

ibiTM WebFOCUS[®]

データアダプタリファレンス

バージョン 9.3.0 | 2024 年 4 月

目次

目次	2
アダプタの概要	48
リクエストの処理	48
アダプタの機能	49
アダプタの処理	49
SQL リクエストの処理	50
ibi WebFOCUS リクエストの処理	51
リレーショナルおよび非リレーショナルアダプタ	51
データ管理	53
データソースの定義	53
リクエストの処理	53
マスターファイル	54
マスターファイルでのファイル宣言の指定	55
マスターファイルでのセグメント宣言の指定	56
マスターファイルでのフィールド宣言の指定	57
MISSING 属性	59
アクセスファイル	59
アクセスファイルでのセグメント宣言の指定	59
主キー	61
一時項目 (DEFINE) の作成	61
DEFINE コマンドによる一時項目 (DEFINE) の作成	61
マスターファイルでの一時項目 (DEFINE) の定義	62
世紀をまたがる日付	63
世紀をまたがる日付の SET コマンド	63
世紀をまたがる日付の実装	64
マスターファイルの構文	65
世紀をまたがる日付設定のファイルレベルでの追加	65

世紀をまたがる日付設定のフィールドレベルでの追加	65
世紀をまたがる日付の DEFINE コマンドによる追加	66
世紀をまたがる日付の解釈	66
シノニムの管理オプション	67
データタイプサポートレポート	70
データタイプレポートを表示するには	70
数値カラムの精度と小数点以下の桁数の変更	71
デフォルトの精度と小数点以下の桁数の上書き変更	71
精度および小数点以下の桁数の属性の設定	73
SQLENGINE SET によるメタデータサービス	74
アプリケーションによるメタデータのアクセス	74
カラム情報の取得 (Db2 のみ)	75
カラムデータのソート順序の指定	75
ユーザ定義のメタデータの取得	76
ユーザ定義のメタデータの指定	76
テーブルのリストを返す	77
上位互換性の維持	78
その他のマスターファイル属性	79
テーブルの説明の記述	80
テーブルの説明の記述	80
Oracle テーブルの説明の記述	80
カラムの説明の記述	80
カラムの説明の記述	81
カラムの説明の記述	81
DEFINE フィールドの説明の記述	82
代替のカラムタイトルの指定	82
デフォルトのカラムタイトルの変更	82
デフォルトのカラムタイトルの変更	83
デフォルトカラムタイトルの 2 行カラムタイトルへの変更	83
カラムタイトルの下線の制御	84
DEFINE フィールドのカラムタイトルの指定	84
最適化の設定	84

リクエストの最適化	84
リクエストの最適化	85
最適化を OFF に指定して RDBMS に渡された SQL リクエスト	85
最適化を ON に指定して RDBMS に渡された SQL リクエスト	86
最適化で生成される SQL 構文の例	87
定数として定義された一時項目 (DEFINE) を転送するリクエストの最適化	87
数値定数の転送	87
CHAR 定数の転送	88
CHAR および INTEGER 定数の転送	88
Amazon Athena アダプタの使用	90
Amazon Athena アダプタの概要	90
Amazon Athena 環境の準備	90
Amazon Athena アダプタの構成	91
アダプタを構成するには	91
Amazon Athena ODBC アダプタの構成設定	92
Amazon Athena JDBC アダプタの構成設定	93
Amazon Athena でのシノニムの作成	95
シノニムを作成するには	96
データタイプのサポート	97
制限事項	97
Amazon AWS S3 アダプタの使用	98
AWS IAM および AWS S3 の構成	98
AWS IAM および AWS S3 を構成するには	99
AWS S3 アダプタの構成	101
AWS S3 アダプタを構成するには	101
AWS S3 のマッピングの構成	103
AWS S3 のマッピングを構成するには	103
AWS S3 の接続属性	105
Amazon Redshift アダプタの使用	107
Amazon Redshift アダプタの概要	107

Amazon Redshift ODBC 環境の準備	108
Amazon Redshift (ODBC) アダプタの構成	108
Amazon Redshift アダプタを構成するには	109
Amazon Redshift アダプタの構成設定 (ODBC)	110
Amazon Redshift (JDBC) アダプタの構成	111
Amazon Redshift アダプタを構成するには	111
Amazon Redshift アダプタの構成設定 (JDBC)	112
Amazon Redshift アダプタの拡張バルクロード構成	113
Amazon Redshift でのシノニムの作成	114
シノニムを作成するには	115
データタイプのサポート	116
Amazon Redshift でのダイレクトパススルーの使用	116
Apache Drill アダプタの使用	117
Apache Drill 環境の準備	117
Java CLASSPATH を構成するには	118
Apache Drill アダプタの構成	120
Apache Drill アダプタを構成するには	120
Apache Drill の接続属性	122
オープンソース Drill JDBC ドライバを使用した Apache Drill URL の構成設定	124
MapR JDBC ドライバを使用した Apache Drill URL の構成設定	125
トラブルシューティング	126
Apache Drill でのシノニムの作成	126
シノニムを作成するには	126
Apache Drill のシノニム作成パラメータ	128
Db2/Db2 Warehouse アダプタの使用	130
Db2 環境の準備	130
Windows で環境を設定するには	130
UNIX で環境を設定するには	131
Db2 アダプタの構成	132
接続属性の宣言	132

接続属性を定義するには	133
CLI での Db2 の接続属性	134
JDBC での Db2 の接続属性	136
接続属性の手動宣言	137
接続属性の宣言	138
接続情報の更新	138
デフォルト接続の上書き変更	139
デフォルト接続の変更	140
デフォルト接続の選択	140
接続範囲の制御	141
AUTODISCONNECT による接続範囲の制御	141
AUTODISCONNECT による接続範囲の制御	141
Db2 メタデータの管理	142
シノニムの作成	142
シノニムを作成するには	142
Db2 のシノニム作成パラメータ	143
生成されたシノニムの例	147
Db2 テーブルとカラムの属性のシノニムへのマッピング	148
アクセスファイルのキーワード	148
シノニムの管理	151
Db2 データタイプのサポート	151
可変長データタイプのマッピングの制御	151
可変長データタイプのマッピングの制御	152
SQL 式内での末尾ブランク	152
数値カラムの精度と小数点以下の桁数の変更	153
Db2 のストアードプロシジャに対するレポートの実行	153
ストアードプロシジャのシノニムの作成	154
Db2 のストアードプロシジャ CustOrders のシノニム	155
ストアードプロシジャのシノニム作成パラメータ	156
ストアードプロシジャに対するレポートの実行	157
TABLE コマンドを使用したストアードプロシジャに対するレポートの実行	158
SELECT を使用したストアードプロシジャに対するレポートの実行	159

Db2 環境のカスタマイズ	160
デフォルトのテーブルスペースの指定	160
テーブルのデフォルト記憶域の指定	160
ロックのタイプの制御	161
ロックタイプの制御	161
インデックススペースのデフォルトのパラメータの上書き変更	163
IXSPACE の設定	163
NONBLOCK モードの有効化	164
NONBLOCK モードの有効化	164
カラム名の制御	165
カラム名の制御	165
更新または削除された行数の取得	166
更新または削除された行数の取得	166
エンドユーザ情報の設定	167
エンドユーザ情報の設定	167
エンドユーザ情報の照会	168
命名規則の設定	168
命名規則の設定	169
HOLD DBMS 作成の制御	169
DBMS 作成の制御	169
Db2 最適化の設定	170
非等価条件 WHERE ベース LEFT OUTER JOIN の最適化	170
条件付き LEFT OUTER JOIN の指定	171
WHERE ベース OUTER JOIN の最適化条件	173
非等価条件 LEFT OUTER JOIN の最適化	173
検索処理のブロックサイズの指定	174
配列検索のブロックサイズの指定	174
挿入処理のブロックサイズの指定	175
バルクインサート API の省略	175
バルクインサート API の動作	176
集計の最適化による効率の改善	176
集計の最適化の設定	176

Db2 Cube Views の使用	177
Db2 Cubes Views のメタデータのマッピング	177
Db2 Cube View のマスターファイルの例	177
SQL パススルーによる Db2 ストアドプロシジャの呼び出し	179
ストアドプロシジャの呼び出し	179
ストアドプロシジャの呼び出し	180
ストアドプロシジャの例	181
ESRI ArcGIS アダプタの使用	183
ESRI ArcGIS Online アプリケーションの作成	183
ESRI ArcGIS アプリケーションを作成するには	184
ESRI ArcGIS の接続属性	187
プレミアム API コールを使用した ESRI ArcGIS アダプタでのメタデータお よびサンプルレポートの作成	189
プレミアム API コールによるメタデータおよびサンプルレポートを作成す るには	189
メタデータおよびサンプルレポート	191
ESRI ArcGIS アダプタのメタデータ	191
ESRI ArcGIS アダプタのサンプルレポート	192
Excel アダプタの使用	194
Excel アダプタの構成	194
アダプタの構成	195
Excel の接続属性	196
接続属性の手動宣言	197
接続属性の宣言	198
デフォルト接続の上書き変更	198
デフォルト接続の変更	198
デフォルト接続の選択	199
接続範囲の制御	199
接続範囲の制御	199
Excel メタデータの管理	200
アダプタの識別	200

アダプタの識別	200
シノニムの作成	201
シノニムを作成するには	201
Excel のシノニム作成パラメータ	202
生成されたシノニムの例	205
アクセスファイルのキーワード	205
シノニムの管理	206
データタイプのサポート	207
数値カラムの精度と小数点以下の桁数の変更	207
Excel 環境のカスタマイズ	207
タイムアウト制限の指定	207
TIMEOUT コマンドの発行	208
更新または削除された行数の取得	208
更新または削除された行数の取得	208
Excel 最適化の設定	209
Excel (直接取得) アダプタの使用	210
Excel (直接取得) アダプタの構成	210
Excel (直接取得) アダプタを構成するには	210
Excel (直接取得) メタデータの管理	211
シノニムの作成	211
シノニムを作成するには	211
アダプタ設定の変更	213
Fixed-Format File および Delimited File アダプタの使用	215
Fixed-Format File および Delimited File 環境の準備	215
Fixed-Format File および Delimited File アダプタの構成	216
Fixed-Format File および Delimited File アダプタを構成するには	216
WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで Fixed-Format File および Delimited File の FTP サーバ接続を構成するには	218
FTP 接続パラメータ	219
SFTP プロトコルによるバルクロードで複数ユーザを構成するには	220

UNIX、Windows ファイルシステムでの Fixed-Format File または Delimited File のパス指定	222
Fixed-Format File および Delimited File のメタデータの管理	223
シノニムの作成	223
Fixed-Format File のシノニムの作成	223
Fixed-Format File のシノニムを作成するには	223
Windows、UNIX ファイルシステムでの Fixed-Format File のシノニム作成 パラメータ	225
シノニムの管理	229
Delimited File のシノニムの作成	229
Delimited File のシノニムを作成するには	230
Delimited File のシノニム作成パラメータ	231
DFIX データファイルおよび生成されたシノニムのサンプル	236
ヘッダおよび囲み文字を使用したサンプルの DFIX データファイル	238
区切り文字の USAGE および ACTUAL フォーマット	239
Delimited File の更新	240
Google Sheets アダプタの使用	242
Google Sheets アダプタの概要	242
Google Sheets プロジェクトの作成	242
Google プロジェクトを作成するには	242
Google Sheets アダプタの構成	254
Google Sheets アダプタを構成するには	255
Google Drive アダプタの構成	259
Google Drive アダプタを構成するには	259
Google Sheets の接続属性	262
Google Sheets アダプタでのメタデータ作成	265
シノニムを作成するには	265
Google Sheets のシノニム作成パラメータ	266
HOLD コマンドを使用した Google スプレッドシートまたは Google スライ ドの作成	267
WebFOCUS HOLD コマンドを使用した Google スプレッドシートの作成	267
HOLD コマンドを使用した Google スライドの作成	268

HP Vertica アダプタの使用	271
HP Vertica 環境の準備	271
Windows および UNIX で環境を設定するには	271
HP Vertica アダプタの構成	272
アダプタの構成	273
HP Vertica の接続属性	274
接続属性の手動宣言	275
接続属性の宣言	276
デフォルト接続の上書き変更	276
デフォルト接続の変更	276
デフォルト接続の選択	277
HP Vertica メタデータの管理	277
アダプタの識別	277
アダプタの識別	278
シノニムの作成	278
シノニムを作成するには	278
HP Vertica のシノニム作成パラメータ	280
生成されたシノニムの例	283
アクセスファイルのキーワード	283
シノニムの管理	285
データタイプサポートレポート	285
数値カラムの精度と小数点以下の桁数の変更	286
HP Vertica 環境のカスタマイズ	286
タイムアウト制限の指定	286
TIMEOUT コマンドの発行	286
長時間実行中リクエストのキャンセル	287
長時間実行中のリクエストをキャンセルするには	287
更新または削除された行数の取得	287
更新または削除された行数の取得	288
HP Vertica 最適化の設定	288
JDBC アダプタの使用	289

JDBC 環境の準備	289
Windows および UNIX で環境を設定するには	290
JDBC アダプタの構成	290
アダプタの構成	291
JDBC の接続属性	292
接続属性の手動宣言	293
接続属性の宣言	294
デフォルト接続の上書き変更	294
デフォルト接続の変更	295
デフォルト接続の選択	295
接続範囲の制御	296
接続範囲の制御	296
JDBC メタデータの管理	296
アダプタの識別	297
アダプタの識別	297
データベーステーブルへのアクセス	297
シノニムの作成	298
シノニムを作成するには	298
JDBC のシノニム作成パラメータ	299
生成されたシノニムの例	303
アクセスファイルのキーワード	303
シノニムの管理	305
データタイプサポートレポート	305
数値カラムの精度と小数点以下の桁数の変更	305
JDBC 環境のカスタマイズ	306
タイムアウト制限の指定	306
TIMEOUT コマンドの発行	306
長時間実行中リクエストのキャンセル	307
長時間実行中のリクエストをキャンセルするには	307
更新または削除された行数の取得	307
更新または削除された行数の取得	307
JDBC 最適化の設定	308

