています。

条件付きスタイルのグラフへの適用

```
TYPE=DATA,[COLUMN|ACROSSCOLUMN=Nn,]COLOR=color, [WHEN=expression,]$
```

説明

DATA

色が適用されているグラフコンポーネントとしてデータを識別します。この値は、構文の開始位置に記述されている必要があります。

COLUMN|ACROSSCOLUMN

色を適用するグラフサブコンポーネントです。有効なグラフサブコンポーネントは、COLUMN および ACROSSCOLUMN です。

Nn

レポート内の位置に基づいて列を選択します。この値を特定するには、BY フィールド、表示フィールド、ROW-TOTAL フィールドの個数を、NOPRINT フィールドを含めて左から右へ数えます。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

color

グラフのコンポーネントまたはサブコンポーネントに適用する色を識別します。有効な色のリストは、レポートデータのフォーマットを参照してください。

expression

指定した色をグラフのコンポーネントに適用する条件を指定するブール式です。この式は、COMPUTE コマンドの右側で有効である必要があります。詳細は、式の使用を参照してください。

注意：IF...THEN ... ELSE は、WHEN 句では必要なく、サポートされていません.

WHEN 式内の数値以外のリテラルは、すべて一重引用符 (') 内で指定される必要があります。

グラフへの条件付きスタイルの適用

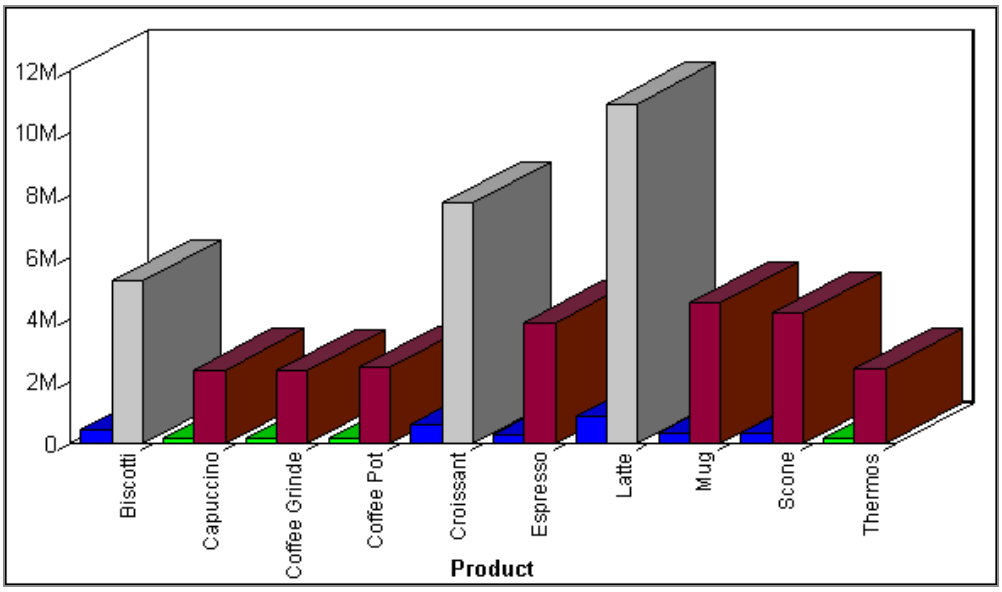
次の例は、条件付きスタイルをどのようにグラフに適用できるかを示しています。

```
   GRAPH FILE GGSALES
   SUM UNITS DOLLARS ACROSS PRODUCT
   ON GRAPH SET STYLE *
1.TYPE=DATA,COLOR=BLUE,$
2.TYPE=DATA,ACROSSCOLUMN=N2,COLOR=FIREBRICK,$
3.TYPE=DATA,ACROSSCOLUMN=N2,COLOR=SILVER,WHEN=N2 GT 5000000,$
```

```
4.TYPE=DATA,ACROSSCOLUMN=N1,COLOR=LIME,WHEN=N1 LT 200000,$
   ENDSTYLE
   END
```

1. この行は、データのデフォルト色に青色を指定します。
2. この行は、DOLLARS フィールドのデフォルト色としてレンガ色を指定します。
3. この行は、DOLLARS の値が 5,000,000 より大きい場合、DOLLARS フィールドに銀色を指定します。
4. この行は、UNITS の値が 200,000 より小さい場合、UNITS フィールドに黄緑色を指定します。

出力結果は次のとおりです。



グラフの別リソースへのリンク

グラフ内の 1 つまたは複数の値にプロシジャまたは URL のリンクを追加して、そのグラフのさらに詳細な情報を表示するレベルまでドリルダウンさせることができます。グラフを実行すると、ここで指定した値が起点 (ホットスポット) となり、リンク先のプロシジャ、JavaScript 関数、または URL を呼び出します。

JSURLS パラメータは、HTML グラフに JavaScript または VBScript ファイルを追加します。これにより、任意の JavaScript または VBScript 関数を使用して WebFOCUS HTML グラフの表示をカスタマイズすることができます。JavaScript および VBScript ファイルは最

後にロードされ、また、リストに記述された順にロードされるため、HTML ページのカスタマイズに制約を与えることはありません。

この機能は、HTML ドキュメントを出力するグラフフォーマットのすべてで機能します (例、JSCHART、PCHOLD FORMAT PNG)。

また、WebFOCUS グラフの実行時に、定義済みの JavaScript 関数セットが呼び出されます。JSURLS を使用して、これらのデフォルト関数を無効にしたり、編集したりすることができます。定義済み JavaScript 関数の一覧を確認するには、/ibi/WebFOCUSxx/webapps/WEB-INF/jsp/global/ibigl.js を参照してください。

構文は次のとおりです。

```
SET JSURLS='/file1 [/file2] [/file3]...'
```

説明

/file1 [/file2] [/file3]...

JavaScript または VBScript が格納されたファイルです。複数の js ファイルを指定する場合は、それぞれをブランクで区切り、値を一重引用符 (') で囲む必要があります。これらのファイルは、Web サーバがアクセス可能な場所に保存しておく必要があります。この値の合計長さは、256 バイトに制限されます。

これらのファイルは URL で参照することができます。

グラフの別リクエストへのリンク

```
TYPE=DATA,[COLUMN|ACROSSCOLUMN=Nn,]COLOR=color,[WHEN=expression,]
FOCEXEC=fex[(parameters ...),]$
```

説明

DATA

色を適用するグラフコンポーネントとしてデータを識別します。TYPE 属性とその値は、構文の先頭部分に配置する必要があります。

COLUMN|ACROSSCOLUMN

色を適用するグラフサブコンポーネントです。有効なグラフサブコンポーネントは、COLUMN および ACROSSCOLUMN です。

color

グラフのコンポーネントまたはサブコンポーネントに適用する色を識別します。有効な色のリストは、レポートデータのフォーマットを参照してください。

Nn

レポート内の位置に基づいて列を選択します。この値を特定するには、BY フィールド、表示フィールド、ROW-TOTAL フィールドの個数を、NOPRINT フィールドを含めて左から右へ数えます。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

FOCEXEC=fex

ユーザがレポートオブジェクトを選択するときに実行されるリンク先プロシジャのファイル名を識別します。

parameters

プロシジャに渡される値です。次の方法を組み合わせることにより、1 つ以上の値を渡すことが可能です。

- 定数値の指定。値は一重引用符 (') で囲みます。
- グラフフィールドの名前または位置の指定。
- 渡される値を含むダイアログマネージャ変数名の指定。
- 変数は、インラインスタイルシート内のみで使用することができます。

注意：通常、変数は定数値を渡すために使用されます。このとき、値は一重引用符 (') で囲みます。以下はその例です。

```
'&ABC'.
```

値を渡す方法は、ハイパーリンクの実行に使用した方法により異なります。値は 1 つ以上渡すことができます。値の文字列全体を括弧で囲み、それぞれをブランクで区切る必要があります。

expression

指定した色をグラフのコンポーネントに適用する条件を指定するブール式です。この式は、COMPUTE コマンドの右側で有効である必要があります。詳細は、式の使用を参照してください。

注意：IF ...THEN ...ELSE は、WHEN 句では必要なく、サポートされていません.

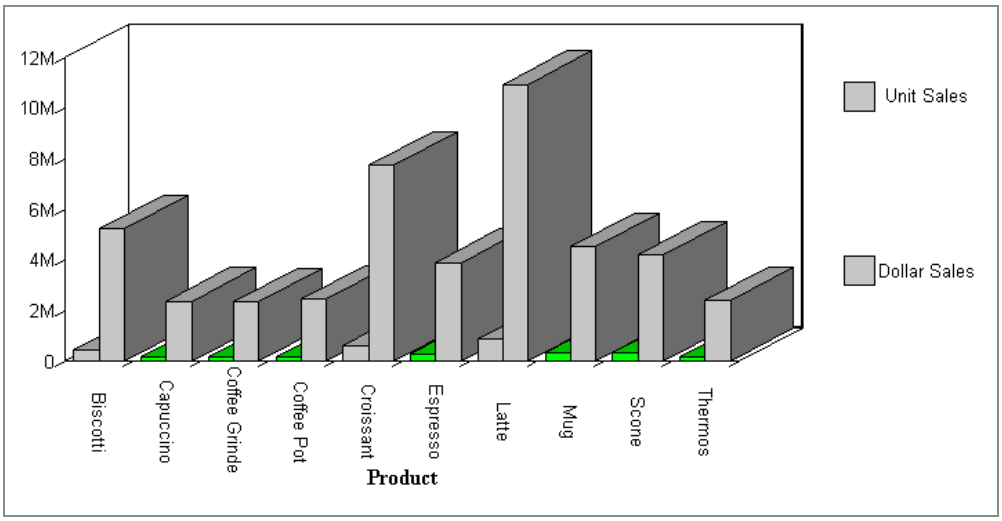
WHEN 式内の数値以外のリテラルは、すべて一重引用符 (') 内で指定される必要があります。

追加レポートまたはグラフへのリンク

次の例では、UNITS の値が 400,000 より小さい場合、黄緑色が指定され、詳細レポートへドリルダウンすることができます。

```
GRAPH FILE GGSALES
SUM UNITS DOLLARS ACROSS PRODUCT
ON GRAPH SET STYLE *
TYPE=DATA,COLOR=SILVER,$
TYPE=DATA,ACROSSCOLUMN=N1,COLOR=LIME,WHEN=N1 LT 400000,FOCEXEC=GRAPH2,$
ENDSTYLE
END
```

出力結果は次のとおりです。



URL へのリンク

任意のコンポーネントから任意の URL へのリンクを定義することができます。この URL には、Web ページ、Web サイト、Servlet プログラム、World Wide Web 以外のリソース（Email アプリケーションなど） を含めることが可能です。リンクの定義後、コンポーネントを選択して URL にアクセスすることができます。

動的なリンクを作成することもできます。動的リンクを使用すると、選択したコンポーネントの値が URL に渡されます。このリソースは、渡された値を使用して、返される結果を動的に決定します。1 つまたは複数のパラメータを渡すことができます。詳細は、パラメータの作成を参照してください。

```
TYPE=type, [subtype], URL=url[(parameters ...)], [TARGET=frame,] [ALT =
'description',] $
```

説明

type

リンクを実行するために、ユーザが Web ブラウザで選択するレポートまたはグラフコンポーネントを特定します。TYPE 属性とその値は、構文の先頭部分に配置する必要があります。

subtype

追加の属性です。COLUMN、LINE、ITEM などがあり、フォーマット設定対象のレポートコンポーネントを特定するために必要です。コンポーネントの指定についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

url

任意の有効な URL を特定します。これには、WebFOCUS Servlet プログラムを指定する URL や、括弧で囲まれたレポートフィールド名も含まれます。この値はリンクのジャンプ先として有効な URL である必要があります。

注意

- URL=url 引数の最大長は、関連する variable=object パラメータを含めて、ブラウザで許容される最大バイト数が上限になります。使用するブラウザでのこの制限についての詳細は、ブラウザベンダーのサポートサイトを参照してください。 URL 引数は 2 行以上に拡張することができます。詳細は、WebFOCUS スタイルシートの作成と管理を参照してください。

  URL の長さは、ブラウザで許容される最大バイト数が上限になります。使用するブラウザでのこの制限についての詳細は、ブラウザベンダーのサポートサイトを参照してください。

- URL がパラメータを持つ WebFOCUS Servlet プログラムを参照する場合、URL を疑問符 (?) で終了する必要があります

parameters

URL に渡される値です。詳細は、パラメータの作成を参照してください。

frame

ドリルダウンリンクからの出力が表示される Web ページのターゲットフレームを指定します。詳細は、ターゲットフレームの指定を参照してください。

description

HTML レポートでサポートされる、セクション 508 (米国リハビリテーション法 508 条) アクセシビリティに準拠したリンクの説明テキストを指定します。説明は、一重引用符 (') で囲みます。

この説明は、レポート出力のリンク上にマウスポインタ (カーソル) を置いたときのポップアップ説明としても表示されます。

JavaScript 関数へのリンク

スタイルシートを使用して、任意のレポートまたはグラフコンポーネントから JavaScript 関数へのリンクを定義することができます。リンクの定義後、レポートまたはグラフコンポーネントを選択して JavaScript 関数を実行することができます。

プロシジャおよび URL へのドリルダウンリンクと同様に、レポートまたはグラフコンポーネントの値を JavaScript 関数へ渡すオプションパラメータを指定することができます。関数は、渡された値を使用して結果を動的に決定し、結果をブラウザに送信します。詳細は、パラメータの作成を参照してください。

注意

- ここで、JavaScript 関数は別の JavaScript 関数を呼び出すことができます。
- JavaScript 関数を実行する場合、ターゲットフレームを指定することはできません。ただし、JavaScript 関数自体の結果としてターゲットフレームを指定することは可能です。

```
TYPE=type, [subtype], JAVASCRIPT=function[(parameters ...)], $
```

説明

type

リンクを実行するために、ユーザが Web ブラウザで選択するレポートコンポーネントを特定します。TYPE 属性とその値は、構文の先頭部分に配置する必要があります。

subtype

追加の属性です。COLUMN、LINE、ITEM などがあり、フォーマット設定対象のレポートコンポーネントを特定するために必要です。詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

function

レポートコンポーネントの選択時に実行する JavaScript 関数を指定します。

すべての関連パラメータを含む JAVASCRIPT=*function* 引数は最大 2400 バイトの長さにすることができ、2 行以上に拡張することができます。単一引数を複数行に分割するには、1 つ目の行末に継続を示す円記号 (¥) を使用する必要があります。ブランクを区切り文字として使用している位置で引数を分割する場合、ブランクは、円記号 (¥) の前、または次の行の先頭に配置する必要があります。円記号 (¥) は区切り文字としては動作しません。

引数が適切に 2 行に分割されます。

```
JAVASCRIPT=myfunc(COUNTRY ¥
CAR MODEL 'ABC'),$
```

注意

- SET JSURLS コマンドまたはダイアログマネージャの -HTMLFORM コマンドを使用して、関数が定義された HTML ドキュメントにグラフを埋め込むことができます。
- -HTMLFORM で呼び出した HTML ドキュメントがある場合、拡張子は (.HTML ではなく) .HTM であることを確認します。

HTMLFORM コマンドについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

parameters

JavaScript 関数に渡される値です。詳細は、パラメータの作成を参照してください。

複数ドリルダウンリンクの作成

```
TYPE=type, [subtype], DRILLMENUITEM='description'|'DrillDown n',
     type_of_link
```

説明

type

リンクを実行するために、ユーザが Web ブラウザで選択するコンポーネントを特定します。TYPE 属性とその値は、構文の先頭部分に配置する必要があります。

subtype

追加の属性です。COLUMN、LINE、ITEM などがあり、フォーマット設定対象のコンポーネントを特定するために必要です。レポートコンポーネントの特定についての詳細は、WebFOCUS スタイルシートを使用したレポートコンポーネントの選択を参照してください。

description

出力結果で、ドリルダウンオプションのポップアップメニューに表示されるテキストです。デフォルト値は ドリルダウン *n* で、*n* はドリルダウン 1、ドリルダウン 2 などの連続する整数です。

type_of_link

リンクのタイプです。たとえば、詳細レポートや URL へのリンクです。有効な値についての詳細は、ドリルダウンリンクの概要を参照してください。

構文ガイドライン

概要リクエストのコンポーネントから複数のドリルダウンリンクを作成し、これらのドリルダウンリンクに、サポートされる動作の任意の組み合わせを指定することができます。たとえば、詳細レポート、グラフ、JavaScript 関数、URL にドリルダウンするリンクを作成することができます。概要リクエストのコンポーネントをクリックすると、ポップアップメニューにドリルダウンオプションのリストが表示されます。

この機能は、見出しおよび脚注には適用されません。

複数ドリルダウンリンクを作成する際は、最初の DRILLMENUITEM の前に単一ドリルダウン動作 (例、プロシジャ、URL) を指定することはできません。

DRILLMENUITEM キーワードにより作成されるメニューには、カスケードスタイルシートファイルを使用してスタイルが設定されます。このファイルは、/ibi/WebFOCUS*xx*/ibi_apps/ibi_html/javaassist/ibi/html/js/multidrill.css です。ここで *xx* は、WebFOCUS のバージョンおよび主要リリース番号を表します。

ヒント：/ibi/WebFOCUS93/ibi_apps/ibi_html/javaassist/ibi/html/js/multidrill.css ファイルを編集することで、DRILLMENUITEM メニューのフォント、サイズ、色を変更することができます。

- /ibi/WebFOCUS93/ibi_apps/ibi_html/javaassist/ibi/html/js/multidrill.css ファイルのバックアップを作成します。
- multidrill.css ファイルに必要な変更を加えます。
- multidrill.css.gz ファイルの名前を変更します。このファイルは、multidrill.css と同一のパスに格納されています。
- キャッシュをクリアします。
- WebFOCUS Reporting Server を再起動します。

ドリルダウンリンクの概要

可能なリンク先は次のとおりです。

- 別のリクエスト。このスタイルシート属性は FOCEXEC です。
- URL。このスタイルシート属性は URL です。有効な URL を渡します。

  URL の長さは、ブラウザで許容される最大バイト数が上限になります。使用するブラウザでのこの制限についての詳細は、ブラウザベンダーのサポートサイトを参照してください。
- フィールドの URL。このスタイルシート属性は URL です。有効な値の URL を持つレポートフィールド名を、リンクのジャンプ先に渡します。
- JavaScript 関数。このスタイルシート属性は JAVASCRIPT です。

パラメータの作成

パラメータを使用することにより、リンク (ドリルダウン) されたレポートに条件を指定することができます。パラメータを定義することにより、ホットスポットをクリックした際に取得する情報の量と種類を制御することができます。

詳細は、レポートの別リソースへのリンクを参照してください。

グラフへのラベルの追加

グラフにラベルを追加すると、そのグラフのデータが示す重要な情報を明確にすることができます。グラフに見出し、脚注 (またはそのいずれか)、および横 (X)、縦 (Y) の軸ラベルを追加することができます。

HEADING 句または FOOTING 句を使用することにより、グラフに見出しまたは脚注を追加することができます。構文は、TABLE リクエストの見出しおよび脚注に使用した句の構文と同一です。グラフの見出しおよび脚注にフィールド値を埋め込むこともできます。この機能は、複数のソートフィールドを含むグラフに注釈を追加するときに特に役立ちます。

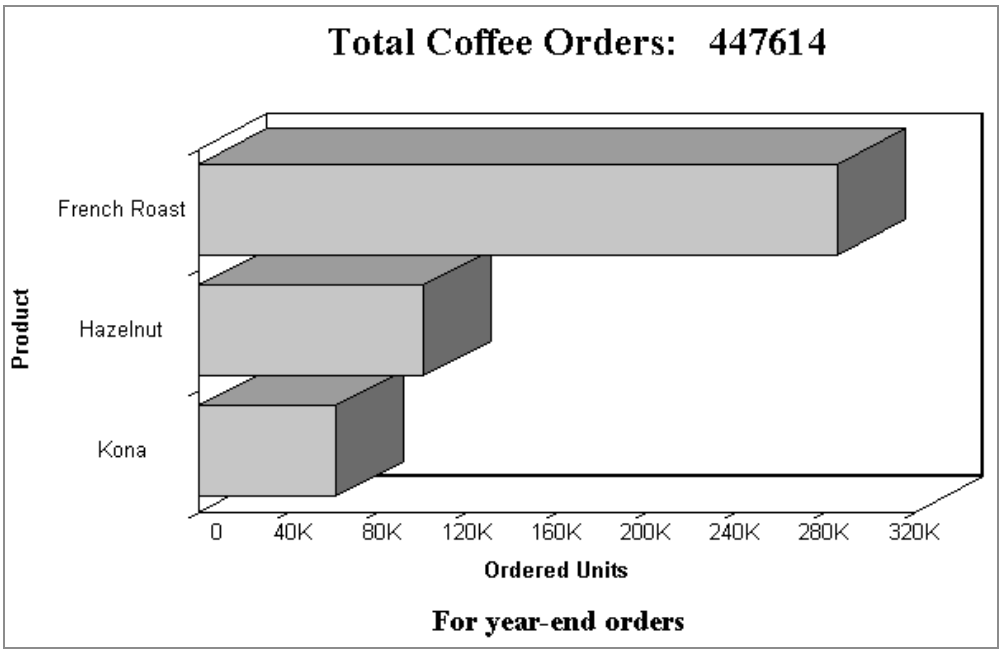
見出し、脚注、埋め込みフィールドについての詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。

グラフへの見出しおよび脚注の追加

次の例は、埋め込みフィールド値を含む見出しのグラフへの追加方法を示しています。この例の見出しは「Total Coffee Orders」で、埋め込みフィールド TOT.QUANTITY が含まれています。

```
GRAPH FILE GGORDER
HEADING CENTER
"Total Coffee Orders: <TOT.QUANTITY "
SUM QUANTITY
BY PRODUCT_DESC
WHERE PRODUCT_DESC EQ 'French Roast' OR 'Hazelnut' OR 'Kona'
FOOTING CENTER
"For year-end orders"
END
```

出力結果は次のとおりです。



グラフへの縦 (Y 軸) および横 (X 軸) ラベルの追加

縦 (Y 軸) および横 (X 軸) グラフラベルは、リクエストに指定された表示フィールドとソートフィールドに従って表示されます。グラフ上に表示されるタイトルは、その特定のフィールドのマスターファイルに表示されるタイトルです。

グラフの縦 (Y 軸) タイトルは、表示フィールドにより決定されます。Y 軸ラベルが複数存在する場合、Y 軸ラベルは表示されません。その代わりに、ラベルは凡例に表示されます。

グラフの横 (X 軸) タイトルは、ソートフィールドにより決定されます。

GRAPH リクエスト内に AS 句を使用することにより、タイトルを置き換えることができます。

グラフへのカスタムスタイルの適用

スタイルシートおよび SET コマンドを使用することにより、グラフをカスタマイズすることができます。さらに、グラフの幅と高さの設定、X 軸と Y 軸の固定スケールの設定、グラフエディタの有効化、グラフ API コールの使用によるカスタマイズも可能です。

グラフの見出しおよび脚注のカスタマイズについての詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。

グラフの高さと幅の設定

グラフの幅 (横軸) は、周囲のあらゆるテキストを含めて、自動的に 760 ピクセルに設定されます。グラフの幅を設定する場合、左マージンに沿って表示される縦軸およびラベルを表示するのに必要なすべてのテキストが収まるよう考慮する必要があります。

表示スペースを最大化するために、AS 句または DECODE 式のいずれかを使用してラベルサイズを制限することができます。

グラフの高さ (縦軸) は、自動的に 400 ピクセルに設定されます。

縦軸は、グラフのプロット値の全範囲が含まれるように自動的に設定 (VAUTO=ON) されます。軸の高さは、見出しおよび脚注の領域を計算に入れ、縦のクラスマーカーに端数を丸めた最適値を表示することを考慮した上で、可能な限り高く設定されます。値の全範囲は、特定の間隔に分割されます。この間隔は、「クラス」と呼ばれます。スケールは、軸に表示する間隔として最適な 10 の倍数および 10 の累乗でクラス値を均等配分するよう設定されます。

グラフの幅設定

```
SET HAXIS={nn|760}
```

説明

{nn|760}

正の数値です。デフォルト値は 760 ピクセルです。

注意：SVG グラフでの最大 HAXIS サイズは 40 インチです。

グラフの高さ設定

```
SET VAXIS={nn|400}
```

説明

{nn|400}

正の数値です。デフォルト値は 400 ピクセルです。

注意：SVG グラフでの最大 VAXIS サイズは 40 インチです。

SET パラメータによるグラフのカスタマイズ

GRAPH 環境には、グラフの外観を制御し、リクエストの実行時にも制御可能なパラメータのセットが含まれます。

たとえば、BSTACK パラメータを使用すると、棒グラフの棒が横並びで表示されるのではなく、積み重ねられて表示されます。

GRAPH リクエストでの SET パラメータの使用

GRAPH 環境を制御するパラメータを設定するには、SET パラメータの適当な組み合わせを使用します。

```
SET parameter=value,parameter=value...
```

サポートする GRAPH パラメータのリストは、グラフの SET パラメータ値および関数を参照してください。

注意

- それぞれの新しい行で SET コマンドを繰り返します。
- 複数のパラメータを 1 行に入力するときは、区切り文字にカンマ (,) を使用します。
- パラメータの一意の短縮名を使用することができます。それらの一意性を確認する必要があります。

GRAPH リクエストでの SET パラメータの使用

次のリクエストは、グラフの高さ (Y 軸) と幅 (X 軸) の設定方法を示しています。

```
SET HAXIS=75,VAXIS=40
GRAPH FILE filename
.
.
.
END
```

グラフの SET パラメータ値および関数

グラフの SET パラメータ	値	パラメータ関数
3D	ON|OFF	ON に設定されている場合、3D グラフが生成されます。OFF に設定されている場合、2D グラフが生成されます。デフォルト値は ON です。
AUTOTICK	ON|OFF	ON に設定されている場合、目盛りの間隔は自動的に設定されます。デフォルト値は ON です。HTICK および VTICKも参照してください。
BARNUMB	ON|OFF	ON に設定されている場合、棒グラフの棒の末端または円グラフのスライスに値の集計を表示します。デフォルト値は OFF です。

グラフの SET パラメータ	値	パラメータ関数
BSTACK	ON|OFF	ON に設定されている場合、棒グラフの棒は並べて表示されず、積み重ねられます。デフォルト値は OFF です。
GRAPHEDIT	graphedit	WebFOCUS バージョン 8.0 以降では、このパラメータは廃止されました。
GRID	ON|OFF	ON に設定されている場合、グラフの横および縦クラスマークで罫線が描画されます。VGRID も参照してください。デフォルト値は OFF です。
HAUTO	ON|OFF	ON に設定されている場合、ユーザが上書きしない限り、横軸の目盛りが自動的に指定されます。OFF に設定されている場合、ユーザは HMAX および HMIN に値を指定する必要があります。デフォルト値は ON です。
HAXIS		横軸の幅をバイト数で指定します。このパラメータは、オフラインで生成したグラフに対して調整することができます。HAXIS は、グラフの幅が表示領域の幅に応じて自動的に調整されることから、オンライン表示では無視されます。

グラフの SET パラメータ	値	パラメータ関数
HCLASS	nnn	AUTOTICK=OFF に設定されている場合、横間隔マークを指定します。HTICK も参照してください。
HISTOGRAM	ON|OFF	ON に設定されている場合、横軸の値が数値でないとき、曲線ではなく、ヒストグラムが生成されます。デフォルト値は ON です。
HMAX	nnn	自動目盛りが使用されない (HAUTO=OFF) 場合、横軸上の最大値を指定します。
HMIN	nnn	自動目盛りが使用されない (HAUTO=OFF) 場合、横軸上の最小値を指定します。
HSTACK	ON|OFF	ON に設定されている場合、ヒストグラム上の棒は並べて表示されず、積み重ねられます。デフォルト値は OFF です。
HTICK	nnn	AUTOTICK が OFF に設定されている場合、横軸の目盛り間隔を指定します。HCLASS も参照してください。
LOOKGRAPH	option	グラフのタイプを指定します。詳細は、LOOKGRAPH によるグラフスタイルの決定を参照してください。

グラフの SET パラメータ	値	パラメータ関数
PIE	ON|OFF	ON に設定されている場合、円グラフを指定します。デフォルト値は OFF です。
VAUTO	ON|OFF	ON に設定されている場合、ユーザが上書きしない限り、縦軸の目盛りが自動的に指定されます。OFF に設定されている場合、ユーザは VMAX および VMIN に値を指定する必要があります。デフォルト値は ON です。
VAXIS		ページの長さを行数で指定します。このパラメータは、オフラインで生成したグラフに対して調整することができます。VAXIS は、グラフの高さが表示領域の高さに応じて自動的に調整されることから、オンライン表示では無視されます。
VCLASS	nnn	AUTOTICK=OFF に設定されている場合、縦間隔マークを指定します。VTICK も参照してください。
VGRID	ON|OFF	ON に設定されている場合、グラフの横および縦クラスマークで罫線が描画されます。GRID も参照してください。デフォルト値は OFF です。

グラフの SET パラメータ	値	パラメータ関数
VMAX	nnn	自動目盛りが使用されない (VAUTO=OFF) 場合、縦軸上の最大値を指定します。
VMIN	nnn	自動目盛りが使用されない (VAUTO=OFF) 場合、縦軸上の最小値を指定します。
VTICK	nnn	AUTOTICK が OFF に設定されている場合、縦軸の目盛り間隔を指定します。VCLASS も参照してください。
VZERO	ON|OFF	Y 軸の値が格納されるか、無視されるかを決定します。ON に設定されている場合、Y 軸に沿ったミッシングデータは 0 (ゼロ) として処理されます。OFF に設定されている場合、Y 軸に沿ったミッシングデータは無視されます。値はプロットマトリックスには格納されません。デフォルト値は OFF です。

X 軸固定スケールの設定

横方向スケールは、グラフが作成される値の全範囲が含まれるように自動的に設定 (HAUTO=ON) されます。値の全範囲は、特定の間隔に分割されます。この間隔は、「クラス」と呼ばれます。スケールは、軸にプロットされる間隔が、10 の倍数の適正値および累乗に端数が処理されたクラス値になるように正規化されます。

上限および下限に固定値 (一定のスケールを必要とする一連のグラフを作成するときに役立ちます) を割り当てるには、自動スケールメカニズムをオフにして、HAUTO=OFF を設定することにより新しい制限値を設定します。

X 軸固定スケールの設定

```
SET HAUTO=OFF, HMAX=nn, HMIN=nn
```

説明

HAUTO

自動目盛り機能です。HAUTO が ON に設定されている場合、HMAX および HMIN の値はすべて上書きされます。

HMAX=nn

横軸に上限を設定します。デフォルト値は 0 です。

HMIN=nn

HAUTO が OFF に設定されている場合、横軸の下限を設定します。デフォルト値は 0 です。

注意

- 1 行に複数の SET パラメータを入力する場合、区切り文字にカンマ (,) を使用します。
- すべてのデータ値を組み込まない制限を定義する場合、OVER または UNDER が表示され、抽出されたデータの一部がグラフに反映されていないことを示します。

Y 軸固定スケールの設定

縦方向スケール固定に上限および下限 (一定のスケールを必要とする一連のグラフを作成するときに役立ちます) を指定するには、自動スケールメカニズムをオフにして、SET VAUTO=OFF コマンドを使用し、固定値を設定します。

Y 軸固定スケールの設定

```
SET VAUTO=OFF, VMAX=nn, VMIN=nn
```

説明

VAUTO

自動目盛り機能です。VAUTO が ON に設定されている場合、VMAX および VMIN の値はすべて上書きされます。

VMAX=nn

VAUTO が OFF に設定されている場合、縦軸の上限を設定します。デフォルト値は 0 です。

VMIN=nn

VAUTO が OFF に設定されている場合、縦軸の下限を設定します。デフォルト値は 0 です。

注意

- 1 行に複数の SET パラメータを入力する場合、区切り文字にカンマ (,) を使用します。
- すべてのデータ値を組み込まない制限を定義する場合、OVER または UNDER が表示され、抽出されたデータの一部がグラフに反映されていないことを示します。

グラフ API および HTML5 JSON プロパティによるグラフのカスタマイズ

GRAPH リクエストの ON GRAPH SET STYLE * コマンドと ENDSTYLE コマンドの間に API コールを手動で追加することで、グラフ出力をさらに強化することができます。HTML5 グラフを作成する場合は、JSON (JavaScript Object Notation) メソッドおよびプロパティを追加することもできます。変更を保存すると、その変更に対応する API コールおよびプロパティが、グラフスタイルシートに書き込まれます。

リクエストのスタイルシートセクションに JSON と API コールを追加した場合、最初に API コールが解析され、次に JSON が解析されます。そのため、API コールと同一のプロパティを JSON が設定する場合、JSON のプロパティが優先されます。一般に、前の宣言で設定されたプロパティは、後の宣言で設定された宣言で上書きされます。

グラフ API によるグラフのカスタマイズ

```
GRAPH FILE filename
graph commands
ON GRAPH SET STYLE *
*GRAPH_SCRIPT
API calls
*END
```

```
*GRAPH_JS
JSON
*END
```

```
WEBFOCUS StyleSheet commands
ENDSTYLE
END
```

説明

filename

グラフのデータソースを指定します。

API calls

API コールです。API コールは、GRAPH_SCRIPT ブロックの *GRAPH_SCRIPT 宣言と *END 宣言の間に追加する必要があります。リクエストには、スタイルセクション内の任意の位置に複数の GRAPH_SCRIPT ブロックを含めることができます。

JSON

HTML5 グラフ出力に適用する JSON メソッドおよびプロパティです。これらは、GRAPH_JS ブロックの *GRAPH_JS 宣言と *END 宣言の間に追加する必要があります。リクエストには、スタイルセクション内の任意の位置に複数の GRAPH_JS ブロックを含めることができます。

WEBFOCUS StyleSheet commands

スタイルシートコマンドについての詳細は、WebFOCUS スタイルシートの作成と管理を参照してください。

グラフ API によるグラフのカスタマイズ

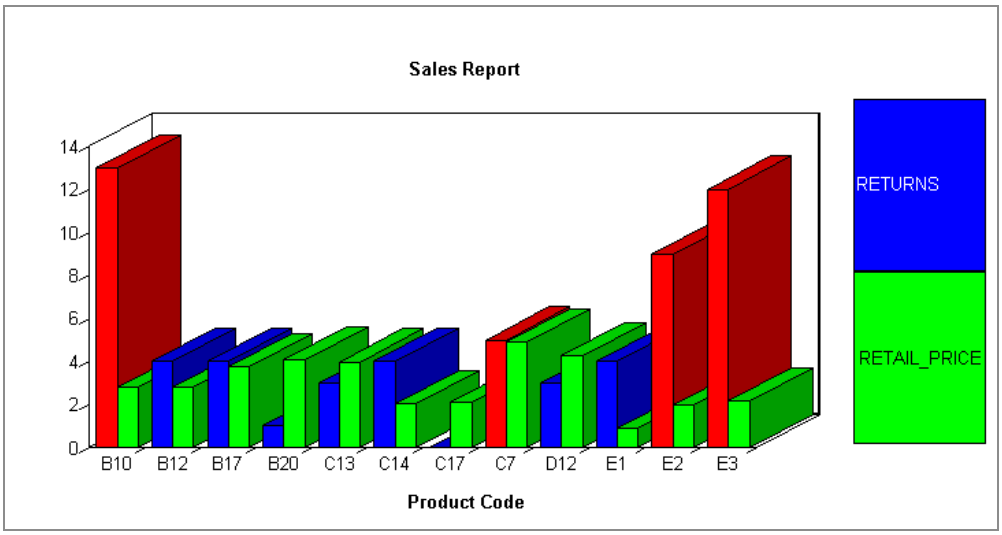
次の注釈付きサンプルは、ON GRAPH SET STYLE * を使用してグラフをカスタマイズする方法を示しています。グラフ API コードは、リクエスト内に太字で表示されています。

```
   GRAPH FILE SALES
   SUM RETURNS
   RETAIL_PRICE
   ACROSS PROD_CODE AS 'Product Code'
   ON GRAPH SET STYLE *
   *GRAPH_SCRIPT
1. setLegendMarkerPosition(4);
2. setO1LabelRotate(0);
3. setTitleString("Sales Report");
4. setTextJustHoriz(getTitle(),1);
   *END
   DEFMACRO=COND0001, MACTYPE=RULE, WHEN=RETURNS GT 4,$
   TYPE=DATA,MACRO=COND0001,ACROSSCOLUMN=RETURNS,COLOR=RED,$
   ENDSTYLE
   END
```

説明

1. 凡例テキストを凡例マーカー内に表示します。
2. X 軸ラベルを横方向に表示します。
3. タイトル (Sales Report) を表示します。
4. タイトルを中央揃えにします。

出力結果は次のとおりです。



イメージファイルとしてのグラフの保存

GRAPHSERVURL パラメータまたは WebFOCUS Reporting Server 上の JSCOM3 構成を使用することにより、グラフ出力をイメージファイルとして保存することができます。グラフ出力をイメージファイルとして保存する方法は、TABLE リクエストや GRAPH リクエストからの出力など、複数の出力を含む PDF ファイルまたは HTML ファイルを 1 つ作成する場合に役立ちます。この種のレポートは、ReportCaster を使用して配信することができます。

詳細は、イメージファイルとしてのグラフの保存 (GRAPHSERVURL 使用)を参照してください。

イメージファイルとしてのグラフの保存 (GRAPHSERVURL 使用)

GRAPHSERVURL パラメータにより、WebFOCUS Reporting Server が Windows、UNIX マシンのいずれかにインストールされているサーバ環境に対して実行しているユーザは、グラフ出力を GIF ファイルとして保存することができます。GIF イメージは、PDF または HTML レポート内に埋め込むことができます。

GRAPHSERVURL パラメータは、WebFOCUS Graph Servlet がインストールされているマシンに HTTP リクエストを送信します。グラフイメージは、WebFOCUS Graph Servlet により作成され、イメージは WebFOCUS Reporting Server の一時的なロケーション (アロケーションが指定されていない場合) または FILEDEF コマンドで指定されたロケーションに返信

されます。アロケーションウィザードを使用して FILEDEF コマンドを作成することができます。

イメージファイルとしてのグラフの保存 (GRAPHSERVURL 使用)

手順

1. 使用しているプラットフォームに対応する『ibi™ WebFOCUS® インストールガイド』の要件に従って JDK をインストールします。
2. イメージを作成するプロシジャを作成し、プロシジャ内の GRAPHSERVURL に WebFOCUS Graph Servlet を呼び出す URL を指定します。以下はその例です。

   ```
   SET GRAPHSERVURL=http://hostname/ibi_apps/IBIGraphServlet
   ```

   説明

   hostname

   WebFOCUS がインストールされているマシン名です。

   詳細は、イメージファイルとしてのグラフの保存を参照してください。
3. ブラウザを使用して、このプロシジャを WebFOCUS Reporting Server 上で直接実行します。

イメージファイルとしてのグラフの保存

```
[FILEDEF filename DISK drive:¥...¥filename.fmt]
SET GRAPHSERVURL= graph_servlet_URL
GRAPH FILE file
graph commands
ON GRAPH HOLD AS filename FORMAT fmt
END
```

説明

FILEDEF

指定したロケーションにイメージファイルを保存します。

filename

イメージファイルに付ける名前です。この名前は、FILEDEF コマンドのファイル名と一致させる必要があります。ファイル名の入力を要求するには、&FILENAME などの変数をプロシジャに記述します。

fmt

グラフの格納先とするイメージファイルのタイプです。有効な値は、PNG、SVG、GIF、JPG です。

graph_servlet_URL

WebFOCUS Graph Servlet を呼び出す URL です。最大文字数は 256 バイトです。

file

作成するレポートのデータソース名です。

注意：永続的な場所に保存されているイメージを挿入するには、イメージファイルの完全修飾パスを指定する必要があります。たとえば、GIF ファイルを挿入する場合は、次のように指定します。

```
TYPE=REPORT, IMAGE=drive:¥...¥ filename.gif
```

説明

drive:¥...¥filename.gif

GIF ファイルが存在するロケーションのパスです。WebFOCUS Reporting Server が同一ドライブ上にインストールされている必要があります。

GIF イメージの PDF レポートへの挿入

次の例は、GRAPH リクエストから GIF ファイルを作成し、PDF レポートに埋め込む方法を示しています。

1. EDA パスまたはアプリケーションパスへアクセスできるロケーションにリモートプロシジャを作成します。たとえば、ローカルサーバに対して実行する場合は次のようになります。

```
SET GRAPHSERVURL= http://localhost/ibi_apps/IBIGraphServlet
GRAPH FILE CENTORD
SUM LINEPRICE
ACROSS PLANTLNG AS 'Plant'
ON GRAPH HOLD AS PLANT FORMAT GIF
END
```

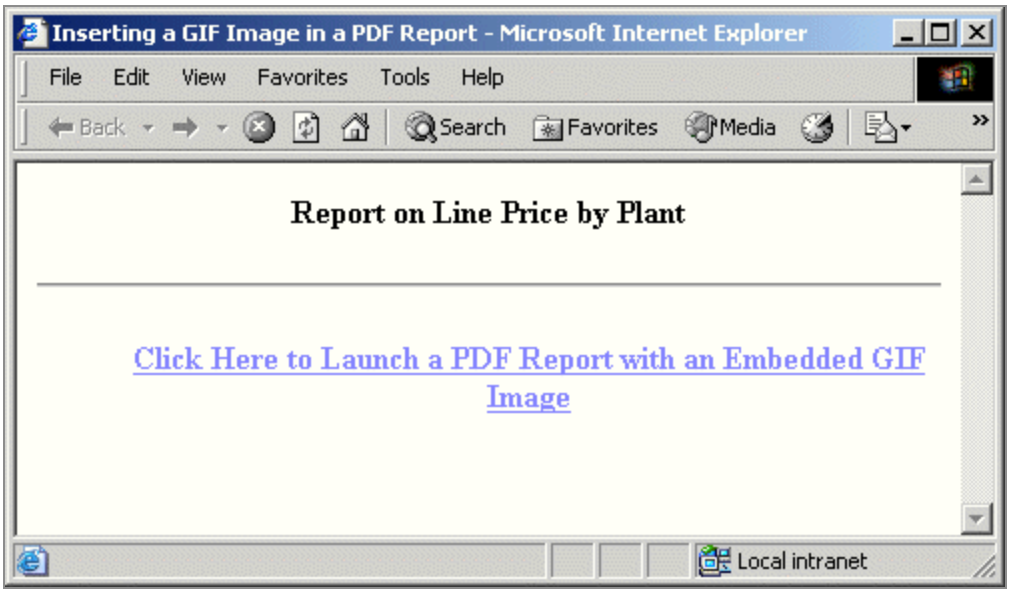
```
TABLE FILE CENTORD
SUM LINEPRICE
BY PLANTLNG AS 'Plant'
ON TABLE SET STYLE *
TYPE=REPORT, IMAGE=plant.gif, POSITION=(4 0), SIZE=(5 3), $
ENDSTYLE
ON TABLE PCHOLD FORMAT PDF
END
```

注意：JSCOM3 を使用している場合、プロシジャから SET GRAPHSERVURL パラメータを削除することができます。

2. プロシジャ名を HOLDGIF として保存します。
3. プロシジャ名 HOLDGIF をハイパーリンクから呼び出す HTML ファイルを作成します。以下はその例です。

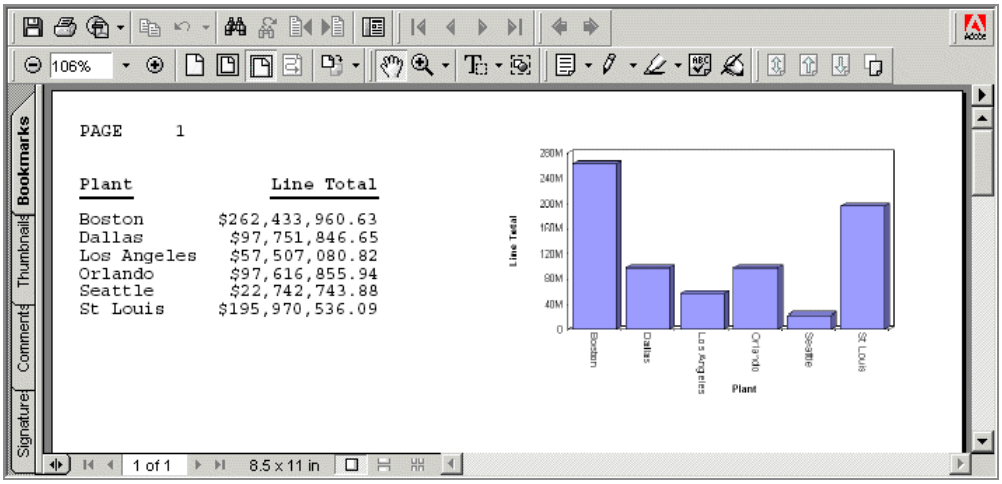
```
<HTML>
<HEAD>
<TITLE> Inserting a GIF Image in a PDF Report </TITLE>
</HEAD>
<BODY>
<H4 ALIGN=CENTER>Report on Line Price by Plant</H4>
<HR>
<P><FONT SIZE=+2></FONT></P>
<UL TYPE=SQUARE
<LI><A HREF="http://localhost/ibi_apps/WFServlet?IBIF_ex=holdgif">
<H4 ALIGN=CENTER>Click Here to Launch a PDF Report with an Embedded
    GIF Image</H4></A>
</UL>
</BODY>
</HTML>
```

次のような起動ページが表示されます。



これで、ブラウザからレポートを実行する準備が整いました。ReportCaster を使用してレポートを配信するには、実際のプロシジャ (ここでは、HOLDGIF) をスケジュールします。

4. リンクをクリックし、レポートを実行します。出力結果は次のとおりです。



注意：このプロシジャを BI Portal レポートとして実行するには、IMAGE 属性を含むスタイルシートの開始位置に -MRNOEDIT コマンドを追加します。これにより、BI Portal がリポジトリ内の GIF ファイルを検索しなくなります。指定されたイメージは、WebFOCUS Reporting Server 上に存在する必要があります。

-MRNOEDIT 構文は、大文字と小文字を区別しません。単一行または複数行で使用することができます。以下はその例です。

単一行

```
-MRNOEDIT TYPE=REPORT, IMAGE=PLANT.gif, POSITION=(4 0), SIZE=(5 3), $
```

または

```
-MRNOEDIT TYPE=REPORT, IMAGE=PLANT.gif,
-MRNOEDIT POSITION=(4 0), SIZE=(5 3), $
```

複数行

```
-MRNOEDIT BEGIN
TYPE=REPORT, IMAGE=PLANT.gif, POSITION=(4 0), SIZE=(5 3), $
-MRNOEDIT END
```

グラフ保存上の注意

WebFOCUS 管理コンソールの [出力先変更設定] 下で [保存レポート] のチェックをオンにすると、出力ファイルを保存するか開くかを選択するウィンドウが表示されます。プロシジャに、出力ファイルに AS 名を指定する PCHOLD コマンドが含まれている場合、ファイルを保存する際に、その名前が保持されます。AS 名が指定されていない場合、無作為なファイル名が生成されます。

重要：PNG、SVG、GIF、または JPG フォーマットでプロシジャに指定された WebFOCUS GRAPH リクエストの [保存レポート] 機能を活用するには、WebFOCUS 管理コンソールで次のことを実行する必要があります。

- .htm 拡張子に関しては、[保存レポート] を [yes] に設定します。

  サーバサイド GRAPH リクエストを実行すると、実際のグラフ出力へのリンクを含む HTM ファイルが作成されます。このグラフ出力は、一時イメージファイルとして、.jpg、.gif、.svg、.png のいずれかの拡張子で保存されているファイルです。

  GRAPH リクエスト実行時に、出力を開くか保存するかの選択が要求された場合、[保存] を選択すると、この出力は、グラフイメージの参照のみが含まれた HTM ファイルとして保存されます。これは、WebFOCUS Client 構成の一時ファイルの期限切れ設定によって、最終的には期限切れとなり、サーバから削除されます。

  GRAPH リクエストの出力を保存するには、保存された HTM ファイルを開き、グラフイメージを右クリックして [名前を付けて画像を保存] を選択し、ディスクに永久保存します。これを実行すると、HTM 出力ファイルでは、保存されたイメージファイルを完全参照するよう変更することができます。

- [保存] をクリックして、[出力先変更設定] パネルで加えた変更を保存します。

財務レポートの作成

FML (Financial Modeling Language) は、貸借対照表、連結、または予算などの財務指向データの作成、計上、提示に関連する特別なニーズに対応します。これらのレポートは、計算が行間または列間で実行され、各行が入力データから直接集計可能、あるいはデータの関数として計算される一意または一連のエントリを表す点で、他のレポートとは異なります。

FML レポート

FML は、TABLE コマンドを統合した拡張機能です。FOR 句と RECAP コマンドを追加することにより、広範囲のアプリケーションを処理することができます。

注意：MORE は FML リクエストではサポートされません。

FML は、ダイアログマネージャと連結して、「WHAT IF」を使用した条件シナリオの評価、および完全な意思決定サポートシステムの開発に使用することができます。これらのシステムは、標準的な財務表に加えて、統計分析およびグラフなど、ビジネスインテリジェンス機能を利用することができます。

FML を使用したプロシジャはデータの種類には依存せず、他のリクエストと同様に、簡単に変更することができます。FML には、次の機能が含まれます。

- **行と列のフォーマット**　結果を行ごとまたは列ごとに指定することができます。詳細は、行間の計算実行を参照してください。
- **中間結果**　FML 結果を外部ファイルに保存し、後の分析で使用することができます。これは、中間結果を作成し、最終プロシジャで結果を連結する際に役立ちます。詳細は、中間レポート結果の保存と取得を参照してください。
- **インラインデータ入力**　FML では、データソースから抽出するデータ値に加えて、プロシジャ内で定数を指定することができます。詳細は、リクエストの直接データ入力を参照してください。
- **再帰的レポート**　ある期間またはフィールドの最終値が次回の開始値になるレポートを作成できます。たとえば、キャッシュフロー予測を作成するために再帰的レポートを使用することができます。詳細は、再帰的モデルの作成を参照してください。
- **勘定科目表または類似階層の動的レポート**　最新の構造とその値を反映する情報を自動検索し、情報の構成が変わるたびに変化するレポートを作成することができます。マ

スターファイルやリクエストを変更する必要はありません。詳細は、動的階層レポートを参照してください。

FML リクエストのサンプル

次の例では、2 年間の結果を比較する簡単な資産表を作成します。この作業を通して、FML のさまざまな重要機能について説明します。プロシジャ行の左側の番号は、リクエストに続く説明に対応します。

```
   TABLE FILE FINANCE
   HEADING CENTER
   "COMPARATIVE ASSET SHEET </2"
   SUM AMOUNT ACROSS HIGHEST YEAR
   WHERE YEAR EQ '1983' OR '1982'
1.FOR ACCOUNT
2. 1000           AS 'UTILITY PLANT'                 LABEL   UTP     OVER
2. 1010 TO 1050   AS 'LESS ACCUMULATED DEPRECIATION' LABEL   UTPAD   OVER
3.BAR                                                                OVER
4.RECAP UTPNET = UTP-UTPAD; AS 'TOTAL PLANT-NET'                     OVER
   BAR                                                                OVER
    2000 TO 3999  AS 'INVESTMENTS'                   LABEL   INV     OVER
5."CURRENT ASSETS"                                                   OVER
    4000          AS 'CASH'                          LABEL   CASH    OVER
    5000 TO 5999  AS 'ACCOUNTS RECEIVABLE-NET'       LABEL   ACR     OVER
    6000          AS 'INTEREST RECEIVABLE'           LABEL   ACI     OVER
    6500          AS 'FUEL INVENTORY'                LABEL   FUEL    OVER
    6600          AS 'MATERIALS AND SUPPLIES'        LABEL   MAT     OVER
    6900          AS 'OTHER'                         LABEL   MISC    OVER
    BAR                                                              OVER
   RECAP TOTCAS=CASH+ACR+ACI+FUEL+MAT+MISC;AS 'TOTAL CURRENT ASSETS' OVER
    BAR                                                              OVER
    7000          AS 'DEFERRED DEBITS'               LABEL   DEFDB   OVER
```

```
    BAR                                                              OVER
6.RECAP TOTAL = UTPNET+INV+TOTCAS+DEFDB; AS 'TOTAL ASSETS'           OVER
    BAR AS '='
    FOOTING
    "</2 *** PRELIMINARY ASSET SHEET BASED ON UNAUDITED FIGURES ***"
   END
```

1. FOR および OVER は、行ごとにレポートを構成する FML 句です。
2. LABEL は、RECAP 計算で使用する行に変数名を指定します。

1000 および 1010 TO 1050 はタグで、FOR フィールドのデータ値である FINANCE データソースの ACCOUNT を識別します。レポートの行は、単一データ値 (ここでは 1000)、複数データ値、または値の範囲 (ここでは 1010 TO 1050) を表すタグに関連付けることができます。

3. BAR は、RECAP 計算実行前の数値フィールドに下線を追加します。
4. RECAP コマンドは、レポートで LABEL により識別されている値を基に新しい値を作成します。この場合、UTPNET という値が UTP と UTPAD から取得されます。この値は、レポートで意味が通じるように AS 句により「TOTAL PLANT-NET」という名前に変更されます。
5. 下線と同様に、テキストを FML レポートの任意の位置に追加することができます。
6. この RECAP コマンドは、データソースから直接抽出した値、および実行済みの RECAP 計算 (UTPNET および TOTCAS) から取得した値から合計 (TOTAL ASSETS) を計算します。

出力結果は次のとおりです。

COMPARATIVE ASSET SHEET

	YEAR 1983	1982
UTILITY PLANT	1,430,903	1,294,611
LESS ACCUMULATED DEPRECIATION	249,504	213,225
TOTAL PLANT-NET	1,181,399	1,081,386
INVESTMENTS	818	5,639
CURRENT ASSETS		
CASH	4,938	4,200
ACCOUNTS RECEIVABLE-NET	28,052	23,758
INTEREST RECEIVABLE	15,945	10,206
FUEL INVENTORY	35,158	45,643
MATERIALS AND SUPPLIES	16,099	12,909
OTHER	1,264	1,743
TOTAL CURRENT ASSETS	101,456	98,459
DEFERRED DEBITS	30,294	17,459
TOTAL ASSETS	1,313,967	1,202,943

*** PRELIMINARY ASSET SHEET BASED ON UNAUDITED FIGURES ***

データからの行作成

TABLE リクエストは通常、BY 句に従ってレポートの行をソートします。抽出したデータは、リクエストに応じて昇順または降順にソートされます。条件選択句を使用して行を特定のサブセットに制限することができます。ただし、次の条件が適用されます。

- 行はソート順で表示される。
- ファイルから抽出された値を持つ行のみが表示される。
- テキストの挿入は、ソートフィールドにより値が次のように変更された場合にのみ可能。

```
ON DIVISION SUBFOOT
```

- 計算の挿入は、ソートフィールドにより値が次のように変更された場合にのみ可能。

```
ON DIVISION RECAP
```

これとは対照的に、FML FOR 句はレポートを行単位で構成するマトリックスを作成します。この構成により、レポートに組み込まれたデータとその出力結果をより制御しやすくすることができます。次の処理を実行することができます。

- データソースフィールドの特定のデータ値に対するレポートを作成し、特定のデータ値を共通ラベルで併合して計算に使用する。
- リクエストにデータを直接入力し、データソースから抽出されたデータに補足する。
- レポートのソート区切りに関連しない点に、テキスト、下線、計算を含める。
- 再帰的処理を実行し、結果保存された中間計算値を次の計算の開始位置として使用する。
- データが抽出されない行を非表示にする。
- 行をラベルで、列を番号、アドレス、値のいずれかで識別する。これにより、スプレッドシートと同様に、行と列の交点に位置するセルを指定することができます。

データソースからの FOR フィールド値の抽出

行を指定する構文は次のとおりです。

```
FOR fieldname [AS 'coltitle'] value [OR value OR...] [AS 'text']
[LABEL label] OVER
.
.
.
[value [OR value ...]] [AS 'text'] [LABEL label]
END
```

説明

fieldname

FML レポートの FOR フィールドです。

coltitle

レポート出力に表示する FOR フィールドのフィールドタイトルです。

value

レポートのこの行に抽出されたデータを記述する値です。「タグ値」とも呼ばれます。

AS 'text'

出力結果のタグ値を置き換える名前を指定します。テキストは一重引用符 (') で囲みます。

label

RECAP 式で参照される行にラベルを指定します。ラベルの最大値は 66 バイトです。ブランクや特殊文字は使用できません。指定する明示的ラベルは、それぞれ一意である必要があります。

明示的ラベルを指定した場合でも、位置ラベル (例、R1、R2) は内部的に保持されます。

デフォルト設定では、FOR フィールド (例、1010) のタグ値は、FML マトリックスに一度だけ追加することができます。FOR フィールドのある値をマトリックスに複数回追加するには、FORMULTIPLE パラメータを有効にします。FORMULTIPLE パラメータのデフォルト値は OFF です。

FML リクエストで使用されるタグ値を返す FMLFOR、FMLLIST、FMLINFO 関数についての詳細は、『ibi™ WebFOCUS® 関数リファレンス』を参照してください。

データソース値からの行の作成

企業の各アカウントに使用する財務データを含む、次のデータソースがあることを想定します。

```
CHART OF ACCOUNTS

ACCOUNT          DESCRIPTION

1010             CASH ON HAND
1020             DEMAND DEPOSITS
1030             TIME DEPOSITS
```

```
1100             ACCOUNTS RECEIVABLE
1200             INVENTORY
.                     .
.                     .
.                     .
```

FML で FOR 句を使用することにより、次の TABLE リクエストを発行することができます。ここでは、ACCOUNT の各値がタグ (例、1010、1020) により個々の行に表示されています。

```
TABLE FILE LEDGER
SUM AMOUNT
FOR ACCOUNT
1010 OVER
1020 OVER
1030 OVER
1100 OVER
1200
END
```

出力結果は次のとおりです。

```
      AMOUNT
      ------
1010   8,784
1020   4,494
1030   7,961
1100  18,829
1200  27,307
```

複数レコードからの行作成

データソースの複数の値を組み合わせて FML レポートの行に使用する場合、いくつかの方法があります。次のものを使用することができます。

- OR 句 - 1 つの式で複数のタグ値の合計を計算するときに使用します。詳細は、OR 句による行の合計値を参照してください。
- TO 句 - レポートに含めるタグ値の範囲を識別するときに使用します。詳細は、TO 句による値範囲の指定を参照してください。
- マスク文字 - タグを 1 つずつ指定せず、グループとして指定するときに使用します。詳細は、マスク文字によるタグ値の抽出を参照してください。

デフォルト設定では、FOR フィールド値は FML マトリックスの 1 行のみに指定できますが、FORMULTIPLE パラメータを ON に設定することにより、FML マトリックスの複数の行に 1 つのデータ値を指定することができます。たとえば、ある値を 1 行に単独で指定、別の行の範囲の一部として使用、さらに別の行の計算に含めることができます。

また、1 行に含める複数のタグを外部ファイルから抽出することもできます。

OR 句による行の合計値

1 つのレポート行に複数存在するタグ値の合計には、FOR 句で OR 句を使用します。構文は次のとおりです。

```
FOR fieldname
value1 OR value2 [OR valuen...] [AS 'text'] [LABEL label] [OVER]
.
.
.
```

説明

fieldname

データソース内に存在するフィールド名です。

value1, value2, valuen

抽出され、合計が計算されるタグ値です。

AS 'text'

タグ値の組み合わせにタイトルを指定します。テキストは一重引用符 (') で囲みます。

label

RECAP 式で参照される行にラベルを指定します。ラベルの最大値は 66 バイトです。ブランクや特殊文字は使用できません。指定する明示的ラベルは、それぞれ一意である必要があります。

明示的ラベルを指定した場合でも、位置ラベル (例、R1、R2) は内部的に保持されます。

行内の値の合計

次のモデルでは、3 つのタグ値 (1010、1020、1030) の合計 (CASH) が計算されます。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 OR 1020 OR 1030  AS 'CASH'                    OVER
1100                  AS 'ACCOUNTS RECEIVABLE'     OVER
1200                  AS 'INVENTORY'
END
```

出力結果は次のとおりです。

```
                     AMOUNT
                     ------
CASH                 21,239
ACCOUNTS RECEIVABLE  18,829
INVENTORY            27,307
```

TO 句による値範囲の指定

1 つのレポート行に指定されたタグ範囲値の合計には、FOR 句で TO 句を使用します。構文は次のとおりです。

```
FOR fieldname
value1 TO value2 [AS 'text'] [LABEL label] [OVER]
```

説明

fieldname

データソース内に存在するフィールド名です。

value1

範囲の最下限を示すタグ値です。

TO

必須の句です。

value2

範囲の最上限を示すタグ値です。

AS 'text'

タグ値の組み合わせにタイトルを指定します。テキストは一重引用符 (') で囲みます。

label

RECAP 式で参照される行にラベルを指定します。ラベルの最大値は 66 バイトです。ブランクや特殊文字は使用できません。指定する明示的ラベルは、それぞれ一意である必要があります。

明示的ラベルを指定した場合でも、位置ラベル (例、R1、R2) は内部的に保持されます。

値の範囲の特定

LEDGER システムの CASH アカウントは、タグ 1010、1020、1030 により識別されるため、1010 から 1030 の範囲を指定することができます。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 TO 1030 AS 'CASH'
END
```

マスク文字によるタグ値の抽出

タグフィールドが文字フォーマットの場合、マスク文字を使用した検索が可能です。マスク文字にはドル記号 ($) を使用します。以下はその例です。

```
A$$D
```

A で開始し、D で終了する 4 バイトの値に一致する値を検索します。ドル記号で示された 2 バイトは、任意の文字を示します。これは、タグ値を 1 つずつ指定することなく、タグ値のグループを指定する場合に役立ちます。

マスク文字によるタググループの検索

次の例では、マスク表現 10$$ を指定することにより、「10」で開始する 4 バイトのアカウントに関連した金額でレポートの CASH 行が生成されます。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$$ AS 'CASH'                 OVER
1100 AS 'ACCOUNTS RECEIVABLE'  OVER
1200 AS 'INVENTORY'
END
```

出力結果は次のとおりです。

```
                     AMOUNT
                     ------
CASH                 21,239
ACCOUNTS RECEIVABLE  18,829
INVENTORY            27,307
```

複数行での同一 FOR フィールド値の使用

次の構文を FML リクエストの前または中に記述することにより、FOR フィールドのある値を複数行に使用することができます。この値は、値のみを指定、範囲の一部として使用、または計算に含めることができます。

```
SET FORMULTIPLE={ON|OFF}
```

または

```
ON TABLE SET FORMULTIPLE {ON|OFF}
```

説明

ON

FML リクエストの複数行で FOR フィールドの特定の値を参照することができます。

FORMULTIPLE を ON に設定すると、データソースから抽出した値は、タグが一致するレポート出力の各行に挿入されます。

OFF

1 つの値を複数行に指定することはできません。デフォルト値は OFF です。

FORMULTIPLE を OFF に設定すると、複数のタグは、その参照方法 (OR 句、TO 句、マスク文字) に関わらず、まずは完全に一致するタグまたは範囲の終了位置に対して評価され、次にマスク文字、最後に範囲内の値に対して評価されます。たとえば、値がまず完全一致として、次に範囲の一部として指定されている場合は、完全一致が表示されます。優先順位が同じ (例、完全一致と範囲の終了位置) であるタグを含む行が複数あり、これらの行に値が一致する場合の表示結果は予想できません。

詳細は、動的階層レポートを参照してください。

複数行での同一値の参照

次のリクエストは、アカウント 1010、1020、1030 のタグ値を抽出し、対応する値を個々に表示したリストを作成します。その後、同じ値を集計し、その合計値を TOTAL CASH として表示します。同様に、アカウント 1100 および 1200 のタグ値は詳細項目として表示され、その合計は TOTAL NON-CASH ASSETS として表示されます。

```
SET FORMULTIPLE=ON
TABLE FILE LEDGER
SUM AMOUNT
FOR ACCOUNT
1010 AS 'CASH ON HAND'                OVER
1020 AS 'DEMAND DEPOSITS'             OVER
1030 AS 'TIME DEPOSITS'               OVER
BAR                                   OVER
1010 OR 1020 OR 1030 AS 'TOTAL CASH'  OVER
" "                                   OVER
1100 AS 'ACCOUNTS RECEIVABLE'         OVER
1200 AS 'INVENTORY'                   OVER
BAR                                   OVER
1100 TO 1200 AS 'TOTAL NON-CASH ASSETS'
END
```

出力結果は次のとおりです。

```
                      AMOUNT
                      ------
```

```
CASH ON HAND            8,784
DEMAND DEPOSITS         4,494
TIME DEPOSITS           7,961
                       ------
TOTAL CASH             21,239

ACCOUNTS RECEIVABLE    18,829
INVENTORY              27,307
                       ------
TOTAL NON-CASH ASSETS  46,136
```

外部ファイルからのタグの使用

この例では、FML レポート行の値は、次のタグを含む外部ファイル (CASHSTUF) から抽出されます。

```
1010
1020
1030
```

次の TABLE リクエストは、外部ファイルから抽出したタグ値を使用し、アカウント 1010、1020、1030 の値の合計を FML レポートの CASH 行に表示します。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
(CASHSTUF)  AS 'CASH'                 OVER
1100        AS 'ACCOUNTS RECEIVABLE'
END
```

ファイル名は、括弧で囲む必要があります。

BY 句を使用した FML リクエスト

TABLE リクエストでは、FOR 句は 1 つのみ使用できます。ソート順を制御する BY 句を部分的に置き換えます。ただし、リクエストには BY 句を 32 個まで含めることができます。通常、BY 句は FML レポートに主 (外部) ソートフィールドを指定し、FOR 句は副 (内部) ソートフィールドを指定します。FOR 句を使用したリクエストでは、BY ROWS OVER 句は サポートされません。

BY 句と FOR 句を併用した FML リクエスト

次の例では、ACCOUNT (内部ソートフィールド) のレポート結果は、REGION (外部ソートフィールド) でソートされます。

```
DEFINE FILE REGION
CUR_YR=E_ACTUAL;
LAST_YR=.831*CUR_YR;
REGION/A4=IF E_ACTUAL NE 0 OR E_BUDGET NE 0 THEN 'EAST' ELSE 'WEST';
END

TABLE FILE REGION
HEADING CENTER
"CURRENT ASSETS FOR REGION <REGION"
" "
SUM CUR_YR LAST_YR
BY REGION NOPRINT
FOR ACCOUNT
10$$ AS 'CASH'                       OVER
1100 AS 'ACCOUNTS RECEIVABLE'        OVER
1200 AS 'INVENTORY'                  OVER
BAR                                  OVER
RECAP CUR_ASSET/I5C = R1 + R2 + R3;
END
```

出力結果は次のとおりです。

```
          CURRENT ASSETS FOR REGION EAST

                              CUR_YR         LAST_YR
                              ------         -------
CASH                        9,511.00        7,903.64
ACCOUNTS RECEIVABLE               .               .
INVENTORY                         .               .
                      --------------  --------------
CUR_ASSET                     9,511           7,903
```

ソートフィールド値は RECAP コマンドで使用することができます。これにより、このモデルでは主ソート区切りごとに異なる動作を実行することができます。たとえば、次のリクエストは、EAST 地域の 0 (ゼロ) 以外の値のみを計算します。

```
RECAP X=IF REGION EQ 'EAST' THEN .25*CASH ELSE 0;
AS 'AVAILABLE FOR DIVIDENDS'
```

詳細は、行間の計算実行を参照してください。

リクエストの直接データ入力

場合によっては、財務レポートを構築する際にいくつかの定数 (例、為替レート、インフレ率) を追加する必要があります。モデルのデータ値には、データソースから抽出する必要がないものもあります。FML を使用することにより、リクエストにデータを直接入力することができます。

リクエストの直接データ入力

```
DATA value,[..., value],$ [AS 'text'] [LABEL label] OVER
```

説明

value

入力する値を指定します。リストの値は、カンマ (,) を使用して区切る必要があります。リストは、カンマとドル記号 (,$) で終了する必要があります。

AS 'text'

データ行にタイトルを指定します。テキストは一重引用符 (') で囲みます。

この値を入力しないと、レポートの行タイトルはブランクになります。

label

RECAP 計算で使用する名前をデータ行に指定します。ラベルの最大値は 66 バイトです。ブランクや特殊文字は使用できません。指定する明示的ラベルは、それぞれ一意である必要があります。

リクエストの直接データ入力

次の例では、2 つの値 (.87 と 1.67) がそれぞれユーロとポンドの為替レートとして指定されています。

```
DEFINE FILE LEDGER
EUROS/I5C=AMOUNT;
POUNDS/I5C=3.2*AMOUNT;
END

TABLE FILE LEDGER
SUM EUROS AS 'EUROPE,DIVISION'
POUNDS AS 'ENGLISH,DIVISION'
FOR ACCOUNT
1010 AS 'CASH--LOCAL CURRENCY' LABEL CASH          OVER
DATA .87, 1.67 ,$ AS 'EXCHANGE RATE' LABEL EXCH    OVER
RECAP US_DOLLARS/I5C = CASH * EXCH;
END
```

指定された値は、レポートに指定された列数に到達するまで 1 列ごとに抽出されます。

出力結果は次のとおりです。

	EUROPE DIVISION	ENGLISH DIVISION
CASH--LOCAL CURRENCY	8,784	28,108
EXCHANGE RATE	.87	1.67
US_DOLLARS	7,642	46,940

行間の計算実行

RECAP コマンドにより、レポート行内のデータを計算し、新しい行を生成することができます。計算結果として出力される値の名前とフォーマット、および実行する計算式を指定する必要があります。RECAP 計算は行の間で実行されるため、計算の行をそれぞれ一意に識別する必要があります。デフォルト設定では、FML は R1、R2 などの行ラベルを使用して、出力結果を一意的に識別します。ただし、意味のあるラベル名を指定することもできます。詳細は、計算の行参照を参照してください。

行間の計算の定義

```
RECAP calcname[/format]=expression; [AS 'text']
```

説明

RECAP

必須のコマンド名です。1 行目に単独で指定する必要があります。

calcname

計算した値に指定される名前です。名前には 66 バイトまで使用できます。1 バイト目はアルファベットである必要があります。この名前は、明示的ラベルとしても使用されます。詳細は、計算の行参照を参照してください。

format

計算値の USAGE フォーマットです。長さは、列幅以内である必要があります。デフォルト値は、計算値が表示される列のフォーマットです。

expression

DEFINE コマンドとともに使用可能な任意の計算式です。ただし、IF... THEN ... ELSE 構文、関数は使用できますが、DECODE、EDIT、日付フォーマットのフィールドは使用できません。この式は、必要に応じて複数行に拡張することができます。式の末尾にはセミコロン (;) が必要です。詳細は、関数を使用した RECAP 計算および『ibi™ WebFOCUS® 関数リファレンス』を参照してください。

式には、デフォルトの FML 位置ラベル (R1、R2 など) を使用した特定の行への参照を含めることができます。また、柔軟な表記法を利用して行、列、セルを参照することもできます。Rn 参照は、モデル内で事前に評価された行に対してのみ使用することができます。詳細は、計算の行参照、計算の列参照およびセル参照を使用した計算を参照してください。

AS 'text'

デフォルトの行タイトルを変更します。デフォルト設定では、RECAP 値の名前が出力の行タイトルとして表示されます。AS 句は、デフォルト値を置き換えます。テキストは一重引用符 (') で囲みます。

RECAP の使用上の注意

- RECAP 式は、ラベル (明示またはデフォルト) を使用してモデルの他の行を参照します。RECAP 式で参照されたラベルは、リクエスト内にも指定されている必要があります。
- RECAP の結果に指定されたフォーマットは、フィールドのフォーマットよりも優先されます。以下はその例です。

  ```
  RECAP TOTVAL/D6.2S=IF R1 GT R4 THEN R4 ELSE R1;
  AS 'REDUCED VALUE'
  ```

  TOTVAL/D6.2S は、列内のデータフォーマットに関わらず、レポートの各列に整数部が 6 桁で小数第 2 位までの値を表示します。値が 0 (ゼロ) の場合はブランクを表示します。この機能は、整数列にパーセントを表示する場合に使用できます。
- 中間合計は、FML ではサポートされていません。
- RETYPE コマンドをサポートする環境において、RETYPE は、フィールドフォーマットの再定義された FML ラベルを認識しません。
- Rn 参照 (デフォルトの位置行ラベル) は、モデル内で事前に評価された行に対してのみ使用することができます。

計算の行参照

FML は、デフォルト位置ラベルを TAG、DATA、RECAP、PICKUP 行に指定します。これらの位置ラベルには、自動的に「R」が接頭語として使用されます。そのため、モデルの 1 行目は R1、2 行目は R2 となります。これらのラベルは、RECAP 式での行の参照に使用できます。

注意：下線、ブランク行、またはテキストが含まれた行は式で参照する必要がないため、これらのタイプの行にはデフォルトラベルは割り当てられません。

RECAP 式で行を参照するには、次のいずれかの方法を使用します。

- FML により指定された位置行ラベルの使用。
- 独自の明示的行ラベルの作成。

  注意：Rn 形式のラベル名は、FML が割り当てるデフォルト位置行ラベルに使用されるため、この形式で明示的なラベルを作成することはお勧めしません。この形式を使用

すると、後続の RECAP で問題が発生する場合があります。

- 位置ラベルと明示的行ラベルの使用。

明示的ラベルを指定した場合でも、位置ラベル (例、R1、R2) は内部的に保持されます。

等号 (=) の左側の計算値に割り当てられた名前をラベルとして使用できるため、RECAP 行に明示的ラベルは必要ありません。

ラベルは、RECAP 計算のほか、FML レポート行のフォーマットにも使用できます。詳細は、FML レポートのフォーマットを参照してください。

明示的行ラベル指定

```
rowtype [AS 'text'] LABEL label [OVER]
```

説明

rowtype

TAG、DATA、または PICKUP のいずれかの行タイプです。

AS 'text'

レポートの行名を変更します。テキストは一重引用符 (') で囲みます。

label

RECAP 式またはスタイルシートで参照するために行にラベルを指定します。ラベルの最大値は 66 バイトです。ブランクや特殊文字は使用できません。指定する明示的ラベルは、それぞれ一意である必要があります。

注意：Rn 形式のラベル名は、FML が割り当てるデフォルト位置行ラベルに使用されるため、この形式で明示的なラベルを作成することはお勧めしません。この形式を使用すると、後続の RECAP で問題が発生する場合があります。

明示的ラベルを指定した場合でも、位置ラベル (例、R1、R2) は内部的に保持されます。

RECAP 式の暗黙的行ラベル参照

次の例では、FML は、アカウント 1010 に暗黙的ラベル R1 を、アカウント 1020 に暗黙的ラベル R2 を、アカウント 1030 に暗黙的ラベル R3 を指定します。BAR 行にはラベルが指定されないため、RECAP 行には暗黙的ラベル R4 を指定します。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'      OVER
1020 AS 'DEMAND DEPOSITS'   OVER
1030 AS 'TIME DEPOSITS'     OVER
BAR                         OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
END
```

出力結果は次のとおりです。

```
                 AMOUNT
                 ------
CASH ON HAND      8,784
DEMAND DEPOSITS   4,494
TIME DEPOSITS     7,961
                 ------
TOTAL CASH       21,239
```

RECAP 式の明示的行ラベル参照

次のリクエストは、ラベル CA、AR、INV を RECAP 式で参照される 3 つのタグ行に指定します。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$$ AS 'CASH'                    LABEL CA    OVER
1100 AS 'ACCOUNTS RECEIVABLE'     LABEL AR    OVER
1200 AS 'INVENTORY'               LABEL INV   OVER
BAR                                           OVER
RECAP CURASST/I5C = CA + AR + INV;
END
```

出力結果は次のとおりです。

```
                     AMOUNT
                     ------
CASH                 21,239
ACCOUNTS RECEIVABLE  18,829
INVENTORY            27,307
                     ------
CURASST              67,375
```

この RECAP 値を後で参照するには、明示的ラベルである CURASST を指定します。

繰り返し行ラベルの使用

レポート内で同じ行を繰り返して指定する場合があります。次の例では、CASH というアカウントが財務分析の ASSETS ステートメントおよび CASH FLOW ステートメントに指定されています。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
"ASSETS"                          OVER
10$$ AS 'CASH' LABEL TOTCASH      OVER
.
.
"CASH FLOW"                       OVER
RECAP SAMECASH/I5C = TOTCASH; AS 'CASH'
END
```

2 回目に CASH 行を参照する際は、RECAP 計算 (新しい名前を指定) を使用し、ラベルを参照することができます。ラベルの参照には、CASH が最初に使用された行の明示的ラベル (TOTCASH) または暗黙的ラベル (R1) を使用できます。

ヒント：FORMULTIPLE パラメータが ON に設定されている場合は、別の名前を指定することなく行を繰り返し使用することができます。詳細は、複数レコードからの行作成を参照してください。

計算の列参照

FML レポートは、行と同様に、明示的ラベルを使用して列を参照することができます。列参照には、以下を使用することができます。

- 列番号。
- RECAP 式内での連続列表記。たとえば、(2,5) は列 2 から列 5 を示します。
- 列アドレス。
- 1 つおき、2 つおきなどを表す係数。
- 列参照番号の生成を制御する列表記。
- 列の値。

RECAP 式への列宣言の適用

次のリクエストは、4 行 × 3 列の FML マトリックスを生成します。

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END

TABLE FILE LEDGER
SUM CUR_YR AS 'CURRENT,YEAR'
   LAST_YR AS 'LAST,YEAR'
COMPUTE CHANGE/I5C = CUR_YR - LAST_YR;
FOR ACCOUNT
1010 AS 'CASH ON HAND'                  OVER
1020 AS 'DEMAND DEPOSITS'               OVER
1030 AS 'TIME DEPOSITS'                 OVER
BAR                                     OVER
RECAP TOTCASH/I5C = R1 + R2 + R3; AS 'TOTAL CASH'
END
```

レポートの列およびマトリックスのセルは、他の FML レポートで参照することができます。

出力結果は次のとおりです。

	CURRENT YEAR	LAST YEAR	CHANGE
CASH ON HAND	8,784	7,216	1,568
DEMAND DEPOSITS	4,494	3,483	1,011
TIME DEPOSITS	7,961	6,499	1,462
TOTAL CASH	21,239	17,198	4,041

たとえば、第 2 列の 3 行目を参照することにより、別の FML レポートでその値 (6499) を使用することができます。詳細は、セル参照を使用した計算を参照してください。

列番号参照を使用した計算

計算は、1 つの列に対して実行することも、指定した一連の列に対して実行することもできます。列を識別するには、ラベル名の後に列番号を指定します。列番号は括弧で囲みます。

RECAP 式の列番号参照

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END

TABLE FILE LEDGER
SUM CUR_YR AS 'CURRENT,YEAR'
LAST_YR AS 'LAST,YEAR'
FOR ACCOUNT
1010 AS 'CASH ON HAND'                                  OVER
1020 AS 'DEMAND DEPOSITS'                               OVER
1030 AS 'TIME DEPOSITS'                                 OVER
BAR                                                     OVER
RECAP TOTCASH/I5C = R1 + R2 + R3; AS 'TOTAL CASH'  OVER
" "                                                     OVER
RECAP GROCASH(2)/F5.2 = 100*TOTCASH(1)/TOTCASH(2) - 100;
AS 'CASH GROWTH(%)'
END
```

2 つ目の RECAP 式では、次のことに注目します。

- TOTCASH(1) は、列 1 の TOTAL CASH の値を参照します。
- TOTCASH(2) は、列 2 の TOTAL CASH の値を参照します。
- 計算結果は、CASH GROWTH(%) というラベルが付いた行の列 2 に表示されます。

RECAP 値は、指定した列に対してのみ計算されます。

出力結果は次のとおりです。

	CURRENT YEAR	LAST YEAR
CASH ON HAND	8,784	7,216
DEMAND DEPOSITS	4,494	3,483
TIME DEPOSITS	7,961	6,499
TOTAL CASH	21,239	17,198
CASH GROWTH(%)		23.50

データ検索の完了後、列が 1 つの場合はすべて一度に計算され、列が複数ある場合は 1 つずつ計算されます。

連続列参照を使用した計算

RECAP で連続する列群が必要な場合、最初と最後の列番号をカンマ (,) で区切ります。たとえば、DIFFERENCE (2,5) は、列 2 から列 5 までの結果を計算することを示します。

連続列の RECAP 計算

次の例では、ATOT に対する RECAP 計算は、リクエストに指定されたとおり、列 2 と 列 3 でのみ実行されます。列 1 に対しては実行されません。

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
NEXT_YR/I5C=1.13*CUR_YR + 222;
END

TABLE FILE LEDGER
SUM NEXT_YR CUR_YR LAST_YR
FOR ACCOUNT
10$$ AS 'CASH'                        OVER
1100 AS 'ACCOUNTS RECEIVABLE'         OVER
1200 AS 'INVENTORY'                   OVER
BAR                                   OVER
RECAP ATOT(2,3)/I5C = R1 + R2 + R3;
AS 'ASSETS--ACTUAL'
END
```

出力結果は次のとおりです。

	NEXT_YR	CUR_YR	LAST_YR
CASH	25,992	21,239	17,198
ACCOUNTS RECEIVABLE	21,941	18,829	15,954
INVENTORY	31,522	27,307	23,329
ASSETS--ACTUAL		67,375	56,481

列アドレス参照を使用した計算

各列ではなく、1 つまたは 2 つおきの列に対して計算を実行する場合は、係数または列アドレスを指定することができます。列アドレスは、列の各ソート値にデータフィールドが複数表示される場合に役立ちます。

RECAP 式の列アドレス使用

式の左辺のフォーマットは次のとおりです。

```
value(s,e,i)[/format]=
```

説明

value

RECAP 計算の結果に指定される名前です。

s

開始列です。

e

終了列です。アスタリスク (*) はすべての列を示します。

i

増分係数です。

format

計算値の USAGE フォーマットです。デフォルト値は、元の列のフィールドフォーマットです。

RECAP 式への列アドレスの適用

次のステートメントでは、月別に 2 列作成されます。

```
SUM ACTUAL AND FORECAST ACROSS MONTH
```

ACTUAL データのみで計算を実行する場合、次のような RECAP を使用して結果の配置を制御します。

```
RECAP calcname(1,*,2)=expression;
```

アスタリスク (*) は、奇数番号の付いたすべての列で RECAP を続行することを示します (列 1 から開始して、増分値 2 で最後まで実行)。

相対列アドレス参照を使用した計算

計算では、特定の列を基準として使用し、その列からの変位によって、その他すべての列を参照することができます。基準にする列の 1 つ左の列の変位は -1、1 つ右の列の変位は +1 です。以下はその例です。

```
COMP=FIX(*)-FIX(*-1);
```