JSON アダプタの使用	309
JSON 環境の準備	309
JSON アダプタの構成	309
アダプタを構成するには	309
JSON ドキュメントとマスターファイルの関連付け	310
JSON ドキュメントとマスターファイルの関連付け	311
JSON カラムに対する FILEDEF コマンドの発行	312
JSON コレクションに対する FILEDEF コマンドの発行	312
JSON メタデータの管理	313
シノニムの作成	313
シノニムを作成するには	313
JSON のシノニム作成パラメータ	315
シノニムの管理	317
リレーショナル DBMS の JSON データタイプからの JSON ドキュメントへのアクセス	317
WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのツールを使用して RDBMS から JSON データにアクセスするには	318
RDBMS から JSON データに手動でアクセスするには	320
変換	324
数値	324
JSON の日付	324
JSON での日付の使用	325
Microsoft Access アダプタの使用	326
Microsoft Access 環境の準備	326
Windows で環境を設定するには	326
Microsoft Access アダプタの構成	326
接続属性の宣言	326
アダプタの構成	327
Microsoft Access の接続属性	328
接続属性の手動宣言	329
接続属性の宣言	330

接続情報の更新	330
デフォルト接続の上書き変更	331
デフォルト接続の変更	331
デフォルト接続の選択	332
接続範囲の制御	332
接続範囲の制御	332
Microsoft Access メタデータの管理	333
シノニムの作成	333
シノニムを作成するには	333
Microsoft Access のシノニム作成パラメータ	335
生成されたシノニムの例	338
アクセスファイルのキーワード	338
シノニムの管理	340
Microsoft Access データタイプのサポート	340
国際言語サポートの有効化	340
国際言語サポートの有効化	341
数値カラムの精度と小数点以下の桁数の変更	342
Microsoft Azure Synapse Analytics アダプタの使用	343
Microsoft Azure Synapse Analytics 環境の準備	343
Microsoft Azure Synapse Analytics アダプタの構成	343
接続属性の宣言	344
アダプタの構成	344
Microsoft Azure Synapse Analytics の接続属性	345
Windows での接続属性の手動宣言	347
デフォルト接続の上書き変更	348
デフォルト接続の変更	348
デフォルト接続の選択	349
接続範囲の制御	349
接続範囲の制御	349
Microsoft Azure Synapse Analytics メタデータの管理	350
シノニムの作成	350

シノニムを作成するには	351
Microsoft Azure Synapse Analytics のシノニム作成パラメータ	352
生成されたシノニムの例	355
Microsoft SQL ODBC テーブルコメントのシノニムへのマッピング	356
アクセスファイルのキーワード	356
シノニムの管理	359
Microsoft Azure Synapse Analytics データタイプのサポート	359
SQL 式内での末尾ブランク	359
数値カラムの精度と小数点以下の桁数の変更	359
読み取り専用フィールドのサポート	360
読み取り専用フィールドのサポート	360
Microsoft Azure Synapse Analytics のストアードプロシジャに対するレポー トの実行	361
ストアードプロシジャのシノニムの作成	361
Microsoft Azure Synapse Analytics ストアドプロシジャ CustOrders のシノニ ム	362
ストアードプロシジャのシノニム作成パラメータ	363
ストアードプロシジャに対するレポートの実行	365
TABLE コマンドを使用したストアードプロシジャに対するレポートの実行	365
SELECT を使用したストアードプロシジャに対するレポートの実行	366
Microsoft Azure Synapse Analytics 環境のカスタマイズ	367
タイムアウト制限の指定	368
TIMEOUT コマンドの発行	368
長時間実行中リクエストのキャンセル	368
長時間実行中のリクエストをキャンセルするには	369
ログイン待機時間の指定	369
ログイン待機時間の指定	369
更新または削除された行数の取得	370
更新または削除された行数の取得	370
Microsoft Azure Synapse Analytics 最適化の設定	371
一時項目 (DEFINE) に NULL 値が含まれる場合のリクエストの最適化	371
強化された集計制御の設定	371

常に FALSE になる条件を含む IF-THEN-ELSE 最適化の使用	372
DEFINE 式の最適化での SQL 制限	372
検索処理でのブロックサイズの指定	373
配列検索のブロックサイズの指定	374
挿入処理のブロックサイズの指定	374
バルクインサート API の省略	375
バルクインサート API の動作	376
非等価条件 WHERE ベース LEFT OUTER JOIN の最適化	376
条件付き LEFT OUTER JOIN の指定	376
WHERE ベース OUTER JOIN の最適化条件	378
非等価条件 LEFT OUTER JOIN の最適化	379
SQL パススルーによる Microsoft Azure Synapse Analytics ストアドプロシ ジャの呼び出し	380
ストアドプロシジャの呼び出し	380
ストアドプロシジャの呼び出し	381
ストアドプロシジャの例	382
ストアドプロシジャのアプリケーションエラーの収集	383
Microsoft SQL Server アダプタの使用	384
Microsoft SQL Server 環境の準備	384
Windows で環境を設定するには (OLE DB)	384
UNIX で環境を設定するには	384
Microsoft SQL Server のリモートアクセス	385
XA のサポート	386
Microsoft SQL Server アダプタの構成	386
接続属性の宣言	386
アダプタの構成	387
Microsoft SQL Server の接続属性	388
Windows での接続属性の手動宣言	390
Windows での接続属性の宣言	392
UNIX での接続属性の手動宣言	392
UNIX での接続属性の宣言	393
デフォルト接続の上書き変更	393

デフォルト接続の変更	393
デフォルト接続の選択	394
接続範囲の制御	394
接続範囲の制御	395
Microsoft SQL Server メタデータの管理	395
シノニムの作成	395
シノニムを作成するには	397
Microsoft SQL Server のシノニム作成パラメータ	398
生成されたシノニムの例	402
Microsoft SQL テーブルのコメントのシノニムへのマッピング	402
アクセスファイルのキーワード	403
シノニムの管理	405
Microsoft SQL Server データタイプのサポート	405
可変長データタイプのマッピングの制御	405
可変長データタイプのマッピングの制御	406
国際言語サポートの有効化	407
国際言語サポートの有効化	408
SQL 式内での末尾ブランク	409
数値カラムの精度と小数点以下の桁数の変更	409
読み取り専用フィールドのサポート	409
読み取り専用フィールドのサポート	410
Microsoft SQL Server のストアドプロシジャに対するレポートの実行	410
ストアドプロシジャのシノニムの作成	411
Microsoft SQL Server ストアドプロシジャ CustOrders のシノニム	412
ストアドプロシジャのシノニム作成パラメータ	412
ストアドプロシジャに対するレポートの実行	414
TABLE コマンドを使用したストアドプロシジャに対するレポートの実行	415
SELECT を使用したストアドプロシジャに対するレポートの実行	416
Microsoft SQL Server 環境のカスタマイズ	417
タイムアウト制限の指定	417
TIMEOUT コマンドの発行	417
長時間実行中リクエストのキャンセル	418

長時間実行中のリクエストをキャンセルするには	418
カーソルタイプの指定	419
カーソルタイプの指定	419
ログイン待機時間の指定	420
ログイン待機時間の指定	420
NONBLOCK モードの有効化	420
NONBLOCK モードの有効化	421
更新または削除された行数の取得	421
更新または削除された行数の取得	422
トランザクションの制御	422
トランザクションの制御	422
トランザクションの分離レベルの指定	423
SET コマンドによるトランザクション分離レベルの指定	424
Microsoft SQL Server 最適化の設定	424
一時項目 (DEFINE) に NULL 値が含まれる場合のリクエストの最適化	425
強化された集計制御の設定	425
常に FALSE になる条件を含む IF-THEN-ELSE 最適化の使用	426
DEFINE 式の最適化での SQL 制限	426
検索処理でのブロックサイズの指定	427
配列検索のブロックサイズの指定	428
挿入処理のブロックサイズの指定	428
バルクインサート API の省略	429
バルクインサート API の動作	429
非等価条件 WHERE ベース LEFT OUTER JOIN の最適化	430
条件付き LEFT OUTER JOIN の指定	430
WHERE ベース OUTER JOIN の最適化条件	432
非等価条件 LEFT OUTER JOIN の最適化	433
ヒントによるオプティマイザの効率化	433
ヒントの設定	434
Microsoft SQL ヒントの設定	434
SQL パススルーによる Microsoft SQL Server ストアドプロシジャの呼び出し	435

ストアードプロシジャの呼び出し	435
ストアードプロシジャの呼び出し	436
ストアードプロシジャの例	436
ストアードプロシジャのアプリケーションエラーの収集	437
Microsoft SQL Server ODBC の互換性	438
Microsoft SQL Server ODBC の接続属性	438
Microsoft SQL Server ODBC のカーソル	439
UNIQUEIDENTIFIER および BIT データタイプのマッピング	439
Microsoft SQL Server ODBC アダプタの使用	440
Microsoft SQL Server ODBC 環境の準備	440
Microsoft SQL Server のリモートアクセス	440
Microsoft SQL Server ODBC アダプタの構成	441
接続属性の宣言	441
アダプタの構成	442
Microsoft SQL Server ODBC の接続属性	443
Windows での接続属性の手動宣言	444
デフォルト接続の上書き変更	445
デフォルト接続の変更	445
デフォルト接続の選択	446
接続範囲の制御	446
接続範囲の制御	446
Microsoft SQL Server ODBC メタデータの管理	447
シノニムの作成	447
シノニムを作成するには	448
Microsoft SQL Server ODBC のシノニム作成パラメータ	450
生成されたシノニムの例	453
Microsoft SQL ODBC テーブルコメントのシノニムへのマッピング	453
アクセスファイルのキーワード	454
常時暗号化のサポート	456
シノニムの管理	456
Microsoft SQL Server ODBC データタイプのサポート	457

SQL 式内での末尾ブランク	457
数値カラムの精度と小数点以下の桁数の変更	457
読み取り専用フィールドのサポート	458
読み取り専用フィールドのサポート	458
Microsoft SQL Server ODBC のストアードプロシージャに対するレポートの実 行	458
ストアードプロシージャのシノニムの作成	459
Microsoft SQL Server ODBC ストアドプロシージャ CustOrders のシノニム	460
ストアードプロシージャのシノニム作成パラメータ	460
ストアードプロシージャに対するレポートの実行	462
TABLE コマンドを使用したストアードプロシージャに対するレポートの実行	463
SELECT を使用したストアードプロシージャに対するレポートの実行	464
Microsoft SQL Server ODBC アダプタの設定	465
バルクロードサービス	465
データタイプマッピングのカスタマイズ	467
機能診断	468
メタデータ	469
その他の設定	470
読み取り/書き込みの最適化	470
DBMS セッションパラメータ	471
SQL の生成	472
Microsoft SQL Server ODBC 環境のカスタマイズ	473
タイムアウト制限の指定	473
TIMEOUT コマンドの発行	474
長時間実行中リクエストのキャンセル	474
長時間実行中のリクエストをキャンセルするには	474
ログイン待機時間の指定	475
ログイン待機時間の指定	475
更新または削除された行数の取得	475
更新または削除された行数の取得	476
トランザクションの分離レベルの指定	476
SET コマンドによるトランザクション分離レベルの指定	476

Microsoft SQL Server ODBC 最適化の設定	477
一時項目 (DEFINE) に NULL 値が含まれる場合のリクエストの最適化	478
強化された集計制御の設定	478
常に FALSE になる条件を含む IF-THEN-ELSE 最適化の使用	479
DEFINE 式の最適化での SQL 制限	479
検索処理でのブロックサイズの指定	480
配列検索のブロックサイズの指定	481
挿入処理のブロックサイズの指定	481
バルクインサート API の省略	482
バルクインサート API の動作	482
非等価条件 WHERE ベース LEFT OUTER JOIN の最適化	483
条件付き LEFT OUTER JOIN の指定	483
WHERE ベース OUTER JOIN の最適化条件	485
非等価条件 LEFT OUTER JOIN の最適化	486
ヒントによるオプティマイザの効率化	486
ヒントの設定	487
Microsoft SQL ヒントの設定	487
SQL パススルーによる Microsoft SQL Server ODBC ストアドプロシジャの 呼び出し	488
ストアドプロシジャの呼び出し	488
ストアドプロシジャの呼び出し	489
ストアドプロシジャの例	490
ストアドプロシジャのアプリケーションエラーの収集	491
MySQL アダプタの使用	492
MySQL 環境の準備	492
Windows で MySQL 環境を準備するには	492
UNIX で MySQL 環境を準備するには	493
MySQL と Unicode	495
MySQL アダプタの構成	495
接続属性の宣言	495
アダプタの構成	496
MySQL の接続属性	497

接続属性の手動宣言	499
接続属性の宣言	500
ユーザの認証	500
デフォルト接続の上書き変更	500
アクセスする接続の選択	501
アクセスする接続の選択	501
接続範囲の制御	502
接続範囲の制御	502
MySQL メタデータの管理	503
シノニムの作成	503
シノニムを作成するには	503
MySQL のシノニム作成パラメータ	504
生成されたシノニムの例	508
アクセスファイルのキーワード	508
シノニムの管理	510
MySQL データタイプのサポート	511
数値カラムの精度と小数点以下の桁数の変更	511
MySQL アダプタ環境のカスタマイズ	511
PASSRECS	511
PASSRECS の設定	512
トランザクションの分離レベルの指定	512
SET コマンドによるトランザクション分離レベルの指定	512
長時間実行中リクエストのキャンセル	513
長時間実行中のリクエストをキャンセルするには	513
MySQL 最適化の設定	514
Netezza アダプタの使用	515
Netezza 環境の準備	515
Windows および UNIX で環境を設定するには	515
Netezza の Unicode サポート	516
Netezza アダプタの構成	516
接続属性の宣言	517

アダプタの構成	517
Netezza JDBC の接続属性	518
接続属性の手動宣言	520
接続属性の宣言	520
デフォルト接続の上書き変更	521
デフォルト接続の変更	521
デフォルト接続の選択	522
Netezza メタデータの管理	522
アダプタの識別	522
アダプタの識別	522
データベーステーブルへのアクセス	523
シノニムの作成	523
シノニムを作成するには	523
Netezza のシノニム作成パラメータ	525
生成されたシノニムの例	528
アクセスファイルのキーワード	528
シノニムの管理	530
Netezza データタイプのサポート	530
数値カラムの精度と小数点以下の桁数の変更	531
Netezza 環境のカスタマイズ	531
タイムアウト制限の指定	531
TIMEOUT コマンドの発行	531
長時間実行中リクエストのキャンセル	532
長時間実行中のリクエストをキャンセルするには	532
更新または削除された行数の取得	532
更新または削除された行数の取得	533
HOLD DBMS 作成の制御	533
DBMS 作成の制御	533
Netezza テーブルの作成	534
Netezza 最適化の設定	534
非等価条件 WHERE ベース LEFT OUTER JOIN の最適化	534
条件付き LEFT OUTER JOIN の指定	535

WHERE ベース OUTER JOIN の最適化条件	537
非等価条件 LEFT OUTER JOIN の最適化	537
ODBC アダプタの使用	539
ODBC 環境の準備	539
ODBC アダプタの構成	539
接続属性の宣言	540
アダプタの構成	540
ODBC の接続属性	541
接続属性の手動宣言	543
接続属性の宣言	544
接続情報の更新	545
デフォルト接続の上書き変更	546
デフォルト接続の変更	546
デフォルト接続の選択	547
接続範囲の制御	547
接続範囲の制御	547
ODBC メタデータの管理	548
アダプタの識別	548
アダプタの識別	548
データベーステーブルへのアクセス	548
シノニムの作成	549
シノニムを作成するには	549
ODBC のシノニム作成パラメータ	550
生成されたシノニムの例	554
アクセスファイルのキーワード	554
シノニムの管理	556
データタイプのサポート	556
数値カラムの精度と小数点以下の桁数の変更	557
ODBC 環境のカスタマイズ	557
タイムアウト制限の指定	557
TIMEOUT コマンドの発行	557

更新または削除された行数の取得	558
更新または削除された行数の取得	558
トランザクションの分離レベルの指定	559
SET コマンドによるトランザクション分離レベルの指定	559
ODBC 最適化の設定	560
Oracle アダプタの使用	561
Oracle 環境の準備	561
Windows で Oracle 環境を準備するには	561
UNIX で Oracle 環境を準備するには	561
リモートの Oracle データベースサーバへの接続	563
XA のサポート	563
Unicode のための Oracle の構成	563
NLS_LANG 環境変数の設定	564
NLS_LANG 変数に英語 (米国) UTF-8 を設定する例	564
Oracle アダプタの構成	565
接続属性の宣言	565
アダプタの構成	566
Oracle の接続属性	567
接続属性の手動宣言	568
接続属性の宣言	569
接続情報の更新	570
デフォルト接続の上書き変更	571
デフォルト接続の変更	571
デフォルト接続の選択	572
接続範囲の制御	572
接続範囲の制御	572
Oracle メタデータの管理	573
シノニムの作成	573
シノニムを作成するには	574
Oracle のシノニム作成パラメータ	575
生成されたシノニムの例	578

Oracle のコメントのシノニムへのマッピング	579
アクセスファイルのキーワード	579
シノニムの管理	581
単一の SQL リクエストでの複数のデータベースサーバへのアクセス	581
Oracle データタイプのサポート	582
可変長データタイプのマッピングの制御	582
可変長データタイプのマッピングの制御	583
CHAR および VARCHAR2 データタイプについて	584
ORACHAR の設定	584
SQL 式内での末尾ブランク	585
数値カラムの精度と小数点以下の桁数の変更	585
NUMBER データタイプについて	585
ORANUMBER の設定	586
Oracle アダプタの設定	586
Oracle ストアドプロシジャに対するレポートの実行	592
ストアドプロシジャのシノニムの作成	593
ストアドプロシジャのシノニム作成パラメータ	594
ストアドプロシジャに対するレポートの実行	596
TABLE コマンドを使用したストアドプロシジャに対するレポートの実行	596
SELECT を使用したストアドプロシジャに対するレポートの実行	597
Oracle 環境のカスタマイズ	598
システム日付および時刻のソースの選択	598
システム日時のソースの選択	599
デフォルトのテーブルスペースの指定	599
DBSPACE の設定	599
インデックススペースのデフォルトのパラメータの上書き変更	600
IXSPACE の設定	600
NONBLOCK モードの有効化	601
NONBLOCK モードの有効化	602
更新または削除された行数の取得	602
更新または削除された行数の取得	603
ストアドプロシジャのパラメータの最大数の指定	603

SPMAXPRM の設定	603
Oracle 最適化の設定	604
一時項目 (DEFINE) に NULL 値が含まれる場合のリクエストの最適化	604
強化された集計制御の設定	605
集計リクエストの最適化	605
検索処理でのブロックサイズの指定	606
配列検索のブロックサイズの指定	607
挿入処理のブロックサイズの指定	607
集計の最適化による効率の改善	608
集計の最適化の設定	608
非等価条件 WHERE ベース LEFT OUTER JOIN の最適化	609
条件付き LEFT OUTER JOIN の指定	609
WHERE ベース OUTER JOIN の最適化条件	611
非等価条件 LEFT OUTER JOIN の最適化	612
ヒントによるオプティマイザ効率の改善	612
ヒントの設定	613
Oracle ヒントの組み合わせ設定	613
SQL パススルーによる Oracle ストアドプロシジャの呼び出し	614
ストアドプロシジャの呼び出し	614
REFCURSOR を使用する Oracle ストアドプロシジャ	616
REFCURSOR を使用しない Oracle ストアドプロシジャ	617
PostgreSQL アダプタの使用	619
PostgreSQL 環境の準備	619
Windows および UNIX で環境を設定するには	619
PostgreSQL アダプタの構成	620
接続属性の宣言	620
アダプタの構成	621
PostgreSQL の接続属性	622
接続属性の手動宣言	623
接続属性の宣言	624
デフォルト接続の上書き変更	624

デフォルト接続の変更	625
デフォルト接続の選択	625
接続範囲の制御	626
接続範囲の制御	626
PostgreSQL メタデータの管理	626
アダプタの識別	626
アダプタの識別	627
データベーステーブルへのアクセス	627
シノニムの作成	627
シノニムを作成するには	628
PostgreSQL のシノニム作成パラメータ	629
生成されたシノニムの例	632
アクセスファイルのキーワード	633
シノニムの管理	634
PostgreSQL データタイプのサポート	635
A256V のデータタイプマッピング	635
数値カラムの精度と小数点以下の桁数の変更	635
PostgreSQL 環境のカスタマイズ	636
タイムアウト制限の指定	636
TIMEOUT コマンドの発行	636
長時間実行中リクエストのキャンセル	636
長時間実行中のリクエストをキャンセルするには	637
更新または削除された行数の取得	637
更新または削除された行数の取得	637
PostgreSQL 最適化の設定	638
Python アダプタの使用	639
WebFOCUS で使用する Python スクリプト記述に関するガイドライン	639
Python アダプタの使用要件	642
Python アダプタの構成	644
Python アダプタを構成するには	644
Python 関数のシノニムの作成	645

Python 関数のシノニムを作成するには	646
ユーザ定義の Python スクリプトの実行	648
ユーザ定義の Python 関数の実行	648
Python スクリプトの実行	648
WebFOCUS Reporting Server リモートサーバアダプタの使用	652
リモート ibi WebFOCUS Reporting Server の構成	652
アダプタの構成	652
接続属性	653
メタデータの管理	656
シノニムの作成	656
シノニムを作成するには	657
シノニム作成パラメータ	658
リモートデータソースのシノニム	660
シノニムの管理	662
アクセスファイルのキーワード	662
アクセスファイルへのサーバ名の保存	663
SUFFIX=EDA シノニムの使用	663
リモートの複数セグメントマスターファイルおよびアクセスファイル	665
数値カラムの精度と小数点以下の桁数の変更	666
ストアードプロシジャの実行	667
次のリクエストへの接続の設定	667
セッションの以降の接続を設定	667
接続ステータスのテスト	668
SET CONNECTION_ATTRIBUTES による代替 ID とパスワードの組み合わせ の使用	668
REMOTE USER および REMOTE PASS での代替 ID とパスワードの組み合わ せの使用	669
使用可能なサーバの確認	669
リモート実行を使用したストアードプロシジャの実行	670
リモート接続でのローカル保存コマンドの実行	670
クエリアダプタの設定	671

REST アダプタの使用	673
REST アダプタの構成	673
接続属性の宣言	673
アダプタの構成	674
REST アダプタの接続属性	675
RESTful Web サービスのセキュリティ	680
連鎖実行用のアダプタを構成するには	681
連鎖認証の構成	683
認証作業への追加パラメータの受け渡し	684
認証作業への追加パラメータの受け渡し	685
RESTful Web サービスのメタデータの管理	686
シノニムの作成	686
シノニムを作成するには	686
RESTful Web サービスのシノニム作成パラメータ	688
シノニムの管理	691
マスターファイルの属性	691
アクセスファイルの属性	692
文字フィールドのデフォルトフィールド長の設定	693
POST リクエストの JSON シノニム での XDEFAULT 属性自動挿入	693
サンプル REST シノニム	693
REST POST リクエストでのテキストデータおよびバイナリデータの送信	696
REST POST リクエストからのバイナリデータの読み取りおよび出力のファイル保存	697
POST リクエストでのマルチパート/フォームデータの送信	698
REST レスポンスでの CSV フォーマットのサポート	700
JSON 本文での値が存在しないラベルのサポート	704
JSON 本文での値が存在しないラベルの送信	705
WHERE 句でのパラメータ化された URL への複数値の送信	705
Salesforce.com アダプタの使用	708
Salesforce.com アダプタの構成	708
アダプタの構成	708

Salesforce.com の接続パラメータ	709
Salesforce.com による一括取得	713
Salesforce.com でのシノニムの作成	716
Salesforce.com オブジェクトのシノニムを作成するには	716
Salesforce.com オブジェクトのクラスタ JOIN を作成するには	718
削除済みレコードの取得の制御	722
ブロックサイズの制御	723
Salesforce.com の日付リテラルおよび日付定数の使用	723
Salesforce.com による集計の無効化	724
SAP BW アダプタの使用	725
SAP BW 環境の準備	725
権限オブジェクト、権限フィールド、および必須値のリスト	726
複数のシステムへのアクセス	729
SAP BW アダプタの構成	730
接続属性の宣言	730
アダプタの構成	730
SAP BW の接続属性	731
SAP BW アダプタの混合コードページ環境のサポート	735
文字変換テーブル	735
変換テーブルを生成するには	735
変換テーブルをサーバにダウンロードするには	737
BEx クエリの作成	741
SAP BW の用語	741
BEx クエリの用語	742
新規ビジネスエクスプローラクエリの定義	743
クエリに対するインフォプロバイダを選択するには	744
クエリを作成するには	744
クエリをフィルタするには	745
クエリの特性の制限	745
特性を制限するには	746
キー数値の制限と計算	746

キー数値を制限するには	746
キー数値を計算するには	747
論理式を定義するには	748
クエリプロパティの表示および OLAP へのリリース	748
WebFOCUS による SAP BW レポートの作成	749
SAP BW レポートの概念の概要	749
カラムレポートおよび階層レポートの理解	751
階層レポートの要件	752
シノニムの階層表現	752
ディメンション宣言	753
レベル階層の定義	754
親子階層の定義	755
ディメンションプロパティ	757
レベル階層を使用したリクエストのサンプル	757
親子階層を使用したリクエストのサンプル	758
階層レポート	759
親子階層の表示	759
階層全体に対するレポート	762
階層メンバーの選択	764
メンバーの選択と親の追加	765
メンバーの選択と子の追加	766
メンバーの選択と上位レベルの一括表示	767
メンバーの選択と下位レベルの表示	768
メンバーの選択と上位および下位レベルの一括表示	769
階層レベル範囲のメンバーの表示	770
表示メンバーの選択とメンバー値による選別	771
メンバー選択とデータ選別の比較	775
WHEN を伴わない SHOW の使用	776
SKIP-LINE の使用	777
メジャーでの条件ソートオプションの使用	778
ディメンションプロパティでの条件ソートオプションの使用	780
HIERARCHY フィールドでの条件ソートオプションの使用	781
WHERE TOTAL によるメジャーの選別	782

2 つの BY HIERARCHY 句の使用	783
BY HIERARCHY 句と BY 句の使用	785
BY HIERARCHY を伴わない ON HIERARCHY の使用	787
複数の表示コマンドを伴う階層レポート	788
階層レポートでの複数表示コマンドの使用	788
カラムレポート	790
有効または無効なスライス - キューブのスライスロジック	790
スライスレポートでの単一のディメンションの表示	791
スライスレポートでの複数のディメンションの表示	791
変数を使用したレポート	792
完全集計および部分集計	793
完全集計モードのサポート	794
部分集計モードのサポート	795
時間依存階層サポート	796
マスターファイル内の時間依存階層	797
時間依存階層に対するレポートの作成	799
逆符号でのメジャーの表示	801
逆符号でのメジャーの表示	801
逆符号でのメジャーの表示	802
レポートの規則	804
表示コマンドおよび演算接頭語のサポート	804
すべての階層に対するレポートの規則	805
親子階層に対するレポートの規則	806
レポート作成の一般的なヒント	806
SAP BW の階層のサポート	807
動的階層の作成	807
動的ディメンションの作成	807
ディメンションの作成	808
SAP BW メタデータの管理	808
シノニムの作成	809
シノニムを作成するには	809
SAP BW の拡張シノニム作成パラメータ	811

複数言語のタイトルと説明	814
コードページおよび複数言語のシノニム	815
シノニムの管理	815
メタデータのマッピング	816
BW インフォプロバイダの OLE/DB キューブへのマッピング	816
マッピングの例	817
マスターファイルの構文規則	818
ディメンション - OMATERIAL	819
変数のタイプ	820
カスタマイズの設定	820
空のレポートのセル設定の指定	820
メンバーのサンプリング設定の指定	821
ブロックサイズの指定	822
プロパティの制限の指定	822
未割り当てメンバーの設定の指定	823
集計演算子の確認の無効化	824
時間依存階層に対するレポート作成	824
リクエストでの大規模インスタンスのサポート	825
リテラル文字列による SAP VARIABLES 句の作成	825
BEx 構造のサポート	826
構造をディメンションまたは直積として表す	827
SQL を使用した SAP BW リクエストの生成	830
SAP BW からの値の取得	830
階層からの項目の選択	831
階層の完全なパスの選択	832
SAP ERP アダプタの使用	842
SAP 環境の準備	842
USER 用の権限オブジェクト、権限フィールド、および必須値のリスト	843
バックグラウンド処理用の権限オブジェクトのリスト	846
アダプタコンポーネント用の SAP 環境の準備	846
SAP にログオンするには	846

開発クラスまたはパッケージを作成するには	849
汎用グループを作成するには	849
Unicode チェックを無効にするには	850
汎用グループを有効にするには	851
汎用グループを確認するには	852
ZXXX* 一時オブジェクトの移送/移動のカスタマイズ	853
移送制御	854
複数の SAP システムへのアクセス	854
SAP アダプタの構成	855
アダプタの構成	855
SAP の接続属性	856
SAP ECC または SAP S/4HANA アダプタを構成するには	861
接続範囲の制御	863
接続範囲の制御	864
SAP ECC への COMMIT リクエストの送信の制御	864
SAP ECC への COMMIT リクエストの送信の制御	865
SAP コンポーネントのインストールと検査	865
SAP コンポーネントをインストールするには	866
SAP コンポーネントのインストールを確認するには	866
SAP 環境での構成後の作業	867
ZXXXREPTS 一時オブジェクトの移送/移動のカスタマイズ	868
移送用のデータファイル名	868
変更依頼情報ファイル	869
依頼の移送	869
SAP の予約済みの名前	870
依頼の移送	871
SAP メタデータの管理	872
シノニムの作成	872
SAP シノニムのタイプ	873
シノニムを作成するには	874
テーブルシノニムの作成パラメータ	875
複数言語のタイトルと説明	877