この式では、ある期間から次の期間までの固定資産の変化を参照することができます。COMP=FIX(*) は、COMP=FIX と同等です。

前の列を参照する場合、その列はすでに抽出されているか、その値が 0 (ゼロ) である必要があります。

RECAP 式への相対列アドレス適用

次の例は、列 1 から列 2 への現金の変動 (CHGCASH) を計算します。

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
NEXT_YR/I5C=1.13*CUR_YR + 222;
END

TABLE FILE LEDGER
SUM NEXT_YR CUR_YR LAST_YR
FOR ACCOUNT
10$$ AS 'TOTAL CASH' LABEL TOTCASH           OVER
" "                                          OVER
RECAP CHGCASH(1,2)/I5SC = TOTCASH(*) - TOTCASH(*+1); AS 'CHANGE IN CASH'
END
```

出力結果は次のとおりです。

	NEXT_YR	CUR_YR	LAST_YR
TOTAL CASH	25,992	21,239	17,198
CHANGE IN CASH	4,753	4,041	

列参照番号生成の制御

CNOTATION パラメータは、リクエストに対して作成された内部マトリックスの各列に連続列番号を割り当てます。レポートで使用する列に対して、列参照番号の作成を制御するには、CNOTATION 列表記コマンドを使用します。

列番号は、内部マトリックスのフィールドを参照するため、データの取得および集計が完了した後で割り当てられます。フィールドが作成され、レポートに表示される場合、そのフィールドは内部マトリックスに保存されます。また、レポートに表示されないフィールドでも、内部マトリックスに保存される場合もあります。内部マトリックスに保存されるフィールドには、一時項目、フォーマットが再設定されたフィールド値、BY フィールド、NOPRINT オプションが指定されたフィールド、FORECAST または REGRESS などの RECAP 計算があります。デフォルト設定では、内部マトリックスのその他すべてのフィールドに列番号が割り当てられるようになっています。つまり、リクエストでフィールド値を正しく参照するには、内部的に作成されたすべてのフィールドを考慮する必要があります。

列参照番号のデフォルトの割り当てを変更するには、SET CNOTATION=PRINTONLY コマンドを使用します。このコマンドにより、レポート出力に表示されるフィールドのみに列番号が割り当てられます。このパラメータを COMPUTE および RECAP コマンドで使用すると、リクエスト内でそのフィールドを参照することができます。

作成する列番号の制御

```
SET CNOTATION={ALL|PRINTONLY|EXPLICIT}
```

説明

ALL

内部マトリックスのすべての列に列番号を割り当てます。デフォルト値は ALL です。

PRINTONLY

レポート出力に表示するフィールドのみに列番号を割り当てます。

EXPLICIT

表示または非表示に関係なく、リクエストで参照されるすべてのフィールドに列参照番号を割り当てます。

注意：CNOTATION は、ON TABLE 句ではサポートされません。

列値参照を使用した計算

レポートを ACROSS 句でソートすると、抽出されたすべての値は、適切な列に配置されます。各列には、ACROSS フィールド値の 1 つで構成されるタイトルが含まれます。RECAP 式でこの値を使用することにより、データ列全体を参照することができます。

RECAP 式の値による列参照

次のリクエストは、ACROSS フィールド (YEAR) の値に依存する係数を使用して、毎年の在庫コストを計算します。その後、資産の合計から毎年の在庫コストを減算して利益を計算します。

```
TABLE FILE LEDGER
SUM AMOUNT ACROSS YEAR
FOR ACCOUNT
10$$ AS 'CASH' LABEL CASH                          OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL RECEIVE        OVER
BAR                                                OVER
1200 AS 'INVENTORY VALUE' LABEL INVENT             OVER
RECAP INVENTORY_FACTOR/F5.2 = IF YEAR LT '1986'
   THEN 1.1 ELSE 1.25; AS 'INVENTORY COST FACTOR' OVER
RECAP INVENTORY_COST = INVENTORY_FACTOR * INVENT;
   AS 'INVENTORY COST'                             OVER
BAR                                                OVER
RECAP PROFIT = CASH + RECEIVE - INVENTORY_COST;
END
```

出力結果は次のとおりです。

	YEAR		
	1985	1986	1987
CASH	5,663	7,001	8,575
ACCOUNTS RECEIVABLE	5,295	6,250	7,284
INVENTORY VALUE	7,754	9,076	10,477
INVENTORY COST FACTOR	1.10	1.25	1.25
INVENTORY COST	8,529	11,345	13,096
PROFIT	2,429	1,906	2,763

行列参照を使用した計算

末尾の ACROSS 値の後に RECAP ステートメントを配置することにより、マトリックス内の複数の RECAP コマンドにスタイルを設定することができます。

マトリックス内の複数 RECAP ステートメントへのスタイル設定

```
TABLE FILE GGSALES
SUM UNITS
BY PRODUCT
ACROSS REGION
RECAP
TTL1/I8 = C1+C2+C3+C4;
TTL2/D12.2 = TTL1*1.25;

ON TABLE SET STYLE *
TYPE=DATA, COLUMN=TTL1 (*), COLOR=BLUE, BACKCOLOR=SILVER, STYLE=BOLD, $
TYPE=DATA, COLUMN=TTL2 (*), COLOR=RED, BACKCOLOR=AQUA, STYLE=BOLD, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	Region					
	Midwest	Northeast	Southeast	West	TTL1	TTL2
Product						
Biscotti	86105	145242	119594	70436	421377	526,721.25
Capuccino	.	44785	73264	71168	189217	236,521.25
Coffee Grinder	50393	40977	47083	48081	186534	233,167.50
Coffee Pot	47156	46185	49922	47432	190695	238,368.75
Croissant	139182	137394	156456	197022	630054	787,567.50
Espresso	101154	68127	68030	71675	308986	386,232.50
Latte	231623	222866	209654	213920	878063	1,097,578.75
Mug	86718	91497	88474	93881	360570	450,712.50
Scone	116127	70732	73779	72776	333414	416,767.50
Thermos	46587	48870	48976	45648	190081	237,601.25

セル参照を使用した計算

行と列の交点 (r, c) を表すセル表記フォーマットを使用して、行と列を参照することができます。

RECAP 式のセル表記による行列参照

式内で行と列を参照するには、次の表記を使用することができます。

```
E(r,c)
```

説明

E

必須定数です。

r

行番号です。

c

列番号です。現在の列は、アスタリスク (*) で示します。

RECAP 式のセル表記による 列参照

次のリクエストでは、2 つの RECAP 式で第 3 行 (PROFIT) の 4 つの列 (1、2、3、4) での値の減算を実行することにより、VARIANCE (EVAR および WVAR) の値を抽出します。これらの値の識別には、セル表記 (r,c) を使用します。

```
TABLE FILE REGION
SUM E_ACTUAL E_BUDGET W_ACTUAL W_BUDGET
FOR ACCOUNT
3000 AS 'SALES'                          OVER
3100 AS 'COST'                           OVER
BAR                                      OVER
RECAP PROFIT/I5C = R1 - R2;              OVER
" "                                      OVER
RECAP EVAR(1)/I5C = E(3,1) - E(3,2);
AS 'EAST--VARIANCE'                      OVER
RECAP WVAR(3)/I5C = E(3,3) - E(3,4);
AS 'WEST--VARIANCE'
END
```

出力結果は次のとおりです。

```
                E_ACTUAL  E_BUDGET  W_ACTUAL  W_BUDGET
                --------  --------  --------  --------
SALES              6,000     4,934     7,222     7,056
COST               4,650     3,760     5,697     5,410
                  ------    ------    ------    ------
PROFIT             1,350     1,174     1,525     1,646

EAST--VARIANCE       176
WEST--VARIANCE                          -121
```

注意：この例では、セル表記のほか、列番号の設定についても示しています。列 1 の EAST--VARIANCE と列 3 の WEST--VARIANCE の表示は、リクエストで括弧内に指定された番号 (EVAR (1) および WVAR (3)) によりそれぞれ制御されています。EVAR (1) and WVAR (3).

関数を使用した RECAP 計算

RECAP 行にユーザ独自の計算ルーチンを指定して、特殊計算を実行することができます。この機能は、計算が数学的に複雑である場合、または広範囲な参照テーブルが必要な場合に役立ちます。

ユーザが作成した関数は、FORTRAN、COBOL、PL/1、BAL などの呼び出しプロセスをサポートする言語で、サブルーチンとしてコード化されます。ユーザ定義関数の作成についての詳細は、『ibi™ WebFOCUS® 関数リファレンス』を参照してください。

RECAP コマンドでの関数呼び出し

```
RECAP calcname[(s,e,i)][/format]=function
(input1,...,inputn,'format2');
```

説明

calcname

計算した値に指定される名前です。

(s,e,i)

値を表示する列の開始値 (s)、終了値 (e)、増分値 (i) を指定します。これらの値を省略すると、値はすべての列に表示されます。

format

計算のフォーマットはオプションです。デフォルト値は、列のフィールドフォーマットです。計算がサブルーチンのみで構成される場合、サブルーチンの出力値 (format2) のフォーマットが計算フォーマットと一致していることを確認します。このフォーマットによる表示が列に収まらない場合、列にはアスタリスク (*) が表示されます。

function

8 バイト以内の関数名です。この名前には、行ラベルとは異なる値を指定する必要があります。以下の特殊文字を含めることはできません。

```
= -, / ()
```

input1, inputn

関数を呼び出す入力引数です。定数、文字、行列参照 (R 表記、E 表記、ラベル)、または他の RECAP 計算の名前を含めることができます。

関数に渡される値の番号とタイプが、関数にコード化された引数と一致することを確認します。

format2

戻り値のフォーマットです。一重引用符 (') で囲みます。

RECAP コマンドでの関数の呼び出し

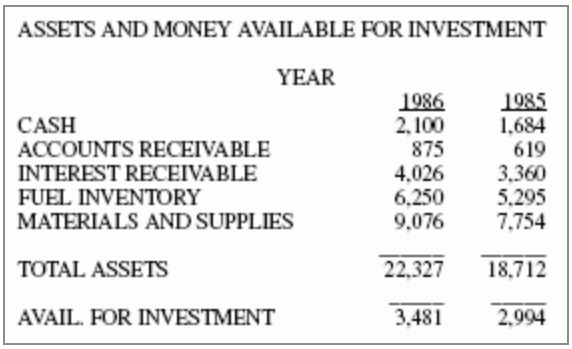
手元現金、総資産、および現在の日付を計算する INVEST という名称の関数が存在することを想定します (INVEST は付属のライブラリには含まれていません)。企業資産のアカウントを出力し、投資できる資金を計算するレポートを作成するには、まず INVEST 関数を呼び出すリクエストを作成します。

現在の日付は &YMD システム変数から取得されます。レポート内の日付を非表示にするには、NOPRINT オプションを設定します。日付は、次の RECAP ステートメントへの入力としてのみ使用されます。

以下は、リクエストの例です。

```
TABLE FILE LEDGER
HEADING CENTER
"ASSETS AND MONEY AVAILABLE FOR INVESTMENT </2"
SUM AMOUNT ACROSS HIGHEST YEAR
IF YEAR EQ 1985 OR 1986
FOR ACCOUNT
1010 AS 'CASH'                                  LABEL CASH      OVER
1020 AS 'ACCOUNTS RECEIVABLE'                   LABEL ACR       OVER
1030 AS 'INTEREST RECEIVABLE'                   LABEL ACI       OVER
1100 AS 'FUEL INVENTORY'                        LABEL FUEL      OVER
1200 AS 'MATERIALS AND SUPPLIES'                LABEL MAT       OVER
BAR                                                             OVER
RECAP TOTCAS = CASH+ACR+ACI+FUEL+MAT; AS 'TOTAL ASSETS'         OVER
BAR                                                             OVER
RECAP THISDATE/A8 = &YMD; NOPRINT                               OVER
RECAP INVAIL = INVEST(CASH,TOTCAS,THISDATE,'D12.2'); AS
          'AVAIL.FOR INVESTMENT'                              OVER
BAR AS '='
END
```

出力結果は次のとおりです。



ASSETS AND MONEY AVAILABLE FOR INVESTMENT

	YEAR	
	1986	1985
CASH	2,100	1,684
ACCOUNTS RECEIVABLE	875	619
INTEREST RECEIVABLE	4,026	3,360
FUEL INVENTORY	6,250	5,295
MATERIALS AND SUPPLIES	9,076	7,754
TOTAL ASSETS	22,327	18,712
AVAIL. FOR INVESTMENT	3,481	2,994

テキスト行の挿入

FML レポートの任意の位置にテキストを挿入するには、まずテキストを 1 行に入力し、これを二重引用符 (") で囲みます。レポートの外観を向上させるために、ブランク行を挿入することもできます。

さらに、テキスト行にデータ変数ラベルを指定することにより、FML レポートで開発したデータをテキスト行に含めることもできます。

テキストの挿入

次の例では、3 行のテキスト (ブランク 1 行、テキスト 2 行) が挿入されます。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
" --- CASH ACCOUNTS ---"              OVER
1010 AS 'CASH ON HAND'                OVER
1020 AS 'DEMAND DEPOSITS'             OVER
1030 AS 'TIME DEPOSITS'               OVER
" "                                   OVER
" --- OTHER CURRENT ASSETS ---"       OVER
1100 AS 'ACCOUNTS RECEIVABLE'         OVER
1200 AS 'INVENTORY'
END
```

出力結果は次のとおりです。

```
                      AMOUNT
                      ------
 --- CASH ACCOUNTS ---
```

```
CASH ON HAND          8,784
DEMAND DEPOSITS       4,494
TIME DEPOSITS         7,961

 --- OTHER CURRENT ASSETS ---
ACCOUNTS RECEIVABLE  18,829
INVENTORY           27,307
```

ブランク行を作成するには、リクエストの 1 行に 1 つのブランクを二重引用符 (") で囲んで入力します。

テキスト行へのデータ変数の挿入

```
"text <label[(c)][>]"
```

説明

<

ラベルを囲むために必須の左キャレット (<) です。

label

明示的または暗黙的な行ラベルです (RECAP では、計算された値がラベルとして使用されます)。

c

セルの列番号を示すオプションの識別子です。ただし、レポートに複数の列が存在する場合は、この識別子が常に必要です。使用する場合は、識別子を括弧で囲みます。

>

配置を明確にするために使用するオプションの右キャレット (>) です。

テキスト行へのデータ変数の挿入

次の例では、RECAP 値である CURASST は、NOPRINT コマンドにより非表示になりますが、テキスト行にデータ変数として挿入されます。

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$$ AS 'Cash'                    LABEL CA    OVER
1100 AS 'Accounts Receivable'     LABEL AR    OVER
1200 AS 'Inventory'               LABEL INV   OVER
RECAP CURASST/I5C = CA + AR + INV; NOPRINT   OVER
"Current Assets: <CURASST"
END
```

出力結果は次のとおりです。

```
                      AMOUNT
                      ------
Cash                  21,239
Accounts Receivable   18,829
Inventory             27,307
Current Assets:67,375
```

FML レポートへのフィールドの追加

リクエストは、レポート内のフィールド数を制御します。たとえば、リクエストに表示コマンド SUM AMOUNT AND FORECAST が含まれる場合、そのレポートには、AMOUNT および FORECAST という 2 つのフィールドが含められます。

FML リクエストにフィールドを追加するには、TABLE リクエストと同様に、COMPUTE コマンドを使用して一時項目を作成するか、ブランク、フィールドタイトル、およびフィールドフォーマットを指定します。

FML レポートへのフィールドの追加

次の例では、COMPUTE コマンドを使用して一時項目 CHANGE を生成し、FML レポートに新しいフィールドとして表示します。次のリクエストは、4 行 × 3 列の FML マトリックスを生成します。

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END
```

```
TABLE FILE LEDGER
SUM CUR_YR AS 'CURRENT,YEAR'
   LAST_YR AS 'LAST,YEAR'
COMPUTE CHANGE/I5C = CUR_YR - LAST_YR;
FOR ACCOUNT
1010 AS 'CASH ON HAND'                  OVER
1020 AS 'DEMAND DEPOSITS'               OVER
1030 AS 'TIME DEPOSITS'                 OVER
BAR                                     OVER
RECAP TOTCASH/I5C = R1 + R2 + R3; AS 'TOTAL CASH'
END
```

出力結果は次のとおりです。

	CURRENT YEAR	LAST YEAR	CHANGE
CASH ON HAND	8,784	7,216	1,568
DEMAND DEPOSITS	4,494	3,483	1,011
TIME DEPOSITS	7,961	6,499	1,462
TOTAL CASH	21,239	17,198	4,041

注意：指定した計算は、レポートの各タグ (RECAP 行) で実行されます。ただし、RECAP 行は計算を変更する場合があります。

新しい期間の列の追加

次のリクエストは、レポートに将来の期間 (ここでは 1999) を追加します。

```
DEFINE FILE LEDGER
CUR_YR/P5C=AMOUNT;
LAST_YR/P5C=.87*AMOUNT - 142;
END

TABLE FILE LEDGER
SUM AMOUNT
ACROSS YEAR AND COMPUTE 1999/P5C = 2.5*AMOUNT;
FOR ACCOUNT
1010 AS 'CASH ON HAND'                    OVER
1020 AS 'DEMAND DEPOSITS'                 OVER
1030 AS 'TIME DEPOSITS'                   OVER
BAR                                       OVER
RECAP TOTCASH/P5C = R1 + R2 + R3; AS 'TOTAL CASH' OVER
RECAP CHANGE(2,*) = TOTCASH(*) - TOTCASH(*-1);
END
```

出力結果は次のとおりです。

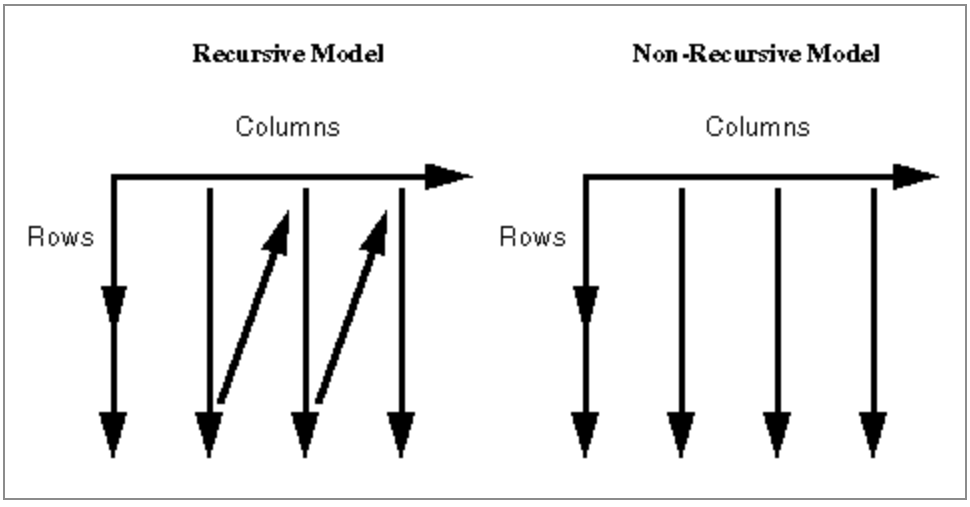
	YEAR			
	1985	1986	1987	1999
CASH ON HAND	1,684	2,100	5,000	4,210
DEMAND DEPOSITS	619	875	3,000	1,548
TIME DEPOSITS	3,360	4,026	575	8,400
TOTAL CASH	5,663	7,001	8,575	14,158
CHANGE		1,338	1,574	5,583

再帰的モデルの作成

異なる期間を含むモデルでは、ある期間の終了値を次の期間の開始値として使用する場合がよくあります。このような状況を記述する計算には、次の 2 つの特性があります。

- 1 つ以上の RECAP 行のラベルは、他行の複製である。これらは、特定の値を再計算するために、繰り返し使用されます。
- その時点ではまだ記述されていない、後でモデルに指定されるラベルを参照することができる。必要なラベルが最後までモデルに存在しない場合は、メッセージが表示されます。

再帰的モデルでは、フィールドを 1 つずつ順番に生成する必要があります。非再帰的モデルでは、フィールドをすべて同時に生成することができます。下図は、これを図式化したものです。



FML では、2 つのモデル条件のいずれかで切り替えが必要なこと (異なる行によるラベルの再利用、または計算へのラベル参照の転送) が認識された直後、順序が自動的に切り替えられます。

再帰的モデルの作成

次の例は、再帰的モデルを表します。ある年の ENDCASH が翌年の STARTING CASH となることが分かります。

```
DEFINE FILE REGION
CUR_YR=E_ACTUAL;
LAST_YR=.831*CUR_YR;
NEXT_YR=1.2297*CUR_YR;
END

TABLE FILE REGION
SUM LAST_YR CUR_YR NEXT_YR
FOR ACCOUNT
10$$ AS 'STARTING CASH' LABEL STCASH        OVER
RECAP STCASH(2,*) = ENDCASH(*-1);           OVER
" "                                         OVER
3000 AS 'SALES' LABEL SLS                   OVER
3100 AS 'COST' LABEL COST                   OVER
BAR                                         OVER
RECAP PROFIT/I5C = SLS - COST;              OVER
" "                                         OVER
RECAP ENDCASH/I5C = STCASH + PROFIT;
END
```

出力結果は次のとおりです。

	LAST_YR	CUR_YR	NEXT_YR
STARTING CASH	7,903.64	9,024.00	10,374.00
SALES	4,986.00	6,000.00	7,378.20
COST	3,864.15	4,650.00	5,718.11
PROFIT	1,121	1,350	1,660
ENDCASH	9,024	10,374	12,034

動的階層レポート

フィールド間の階層関係はマスターファイルに定義され、FML を使用して自動的に表示されます。親フィールドと子フィールドがデータ値を共有し、その関係が階層関係である必要があります。親フィールドと子フィールドのフォーマットは、いずれも数値または文字である必要があります。

たとえば、以下を想定します。

- 従業員のデータソースに、従業員の ID とその従業員のマネージャの ID の両方が格納されている。

  または

- 総勘定元帳データソースに、勘定科目番号フィールドと勘定科目の親フィールドの両方が格納されている。

これらのフィールドを検証することにより、全組織図または勘定科目構造図 (チャート) を作成することができます。ただし、従来の FML レポートにチャートを出力するには、リクエスト構文に従業員の ID またはアカウント番号をレポートでの表示順に指定する必要があります。従業員またはアカウントが追加、削除、または転送される場合、変更を組織構造に反映するためにはリクエストを変更する必要があります。以下はその例です。

```
TABLE FILE EMPLOYEE
PRINT DEPARTMENT CURR_JOBCODE
FOR EMP_ID
999999999   OVER
222222222   OVER
 .
 .
 .
```

それに対して、マスターファイルの 2 フィールド間の階層的な関係は FML 階層により定義できます。また、この情報をメモリにロードすることもできます。その後、FML は子の関係を表す行を動的に作成し、レポートに表示します。構造は、階層の任意の点で開始することができます。この例で示された EMP_ID は「階層フィールド」と呼ばれます。

FML 階層の要件

1. 2 つのフィールド間に階層関係を定義するには、マスターファイルで PROPERTY=PARENT_OF および REFERENCE=hierarchyfld 属性を指定します。

   階層はメモリにロードする必要があります。ロードされた階層は、「チャート」と呼ばれます。階層がマスターファイルに定義され、FML リクエストにより参照されると、階層は自動的にロードされます。FML リクエストで参照されていないマスターファイル、または FML リクエストで参照されているマスターファイルに結合しているマスターファイルに定義された階層を使用するには、FML リクエストの発行前に LOAD CHART コマンドを発行します。

   チャートは 16 個までロードできます。チャートは、セッションの終了時に自動的にアンロードされます。

2. 階層の特定の位置で開始する階層データを抽出するには、FML リクエストの FOR 句で GET/WITH CHILDREN または ADD 句を使用します。

FML 階層を使用する場合、FOR フィールドは次のいずれかに指定する必要があります。

- 階層フィールド

  または

- 階層フィールドを含むユニークセグメントに結合された JOIN フィールド。この場合、JOIN フィールドを階層フィールドにする必要があります。結合が一意であるための条件は、階層がクロスリファレンスセグメントで定義されている場合のみに適用されます。

FOR フィールドが、親子階層内に存在するか、親子階層にリンクしている必要があります。親子階層にリンクしている場合は、階層フィールドを含む取引データを、その階層定義を含む他のデータソースに結合することができます。

FML リクエストと同様に、タグ値のファイルにデータが存在しない場合にも、タグ付きの行はミッシングデータを表すピリオド (.) を伴って表示されます。これは、タグ付きの行の定義に WHEN EXISTS 句を追加することにより上書きすることができます。WHEN EXISTS により、行はタグにデータが存在する場合にのみ表示されます。

注意：HTML 出力に階層インデントを保持するには、SHOWBLANKS=ON 設定を有効にしておく必要があります。

マスターファイルでの階層の定義

CENTGL マスターファイルには、勘定科目表の階層が格納されています。GL_ACCOUNT_PARENT は、階層の親フィールドです。GL_ACCOUNT フィールドは、階層フィールドです。GL_ACCOUNT_CAPTION フィールドは、階層フィールドの説明キャプションとして使用することができます。

```
FILE=CENTGL     ,SUFFIX=FOC
SEGNAME=ACCOUNTS,SEGTYPE=S01
FIELDNAME=GL_ACCOUNT,           ALIAS=GLACCT,  FORMAT=A7,
          TITLE='Ledger,Account', FIELDTYPE=I, $
FIELDNAME=GL_ACCOUNT_PARENT,    ALIAS=GLPAR,   FORMAT=A7,
          TITLE=Parent,
          PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT, $
FIELDNAME=GL_ACCOUNT_TYPE,      ALIAS=GLTYPE,  FORMAT=A1,
          TITLE=Type,$
FIELDNAME=GL_ROLLUP_OP,         ALIAS=GLROLL,  FORMAT=A1,
          TITLE=Op, $
FIELDNAME=GL_ACCOUNT_LEVEL,     ALIAS=GLLEVEL, FORMAT=I3,
```

```
          TITLE=Lev, $
FIELDNAME=GL_ACCOUNT_CAPTION,   ALIAS=GLCAP,   FORMAT=A30,
          TITLE=Caption,
          PROPERTY=CAPTION, REFERENCE=GL_ACCOUNT, $
FIELDNAME=SYS_ACCOUNT,          ALIAS=ALINE,   FORMAT=A6,
          TITLE='System,Account,Line', MISSING=ON, $
```

CENTSYSF データソースには、詳細レベルの財務データが格納されています。これは、Century Corporation という架空会社で使用する未連結の財務データです。このデータソースでは、あたかも外部の会計システムからデータを取得するように、CENTGL データベースから分離して処理を行えるように設計されています。このデータソースは、異なる勘定科目システム (SYS_ACCOUNT) を使用しますが、CENTGL の SYS_ACCOUNT フィールドと結合することができます。このデータは、通例の記号 (費用はプラス、収益はマイナス) を使用しています。

```
FILE=CENTSYSF     ,SUFFIX=FOC
SEGNAME=RAWDATA   ,SEGTYPE=S2
FIELDNAME=SYS_ACCOUNT   ,  ,A6       , FIELDTYPE=I,
          TITLE='System,Account,Line', $
FIELDNAME=PERIOD        ,  ,YYM      , FIELDTYPE=I, $
FIELDNAME=NAT_AMOUNT    ,  ,D10.0    , TITLE='Month,Actual', $
FIELDNAME=NAT_BUDGET    ,  ,D10.0    , TITLE='Month,Budget', $
FIELDNAME=NAT_YTDAMT    ,  ,D12.0    , TITLE='YTD,Actual', $
```

FML 階層の表示

GET CHILDREN および WITH CHILDREN コマンドは、FML レポートの階層データを動的に抽出、表示します。GET CHILDREN は、子のみを表示し、コマンド内で参照した親値は表示しません。WITH CHILDREN は、親とその子を順に表示します。

FML 階層の表示

```
TABLE FILE filename{PRINT|SUM} ...
FOR hierarchyfld
parentvalue {GET|WITH} CHILD[REN] [n|ALL]
 [AS CAPTION|'text'] [LABEL label]
.
```

```
.
.
END
```

説明

filename

FML リクエストで使用されるファイル名です。このリクエストの階層が自動的にロードされない場合は、LOAD CHART コマンドの発行時にすでにロードされていることが考えられます。

hierarchyfld

階層フィールド名です。リクエストが JOIN 構造を参照する場合、その名前はホストファイルのフィールド名である必要があります。エイリアスはサポートされません。

parentvalue

子を抽出するための親値です。

GET CHILDREN

指定した parentvalue の最初の子から開始する階層を表示します。表示には親は含まれません。これは、FML 構文 CHILD1 OVER CHILD2 OVER... に相当します。

WITH CHILDREN

指定した parentvalue から開始する階層を表示します。表示には親が含まれます。これは、FML 構文 parentvalue OVER CHILD1 OVER CHILD2 OVER ... に相当します。

n|ALL

1 から 99 の正の整数で、表示する階層レベル数を指定します。100 以上の値が指定された場合は警告メッセージが表示され、n は 99 に設定されます。デフォルト値は 1 です。n が省略されると、直系の子のみが表示されます。GET CHILDREN 2 または WITH CHILDREN 2 は、直系の子と孫を表示します。GET CHILDREN 99 または WITH CHILDREN 99 は、99 レベルまでの子孫を表示します。ALL は 99 と同義です。各子インスタンスは、その次の子インスタンスの前に出力されます。階層フィールドのレベルが連続している場合、次のレベルは前のレベルからブランク 2 個分でインデントされます。

CAPTION

表示するキャプション値は、マスターファイルに CAPTION として定義されたフィールドから抽出されることを示します。

AS CAPTION 句は、GET/WITH CHILDREN または ADD 構文を使用しない場合でも、タグ付き行でサポートされます。ただし、キャプション値をロード、表示するには、階層を定義 (PARENT_OF 属性を指定) する必要があります。階層が定義されていない場合、AS CAPTION 句は無視されます。

'text'

階層フィールド値の行タイトルとして使用するテキストです。マスターファイルに定義された CAPTION フィールドは、レポート出力のキャプションとしては使用されません。

label

明示的な行ラベルです。生成された行には、それぞれ指定したラベルが割り当てられます。

注意：階層のソートは、まず親フィールド、次に親フィールドの階層フィールドにより行われます。

FML 階層の表示

次のリクエストは、3000 で始まる 2 階層のアカウント番号を表示します。

```
SET SHOWBLANKS=ON
TABLE FILE CENTGL
PRINT GL_ACCOUNT_PARENT
FOR GL_ACCOUNT
3000 WITH CHILDREN 2
END
```

出力結果は次のとおりです。

```
             Parent
             ------
3000         1000
  3100       3000
    3110     3100
    3120     3100
    3130     3100
    3140     3100
    3200     3000
    3300     3200
    3400     3200
    3500     3200
    3600     3200
```

```
    3700     3200
    3800     3200
    3900     3200
```

注意

- リクエストに WITH CHILDREN ではなく GET CHILDREN を指定すると、親値 (3000) の行はレポート出力に表示されません。
- HTML 出力にインデントを保持するには、SET SHOWBLANKS=ON コマンドを有効にしておく必要があります。

FML 階層のキャプション表示

次のリクエストは、アカウントの 2 階層チャートを表示します。1000 (階層の最上位) で開始し、アカウント番号ではなくキャプションフィールド値を表示します。

```
SET SHOWBLANKS=ON
TABLE FILE CENTGL
PRINT GL_ACCOUNT_PARENT
FOR GL_ACCOUNT
1000 WITH CHILDREN 2 AS CAPTION
END
```

出力結果は次のとおりです。

```
                                  Parent
                                  ------
Profit Before Tax
  Gross Margin                    1000
    Sales Revenue                 2000
    Cost Of Goods Sold            2000
  Total Operating Expenses        1000
    Selling Expenses              3000
    General + Admin Expenses      3000
  Total R+D Costs                 1000
    Salaries                      5000
    Misc.Equipment               5000
```

注意：リクエストに WITH CHILDREN ではなく GET CHILDREN を指定すると、親値 (1000、Profit Before Tax) の行はレポート出力に表示されません。

FML 階層の統合

ADD コマンドは、FML レポート出力の 1 行に含まれる複数の階層レベルを統合します。ADD は単独で使用することも、GET CHILDREN または WITH CHILDREN とともに使用することもできます。ADD は、SUM コマンドを含むリクエストに使用することができます。また、統合されたデータではなく、詳細レベルのデータに使用することもできます。

ADD を単独で使用する場合、レポート出力の 1 行に含まれる親と子を集計して、その行の数値データを統合します。これは、FML 構文 parentvalue OVER CHILD1 OVER CHILD2OVER ... に相当します。

ADD を GET CHILDREN と併用すると、指定した親値の子が 1 行ずつ表示されます。各行は、各子とその子孫の合計値です。表示する子のレベル数 (レポート出力に生成される行数の決定)、および計算量を指定することができます。デフォルト設定では、レポート出力には直系の子のみに行が指定され、各子の合計にはその子孫がすべて含まれます。

ADD を WITH CHILDREN と併用すると、まず親とそのすべての子孫の値の合計がレポート出力の行として表示されます。その後、GET CHILDREN ADD により表示された行と同じ行が続きます。

FML レポートの複数行でデータレコードを使用する (例、詳細および合計行の表示、複数レベルの詳細データの統合) には、次の設定が必要です。

```
SET FORMULTIPLE=ON
```

FML 階層の概要行の作成

```
TABLE FILE filenameSUM ...
FOR hierarchyfld
parentvalue ADD [n|ALL]
 [AS CAPTION|'text'] [LABEL label]
.
.
.
END
```

説明

filename

FML リクエストで使用されるファイル名です。このリクエストの階層が自動的にロードされない場合は、LOAD CHART コマンドの発行時にすでにロードされていることが考えられます。

hierarchyfld

階層フィールド名です。リクエストが JOIN 構造を参照する場合、その名前はホストファイルのフィールド名である必要があります。エイリアスはサポートされません。

parentvalue

階層での集計の開始位置を決定する親値です。

ADD

1 行に親と *n* レベルまでの子孫を表示し、その行に表示された数値データの合計を計算します。これは、FML 構文 parentvalue OVER CHILD1 OVER CHILD2 OVER ... に相当します。

子の合計のみを表示するには、親行と合計行を表示し、RECAP を使用して合計行と親行の減算を実行する必要があります。以下はその例です。

```
FOR ...
parentvalue                 OVER
parentvalue ADD 1           OVER
RECAP CHILDSUM = R2-R1;
```

n|ALL

1 から 99 までの正の整数で、集計する階層レベル数を指定します。デフォルト値は ALL です。*n* が省略されると、すべての子の値が合計に含められます。*n* が 1 の場合、直系の子のみが含まれます。*n* が 2 の場合、直系の子と孫が含まれます。ADD 99 は、99 レベルまでの子孫を集計します。ALL は 99 と同義です。

CAPTION

親値のキャプションが合計行を表示することを示します。

AS CAPTION 句は、GET CHILDREN または ADD 構文を使用しない場合も、タグ付き行でサポートされます。ただし、キャプション値をロード、表示するには、階層を定義 (PARENT_OF 属性を指定) する必要があります。階層が定義されていない場合、AS CAPTION 句は無視されます。

'text'

集計行のタイトルとして使用するテキストです。マスターファイルに定義された CAPTION フィールドは、レポート出力のキャプションとしては使用されません。

label

明示的な行ラベルです。生成された行には、それぞれ指定したラベルが割り当てられます。

FML 階層の概要行の表示

CENTSYSF データソースには、詳細レベルの財務データが格納されています。CENTGL データソースのアカウント階層をこの財務データとともに使用するには、2 つのデータソースを結合します。CENTSYSF 内のデータは、自然記号とともに格納されます。これは、収益と負債が負の値で格納されることを意味します。このリクエストで使用される階層の一部には、正のデータのみが含まれます。

階層はホストセグメントに定義されているため、結合はユニークである必要はないことに注意してください。

まず、WITH CHILDREN コマンドにより、Selling Expenses (3100) で開始する階層行が表示されます。詳細レベルのデータソースでは、子を持たないアカウントのみにデータが入力されます。次に、ADD コマンドにより、アカウント 3100 とその子孫の合計値を表示する行が 1 つ作成されます。

```
SET SHOWBLANKS=ON
SET FORMULTIPLE=ON
JOIN SYS_ACCOUNT IN CENTGL TO ALL SYS_ACCOUNT IN CENTSYSF
TABLE FILE CENTGL
SUM NAT_AMOUNT/D10.0 NAT_YTDAMT/D10.0
FOR GL_ACCOUNT
3100 WITH CHILDREN ALL AS CAPTION OVER
BAR                                OVER
3100 ADD AS CAPTION
IF PERIOD EQ '2002/03'
END
```

出力結果は次のとおりです。

	Month Actual	YTD Actual
Selling Expenses	.	.
Advertising	.	.
TV/Radio	1,049,146.	2,954,342.
Print Media	244,589.	721,448.
Internet Advertising	9,542.	29,578.
Promotional Expenses	53,719.	151,732.
Joint Marketing	97,135.	289,799.
Bonuses/Commisions	100,188.	304,199.
Selling Expenses	1,554,319.	4,451,098.

FML 階層データの任意のレベルおよび層への統合

```
TABLE FILE filename
SUM ...
FOR hierarchyfld
parentvalue {GET|WITH} CHILD[REN] [n|ALL] ADD [m|ALL]
 [AS CAPTION|'text'] [LABEL label]
.
.
.
END
```

説明

filename

FML リクエストで使用されるファイルの名前です。このリクエストの階層が自動的にロードされない場合、LOAD CHART コマンドの発行により、事前にロードされています。

hierarchyfld

階層フィールド名です。リクエストが JOIN 構造を参照する場合、その名前はホストファイルのフィールド名である必要があります。エイリアスはサポートされません。

parentvalue

階層での集計の開始位置を決定する親値です。

GET|WITH

GET は、レポートで生成された最初の行が親値の最初の子の統合行であることを指定します。WITH は、レポートで生成された最初の行が親値の統合行で、その後に *n* で指定したレベルまでの各子とその子孫の統合行が続くことを指定します。

n|ALL

1 から 99 の正の整数で、表示する子のレベル数を指定します。各子の出力行には、その子と ADD オプションで指定されたレベルまでの子孫の合計が含まれます。デフォルト値は 1 です。*n* が省略されると、レポートには直系の子に対して 1 行ずつ表示されます。*n* が 2 の場合、レポートには直系の子と孫に対して 1 行ずつ表示されます。ALL は 99 と同義です。

ADD

GET CHILDREN または WITH CHILDREN コマンドにより生成された各行の *m* で指定されたレベルまで階層の合計が計算されます。

m|ALL

1 から 99 までの正の整数です。レポート出力の各行に統合する子のレベル数を指定します。100 以上の値が指定された場合、警告メッセージが表示され、*m* は 99 に設定されます。デフォルト値は ALL です。*m* が省略されると、統合行はすべての子の合計を計算します。*m* が 2 の場合、直系の子と孫のみがレポート出力のそれぞれの行に統合されます。ADD 99 は、99 レベルまでの子孫を集計します。ALL は 99 と同義です。

CAPTION

親値のキャプションが合計行を表示することを示します。

AS CAPTION 句は、GET CHILDREN または ADD 構文を使用しない場合も、タグ付き行でサポートされます。ただし、キャプション値をロード、表示するには、階層を定義 (PARENT_OF 属性を指定) する必要があります。階層が定義されていない場合、AS CAPTION 句は無視されます。

'text'

集計行のタイトルとして使用するテキストです。マスターファイルに定義された CAPTION フィールドは、レポート出力のキャプションとしては使用されません。

label

明示的な行ラベルです。生成された行には、それぞれ指定したラベルが割り当てられます。

FML 階層データの統合

次のリクエストは、1 つ目の WITH CHILD コマンドを使用して、アカウント 3100 で開始する階層の詳細データを表示します。2 つ目の WITH CHILD コマンドにより、親アカウント (3100) とその直系の子との統合行が作成されます。

```
SET SHOWBLANKS=ON
SET FORMULTIPLE=ON
JOIN SYS_ACCOUNT IN CENTGL TO ALL SYS_ACCOUNT IN CENTSYSF
TABLE FILE CENTGL
SUM NAT_AMOUNT/D10.0 NAT_YTDAMT/D10.0
FOR GL_ACCOUNT
3100 WITH CHILDREN ALL AS CAPTION      OVER
" "                                    OVER
BAR AS =                               OVER
" "                                    OVER
3100 WITH CHILDREN ADD AS CAPTION
IF PERIOD EQ '2002/03'
END
```

階層はホストセグメントに定義されているため、結合はユニークである必要はないことに注意してください。

次のレポート出力の上部は詳細レベルのデータを示し、下部は、集計データを表示しています。下部は、集計データを表示しています。レポートの連結データには次の行が表示されます。

- 親の値を表す行が表示されます。この行には、すべてのレベルのすべての子を合計した値が含まれます。
- 勘定科目 Selling Expenses (3100) の直系の子の値を表す行がそれぞれ 1 行表示されます。これらの行には、Advertising、Promotional Expenses、Joint Marketing、Bonuses/Commisions があります。
- Advertising の行には、そのすべての子を合計した値が含まれます。子が複数レベル存在する場合は、すべての値が合計に追加されます。3100 の他の直系の子には子孫が存在しないため、各行の合計は親値のみで構成されます。

	Month Actual	YTD Actual
Selling Expenses	.	.
Advertising	.	.
TV/Radio	1,049,146.	2,954,342.
Print Media	244,589.	721,448.
Internet Advertising	9,542.	29,578.
Promotional Expenses	53,719.	151,732.
Joint Marketing	97,135.	289,799.
Bonuses/Commisions	100,188.	304,199.
Selling Expenses	1,554,319.	4,451,098.
Advertising	1,303,277.	3,705,368.
Promotional Expenses	53,719.	151,732.
Joint Marketing	97,135.	289,799.
Bonuses/Commisions	100,188.	304,199.

WITH CHILDREN ではなく GET CHILDREN を使用すると、最上行が出力の各部分から削除されます。残りの行は変更されません。

	Month Actual	YTD Actual
Advertising	.	.
TV/Radio	1,049,146.	2,954,342.
Print Media	244,589.	721,448.
Internet Advertising	9,542.	29,578.
Promotional Expenses	53,719.	151,732.
Joint Marketing	97,135.	289,799.
Bonuses/Commisions	100,188.	304,199.
Advertising	1,303,277.	3,705,368.
Promotional Expenses	53,719.	151,732.
Joint Marketing	97,135.	289,799.
Bonuses/Commisions	100,188.	304,199.

次のリクエストは、アカウント 2000 およびその直系の子と孫を統合した行を表示します。