コードページおよび複数言語のシノニム	877
LDB シノニムの作成パラメータ	878
論理データベースの制限事項	880
汎用モジュールシノニムの作成パラメータ	881
BAPI シノニムの作成パラメータ	883
汎用モジュールおよび BAPI の制限事項	885
QUERY シノニムの作成パラメータ	885
SAP クエリのシノニムのバリエーション	887
IDOC 用のシノニムの作成	888
SAP テーブルのシノニムの作成	889
シノニムの管理	890
個別のテーブルに対する SAP テーブルクラスのサポート	890
SAP 汎用モジュールのサポート	891
SAP データタイプサポート	894
SAP Open/SQL のサポート	897
SAP の高度な機能	898
ユーザセキュリティのサポート	898
カスタムセキュリティグジットの使用	898
サンプルユーザ権限チェック	899
シノニム作成時のサンプルグジット認可チェック	900
BAPI のサポート	901
結合のサポート	901
レポートの処理モードの設定	902
ダイアログおよびバッチ実行モードの設定	902
ポーリング間隔のバッチ実行への設定	903
SAP リクエストの生成	904
すべてのレコードの取得	904
選択したフィールドの取得	905
2 つのテーブルの結合	906
SAP Hana アダプタの使用	908
SAP Hana 環境の準備	908

Windows および UNIX で環境を設定するには	908
SAP Hana アダプタの構成	909
接続属性の宣言	909
アダプタの構成	910
SAP Hana の接続属性	911
接続属性の手動宣言	912
接続属性の宣言	913
デフォルト接続の上書き変更	913
デフォルト接続の変更	913
デフォルト接続の選択	914
SAP Hana メタデータの管理	914
アダプタの識別	915
アダプタの識別	915
シノニムの作成	915
シノニムを作成するには	916
SAP Hana のシノニム作成パラメータ	917
生成されたシノニムの例	920
アクセスファイルのキーワード	921
シノニムの管理	922
データタイプサポートレポート	923
数値カラムの精度と小数点以下の桁数の変更	923
SAP Hana 環境のカスタマイズ	923
タイムアウト制限の指定	923
TIMEOUT コマンドの発行	924
長時間実行中リクエストのキャンセル	924
長時間実行中のリクエストをキャンセルするには	924
更新または削除された行数の取得	925
更新または削除された行数の取得	925
SAP Hana 最適化の設定	925
Snowflake Cloud Data Warehouse アダプタの使用	926
Snowflake JDBC 環境の準備	926

Snowflake Cloud Data Warehouse ODBC 環境の準備	927
Snowflake Cloud Data Warehouse アダプタの構成 - JDBC	927
Snowflake Cloud Data Warehouse アダプタを構成するには - JDBC	927
Snowflake Cloud Data Warehouse アダプタの構成設定 - JDBC	928
Snowflake Cloud Data Warehouse アダプタの構成 - ODBC	931
Snowflake Cloud Data Warehouse アダプタを構成するには - ODBC	931
Snowflake Cloud Data Warehouse アダプタの構成設定 - ODBC	932
Snowflake Cloud Data Warehouse (JDBC) でのシノニムの作成	936
シノニムを作成するには	936
Snowflake Cloud Data Warehouse (JDBC) のシノニム作成パラメータ	938
Snowflake Cloud Data Warehouse (ODBC) でのシノニムの作成	940
シノニムを作成するには	941
Snowflake Cloud Data Warehouse (ODBC) のシノニム作成パラメータ	942
データタイプのサポート	944
Snowflake Cloud Data Warehouse アダプタの設定 - JDBC	945
バルクロードサービス	945
データタイプマッピングのカスタマイズ	947
メタデータ	948
その他の設定	948
Snowflake Cloud Data Warehouse アダプタの設定 - ODBC	949
バルクロードサービス	949
データタイプマッピングのカスタマイズ	951
メタデータ	952
その他の設定	952
Sybase アダプタの使用	954
Sybase 環境の準備 - OCS	954
interfaces ファイルのパスの特定	954
Sybase Server 名の指定	955
Sybase サーバ名の指定	955
リモートの Sybase Server へのアクセス	955
Unicode サポート	956

XA のサポート (ASE)	956
Sybase 環境の準備 - JDBC	957
Windows および UNIX で JDBC 環境を設定するには	957
Sybase アダプタの構成	957
接続属性の宣言	958
アダプタの構成	958
Sybase の接続属性 - OCS	959
Sybase の接続属性 - JDBC	960
接続属性の手動宣言	962
接続属性の宣言	962
デフォルト接続の上書き変更	963
デフォルト接続の変更	963
デフォルト接続の選択	964
接続情報の更新	964
接続範囲の制御	966
接続範囲の制御	966
Sybase メタデータの管理	966
シノニムの作成	967
シノニムを作成するには	967
Sybase のシノニム作成パラメータ	968
生成されたシノニムの例	972
Sybase IQ テーブルのコメントのシノニムへのマッピング	972
アクセスファイルのキーワード	973
シノニムの管理	975
同一 Sybase Server 上の別データベースへのアクセス	975
Sybase データタイプのサポート	976
可変長データタイプのマッピングの制御	976
Sybase ASE - 可変長データタイプのマッピングの制御	976
可変長データタイプのマッピングの制御 (Sybase ASE)	977
Sybase IQ - 可変長データタイプのマッピングの制御	978
可変長データタイプのマッピングの制御 (Sybase IQ)	978
SQL 式内での末尾ブランク	979

数値カラムの精度と小数点以下の桁数の変更	979
Sybase ストアドプロシジャに対するレポートの実行	980
ストアドプロシジャのシノニムの作成	980
ストアドプロシジャのシノニム作成パラメータ	981
ストアドプロシジャに対するレポートの実行	984
TABLE コマンドを使用したストアドプロシジャに対するレポートの実行	984
SELECT を使用したストアドプロシジャに対するレポートの実行	985
Sybase 環境のカスタマイズ	986
NONBLOCK モードの有効化	986
NONBLOCK モードの有効化	987
更新または削除された行数の取得	987
更新または削除された行数の取得	988
Sybase ASE および IQ でのパスワード暗号化の有効化	988
パスワード暗号化の有効化 (ASE および IQ)	989
Sybase 最適化の設定	990
検索処理でのブロックサイズの指定	990
配列検索のブロックサイズの指定	990
挿入処理のブロックサイズの指定	991
バルクインサート API の省略	991
バルクインサート API の動作	992
SQL パススルーを使用した Sybase のストアドプロシジャの呼び出し	992
Sybase ストアドプロシジャの呼び出し	993
Sybase のストアドプロシジャ	993
非連鎖の Sybase ストアドプロシジャの実行	993
Teradata アダプタの使用	995
Teradata 環境の準備	995
環境を ODBC または CLI を使用して Microsoft Windows 上に設定するには	995
UNIX で環境を設定するには	995
Teradata CLI および ODBC アダプタの Unicode 用の構成	996
Teradata アダプタの構成	996
接続属性の宣言	996

アダプタの構成	997
Teradata の接続属性	998
接続属性の手動宣言	999
接続属性の宣言	1000
Teradata 用の接続情報の更新	1001
デフォルト接続の上書き変更	1002
デフォルト接続の変更	1002
デフォルト接続の選択	1003
接続範囲の制御	1003
接続範囲の制御	1003
Teradata メタデータの管理	1004
シノニムの作成	1004
シノニムを作成するには	1004
Teradata のシノニム作成パラメータ	1005
生成されたシノニムの例	1009
Teradata コメントのシノニムへのマッピング	1009
アクセスファイルのキーワード	1010
シノニムの管理	1012
Teradata データタイプのサポート	1012
可変長データタイプのマッピングの制御	1012
可変長データタイプのマッピングの制御	1012
SQL 式内での末尾ブランク	1013
数値カラムの精度と小数点以下の桁数の変更	1013
Teradata ストアドプロシジャに対するレポートの実行	1014
ストアドプロシジャのシノニムの作成	1014
ストアドプロシジャのシノニム作成パラメータ	1015
ストアドプロシジャに対するレポートの実行	1017
TABLE コマンドを使用したストアドプロシジャに対するレポートの実行	1017
SELECT を使用したストアドプロシジャに対するレポートの実行	1018
Teradata 環境のカスタマイズ	1019
リクエストのカラム見出しの制御	1020
コマンドによるカラム見出しの制御	1020

NONBLOCK モードの有効化	1020
NONBLOCK モードの有効化	1021
更新または削除された行数の取得	1021
更新または削除された行数の取得	1022
Teradata 接続内でのトランザクションモードの設定	1022
Teradata 接続のトランザクションモードの設定	1022
Teradata 最適化の設定	1023
挿入処理のブロックサイズの指定	1024
挿入処理のブロックサイズの指定	1024
集計の最適化による効率の改善	1024
非等価条件 WHERE ベース LEFT OUTER JOIN の最適化	1025
条件付き LEFT OUTER JOIN の指定	1025
WHERE ベース OUTER JOIN の最適化条件	1027
非等価条件 LEFT OUTER JOIN の最適化	1028
SQL パススルーを使用した Teradata マクロまたはストアードプロシジャの 呼び出し	1028
マクロの呼び出し	1029
ストアードプロシジャの呼び出し	1029
Web サービスアダプタの使用	1031
Web サービスアダプタの構成	1031
接続属性の宣言	1031
アダプタの構成	1032
Web サービスの接続属性	1033
接続属性の手動宣言	1036
接続属性の宣言	1038
Web サービスのセキュリティ	1038
連鎖実行用のアダプタを構成するには	1039
連鎖認証の構成	1041
認証作業への追加パラメータの受け渡し	1042
Web サービスのメタデータの管理	1043
シノニムの作成	1044
シノニムを作成するには	1044

Web サービスのシノニム作成パラメータ	1045
シノニムの管理	1047
マスターファイルの属性	1048
アクセスファイルの属性	1048
静的 JOIN の使用	1050
ibi Data Migrator デスクトップインターフェースのシノニムエディタによる 静的 JOIN の追加	1051
JOIN オプション	1055
WSDL ファイルの概要	1056
FindAddress Web サービスのマスターファイルを生成する WSDL セグメン ト	1057
SOAP ヘッダを使用した WSDL ドキュメント	1062
出力ヘッダを使用した WSDL ドキュメント	1064
複数のヘッダ	1065
複合データタイプを使用した WSDL ドキュメント	1065
抽象入力データタイプを使用した WSDL ドキュメント	1067
属性とデフォルト値のマッピング	1067
属性とデフォルト値をマッピングした WSDL ドキュメント	1068
入力値としての配列の使用	1068
入力値としての配列の使用	1069
XML 要素とその属性への入力値としての配列の使用	1071
XML 要素とその属性への入力値としての配列の使用	1071
WSDL スキーマでのネームスペース修飾子の編集	1072
データタイプのサポート	1073
日付時間の処理	1075
ISO 標準日付時間表記の有効化	1076
プロシジャに FILEDEF SOAPTSCQ を使用した SOAP リクエストの取り込 み	1076
FILEDEF SOAPTSCG を使用した SOAP リクエストの取り込み	1076
WebFOCUS Client アダプタの使用	1078
アダプタを構成するには	1078
アダプタでのシノニムとサンプルの作成	1079

XML アダプタの使用	1080
XML 環境の準備	1080
XML アダプタの構成	1080
アダプタの構成	1080
XML メタデータの管理	1081
シノニムの作成	1081
シノニムを作成するには	1082
XML のシノニム作成パラメータ	1083
empxml xsd のシノニムの作成	1085
スキーマサブセットからのシノニム作成	1088
シノニムの管理	1090
リレーショナル DBMS の XML データタイプからの XML ドキュメントへの アクセス	1090
WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのツールを使用して RDBMS から XML データにアクセスするには	1091
RDBMS から XML データに手動でアクセスするには	1092
静的 JOIN の使用	1094
SQL Server によってエクスポートされたスキーマから生成されたマスター ファイルとアクセスファイルの変更	1095
JOIN 構造の例 - 元の構造と変更後の構造	1096
データタイプのサポート	1106
ACTUAL および USAGE 属性の設定	1109
文字列の長さの変更	1109
可変長と固定長の文字列の切り替え	1109
文字列長さの変更	1110
数値カラムの精度と小数点以下の桁数の変更	1111
変換	1111
数値	1111
XML の日付	1111
XML での日付の使用	1112
XML ドキュメントとマスターファイルの関連付け	1112

FILEDEF を使用した XML ドキュメントとマスターファイルの関連付け	1113
XA のサポート	1116
XA トランザクション管理	1116
サポート対象のインターフェース	1117
実装	1117
ベンダー環境	1118
XA 構成ファイルの作成	1119
集計の最適化のサポート	1120
リレーショナルアダプタおよび集計 SQL クエリ	1120
RDBMS での集計の最適化	1120
集計の最適化の設定	1121
集計最適化の使用時の注意	1121
クラスタ JOIN	1123
埋め込み JOIN	1123
埋め込み JOIN のマスターファイル	1124
マスターファイルでの等価 JOIN の定義	1124
COBOL ファイル記述の変換	1127
COBOL ファイル記述によるシノニムの作成	1127
COBOL ファイル記述の変換の制御	1128
COBOL ファイル記述のカスタマイズオプション	1128
REDEFINE フィールド	1131
REDEFINE フィールドの生成	1131
コメントとしてのレベル 88	1133
コメントとしてのレベル 88 の生成	1133
セグメントとしての OCCURS	1134
単純な固定フォーマットの OCCURS	1134
可変フォーマットの OCCURS	1135
ネストされた OCCURS	1136
ORDER フィールド	1137
ネストされた OCCURS での ORDER フィールドの順序	1138

ゾーン数値フィールドの使用	1138
ゾーン数値フィールドの使用	1138
数値フィールド編集オプション	1140
数値フィールド編集オプション	1140
フィールド名情報	1141
フォーマット変換に関する一般的注意事項	1143
INPUT としての複数レコード	1146
INPUT としての複数レコード	1146
フィールド最大数	1147
フィールド最大数	1147
2000 年	1149
COBOL FD の構文要件	1149
COBOL FD 構文のガイドライン	1149
データセット圧縮イグジット - ZCOMP	1151
ZCOMP イグジットの呼び出し	1151
ZCOMP のロードの結果	1153
レコードの転送	1153
ZCOMP C ソースファイル (zcomp.c)	1153
ZCOMP ヘッダファイル (zcomp.h)	1156
ZCOMP1 BAL パラメータリスト	1158
ダイナミックプライベートユーザイグジット	1160
FOCSAM および GETPRV ユーザイグジット	1160
ダイナミックプライベートユーザイグジットを使用するには	1161
GETPRV イグジットの全般機能	1161
GETPRV イグジット使用時の機能上の要件	1162
GETPRV イグジットの物理実装	1164
GETPRV でのデータアクセスに使用するマスターファイル	1164
GETPRV でのデータアクセスに使用するアクセスファイル	1165
呼び出しパラメータと作業領域	1166
呼び出しパラメータ	1166
作業領域制御ブロックパラメータ	1169

呼び出しパラメータおよび作業領域制御ブロックパラメータのサンプル C ヘッダファイル実装	1172
SUFFIX=PRIVATE のサンプル (getprv.c)	1174
特殊文字および予約語の確認	1177
特殊文字の確認	1177
予約語の確認	1178
Legal and Third-Party Notices	1180

アダプタの概要

アダプタの使用方法について説明します。

クライアント/サーバアーキテクチャでは、クライアントはアダプタを使用することにより、任意のオペレーティングシステムのデータソースのデータを、SQL ステートメントを通じて仮想的に管理することができます。

クライアントは、ibi™ WebFOCUS® Reporting Server 上のデータを処理するリクエストを生成します。WebFOCUS® Reporting Server はデータソースとして機能し、複数のクライアントからデータアクセスおよび処理を実行するリクエストを受け取ります。クライアント/サーバアーキテクチャでは、従来の単一のシステムはフロントエンドとバックエンドに分割されます。ワークロードはクライアントとサーバ間で分散されます。通信ソフトウェアによってクライアントとサーバ間のリンクが確立され、任意の通信プロトコルとインタフェースが確立されます。

リクエストの処理

サーバは SQL リクエストを処理することができます。これにより SQL を使用するクライアントアプリケーションを幅広くサポートすることができます。SQL は、リレーショナルおよび非リレーショナルデータを処理するリクエストの標準のアクセス言語です。

また、サーバは SQL より強力な機能の ibi™ WebFOCUS® Language と呼ばれる 4-GL リクエスト言語を備えており、広範なリクエストを処理することができます。特にアーキテクチャがリレーショナルモデルに基づいていない、非リレーショナルのデータソースに対するリクエストを処理することが可能です。

クライアント/サーバアーキテクチャのサーバは、データアクセスおよびデータ処理に使用されます。サーバはリクエストのデータを受け取り、処理して、クライアントにアンサーセットまたはメッセージを返します。サーバを構成するさまざまなサブシステム内にアダプタが含まれています。

アダプタの機能

サーバは、アダプタを使用してデータソースにアクセスします。サーバは、クライアントから SQL または の 2 つのタイプのリクエストを受け取ることができます。SQL または WebFOCUS® の 2 つのタイプのリクエストを受け取ることができます。

サーバがクライアントから受け取ったリクエストは、標準のフォーマットでアダプタに渡されます。アダプタはリクエストを受け取り、固有のデータ処理言語 (DML) に変換し、その API を使用してデータソースへの呼び出しを実行します。このようなアダプタの動作によって、サーバはデータソースの複雑な処理から解放されます。アプリケーションでは、SQL ステートメント、WebFOCUS® コマンド、ストアドプロシジャを実行または呼び出すことができます。

多様なデータソースに対応したアダプタが用意されています。アダプタはそれぞれアクセスするデータソース向けに設計されているため、SQL または WebFOCUS とデータソースの DML 間での変換が可能です。アダプタは、構文、機能、スキーマ、データタイプ、カタログ、データ表現、メッセージ処理、アンサーセットの抽出の相違など、製品間の差異を吸収するためのソリューションを提供します。

読み取り/書き込みに関する考慮事項

アダプタに読み取り/書き込み機能がある場合、アプリケーションからデータソースにデータが追加されます。

これを実行するには、データソースへの接続でアダプタが使用するアプリケーション ID に適切な権限が与えられる必要があります。ほとんどの SQL データソースでは、アダプタに書き込みを許可する通常の GRANTS が、CONNECT、RESOURCE ロールに追加されます。正確な権限セットについては、各ベンダーのマニュアルを参照してください。

アダプタの処理

アダプタは、データ管理リクエストをデータソースに送信し、アンサーセット、メッセージのいずれかを要求元に送信することで、データインターフェースとデータソース間の通信を管理します。

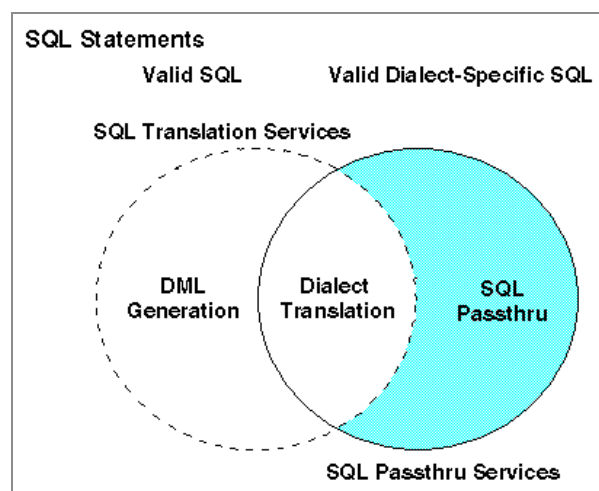
アダプタは、これを次のように処理します。

1. リクエストを適合する DML に変換する。
2. 標準接続コールを使用して、ターゲットデータソースに接続する。その後、アダプタはデータソースにリクエストを送信します。
3. データソースがリクエストを処理する。

4. 以降の処理がクライアントで行えるよう、結果またはエラーの状態がクライアントアプリケーションに返信される。

SQL リクエストの処理

クライアントアプリケーションから受け取る SQL リクエストに応じて、異なる動作を実行するようにサーバを構成することができます。図に示すように、サーバは以下の手順で SQL リクエストのデータを処理します。



- **ダイレクトパススルー** ダイレクトパススルーを有効にした場合、サーバは指定した RDBMS に SQL リクエストを直接渡します。ターゲット RDBMS (データベースエンジン) の名前はサーバプロファイルに指定しますが、クライアントアプリケーションが指定することもあります。クライアントアプリケーションを通じて Full Function Server または HUB Server を呼び出した場合、一時的にダイレクトパススルーモードで処理することが可能です。ユーザは、ダイレクトパススルーの有効/無効の切り替えに責任を負います。
- **SQL 処理** サーバプロファイルにデータベースエンジンを設定していない、またはクライアントアプリケーションにより設定されている場合、ダイレクトパススルーは有効となりません。代わりに HUB Server または Full Function Server のデフォルトの動作が実行され、受け取った SQL リクエストが有効かどうかを検査します。次にそれらが SQL リクエストを処理できるかどうかが判別されます。
 - リクエストが一定の要件を満たす場合、サーバは RDBMS にリクエストを直接渡します。これは「自動パススルー」と呼ばれます。
 - リクエストの構文が RDBMS の構文と一致しない場合、サーバはリクエストを内部 DML に変換してアダプタに渡します。アダプタは固有の DML を生成して、リ

クエストを RDBMS に渡します。これは「SQL 変換」と呼ばれます。

ibi WebFOCUS リクエストの処理

サーバが受け取った WebFOCUS リクエストはアダプタに渡されます。アダプタはリクエストを分析して DBMS 固有の DML を生成します。リクエストのそれらの部分は DML と同じです。WebFOCUS は DML より強力な言語であるため、リクエストの一部に同じ部分は含まれません。アダプタは生成された DML を DBMS に渡して処理します。DBMS からアンサーセットが返されると、WebFOCUS はアダプタが DBMS に渡すことができなかったリクエストのそれらの部分を処理します。

このマニュアルには、SQL 構文ではサポートされないが、WebFOCUS 構文ではサポートされる便利な機能の説明が含まれています。

WebFOCUS 言語についての詳細は、以下のマニュアルを参照してください。

- 『ibi™ WebFOCUS® Language リファレンス』
- 『ibi™ WebFOCUS® メタデータリファレンス』
- 『ibi™ WebFOCUS® アプリケーション作成ガイド』

リレーショナルおよび非リレーショナルアダプタ

アダプタは、リレーショナルおよび非リレーショナルの両方のデータソースのアンサーセットを検索することができます。リレーショナルおよび非リレーショナルデータ構造のアーキテクチャはさまざまであるため、リレーショナルおよび非リレーショナルアダプタはそれらの相違に対応します。たとえば、リレーショナルアダプタは、テーブルの行と列にデータが格納されたデータソースを処理するよう設計されています。非リレーショナルアダプタは、階層データソースやネットワークデータソース、シーケンシャルファイルやインデックスファイルシステムなど、それぞれ種類の異なるデータソースのアーキテクチャに対応するよう設計されています。

以下は、リレーショナル/非リレーショナルアダプタの主な機能のリストです。

機能	リレーショナルアダプタ	非リレーショナルアダプタ
マスターファイルの一時項目 (DEFINE)	可能	可能
アクセスコントロール	可能	可能
トランザクション管理コマンド	可能	不可

リレーショナルアダプタおよび非リレーショナルアダプタを使用すると、次のことが可能になります。

- データソースが、データの結合、ソート、集計に必要な処理を実行することができます。このためデータソースとサーバ間の通信量は減少し、ユーザの応答時間は向上します。アダプタは、クエリを可能な限り最適化しようと試みます。
- DML ステートメントを通じてデータソースと通信することができます。トレース機能を使用して、これらの DML ステートメントを参照することができます。これらのトレースは、プロシジャのデバッグやアダプタのパフォーマンス分析に便利です。トレースでは、以下の点に注意してください。
 - トレースはすべてのリレーショナルアダプタに共通します。
 - 非リレーショナルアダプタでは、データ構造および DML 呼び出しの相違に対応するためそれぞれ異なります。
- 一時項目 (DEFINE) と DBA によるアクセスコントロールの両方がサポートされます。
- リレーショナルアダプタには、データベース管理者が接続の開始と終了を制御したり、コミットまたはロールバックのトランザクションを実行するタイミングを選択したりするための、さまざまな COMMIT コマンド、接続コマンド、スレッド制御コマンドが用意されています。このレベルのトランザクション制御は、非リレーショナルアダプタではサポートされません。Full Function Server モードでは、サーバのローカルのすべてのリレーショナルデータソースに対して COMMIT/ROLLBACK が適用されます。HUB Server が有効な場合、COMMIT はリモートデータサーバに対しても実行されます。All PREPARE ハンドルは削除されます (これはブロードキャスト COMMIT です)。

このマニュアルには、これらの類似点と相違点の詳細が記載されています。

- 全般情報として、すべてのアダプタに共通する情報のほか、すべてのリレーショナルアダプタのみに共通する情報、すべての非リレーショナルアダプタのみに共通する情報があります。
- カスタム情報として、特定のアダプタにのみ適用される情報があります。

データ管理

データを管理する場合、サーバおよび通信構成ファイルを変更する必要があります。最初の手順は、データをどこにどのように定義するか、およびサーバとアダプタが処理フローの管理に果たす役割を理解することです。

データソースの定義

テーブルまたはビューにアクセスするには、最初にマスターファイルとそれに関連するアクセスファイルの 2 つのファイルを使用して、それらのテーブルまたはビューを定義しなければなりません。

マスターファイルとアクセスファイルは、テーブル全体またはテーブルの一部を定義することができます。また、複数のマスターファイルとアクセスファイルを組み合わせることによって、同じテーブルの異なるカラムのサブセットを定義したり、単一のマスターファイルとアクセスファイルを組み合わせることによって、複数のテーブルを定義することができます。

注意：この説明の「テーブル」という用語は、基本テーブルとデータソースのビューの両方を指します。マスターファイルでは、カンマ区切りフォーマットのキーワードを使用してデータソーステーブルのカラムを定義します。アクセスファイルには、データソーステーブルを定義する他のパラメータが含まれています。クエリの実行および非 SQL データソースにアクセスする DML の作成に、両方のファイルが必要となるアダプタもあります。

リクエストの処理

リクエストを処理する場合、サーバとアダプタ間で制御がやり取りされます。処理中に選択した情報が、以下のようにマスターファイルおよびアクセスファイルから読み込まれます。

サーバによるリクエストの処理手順は、以下のとおりです。

1. テーブルを識別するためリクエストが解析されます。
2. テーブルのマスターファイルが読み込まれます。
3. マスターファイルの SUFFIX の値がチェックされます (SUFFIX はデータソースのタイプを示します)。
4. 適切なアダプタに制御が渡されます。

アダプタの処理は、以下のとおりです。

1. 対応するアクセスファイルの場所を特定します。
2. マスターファイルおよびアクセスファイルの情報を使用して、リクエストの完成に必要な DML ステートメントを生成します (必要な場合)。
3. データソースに DML ステートメントを渡します。
4. データソースにより生成されたアンサーセットを取得します。
5. サーバに制御を返します。

リクエストの要件に応じて、返されたデータに対して他の処理を実行します。

マスターファイル

マスターファイルは論理データソースを定義します。論理データソースは、同一タイプの 1 つまたは複数の物理データソースで構成することができます。各セグメントが物理データソースとなります。

マスターファイルには、以下の 3 つのタイプの宣言が含まれています。

宣言のタイプ	説明
ファイル	ファイルを指定して、データソースのタイプを定義します。
セグメント	テーブル、ファイル、ビュー、セグメントを指定します。
フィールド	テーブルのカラム、ビュー、ファイルのフィールドを定義します。

次のガイドラインが適用されます。

- 宣言には、属性と値の組み合わせを記述します。複数の組み合わせを列記するには、それぞれをカンマ (,) で区切ります。
- 各宣言は、それぞれ別の行から開始する必要があります。1 つの宣言を複数行にわたって記述することができますが、属性と値の組み合わせは一对として同一行に記述する必要があります。

- ファイル、セグメント、フィールド、エイリアスの名前に、予約語を使用することはできません。予約語を指定すると、構文エラーが発生します。

マスターファイルでのファイル宣言の指定

マスターファイルは、少なくとも 2 つの属性を含むファイル宣言から始まります。

FILENAME (FILE)