```
SET SHOWBLANKS=ON
SET FORMULTIPLE=ON
JOIN SYS_ACCOUNT IN CENTGL TO ALL SYS_ACCOUNT IN CENTSYSF
TABLE FILE CENTGL
SUM NAT_AMOUNT/D10.0 NAT_YTDAMT/D10.0
FOR GL_ACCOUNT
2000 WITH CHILDREN 2 ADD AS CAPTION
IF PERIOD EQ '2002/03'
END
```

出力結果は次のとおりです。

	Month Actual	YTD Actual
Gross Margin	-4,513,659.	-13,080,549.
Sales Revenue	-10,398,305.	-30,877,546.
Retail Sales	-8,237,253.	-24,539,197.
Mail Order Sales	-1,138,414.	-3,403,387.
Internet Sales	-1,022,638.	-2,934,962.
Cost Of Goods Sold	5,884,646.	17,796,997.
Variable Material Costs	4,415,560.	13,410,629.
Direct Labor	961,143.	2,920,449.
Fixed Costs	507,943.	1,465,919.

階層の手動ロード

ほとんどの場合、階層はリクエスト構文の結果として自動的にロードされます。ただし、1 つのマスターファイルに定義した階層を、階層ファイルに結合していないデータソース (同じ階層フィールドを含む) に対して使用する必要がある場合、LOAD CHART コマンドを使用して階層データを手動でロードすることができます。

ロード可能なチャート数の上限は、利用可能なメモリにより決定されます。チャートは、セッションの終了時に自動的にアンロードされます。

チャートをロードするには、親の値およびその値に関連する子のリストを生成する TABLE リクエストを実行します。

```
TABLE FILE chartfile
BY parentfield BY hierarchyfield
[SUM captionfield]
END
```

結果として生成されるチャートには、次の情報が格納されています。リクエストにAS CAPTION 句が使用されている場合は、関連するキャプションも格納されます。

```
parentfield     hierarchyfield
-----------     --------------
parentvalue1    child1
parentvalue1    child2
parentvalue2    child3
   .
   .
   .
```

別のマスターファイルからの階層のロード

あるマスターファイルに定義した階層を、階層ファイルには結合されていないが、同じ階層フィールドを持つデータソースに対して使用する必要がある場合、階層データを手動でロードすることができます。

ロード可能なチャート数の上限は、利用可能なメモリにより決定されます。チャートは、WebFOCUS の終了時に自動的にアンロードされます。

```
LOAD CHART chartfile[.sega].hierarchyfld
   [FOR requestfile[[.segb].fieldb]]
```

説明

chartfile

階層情報を含むマスターファイル名です。

sega

階層フィールドを含むセグメント名です。セグメント名は、構造内の別のセグメントのフィールド名が階層フィールド名と同一である場合にのみ必要です。

hierarchyfld

階層フィールドです。マスターファイルで複数の階層を定義することができるため、これは必須です。

FOR

FML リクエストで使用されていないマスターファイルで定義される、階層をロードします。たとえば、マスターファイル B に階層情報が含まれており、このマスターファイルがマスターファイル A に結合されておらず、リクエストでマスターファイル A を使用する場合は、FML リクエストの実行前に次の LOAD CHART コマンドを発行します。

```
LOAD CHART B.FLDB FOR A.FLDA
TABLE FILE A ...
```

requestfile

FML リクエストで使用されるマスターファイル名です。

segb

FML リクエストで使用されているマスターファイルの階層フィールド値を含むセグメント名です。*sega* と同一名の場合は必要ありません。

fieldb

階層フィールドの値を含む FML リクエストに指定されたマスターファイルのフィールドです。階層フィールドと同一名の場合は必要ありません。

注意

- 同一階層に LOAD CHART コマンドを複数回発行すると、メモリに存在する以前のバージョンは新しい階層で上書きされます。
- 階層ファイルに動的に結合されたデータソースに LOAD CHART コマンドを発行する場合、まず JOIN コマンドを発行する必要があります。

FML 階層の使用上の注意

- HOLDATTR が ON に設定されているときは、PROPERTY および REFERENCE は、HOLD マスターファイルに継承されます。
- 次の設定は、FML リクエストの複数行 (例、詳細行と概要行) でのデータレコードの使用に必要です。

```
SET FORMULTIPLE=ON
```

- ESSBASE などのロールアップデータソースからレポートを作成する際は、親インスタンスに格納されたデータ値は、そのすべての子孫の集計です。統合されたデータには ADD 機能は使用しません。
- 同一のデータを異なる階層フィールド値に複数回定義できる共有メンバー (例、ESSBASE) を含むデータソースからレポートを作成する際は、2 つの異なる親により共有されるデータは集計演算で 2 回数えられます。集計が 2 回実行されないようにするには、共有フィールドに対する SUM コマンドで FST. 演算子を使用します。
- レポート出力が HTML フォーマットの場合、階層インデントを保持するには、SHOWBLANKS=ON 設定を有効にしておく必要があります。

行タイトルのカスタマイズ

データを正確に識別するために FML レポート内の行タイトルをカスタマイズすることができます。AS 句を使用して、TAG、DATA、RECAP、PICKUP 行に新しいタイトルを指定することができます。

FML の行タイトルのカスタマイズ

TAG 行では、次の構文を使用します。

```
value AS {'title'|CAPTION}
```

DATA または PICKUP 行では、次の構文を使用します。

```
value AS 'title'
```

RECAP 行では、次の構文を使用します。

```
RECAP calcname[/format]=expression; AS 'title'
```

説明

value

レポートの実行に使用する値です。値の取得方法には、データソースまたは外部ファイル (タグにより提示) からの抽出、ユーザによるリクエストへの直接入力、作業ファイルからの取得があります。

title

カスタマイズされた行タイトルです。ブランクが埋め込まれている場合は、一重引用符 (') で囲みます。

TAG、DATA、PICKUP 行では、デフォルトの行タイトルは *value* です。

RECAP 行では、デフォルトの行タイトルは *calcname* です。

CAPTION

階層データソースのマスターファイルでは、CAPTION はキャプションフィールドを使用して TAG 行を識別します。マスターファイル内の階層は、FOR フィールドの PARENT-OF を定義します。

calcname

RECAP 計算により得られた値です。

タグ行タイトルの変更

次の例では、行タイトル CASH ON HAND および DEMAND DEPOSITS により、対応するタグを識別しやすくなります。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' OVER
1020 AS 'DEMAND DEPOSITS'
END
```

割り当てられた行タイトルにはブランクが埋め込まれているため、一重引用符 (') が必要です。

出力結果は次のとおりです。

```
                  AMOUNT
                  ------
CASH ON HAND       8,784
DEMAND DEPOSITS    4,494
```

AS 句が含まれない場合、タグ値はレポートに表示されます。

RECAP 値の行タイトルのカスタマイズ

次のリクエストは、RECAP 値である TOTCASH に TOTAL CASH というタイトルを作成します。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'      OVER
1020 AS 'DEMAND DEPOSITS'   OVER
1030 AS 'TIME DEPOSITS'     OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
END
```

出力結果は次のとおりです。

```
                 AMOUNT
                 ------
CASH ON HAND      8,784
DEMAND DEPOSITS   4,494
TIME DEPOSITS     7,961
TOTAL CASH       21,239
```

AS 句が含まれない場合は、RECAP 値の名前 (TOTCASH) がレポートに表示されます。

FML レポートのフォーマット

次のフォーマットを使用して、FML レポートの見やすさや表現力を向上することができます。

- **数値列に下線を付ける**　数値列を含むレポートでは、RECAP 計算の前に下線の表示が必要な場合がよくあります。下線文字を指定するには、タグ値の代わりに「BAR」を記述します。
- **改ページを追加する**　改ページを挿入するには、レポートの任意の位置に、タグ値の代わりに「PAGE-BREAK」を記述します。
- **行、列、セルのフォーマットを設定する**　FONT、SIZE、STYLE、COLOR などのスタイルシート属性を、指定した行、列、セルに適用することができます。
- **行、列、セル周囲へ境界線を追加する**　スタイルシートで BORDER 属性を使用することにより、行またはセル周囲の境界線の太さ、スタイル、色を指定することができま

す。境界線 (上下左右) のフォーマットを変更することもできます。

- **テキストまたは数値にインデントを設定する**　FML タグ行、階層、または RECAP 行では、タグ値、ラベルテキスト、またはキャプションテキストに、指定したブランク数分のインデントを設定することができます。FML 階層の行にインデントを適用する場合、指定したブランク数だけ階層の親行がインデントされます。

  注意：HTML または PDF レポートで FML 階層のレベル間のインデントを指定するには、BLANKINDENT 設定を使用します。詳細は、FML 階層行タイトルのインデント を参照してください。

列への下線文字の追加

```
BAR [AS 'character'] OVER
```

説明

character

ハイフン文字 (-) または等号文字 (=) のいずれかです。テキストは一重引用符 (') で囲みます。デフォルト文字は、ハイフン (-) です。

列への下線の追加

次の例では、デフォルトの下線文字 (_) を使用します。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'       OVER
1020 AS 'DEMAND DEPOSITS'    OVER
1030 AS 'TIME DEPOSITS'      OVER
BAR                          OVER
RECAP TOTCASH = R1 + R2 + R3;
END
```

出力結果は次のとおりです。

```
                  AMOUNT
                  ------
```

```
CASH ON HAND      8,784
DEMAND DEPOSITS   4,494
TIME DEPOSITS     7,961
                 ------
TOTCASH          21,239
```

BAR ... OVER 句は、表示フィールドを含む列のみに下線を追加します。

FML レポートでの改ページの指定

FML リクエストに、タグ値の代わりに次の構文を記述します。

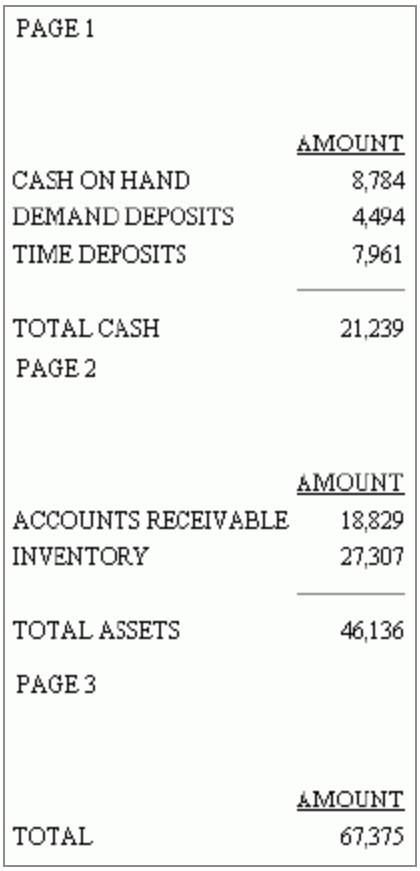
```
PAGE-BREAK OVER
```

FML レポートでの改ページの指定

次の例では、最初の 2 つの RECAP 計算を強調表示するために、これら 2 つの RECAP コマンドの後に改ページが挿入されています。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'                                    OVER
1020 AS 'DEMAND DEPOSITS'                                 OVER
1030 AS 'TIME DEPOSITS'                                   OVER
BAR                                                       OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'             OVER
PAGE-BREAK                                                OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL RECEIVE               OVER
1200 AS 'INVENTORY'  LABEL INVENT                         OVER
BAR                                                       OVER
RECAP TOTASSET = RECEIVE + INVENT; AS 'TOTAL ASSETS'      OVER
PAGE-BREAK                                                OVER
RECAP TOTAL = TOTCASH + TOTASSET;
END
```

出力結果は次のとおりです。



PAGE 1

	AMOUNT
CASH ON HAND	8,784
DEMAND DEPOSITS	4,494
TIME DEPOSITS	7,961
TOTAL CASH	21,239

PAGE 2

	AMOUNT
ACCOUNTS RECEIVABLE	18,829
INVENTORY	27,307
TOTAL ASSETS	46,136

PAGE 3

	AMOUNT
TOTAL	67,375

FML レポートの行、列、セルへのフォーマット設定

```
TYPE=type, [COLUMN=column] [LABEL={Rn|label}], format_def, $
```

説明

type

フォーマットを設定するコンポーネントを識別します。有効な値には、次のものがあります。

REPORT - 指定したラベルの付いた行を表示します。

DATA - 指定したラベルの付いた行のうち、ユーザ入力データ値を含むものを表示します。

FREETEXT - 指定したラベルの付いたテキスト行またはブランク行を表示します。

UNDERLINE - BAR により生成された下線を表示します。下線のフォーマットは PDF ではサポートされますが、HTML レポートではサポートされません。

column

特定の列を識別します。列は、割り当てられた名前、または行内の位置から識別することができます。

LABEL

FML 行の識別およびフォーマット設定に使用する制御要素です。

ラベルは、計算またはフォーマット設定を実行する行の識別に使用されます。TAG または DATA 行のラベルは、レポート出力には表示されません。FML コード内の行を識別するためだけに使用されます。RECAP 行では、計算値の名前がラベルに使用されます。別のタイトルが指定されない限り、レポートにはこの値が表示されます。

label - 明示的ラベルです。行をより正確に識別するために割り当てます。

format_def

FONT、SIZE、STYLE、COLOR などのフォーマット定義です。詳細は、FML レポートのフォーマットを参照してください。

注意：セルにフォーマットを設定するには、スタイルシートで COLUMN=... plus LABEL= ... を使用して、セルを行と列との交点として識別します。

FML レポートでの行のフォーマット

ここでは、行ラベルおよび行データで構成される FML 行の識別およびフォーマットの設定方法について説明します。次のリクエストでは、スタイルシートの LABEL 属性は、スタイルに ITALIC (斜体) が設定された 3 つの TAG 行を識別します。

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND' LABEL COH OVER
1020 AS 'DEMAND DEPOSITS' LABEL DD OVER
1030 AS 'TIME DEPOSITS' LABEL TD OVER
BAR OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
```

```
TYPE = REPORT, LABEL = COH, STYLE = ITALIC, $
TYPE = REPORT, LABEL = DD,  STYLE = ITALIC, $
TYPE = REPORT, LABEL = TD,  STYLE = ITALIC, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	AMOUNT
CASH ON HAND	*8,784*
DEMAND DEPOSITS	*4,494*
TIME DEPOSITS	*7,961*
TOTAL CASH	21,239

FML レポート TAG 行への太字の適用

次のリクエストは、カスタマイズした行タイトル CASH および AMOUNT 列の関連データに太字を適用します。スタイルシートでは、フォーマットを設定するコンポーネントを識別するために、明示的ラベル CA を使用します。

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$$ AS 'CASH'                 LABEL CA   OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL AR   OVER
1200 AS 'INVENTORY'            LABEL INV  OVER
RECAP CURASST/I5C = CA + AR + INV;
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = REPORT, LABEL = CA, STYLE = BOLD, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	AMOUNT
CASH	**21,239**
ACCOUNTS RECEIVABLE	18,829
INVENTORY	27,307
CURASST	67,375

FML マトリックスセルへの太字の適用

次のリクエストは、行タイトル CASH の AMOUNT のデータ値に太字が設定されたレポートを生成します。ただし、行タイトル CASH には太字は適用しません。これを実行するには、太字を適用するセルをスタイルシートに指定します。この例では、行 (CA) の列 (N2) に太字の適用が指定されています。

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
10$$ AS 'CASH'                 LABEL CA   OVER
1100 AS 'ACCOUNTS RECEIVABLE' LABEL AR   OVER
1200 AS 'INVENTORY'            LABEL INV  OVER
RECAP CURASST/I5C = CA + AR + INV;
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = REPORT, COLUMN = N2, LABEL = CA, STYLE = BOLD, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	AMOUNT
CASH	**21,239**
ACCOUNTS RECEIVABLE	18,829
INVENTORY	27,307
CURASST	67,375

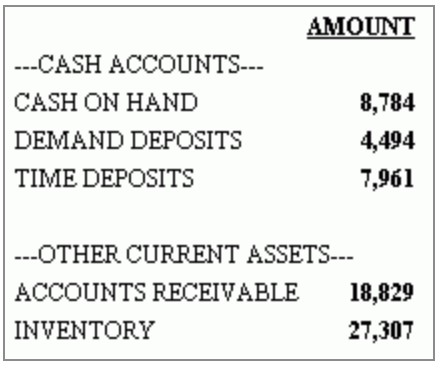
FML レポート列への太字の適用

次のリクエストは、AMOUNT 列を名前で識別し、そのタイトルとデータのフォーマットを太字に設定します。この列が N2 として識別された場合でも、同一の結果になります。

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
"---CASH ACCOUNTS---"                  OVER
1010 AS 'CASH ON HAND'                 OVER
1020 AS 'DEMAND DEPOSITS'              OVER
1030 AS 'TIME DEPOSITS'                OVER
" "                                    OVER
"---OTHER CURRENT ASSETS---"           OVER
1100 AS 'ACCOUNTS RECEIVABLE'          OVER
1200 AS 'INVENTORY'
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = REPORT, COLUMN = AMOUNT, STYLE = BOLD, $
```

```
ENDSTYLE
END
```

出力結果は次のとおりです。



	AMOUNT
---CASH ACCOUNTS---	
CASH ON HAND	8,784
DEMAND DEPOSITS	4,494
TIME DEPOSITS	7,961
---OTHER CURRENT ASSETS---	
ACCOUNTS RECEIVABLE	18,829
INVENTORY	27,307

テキスト行への太字の適用

次のリクエストは、テキストのスタイルに太字を使用します。この例では、1 つのスタイルがテキスト 2 行に適用されるため、これらの行を区別するラベルは必要ありません。

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
"---CASH ACCOUNTS---"  LABEL CA         OVER
1010 AS 'CASH ON HAND'                  OVER
1020 AS 'DEMAND DEPOSITS'               OVER
1030 AS 'TIME DEPOSITS'                 OVER
" "                                     OVER
"---OTHER CURRENT ASSETS---"  LABEL OCA OVER
1100 AS 'ACCOUNTS RECEIVABLE'           OVER
1200 AS 'INVENTORY'
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = FREETEXT, STYLE = BOLD, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	AMOUNT
---CASH ACCOUNTS---	
CASH ON HAND	8,784
DEMAND DEPOSITS	4,494
TIME DEPOSITS	7,961
---OTHER CURRENT ASSETS---	
ACCOUNTS RECEIVABLE	18,829
INVENTORY	27,307

FML レポートテキスト行フォーマットの個別設定

次のリクエストでは、SIZE 属性を定義することにより、CASH ACCOUNTS と OTHER CURRENT ASSETS を識別します。各行を個別に識別し、フォーマットを設定するために、ラベル CA と OCA が使用されています。

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
" --- CASH ACCOUNTS ---" LABEL CA            OVER
1010 AS 'CASH ON HAND'                        OVER
1020 AS 'DEMAND DEPOSITS'                     OVER
1030 AS 'TIME DEPOSITS'                       OVER
" "                                           OVER
" --- OTHER CURRENT ASSETS ---" LABEL OCA    OVER
1100 AS 'ACCOUNTS RECEIVABLE'                 OVER
1200 AS 'INVENTORY'
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = FREETEXT, LABEL = CA, STYLE = BOLD, SIZE = 12, $
TYPE = FREETEXT, LABEL = OCA, STYLE = BOLD, SIZE = 10, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	AMOUNT
---CASH ACCOUNTS---	
CASH ON HAND	8,784
DEMAND DEPOSITS	4,494
TIME DEPOSITS	7,961
---OTHER CURRENT ASSETS---	
ACCOUNTS RECEIVABLE	18,829
INVENTORY	27,307

テキスト行でのテキストおよび変数スタイル設定

次のリクエストでは、テキスト行のテキストと変数コンポーネントのスタイルは個別に設定されます。テキストの Current Assets は斜体で、RECAP 計算から得られた変数は太字で表示されます。

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT AS 'Amount' FOR ACCOUNT
10$$ AS 'Cash'                    LABEL CA    OVER
1100 AS 'Accounts Receivable'     LABEL AR    OVER
1200 AS 'Inventory'               LABEL INV   OVER
RECAP CURASST/I5C = CA + AR + INV; NOPRINT  OVER
"Current Assets: <CURASST"
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID=OFF, $
TYPE = FREETEXT, OBJECT = TEXT, ITEM = 1, SIZE = 12, STYLE = ITALIC, $
TYPE = FREETEXT, OBJECT = FIELD, ITEM = 1, STYLE = BOLD, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	AMOUNT
CASH ON HAND	*8,784*
DEMAND DEPOSITS	*4,494*
TIME DEPOSITS	*7,961*
TOTAL CASH	21,239

FML RECAP 行への太字の適用

次のリクエストは、RECAP 行の行タイトルおよび計算値に太字を適用します。スタイルシートの RECAP ラベルは TOTCASH です。RECAP では、計算値に割り当てられた名前は明示的ラベルとして使用されます。

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'    LABEL CASH    OVER
1020 AS 'DEMAND DEPOSITS' LABEL DD      OVER
1030 AS 'TIME DEPOSITS'   LABEL TD      OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = REPORT, LABEL = TOTCASH, STYLE = BOLD, $
```

```
TYPE = REPORT, LABEL = CASH, COLUMN = N1, STYLE = ITALIC, $
TYPE = REPORT, LABEL = DD, COLUMN = N1, STYLE = ITALIC, $
TYPE = REPORT, LABEL = TD, COLUMN = N1, STYLE = ITALIC, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	AMOUNT
CASH ON HAND	8,784
DEMAND DEPOSITS	4,494
TIME DEPOSITS	7,961
TOTAL CASH	**21,239**

行とセルへの境界線追加およびフォーマット設定

行またはセルの周囲に統一境界線を指定するには、次の構文を使用します。

```
TYPE=REPORT, LABEL=row_label, [COLUMN=column,] BORDER=option,
[BORDER-STYLE=line_style,] [BORDER-COLOR={color|RGB® g b)},] $
```

境界線の上下左右に異なる属性を指定するには、次の構文を使用します。

```
TYPE=REPORT, LABEL=row_label, [COLUMN=column,] BORDER-position=option,
[BORDER-[position-]STYLE=line_style,]
[BORDER-[position-]COLOR={color|RGB(r g b)},] $
```

境界線の上下左右に異なる属性を指定するには、次の構文を使用します。

```
TYPE=REPORT, LABEL=row_label, [COLUMN=column,] BORDER-position=option,
[BORDER-[position-]STYLE=line_style,]
[BORDER-[position-]COLOR={color|RGB(r g b)},] $
```

説明

row_label

指定した境界線スタイルが適用される行です。

column

行ラベルと使用して、境界線スタイルが適用されるセル (行と列の交点) を指定します。

option

次のいずれかの値です。

ON - 見出しまたは脚注に境界線を表示します。ON を指定した場合、MEDIUM と同一の線が生成されます。

OFF - 見出しまたは脚注に境界線を表示しません。デフォルト値は OFF です。

LIGHT - 細い線を指定します。見出しまたは脚注に細い線を指定したり、1 つ以上の境界位置を指定したりできます。

MEDIUM - 標準の太さの線を指定します。ON では、線の太さに MEDIUM が使用されます。見出しまたは脚注に細い線を指定したり、1 つ以上の境界位置を指定したりできます。標準の太さの線を設定することにより、GRID 属性で作成された線の均一性を確実に得ることができます。

HEAVY - 太い線を指定します。見出しまたは脚注に太い線を指定したり、1 つ以上の境界位置を指定したりできます。

width - 単位にポイントを使用して線幅を指定します。72 ポイントは、1 インチです。見出しや脚注の全体、および 1 つ以上の境界位置の線幅をポイント単位で指定することができます。ポイントを使用して指定する線幅の表示は、HTML と PDF 出力で異なります。表示フォーマットに関係なく、外観を統一するには、LIGHT、MEDIUM、または HEAVY を使用します。

position

フォーマットを設定する境界線を指定します。有効な値は、TOP、BOTTOM、LEFT、RIGHT です。

BORDER キーワードに位置識別子を指定することができます。これにより、任意の境界線の幅、スタイル、色にフォーマットを個別に設定することができます。

line_style

境界線のスタイルを設定します。WebFOCUS スタイルシートは、すべての標準カスケードスタイルシートの線スタイルをサポートします。HTML で使用できる 3 次元スタイルは、アスタリスク (*) で示されたものだけに限られています。有効な値には、次のものが

あります。

```
NONE
SOLID
DOTTED
DASHED
DOUBLE*
GROOVE*
RIDGE*
INSET*
OUTSET*
```

color

定義済みの色です。デフォルト値は、BLACK です。

ディスプレイデバイスまたは出力デバイスが指定した色をサポートしない場合、灰色に置き換えられます。

RGB

赤、緑、青の組み合わせを使用して、色を指定します。

(r g b)

赤、緑、青のそれぞれの最適な輝度です。0 (ゼロ) から 255 までの値が使用されます。0 (ゼロ) は最も輝度が低く、255 は最も輝度が高いことを示します。色の構成要素の輝度をすべて同じ値に設定すると、灰色になります。

注意：HTML レポートで BORDERS を使用するには、カスケードスタイルシートが ON に設定されている必要があります。このコードは PDF レポートでは必要ありません。

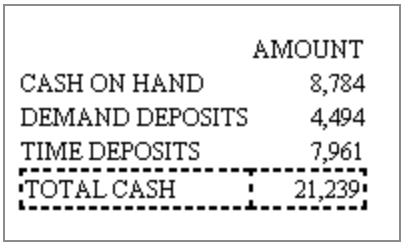
統一境界線による行の強調

次のリクエストは、ラベル TOTCASH により識別される RECAP 行周囲に標準の太さの点線を指定します。HTML レポートで BORDERS を使用するには、カスケードスタイルシートが ON に設定されている必要があります。

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'    LABEL CASH    OVER
1020 AS 'DEMAND DEPOSITS' LABEL DD      OVER
1030 AS 'TIME DEPOSITS'   LABEL TD      OVER
```

```
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF, $
TYPE = REPORT, LABEL = TOTCASH, BORDER = MEDIUM,
  BORDER-STYLE = DASHED, $
ENDSTYLE
END
```

出力結果は次のとおりです。



	AMOUNT
CASH ON HAND	8,784
DEMAND DEPOSITS	4,494
TIME DEPOSITS	7,961
TOTAL CASH	21,239

異なる上下左右の境界線による行の強調

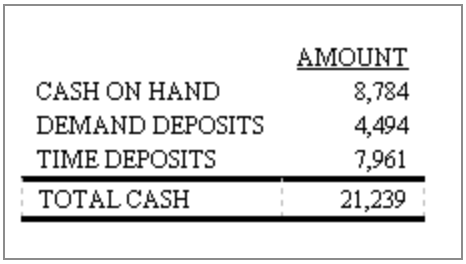
次のリクエストは、ラベル TOTCASH により識別される RECAP 行の上下に黒の太線を、その行の各列の左右に銀色の細い点線を指定します。

HTML レポートで BORDERS を使用するには、カスケードスタイルシートが ON に設定されている必要があります。

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'    LABEL CASH    OVER
1020 AS 'DEMAND DEPOSITS' LABEL DD      OVER
1030 AS 'TIME DEPOSITS'   LABEL TD      OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF,$
TYPE = REPORT, LABEL = TOTCASH,
     BORDER-TOP = HEAVY,
     BORDER-BOTTOM = HEAVY,
     BORDER-LEFT = LIGHT,
     BORDER-RIGHT = LIGHT,
```

```
     BORDER-TOP-STYLE = SOLID,
     BORDER-BOTTOM-STYLE = SOLID,
     BORDER-LEFT-STYLE = DOTTED,
     BORDER-RIGHT-STYLE = DOTTED,
     BORDER-LEFT-COLOR = 'SILVER',
     BORDER-RIGHT-COLOR = 'SILVER', $
ENDSTYLE
END
```

出力結果は次のとおりです。



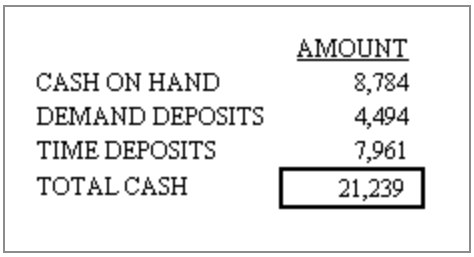
	AMOUNT
CASH ON HAND	8,784
DEMAND DEPOSITS	4,494
TIME DEPOSITS	7,961
TOTAL CASH	21,239

セル周囲への統一境界線の追加

次のリクエストは、ラベル TOTCASH により識別された行の 2 つ目の列のセル周囲に標準の太さの境界線を指定します。LABEL と COLUMN の値の組み合わせによりセルが識別されます。BORDERS を使用するには、カスケードスタイルシートが ON に設定されている必要があります。

```
SET PAGE-NUM=OFF
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
1010 AS 'CASH ON HAND'    LABEL CASH    OVER
1020 AS 'DEMAND DEPOSITS' LABEL DD      OVER
1030 AS 'TIME DEPOSITS'   LABEL TD      OVER
RECAP TOTCASH = R1 + R2 + R3; AS 'TOTAL CASH'
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLESHEET *
TYPE = REPORT, GRID = OFF,$
TYPE = REPORT, LABEL = TOTCASH, COLUMN = N2, BORDER = MEDIUM, $
ENDSTYLE
END
```

出力結果は次のとおりです。



	AMOUNT
CASH ON HAND	8,784
DEMAND DEPOSITS	4,494
TIME DEPOSITS	7,961
TOTAL CASH	21,239

FML ラベル、タグ、キャプションへのインデント指定

```
FOR forfield [IN k]
tag [[GET CHILDREN|WITH CHILDREN] n] INDENT m [AS ['text'|CAPTION]]
[OVER]
```

または

```
RECAP fieldname[/format]=expression; INDENT m [AS 'text']
```

説明

forfield

レポートに含める値を持つデータソースのフィールドです。

k

FML レポート内の FOR 値の開始列です。

tag

FML レポートの行に表示される *forfield* の値です。

n

FML レポートで表示する FML 階層レベル数です。

m

FML 行または階層のタグ値、ラベル、またはキャプションのインデントに使用するブランク数を指定する正の整数です。0 (ゼロ) は使用できません。FOR コマンドに IN 句が指定されていない場合、インデントは 1 桁目から開始します。IN 句が指定されている場合、

インデントは IN 句に指定された桁から開始します。インデントの最大値は、AS 名の最大値と同じです。

FML 階層をインデントする場合、階層の親行は、インデントとして指定されたブランクの数だけインデントが設定されます。各階層レベル間のインデント幅は、ブランク 2 つです。GET CHILDREN 句が使用された場合、階層出力は親からではなく最初の子から始まるため、階層の 1 行目には、ブランク 2 つ分の追加のインデントが設定されます。GET CHILDREN の使用方法についての詳細は、FML 階層の表示を参照してください。

'text'

FML レポートの行に表示されるラベルです。

CAPTION

マスターファイルに定義されたキャプションフィールドを示します。

OVER

表示される行の最終行ではないことを示します。

fieldname

RECAP コマンドにより計算された値に指定される名前です。

format

RECAP フィールドの USAGE フォーマットです。長さは、列幅以内である必要があります。デフォルト値は、計算値が表示される列のフォーマットです。

expression

RECAP フィールド値の計算方法を記述した式です。

FML 階層のタグ行のインデント

次のリクエストは、タグ値 3000 の 2 行目のラベルを 5 バイト分インデントします。このリクエストでは GET CHILDREN 句が使用されているため、タグ値 3000 の 3 行目に定義された FML 階層の 1 行目は 7 バイト分 (5 + 2) インデントされます。

```
SET FORMULTIPLE=ON
TABLE FILE CENTGL
PRINT GL_ACCOUNT_PARENT
FOR GL_ACCOUNT
3000                          AS 'Not Indented'        OVER
```

```
3000                 INDENT 5  AS 'Indented 5'          OVER
3000 GET  CHILDREN 2 INDENT 5  AS 'Hierarchy Indented 5'
END
```

出力結果は次のとおりです。

```
                              Parent
                              ------
Not Indented                  3000
     Indented 5               3000
       Hierarchy Indented 5    3000
         Hierarchy Indented 5  3100
         Hierarchy Indented 5  3100
         Hierarchy Indented 5  3100
         Hierarchy Indented 5  3100
         Hierarchy Indented 5  3000
         Hierarchy Indented 5  3200
         Hierarchy Indented 5  3200
         Hierarchy Indented 5  3200
         Hierarchy Indented 5  3200
         Hierarchy Indented 5  3200
         Hierarchy Indented 5  3200
         Hierarchy Indented 5  3200
```

FML RECAP 行のインデント

次のリクエストは、Digital と Analog 別の価格、コスト、在庫量の合計を計算します。1 つ目の RECAP コマンドは各列の合計を計算し、ラベルを 5 バイト分インデントします。2 つ目の RECAP コマンドは利益を計算し、ラベルを 10 バイト分インデントします。

```
SET FORMULTIPLE=ON
TABLE FILE CENTINV
SUM PRICE COST QTY_IN_STOCK
FOR PRODTYPE
Digital                                          OVER
Analog                                           OVER
BAR                                              OVER
RECAP TOTAL = R1 + R2; INDENT 5  AS 'Total:'     OVER
BAR                                              OVER
RECAP PROFIT(2) = TOTAL(1) - TOTAL(2); AS 'Profit:'INDENT 10
END
```

出力結果は次のとおりです。

```
                                    Our    Quantity
                         Price:     Cost:In Stock:
                         ------     -----  ---------
Digital                  4,080.00      3,052.00     119143
Analog                   1,883.00      1,371.00     139345
                    ------------  ------------    -------
     Total:              5,963.00      4,423.00     258488
                    ------------  ------------    -------
          Profit:                     1540.00
```

FML 階層行タイトルのインデント

FML 階層では、階層間の関係を明確にするため、各レベルで値のキャプション (タイトル) がインデントされます。階層内の各レベル間のインデントの指定には、HTML または PDF で BLANKINDENT パラメータを使用します。インデントを設定する場合、各階層レベルにデフォルトのインデントを使用することも、任意のインデント値を指定することも可能です。キャプションにインデントを設定した HTML レポートを出力するには、BLANKINDENT パラメータを ON に設定するか、特定の値を指定する必要があります。

SET BLANKINDENT でインデントが設定された列の幅が十分ではなく、インデントされたフィールド値が列幅に収まらない場合でも、列幅は再定義されません。テーブルベースのフォーマットでは、必要に応じて列幅が自動的に調整されますが、位置ベースのフォーマットでは (例、PDF、PPTX)、列の位置にずれが発生します。スタイルシート構文を使用すると、インデントされた値の表示に十分な列幅を確保し、後続の列を移動することができます。必要な列幅を確保するには、スタイルシートの SQUEEZE = n 属性を使用して列幅を変更します。

関連機能を使用して、階層の親行の前、あるいは FML リクエストの FML タグまたは RECAP 行の前のブランク数を変更することができます。詳細は、FML レポートのフォーマットを参照してください。

HTML レポート FML 階層キャプションのインデント

```
SET BLANKINDENT={ON|OFF|n}
ON TABLE SET BLANKINDENT {ON|OFF|n}
```

説明

ON

FML 階層キャプションをキャプションの前に設定されたブランクごとに 0.125 単位インデントします。FML 階層の子レベルでは、対象の行とその 1 つ前の行との行間に設定されたスペースごとに 0.125 単位ずつインデントします。

OFF

HTML レポートでは、FML 階層のキャプションに適用するインデントをオフにします。デフォルト値は OFF です。他のフォーマットでは、デフォルトのインデントであるブランク 2 つ分を使用します。

n

UNITS パラメータで定義した単位に基づいて明示的に指定する値です。この値は、キャプション前に設定されたバイト数により乗算されます。FML 階層内の子レベルでは、対象の行とその上の行とのインデントの差と *n* の乗算値だけインデントされます。デフォルトのブランク数は 2 です。0 (ゼロ) を指定すると、OFF を指定した場合と同一のレポート出力が生成されます。*n* には、負の数は使用できません。

FML 階層キャプションでのデフォルトインデントの使用

次のリクエストは、デフォルトのインデントを使用して HTML レポートを作成します。

```
SET PAGE-NUM=NOPAGE
SET BLANKINDENT=ON
SET FORMULTIPLE=ON
TABLE FILE CENTGL
PRINT GL_ACCOUNT_PARENT
```

```
FOR GL_ACCOUNT
3000                  AS CAPTION      OVER
3000 GET  CHILDREN 2  AS CAPTION
ON TABLE SET ONLINE-FMT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
TYPE = REPORT, GRID = OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	Parent
Total Operating Expenses	1000
Selling Expenses	3000
Advertising	3100
Promotional Expenses	3100
Joint Marketing	3100
Bonuses/Commisions	3100
General + Admin Expenses	3000
Salaries-Corporate	3200
Company Benefits	3200
Depreciation Expenses	3200
Gain/(Loss) Sale of Equipment	3200
Leasehold Expenses	3200
Interest Expenses	3200
Utilities	3200

FML 階層キャプションインデント値の指定

次のリクエストは、FML 階層の各レベルに 0.25 のインデントを指定します。この値は、デフォルトの測定単位であるインチで表されます。

```
SET PAGE-NUM=NOPAGE
SET BLANKINDENT=.25
SET FORMULTIPLE=ON
TABLE FILE CENTGL
PRINT GL_ACCOUNT_PARENT
```

```
FOR GL_ACCOUNT
3000                 AS CAPTION          OVER
3000 GET  CHILDREN 2 AS CAPTION

ON TABLE SET STYLE *
TYPE = REPORT, GRID = OFF, $
ENDSTYLE
END
```

出力結果は次のとおりです。

	Parent
Total Operating Expenses	1000
Selling Expenses	3000
Advertising	3100
Promotional Expenses	3100
Joint Marketing	3100
Bonuses/Commisions	3100
General + Admin Expenses	3000
Salaries-Corporate	3200
Company Benefits	3200
Depreciation Expenses	3200
Gain/(Loss) Sale of Equipment	3200
Leasehold Expenses	3200
Interest Expenses	3200
Utilities	3200

行の非表示

レポートに行を表示することなく、計算での使用のみを目的として TAG 行のデータを検索する場合があります。タグ行を非表示にするには、TABLE リクエストと同様に、行の宣言に NOPRINT を追加します。

また、TAG 行の値にデータが存在しない場合に、行を非表示にすることができます。詳細は、データのない行の非表示を参照してください。

さらに、RECAP コマンドに NOPRINT とその後ろにセミコロン (;) を記述することにより、RECAP 行を非表示にすることができます。これは、他の計算への入力として使用される中間 RECAP 値を非表示にする際に役立ちます。

TAG 行の非表示

次のリクエストは、COST の値を計算に使用しますが、レポートの行には表示しません。

```
DEFINE FILE REGION
AMOUNT/I5C=E_ACTUAL;
END

TABLE FILE REGION
SUM AMOUNT FOR ACCOUNT
3000 AS 'SALES' LABEL SLS           OVER
3100 AS 'COST' LABEL COST NOPRINT   OVER
RECAP PROFIT/I5C = SLS - COST;      OVER
" "                                 OVER
RECAP ROS/F6.2 = 100*PROFIT/SLS;
AS 'RETURN ON SALES'
END
```

出力結果は次のとおりです。

	AMOUNT
SALES	6,000
PROFIT	1,350
RETURN ON SALES	22.50

データのない行の非表示

タグ行には、タグ値のファイル内にデータが存在しない場合にもテキスト (ミッシングデータを表すピリオド (.)) が表示されます。TAG 行の定義に WHEN EXISTS 句を追加することにより、この設定を上書きすることができます。WHEN EXISTS により、行はタグにデータが存在する場合にのみ表示されます。この機能は、1 つのモデルが企業の異なる部門に適用された場合などに役立ちます。

データのない行の非表示

次の例では、変数 DIVISION には Divison 1 と Divison 2 が含まれていることを想定します。次のリクエストは、1 つの FML レポートに貸借対照表を表示します。各部門に関係のない行は、表示されません。

```
TABLE FILE LEDGER
HEADING CENTER
"BALANCE SHEET FOR DIVISION <DIVISION"
" "
SUM AMOUNT
BY DIVISION NOPRINT
ON DIVISION PAGE-BREAK
FOR ACCOUNT
2000 AS 'LAND' WHEN EXISTS LABEL LD          OVER
2100 AS 'CAR LOANS' WHEN EXISTS LABEL LOAN   OVER
   .
   .
   .
```

中間レポート結果の保存と取得

多くのレポートでは、以前のレポートで作成された結果が必要です。これは、中間値の格納場所が指定されている場合にのみ可能です。中間値が必要な例には、貸借対照表で資本を計算する前に、損益計算書での純利益の計算が必要な場合が挙げられます。FML は、モデル (複数可) から選択した行を作業ファイルに追加することにより、行を保存することができます。追加された行は作業ファイルから抽出され、再利用されます。

デフォルトの作業ファイルは FOCPOST です。これはカンマ区切りのファイルで、FOCPOST マスターファイルが利用可能な場合にレポートを直接作成することができます。リクエストで作業ファイルを使用するには、データを追加するレポートの実行前、およびデータを抽出するレポートの実行に、FOCPOST ddname に物理的な名前を割り当てる必要があります。リクエストで作業ファイルを使用するには、データを追加するレポートの実行前、およびデータを抽出するレポートの実行に、FOCPOST ddname に物理的な名前を割り当てる必要があります。

ファイルに物理的な名前を割り当てるには、リクエストを実行する前にコマンドを発行します。Windows および UNIX では FILEDEF コマンドを使用します。

FML レポートは、リクエストに直接入力したデータから作成することはできませんが、カンマ区切り作業ファイルに格納されたデータのみから作成することができます。

データの保存

TAG、RECAP、DATA 行は、ファイルに出力して保存することができます。保存した行は、DATA 行に値を追加した場合と同様に使用することができます。

行のオプションにもよりますが、行はレポートで通常どおりに処理され、その後ファイルに追加されます。行のラベルが最初に記述され、カンマ (,) で区切られた列の数値、終了文字 ($) の順に記述されます。詳細は、行の作業ファイルへの追加を参照してください。

注意：実際にレポートに表示されるフィールドのみが出力されます。表示されないフィールドは出力されません。この種のフィールドには、NOPRINT オプションが指定されているフィールド、リクエストでフォーマットを再設定するときに生成される追加のフィールド、演算で指定するフィールドがあります。

データのファイルへの追加

以下は、TAG、RECAP、DATA 行を保存する構文です。

```
POST [TO ddname]
```

説明

ddname

データを追加する作業ファイルに割り当てられた論理名です。

この構文は、作業ファイルに追加する任意の行に追加します。

行の作業ファイルへの追加

次のリクエストは、FML レポートを作成し、2 つのタグ行を LEDGEOUT 作業ファイルに追加します。

```
FILEDEF LEDGEOUT DISK [APP]¥LEDGEOUT.DAT
```

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
```

```
LAST_YR/I5C=.87*CUR_YR - 142;
END
```

```
TABLE FILE LEDGER
SUM CUR_YR LAST_YR
FOR ACCOUNT
1100 LABEL AR POST TO LEDGEOUT OVER
1200 LABEL INV POST TO LEDGEOUT
END
```

出力結果は次のとおりです。

	CUR_YR	LAST_YR
1100	18,829	15,954
1200	27,307	23,329

作業ファイルからのデータの抽出

作業ファイルに追加した行を抽出し、FML リクエストに次の句を追加することにより、DATA 行に入力されたときと同様に使用することができます。

```
DATA PICKUP [FROM ddname] id1 [OR id2 ...] [LABEL label] [AS 'text']
```

説明

ddname

データが抽出される作業ファイルの論理名です。

id

作業ファイルに追加され、現在抽出されているデータ行に割り当てられるラベルです。

label

抽出するデータに割り当てられるラベルです。

抽出するデータに割り当てるラベルは、追加したデータのラベル (id) 値に一致させることができます。値を一致させる必要はありません。

LABEL および AS 句を含めることはできますが、WHEN EXISTS はサポートされていません。

注意：抽出されたフィールドは、レポートのメモリリポジトリ内 (内部マトリックス) のすべてのフィールド (表示、非表示に関わらず) にマッピングされます。マトリックスに出力ファイルのフィールドと関係を持たない列が含まれる場合、抽出された値は位置揃えされないことがあります。たとえば、PICKUP リクエストでフィールドフォーマットを再設定する場合、そのフィールドは内部マトリックスでは 2 つの列で表されます。ただし、出力されるファイルには、このフィールドを表す値は 1 つしか存在せず、抽出された値は、マトリックス上の対応する列へは適切にマッピングされません。

作業ファイルからのデータの抽出

次の例では、LEDGER データソースと LEDGEOUT 作業ファイル内のデータは、RECAP 計算に使用されます。このファイルの作成方法についての詳細は、行の作業ファイルへの追加を参照してください。

ヒント：リクエストの実行前に、Windows、UNIX では FILEDEF コマンドを発行して、ファイルに論理名を割り当てる必要があります。アロケーションウィザードを使用して、FILEDEF コマンドを作成することができます。

```
DEFINE FILE LEDGER
CUR_YR/I5C=AMOUNT;
LAST_YR/I5C=.87*CUR_YR - 142;
END
```

```
TABLE FILE LEDGER
SUM CUR_YR LAST_YR
FOR ACCOUNT
1010 TO 1030 AS 'CASH' LABEL CASH   OVER
DATA PICKUP FROM LEDGEOUT AR  AS 'ACCOUNTS RECEIVABLE' LABEL AR   OVER
DATA PICKUP FROM LEDGEOUT INV AS 'INVENTORY' LABEL INV            OVER
BAR                                 OVER
RECAP CUR_ASSET/I5C = CASH + AR + INV;
END
```

出力結果は次のとおりです。

	CUR_YR	LAST_YR
CASH	21,239	17,198
ACCOUNTS RECEIVABLE	18,829	15,954
INVENTORY	27,307	23,329
CUR_ASSET	67,375	56,481

次の行は、LEDGEOUT から 2 つのアカウントの合計を抽出するために使用することができます。

```
DATA PICKUP FROM LEDGEOUT AR OR INV
AS 'ACCTS REC AND INVENTORY'
```

注意：PICKUP ファイルの行は、標準のカンマ区切りフォーマットで保存されるため、行はあらかじめファイルとして取得しておくことや、ユーザが直接作成しておくことができます。

FML レポートからの HOLD ファイルの作成

FML で作成したレポートは、TABLE 言語で作成したレポートと同様に HOLD ファイルに抽出することができます。

この場合、各行に指定されるタグ値のセットは、モデルに AS テキストで指定される説明フィールドで識別します。行にテキストが指定されていない場合は、最初のタグ値が自動的に使用されます。このため、1 行につき 1 つのみタグを持ち、テキストが指定されていない単純なモデルでは、HOLD ファイルの行にはタグ値が 1 つのみ含まれます。RECAP 計算から抽出された行は、HOLD ファイルの一部を形成します。テキスト行 (BAR 行を含む) は、省略されます。

RECAP で HOLD を使用するには、RECAP フィールドのフォーマットは元のフィールドのフォーマットと同一である必要があります。

この機能を使用することにより、計算の結果である行を HOLD ファイルに新しく作成することができます。作成された HOLD ファイルは、さまざまな TABLE リクエストで使用することができます。

注意：HOLD ファイルの作成時に、RECAP 行のフォーマットを再設定することはできません。

FML レポートからの HOLD ファイルの作成

次のリクエストは、CASH、ACCOUNTS RECEIVABLE、INVENTORY、および RECAP 行である CURRENT ASSETS のレコードを含む HOLD ファイルを作成します。

```
TABLE FILE LEDGER
SUM AMOUNT FOR ACCOUNT
```

```
1010 TO 1030 AS 'CASH'                                OVER
1100 AS 'ACCOUNTS RECEIVABLE'                         OVER
1200 AS 'INVENTORY'                                   OVER
RECAP CA = R1 + R2 + R3; AS 'CURRENT ASSETS'
ON TABLE HOLD
END
```

HOLD ファイルにクエリを実行します。

```
>
?HOLD

DEFINITION OF HOLD FILE:HOLD

FIELDNAME          ALIAS          FORMAT

                   E01            A 19
AMOUNT             E02            I5C
```

その後、以下を使用して HOLD ファイルからレポートを作成します。

```
TABLE FILE HOLD
PRINT E01 E02
END
```

出力結果は次のとおりです。

```
                     AMOUNT
                     ------
CASH                 21,239
ACCOUNTS RECEIVABLE  18,829
INVENTORY            27,307
CURRENT ASSETS       67,375
```

自由形式レポートの作成

独自に設計したレイアウトを使用して、データを制限のない形式 (自由形式) で表現することができます。

表形式レポートおよびマトリックスレポートは、レコードの比較を目的として列と行でデータを表現し、グラフ形式レポートは、図表やグラフを使用してデータを視覚的に表現します。これに対し、自由形式レポートでは、ページ上のデータの配置を選択することができます。自由形式レポートは、レポートの各ページにデータソースレコードのカスタムデータを表示するときに役立ちます。

注意：自由形式レポートは、PDF、HTML、スタイルフォーマットで使用することができます。HTML 出力では、レポートのすべてのページが単一の HTML ページに表示されます。PDF 出力では、改ページが保持されます。

自由形式レポートの作成

自由形式レポートは、一覧およびマトリックス形式 (PRINT、LIST、SUM、COUNT) を制御する表示コマンドを省略した TABLE リクエストを作成することができます。リクエストには、次のレポートコンポーネントが含まれます。

見出し	レポートの本文が格納されます。レポートを構成するテキスト、図形文字、およびデータフィールドを表示します。
脚注	レポートの脚注が格納されます。レポートの各ページの下部に表示されるテキストです。脚注には、見出しと同一のテキストおよびデータフィールドを表示することができます。
演算接頭語	フィールドの演算および操作を示します。

一時項目	データソースの既存のフィールドから新しい値を取得します。
BY 句	レポートのソート順序を指定し、各ページに含めるレコード数を決定します。
WHERE 条件	レポートのレコードを選択します。

自由形式レポートの作成では、次の作業が可能です。

- テキスト、データフィールド、図形文字を含むレポートの設計。詳細は、自由形式レポートの設計を参照してください。
- レポートのレイアウトのカスタマイズ。詳細は、自由形式レポートのレイアウト指定を参照してください。
- レポートに含めるレコードとソート順序の選択。詳細は、自由形式レポートでのレコードのソートおよび選択を参照してください。

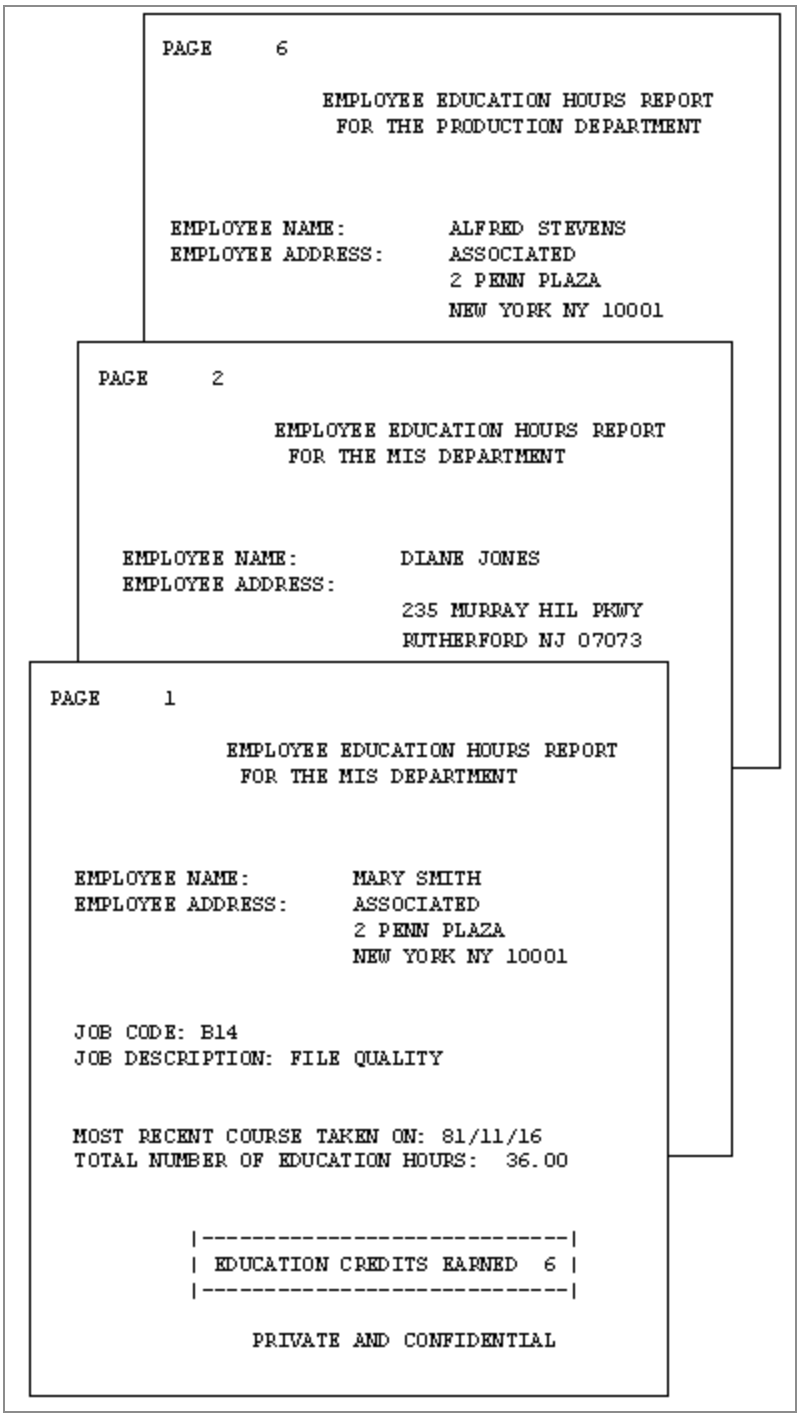
自由形式レポートの作成

ある企業の人事部長が、企業の教育方針の管理を担当していることを想定します。この企業の教育方針には、経費を使用した社外受講時間数は、社内受講時間数に基づいて決定されることが定められてます。

受講時間を効率よく管理するためには、各従業員の社内受講時間を表すレポートが必要です。レポートは各従業員の人事ファイルに保存し、社外受講を許可する際に参照する必要があるため、各従業員の情報は個々のページに出力します。

ここでは、この要件を満たすために、各従業員を個々のページに出力する EMPLOYEE EDUCATION HOURS REPORT を作成します。レポートの 1 ページ目と 2 ページ目には MIS 部の従業員の情報が、6 ページ目には PRODUCTION 部の従業員の情報が表示されていることに注意してください。

下図は、「EMPLOYEE EDUCATION HOURS REPORT」という名前のプロシジャを実行した場合の出力結果を表します。


PAGE 6
EMPLOYEE EDUCATION HOURS REPORT
FOR THE PRODUCTION DEPARTMENT
EMPLOYEE NAME: ALFRED STEVENS
EMPLOYEE ADDRESS: ASSOCIATED
2 PENN PLAZA
NEW YORK NY 10001
PAGE 2
EMPLOYEE EDUCATION HOURS REPORT
FOR THE MIS DEPARTMENT
EMPLOYEE NAME: DIANE JONES
EMPLOYEE ADDRESS:
235 MURRAY HIL PKWY
RUTHERFORD NJ 07073
PAGE 1
EMPLOYEE EDUCATION HOURS REPORT
FOR THE MIS DEPARTMENT
EMPLOYEE NAME: MARY SMITH
EMPLOYEE ADDRESS: ASSOCIATED
2 PENN PLAZA
NEW YORK NY 10001
JOB CODE: B14
JOB DESCRIPTION: FILE QUALITY
MOST RECENT COURSE TAKEN ON: 81/11/16
TOTAL NUMBER OF EDUCATION HOURS: 36.00
EDUCATION CREDITS EARNED 6
PRIVATE AND CONFIDENTIAL

「EMPLOYEE EDUCATION HOURS REPORT」のリクエスト

次のリクエストは、EMPLOYEE EDUCATION HOURS REPORT を生成します。リクエストの左の番号は、このリクエストの後にある注釈に対応しています。

```
1.SET STYLE = OFF
   SET STYLEMODE=FIXED
   SET ONLINE-FMT = PDF
2.DEFINE FILE EMPLOYEE
      CR_EARNED/I2 = IF ED_HRS GE 50 THEN 9
         ELSE IF ED_HRS GE 30 THEN 6
         ELSE 3;
      END
3.TABLE FILE EMPLOYEE
   BY DEPARTMENT
4.HEADING
   " "
   "<13>EMPLOYEE EDUCATION HOURS REPORT"
5. "<14>FOR THE <DEPARTMENT DEPARTMENT"
6. "</2"
   "EMPLOYEE NAME:    <23><FIRST_NAME <LAST_NAME>"
   "EMPLOYEE ADDRESS: <23><ADDRESS_LN1>"
   "<23><ADDRESS_LN2>"
   "<23><ADDRESS_LN3>"
   "</1"
   "JOB CODE: <JOBCODE>"
   "JOB DESCRIPTION: <JOB_DESC>"
   "</1"
7."MOST RECENT COURSE TAKEN ON: <MAX.DATE_ATTEND>"
   "TOTAL NUMBER OF EDUCATION HOURS: <ED_HRS>"
   "</1"
8. "<10>|-------------------------------------|"
9. "<10>| EDUCATION CREDITS EARNED <CR_EARNED>|"
   "<10>|-------------------------------------|"
10.BY EMP_ID NOPRINT PAGE-BREAK
11.WHERE ED_HRS GT 0
12.FOOTING
   "<15>PRIVATE AND CONFIDENTIAL"
   END
```

次に、サンプルレポートを作成するリクエストの、それぞれの行の役割について説明します。

1. ブラウザでレポートを希望の形式で表示するには、2 つの SET コマンドが必要です。SET STYLE = OFF コマンドは、デフォルトのスタイルシートパラメータを無視するため、自由形式設計が可能になります。SET STYLEMODE = FIXED は、HTML 形式をオフにします。レポート作成者は、スポットマーカーおよびスキップラインコマンドを使用して、レポート内の項目の配置を決定することができます。
2. DEFINE コマンドは、レポートの一時項目を作成します。演算は、社外受講単位取得についての企業方針を反映します。その結果は CR_EARNED に格納され、レポートの後部に表示されます。
3. 自由形式レポートは、標準の TABLE FILE コマンドで開始します。サンプルレポートは、EMPLOYEE データソースを使用します。
4. HEADING コマンドで開始する見出しにはレポートの本文が定義されます。レポートに表示されるテキストおよびデータフィールドのほとんどは、見出しに指定されます。このリクエストの見出しには、2 つ目の BY 句である BY EMP_ID NOPRINT PAGE-BREAK までが含まれます。
5. この行は、以下を示します。
   - ページの見出し内の 2 行目のテキスト。
   - 「<DEPARTMENT」というテキストに埋め込まれたデータフィールド。
   - 行の開始位置。「<14>」は 14 桁を表します。
6. スキップラインコマンドを使用することで、レポートの読みやすさを向上させることができます。コマンド「</2」のみが 1 行に記述されると、ブランク行が 2 行生成されます。これは、出力結果のページ見出しと従業員名 (EMPLOYEE NAME) 間のブランク行で確認することができます。
7. この行は、演算接頭語を使用して、自由形式レポートでのフィールド計算の実行方法を示します。ここでは、最後の受講日 (DATE_ATTEND フィールドの最大値) が要求されています。
8. この行とそれに続く 2 行は、レポートの図形の作成に特別な文字が使用されることを示します。「EDUCATION CREDITS EARNED」を囲む枠線は、プロポーショナルフォントで表示される出力では、調整が必要な場合があります。
9. DEFINE コマンドにより作成されたフィールド値は枠線内に表示されます。従業員の最新受講単位数は強調表示されます。この行は、レポートの本文に一時項目 (DEFINE) を表示できることを示しています。
10. この行は、自由形式レポートでのソートの使用を示します。このレポートでは、1 ペー

ジにつき従業員 1 名のみの情報が出力される必要があります。これは、BY および PAGE-BREAK コマンドを使用することで可能になります。改ページ付きのレポートを作成するには、レポートの出力フォーマットを PDF にする必要があります。

11. 自由形式レポートのレコード選択を指定することができます。WHERE 条件の結果、レポートには社内受講単位数を持つ従業員のみが含まれます。
12. このレポートは人事関連であるため、その機密性を明示することが重要です (ここでは、各レポートページの最下部に「PRIVATE AND CONFIDENTIAL」と挿入)。これには、FOOTING コマンドを使用します。

自由形式レポートの設計

自由形式レポートの本文を設計するには、HEADING および FOOTING コマンドを使用します。これらのコマンドを使用して、次のことを行うことができます。

- レポートにテキスト、データフィールド、および図形文字を挿入する。
- テキストおよびデータを正確な位置に配置し、ブランク行を追加して読みやすくする。

自由形式レポートの本文の定義には HEADING コマンドを使用し、レポートの各ページ下部に表示する内容の定義には FOOTING コマンドを使用します。脚注はオプションです。見出しのみを使用してレポート全体を定義することができます。

自由形式レポートへのテキスト挿入

必要に応じて、自由形式レポートの任意の位置にテキストを指定することができます。この例のサンプルリクエスト (「EMPLOYEE EDUCATION HOURS REPORT」のリクエストを参照) では、次のテキストが使用されます。

- レポートのタイトル

```
"<13>EMPLOYEE EDUCATION HOURS REPORT"
```

- データフィールドのラベル

```
"EMPLOYEE NAME:   <FIRST_NAME <LAST_NAME>"
```

- データフィールドおよび図形文字での表示

```
"<10>| EDUCATION CREDITS EARNED <CR_EARNED>|"
```

- ページの脚注

```
"<15>PRIVATE AND CONFIDENTIAL"
```

自由形式レポートへのデータフィールド挿入

自由形式などの形式に関わらず、データはレポートにおいて不可欠な要素です。リクエストで利用可能なデータフィールドには、マスターファイルのデータフィールド、クロスリファレンスフィールド、DEFINE コマンドで作成された一時項目があります。

この例のサンプルリクエストでは、これら 3 種のデータフィールドがすべて参照されています。詳細は、「EMPLOYEE EDUCATION HOURS REPORT」のリクエストを参照してください。

- ED_HRS は、EMPLOYEE マスターファイル内にあります。

```
"TOTAL NUMBER OF EDUCATION HOURS: <ED_HRS>"
```

- DATE_ATTEND は、EDUCFILE マスターファイル内にあります。DATE_ATTEND は、EMPLOYEE マスターファイル内のクロスリファレンスフィールドです。

```
"MOST RECENT COURSE TAKEN ON: <MAX.DATE_ATTEND>"
```

- CR_EARNED は、TABLE FILE コマンドの前に DEFINE コマンドで作成されます。このフィールドは、次のように参照されます。

```
"<10>| EDUCATION CREDITS EARNED <CR_EARNED>|"
```

データフィールドに演算接頭語を適用することもできます。これにより、特定の値 (例、ソートグループ内の最大値) の選択、または計算 (例、フィールドの平均値の計算) を実行することができます。自由形式レポートでは、任意の演算接頭語を使用することができます。

このサンプルリクエストでは、演算接頭語 MAX は、最後に受講した社内講習の最終日を選択します。

```
"MOST RECENT COURSE TAKEN ON: <MAX.DATE_ATTEND>"
```

すべてのレポートタイプにおいて、演算接頭語を正しく使用するには、データソースの構造を理解している必要があります。

自由形式レポートへの図形文字挿入

レポートには、任意の図形を必要に応じて作成することができます。このサンプルレポートでは、テキストおよび一時項目 (DEFINE) を囲む枠線に特殊文字が使用されています。詳細は、自由形式レポートの作成を参照してください。図形文字の使用には、次のものがあります。

- アスタリスク (*) やその他の特殊文字による主要データフィールドの強調表示。これらの特殊文字は、キーボードから直接入力、または HEXBYT 関数を使用して挿入します。HEXBIT についての詳細は、『ibi™ WebFOCUS® 関数リファレンス』を参照してください。
- レポートページ全体を囲む枠線の使用。これにより、定型フォームのような外観が得られます。
- ページの見出しおよび脚注とレポートの本文の境界線としての二重線の使用。

図形の作成に使用される特殊文字には制限があります。入力および表示可能な文字はワークステーションにより異なり、印刷可能な文字もプリンタにより異なります。図形の作成が困難な場合は、利用可能な文字について企業の担当者に確認してください。

自由形式レポートのレイアウト指定

レポートでブランク行を挿入し、テキストおよびデータフィールドの位置を指定するには、HEADING および FOOTING コマンドのスポットマーカー機能を使用します。

注意：HTML レポートでこの機能を利用するには、リクエストに SET STYLEMODE=FIXED コマンドを記述します。

このサンプルリクエスト (「EMPLOYEE EDUCATION HOURS REPORT」のリクエストという名前の例を参照) では、この機能が示されています。最初の 2 つの例には、レポートへのテキストおよびデータフィールドの配置方法が示されています。3 つ目の例には、ブランク行を追加する方法が示されています。

- スポットマーカー <13> は、指定したテキストをレポートの 13 桁目に配置します。

```
"<13>EMPLOYEE EDUCATION HOURS REPORT"
```

- スポットマーカー <23> は、指定したデータフィールドをレポートの 23 桁目に配置します。

```
"<23><ADDRESS_LN2>"
```

- スポットマーカー </1 が 1 行に単独で記述されている場合、ジョブ説明の後にブランク行が 2 つ挿入されます。

```
"JOB DESCRIPTION: <JOB_DESC>""</1""MOST RECENT COURSE TAKEN ON:
<MAX.DATE_ATTEND>"
```

自由形式レポートを設計する際は、NOPRINT、PAGE-BREAK (PDF 出力のみ)、および UNDER-LINE などのソートフィールドオプションを活用します。このサンプルリクエストでは、別ページに各従業員情報を配置するために PAGE-BREAK が使用されています。

```
BY EMP_ID NOPRINT PAGE-BREAK
```

自由形式レポートでのレコードのソートおよび選択

表形式レポートやマトリックスレポートと同様に、レポートをソートし、条件に基づいてレコードを選択することができます。表形式レポートおよびマトリックスレポートと同じコマンドを使用します。たとえば、レポートのソートには BY 句を、データソースからのレコードの選択には WHERE 条件を定義します。

SQL によるレポートの作成

ユーザは、SQL ステートメントを TABLE フォーマット句と組み合わせたリクエストを発行することができます。

これらのリクエストの組み合わせは、ANSI レベル 2 SQL ステートメントを実行可能な FOCUS リクエストに変換する SQL トランスレータを通してサポートされます。

FOCUS と DBMS データの抽出および分析には、SQL トランスレータを使用することができます。

SQL ステートメントのサポート

SQL 変換サービスは、ANSI レベル 2 に準拠し、SQL ステートメントの多くをサポートします。Reporting Server および特定の RDBMS エンジンでは、*alpha1* CONCAT *alpha2* 構文がサポートされる場合もあります。詳細は、サポートされる SQL ステートメントおよび サポートされない SQL ステートメントを参照してください。

サポートされる SQL ステートメントの多くは、標準 SQL への変換候補の対象となります。この機能により、サーバはインバウンド SQL リクエストを SQL 対応の SUB Server およびデータアダプタへ送信することができます。標準 SQL への変換では、Reporting Server DML (データ処理言語) への変換は実行されず、データロケーションの透過性が維持されます。また、標準 SQL ステートメントを送信先の SQL エンジンで処理可能な言語に変換します。ステートメントの語義は保持されます。

注意：SQL トランスレータは ANSI レベル 2 に準拠するため、以前のバージョンで有効なリクエストには、利用できないものがあります。

サポートされる SQL ステートメント

SQL 変換サービスは、次のステートメントをサポートします。

- SELECT (SELECT ALL および SELECT DISTINCT も含む)。
- EQUIJOIN および EQUIJOIN 以外。

- OUTER JOIN。特定の制限が適用されます。詳細は、SQL JOINを参照してください。
- CREATE VIEW および DROP VIEW。
- PREPARE および EXECUTE。
- テーブル名およびフィールド名の区切り文字を含む識別子。SELECT リスト内のブランクや特殊文字が埋め込まれているテーブル名およびフィールド名は、二重引用符 (") で囲まれている必要があります。
- テーブル名またはテーブルタグにより修飾されたフィールド名。
- UNION [ALL]、INTERSECT [ALL]、EXCEPT [ALL] 演算子。
- WHERE 述語および FROM リスト内のリクエストすべてに対する非相関サブクエリ。
- この機能をサポートする RDBMS に変換される標準 SQL への変換候補であるリクエストに対する相関サブクエリ。相関サブクエリは、FOCUS およびその他の非リレーショナルデータソースでサポートされないことに注意してください。
- SELECT リスト内の数値定数、リテラル、式。
- RDBMS エンジンがスカラ関数タイプをサポートする場合、標準 SQL への変換候補であるクエリのスカラ関数。スカラ関数には、ABS、CHAR、CHAR_LENGTH、CONCAT、COUNTBY、DATE、DAY、DAYS、DECIMAL、EDIT、EXTRACT、FLOAT、HOUR、IF、INT、INTEGER、LCASE、LENGTH、LOG、LTRIM、MICROSECOND、MILLISECOND、MINUTE、MONTH、POSITION、RTRIM、SECOND、SQRT、SUBSTR (または SUBSTRING)、TIME、TIMESTAMP、TRIM、VALUE、UCASE、YEAR があります。
- リテラルまたは数値フィールドとともに使用される連結演算子 「||」。
- 集計関数 (COUNT、MIN、MAX、SUM、AVG)。
- 条件で使用可能な式 (CASE、NULLIF、COALESCE)。
- さまざまなフォーマットの日付、時間、タイムスタンプのリテラル。詳細は、日付、時間、タイムスタンプフィールドの SQL トランスレータサポートを参照してください。
- =ALL、<>ANY、<>SOME 以外の、ANY、SOME、ALL を含むリクエストすべて。
- エンジンが修飾されたサブクエリをサポートする場合、標準 SQL への変換候補であるリクエストの =ALL、<>ANY、<>SOME。
- 特殊レジスタ USER、CURRENT_DATE、CURRENT_TIME、CURRENT_TIMESTAMP、CURRENT_EDASQLVERSION、および CURRENT_TIMEZONE。

- NULL および NOT NULL 述語。
- LIKE および NOT LIKE 述語。
- IN および NOT IN 述語。
- 時間および日付の演算。
- EXISTS および NOT EXISTS 述語。
- 明示的なフィールドの名前、AS 名、またはフィールドの位置を使用して表現される GROUP BY 句。
- 明示的なフィールドの名前または個数を使用した ORDER BY 句。
- レコードのロックを回避する FOR FETCH ONLY 機能。
- CDN 変数が設定されている場合のコンチネンタル式 10 進表記 (CDN)。
- 国際言語サポート (NLS)。

サポートされない SQL ステートメント

SQL 変換サービスは、次のステートメントをサポートしません。

- 1 つの SELECT につき 16 個以上の JOIN。これは、SQL で設定されている制限です。FOCUS では 16 個までの JOIN がサポートされます。
- マスターファイル内の ALIAS 名および出力のフォーマットを設定するフォーマットオプションの使用。
- 一意のフィールド短縮名。
- 一時項目 (DEFINE)。Reporting Server の動的カタログまたはマスターファイルで定義された一時項目 (DEFINE)。
- DML 生成の相関サブクエリ。

SQL トランスレータの予約語

次の語句は、SQL トランスレータとともに使用するマスターファイルのフィールド名に使用することはできません。

- ALL
- COUNT
- SUM
- MAX
- MIN
- AVG
- CURRENT
- DISTINCT
- USER

SQL トランスレータコマンドの使用

サポートされる任意のデータソース (複数可) を使用してレポートを作成する際に、SQL コマンドを使用することができます。レポートのフォーマットに使用される標準 TABLE 句は、幅広いレポート準備オプションを利用するために SQL SELECT ステートメントに追加することができます。

注意：リクエストのデータソースを結合する必要がある場合、SQL ステートメントの発行前に JOIN コマンドを使用するか、SQL ステートメントの WHERE 述語を使用して必要なファイルを動的に結合します。詳細は、SQL JOINを参照してください。

SQL トランスレータコマンドの使用

```
SQL
sql statement;
[ECHO|FILE]
[TABLE phrases]
END
```

説明

SQL

SQL トランスレータを呼び出す SQL コマンド識別子です。

注意：SQL コマンドコンポーネントは、前述の順序で記述する必要があります。

sql statement

サポートされる SQL ステートメントです。ステートメントは、セミコロン (;) で終了する必要があります。複数行にわたって記述することができます。詳細は、サポートされる SQL ステートメントを参照してください。

SQL ステートメントでは、ファイル名は ANSI レベル 2 により 48 バイト以下に制限されています。SQL CREATIVE VIEW ステートメントで生成されたビュー名は 18 バイト以下に制限されています。サブクエリは 15 レベルまでネストすることが可能です。相関サブクエリは、FOCUS およびその他の非リレーショナルデータソースではサポートされません。

ECHO

生成された TABLE リクエストを取得するオプションのデバッグ句です。これらのオプションは、SQL ステートメントの後に配置します。

FILE [name]

指定のプロシジャに変換後の TABLE 句を書き込みます。ファイル名を指定しないと、リクエストの実行時にデフォルト名が割り当てられます。その後、ファイルは削除されます。

TABLE phrases

オプションの TABLE フォーマット句です。詳細は、SQL リクエスト内の TABLE フォーマット句を参照してください。

END または QUIT

プロシジャの終了に必要です。

SQL トランスレータコマンドの使用

次のリクエストには、SQL ステートメントおよび TABLE フォーマットコマンドが含まれています。

```
SQL
SELECT BODYTYPE, AVG(MPG), SUM(SALES)
```

```
FROM CAR
WHERE RETAIL_COST > 5000
GROUP BY BODYTYPE;
TABLE HEADING CENTER
"AVERAGE MPG AND TOTAL SALES PER BODYTYPE"
END
```

SQL リクエスト内の TABLE フォーマット句

SQL リクエストには、TABLE フォーマット句を含めることができますが、その場合は次の規則に従う必要があります。

- TABLE フォーマット句は、SELECT および UNION とともに使用できます。
- フォーマット句は、TABLE で開始します。
- 見出しと脚注の指定、ON 句による動作の定義、ON TABLE SET コマンドの使用を行えます。また、ON TABLE HOLD または ON TABLE PCHOLD を使用して抽出ファイルを作成することができます。さらに、READLIMIT および RECORDLIMIT テストを指定することもできます。

  見出しおよび脚注についての詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。

  ON TABLE HOLD および ON TABLE PCHOLD についての詳細は、レポート出力の保存と再利用を参照してください。
- 追加の表示フィールド、ACROSS フィールド、WHERE および IF 条件 (READLIMIT および RECORDLIMIT テスト以外)、一時項目 (COMPUTE) を指定することはできません。BY 句は無視されます。

SQL SELECT ステートメント

SQL SELECT ステートメントは、リクエストに適用された内容が個別フィールドの表示か集計かにより、TABLE PRINT または TABLE SUM コマンドに変換されます。詳細は、レポートデータの表示を参照してください。

SQL ステートメントの SELECT * は、マスターファイル内の各フィールドを PRINT に変換し、直積のフィールドのすべてを使用します。これにより、ファイルが妥当な数の画面に表示可能である場合、または抽出したデータを再利用するためのファイルへの保存に ON TABLE HOLD または ON TABLE PCHOLD を使用する場合、ファイルをすばやく表示できます。詳細は、レポート出力の保存と再利用を参照してください。

SQL 関数 (COUNT、SUM、MAX、MIN、AVG など) は、SELECT リストおよび HAVING 条件でサポートされています。式は、関数の引数として使用することができます。

関数 COUNT (*) は、マスターファイル内のすべてのフィールドを出力することにより生成されるレコード件数に変換されます。これは、すべてのフィールド上の SELECT から生成される直積のすべての行数を計算することと同じです。

SQL WHERE 述語内の式は、可能な限り TABLE リクエストの対応する WHERE 条件に変換されます。SELECT リスト内の式は、一時項目 (DEFINE) を生成します。SQL HAVING 句は、TABLE リクエストでは対応する WHERE TOTAL 条件にも変換されます。SQL LIKE 演算子は、TABLE リクエストの WHERE 条件内の対応する LIKE 演算子に直接変換されます。TABLE リクエスト内のレコードの選択についての詳細は、レコードの選択を参照してください。

WHERE 式が等号 (=) に基づいてサブクエリと比較されるときは、等価条件を基準とするサブクエリのみがサポートされます。たとえば、WHERE field = (SELECT ...) です。

SQL UNION 演算子は、指定された各データソースに HOLD ファイルを作成する TABLE リクエストに変換されます。その後には、1 つ目のデータソース (旧) と 2 つ目のデータソース (新) の両方からのレコードを組み合わせるオプション HOLD OLD-OR-NEW を含む MATCH コマンドが続きます。詳細は、データソースの結合 (MATCH)を参照してください。

関連情報については、サポートされる SQL ステートメントおよび SQL トランスレータコマンドの使用を参照してください。

FROM 句を伴わない SQL SELECT ステートメントの使用

FROM 句を伴わない SELECT ステートメントは、1 つ以上の定数値で構成される 1 行のアンサーセットの取得でサポートされます。この種の SELECT を発行する場合、マスターファイルは必要ありません。この構文の用途として、関数のテストが挙げられます。たとえば、次の SQL SELECT は、単一のリテラル値および 2 つの関数コールの結果を取得します。

```
SQL
SELECT 'MOD' AS FUNCTION, MOD(6,3) AS MOD1, MOD(5,3) AS MOD2;
```

```
TABLE
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。

FUNCTION	MOD1	MOD2
MOD	0	2

SQL JOIN

SQL JOIN を実行する際は、結合されたフィールドのフォーマットは同一の必要があります。JOIN フィールドにはインデックスは必要はなく、等結合 (EQUIJOIN) 以外がサポートされます。

再帰的 JOIN、OUTER JOIN、INNER JOIN がサポートされます。INNER JOIN がデフォルト値です。

INNER JOIN の作成

INNER JOIN では、2 つの構文がサポートされています。

構文 1

```
SQL
SELECT fieldlist FROM file1 [alias1], file2 [alias2]
[WHERE where_condition];
END
```

構文 2

```
SQL
SELECT fieldlist FROM file1 [alias1] INNER JOIN file2 [alias2]
ON join_condition [INNER JOIN ...]
```

```
[WHERE where_condition];
END
```

説明

fieldlist

データソースから抽出されるフィールドを識別します。

SQL WHERE 述語内の JOIN フィールド名が一意ではない場合は修飾する必要があるため、対応するファイル名またはファイルのエイリアスを指定します。以下はその例です。

```
{file1|alias1}.field1, {file2|alias2}.field2
```

FROM

結合されるデータソースを示します。

file1, file2

結合されるデータソースです。

alias1, alias2

結合されるデータソースのオプションのエイリアスです。

where_condition

JOIN アンサーセットのオプションの選択条件です。この条件を満たさない JOIN 行は、返されるアンサーセットから削除されます。構文 1 で省略した場合、アンサーセットは 2 つのデータソースの直積となります。

join_condition

結合条件です。

OUTER JOIN の作成

```
SQL
SELECT fieldlist FROM file1 {LEFT|RIGHT|FULL} JOIN file2
ON join_condition [{LEFT|RIGHT|FULL} JOIN ...]
WHERE where_condition
```

```
END
```

説明

fieldlist

抽出元のデータソースと抽出するフィールドを識別します。

SQL WHERE 述語内の JOIN フィールド名が一意ではない場合は修飾する必要があるため、対応するファイル名またはファイルのエイリアスを指定します。以下はその例です。

```
{file1|alias1}.field1, {file2|alias2}.field2
```

FROM

結合されるデータソースを示します。

file1, file2

結合されるデータソースです。

alias1, alias2

結合されるデータソースのオプションのエイリアスです。

join_condition

結合条件です。条件は、等式 (例、T1.A=T2.B) を指定する必要があります。

where_condition

JOIN アンサーセットのオプションの選択条件です。この条件を満たさない JOIN 行は、返されるアンサーセットから削除されます。

SQL トランスレータからの JOIN 名の割り当て

SQL トランスレータが発行する JOIN には、次のフォーマットの名前が割り当てられます。

```
SQLJNMnn
```

説明

SQLJNM

SQL トランスレータの JOIN 接頭語です。

nn

JOIN が作成される順序で割り当てられる 01 から 16 までの数値です (FOCUS は最大 16 個の JOIN をサポートします)。1 つ目の JOIN 名は、AS 名である SQLJNM01、2 つ目の JOIN 名は SQLJNM02 で、SQLJNM16 まで同様に名前が割り当てられます。

すべての JOIN は自動的に作成され、SQL トランスレータによりクリアされます。ユーザ指定の JOIN には影響はありません。

SQL JOIN でのフィールドの修飾名の使用

次のステートメントでは、T.A および U.B はフィールドの修飾名です。

```
SQL
 SELECT T.A, T.B
 FROM T, U
 WHERE T.A = U.B;
END
```

SQL の再帰的 JOIN の使用

次のステートメントでは、A と B はともにデータソース CAR のエイリアスです。CAR からの出力は、A の値と同じ値を持つ B の値の組み合わせとなります。

```
SQL
   SELECT A.SEATS, B.SEATS
   FROM CAR A, CAR B
   WHERE A.MODEL = B.MODEL;
END
```

SELECT 句内のフィールド名のすべてが一意であるか、修飾名である必要があります。

SQL FULL OUTER JOIN の使用

次のステートメントでは、B、C、D がそれぞれ異なるデータソースのエイリアスです。