マスターファイルを指定します。

SUFFIX

リクエストの解釈に必要なアダプタを指定します。

ファイル宣言の構文は以下のとおりです。

```
FILE[NAME]=file, SUFFIX=suffix [,,$]
```

説明

file

マスターファイルのファイル名です。このファイル名は文字で開始する必要があります。また、テーブルまたはビューの内容を表す名前にすることをお勧めします。実際のファイルには拡張子 .mas を付ける必要がありますが、この属性の値には拡張子を含めないでください。拡張子.mas を持たないファイルには、8 バイト以内の文字でファイル名を指定することができます。

suffix

リクエストの解釈に必要なアダプタを指定します。たとえば、Oracle アダプタには値として SQLORA を指定します。

マスターファイルでのセグメント宣言の指定

マスターファイルで定義するテーブルごとにセグメント宣言を指定する必要があります。セグメント宣言は、少なくとも 2 つの属性で構成されます。

SEGNAME

単一のテーブルを指定します。

SEGTYPE

行の物理記憶域およびカラム値の一意性を指定します。

セグメント宣言の構文は以下のとおりです。

```
SEGNAME=segname, SEGTYPE=S0 [, $]
```

説明

segname

実際のテーブル名へのリンクとして機能するセグメント名です。FILENAME に指定した名前、実際のテーブル名、任意の名前と同じです。8 バイト以内の文字を指定することができます。

マスターファイルの SEGNAME 値は、アクセスファイルで指定された SEGNAME 値に一致させる必要があります。アクセスファイルのセグメント宣言の TABLENAME には、テーブルの完全修飾名が含まれています。

S0

RDBMS は行の物理記憶域およびカラム値の一意性に責任を有することを示します (ユニークインデックスや制限がある場合)。この値は、常に S0 (S ゼロ) です。

マスターファイルでのフィールド宣言の指定

テーブルの各行は、1 つまたは複数のカラムで構成されます。これらのカラムは、以下のプライマリフィールド属性を持つフィールドとしてマスターファイルに定義されます。

FIELDNAME

フィールド名を指定します。

ALIAS

完全なカラム名を指定します。

USAGE

レポートのフィールドの表示方法を指定します。

ACTUAL

フィールドのデータタイプと長さ (バイト単位) を指定します。

MISSING

フィールドが NULL データを許容するかどうかを指定します。

既存のテーブルまたは定義するビューのシステムカタログテーブル ALL_TAB_COLUMNS を使用して、これらの属性の値を取得することができます。

フィールド定義の構文は、以下のとおりです。

```
FIELD[NAME]=fieldname, [ALIAS=]sqlcolumn, [USAGE=]display_format,  
[ACTUAL=]storage_format [,MISSING={ON|OFF}], $
```

説明

fieldname

フィールド名です。この値はマスターファイル内で一意でなければなりません。この名前には、最大で 48 バイトの、文字、数字、アンダースコア () を組み合わせて使用することができます。名前の先頭は文字でなければなりません。特殊文字や埋め込みブランクを使用することはお勧めしません。マスターファイルでのフィールド宣言の順序は、キーカラムの指定に関連して重要です。詳細は、[主キー](#)を参照してください。

ヒント：名前は、レポート、クライアントアプリケーション、または EDADESCRIBE のデフォルトのカラムタイトルとして表示されるため、データの内容を表すような名前を指定します。

マスターファイルのテーブルのすべてのカラムを定義する必要はありません。

sqlcolumn

カラムの完全修飾名です (アダプタは、この名前を使用して SQL ステートメントを生成します)。この値は、識別子の命名規則に準拠する必要があります。名前は文字から始まり、文字、数字、アンダースコア (_) の組み合わせが後に続きます。埋め込みブランクを使用することはできません。

display_format

表示フォーマットです。この値には、フィールドのタイプと長さを指定する必要があります。必要に応じて、編集オプションを含めることもできます。

表示フォーマットのデータタイプは、ACTUAL フォーマットのデータタイプに一致させる必要があります。たとえば、文字の USAGE データタイプのフィールドには、文字の ACTUAL データタイプを指定する必要があります。

パック 10 進数または浮動小数点数データタイプのフィールドには、正しい小数点以下の桁数 (s) と精度 (p) を指定する必要があります。小数点以下の桁数は小数点の右側の桁数、精度はフィールドの合計長です。

サーバでは、表示フィールドの合計長に小数点およびマイナス記号が含まれます。SQL では、フィールドの合計長に小数点およびマイナス記号は含まれません。たとえば、DECIMAL(5,2) のカラムの USAGE 属性は P7.2 となり、小数点および有効なマイナス記号を含めることができます。

storage_format

データタイプの記憶域フォーマットと長さ (バイト単位) です。

ON

ミッシングデータの NODATA パラメータに指定された文字を表示します。関連する情報については、[MISSING 属性](#)を参照してください。

OFF

値を持たないフィールドにブランクまたは 0 (ゼロ) を表示します。これがデフォルト値です。詳細は、[MISSING 属性](#)を参照してください。

MISSING 属性

テーブルでは、NULL 値はミッシング値または不明な値を表します。ブランクおよび 0 (ゼロ) とは異なります。たとえば NULL 値を許容するカラムは、そのすべての行に値を含める必要がない場合に使用されます (給与データが格納されているテーブルの昇給率など)。

- デフォルトの NODATA 文字はピリオドです。
- テーブル内で NULL データを許容するカラムには、そのテーブル定義に NULL 句を含める必要はありません。デフォルト設定で NULL データが許容されるためです。
- そのテーブルのマスターファイルでは、NULL データを許容するカラムの MISSING 属性を ON に設定する必要があります。この属性のデフォルト値は OFF です。値 OFF はテーブル定義の NOT NULL 属性に対応しています。

NULL データを許容するが、マスターファイルの対応するフィールドの MISSING 属性が OFF のカラムでは、NULL データの値は 0 (ゼロ) またはブランクとして表示されます。

アクセスファイル

マスターファイルには、それぞれ対応するアクセスファイルがあります。アクセスファイルの名前は、マスターファイルの名前に一致させる必要があります。ただし、アクセスファイルの拡張子は、.mas ではなく .acx になります。

アクセスファイルは、マスターファイルのセグメントと、そのセグメントが定義されているテーブルを関連付けることにより、サーバとデータソース間のリンクとして機能します。アクセスファイルはテーブルおよび主キー (存在する場合) を識別します。また、データの論理ソート順序を指定したり、テーブルの記憶域を識別したりする役割もあります。

アクセスファイルでのセグメント宣言の指定

アクセスファイルのセグメント宣言は、マスターファイルの単一のセグメントと実際のテーブルまたはビュー間のリンクを確立します。セグメント宣言を構成する属性は、以下のとおりです。

SEGNAME

単一のテーブルを指定します。

TABLENAME

テーブル名またはビュー名です。テーブル名と同様、所有者 ID も含まれています。

KEYS

主キーを構成するカラムの数です。

KEYORDER

主キーに基づいてデータをソートする際の論理的ソート順です。

アクセスファイルのセグメント宣言の構文は、以下のとおりです。

```
SEGNAME=segname, TABLENAME=owner.tablename databaselink  
[,KEYS={n|0}] [,KEYORDER={LOW|HIGH}] , $
```

説明**segname**

マスターファイルの SEGNAME 値と同一の値です。

owner

デフォルト設定では、ユーザ ID です。

tablename

テーブルまたはビューの名前です。

databaselink

現在接続されているデータベースサーバで使用する DATABASE LINK 名です。

n

主キーを構成するカラムの数です。0 から 16 の値を指定することができます。デフォルト値は 0 (ゼロ) です。詳細は、[主キー](#)を参照してください。

LOW

主キーの、昇順の論理ソート順序を示します。これがデフォルト値です。

HIGH

主キーの、降順の論理ソート順序を示します。

主キー

テーブルの主キーは、テーブルの各行を一意に識別する値を持つフィールド、または複数のフィールドの組み合わせで構成されます。たとえば、社員テーブルではすべての社員に一意的社員番号が割り当てられます。各社員はテーブルの単一の行によって表され、その識別番号によって一意に識別されます。

主キー定義の一部はマスターファイルで定義し、一部はアクセスファイルで定義する必要があります。

- マスターファイルでのフィールド宣言の順序は、キーフィールドを指定する際に重要になります。マスターファイルで主キーを定義するには、主キーを構成するフィールドをセグメント宣言の直後に記述します。残りのフィールドについては、任意の順序で指定することができます。アクセスファイルでは、KEYS 属性はプライマリキーの定義の終了を示します。
- アダプタは、アクセスファイルの KEYS 属性に指定されたフィールド数 (n)、およびマスターファイルに記述された最初の n 個のフィールドを使用して、主キーを識別します。

通常、主キーは、SQL 言語でユニークインデックスを作成することでサポートされます。これにより、重複するキー値の挿入が回避されます。アダプタ自体では、主キーを構成するフィールドにインデックスは必要ありません。ただし、データ整合性とパフォーマンスの両方の観点から、ユニークインデックスの使用が好ましい場合があります。

一時項目 (DEFINE) の作成

これらのタスクを実行するには、DEFINE コマンドを使用します。

DEFINE コマンドによる一時項目 (DEFINE) の作成

```
DEFINE fieldname/format [WITH fieldname]=expression ;$
```

説明

fieldname

一時項目 (DEFINE) の名です。1 から 48 バイトの名前を指定します。フィールド名を修飾することはできません。

format

フィールドの表示フォーマットを指定して、USAGE フォーマットの規則に従います。このオペランドはオプションです。指定しない場合のデフォルト値は D12.2 です。

WITH fieldname

式が定数の場合、必ず指定します。DEFINE の関連セグメントの実フィールドを選択することができます。

expression

数式または論理ステートメントを記述することができます。式には、定数、データベースフィールド、一時項目 (DEFINE) を含めることができます。式は、セミicolonとドル記号 (;\$) で終了する必要があります。

DEFINE ステートメントは、セグメントのすべてのフィールド定義の後に挿入します。一時項目 (DEFINE) で DESCRIPTION または TITLE 属性を使用する場合、これらの属性は別の行に記述する必要があります。

マスターファイルでの一時項目 (DEFINE) の定義

以下の例では、BODY セグメントの最後に PROFIT という一時項目 (DEFINE) が定義されています。

```
SEGMENT=BODY, SEGTYPE=S0 , PARENT=CARREC,$
  FIELDNAME=BODYTYPE           ,ALIAS=BODYTYPE           ,A12,A12,$
  FIELDNAME=DEALER_COST        ,ALIAS=DEALER_COST        ,D8, D8 , $
  FIELDNAME=RETAIL_COST        ,ALIAS=RETAIL_COST        ,D8, D8 , $
  DEFINE PROFIT/D8 = RETAIL_COST - DEALER_COST
  ;DESC=NET_COST, TITLE='NET,COST' , $
```

この DEFINE ステートメントにより、レポートのフィールド名として PROFIT を使用することができます。PROFIT は、RETAIL_COST から DEALER_COST を減算した値と同一の値が格納されたフィールドとして処理されます。

注意

- 一時項目 (DEFINE) を計算するにはデータソース全体を読み取る必要があるため、一時項目 (DEFINE) に選別条件を適用すると、余分なオーバーヘッドが発生する場合があります。
- リレーショナルデータソースおよびリモートデータソースのマスターファイルの一時項目 (DEFINE) がクエリで参照されると、自動パススルーが無効になります。

世紀をまたがる日付

既存のビジネスアプリケーションの多くでは、年の表示に 4 桁ではなく 2 桁を使用しています。年の値として 00 を受け取った場合、最初の 2 桁を 20 世紀の 19 と見なして 1900 年と解釈してしまいます。問題となる世紀を特定する仕組みを導入しない限り、既存のアプリケーションの日付に基づく計算がエラーとなる深刻なリスクが存在します。これは抵当権、保険証、祝祭日、債券、在庫補充、契約書、リース、年金、受取勘定、顧客レコードなど、ほぼすべてのタイプのアプリケーションに影響を及ぼします。

世紀をまたがる日付の機能によって、アプリケーションのファイルおよびフィールドレベルでこの問題を解決することが可能となります。グローバルの設定は変更せずに、ファイルレベルの設定を変更することができます。

この機能は、以下のように有効にすることができます。

- SET コマンドを使用して有効にする。
- マスターファイルのファイルレベルで有効にする。
- マスターファイルのフィールドレベルで有効にする。

世紀をまたがる日付の SET コマンド

年の先頭の 2 桁が指定されていない場合、サーバは世紀を解釈するための SET コマンドを実行します。

```
SET DEFCEINT  
SET YRTHRESH
```

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て検証されます。

世紀をまたがる日付の実装

DEFCENT の構文は、以下のとおりです。

```
SET DEFCENT=nn
```

説明

nn

特定の値を指定しない場合は、19 に設定されます。

YRTHRESH の構文は、以下のとおりです。

```
SET YRTHRESH=nn
```

説明

nn

特定の値を指定しない場合は、0 (ゼロ) に設定されます。

DEFCENT と YRTHRESH を組み合わせることにより、100 年時間枠の基準年が設定されます。この時間枠には任意の 2 桁の年を入力することができ、入力した値に基づいて先頭の 2 桁が設定されます。この時間枠で指定した以外の年については、ユーザのコーディングで対応する必要があります。

2 つのコマンドのデフォルト値は SET DEFCENT=19、SET YRTHRESH=00 です。年のしきい値を指定した場合、その値以上の年は DEFCENT によって割り当てられたと見なされます。その値未満の年は DEFCENT に 1 を加算した値となります。

2 桁の年を解釈する際、DEFCENT と YRTHRESH がどのように適用されるかを考えてみます。

```
SET DEFCENT=19, SET YRTHRESH=80
```

これらのコマンドは 1980 年から 2079 年までの範囲を定義します。2 桁の年のフィールドの値が 99 であれば、1999 年と解釈されます。フィールドの値が 79 であれば 2079 年と解釈されます。フィールドの値が 00 であれば 2000 年と解釈されます。

マスターファイルの構文

SET コマンドを使用せずに、マスターファイルのファイルレベルまたはフィールドレベルに設定を追加することができます。

世紀をまたがる日付設定のファイルレベルでの追加

FDEFCENT の構文は、以下のとおりです。

```
{FDEFCENT|FDFC}=nn
```

説明

nn

特定の値を指定しない場合は、19 に設定されます。

FYRTHRESH の構文は、以下のとおりです。

```
{FYRTHRESH|FYRT}=nn
```

説明

nn

特定の値を指定しない場合は、0 (ゼロ) に設定されます。

世紀をまたがる日付設定のフィールドレベルでの追加

フィールドレベルには、DEFCENT および YRTHRESH を追加することができます。DEFCENT の構文は、以下のとおりです。

```
{DEFCENT|DFC}=nn
```