```
SQL
SELECT
       B.FIELD1 AS B_FIELD1, B.FIELD2 AS B_FIELD2,
       D.FIELD1 AS D_FIELD1, D.FIELD2 AS D_FIELD2
FROM
((FILE1 B FULL OUTER JOIN FILE2 C ON B.FIELD2 = C.FIELD2 )
           FULL OUTER JOIN FILE3 D ON C.FIELD2 = D.FIELD2 )
WHERE B.FIELD1 < 2
END
```

複数の FULL OUTER JOIN がサポートされます。ただし、これらの JOIN により、一時 HOLD ファイルが多数生成される場合があります。

SQL JOIN の考慮事項

- 標準 SQL では、WHERE field='a' は a または A の値を持つフィールドからレコードを検索します。SQL トランスレータは大文字と小文字を区別し、リクエストされた内容に完全に一致する値を返します。この場合、'a' のみが返されます。
- SQL 比較演算子 ANY、SOME、ALL (=ALL、<>ANY、<>SOME を除く) はサポートされます。
- サブ選択は HAVING 条件ではサポートされません。
- 複数のセグメントを含む構造では、下位セグメントが存在しない親セグメントはレポートから除外されます。これが INNER JOIN です。
- SQL トランスレータは、JOIN の作成時に最適化手法を適用します。詳細は、インデックスが最適化された抽出を参照してください。

インデックスの最適化、および最適化済み JOIN ステートメントについての詳細は、使用するサーバのマニュアルを参照してください。

直積スタイルのアンサーセット

SQL トランスレータは、この機能を明示的にオフにしない限り、直積スタイルのアンサーセットを自動的に生成します。ただし、この機能をオフにすると ANSI 標準に準拠しないため、CARTESIAN の設定をオンのままにしておくことをお勧めします。SET CARTESIAN コマンドについての詳細は、データソースの結合 (MATCH)を参照してください。

コンチネンタル式 10 進表記 (CDN)

コンチネンタル式 10 進表記は、カンマ (,) で小数位を、ピリオド (.) で有効桁数を 3 つに区切ります。この表記は SQL トランスレータのリクエストで利用可能です。

桁の分割に CDN を使用

次の例では、「1.2 + SEATS」として定義されたフィールドが作成されます。

```
SET CDN=ON
SQL
   SELECT SEATS + 1,2
   FROM CAR;
END
```

SQL リクエストでのフィールド名の指定

SQL リクエストでは、次の方法でフィールドを指定します。

- **区切り文字を含む識別名**　フィールド名には「.」、「#」、「@」、「_」、「$」を含めることはできますが、フィールド名の先頭にこれらの記号を使用することはできません。これらの記号を含むフィールド名を参照する際、フィールド名を二重引用符 (") で囲む必要があります。
- **修飾フィールド名**　フィールド名およびファイルのエイリアス名を修飾します。ファイルのエイリアス名についての詳細は、SQL JOINの結合に関する説明を参照してください。
- **ブランクおよび特殊文字を含むフィールド名**　SELECT リストでは、ブランクやその他の特殊文字を含むフィールド名を指定することができます。これらのフィールド名は二重引用符 (") で囲む必要があります。特殊文字は、区切り文字を含む識別子としてリストされていない文字で、インストールされた FOCUS 環境の言語別の文字セットには含まれていません。

区切り文字を含む識別子を持つフィールド名の指定

次のフィールド識別子をリクエストに含めることができます。

```
"COUNTRY.NAME"
```

区切り文字を含むフィールド名の修飾

区切り文字を含むフィールド名の COUNTRY.NAME をそのファイル名で修飾するには、次のように指定します。

```
CAR."COUNTRY.NAME"
```

SQL UNION、INTERSECT、EXCEPT 演算子

SQL UNION、INTERSECT、EXCEPT 演算子は、MATCH ロジックを生成します。含まれるファイル数は、MATCH により制限されます。括弧が使用された UNION はサポートされます。

- SELECT A UNION SELECT B は、A または B、あるいは両方の行を抽出します(これは、MATCH 句 OLD-OR-NEW に相当します)。
- INTERSECT は、A と B の両方の行を抽出します (これは、MATCH 句 OLD-OR-NEW に相当します)。
- EXCEPT は A の行を抽出しますが、B の行は抽出しません (これは、MATCH 句 OLD-OR-NEW に相当します)。

MATCH ロジックは、データソースの内容を結合します。詳細は、データソースの結合(MATCH)を参照してください。

数値定数、リテラル、式、関数

SQL SELECT リスト、WHERE 述語、HAVING 句には、数値定数、一重引用符 (') で囲まれたリテラル、式、任意のスカラ関数を含めることができます。内部的には、SELECT フィールド内の各値に対して、一時項目 (DEFINE) が 1 つ作成されます。一時項目 (DEFINE) の値は、アンサーセットに入力されます。

日付、時間、タイムスタンプフィールドの SQL トランスレータサポート

SQL トランスレータには、WHERE 述語内の日付時間フィールドや SELECT ステートメント内のフィールドリストをサポートするために新しいデータタイプが定義されました。

さらに、リレーショナルまたは FOCUS データソースでは、時間またはタイムスタンプフィールドを定義することができます。これらのフィールドは SQL トランスレータからアクセス可能です。値は INSERT および UPDATE ステートメントを使用して入力することができ、SELECT ステートメント内に表示されます。

時間またはタイムスタンプデータ項目 (フィールドまたはリテラル) は、条件で比較することができます。時間やタイムスタンプの値は、その和と差を計算することができます。結果は秒数の差で表されます。T + 2 HOURS や TS + 5 YEARS などの式は許可されます。これらの式は、日付時間関数の呼び出しに変換されます。日付時間関数についての詳細は、『ibi™ WebFOCUS® 関数リファレンス』を参照してください。

マスターファイルの実フィールドおよび一時項目の日付フォーマットはすべて YYYYMMDD に変換されます。コンポーネントが不足したフォーマットを指定した場合、SQL トランスレータは欠落しているコンポーネントのデフォルト値を入力します。月など、日付の一部を指定するには、文字フォーマットの一時項目を使用します。

日付、時間、タイムスタンプフィールドの SQL トランスレータサポート

下表で、fff は小数第 3 位 (ミリ秒) の秒を、ffffff は小数第 6 位 (マイクロ秒) の秒を表します。

次のフォーマットは、SQL トランスレータに入力することができます。

フォーマット	マスターファイル内の USAGE 属性	日付構成要素
日	`YYMD`	`YYYY-MM-DD`
時間	`HH`	`HH`
時間から分まで	`HHI`	`HH.MM`
時間から秒まで	`HHIS`	`HH.MM.SS`
時間からミリ秒まで	`HHISs`	`HH.MM.SS.fff`
時間からマイクロ秒まで	`HHISsm`	`HH.MM.SS.ffffff`
年から時間まで	`HYYMDH`	`YYYY-MM-DD HH`
年から分まで	`HYYMDI`	`YYYY-MM-DD HH.MM`
年から秒まで	`HYYMDS`	`YYYY-MM-DD HH.MM.SS`
年からミリ秒まで	`HYYMDs`	`YYYY-MM-DD HH.MM.SS.fff`
年からマイクロ秒まで	`HYYMDm`	`YYYY-MM-DD HH.MM.SS.ffffff`

注意

- 時間情報は、時間、分、秒、または 1 秒未満の時間に指定することができます。
- 日付情報に使用できる区切り文字は、ハイフン (-) またはスラッシュ (/) のみです。
- 時間情報に使用できる区切り文字は、コロン (:) のみです。
- 日付情報と時間情報の区切り文字には、ブランクのみ使用できます。

SQL トランスレータによる日付時間構成要素の抽出

SQL トランスレータは、日付時間値から構成要素を返す関数をサポートします。構成要素を抽出するには、EXTRACT ステートメントを使用します。

日付、時間、タイムスタンプの値から先頭および末尾のパターンを削除するには、TRIM 関数を使用します。詳細は、『ibi™ WebFOCUS® 関数リファレンス』を参照してください。

SQL トランスレータで使用可能な日付、時間、タイムスタンプ関数

次の関数は、日付構成要素を整数の値として返します。ここで、x は日付時間の値とします。

関数	戻り値
`YEAR(x)`	年
`MONTH(x)`	月数

関数	戻り値
`DAY(x)`	日数
`HOUR(x)`	時間
`MINUTE(x)`	分
`SECOND(x)`	秒
`MILLISECOND(x)`	ミリ秒
`MICROSECOND(x)`	マイクロ秒

SQL トランスレータの日付、時間、およびタイムスタンプ関数の使用

値が「1999-11-23 07:32:16.123456」であるタイムスタンプフィールド TS の戻り値は、次のとおりです。

```
YEAR(TS) = 1999
MONTH(TS) = 11
DAY(TS) = 23
HOUR(TS) = 7
MINUTE(TS) = 32
SECOND(TS) = 16
MILLISECOND(TS) = 123
MICROSECOND(TS) = 123456
```

SELECT ステートメント内の SQL トランスレータの日付、時間、およびタイムスタンプ関数の使用

「VIDEOTR2」という名前の FOCUS データソースに「TRANSDATE」という名前の日付時間フィールドが含まれていることを想定します。

```
SQL
SELECT TRANSDATE,
YEAR(TRANSDATE), MONTH(TRANSDATE),
MINUTE(TRANSDATE)
FROM VIDEOTR2;
FILE VIDSQL
END
```

SQL トランスレータは、関数に次の一時項目 (DEFINE) を生成します。出力を表示するために TABLE リクエストがその後に記述されています。

```
SET COUNTWIDTH=ON
-SET SQLERRNUM = 0;
DEFINE FILE
VIDEOTR2 TEMP
SQLDEF01/I4 MISSING ON NEEDS ALL = HPART(TRANSDATE,'YEAR','I4');
SQLDEF02/I2 MISSING ON NEEDS ALL = HPART(TRANSDATE,'MONTH','I2');
SQLDEF03/I2 MISSING ON NEEDS ALL = HPART(TRANSDATE,'MINUTE','I2');
END
TABLEF FILE VIDEOTR2
PRINT TRANSDATE SQLDEF01 AS 'SQLDEF01' SQLDEF02 AS 'SQLDEF02' SQLDEF03 AS
  'SQLDEF03'
ON TABLE SET CARTESIAN ON
ON TABLE SET ASNAMES ON
ON TABLE SET HOLDLIST PRINTONLY
END
```

出力結果は次のとおりです。

```
TRANSDATE         SQLDEF02  SQLDEF04  SQLDEF05
1999/06/20 04:14      1999         6        14
1991/06/27 02:45      1991         6        45
```

```
1996/06/21 01:16      1996         6        16
1991/06/21 07:11      1991         6        11
1991/06/20 05:15      1991         6        15
1999/06/26 12:34      1999         6        34
1919/06/26 05:45      1919         6        45
1991/06/21 01:10      1991         6        10
1991/06/19 07:18      1991         6        18
1991/06/19 04:11      1991         6        11
1998/10/03 02:41      1998        10        41
1991/06/25 01:19      1991         6        19
1986/02/05 03:30      1986         2        30
1991/06/24 04:43      1991         6        43
1991/06/24 02:08      1991         6         8
1999/10/06 02:51      1999        10        51
1991/06/25 01:17      1991         6        17
```

日付時間構成要素を抽出する SQL トランスレータの EXTRACT 関数の使用

日付時間構成要素を整数値として抽出するには、次の ANSI 標準関数を使用します。

```
EXTRACT(component FROM value)
```

説明

component

YEAR、MONTH、QUARTER、DAY、WEEKDAY、HOUR、MINUTE、SECOND、MILLISECOND、MICROSECOND のいずれかです。

value

日付時間、DATE、TIME、TIMESTAMP フィールド、定数、式のいずれかです。

たとえば、次の式は同等です。

```
EXTRACT(YEAR FROM TS)
YEAR(TS)
```

EXTRACT 関数の使用

```
SELECT D. EXTRACT(YEAR FROM D), EXTRACT(MONTH FROM D),
EXTRACT(DAY FROM D) FROM T
```

このリクエストは、次のような行を生成します。

```
1999-01-01     1999      1       1
2000-03-03     2000      3       3
```

インデックスが最適化された抽出

SQL トランスレータは、最適化コードを生成することにより、クエリのパフォーマンスを向上させます。これにより、抽出エンジンは、セグメントインスタンスのすべてをスキャンすることなく、選択したレコードに直接アクセスすることができます。

インデックスの最適化および最適化済み JOIN ステートメントについての詳細は、使用するプラットフォームに対応するサーバのマニュアルを参照してください。

最適化 JOIN

SQL トランスレータは、SQL 構文内の JOIN を受け入れます。SQL 言語の JOIN は方向性を持ちません。ホストファイルおよびクロスリファレンスファイルの概念は SQL には存在しません。

SQL トランスレータは、各 JOIN を分析し、効率のよい実装方法を識別します。まず、クエリ内の結合候補にコストを割り当てます。

- FOCUS 結合規則に従って、クロスリファレンスフィールドとして加えることが可能なフィールドへの等結合 (EQUIJOIN) は cost = 1 です。これは、WHERE 句内の EQUIJOIN 述語を持つリレーショナルテーブルに対するクエリでは一般的です。
- FOCUS 結合規則に従って、クロスリファレンスフィールドとして加えることができないフィールドへの等結合 (EQUIJOIN) は cost = 16 です。
- 等結合 (EQUIJOIN) 以外または制限のない直積は cost = 256 です。

その後、トランスレータは、クエリの JOIN 構造を作成するためにこれらのコストを使用します。クエリの FROM 内のテーブルの順序は、結合分析の最初の 2 句に影響します。

1. FROM 句で参照される 1 つ目のテーブルから 2 つ目のテーブルへ、2 つ目のテーブルから 3 つ目のテーブルへ cost=1 JOIN が存在する場合、トランスレータはクエリに指定された順序でテーブルを結合します。そうでない場合は、第 2 工程へと続行します。
2. 第 1 工程が受け入れ可能な JOIN 構造の生成に失敗した場合、トランスレータはテーブルを FROM 句で優先されるテーブルに結合することなく、JOIN 構造の生成を試みます。これにより、この工程は常に、クエリで参照される 1 つ目のテーブルをホストテーブルにします。2 つのテーブル間に cost=1 JOIN がない場合、または JOIN を使用するためにテーブルの順序を変更する必要がある場合、トランスレータは第 2 工程を放棄し、第 3 工程を実装します。
3. トランスレータは、まず最もコストの低い JOIN から JOIN 構成を生成し、その後必要に応じてよりコストの高い JOIN を生成します。このソート処理により、テーブルの結合順序が変更される場合があります。このプロシジャが生成する JOIN の効率は、JOIN されているテーブルの相対的なサイズに依存します。

分析の結果、FOCUS の規則によりクロスリファレンスファイルとして加えることができないテーブル (例、インデックスの欠落) への結合が行われる場合、トランスレータはインデックス付きの HOLD ファイルを作成するコードを生成し、このファイルとの結合を導入します。ただし、HOLD ファイルは、結合順序の分析には参加しません。

TABLEF の最適化

パフォーマンスを向上するために、SQL トランスレータは、TABLE コマンドではなく、FOCUS TABLEF コマンドを生成するよう設定することができます。SET SQLTOPTTF コマンド (SQL トランスレータの OPTimization TableF) を使用して、この最適化を利用します。詳細は、パフォーマンスの改善を参照してください。

SQLTOPTTF によるパフォーマンスの向上

```
SET SQLTOPTTF = {ON|OFF}
```

説明

ON

JOIN や GROUP BY 句がない場合など、可能であれば TABLEF コマンドを生成します。デフォルト値は ON です。

OFF

TABLE コマンドを生成します。

パフォーマンスの改善

データ検索とレポート処理のパフォーマンスを改善するには、次の方法があります。

- ネットワーク型、および階層型データソースの一時回転によるデータの代替ビューの作成。
- AUTOPATH 機能による自動代替ファイルビュー。
- 自動インデックス検索 (AUTOINDEX)。
- TABLEF コマンドを使用したソート済みデータの検索。
- SAVEMATRIX パラメータを使用したレポート内部マトリックスの保存。
- 式をマシンコードにコンパイルすることによる処理の高速化。

注意：データソースの種類によっては、これらの方法は使用できない場合があります。詳細は、データアダプタのマニュアルを参照してください。

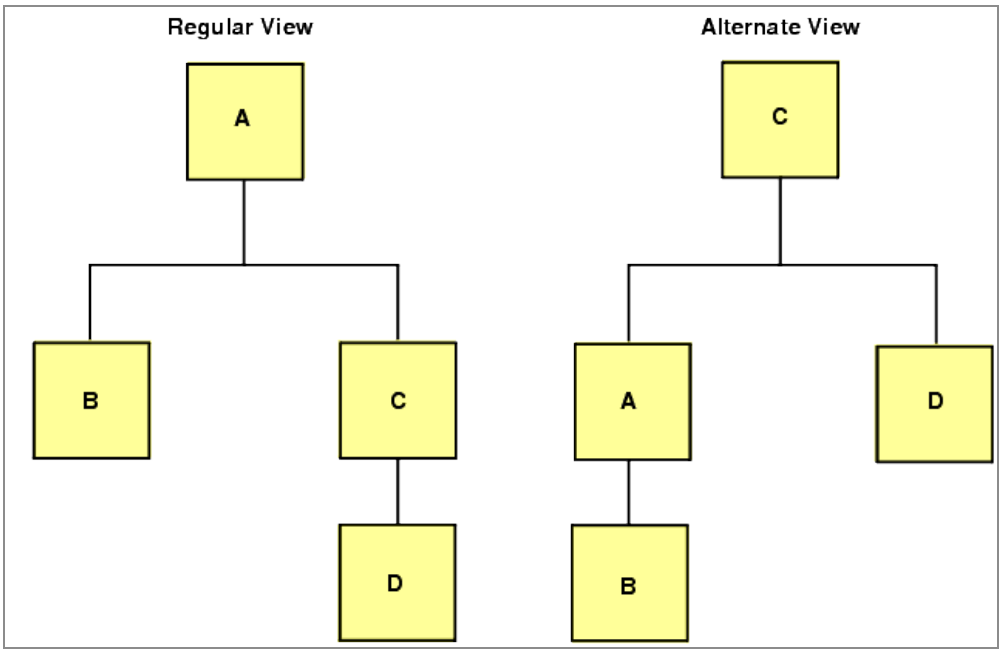
データ構造回転による検索速度の改善

CA-IDMS/DB、FOCUS などの特定のネットワーク型または階層型のデータソースを使用する場合は、データソースを回転して代替ビューを作成することができます。代替ビューにより、一部のセグメント関係が変更され、別の順序でセグメントにアクセスすることが可能になります。代替ビューからレポートを実行すると、次のことが可能になります。

- アクセスパスの変更。たとえば、セグメントを上位へ移動することにより、下位セグメントからより高速にデータにアクセスすることができます。
- データソースのパス構造の変更。このオプションは、ファイル内の異なるパス上に存在する複数のソートフィールドを使用してレポートを作成する場合に、特に役立ちます。ファイル階層のビューを変更することにより、使用するすべてのソートフィールドを同一パス上に配置することができます。

この場合、データの取得が最小参照サブツリーにより制御されることに注意してください。

たとえば、次の通常ビュー (Regular View) および代替ビュー (Alternate View) について考察します。



C は代替ビューのルートセグメントであるため、C の特定インスタンスをより高速に選択することができます。

代替ビューの要求

代替ビューを要求するには、代替ルートセグメントで指定したフィールド名を TABLE コマンドのファイル名にピリオド (.) で区切って追加します。

```
TABLE FILE filename.fieldname
```

データ再構成時の注意

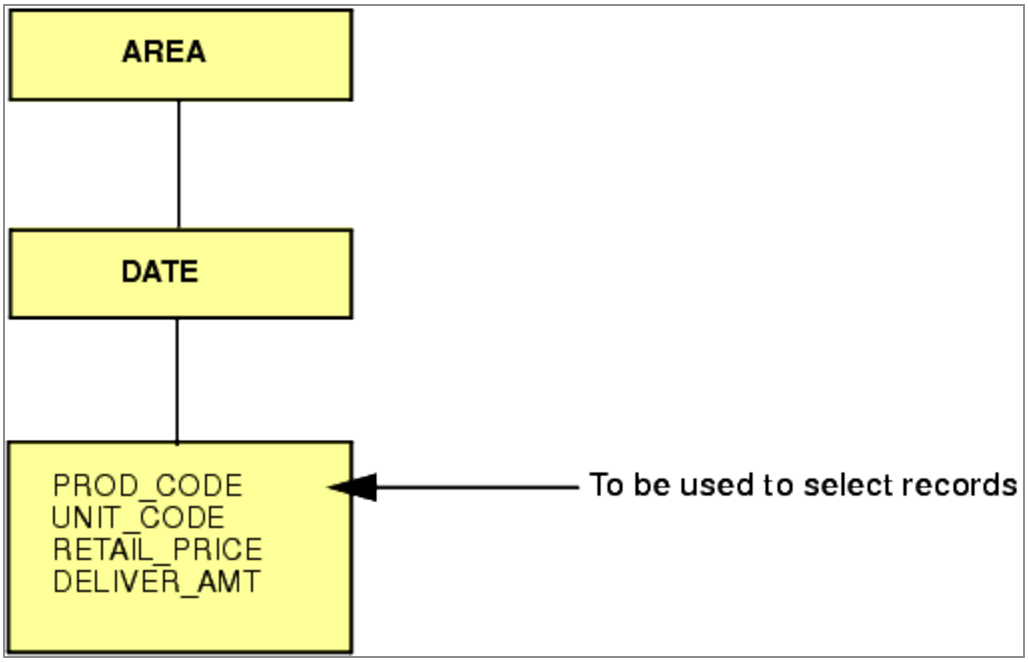
- インデックス付きでないフィールドを使用すると、各セグメントインスタンスは指定のレコードが見つかるまで検索されます。そのため、インデックスフィールドを使用する方が効率的です。
- 特定の子セグメントで代替ビュー機能を使用すると、そのセグメントから検索するデータは、論理順ではなく物理順に検索されます。これは、その子セグメントがリクエストのルートセグメントになり、異なる親の子セグメント間に論理ポインタがない

ためです。

- インデックスフィールドの代替ビューは、検索にインデックスを使用する特殊なケースです。インデックスフィールドで代替ビューを実行すると、検索速度が向上します。ただし、パフォーマンスを実際に改善するには、インデックスフィールドに、WHERE (MONTH EQ 1) OR (MONTH EQ 2) などの等価テストを含める必要があります。
- 代替ビューで指定するフィールド名に修飾名を使用することはできません。また、フィールド名が 12 バイトを超えることはできません。
- 自動インデックス検索 (AUTOINDEX) は、TABLE リクエストで代替ビューに対して呼び出されることはありません。

データの再構成

次のデータ構造について考察します。ここでは、PROD_CODE がインデックスフィールドです。



次のリクエストを発行して PROD_CODE を含むセグメントを階層の最上位へ移動し、そのセグメントのデータへのアクセスを高速にすることができます。

```
TABLE FILE SALES.PROD_CODE
"SALES OF B10 DISTRIBUTED BY AREA"
```

```
SUM UNIT_SOLD AND RETAIL_PRICE
BY AREA
WHERE PROD_CODE EQ 'B10'
ON TABLE COLUMN-TOTAL
END
```

ibi FOCUS データソースの検索速度最適化

AUTOPATH パラメータを ON に設定すると、最適化された検索パスが動的に選択されます。最下位の検索セグメントがエントリポイントになります。これと同等の代替ビュー構文は次のとおりです。

```
TABLE FILE filename.fieldname
```

説明

fieldname

これはインデックス付きではありません。*fieldname* が存在するセグメントから検索を開始します。

システムは、リクエストで参照するフィールド、およびデータソース構造を分析することにより、最適化された検索が適切かどうか決定します。AUTOPATH パラメータについての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

自動インデックス検索

自動インデックス検索 (AUTOINDEX) により、FOCUS データソースでのデータ検索速度が最適化されます。自動インデックス検索を活用するためには、TABLE リクエストで参照する最上位のインデックスフィールドに等価テストまたは範囲テストを含める必要があります。

次の場合、この方法はサポートされません。

- 範囲テストがパック 10 進数のデータ値に適用される。
- リクエストが代替ビュー (TABLE FILE *filename.fieldname*) を指定する。
- リクエストに BY HIGHEST または BY LOWEST コードが含まれる。

AUTOINDEX についての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

インデックス検索の使用

```
SET AUTOINDEX = {ON|OFF}
```

説明

ON

インデックスデータ検索を使用して、検索速度を最適化します。このリクエストには、参照する最上位セグメントのインデックスフィールドに等価テストまたは範囲テストを含める必要があります。デフォルト値は ON です。

OFF

リクエストでインデックスビュー (TABLE FILE *filename.indexed_fieldname*) が指定されず、*indexed_fieldname* に等価テストが含まれない限り、連続データ検索を使用します。この場合、インデックスデータ検索は自動的に実行されます。

インデックス検索使用上の注意

- TABLE リクエストに代替ビュー (TABLE FILE *filename.fieldname*) が含まれるときは、AUTOINDEX は呼び出されません。
- TABLE リクエストに BY HIGHEST または BY LOWEST 句が含まれている場合は、AUTOINDEX が ON でもインデックス検索は実行されません。
- リクエストでインデックスビュー (TABLE FILE *filename.indexed_fieldname*) が指定するときに、インデックス検索が実行される条件は次のとおりです。
    - AUTOINDEX が OFF で、リクエストにインデックスフィールドの等価テストが含まれる。
    - AUTOINDEX が ON で、リクエストにインデックスフィールドに対する等価テストまたは範囲 (FROM ... TO) テストが含まれる。

インデックス検索の使用

次のマスターファイルは、以下のインデックス検索の例で参照されています。

```
FILENAME=SALES,SUFFIX=FOC,
  SEGNAME=STOR_SEG,SEGTYPE=S1,
    FIELDNAME=AREA,ALIAS=LOC,FORMAT=A1,$
  SEGNAME=DATE_SEG,PARENT=STOR_SEG,SEGTYPE=SH1,
    FIELDNAME=DATE,ALIAS=DTE,FORMAT=A4MD, $
  SEGNAME=DEPT,PARENT=DATE_SEG,SEGTYPE=S1,
    FIELDNAME=DEPARTMENT,ALIAS=DEPT,FORMAT=A5,FIELDTYPE=I,$
    FIELDNAME=DEPT_CODE,ALIAS=DCODE,FORMAT=A3,FIELDTYPE=I,$
    FIELDNAME=PROD_TYPE,ALIAS=PTYPE,FORMAT=A10,FIELDTYPE=I,$
  SEGNAME=INVENTORY,PARENT=DEPT,SEGTYPE=S1,$
    FIELDNAME=PROD_CODE,ALIAS=PCODE,FORMAT=A3,FIELDTYPE=I,$
    FIELDNAME=UNIT_SOLD,ALIAS=SOLD,FORMAT=I5,$
    FIELDNAME=RETAIL_PRICE,ALIAS=RP,FORMAT=D5.2M,$
    FIELDNAME=DELIVER_AMT,ALIAS=SHIP,FORMAT=I5,$
```

次のプロシジャには、DEPT_CODE および PROD_CODE の等価テストが含まれています。DEPT_CODE は参照セグメントの上位にあるため、インデックス検索に使用されます。

```
SET AUTOINDEX=ON
TABLE FILE SALES
SUM UNIT_SOLD RETAIL_PRICE
IF DEPT_CODE EQ 'H01'
IF PROD_CODE EQ 'B10'
END
```

TABLE リクエストに同一セグメントの複数のインデックスフィールドに対する等価テスト、または範囲テストが含まれる場合、AUTOINDEX は、セグメント内で参照されている 1 つ目のインデックスを検索に使用します。次のストアドプロシジャには、2 つのインデックスフィールドに対する等価テストが含まれています。DEPT_CODE はマスターファイル内で PROD_TYPE の前に位置するため、AUTOINDEX は検索に DEPT_CODE を使用します。

```
SET AUTOINDEX=ON
TABLE FILE SALES
SUM UNIT_SOLD AND RETAIL_PRICE
IF PROD_TYPE EQ 'STEREO'
IF DEPT_CODE EQ 'H01'
END
```

最上位の参照セグメントにないインデックスフィールドに対して等価テスト、または範囲テストを実行すると、インデックス検索は呼び出されません。次の例では、リクエストに STOR_SEG セグメント内の AREA フィールドへの参照が含まれるため、インデックス検索は実行されません。

```
SET AUTOINDEX=ON
TABLE FILE SALES
SUM UNIT_SOLD AND RETAIL_PRICE
BY AREA
IF PROD_CODE EQ 'B10'
IF PROD_TYPE EQ 'STEREO'
END
```

TABLEF によるデータ検索

TABLEF は TABLE コマンドのバリエーションで、表示に必要な順に保存済みで、追加のソートが不要なデータを高速に検索することができます。

TABLEF を使用すると、データソースからの論理的順序でレコードが検索されます。適用される標準リクエスト構文は、次の規則に従います。

- 任意の BY 句は、データソースの論理的順序に対応する必要があります。BY 句は制御区切りを設定するためにのみ使用し、レコードの順序を変更するためには使用しません。
- ACROSS 句は使用できません。
- 複数の表示コマンドは使用できません。単一の表示コマンドのみ使用できます。
- レポートの実行後、RETYPE、HOLD、および SAVE は使用できません。ただし、ON TABLE HOLD または ON TABLE SAVE をリクエストの一部として含めることで、抽出ファイルを作成することができます。
- NOSPLIT は TABLEF コマンドとともに使用することはできず、FOC037 エラーメッセージが生成されます。
- リクエストおよびデータの両方の自然ソート順序が同一の場合、TABLEF を HOLD ファイルおよび他の非 FOCUS データソースと併用することができます。
- DST. 演算接頭語は使用できません。
- BORDER スタイルは、TABLEF ではサポートされません。
- TABLEF は、SQUEEZE ではサポートされません。

高速テーブル検索による表示

EMPLOYEE データソースから HOLD ファイルを作成し、CURR_SAL、LAST_NAME および FIRST_NAME フィールドでソートしてある場合、次のTABLEF リクエストを発行することができます。

```
TABLEF FILE HOLD
PRINT CURR_SAL AND LAST_NAME AND FIRST_NAME
END
```

マスターファイルと構造図

この付録では、マニュアルの事例で使用するサンプルデータソースの記述およびその構造図について説明します。

EMPLOYEE データソース

EMPLOYEE データソースには、企業の従業員に関するサンプルデータが保存されています。次のセグメントが記述されています。

EMPINFO

従業員の ID 番号、氏名、役職が格納されています。

FUNDTRAN

従業員の自動振込み口座情報が格納されています。これは、ユニークセグメントです。

PAYINFO

従業員の給与履歴が格納されています。

ADDRESS

従業員の自宅の住所が格納されています。

SALINFO

従業員の毎月の給与額が格納されています。

DEDUCT

従業員の毎月の給与控除額が格納されています。

EMPLOYEE データソースには、JOBFILE ファイルおよび EDUCFILE ファイルに属するクロスリファレンスセグメントも含まれています。この付録では、これらのファイルについても説明します。ここでは、次のセグメントが記述されています。

JOBSEG (JOBFILE から)

各従業員に適用する役職名を記述します。

SKILLSEG (JOBFILE から)

各役職に要求されるスキルを列記します。

SECSEG (JOBFILE から)

各役職に要求される機密区分を指定します。

ATTNDSEG (EDUCFILE から)

従業員が社内コースを受講した日付を列記します。

COURSEG (EDUCFILE から)

従業員が受講したコース名を列記します。

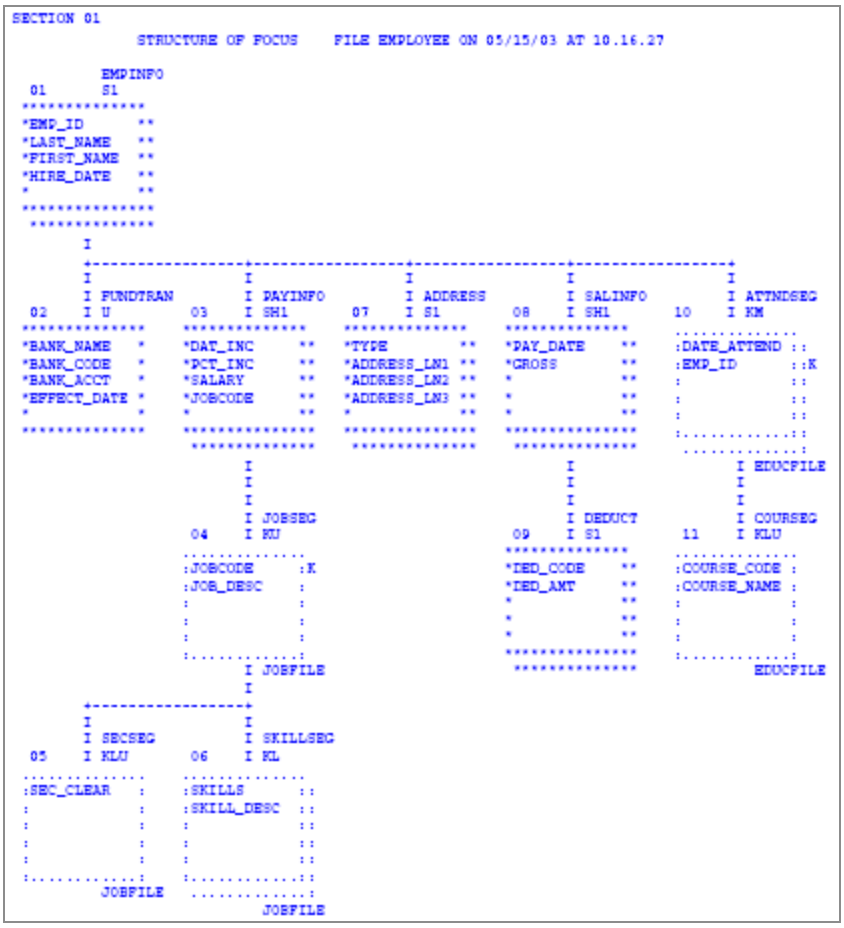
EMPLOYEE マスターファイル

```
FILENAME=EMPLOYEE, SUFFIX=FOC
 SEGNAME=EMPINFO,  SEGTYPE=S1
  FIELDNAME=EMP_ID,        ALIAS=EID,    FORMAT=A9,          $
  FIELDNAME=LAST_NAME,     ALIAS=LN,     FORMAT=A15,         $
  FIELDNAME=FIRST_NAME,    ALIAS=FN,     FORMAT=A10,         $
  FIELDNAME=HIRE_DATE,     ALIAS=HDT,    FORMAT=I6YMD,       $
  FIELDNAME=DEPARTMENT,    ALIAS=DPT,    FORMAT=A10,         $
  FIELDNAME=CURR_SAL,      ALIAS=CSAL,   FORMAT=D12.2M,      $
  FIELDNAME=CURR_JOBCODE,  ALIAS=CJC,    FORMAT=A3,          $
  FIELDNAME=ED_HRS,        ALIAS=OJT,    FORMAT=F6.2,        $
 SEGNAME=FUNDTRAN,  SEGTYPE=U,   PARENT=EMPINFO
  FIELDNAME=BANK_NAME,     ALIAS=BN,     FORMAT=A20,         $
  FIELDNAME=BANK_CODE,     ALIAS=BC,     FORMAT=I6S,         $
  FIELDNAME=BANK_ACCT,     ALIAS=BA,     FORMAT=I9S,         $
  FIELDNAME=EFFECT_DATE,   ALIAS=EDATE,  FORMAT=I6YMD,       $
 SEGNAME=PAYINFO,   SEGTYPE=SH1, PARENT=EMPINFO
  FIELDNAME=DAT_INC,       ALIAS=DI,     FORMAT=I6YMD,       $
  FIELDNAME=PCT_INC,       ALIAS=PI,     FORMAT=F6.2,        $
  FIELDNAME=SALARY,        ALIAS=SAL,    FORMAT=D12.2M,      $
  FIELDNAME=JOBCODE,       ALIAS=JBC,    FORMAT=A3,          $
```

```
SEGNAME=ADDRESS,   SEGTYPE=S1,  PARENT=EMPINFO
 FIELDNAME=TYPE,          ALIAS=AT,     FORMAT=A4,          $
 FIELDNAME=ADDRESS_LN1,   ALIAS=LN1,    FORMAT=A20,         $
 FIELDNAME=ADDRESS_LN2,   ALIAS=LN2,    FORMAT=A20,         $
 FIELDNAME=ADDRESS_LN3,   ALIAS=LN3,    FORMAT=A20,         $
 FIELDNAME=ACCTNUMBER,    ALIAS=ANO,    FORMAT=I9L,         $
SEGNAME=SALINFO,   SEGTYPE=SH1, PARENT=EMPINFO
 FIELDNAME=PAY_DATE,      ALIAS=PD,     FORMAT=I6YMD,       $
 FIELDNAME=GROSS,         ALIAS=MO_PAY, FORMAT=D12.2M,      $
SEGNAME=DEDUCT,    SEGTYPE=S1,  PARENT=SALINFO
 FIELDNAME=DED_CODE,      ALIAS=DC,     FORMAT=A4,          $
 FIELDNAME=DED_AMT,       ALIAS=DA,     FORMAT=D12.2M,      $
SEGNAME=JOBSEG,    SEGTYPE=KU,  PARENT=PAYINFO, CRFILE=JOBFILE,
 CRKEY=JOBCODE,$
SEGNAME=SECSEG,    SEGTYPE=KLU, PARENT=JOBSEG,   CRFILE=JOBFILE, $
SEGNAME=SKILLSEG,  SEGTYPE=KL,  PARENT=JOBSEG,   CRFILE=JOBFILE, $
SEGNAME=ATTNDSEG,  SEGTYPE=KM,  PARENT=EMPINFO,  CRFILE=EDUCFILE,
 CRKEY=EMP_ID,$
SEGNAME=COURSEG,   SEGTYPE=KLU, PARENT=ATTNDSEG, CRFILE=EDUCFILE,$
```

EMPLOYEE 構造図

EMPLOYEE 構造図は次のとおりです。



JOBFILE データソース

JOBFILE データソースには、企業の役職に関するサンプルデータが保存されています。次のセグメントが記述されています。

JOBSEG

各従業員に適用する役職名を記述します。このセグメントの JOBCODE フィールドはインデックスフィールドです。

SKILLSEG

各役職に要求されるスキルを列記します。

SECSEG

各役職に要求される機密区分を指定します (必要な場合)。これは、ユニークセグメントです。

JOBFILE マスターファイル

```
FILENAME=JOBFILE,  SUFFIX=FOC
 SEGNAME=JOBSEG,   SEGTYPE=S1
  FIELDNAME=JOBCODE,    ALIAS=JC, FORMAT=A3,    INDEX=I,$
  FIELDNAME=JOB_DESC,   ALIAS=JD, FORMAT=A25           ,$
 SEGNAME=SKILLSEG, SEGTYPE=S1,  PARENT=JOBSEG
  FIELDNAME=SKILLS,     ALIAS=,   FORMAT=A4            ,$
  FIELDNAME=SKILL_DESC, ALIAS=SD, FORMAT=A30           ,$
 SEGNAME=SECSEG,   SEGTYPE=U,   PARENT=JOBSEG
  FIELDNAME=SEC_CLEAR,  ALIAS=SC, FORMAT=A6            ,$
```

JOBFILE 構造図

```
SECTION 01
              STRUCTURE OF FOCUS    FILE JOBFILE ON 05/15/03 AT 14.40.06

          JOBSEG
  01      S1
 **************
 *JOBCODE     **I
 *JOB_DESC    **
 *            **
 *            **
 *            **
 ***************
  **************
        I
        +-----------------+
        I                 I
        I SECSEG          I SKILLSEG
  02    I U         03    I S1
 **************    *************
 *SEC_CLEAR   *    *SKILLS     **
 *            *    *SKILL_DESC **
 *            *    *           **
 *            *    *           **
 *            *    *           **
 **************    **************
                    *************
```

EDUCFILE データソース

EDUCFILE データソースには、企業の社内コースに関するサンプルデータが保存されています。次のセグメントが記述されています。

COURSEG

各コース名が格納されています。

ATTNDSEG

各コースを受講した従業員を特定します。このセグメントの両方のフィールドがキーフィールドです。このセグメントの EMP_ID フィールドはインデックスフィールドです。

EDUCFILE マスターファイル

```
FILENAME=EDUCFILE, SUFFIX=FOC
 SEGNAME=COURSEG,  SEGTYPE=S1
  FIELDNAME=COURSE_CODE,  ALIAS=CC,   FORMAT=A6,           $
  FIELDNAME=COURSE_NAME,  ALIAS=CD,   FORMAT=A30,          $
 SEGNAME=ATTNDSEG,  SEGTYPE=SH2,  PARENT=COURSEG
  FIELDNAME=DATE_ATTEND,  ALIAS=DA,   FORMAT=I6YMD,        $
  FIELDNAME=EMP_ID,       ALIAS=EID,  FORMAT=A9, INDEX=I,  $
```

EDUCFILE 構造図

```
SECTION 01
              STRUCTURE OF FOCUS    FILE EDUCFILE ON 05/15/03 AT 14.45.44

          COURSEG
  01      S1
 **************
 *COURSE_CODE **
 *COURSE_NAME **
 *            **
 *            **
 *            **
 ***************
```

```
  **************
        I
        I
        I
        I ATTNDSEG
  02    I SH2
 **************
 *DATE_ATTEND **
 *EMP_ID      **I
 *            **
 *            **
 *            **
 ***************
  **************
```

SALES データソース

SALES データソースには、提携したチェーン店を持つ酪農企業に関するサンプルデータが保存されています。次のセグメントが記述されています。

STOR_SEG

製品を購入する店舗を列記します。

DAT_SEG

在庫日データが格納されています。

PRODUCT

各製品の日付別の売上データが格納されています。PROD_CODE フィールドはインデックスフィールドです。RETURNS フィールドと DAMAGED フィールドには、MISSING=ON 属性が指定されています。

SALES マスターファイル