説明

nn

特定の値を指定しない場合は、19 に設定されます。

YRTHRESH の構文は、以下のとおりです。

```
{YRTHRESH|YRT}=nn
```

説明

nn

特定の値を指定しない場合は、0 (ゼロ) に設定されます。

世紀をまたがる日付の DEFINE コマンドによる追加

```
DEFINE FILE EMPLOYEE
  fld/fmt [{DEFCENT|DFC} nn {YRTHRESH|YRT} nn] [MISSING...]=expression;
END
```

DFC および YRT の構文は、フィールドのフォーマット情報に準拠する必要があります。

世紀をまたがる日付の解釈

以下は、マスターファイルのファイルレベルおよびフィールドレベルの両方で、世紀の解釈方法を実装する例です。

```
FILENAME=EMPLOYEE, SUFFIX=FOC, FDEFCENT=20, FYRTHRESH=66,$
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, DEFCENT=19,
YRTHRESH=75,$
```

以下は、DEFINE コマンドによる 2 桁の年のフィールドの変換を示す例です。

```
DEFINE FILE EMPLOYEE
ESHIRE_DATE/YYMD = HIRE_DATE; (The format of HIRE_DATE is I6YM.)
ESHIRE DFC 19 YRT 80 = HIRE_DATE;
END
```

シノニムの管理オプション

WebFOCUS Reporting Server ブラウザインターフェースまたは ibi™ Data Migrator デスクトップインターフェースのいずれかの [アプリケーション] ナビゲーションウィンドウでシノニム名を右クリックすると、次のオプションにアクセスすることができます。

オプション	説明
開く	表示および編集のために、グラフィカルインターフェースを使用してマスターファイルを開きます。また、アクセスファイルを使用している場合にも使用可能になります。
テキストとして編集	このオプションを使用すると、マスターファイルのシノニムを表示したり、手動で編集したりすることができます。 注意： シノニムを更新する場合は、マスターファイルを手動で編集するのではなく、[開く] オプションで提供されるグラフィカルインターフェースを使用することをお勧めします。
アクセスファイルをテキストとして編集	アクセスファイルのシノニムを表示したり、手動で編集したりできます。 注意： このオプションは、シノニムの一部としてアクセスファイルを作成した場合にのみ使用することができます。

オプション	説明
	す。
サンプルデータ	関連するデータソースから最大で 20 行を取得します。
データプロファイリング	データプロファイリングには、シノニムの各フィールドのデータ特性が表示されます。 文字フィールドでは、種類の合計、件数の合計、長さの最大値、長さの最小値、長さの平均値、NULL 値の個数が表示されます。 数値フィールドでは、種類の合計、件数の合計、最大値、最小値、平均値、NULL 値の個数が表示されます。
シノニムのリフレッシュ (適切な場合)	シノニムを再作成します。このオプションは、基になるオブジェクトが変更された場合に使用します。
データ管理	適用できる場合、以下のオプションも続けて指定します。 DBMS テーブルの再作成 データソーステーブルを再作成します。選択したシノニムの再作成を確認するメッセージが表示されます。なお、テーブルは一度削除されてから再作成されます。そのため、処理中はデータが一時的に失われる場合があります。 全データの削除 既存のデータをすべて削除します。データを削除する前に、選択したデータの削除を確認するメッセージが表示されます。 テーブルの削除 テーブルを DBMS から削除します。 サンプルデータの挿入 指定した数のサンプルレコードを挿入します。この場合、すべてのフィールドにカウンタ値が表示されます。 データの表示/変更 データソースのデータを表示するウィンドウを開き、ボタンをクリックして値の挿入、値のフィルタ、データソースの再ロード、ビューのカスタ

オプション	説明
	マイズを行えます。 再編成 元のデータを保持した状態でデータソーステーブルを再作成します。 注意：このオプションは WebFOCUS Reporting Server ブラウザインターフェースでは使用できません。
インパクト分析	選択したシノニムの格納先および使用先を表示するレポートを生成します。このレポートには、シノニムの各インスタンスへのリンクが表示されます。インパクト分析レポートを生成すると、実際に変更を加えることなく、変更により影響を受けるコンポーネントを把握することができます。インパクト分析レポートについての詳細は、『ibi™ WebFOCUS® サーバ管理者ガイド』を参照してください。
従属関係の分析	選択したシノニムに関する情報と、そのシノニムで参照されている他のシノニムおよびオブジェクトに関する情報を表示するレポートを生成します。
コピー	シノニムをクリップボードにコピーします。
削除	シノニムを削除します。シノニムを削除する前に、選択したシノニムの削除を確認するメッセージが表示されます。
切り取り	シノニムを削除して、クリップボードに配置します。
権限	サーバのセキュリティサブジェクト (ロール、グループ、ユーザ) のリスト、および各セキュリティサブジェクトがこのシノニムに対して所有する権限を表示します。
プロパティ	シノニムのプロパティ (物理パス、最終更新日、説明、権限など) を表示します。

データタイプサポートレポート

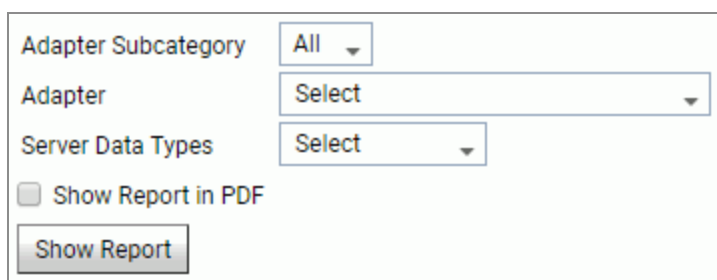
WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

データタイプレポートを表示するには

データタイプレポートを表示するには、次の手順を実行します。

手順

1. メインメニューから [データの取得] を選択します。ウィンドウ下部の [情報と機能診断] ボタンをクリックします。
2. サイドメニューから、[データタイプ] ボタンをクリックします。
下図のように、データタイプレポートのフィルタページが開きます。



3. [アダプタサブカテゴリ] ドロップダウンリストから、レポートに表示するアダプタのサブカテゴリを選択します。
4. [アダプタ] リストボックスから、レポートに表示するアダプタを選択します。
5. [サーバデータタイプ] リストボックスから、レポートに表示するサーバデータタイプを選択します。
6. [レポートの表示] をクリックします。

結果

下図のように、データタイプレポートが表示されます。

Data Types						
Data Type Category	Vendor Data Types	Type Range	Server USAGE	Server ACTUAL	Server DDL	Remarks
DB2, all platforms						
Date-Time	DATE		YYMD	DATE		
	TIME		HHIS	HHIS		
	TIMESTAMP		HYYMDm	HYYMDm		
LOB and Other	CLOB		TX50	TX		
	BLOB		BLOB	BLOB		
	DATALINK		N/A	N/A		
	XML		TX50L	TX		
Numeric	SMALLINT		I6	I2		
	INTEGER		I11	I4		
	BIGINT		P20	P10		
	DECIMAL(p,s)/NUMERIC(p,s)	p=1..31,s=0	Pn	Pk		n=p+1,k=(p/2)+1
	DECIMAL(p,s)/NUMERIC(p,s)	p=1..31,s>0	Pn.m	Pk		n=p+2,m=min(s,31),k=(p/2)+1
	REAL		F9.2	F4		
	DOUBLE		D20.2	D8		
DB2, only for IBM i						

注意：[レポートを PDF で表示] のチェックをオンにすることで、レポートを PDF 表示にすることもできます。

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアードプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

デフォルトの精度と小数点以下の桁数の書き変更

```
ENGINE ADAPTER_ID SET CONVERSION RESET
ENGINE ADAPTER_ID SET CONVERSION format RESET
ENGINE ADAPTER_ID SET CONVERSION format [PRECISION precision [scale]]
ENGINE ADAPTER_ID SET CONVERSION format [PRECISION MAX]
```

説明

ADAPTER_ID

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

RESET

指定している精度と小数点以下の桁数の値をアダプタのデフォルト設定に戻します。SET CONVERSION コマンドの直後に RESET を指定した場合、データタイプはすべてデフォルト設定に戻ります。特定のデータタイプの後に RESET を指定した場合、そのデータタイプのカラムのみがデフォルト設定に戻ります。

format

データソースでサポートされる任意の有効なフォーマットです。利用可能な値は、次のとおりです。

INTEGER - コマンドは INTEGER カラムにのみ適用されます。

DECIMAL - コマンドは DECIMAL カラムにのみ適用されます。

REAL - コマンドは単精度浮動小数点数のカラムにのみ適用されます。適用されるデータベースは、Db2、Microsoft SQL Server、および Sybase のみです。

FLOAT - コマンドは倍精度浮動小数点数のカラムにのみ適用されます。

precision

精度です。1 より大きく、データタイプの許容最大値以下の値を指定する必要があります (MAX の説明を参照)。

scale

小数点以下の桁数です。この設定は、DECIMAL、FLOAT、REAL データタイプで有効です。値を指定しない場合、現在の小数点以下の桁数の設定が引き続き有効になります。デフォルトの小数点以下の桁数は 2 です。

小数点以下の桁数が必要ない場合は、小数点以下の桁数として 0 (ゼロ) を設定する必要があります。

MAX

精度の値を、指定したデータタイプの許容最大値に設定します。

データタイプ	最大精度
INTEGER	11
DECIMAL	18
REAL	9
FLOAT	20

注意：CONVERSION コマンドがプロファイルで有効な場合に、CREATE SYNONYM コマンドを発行すると、CONVERSION コマンドにより設定された小数点以下の桁数と長さがマスターファイルに反映されます。

ただし、SELECT ステートメントを発行した場合、アンサーセットの説明ではマスターファイル内の情報が使用されません。アンサーセットの説明に使用される長さや小数点以下の桁数は、CONVERSION コマンドが有効かどうかによって決まります。

CONVERSION コマンドが有効な場合、アンサーセットの説明では CONVERSION コマンドによって設定された長さや小数点以下の桁数が使用されます。

CONVERSION コマンドが無効な場合、アンサーセットの説明ではデータの実際の長さや小数点以下の桁数が使用されます。

精度および小数点以下の桁数の属性の設定

以下の例では、すべての INTEGER と SMALLINT のフィールドの精度属性を 7 に設定する方法を示しています。

```
ENGINE ADAPTER_ID SET CONVERSION INTEGER PRECISION 7
```

以下の例では、すべての DOUBLE PRECISION フィールドの精度属性を 14、小数点以下の桁数属性を 3 に設定する方法を示しています。

```
ENGINE ADAPTER_ID SET CONVERSION FLOAT PRECISION 14 3
```

以下の例では、すべての INTEGER と SMALLINT のフィールドの精度属性をデフォルト値に設定する方法を示しています。

```
ENGINE ADAPTER_ID SET CONVERSION INTEGER RESET
```

以下の例では、すべてのフィールドの精度および小数点以下の桁数をデフォルト値に設定する方法を示しています。

```
ENGINE ADAPTER_ID SET CONVERSION RESET
```

SQLENGINE SET によるメタデータサービス

サーバがグローバルプロファイルの SET SQLENGINE コマンドを使用して、単一のリレーショナルデータベース管理システム (RDBMS) にアクセスする場合、RDBMS の固有のカatalog に対してサーバへのメタデータ呼び出しが実行されます。

アプリケーションによるメタデータのアクセス

固有のカatalogを直接照会するアプリケーションは、以下のメタデータプロシジャを使用します。

- ODBC アプリケーションでは、ODBC SQL コール。
- API アプリケーションでは、ODBCxxxx コールの EDARPC。

以下の表は、ODBC コールと API コールの関係を示します。

ODBC コール	API コール
SQLTables	ODBCTABL
SQLColumns	ODBCCOLS
SQLPrimaryKeys	ODBCPKEY
SQLStatistics	ODBCSTAT

ODBC コール	API コール
SQLProcedures	ODBCPROC
SQLProcedureColumns	ODBCPRCC
SQLSpecialColumns	ODBCSCOL
SQLColumnPrivileges	ODBCCPRV
SQLForeignKeys	ODBCFKEY
SQLTablePrivileges	ODBCTPRV

以下の 2 つのコマンドを使用して、テーブルおよびカラムメタデータへのクエリを制御または上書き変更することができます。

```
SQL sqlengine SET ODBCCOLSSORT
```

```
SQL sqlengine SET ODBCTABL
```

これらのコマンドは、サポートされる任意のサーバプロファイルに記述することができます。

カラム情報の取得 (Db2 のみ)

クライアントアプリケーションから SQLColumns または ODBCCOLS メタデータコールを発行した場合、デフォルト設定では、サーバは Db2 のカラムを *colno* の順序でソートします。

カラムデータのソート順序の指定

```
SET ODBCCOLSSORT {ON|OFF}
```

説明

ON

サーバは、カラムデータを colno の順序でソートします。これがデフォルト値です。

OFF

カラムデータは、未ソートの順序で返されます。

ユーザ定義のメタデータの取得

クライアントアプリケーションが ODBC または API メタデータでのクエリを実行する場合、サーバは固有の RDBMS カタログに対してデフォルトの SQL を実行する内部プロシジャを実行します。デフォルトの SQL の代わりに、独自の SQL を実行することが可能です。このタイプの上書き処理は、メタデータを処理する任意の内部サーバルーチンに対して指定することができます。

ユーザ定義のメタデータの指定

```
SQL sqlengine SET ODBCxxxx procname
```

説明

ODBCxxxx

内部サーバルーチンの名前です。利用可能な値は、次のとおりです。ODBCTABL、ODBCCOLS、ODBCPKEY、ODBCSTAT、ODBCPROC、ODBCPRCC、ODBCSCOL、ODBCCPRV、ODBCFKEY、ODBCTPRV。

procname

サーバがメタデータへのクエリを受け取った際に実行するプロシジャです。このプロシジャは、サーバ JCL の FOCEXEC ddname の割り当てを通じて使用可能でなければなりません。

注意：この上書きプロシジャを MVS 下のサーバで使用する場合、ユーザごとに約 600K の記憶域が追加されます。

上書きプロシジャを記述する場合、選択リスト (アンサーセットのレイアウト) の保守に注意する必要があります。選択リストは、SQLTables 呼び出しによって返される ODBC の指定に合致しなければなりません。SQLTables 指定レイアウトについては、『ODBC 2.0

Programmer's Reference and SDK Guide』を参照してください。この上書きプロシジャを使用する場合、メタデータ呼び出しとともに送信されるパラメータを正確に解析する必要があります。

上書きプロシジャは、以下のフォーマットでなければなりません。

1. メタデータ呼び出しパラメータを解析するダイアログマネージャのコード。

2.

```
SQL sqlengine SELECT code;
```

3.

```
TABLE
ON TABLE PCHOLD FORMAT ALPHA
END
```

1 と 2 はユーザによる記述です。3 はプロシジャの末尾に常に含める必要があります。

テーブルのリストを返す

以下のサンプルは *qualif.EDARPC.DATA(DB2ODBC1)* に収録されており、接続ユーザに INSERT、UPDATE、DELETE、SELECT の権限を付与するテーブルのリストを返します。ODBCTABL の Db2 上書きプロシジャを記述するためのサンプルです。ODBCTABL 呼び出しのアンサーセットを返す SQL を記述する方法の 1 つです。SELECT リストを保守している限り、関連するクエリを記述することができます。