```
FILENAME=KSALES,   SUFFIX=FOC
 SEGNAME=STOR_SEG, SEGTYPE=S1
  FIELDNAME=STORE_CODE,  ALIAS=SNO,   FORMAT=A3,   $
```

```
  FIELDNAME=CITY,        ALIAS=CTY,   FORMAT=A15,  $
  FIELDNAME=AREA,        ALIAS=LOC,   FORMAT=A1,   $
 SEGNAME=DATE_SEG, PARENT=STOR_SEG, SEGTYPE=SH1,
  FIELDNAME=DATE,        ALIAS=DTE,   FORMAT=A4MD, $
 SEGNAME=PRODUCT,  PARENT=DATE_SEG, SEGTYPE=S1,
  FIELDNAME=PROD_CODE,   ALIAS=PCODE, FORMAT=A3,   FIELDTYPE=I,$
  FIELDNAME=UNIT_SOLD,   ALIAS=SOLD,  FORMAT=I5,   $
  FIELDNAME=RETAIL_PRICE,ALIAS=RP,    FORMAT=D5.2M,$
  FIELDNAME=DELIVER_AMT, ALIAS=SHIP,  FORMAT=I5,   $
  FIELDNAME=OPENING_AMT, ALIAS=INV,   FORMAT=I5,   $
  FIELDNAME=RETURNS,     ALIAS=RTN,   FORMAT=I3,   MISSING=ON,$
  FIELDNAME=DAMAGED,     ALIAS=BAD,   FORMAT=I3,   MISSING=ON,$
```

SALES 構造図

```
SECTION 01
             STRUCTURE OF FOCUS    FILE SALES ON 05/15/03 AT 14.50.28

          STOR_SEG
  01      S1
 **************
 *STORE_CODE  **
 *CITY        **
 *AREA        **
 *            **
 *            **
 ***************
  **************
        I
        I
        I
        I DATE_SEG
  02    I SH1
 **************
 *DATE        **
 *            **
 *            **
 *            **
 *            **
 ***************
  **************
        I
        I
```

```
        I
        I PRODUCT
 03     I S1
**************
*PROD_CODE   **I
*UNIT_SOLD   **
*RETAIL_PRICE**
*DELIVER_AMT **
*            **
***************
 **************
```

CAR データソース

CAR データソースには、車の仕様および売上情報に関するサンプルデータが保存されています。次のセグメントが記述されています。

ORIGIN

車の製造国を列記します。COUNTRY フィールドはインデックスフィールドです。

COMP

車の名前が格納されています。

CARREC

車の型式が格納されています。

BODY

車体の種類、座席、ディーラー、小売価格、売上台数を列記します。

SPECS

車の仕様を列記します。これは、ユニークセグメントです。

WARANT

保証のタイプを列記します。

EQUIP

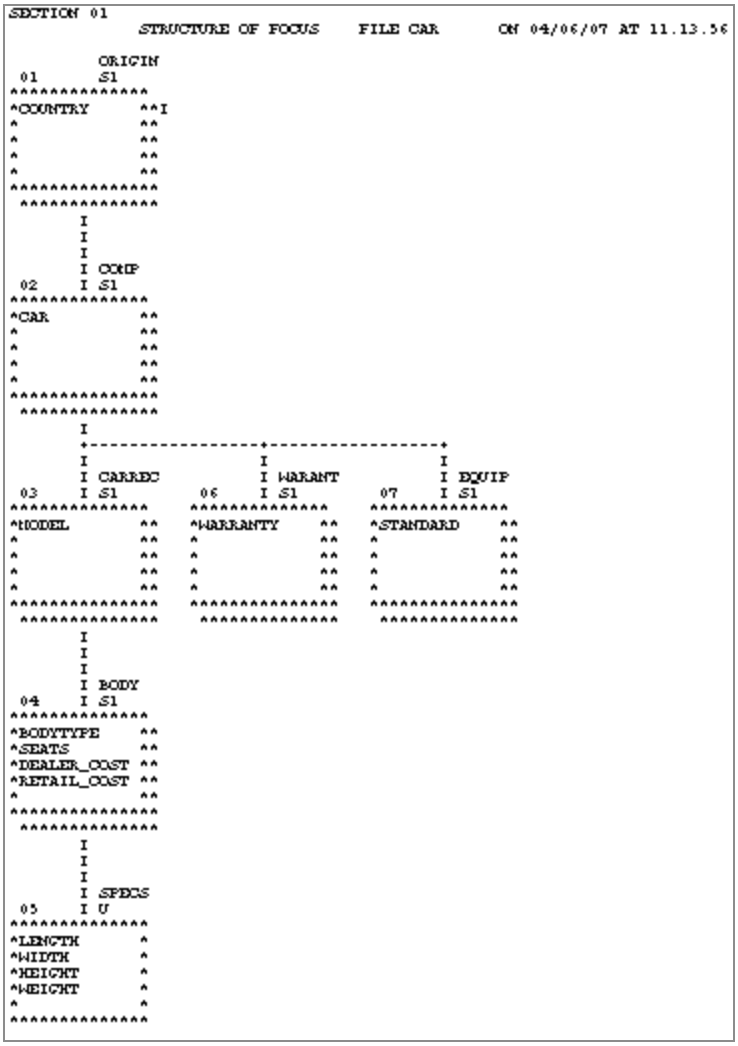
標準装備を列記します。

CAR マスターファイルでは、「ALIAS=」という記述をせずに ALIAS が指定されています。

CAR マスターファイル

```
FILENAME=CAR,SUFFIX=FOC
 SEGNAME=ORIGIN,SEGTYPE=S1
  FIELDNAME=COUNTRY,COUNTRY,A10,FIELDTYPE=I,$
 SEGNAME=COMP,SEGTYPE=S1,PARENT=ORIGIN
  FIELDNAME=CAR,CARS,A16,$
 SEGNAME=CARREC,SEGTYPE=S1,PARENT=COMP
  FIELDNAME=MODEL,MODEL,A24,$
 SEGNAME=BODY,SEGTYPE=S1,PARENT=CARREC
  FIELDNAME=BODYTYPE,TYPE,A12,$
  FIELDNAME=SEATS,SEAT,I3,$
  FIELDNAME=DEALER_COST,DCOST,D7,$
  FIELDNAME=RETAIL_COST,RCOST,D7,$
  FIELDNAME=SALES,UNITS,I6,$
 SEGNAME=SPECS,SEGTYPE=U,PARENT=BODY
  FIELDNAME=LENGTH,LEN,D5,$
  FIELDNAME=WIDTH,WIDTH,D5,$
  FIELDNAME=HEIGHT,HEIGHT,D5,$
  FIELDNAME=WEIGHT,WEIGHT,D6,$
  FIELDNAME=WHEELBASE,BASE,D6.1,$
  FIELDNAME=FUEL_CAP,FUEL,D6.1,$
  FIELDNAME=BHP,POWER,D6,$
  FIELDNAME=RPM,RPM,I5,$
  FIELDNAME=MPG,MILES,D6,$
  FIELDNAME=ACCEL,SECONDS,D6,$
 SEGNAME=WARANT,SEGTYPE=S1,PARENT=COMP
  FIELDNAME=WARRANTY,WARR,A40,$
 SEGNAME=EQUIP,SEGTYPE=S1,PARENT=COMP
  FIELDNAME=STANDARD,EQUIP,A40,$
```

CAR 構造図



LEDGER データソース

LEDGER データソースには、サンプルの会計データが保存されています。このデータソースは、「TOP」という 1 つのセグメントで構成されています。このデータソースは、当初は財務レポートの事例として指定されたものです。このマスターファイルのフィールドにはエイリアスは存在せず、カンマ (,) がプレースホルダとして使用されています。

LEDGER マスターファイル

```
FILENAME=LEDGER, SUFFIX=FOC,$
 SEGNAME=TOP,    SEGTYPE=S2,$
  FIELDNAME=YEAR    ,  , FORMAT=A4, $
  FIELDNAME=ACCOUNT,  , FORMAT=A4, $
  FIELDNAME=AMOUNT ,  , FORMAT=I5C,$
```

LEDGER 構造図

```
SECTION 01
            STRUCTURE OF FOCUS    FILE LEDGER   ON 05/15/03 AT 15.17.08

          TOP
  01      S2
 **************
 *YEAR         **
 *ACCOUNT      **
 *AMOUNT       **
 *             **
 *             **
 ***************
  **************
```

FINANCE データソース

FINANCE データソースには、バランスシート用のサンプル財務データが保存されています。このデータソースは、「TOP」という 1 つのセグメントで構成されています。このデータソースは、当初は財務レポートの事例として指定されたものです。このマスターファイルのフィールドにはエイリアスは存在せず、カンマ (,) がプレースホルダとして使用されています。

FINANCE マスターファイル

```
FILENAME=FINANCE, SUFFIX=FOC,$
 SEGNAME=TOP,     SEGTYPE=S2,$
  FIELDNAME=YEAR   ,  , FORMAT=A4,  $
  FIELDNAME=ACCOUNT,  , FORMAT=A4,  $
  FIELDNAME=AMOUNT ,  , FORMAT=D12C,$
```

FINANCE 構造図

```
SECTION 01
            STRUCTURE OF FOCUS    FILE FINANCE  ON 05/15/03 AT 15.17.08

          TOP
  01      S2
 **************
 *YEAR         **
 *ACCOUNT      **
 *AMOUNT       **
 *             **
 *             **
 ***************
  **************
```

REGION データソース

REGION データソースには、国の東部地区および西部地区のサンプル会計データが保存されています。このデータソースは、「TOP」という 1 つのセグメントで構成されています。このデータソースは、当初は財務レポートの事例として指定されたものです。このマスターファイルのフィールドにはエイリアスは存在せず、カンマ (,) がプレースホルダとして使用されています。

REGION マスターファイル

```
FILENAME=REGION, SUFFIX=FOC,$
 SEGNAME=TOP,    SEGTYPE=S1,$
  FIELDNAME=ACCOUNT,  , FORMAT=A4, $
  FIELDNAME=E_ACTUAL, , FORMAT=I5C,$
  FIELDNAME=E_BUDGET, , FORMAT=I5C,$
  FIELDNAME=W_ACTUAL, , FORMAT=I5C,$
  FIELDNAME=W_BUDGET, , FORMAT=I5C,$
```

REGION 構造図

```
SECTION 01
            STRUCTURE OF FOCUS    FILE REGION   ON 05/15/03 AT 15.18.48

          TOP
  01      S1
 **************
 *ACCOUNT     **
 *E_ACTUAL    **
 *E_BUDGET    **
 *W_ACTUAL    **
 *            **
 ***************
  **************
```

EMPDATA データソース

EMPDATA データソースには、企業の従業員に関するサンプルデータが保存されています。このデータソースは、「EMPDATA」という 1 つのセグメントで構成されています。PIN フィールドはインデックスフィールドです。AREA フィールドは一時項目です。

EMPDATA マスターファイル

```
FILENAME=EMPDATA, SUFFIX=FOC
 SEGNAME=EMPDATA, SEGTYPE=S1
  FIELDNAME=PIN,          ALIAS=ID,       FORMAT=A9,  INDEX=I,    $
  FIELDNAME=LASTNAME,     ALIAS=LN,       FORMAT=A15,             $
  FIELDNAME=FIRSTNAME,    ALIAS=FN,       FORMAT=A10,             $
  FIELDNAME=MIDINITIAL,   ALIAS=MI,       FORMAT=A1,              $
  FIELDNAME=DIV,          ALIAS=CDIV,     FORMAT=A4,              $
  FIELDNAME=DEPT,         ALIAS=CDEPT,    FORMAT=A20,             $
  FIELDNAME=JOBCLASS,     ALIAS=CJCLAS,   FORMAT=A8,              $
  FIELDNAME=TITLE,        ALIAS=CFUNC,    FORMAT=A20,             $
  FIELDNAME=SALARY,       ALIAS=CSAL,     FORMAT=D12.2M,          $
  FIELDNAME=HIREDATE,     ALIAS=HDAT,     FORMAT=YMD,             $
$
DEFINE AREA/A13=DECODE DIV (NE 'NORTH EASTERN' SE 'SOUTH EASTERN'
CE 'CENTRAL' WE 'WESTERN' CORP 'CORPORATE' ELSE 'INVALID AREA');$
```

EMPDATA 構造図

```
SECTION 01
            STRUCTURE OF FOCUS    FILE EMPDATA  ON 05/15/03 AT 14.49.09

          EMPDATA
  01      S1
 **************
 *PIN          **I
 *LASTNAME     **
 *FIRSTNAME    **
 *MIDINITIAL   **
 *             **
 ***************
  **************
```

TRAINING データソース

TRAINING データソースには、従業員のトレーニングコースに関するサンプルデータが保存されています。このデータソースは、「TRAINING」という 1 つのセグメントで構成されて

います。PIN フィールドはインデックスフィールドです。EXPENSES、GRADE、LOCATION の各フィールドには、MISSING=ON 属性が指定されています。

TRAINING マスターファイル

```
FILENAME=TRAINING, SUFFIX=FOC
 SEGNAME=TRAINING, SEGTYPE=SH3
  FIELDNAME=PIN,          ALIAS=ID,     FORMAT=A9,   INDEX=I,    $
  FIELDNAME=COURSESTART,  ALIAS=CSTART, FORMAT=YMD,              $
  FIELDNAME=COURSECODE,   ALIAS=CCOD,   FORMAT=A7,               $
  FIELDNAME=EXPENSES,     ALIAS=COST,   FORMAT=D8.2, MISSING=ON  $
  FIELDNAME=GRADE,        ALIAS=GRA,    FORMAT=A2,   MISSING=ON, $
  FIELDNAME=LOCATION,     ALIAS=LOC,    FORMAT=A6,   MISSING=ON, $
```

TRAINING 構造図

```
SECTION 01
            STRUCTURE OF FOCUS    FILE TRAINING ON 05/15/03 AT 14.51.28

          TRAINING
  01      SH3
 **************
 *PIN         **I
 *COURSESTART **
 *COURSECODE  **
 *EXPENSES    **
 *            **
 ***************
  **************
```

COURSE データソース

COURSE データソースには、教育コースに関するサンプルデータが保存されています。このデータソースは、CRSELIST セグメントのみで構成されています。

COURSE マスターファイル

```
FILENAME=COURSE,   SUFFIX=FOC
 SEGNAME=CRSELIST,  SEGTYPE=S1
  FIELDNAME=COURSECODE,  ALIAS=CCOD,   FORMAT=A7,   INDEX=I,    $
  FIELDNAME=CTITLE,      ALIAS=COURSE, FORMAT=A35,              $
  FIELDNAME=SOURCE,      ALIAS=ORG,    FORMAT=A35,              $
  FIELDNAME=CLASSIF,     ALIAS=CLASS,  FORMAT=A10,              $
  FIELDNAME=TUITION,     ALIAS=FEE,    FORMAT=D8.2, MISSING=ON, $
  FIELDNAME=DURATION,    ALIAS=DAYS,   FORMAT=A3,   MISSING=ON, $
  FIELDNAME=DESCRIPTN1,  ALIAS=DESC1,  FORMAT=A40,              $
  FIELDNAME=DESCRIPTN2,  ALIAS=DESC2,  FORMAT=A40,              $
  FIELDNAME=DESCRIPTN2,  ALIAS=DESC3,  FORMAT=A40,              $
```

COURSE 構造図

```
SECTION 01
             STRUCTURE OF FOCUS    FILE COURSE   ON 05/15/03 AT 12.26.05

         CRSELIST
 01      S1
**************
*COURSECODE  **I
*CTITLE      **
*SOURCE      **
*CLASSIF     **
*            **
***************
 **************
```

JOBHIST データソース

JOBHIST データソースには、従業員の職務に関する情報が保存されています。PIN および JOBUSTART フィールドの両方がキーフィールドです。PIN フィールドはインデックス フィールドです。

JOBHIST マスターファイル

```
FILENAME=JOBHIST, SUFFIX=FOC
SEGNAME=JOBHIST, SEGTYPE=SH2
 FIELDNAME=PIN,          ALIAS=ID,       FORMAT=A9,      INDEX=I ,$
 FIELDNAME=JOBSTART,     ALIAS=SDAT,     FORMAT=YMD,              $
 FIELDNAME=JOBCLASS,     ALIAS=JCLASS,   FORMAT=A8,               $
 FIELDNAME=FUNCTITLE,    ALIAS=FUNC,     FORMAT=A20,              $
```

JOBHIST 構造図

```
SECTION 01
     STRUCTURE OF FOCUS    FILE JOBHIST  ON 01/22/08 AT 16.23.46
    JOBHIST
  01      SH2
 **************
 *PIN         **I
 *JOBSTART    **
 *JOBCLASS    **
 *FUNCTITLE   **
 *            **
 ***************
  **************
```

JOBLIST データソース

JOBLIST データソースには、職務に関する情報が保存されています。JOBCLASS フィールドはインデックスフィールドです。

JOBLIST マスターファイル

```
FILENAME=JOBLIST, SUFFIX=FOC
SEGNAME=JOBSEG,   SEGTYPE=S1
```

```
 FIELDNAME=JOBCLASS,     ALIAS=JCLASS,   FORMAT=A8,    INDEX=I ,$
 FIELDNAME=CATEGORY,     ALIAS=JGROUP,   FORMAT=A25,            $
 FIELDNAME=JOBDESC,      ALIAS=JDESC,    FORMAT=A40,            $
 FIELDNAME=LOWSAL,       ALIAS=LSAL,     FORMAT=D12.2M,         $
 FIELDNAME=HIGHSAL,      ALIAS=HSAL,     FORMAT=D12.2M,         $
DEFINE GRADE/A2=EDIT  (JCLASS,'$$$99');$
DEFINE LEVEL/A25=DECODE GRADE (08 'GRADE 8' 09 'GRADE 9' 10
'GRADE 10' 11 'GRADE 11' 12 'GRADE 12' 13 'GRADE 13' 14 'GRADE 14');$
```

JOBLIST 構造図

```
SECTION 01
              STRUCTURE OF FOCUS    FILE JOBLIST  ON 01/22/08 AT 16.24.52
          JOBSEG
  01      S1
 **************
 *JOBCLASS    **I
 *CATEGORY    **
 *JOBDESC     **
 *LOWSAL      **
 *            **
 ***************
  **************
```

LOCATOR データソース

LOCATOR データソースには、従業員の勤務場所および電話番号に関する情報が保存されています。PIN フィールドはインデックスフィールドです。

LOCATOR マスターファイル

```
FILENAME=LOCATOR, SUFFIX=FOC
SEGNAME=LOCATOR,   SEGTYPE=S1,
 FIELDNAME=PIN,           ALIAS=ID_NO,    FORMAT=A9,   INDEX=I, $
```

```
 FIELDNAME=SITE,          ALIAS=SITE,     FORMAT=A25,          $
 FIELDNAME=FLOOR,         ALIAS=FL,       FORMAT=A3,           $
 FIELDNAME=ZONE,          ALIAS=ZONE,     FORMAT=A2,           $
 FIELDNAME=BUS_PHONE,     ALIAS=BTEL,     FORMAT=A5,           $
```

LOCATOR 構造図

```
SECTION 01
              STRUCTURE OF FOCUS    FILE LOCATOR  ON 01/22/08 AT 16.26.55
          LOCATOR
  01      S1
 **************
 *PIN          **I
 *SITE         **
 *FLOOR        **
 *ZONE         **
 *             **
 ***************
  **************
```

PERSINFO データソース

PERSINFO データソースには、従業員の個人情報が保存されています。PIN フィールドはインデックスフィールドです。

PERSINFO マスターファイル

```
FILENAME=PERSINFO, SUFFIX=FOC
SEGNAME=PERSONAL, SEGTYPE=S1
 FIELDNAME=PIN,           ALIAS=ID,       FORMAT=A9,   INDEX=I,    $
 FIELDNAME=INCAREOF,      ALIAS=ICO,      FORMAT=A35,              $
 FIELDNAME=STREETNO,      ALIAS=STR,      FORMAT=A20,              $
 FIELDNAME=APT,           ALIAS=APT,      FORMAT=A4,               $
 FIELDNAME=CITY,          ALIAS=CITY,     FORMAT=A20,              $
```

```
 FIELDNAME=STATE,         ALIAS=PROV,     FORMAT=A4,              $
 FIELDNAME=POSTALCODE,    ALIAS=ZIP,      FORMAT=A10,             $
 FIELDNAME=COUNTRY,       ALIAS=CTRY,     FORMAT=A15,             $
 FIELDNAME=HOMEPHONE,     ALIAS=TEL,      FORMAT=A10,             $
 FIELDNAME=EMERGENCYNO,   ALIAS=ENO,      FORMAT=A10,             $
 FIELDNAME=EMERGCONTACT, ALIAS=ENAME,     FORMAT=A35,             $
 FIELDNAME=RELATIONSHIP, ALIAS=REL,       FORMAT=A8,              $
 FIELDNAME=BIRTHDATE,     ALIAS=BDAT,     FORMAT=YMD,             $
```

PERSINFO 構造図

```
SECTION 01
     STRUCTURE OF FOCUS    FILE PERSINFO ON 01/22/08 AT 16.27.24
   PERSONAL
  01      S1
 **************
 *PIN          **I
 *INCAREOF     **
 *STREETNO     **
 *APT          **
 *             **
 ***************
  **************
```

SALHIST データソース

SALHIST には、従業員の給与履歴に関するデータが格納されています。PIN フィールドはインデックスフィールドです。PIN および EFFECTDATE フィールドの両方がキーフィールドです。

SALHIST マスターファイル

```
FILENAME=SALHIST,  SUFFIX=FOC
SEGNAME=SLHISTRY,  SEGTYPE=SH2
```

```
 FIELDNAME=PIN,        ALIAS=ID,     FORMAT=A9,     INDEX=I,  $
 FIELDNAME=EFFECTDATE, ALIAS=EDAT,   FORMAT=YMD,              $
 FIELDNAME=OLDSALARY,  ALIAS=OSAL,   FORMAT=D12.2,            $
```

SALHIST 構造図

```
SECTION 01
     STRUCTURE OF FOCUS    FILE SALHIST  ON 01/22/08 AT 16.28.02
    SLHISTRY
  01      SH2
 **************
 *PIN         **I
 *EFFECTDATE  **
 *OLDSALARY   **
 *            **
 *            **
 ***************
  **************
```

VIDEOTRK、MOVIES、ITEMS データソース

VIDEOTRK データソースには、レンタルビデオ事業の顧客、レンタル商品、注文情報に関するサンプルデータが保存されています。このデータソースは、MOVIES データソースまたは ITEMS のデータソースと結合することができます。

VIDEOTRK マスターファイル

```
FILENAME=VIDEOTRK, SUFFIX=FOC
 SEGNAME=CUST,     SEGTYPE=S1
  FIELDNAME=CUSTID,     ALIAS=CIN,          FORMAT=A4,    $
  FIELDNAME=LASTNAME,   ALIAS=LN,           FORMAT=A15,   $
```

```
  FIELDNAME=FIRSTNAME,  ALIAS=FN,           FORMAT=A10,   $
  FIELDNAME=EXPDATE,    ALIAS=EXDAT,        FORMAT=YMD,   $
  FIELDNAME=PHONE,      ALIAS=TEL,          FORMAT=A10,   $
  FIELDNAME=STREET,     ALIAS=STR,          FORMAT=A20,   $
  FIELDNAME=CITY,       ALIAS=CITY,         FORMAT=A20,   $
  FIELDNAME=STATE,      ALIAS=PROV,         FORMAT=A4,    $
  FIELDNAME=ZIP,        ALIAS=POSTAL_CODE,  FORMAT=A9,    $
SEGNAME=TRANSDAT,  SEGTYPE=SH1,  PARENT=CUST
  FIELDNAME=TRANSDATE,  ALIAS=OUTDATE,      FORMAT=YMD,   $
SEGNAME=SALES,     SEGTYPE=S2,   PARENT=TRANSDAT
  FIELDNAME=PRODCODE,   ALIAS=PCOD,         FORMAT=A6,    $
  FIELDNAME=TRANSCODE,  ALIAS=TCOD,         FORMAT=I3,    $
  FIELDNAME=QUANTITY,   ALIAS=NO,           FORMAT=I3S,   $
  FIELDNAME=TRANSTOT,   ALIAS=TTOT,         FORMAT=F7.2S, $
SEGNAME=RENTALS,   SEGTYPE=S2,   PARENT=TRANSDAT
  FIELDNAME=MOVIECODE,  ALIAS=MCOD,         FORMAT=A6, INDEX=I, $
  FIELDNAME=COPY,       ALIAS=COPY,         FORMAT=I2,    $
  FIELDNAME=RETURNDATE, ALIAS=INDATE,       FORMAT=YMD,   $
  FIELDNAME=FEE,        ALIAS=FEE,          FORMAT=F5.2S, $
```

VIDEOTRK 構造図

```
SECTION 01
             STRUCTURE OF FOCUS    FILE VIDEOTRK ON 05/15/03 AT 12.25.19

         CUST
 01      S1
**************
*CUSTID      **
*LASTNAME    **
*FIRSTNAME   **
*EXPDATE     **
*            **
***************
 **************
       I
       I
       I
       I TRANSDAT
 02    I SH1
**************
*TRANSDATE   **
*            **
```

```
*            **
*            **
*            **
***************
 **************
       I
       +-----------------+
       I                 I
       I SALES           I RENTALS
 03    I S2        04    I S2
**************    **************
*PRODCODE    **   *MOVIECODE   **I
*TRANSCODE   **   *COPY        **
*QUANTITY    **   *RETURNDATE  **
*TRANSTOT    **   *FEE         **
*            **   *            **
***************   ***************
 **************    **************
```

MOVIES マスターファイル

```
FILENAME=MOVIES,    SUFFIX=FOC
 SEGNAME=MOVINFO,   SEGTYPE=S1
  FIELDNAME=MOVIECODE,   ALIAS=MCOD,  FORMAT=A6, INDEX=I, $
  FIELDNAME=TITLE,       ALIAS=MTL,   FORMAT=A39,  $
  FIELDNAME=CATEGORY,    ALIAS=CLASS, FORMAT=A8,   $
  FIELDNAME=DIRECTOR,    ALIAS=DIR,   FORMAT=A17,  $
  FIELDNAME=RATING,      ALIAS=RTG,   FORMAT=A4,   $
  FIELDNAME=RELDATE,     ALIAS=RDAT,  FORMAT=YMD,  $
  FIELDNAME=WHOLESALEPR, ALIAS=WPRC,  FORMAT=F6.2, $
  FIELDNAME=LISTPR,      ALIAS=LPRC,  FORMAT=F6.2, $
  FIELDNAME=COPIES,      ALIAS=NOC,   FORMAT=I3,   $
```

MOVIES 構造図

```
SECTION 01
             STRUCTURE OF FOCUS    FILE MOVIES   ON 05/15/03 AT 12.26.05
```

```
         MOVINFO
 01      S1
**************
*MOVIECODE   **I
*TITLE       **
*CATEGORY    **
*DIRECTOR    **
*            **
***************
 **************
```

ITEMS マスターファイル

```
FILENAME=ITEMS,   SUFFIX=FOC
 SEGNAME=ITMINFO, SEGTYPE=S1
  FIELDNAME=PRODCODE,  ALIAS=PCOD,  FORMAT=A6,  INDEX=I, $
  FIELDNAME=PRODNAME,  ALIAS=PROD,  FORMAT=A20,          $
  FIELDNAME=OURCOST,   ALIAS=WCOST, FORMAT=F6.2,         $
  FIELDNAME=RETAILPR,  ALIAS=PRICE, FORMAT=F6.2,         $
  FIELDNAME=ON_HAND,   ALIAS=NUM,   FORMAT=I5,           $
```

ITEMS 構造図

```
SECTION 01
             STRUCTURE OF FOCUS    FILE ITEMS    ON 05/15/03 AT 12.26.05

         ITMINFO
 01      S1
**************
*PRODCODE    **I
*PRODNAME    **
*OURCOST     **
*RETAILPR    **
*            **
***************
 **************
```

Gotham Grinds データソース

Gotham Grinds は複数のデータソースで構成され、それぞれのデータソースには特選品を扱う企業に関するサンプルデータが保存されています。

- GGDEMOG データソースには、コーヒー、グルメ菓子、ギフトなどの特選品を販売する会社である Gotham Grinds の顧客に関する統計データが保存されています。詳細は、GGDEMOGを参照してください。このデータソースは、DEMOG01 セグメントのみで構成されています。
- GGORDER データソースには、Gotham Grinds の注文情報が保存されています。詳細は、GGORDERを参照してください。このデータソースは、ORDER01 と ORDER02 の 2 つのセグメントで構成されています。
- GGPRODS データソースには、Gotham Grinds の製品情報が保存されています。詳細は、GGPRODSを参照してください。このデータソースは、PRODS01 セグメントのみで構成されています。
- GGSALES データソースには、Gotham Grinds の売上情報が保存されています。詳細は、GGSALESを参照してください。このデータソースは、SALES01 セグメントのみで構成されています。
- GGSTORES データソースには、Gotham Grinds が米国内に持つ 12 店舗の個別情報が保存されています。STORES01 セグメントのみで構成されています。

GGDEMOG マスターファイル

```
FILENAME=GGDEMOG, SUFFIX=FOC
 SEGNAME=DEMOG01, SEGTYPE=S1
  FIELD=ST,       ALIAS=E02, FORMAT=A02, INDEX=I,TITLE='State',
   DESC='State',$
  FIELD=HH,       ALIAS=E03, FORMAT=I09, TITLE='Number of Households',
   DESC='Number of Households',$
  FIELD=AVGHHSZ98,ALIAS=E04, FORMAT=I09, TITLE='Average Household Size',
   DESC='Average Household Size',$
  FIELD=MEDHHI98, ALIAS=E05, FORMAT=I09, TITLE='Median Household Income',
   DESC='Median Household Income',$
  FIELD=AVGHHI98, ALIAS=E06, FORMAT=I09, TITLE='Average Household Income',
   DESC='Average Household Income',$
  FIELD=MALEPOP98,ALIAS=E07, FORMAT=I09, TITLE='Male Population',
   DESC='Male Population',$
  FIELD=FEMPOP98, ALIAS=E08, FORMAT=I09, TITLE='Female Population',
```

```
   DESC='Female Population',$
  FIELD=P15TO1998,ALIAS=E09, FORMAT=I09, TITLE='15 to 19',
   DESC='Population 15 to 19 years old',$
  FIELD=P20TO2998,ALIAS=E10, FORMAT=I09, TITLE='20 to 29',
   DESC='Population 20 to 29 years old',$
  FIELD=P30TO4998,ALIAS=E11, FORMAT=I09, TITLE='30 to 49',
   DESC='Population 30 to 49 years old',$
  FIELD=P50TO6498,ALIAS=E12, FORMAT=I09, TITLE='50 to 64',
   DESC='Population 50 to 64 years old',$
  FIELD=P65OVR98, ALIAS=E13, FORMAT=I09, TITLE='65 and over',
   DESC='Population 65 and over',$
```

GGDEMOG 構造図

```
SECTION 01
     STRUCTURE OF FOCUS    FILE GGDEMOG  ON 05/15/03 AT 12.26.05

         GGDEMOG
 01      S1
**************
*ST           **I
*HH           **
*AVGHHSZ98    **
*MEDHHI98     **
*             **
***************
 **************
```

GGORDER マスターファイル

```
FILENAME=GGORDER, SUFFIX=FOC,$
 SEGNAME=ORDER01, SEGTYPE=S1,$
  FIELD=ORDER_NUMBER, ALIAS=ORDNO1,   FORMAT=I6,  TITLE='Order,Number',
   DESC='Order Identification Number',$
  FIELD=ORDER_DATE,   ALIAS=DATE,     FORMAT=MDY, TITLE='Order,Date',
   DESC='Date order was placed',$
  FIELD=STORE_CODE,   ALIAS=STCD,     FORMAT=A5,  TITLE='Store,Code',
   DESC='Store Identification Code (for order)',$
```

```
  FIELD=PRODUCT_CODE, ALIAS=PCD,      FORMAT=A4,  TITLE='Product,Code',
   DESC='Product Identification Code (for order)',$
  FIELD=QUANTITY,     ALIAS=ORDUNITS, FORMAT=I8,  TITLE='Ordered,Units',
   DESC='Quantity Ordered',$
SEGNAME=ORDER02, SEGTYPE=KU, PARENT=ORDER01, CRFILE=GGPRODS, CRKEY=PCD,
CRSEG=PRODS01  ,$
```

GGORDER 構造図

```
SECTION 01
     STRUCTURE OF FOCUS    FILE GGORDER  ON 05/15/03 AT 16.45.48

          GGORDER
  01      S1
 **************
 *ORDER_NUMBER**
 *ORDER_DATE  **
 *STORE_CODE  **
 *PRODUCT_CODE**
 *            **
 ***************
  **************
        I
        I
        I
        I ORDER02
  02    I KU
 ..............
 :PRODUCT_ID  :K
 :PRODUCT_DESC:
 :VENDOR_CODE :
 :VENDOR_NAME :
 :            :
 :............:
```

GGPRODS マスターファイル

```
FILENAME=GGPRODS, SUFFIX=FOC
 SEGNAME=PRODS01, SEGTYPE=S1
  FIELD=PRODUCT_ID, ALIAS=PCD, FORMAT=A4, INDEX=I, TITLE='Product,Code',
   DESC='Product Identification Code',$
  FIELD=PRODUCT_DESCRIPTION, ALIAS=PRODUCT, FORMAT=A16, TITLE='Product',
   DESC='Product Name',$
  FIELD=VENDOR_CODE, ALIAS=VCD, FORMAT=A4, INDEX=I, TITLE='Vendor ID',
   DESC='Vendor Identification Code',$
  FIELD=VENDOR_NAME, ALIAS=VENDOR, FORMAT=A23, TITLE='Vendor Name',
   DESC='Vendor Name',$
  FIELD=PACKAGE_TYPE, ALIAS=PACK, FORMAT=A7, TITLE='Package',
   DESC='Packaging Style',$
  FIELD=SIZE, ALIAS=SZ, FORMAT=I2, TITLE='Size',
   DESC='Package Size',$
  FIELD=UNIT_PRICE, ALIAS=UNITPR, FORMAT=D7.2, TITLE='Unit,Price',
   DESC='Price for one unit',$
```

GGPRODS 構造図

```
SECTION 01
     STRUCTURE OF FOCUS    FILE GGPRODS  ON 05/15/03 AT 12.26.05

         GGPRODS
 01      S1
**************
*PRODUCT_ID  **I
*PRODUCT_DESC**I
*VENDOR_CODE **
*VENDOR_NAME **
*            **
***************
 **************
```

GGSALES マスターファイル

```
FILENAME=GGSALES, SUFFIX=FOC
 SEGNAME=SALES01, SEGTYPE=S1
  FIELD=SEQ_NO, ALIAS=SEQ, FORMAT=I5, TITLE='Sequence#',
   DESC='Sequence number in database',$
  FIELD=CATEGORY, ALIAS=E02, FORMAT=A11, INDEX=I, TITLE='Category',
   DESC='Product category',$
  FIELD=PCD, ALIAS=E03, FORMAT=A04, INDEX=I, TITLE='Product ID',
   DESC='Product Identification code (for sale)',$
  FIELD=PRODUCT, ALIAS=E04, FORMAT=A16, TITLE='Product',
   DESC='Product name',$
  FIELD=REGION, ALIAS=E05, FORMAT=A11, INDEX=I, TITLE='Region',
   DESC='Region code',$
  FIELD=ST, ALIAS=E06, FORMAT=A02, INDEX=I, TITLE='State',
   DESC='State',$
  FIELD=CITY, ALIAS=E07, FORMAT=A20, TITLE='City',
   DESC='City',$
  FIELD=STCD, ALIAS=E08, FORMAT=A05, INDEX=I, TITLE='Store ID',
   DESC='Store identification code (for sale)',$
  FIELD=DATE, ALIAS=E09, FORMAT=I8YYMD, TITLE='Date',
   DESC='Date of sales report',$
  FIELD=UNITS, ALIAS=E10, FORMAT=I08, TITLE='Unit Sales',
   DESC='Number of units sold',$
  FIELD=DOLLARS, ALIAS=E11, FORMAT=I08, TITLE='Dollar Sales',
   DESC='Total dollar amount of reported sales',$
  FIELD=BUDUNITS, ALIAS=E12, FORMAT=I08, TITLE='Budget Units',
   DESC='Number of units budgeted',$
  FIELD=BUDDOLLARS, ALIAS=E13, FORMAT=I08, TITLE='Budget Dollars',
   DESC='Total sales quota in dollars',$
```

GGSALES 構造図

```
SECTION 01
     STRUCTURE OF FOCUS    FILE GGSALES  ON 05/15/03 AT 12.26.05

         GGSALES
 01      S1
**************
*SEQ_NO      **
*CATEGORY    **I
```

```
*PCD         **I
*PRODUCT     **I
*            **
***************
 **************
```

GGSTORES マスターファイル

```
FILENAME=GGSTORES, SUFFIX=FOC
 SEGNAME=STORES01, SEGTYPE=S1
  FIELD=STORE_CODE, ALIAS=E02, FORMAT=A05, INDEX=I, TITLE='Store ID',
   DESC='Franchisee ID Code',$
  FIELD=STORE_NAME, ALIAS=E03, FORMAT=A23, TITLE='Store Name',
   DESC='Store Name',$
  FIELD=ADDRESS1, ALIAS=E04, FORMAT=A19, TITLE='Contact',
   DESC='Franchisee Owner',$
  FIELD=ADDRESS2, ALIAS=E05, FORMAT=A31, TITLE='Address',
   DESC='Street Address',$
  FIELD=CITY, ALIAS=E06, FORMAT=A22, TITLE='City',
   DESC='City',$
  FIELD=STATE, ALIAS=E07, FORMAT=A02, INDEX=I, TITLE='State',
   DESC='State',$
  FIELD=ZIP, ALIAS=E08, FORMAT=A06, TITLE='Zip Code',
   DESC='Postal Code',$
```

GGSTORES 構造図

```
SECTION 01
     STRUCTURE OF FOCUS    FILE GGSTORES ON 05/15/03 AT 12.26.05

         GGSTORES
 01      S1
**************
*STORE_CODE  **I
*STORE_NAME  **
*ADDRESS1    **
*ADDRESS2    **
*            **
```

```
***************
 **************
```

Century Corp データソース

Century Corp は、世界中の小売店を通して製品を流通する家電製品のメーカーです。この企業は、世界中の工場、倉庫、オフィスに何千人もの従業員を配置しています。従業員の役割は、品質のよい製品とサービスを顧客に提供することです。

Century Corp データソースは複数のデータソースで構成され、これらのデータソースには財務、人材、在庫、注文に関するデータが保存されています。最後の 3 つのデータソースは、勘定科目の階層図と併せて使用するよう設計されています。

- CENTCOMP マスターファイルには、店舗のロケーション情報が格納されています。このデータソースは、COMPINFO セグメントのみで構成されています。
- CENTFIN マスターファイルには、財務情報が格納されています。このデータソースは、ROOT_SEG セグメントのみで構成されています。
- CENTHR マスターファイルには、人事情報が格納されています。このデータソースは、EMPSEG セグメントのみで構成されています。
- CENTINV マスターファイルには、在庫情報が格納されています。このデータソースは、INVINFO セグメントのみで構成されています。
- CENTORD マスターファイルには、注文情報が格納されています。このデータソースは、OINFO、STOSEG、PINFO、INVSEG の 4 つのセグメントで構成されています。
- CENTQA マスターファイルには、障害情報が格納されています。このデータソースは、PROD_SEG、INVSEG、PROB_SEG の 3 つのセグメントで構成されています。
- CENTGL マスターファイルには、勘定科目表が格納されています。GL_ACCOUNT_PARENT は、階層の親フィールドです。GL_ACCOUNT フィールドは、階層フィールドです。GL_ACCOUNT_CAPTION フィールドは、階層フィールドの説明キャプションとして使用することができます。
- CENTSYSF マスターファイルには、詳細レベルの財務データが格納されています。CENTSYSF は、異なる勘定科目体系 (SYS_ACCOUNT) を使用しますが、CENTGL の SYS_ACCOUNT フィールドと結合することができます。このデータは、通例の記号 (費用はプラス、収益はマイナス) を使用しています。

CENTCOMP マスターファイル

```
FILE=CENTCOMP, SUFFIX=FOC, FDFC=19, FYRT=00
  SEGNAME=COMPINFO, SEGTYPE=S1, $
  FIELD=STORE_CODE, ALIAS=SNUM, FORMAT=A6, INDEX=I,
   TITLE='Store Id#:',
   DESCRIPTION='Store Id#', $
  FIELD=STORENAME, ALIAS=SNAME, FORMAT=A20,
   WITHIN=STATE,
   TITLE='Store,Name:',
   DESCRIPTION='Store Name', $
  FIELD=STATE, ALIAS=STATE, FORMAT=A2,
   WITHIN=PLANT,
   TITLE='State:',
   DESCRIPTION=State, $
  DEFINE REGION/A5=DECODE STATE ('AL' 'SOUTH' 'AK' 'WEST' 'AR' 'SOUTH'
  'AZ' 'WEST' 'CA' 'WEST' 'CO' 'WEST' 'CT' 'EAST'
  'DE' 'EAST' 'DC' 'EAST' 'FL' 'SOUTH' 'GA' 'SOUTH' 'HI' 'WEST'
  'ID' 'WEST' 'IL' 'NORTH' 'IN' 'NORTH' 'IA' 'NORTH'
  'KS' 'NORTH' 'KY' 'SOUTH' 'LA' 'SOUTH' 'ME' 'EAST' 'MD' 'EAST'
  'MA' 'EAST' 'MI' 'NORTH' 'MN' 'NORTH' 'MS' 'SOUTH' 'MT' 'WEST'
  'MO' 'SOUTH' 'NE' 'WEST'  'NV' 'WEST' 'NH' 'EAST' 'NJ' 'EAST'
  'NM' 'WEST' 'NY' 'EAST' 'NC' 'SOUTH' 'ND' 'NORTH' 'OH' 'NORTH'
  'OK' 'SOUTH' 'OR' 'WEST' 'PA' 'EAST' 'RI' 'EAST' 'SC' 'SOUTH'
  'SD' 'NORTH' 'TN' 'SOUTH' 'TX' 'SOUTH' 'UT' 'WEST' 'VT' 'EAST'
  'VA' 'SOUTH' 'WA' 'WEST' 'WV' 'SOUTH' 'WI' 'NORTH' 'WY' 'WEST'
  'NA' 'NORTH' 'ON' 'NORTH' ELSE ' ');,
   TITLE='Region:',
   DESCRIPTION=Region, $
```

CENTCOMP 構造図

```
SECTION 01
       STRUCTURE OF FOCUS    FILE CENTCOMP ON 05/15/03 AT 10.20.49

          COMPINFO
  01      S1
 **************
 *STORE_CODE  **I
 *STORENAME   **
 *STATE       **
```

```
 *            **
 *            **
 ***************
  **************
```

CENTFIN マスターファイル

```
FILE=CENTFIN, SUFFIX=FOC, FDFC=19, FYRT=00
  SEGNAME=ROOT_SEG, SEGTYPE=S4, $
  FIELD=YEAR, ALIAS=YEAR, FORMAT=YY,
   WITHIN='*Time Period', $
  FIELD=QUARTER, ALIAS=QTR, FORMAT=Q,
   WITHIN=YEAR,
   TITLE=Quarter,
   DESCRIPTION=Quarter, $
  FIELD=MONTH, ALIAS=MONTH, FORMAT=M,
   TITLE=Month,
   DESCRIPTION=Month, $
  FIELD=ITEM, ALIAS=ITEM, FORMAT=A20,
   TITLE=Item,
   DESCRIPTION=Item, $
  FIELD=VALUE, ALIAS=VALUE, FORMAT=D12.2,
   TITLE=Value,
   DESCRIPTION=Value, $
  DEFINE ITYPE/A12=IF EDIT(ITEM,'9$$$$$$$$$$$$$$$$$$') EQ 'E'
   THEN 'Expense' ELSE IF EDIT(ITEM,'9$$$$$$$$$$$$$$$$$$') EQ 'R'
   THEN 'Revenue' ELSE 'Asset';,
   TITLE=Type,
   DESCRIPTION='Type of Financial Line Item',$
  DEFINE MOTEXT/MT=MONTH;,$
```

CENTFIN 構造図

```
SECTION 01
       STRUCTURE OF FOCUS    FILE CENTFIN  ON 05/15/03 AT 10.25.52

          ROOT_SEG
  01      S4
```

```
 **************
 *YEAR         **
 *QUARTER      **
 *MONTH        **
 *ITEM         **
 *             **
 ***************
  **************
```

CENTHR マスターファイル

```
FILE=CENTHR, SUFFIX=FOC
  SEGNAME=EMPSEG, SEGTYPE=S1, $
  FIELD=ID_NUM, ALIAS=ID#, FORMAT=I9,
   TITLE='Employee,ID#',
   DESCRIPTION='Employee Identification Number', $
  FIELD=LNAME, ALIAS=LN, FORMAT=A14,
   TITLE='Last,Name',
   DESCRIPTION='Employee Last Name', $
  FIELD=FNAME, ALIAS=FN, FORMAT=A12,
   TITLE='First,Name',
   DESCRIPTION='Employee First Name', $
  FIELD=PLANT, ALIAS=PLT, FORMAT=A3,
   TITLE='Plant,Location',
   DESCRIPTION='Location of the manufacturing plant',
   WITHIN='*Location', $
  FIELD=START_DATE, ALIAS=SDATE, FORMAT=YYMD,
   TITLE='Starting,Date',
   DESCRIPTION='Date of employment',$
  FIELD=TERM_DATE, ALIAS=TERM_DATE, FORMAT=YYMD,
   TITLE='Termination,Date',
   DESCRIPTION='Termination Date', $
  FIELD=STATUS, ALIAS=STATUS, FORMAT=A10,
   TITLE='Current,Status',
   DESCRIPTION='Job Status', $
  FIELD=POSITION, ALIAS=JOB, FORMAT=A2,
   TITLE=Position,
   DESCRIPTION='Job Position',  $
  FIELD=PAYSCALE, ALIAS=PAYLEVEL, FORMAT=I2,
   TITLE='Pay,Level',
   DESCRIPTION='Pay Level',
   WITHIN='*Wages',$
  DEFINE POSITION_DESC/A17=IF POSITION EQ 'BM' THEN
```

```
 'Plant Manager' ELSE
 IF POSITION EQ 'MR' THEN 'Line Worker' ELSE
 IF POSITION EQ 'TM' THEN 'Line Manager' ELSE
 'Technician';
 TITLE='Position,Description',
 DESCRIPTION='Position Description',
 WITHIN='PLANT',$
DEFINE BYEAR/YY=START_DATE;
 TITLE='Beginning,Year',
 DESCRIPTION='Beginning Year',
 WITHIN='*Starting Time Period',$
```

```
DEFINE BQUARTER/Q=START_DATE;
 TITLE='Beginning,Quarter',
 DESCRIPTION='Beginning Quarter',
 WITHIN='BYEAR',
DEFINE BMONTH/M=START_DATE;
 TITLE='Beginning,Month',
 DESCRIPTION='Beginning Month',
 WITHIN='BQUARTER',$
DEFINE EYEAR/YY=TERM_DATE;
 TITLE='Ending,Year',
 DESCRIPTION='Ending Year',
 WITHIN='*Termination Time Period',$
DEFINE EQUARTER/Q=TERM_DATE;
 TITLE='Ending,Quarter',
 DESCRIPTION='Ending Quarter',
 WITHIN='EYEAR',$
DEFINE EMONTH/M=TERM_DATE;
 TITLE='Ending,Month',
 DESCRIPTION='Ending Month',
 WITHIN='EQUARTER',$
DEFINE RESIGN_COUNT/I3=IF STATUS EQ 'RESIGNED' THEN 1
 ELSE 0;
 TITLE='Resigned,Count',
 DESCRIPTION='Resigned Count',$
DEFINE FIRE_COUNT/I3=IF STATUS EQ 'TERMINAT' THEN 1
 ELSE 0;
 TITLE='Terminated,Count',
 DESCRIPTION='Terminated Count',$
DEFINE DECLINE_COUNT/I3=IF STATUS EQ 'DECLINED' THEN 1
 ELSE 0;
 TITLE='Declined,Count',
 DESCRIPTION='Declined Count',$
DEFINE EMP_COUNT/I3=IF STATUS EQ 'EMPLOYED' THEN 1
 ELSE 0;
 TITLE='Employed,Count',
```

```
   DESCRIPTION='Employed Count',$
  DEFINE PEND_COUNT/I3=IF STATUS EQ 'PENDING' THEN 1
   ELSE 0;
   TITLE='Pending,Count',
   DESCRIPTION='Pending Count',$
  DEFINE REJECT_COUNT/I3=IF STATUS EQ 'REJECTED' THEN 1
   ELSE 0;
   TITLE='Rejected,Count',
   DESCRIPTION='Rejected Count',$
  DEFINE FULLNAME/A28=LNAME||', '|FNAME;
   TITLE='Full Name',
   DESCRIPTION='Full Name:Last, First', WITHIN='POSITION_DESC',$
```

```
  DEFINE SALARY/D12.2=IF BMONTH LT 4 THEN PAYLEVEL * 12321
   ELSE IF BMONTH GE 4 AND BMONTH LT 8 THEN PAYLEVEL * 13827
   ELSE PAYLEVEL * 14400;,
   TITLE='Salary',
   DESCRIPTION='Salary',$
  DEFINE PLANTLNG/A11=DECODE PLANT (BOS 'Boston' DAL 'Dallas'
   LA 'Los Angeles' ORL 'Orlando' SEA 'Seattle' STL 'St Louis'
   ELSE 'n/a');$
```

CENTHR 構造図

```
SECTION 01
     STRUCTURE OF FOCUS    FILE CENTHR   ON 05/15/03 AT 10.40.34

          EMPSEG
  01      S1
 **************
 *ID_NUM      **
 *LNAME       **
 *FNAME       **
 *PLANT       **
 *            **
 ***************
  **************
```

CENTINV マスターファイル

```
FILE=CENTINV, SUFFIX=FOC, FDFC=19, FYRT=00
 SEGNAME=INVINFO, SEGTYPE=S1, $
  FIELD=PROD_NUM, ALIAS=PNUM, FORMAT=A4, INDEX=I,
   TITLE='Product,Number:',
   DESCRIPTION='Product Number', $
  FIELD=PRODNAME, ALIAS=PNAME, FORMAT=A30,
   WITHIN=PRODCAT,
   TITLE='Product,Name:',
   DESCRIPTION='Product Name', $
  FIELD=QTY_IN_STOCK, ALIAS=QIS, FORMAT=I7,
   TITLE='Quantity,In Stock:',
   DESCRIPTION='Quantity In Stock', $
  FIELD=PRICE, ALIAS=RETAIL, FORMAT=D10.2,
   TITLE='Price:',
   DESCRIPTION=Price, $
  FIELD=COST, ALIAS=OUR_COST, FORMAT=D10.2,
   TITLE='Our,Cost:',
   DESCRIPTION='Our Cost:', $
  DEFINE PRODCAT/A22 = IF PRODNAME CONTAINS 'LCD'
   THEN 'VCRs' ELSE IF PRODNAME
   CONTAINS 'DVD' THEN 'DVD' ELSE IF PRODNAME CONTAINS 'Camcor'
   THEN 'Camcorders'
   ELSE IF PRODNAME CONTAINS 'Camera' THEN 'Cameras' ELSE IF PRODNAME
   CONTAINS 'CD' THEN 'CD Players'
   ELSE IF PRODNAME CONTAINS 'Tape' THEN 'Digital Tape Recorders'
   ELSE IF PRODNAME CONTAINS 'Combo' THEN 'Combo Players'
   ELSE 'PDA Devices'; WITHIN=PRODTYPE, TITLE='Product Category:' ,$
  DEFINE PRODTYPE/A19 = IF PRODNAME CONTAINS 'Digital' OR 'DVD' OR 'QX'
   THEN 'Digital' ELSE 'Analog';,WITHIN='*Product Dimension',
   TITLE='Product Type:',$
```

CENTINV 構造図

```
SECTION 01
       STRUCTURE OF FOCUS    FILE CENTINV  ON 05/15/03 AT 10.43.35

          INVINFO
  01      S1
 **************
```

```
 *PROD_NUM    **I
 *PRODNAME    **
 *QTY_IN_STOCK**
 *PRICE       **
 *            **
 ***************
  **************
```

CENTORD マスターファイル

```
FILE=CENTORD, SUFFIX=FOC
 SEGNAME=OINFO, SEGTYPE=S1, $
  FIELD=ORDER_NUM, ALIAS=ONUM, FORMAT=A5, INDEX=I,
   TITLE='Order,Number:',
   DESCRIPTION='Order Number', $
  FIELD=ORDER_DATE, ALIAS=ODATE, FORMAT=YYMD,
   TITLE='Date,Of Order:',
   DESCRIPTION='Date Of Order', $
  FIELD=STORE_CODE, ALIAS=SNUM, FORMAT=A6, INDEX=I,
   TITLE='Company ID#:',
   DESCRIPTION='Company ID#', $
  FIELD=PLANT, ALIAS=PLNT, FORMAT=A3, INDEX=I,
   TITLE='Manufacturing,Plant',
   DESCRIPTION='Location Of Manufacturing Plant',
   WITHIN='*Location',$
  DEFINE YEAR/YY=ORDER_DATE;,
   WITHIN='*Time Period',$
  DEFINE QUARTER/Q=ORDER_DATE;,
   WITHIN='YEAR',$
  DEFINE MONTH/M=ORDER_DATE;,
   WITHIN='QUARTER',$
 SEGNAME=PINFO, SEGTYPE=S1, PARENT=OINFO, $
  FIELD=PROD_NUM, ALIAS=PNUM, FORMAT=A4,INDEX=I,
   TITLE='Product,Number#:',
   DESCRIPTION='Product Number#', $
  FIELD=QUANTITY, ALIAS=QTY, FORMAT=I8C,
   TITLE='Quantity:',
   DESCRIPTION=Quantity, $
  FIELD=LINEPRICE, ALIAS=LINETOTAL, FORMAT=D12.2MC,
   TITLE='Line,Total',
   DESCRIPTION='Line Total', $
  DEFINE LINE_COGS/D12.2=QUANTITY*COST;,
   TITLE='Line,Cost Of,Goods Sold',
```

```
   DESCRIPTION='Line cost of goods sold', $
  DEFINE PLANTLNG/A11=DECODE PLANT (BOS 'Boston' DAL 'Dallas'
   LA 'Los Angeles' ORL 'Orlando' SEA 'Seattle' STL 'St Louis'
   ELSE 'n/a');
 SEGNAME=INVSEG, SEGTYPE=DKU, PARENT=PINFO, CRFILE=CENTINV,
  CRKEY=PROD_NUM,   CRSEG=INVINFO,$
 SEGNAME=STOSEG, SEGTYPE=DKU, PARENT=OINFO, CRFILE=CENTCOMP,
  CRKEY=STORE_CODE, CRSEG=COMPINFO,$
```

CENTORD 構造図

```
SECTION 01
      STRUCTURE OF FOCUS    FILE CENTORD  ON 05/15/03 AT 10.17.52

          OINFO
  01      S1
 **************
 *ORDER_NUM   **I
 *STORE_CODE  **I
 *PLANT       **I
 *ORDER_DATE  **
 *            **
 ***************
  **************
        I
        +-----------------+
        I                 I
        I STOSEG          I PINFO
  02    I KU        03    I S1
 ..............    **************
 :STORE_CODE  :K   *PROD_NUM    **I
 :STORENAME   :    *QUANTITY    **
 :STATE       :    *LINEPRICE   **
 :            :    *            **
 :            :    *            **
 :............:    ***************
  JOINED  CENTCOMP  **************
                          I
                          I
                          I
                          I INVSEG
                    04    I KU
                   ..............
```

```
                    :PROD_NUM    :K
                    :PRODNAME    :
                    :QTY_IN_STOCK:
                    :PRICE       :
                    :            :
                    :............:
                     JOINED  CENTINV
```

CENTQA マスターファイル

```
FILE=CENTQA, SUFFIX=FOC, FDFC=19, FYRT=00
 SEGNAME=PROD_SEG, SEGTYPE=S1, $
  FIELD=PROD_NUM, ALIAS=PNUM, FORMAT=A4, INDEX=I,
   TITLE='Product,Number',
   DESCRIPTION='Product Number', $
 SEGNAME=PROB_SEG, PARENT=PROD_SEG, SEGTYPE=S1, $
  FIELD=PROBNUM, ALIAS=PROBNO, FORMAT=I5,
   TITLE='Problem,Number',
   DESCRIPTION='Problem Number',
   WITHIN=PLANT,$
  FIELD=PLANT, ALIAS=PLT, FORMAT=A3, INDEX=I,
   TITLE=Plant,
   DESCRIPTION=Plant,
   WITHIN=PROBLEM_LOCATION,$
  FIELD=PROBLEM_DATE, ALIAS=PDATE, FORMAT=YYMD,
   TITLE='Date,Problem,Reported',
   DESCRIPTION='Date Problem Was Reported', $
  FIELD=PROBLEM_CATEGORY, ALIAS=PROBCAT, FORMAT=A20, $
   TITLE='Problem,Category',
   DESCRIPTION='Problem Category',
   WITHIN=*Problem,$
  FIELD=PROBLEM_LOCATION, ALIAS=PROBLOC, FORMAT=A10,
   TITLE='Location,Problem,Occurred',
   DESCRIPTION='Location Where Problem Occurred',
   WITHIN=PROBLEM_CATEGORY,$
  DEFINE PROB_YEAR/YY=PROBLEM_DATE;,
   TITLE='Year,Problem,Occurred',
   DESCRIPTION='Year Problem Occurred',
   WITHIN=*Time Period,$
  DEFINE PROB_QUARTER/Q=PROBLEM_DATE;
   TITLE='Quarter,Problem,Occurred',
   DESCRIPTION='Quarter Problem Occurred',
   WITHIN=PROB_YEAR,$
```

```
  DEFINE PROB_MONTH/M=PROBLEM_DATE;
   TITLE='Month,Problem,Occurred',
   DESCRIPTION='Month Problem Occurred',
   WITHIN=PROB_QUARTER,$
  DEFINE PROBLEM_OCCUR/I5 WITH PROBNUM=1;,
   TITLE='Problem,Occurrence'
   DESCRIPTION='# of times a problem occurs',$
  DEFINE PLANTLNG/A11=DECODE PLANT (BOS 'Boston' DAL 'Dallas'
   LA 'Los Angeles' ORL 'Orlando' SEA 'Seattle' STL 'St Louis'
   ELSE 'n/a');$
 SEGNAME=INVSEG, SEGTYPE=DKU, PARENT=PROD_SEG, CRFILE=CENTINV,
  CRKEY=PROD_NUM, CRSEG=INVINFO,$
```

CENTQA 構造図

```
SECTION 01
       STRUCTURE OF FOCUS    FILE CENTQA   ON 05/15/03 AT 10.46.43

          PROD_SEG
  01      S1
 **************
 *PROD_NUM    **I
 *            **
 *            **
 *            **
 *            **
 ***************
  **************
        I
        +-----------------+
        I                 I
        I INVSEG          I PROB_SEG
  02    I KU        03    I S1
 ..............    **************
 :PROD_NUM    :K   *PROBNUM     **
 :PRODNAME    :    *PLANT       **I
 :QTY_IN_STOCK:    *PROBLEM_DATE**
 :PRICE       :    *PROBLEM_CAT>**
 :            :    *            **
 :............:    ***************
  JOINED  CENTINV   **************
```

CENTGL マスターファイル

```
FILE=CENTGL ,SUFFIX=FOC
 SEGNAME=ACCOUNTS, SEGTYPE=S1
  FIELDNAME=GL_ACCOUNT, ALIAS=GLACCT, FORMAT=A7,
   TITLE='Ledger,Account', FIELDTYPE=I, $
  FIELDNAME=GL_ACCOUNT_PARENT, ALIAS=GLPAR, FORMAT=A7,
   TITLE=Parent,
   PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT, $
  FIELDNAME=GL_ACCOUNT_TYPE, ALIAS=GLTYPE, FORMAT=A1,
   TITLE=Type,$
  FIELDNAME=GL_ROLLUP_OP, ALIAS=GLROLL, FORMAT=A1,
   TITLE=Op, $
  FIELDNAME=GL_ACCOUNT_LEVEL, ALIAS=GLLEVEL, FORMAT=I3,
   TITLE=Lev, $
  FIELDNAME=GL_ACCOUNT_CAPTION, ALIAS=GLCAP, FORMAT=A30,
   TITLE=Caption,
   PROPERTY=CAPTION, REFERENCE=GL_ACCOUNT, $
  FIELDNAME=SYS_ACCOUNT, ALIAS=ALINE, FORMAT=A6,
   TITLE='System,Account,Line', MISSING=ON, $
```

CENTGL 構造図

```
SECTION 01
       STRUCTURE OF FOCUS    FILE CENTGL   ON 05/15/03 AT 15.18.48

          ACCOUNTS
  01      S1
 **************
 *GL_ACCOUNT  **I
 *GL_ACCOUNT_>**
 *GL_ACCOUNT_>**
 *GL_ROLLUP_OP**
 *            **
 ***************
  **************
```

CENTSYSF マスターファイル

```
FILE=CENTSYSF ,SUFFIX=FOC
 SEGNAME=RAWDATA ,SEGTYPE=S2
  FIELDNAME = SYS_ACCOUNT ,  ,A6    , FIELDTYPE=I,
   TITLE='System,Account,Line', $
  FIELDNAME = PERIOD      ,  ,YYM   , FIELDTYPE=I,$
  FIELDNAME = NAT_AMOUNT  ,  ,D10.0 , TITLE='Month,Actual', $
  FIELDNAME = NAT_BUDGET  ,  ,D10.0 , TITLE='Month,Budget', $
  FIELDNAME = NAT_YTDAMT  ,  ,D12.0 , TITLE='YTD,Actual',   $
  FIELDNAME = NAT_YTDBUD  ,  ,D12.0 , TITLE='YTD,Budget',   $
```

CENTSYSF 構造図

```
SECTION 01
       STRUCTURE OF FOCUS    FILE CENTSYSF   ON 05/15/03 AT 15.19.27

          RAWDATA
  01      S2
 **************
 *SYS_ACCOUNT **I
 *PERIOD      **I
 *NAT_AMOUNT  **
 *NAT_BUDGET  **
 *            **
 ***************
  **************
```

CENTSTMT マスターファイル

```
FILE=CENTSTMT, SUFFIX=FOC
 SEGNAME=ACCOUNTS, SEGTYPE=S1
  FIELD=GL_ACCOUNT, ALIAS=GLACCT,  FORMAT=A7,
   TITLE='Ledger,Account', FIELDTYPE=I, $
  FIELD=GL_ACCOUNT_PARENT, ALIAS=GLPAR, FORMAT=A7,
   TITLE=Parent,
   PROPERTY=PARENT_OF, REFERENCE=GL_ACCOUNT, $
```

```
  FIELD=GL_ACCOUNT_TYPE, ALIAS=GLTYPE, FORMAT=A1,
   TITLE=Type,$
  FIELD=GL_ROLLUP_OP, ALIAS=GLROLL, FORMAT=A1,
   TITLE=Op, $
  FIELD=GL_ACCOUNT_LEVEL, ALIAS=GLLEVEL, FORMAT=I3,
   TITLE=Lev, $
  FIELD=GL_ACCOUNT_CAPTION, ALIAS=GLCAP, FORMAT=A30,
   TITLE=Caption,
   PROPERTY=CAPTION, REFERENCE=GL_ACCOUNT, $
 SEGNAME=CONSOL, SEGTYPE=S1, PARENT=ACCOUNTS, $
  FIELD=PERIOD, ALIAS=MONTH, FORMAT=YYM, $
  FIELD=ACTUAL_AMT, ALIAS=AA, FORMAT=D10.0, MISSING=ON,
   TITLE='Actual', $
  FIELD=BUDGET_AMT, ALIAS=BA, FORMAT=D10.0, MISSING=ON,
   TITLE='Budget', $
  FIELD=ACTUAL_YTD, ALIAS=AYTD, FORMAT=D12.0, MISSING=ON,
   TITLE='YTD,Actual', $
  FIELD=BUDGET_YTD, ALIAS=BYTD, FORMAT=D12.0, MISSING=ON,
   TITLE='YTD,Budget', $
```

CENTSTMT 構造図

```
SECTION 01
     STRUCTURE OF FOCUS    FILE CENTSTMT ON 05/15/03 AT 14.45.44

          ACCOUNTS
  01      S1
 **************
 *GL_ACCOUNT  **I
 *GL_ACCOUNT_>**
 *GL_ACCOUNT_>**
 *GL_ROLLUP_OP**
 *            **
 ***************
  **************
        I
        I
        I
        I CONSOL
  02    I S1
 **************
 *PERIOD      **
 *ACTUAL_AMT  **
```

```
*BUDGET_AMT  **
*ACTUAL_YTD  **
*            **
**************
 **************
```

エラーメッセージ

エラーメッセージに関するテキストまたは説明は、エラーファイルに表示することができます。すべての WebFOCUS エラーメッセージは、8 つのシステムエラーファイルに格納されています。

- UNIX、Windows では、ファイルの拡張子は .err です。

メッセージの表示

メッセージのテキストおよび説明を表示するには、プロシジャで次のクエリコマンドを発行します。

```
? n
```

説明

n

メッセージ番号です。

メッセージ番号およびテキストが、メッセージの詳細説明 (存在する場合) とともに表示されます。たとえば、次のコマンドを発行します。

```
?210
```

次のメッセージが表示されます。

```
(FOC210)          数値以外のデータがあります:
数値でなければならないところに英文字があります。
```

テーブル構文の概要と制限事項

ここでは、WebFOCUS レポートコマンドおよびオプションの概要について説明します。

TABLE 構文の概要

TABLE リクエストの構文は、次のとおりです。

```
DEFINE FILE filename    CLEAR|ADD
tempfield  [/format] [{DEFCENT|DFC} {cc|19} {YRTHRESH|YRT} {[-]yy|0}]
    [MISSING {ON|OFF} [NEEDS] {SOME|ALL} [DATA]]
    [(GEOGRAPHIC_ROLE = georole)] [WITH realfield]
    [TITLE 'line1[,line2 ...']]
  [DESCRIPTION 'description'] = expression;
tempfield  [/format]    REDEFINES qualifier.fieldname = expression;
.
.
.
END
TABLE  FILE  filename
HEADING  [CENTER]
"text"
{display_command}   [SEG.] field   [/R|/L|/C]   [/format]
{display_command}   [prefixop.] [field]  [/R|/L|/C]  [/format]
  [NOPRINT|AS 'title1,...,title5']   [AND|OVER]   [obj2...obj1024]
      [WITHIN field]   [IN [+]n]
COMPUTE field  [/format] [(GEOGRAPHIC_ROLE = georole)] =
        expression; [AS 'title,...,title5']  [IN [+]n]
[AND]  ROW-TOTAL      [/R|/L|/C]    [/format][AS 'name']
[AND]  COLUMN-TOTAL   [/R|/L|/C]    [AS 'name']
ACROSS  [HIGHEST]  sortfieldn       [IN-GROUPS-OF qty]
    [NOPRINT|  AS 'title1,...,title5']
BY  [HIGHEST]  sortfieldn           [IN-GROUPS-OF qty]
    [NOPRINT|  AS 'title1,...,title5']
BY  [HIGHEST|LOWEST{n}]  TOTAL  [prefix_operator]  {field|code_value}
RANKED [AS 'name'] BY {TOP|HIGHEST|LOWEST} [n] field
       [PLUS OTHERS AS 'othertext']
      [IN-GROUPS-OF qty [TILES [TOP m]] [AS 'heading']]
      [NOPRINT|AS 'title1,...,title5']
```

```
{BY|ACROSS} sortfield IN-RANGES-OF value [TOP limit]
ON   sfld  option1  [AND]  option2   [WHEN expression;...]
ON   sfld  RECAP   fld1  [/fmt] = FORECAST (fld2, intvl, npredct,
  '{MOVAVE|EXPAVE}',npnt);
```

```
ON sfld RECAP fld1[/fmt] = FORECAST(fld2, interval, npredict, 'DOUBLEXP',
   npoint1, npoint2);
```

```
ON sfld RECAP fld1[/fmt] = FORECAST(fld2, interval, npredict, 'SEASONAL',
   nperiod, npoint1, npoint2, npoint3);
```

```
ON   sfld  RECAP   fld1  [/fmt] = FORECAST (fld2, intvl, npredct,
'REGRESS');
```

```
ON {sortfield|TABLE} RECAP y[/fmt] = REGRESS(n, x1, [x2, [x3,]] z);
ON   sfld  RECAP   fld1  [/fmt] = FORECAST (infield, interval, npredict,
  'DOUBLEXP',npoint, npoint2);
ON  sfld   RECAP   fld1  [/fmt] = FORECAST (infield, interval, npredict,
  'SEASONAL', nperiod, npoint, npoint2, npoint3);{BY|ON} fieldname
SUBHEAD
  [NEWPAGE]
"text"
```

```
{BY|ON} fieldname SUBFOOT [WITHIN] [MULTILINES][NEWPAGE]
"text" [<prefop.fieldname ... ]"        [WHEN expression;]
```

```
WHERE   [TOTAL]  expression
WHERE   {RECORDLIMIT|READLIMIT}  EQ  n
IF      [TOTAL]  field relation  value  [OR value...]
WHERE_GROUPED expression
ON TABLE  SET parameter value
ON TABLE  HOLD [VIA program][AS name] [FORMAT format] [DATASET dataset]
              [MISSING {ON|OFF}] [PERSISTENCE {STAGE|PERMANENT}]
ON TABLE  {PCHOLD|SAVE|SAVB} [AS name] [FORMAT format]  [MISSING {ON|OFF}]
ON TABLE  NOTOTAL
ON TABLE  COLUMN-TOTAL [/R|/L|/C]  [AS 'name']  fieldname
ON TABLE  {ROW-TOTAL|ACROSS-TOTAL}[/R|/L|/C][format] [AS 'name'] fldname
{BY|ON}  sfld [AS 'text1']   {SUBTOTAL|SUB-TOTAL|SUMMARIZE|RECOMPUTE}
  [MULTILINES] [pref. ] [field1 [pref. ] field2 ...] [AS 'text2']
  [WHEN expression;]
{ACROSS|ON}  sfld [AS 'text1'] {SUBTOTAL|SUB-TOTAL|SUMMARIZE|RECOMPUTE}
```

```
   [AS 'text2']  [COLUMNS c1 [AND c2 ...]]
ON  TABLE {SUBTOTAL|SUB-TOTAL|SUMMARIZE|RECOMPUTE}
  [pref. ] [field1 [pref. ] field2 ...] [AS 'text2']
FOOTING  [CENTER]  [BOTTOM]
"text"
MORE
FILE  file2
   [IF field relation value [OR value...]|WHERE expression]
{END|RUN|QUIT}
```

階層レポート構文の概要

```
SUM [FROLL.]measure_field ...
BY hierarchy_field [HIERARCHY [WHEN expression_using_hierarchy_fields;]
  [SHOW [TOP|UP n] [TO {BOTTOM|DOWN m}] [byoption [WHEN condition] ...] ]
[WHERE expression_using_dimension_data]
[ON hierarchy_field HIERARCHY [WHEN expression_using_hierarchy_fields;]
  [SHOW [TOP|UP n] [TO BOTTOM|DOWN m] [byoption [WHEN condition] ...]]
```

TABLEF 構文の概要

TABLEF リクエストの構文は、次のとおりです。

```
TABLEF FILE filename
HEADING [CENTER]
"text"
```

```
{display_command}  [SEG.]field  [/R|/L|/C]  [/format]
{display_command}  [prefixop.][field] [/R|/L|/C] [/format]
    [NOPRINT|AS 'title1,...,title5']    [AND|OVER]    [obj2...obj495]
    [IN n]
```

```
COMPUTE field [/format]=expression; [AS 'title1,...title5']
[AND] ROW-TOTAL [AND] COLUMN-TOTAL
```

```
BY [HIGHEST] keyfieldn [NOPRINT]
```

```
ON keyfield option1 [AND] option2...
```

```
WHERE [TOTAL] expression
```

```
IF [TOTAL] field relation value [OR value...]
```

```
ON TABLE SET parameter value
```

```
ON TABLE HOLD [VIA program] [AS name] [FORMAT format]  [MISSING {ON|OFF}]
ON TABLE PCHOLD             [AS name] [FORMAT format]  [MISSING {ON|OFF}]
ON TABLE SAVE               [AS name] [FORMAT format]  [MISSING {ON|OFF}]
ON TABLE SAVB               [AS name] [FORMAT format]  [MISSING {ON|OFF}]
```

```
ON TABLE  NOTOTAL
ON TABLE  COLUMN-TOTAL fieldname
ON TABLE  ROW-TOTAL fieldname
```

```
 FOOTING [CENTER] [BOTTOM]
 "text"

{END|RUN|QUIT}
```

注意

- TABLEF リクエストで使用可能な演算接頭語には、AVE.、ASQ.、MAX.、MIN.、FST.、LST.、CNT.、SUM. があります。
- TABLEF リクエストでは、以下を使用することはできません。
  - DST.、PCT.、PCT.CNT.、RPCT.、TOT. 演算接頭語
  - TABPAGENO、TABLASTPAGE、BYLASTPAGE 変数
  - SET SQUEEZE
  - 境界スタイル
  - ACROSS 句

- WITHIN 句
- RANKED BY 句
- NOSPLIT
- 複数の表示コマンドを使用したリクエスト (複数動詞のリクエスト)

MATCH 構文の概要

MATCH リクエストの構文は、次のとおりです。

```
MATCH FILE filename    (the OLD file) report request
BY field1 [AS sortfield]
MORE
FILE file3
subrequest
RUN
.
.
.
FILE filename2         (the NEW file) report request
BY field1 [AS sortfield1]
.
.
.
[AFTER MATCH HOLD [AS filename]  matchtype]
MORE
FILE file4
subrequest
END
```

説明

matchtype

次のいずれかです。

```
OLD
```

```
NEW
```

```
OLD-NOT-NEW
```

```
NEW-NOT-OLD
```

```
OLD-AND-NEW
```

```
OLD-OR-NEW
```

```
OLD-NOR-NEW
```

FOR 構文の概要

FOR ステートメントの正式な構文は、次のとおりです。

```
FOR fieldname [NOPRINT]row [OVER row]
.
.
.
.
END
```

説明

row

次のいずれかです。

```
tag [OR tag...][options]
[fieldname]
DATA n,[n,....] $
DATA PICKUP [FROM filename] tag [LABEL label] [AS 'text']
RECAP name[/format]=expression;
BAR [AS 'character'] [OVER]
"text"
parentvalue {GET|WITH} CHILD[REN] [{n|ALL}] [ADD [m|ALL]]
```

```
   [AS {CAPTION|'text'}] [LABEL label]parentvalue ADD [{m|ALL}] [AS
{CAPTION|'text'}] [LABEL label]
PAGE-BREAK [OVER]
```

tag

次のいずれかです。

```
value [OR value...] value TO value
```

options

次のいずれかです。

```
AS 'text'
[INDENT m]
NOPRINT
[LABEL label]
WHEN EXISTS
[POST [TO filename]]
```

TABLE の制限事項

次の制限事項は、TABLE リクエストに適用されます。

- TABLE リクエストでは、動詞オブジェクトの数に制限はありません。ただし、レポート出力フォーマットにサポートされるフィールドの数に制限がある場合、オペレーティングシステムに最大レコード長が設定され、この制限値にすべてのフィールドが収まらない場合、または出力の格納に必要なメモリ量を確保できない場合には、エラーが発生します。
- MATCH リクエストで参照可能な動詞オブジェクトの最大数は 495 です。
- レポート出力の最大フィールド数は 1024 です。
- リクエスト内または単一マスターファイル内のすべてのフィールドを合計した長さは 256 キロバイトです。
- ソートフィールドの総数 (BY と ACROSS の合計) は 128 です。

場合によっては内部ソートが自動的に生成されるため、その個数も総数に含める必要があります。HOLD FORMAT FOCUS/XFOCUS を使用すると、一意性を確保するために FOCLIST が BY フィールドとして追加されます。

- 出力レコードの最大サイズは 256 キロバイトです (FORMAT/USAGE)。
- 出力レコードの最大サイズは FOCUS パーティションでは 2 ギガバイト、XFOCUS では 32 ギガバイトです。
- 単一の DEFINE、COMPUTE、-SET、または WHERE 句の内部式表現の最大サイズは 64 キロバイトです。

  この内部表現は、逆ポーランド記法 (後置記法) で生成され、入力式に比べてサイズが著しく小さくなります。また、式の定数 (DECODE や IF ... THEN ... ELSE 条件内の式) は式表現の外部に格納されるため、式表現自体に必要な領域が減少します。
- 構造またはファイル内のセグメントの最大数は 512 です。

  つまり、合計 511 個の JOIN を同時に有効にすることができます。

  注意：FOCUS データソースの場合、セグメントの最大数は 64 です。

  JOIN で使用可能なフィールドの最大ペア数は 128 です。
- 文字フィールドの最大サイズは 4 キロバイトです (UTF の場合、12 キロバイト)。
- TABLE リクエスト内の表示コマンドの最大数は 64 です。

リクエストのフィールド参照

レポートを作成する際に、リクエストのさまざまな箇所でフィールドを参照します。たとえば、表示コマンド (PRINT、SUM)、ソート句 (BY、ACROSS)、選択条件 (WHERE、WHERE TOTAL、IF) でフィールドを参照します。

フィールドを参照するには、いくつかの方法があります。次の処理を実行することができます。

- マスターファイルで指定されたエイリアス、マスターファイルで定義されたフィールド名、またはフィールド名やエイリアスの一意の最小短縮名を使用して、個別フィールドを参照する。詳細は、個別フィールドの参照を参照してください。
- 長い修飾フィールド名を使用して、フィールドを参照する。詳細は、修飾フィールド名によるフィールドの参照を参照してください。
- 1 つのフィールド名のみを使用して、セグメントの全フィールドを参照する。詳細は、セグメント全フィールドの参照を参照してください。

現在アクティブなデータソース、または指定したマスターファイルに含まれる全フィールドのリストを表示することもできます。詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

個別フィールドの参照

次のいずれかの方法を使用して、フィールドを個別に参照することができます。

- マスターファイルで定義したフィールド名を使用する。
- マスターファイルで定義したエイリアス (フィールド名の別名) を使用する。
- フィールド名またはエイリアスの一意の最小短縮名を使用する。短縮名は一意である必要があります。名前が重複する場合、エラーメッセージが表示されます。
- マスターファイルで定義されるフィールド名の末尾に文字「S」を追加する。

個別フィールドの参照

次のリクエストでは、DEPARTMENT は完全フィールド名、DPT はエイリアス、DEP は DEPARTMENT の一意の短縮名です。これらの例は、すべて同一の出力を作成します。

```
1.TABLE FILE EMPLOYEE
   PRINT DEPARTMENT
   END
```

```
2.TABLE FILE EMPLOYEE
   PRINT DPT
   END
```

```
3.TABLE FILE EMPLOYEE
   PRINT DEP
   END
```

注意：使用する短縮名が一意ではない場合、次のメッセージが表示されます。

```
(FOC016) フィールド名の省略形が、一意でありません :D
```

修飾フィールド名によるフィールドの参照

フィールド名およびエイリアスの最大長は 512 バイトです。また、フィールド名は、最大 2 つの修飾子および修飾文字を先頭に付けることで修飾することもできます。例外として、FOCUS データソースのマスターファイルで記述するテキストフィールド名およびインデックスフィールド名の長さは 12 バイトに制限されます。ただし、テキストフィールドおよびインデックスフィールドのエイリアスには、通常のフィールド名と同一の長さを使用することができます。フィールド名は、TITLE 属性または AS 句で別名を指定しない限り、常にレポートのフィールドタイトルとして表示されます。詳細は、見出し、脚注、タイトル、ラベルの使用を参照してください。

指定したフィールドの修飾子として、ファイル名、セグメント名、またはその両方を使用することができます。

長い修飾フィールド名の有効化

SET FIELDNAME コマンドは、長い (512 バイト以内) 修飾フィールド名を有効にします。

```
SET FIELDNAME = fieldname
```

説明

fieldname

長い修飾フィールド名の有効状態を指定します。有効な修飾子には、次のものがあります。

NEW - 512 バイトの修飾フィールド名がサポートされることを示します。デフォルト値は NEW です。

NOTRUNC - 最大 512 バイトまでサポートします。一意のフィールド短縮名は使用できません。

OLD は現在動作せず、NEW として機能します。

修飾フィールド名によるフィールド参照

```
EMPLOYEE.EMPINFO.EMP_ID
```

EMPLOYEE ファイルの EMPINFO セグメント内のフィールド EMP_ID の完全修飾名です。

長い修飾フィールド名使用上の注意

?SET は FIELDNAME の現在の値を表示します。さらに、「&FOCFIELDNAME」と呼ばれるダイアログマネージャ変数が利用できます。&FOCFIELDNAME は NEW または NOTRUNC の値を持つことができます。

セグメント全フィールドの参照

すべてのセグメントフィールドを表示するレポートを生成する場合、各フィールドを指定することなく、セグメント全体を参照することができます。セグメント内の任意のフィールドを、SEG. 演算子を含めて指定します。

セグメント全フィールドの参照

GGPRODS マスターファイルのセグメント PRODS01 には、PRODUCT_ID、PRODUCT_DESCRIPTION、VENDOR_CODE、VENDOR_NAME、PACKAGE_TYPE、SIZE、および UNIT_PRICE フィールドが格納されています。

```
SEGMENT=PRODS01
FIELDNAME = PRODUCT_ID
FIELDNAME = PRODUCT_DESCRIPTION
FIELDNAME = VENDOR_CODE
FIELDNAME = VENDOR_NAME
FIELDNAME = PACKAGE_TYPE
FIELDNAME = SIZE
FIELDNAME = UNIT_PRICE
```

セグメント内の各フィールドデータを含むレポートを作成するには、次のリクエストのいずれかを発行します。

```
1.TABLE FILE GGPRODS
   PRINT PRODUCT_ID AND PRODUCT_DESCRIPTION AND VENDOR_CODE AND
    VENDOR_NAME AND PACKAGE_TYPE AND SIZE AND UNIT_PRICE
   END
```

```
2.TABLE FILE GGPRODS
   PRINT SEG.PRODUCT_ID
   END
```

フィールド名のリスト表示

現在のアクティブデータソースに含まれる全フィールドのリストを表示する場合、?F フィールド名クエリを発行することが可能です。

リクエスト処理を中止することなく、フィールド名のリストを表示する必要がある場合、またはフィールド名のスペルチェックを必要とする場合に役立ちます。66 バイトのフィールド名全体を表示することができます。

?F クエリは、WebFOCUS のエディタから発行することができます。

すべてのクエリコマンド (?) についての詳細は、『ibi™ WebFOCUS® アプリケーション作成ガイド』を参照してください。

フィールド名、エイリアス、フォーマット情報リスト

?FF クエリは、指定したマスターファイルに対する、フィールド名、エイリアス、フォーマット情報を、セグメントごとに表示します。

?FF クエリは、WebFOCUS のエディタから発行することができます。

使用するソフトウェアが MODIFY および FSCAN をサポートする場合、これらの機能から ?FF を発行することもできます。

注意

- 重複フィールド名が指定文字列と一致する場合、その表示には ?F と ?FF の両方のセグメント名により修飾されたフィールド名が含まれます。
- 31 バイトを超えるフィールド名は、表示が切り取られ、32 バイト目にファイル名が表示よりも長いことを示すキャレット記号 (>) が追加されます。

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

ibi, the ibi logo, FOCUS, and TIBCO are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: https://scripts.sil.org/OFL

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at https://www.cloud.com/legal.

Copyright © 2021-2025. Cloud Software Group, Inc. All Rights Reserved.