サーバプロファイルでは、SQL DB2 SET ODBCTABL DB2ODBC1 を実行して、サーバ上で以下の RPC リクエストを実行します。

```
ODBCTABL ,<NULL>,,,,0,0,*
```

以下の例は、上記の ODBCTABL 呼び出しと一致しています。

```
-*
-* Dialogue Manager code to parse the ODBCTABL parameter list
-*
-DEFAULTS 1=' ' ,2='%',3='%'
-IF &2 NE '<NULL>' THEN GOTO LAB1;
-SET &2 = '%';
-LAB1
-IF &3 NE ' ' THEN GOTO LAB2;
-SET &3 = '%';
-LAB2
```

```

-＊
-＊ SQL SELECT code
-＊
SQL DB2
SELECT ' ',T2.CREATOR,T2.NAME,'TABLE',' '
FROM SYSIBM.SYSTABAUTH T1,SYSIBM.SYSTABLES T2
WHERE T1.GRANTEE = USER
AND T1.TTNAME LIKE '&2'
AND T1.TCREATOR LIKE '&3'
AND T2.TYPE = 'T'
AND (T1.DELETEAUTH IN ('G','Y')
OR T1.INSERTAUTH IN ('G','Y')
OR T1.SELECTAUTH IN ('G','Y')
OR T1.UPDATEAUTH IN ('G','Y'))
AND T1.TTNAME = T2.NAME
AND T1.TCREATOR = T2.CREATOR
UNION

```

```

SELECT ' ',T2.CREATOR,T2.NAME,'VIEW',' '
FROM SYSIBM.SYSTABAUTH T1,SYSIBM.SYSTABLES T2
WHERE T1.GRANTEE = USER
AND T1.TTNAME LIKE '&1'
AND T1.TCREATOR LIKE '&2'
AND T2.TYPE = 'V'
AND (T1.DELETEAUTH IN ('G','Y')
OR T1.INSERTAUTH IN ('G','Y')
OR T1.SELECTAUTH IN ('G','Y')
OR T1.UPDATEAUTH IN ('G','Y'))
AND T1.TTNAME = T2.NAME
AND T1.TCREATOR = T2.CREATOR
ORDER BY CREATOR,NAME;
-＊
-＊ The following code must always be present
-＊
TABLE
ON TABLE PCHOLD FORMAT ALPHA
END

```

上位互換性の維持

バージョン 4.3.x 以前では、インストール時に拡張カタログ (SYSOWNER テーブル) が作成されていました。これにより、SQLTables または ODBC TABL 呼び出しによって返されたテーブルのリストを制御することが可能でした。バージョン 5.1.0 以降では、このテーブルは作

成されません。サーバの旧バージョンのこのテーブルを使用して、引き続きテーブルのリストを管理する必要がある場合、サポートされるサーバプロファイルで以下のコマンドを実行します。

```
SQL sqlengine SET OWNERID ownerid
```

説明

sqlengine

データソースを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ownerid

拡張カタログテーブル SYSOWNER の作成者または所有者を識別します。このコマンドを使用した場合、テーブルのリストを提供するため生成される SQL は、SYSOWNER テーブルの所有者名によって制限されます。

このコマンドは、従来のサーバのリリースでリレーショナルゲートウェイを構成しており、同じ構成を継続するユーザにおいてのみサポートされます。

その他のマスターファイル属性

ここでは、サーバおよび通信構成ファイルを修正する手順について説明します。

以下の属性を使用して、テーブルおよびカラムの情報を記述することが可能です。

REMARKS

テーブルについて説明する属性 (オプション) です。

DESCRIPTION

カラムについて説明する属性 (オプション) です。

TITLE

レポートの FIELDNAME 値 (通常、カラムタイトルとして使用される) を置換する、代替のカラムタイトルを指定する属性 (オプション) です。

サーバでは、API または ODBC を使用するクライアントツールは TITLE 属性を使用しません。サーバカタログを直接照会する場合にのみ、TITLE 属性を使用することができます。

テーブルの説明の記述

REMARKS 属性を使用して、テーブルの説明を記述することができます。

テーブルの説明の記述

構文は次のとおりです。

```
REMARKS=text , $
```

説明

text

単一行のテキストです。78 バイト以内で記述することができます。テキストにカンマ (,) が含まれている場合は、そのテキストを一重引用符 (') で囲む必要があります。

REMARKS 属性は、マスターファイル中の複数行に分けて記述することはできません。複数行に記述する必要がある場合、各行に属性を指定します。

Oracle テーブルの説明の記述

以下は、Oracle テーブル SAMPLE の説明を記述する例です。

```
FILE=SAMPLE,SUFFIX=SQLORA,REMARKS=This is a sample Oracle table., $
```

カラムの説明の記述

DESCRIPTION 属性を使用して、テーブルのカラムに関するコメントを記述することができます。

別の方法として、カラム宣言、セグメント宣言、テーブル宣言のいずれかで、終了を表すドル記号 (\$) の後にコメントを入力することで説明を追加することもできます。また、コメント専用の行を作成するには、宣言の後に新しい行を挿入し、その行の先頭にドル記号 (\$) を

配置します。サーバは、このような方法で追加されたコメントをクライアントアプリケーションで使用することはできません。

マスターファイルで DEFINE フィールドを使用している場合、各行に DESCRIPTION 属性を記述する必要があります。DEFINE の後のセミコロン (;) は、DESCRIPTION= と同一の行に配置する必要があります。

カラムの説明の記述

構文は次のとおりです。

```
DESC[RIPTION]=text , $
```

説明

text

単一行のテキストです。44 バイト以内の文字列を指定します。

テキストにカンマ (,) が含まれている場合は、そのテキストを一重引用符 (') で囲む必要があります。

DESCRIPTION 属性は、マスターファイル中の複数行に分けて記述することはできません。複数行に記述する必要がある場合、各行に属性を指定します。

注意：スペースを節約するには、ALIAS および FIELDNAME 属性と同じ行に説明を記述します。

カラムの説明の記述

以下は、カラム UNITS の説明を記述する例です。説明にカンマ (,) が含まれているため、説明を一重引用符 (') で囲む必要があります。

```
FIELD=UNITS,ALIAS=QTY,FORMAT=I6,DESC='This is quantity sold, not  
returned', $
```

DEFINE フィールドの説明の記述

以下は、DEFINE フィールド ITEMS_SOLD の説明を記述する例です。

```
DEFINE ITEMS_SOLD/D8=ORDERED-INVENTORY  
;DESC=DAMAGED ITEMS NOT INCLUDED,$
```

代替のカラムタイトルの指定

レポートを作成した場合、テーブルのカラム名がレポートの各カラムのデフォルトのタイトルとして表示されます。ただし、TITLE 属性を指定することにより、デフォルトのカラムタイトルを変更することができます。

サーバでは、API または ODBC を使用するクライアントツールは TITLE 属性を使用しません。サーバカタログを直接照会する場合にのみ、あるいは WebFOCUS (Windows バージョン) を使用する場合に TITLE 属性を使用することができます。

レポートリクエストで FIELDNAME および TITLE 属性を AS 句とともに使用して、それらの属性を上書きすることができます。既存の TITLE 属性を上書きするには、SET TITLE コマンドを使用します。HOLD ファイルのマスターファイルに TITLE 属性を適用するには、SET HOLDATTR コマンドを使用します。

カラムに AVE などの接頭語を使用している場合、レポートの TITLE 属性は無効となります。この場合、AS 句を使用して接頭語付きのカラムの代替のカラムタイトルを指定することができます。

マスターファイルで DEFINE フィールドを使用している場合、その行に TITLE 属性を記述する必要があります。DEFINE の後のセミコロンは TITLE= と同じ行に指定しなければなりません。

デフォルトのカラムタイトルの変更

構文は次のとおりです。

```
TITLE='text' ,  
TITLE='text,text,...' ,  
TITLE='text /' ,
```

説明

text

64 バイト以内の文字列を指定します。

テキストは最大 5 行に分けて記述することができます。テキストは、一重引用符 (') で囲む必要があります。レポート出力のテキストを複数行に分割するには、カンマ (,) を使用します。

代替カラムタイトルの後ろにブランクを挿入するには、ブランク挿入後にスラッシュ (/) を追加し、終了を示す一重引用符 (') を入力します。

TITLE 属性は、マスターファイル中の複数行に分けて記述することはできません。複数行に記述する必要がある場合、各行に属性を指定します。

デフォルトのカラムタイトルの変更

以下は、デフォルトのカラムタイトル LNAME を Client Name に変更する例です。

```
FIELD=LNAME,ALIAS=LN,FORMAT=A15,TITLE='Client Name',
```

```
Client Name
-----
```

デフォルトカラムタイトルの 2 行カラムタイトルへの変更

以下は、デフォルトのカラムタイトル LNAME を Client と Name の 2 行のカラムタイトルに変更する例です。

```
FIELD=LNAME,ALIAS=LN,FORMAT=A15,TITLE='Client,Name',
```

```
Client
Name
-----
```

カラムタイトルの下線の制御

以下は、代替のカラムタイトルの下線の長さを制御する例です。

```
FIELD=LNAME,ALIAS=LN,FORMAT=A15,TITLE='Client,Name' /', $
Client
Name
-----
```

DEFINE フィールドのカラムタイトルの指定

以下は、DEFINE フィールド ITEMS_SOLD のデフォルトのカラムタイトルを Items と Sold の 2 行のカラムタイトルに変更する例です。

```
DEFINE ITEMS_SOLD/D8=ORDERED-INVENTORY;TITLE='Items,Sold', $
Items
Sold
-----
```

最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

リクエストの最適化

アダプタは RDBMS の結合、ソート、集計機能を使用する SQL ステートメントを作成することにより、DML リクエストを最適化することができます。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、メニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

リクエストの最適化

```
SQL ADAPTER_ID SET {OPTIMIZATION|SQLJOIN} setting
```

説明

ADAPTER_ID

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

SQLJOIN

OPTIMIZATION の同義語です。

setting

最適化の設定です。次の値が有効です。

ON - アダプタは、RDBMS の結合、ソート、および集計機能を利用する SQL ステートメントを作成します。場合によっては、乗算の影響により最適化が無効になることがあります。ただし、PRINT ベースおよび LIST ベースのレポートリクエストでは、結合されなかったユニークセグメントや乗算された行により最適化が無効になることはありません。これがデフォルト値です。

OFF - アダプタは、各テーブルから単純なデータ取得を実行する SQL ステートメントを作成します。レポートの生成に必要な集計、ソート、結合はすべて、サーバ側のアドレス空間または仮想マシンで処理されます。

両方の OPTIMIZATION 設定で生成されるレポートは同一です。

最適化を OFF に指定して RDBMS に渡された SQL リクエスト

この例では、最適化を無効にして生成された SQL ステートメントを示します。このレポートリクエストは、EMPINFO と FUNDTTRAN の 2 つのテーブルを結合します。この例では、トレースコンポーネントの SQLAGGR および STMTRACE が有効になっています。

最適化を無効にした場合、データアダプタは 2 つの SELECT ステートメントを生成します。最初の SELECT は、EMPINFO テーブルの行のうち、DEPARTMENT カラムに MIS 値が含まれた行をすべて取得します。2 番目の SELECT は、ホストフィールド値 (EMP_ID) に対してパラメータマーカーを使用することで、クロスリファレンスされた FUNDTTRAN テーブルか

ら、EMPINFO 行に対応する行を取得します。パラメータマーカは、使用する RDBMS に応じて異なります (例、?、:000n、:H)。両方の SELECT ステートメントからアンサーセットは正しく取得されますが、結合、ソート、集計処理を実行するのはサーバです。

```
SQL ADAPTER_ID SET OPTIMIZATION OFF
JOIN EMP_ID IN EMPINFO TO ALL WHO IN FUNDTRAN AS J1
TABLE FILE EMPINFO
SUM AVE.CURRENT_SALARY ED_HRS BY WHO BY LAST_NAME
IF DEPARTMENT EQ 'MIS'
END
```

トレース処理では、以下の出力が表示されます。

```
(FOC2510) 次の理由で FOCUS で JOIN が実行されました。
(FOC2511) 最適化は OFF に設定されています。
(FOC2590) 次の理由で集計されませんでした。
(FOC2592) 最適化 / JOIN は行われていません。
SELECT T1.EID,T1.LN,T1.DPT,T1.CSAL,T1.OJT
FROM 'USER1'.'EMPINFO' T1 WHERE (T1.DPT = 'MIS') FOR FETCH ONLY;
SELECT T2.EID FROM 'USER1'.'FUNDTRAN' T2 WHERE (T2.EID = ?)
FOR FETCH ONLY;
```

最適化を ON に指定して RDBMS に渡された SQL リクエスト

最適化を有効にした場合、データアダプタは結合、ソート、集計処理を統合した 1 つの SELECT ステートメントを生成します。RDBMS では、リクエストの管理と処理を行います。サーバは、レポートのフォーマット設定のみを行います。

```
SQL ADAPTER_ID SET OPTIMIZATION ON
JOIN EMP_ID IN EMPINFO TO ALL WHO IN FUNDTRAN AS J1
TABLE FILE EMPINFO
SUM AVE.CURRENT_SALARY ED_HRS BY WHO BY LAST_NAME
IF DEPARTMENT EQ 'MIS'
END
```

トレース処理では、以下の出力が表示されます。

```
AGGREGATION DONE ...
SELECT T2.EID,T1.LN, AVG(T1.CSAL), SUM(T1.OJT)
```

```
FROM 'USER1'.'EMPINFO' T1,'USER1'.'FUNDTRAN' T2
WHERE (T2.EID = T1.EID) AND (T1.DPT = 'MIS')
GROUP BY T2.EID,T1.LN
ORDER BY T2.EID,T1.LN;
```

最適化で生成される SQL 構文の例

生成される SQL 構文は、RDBMS ごとに若干異なります。ただし、アダプタのメッセージは共通で、生成される SQL ステートメントは類似しているため、生成される SQL 構文のほとんどの例は、すべてのリレーショナルアダプタに適用されます。

定数として定義された一時項目 (DEFINE) を転送するリクエストの最適化

定数として定義された一時項目 (DEFINE) は、SQL データベースエンジン (RDBMS) に直接渡され、RDBMS の結合、ソート、集計機能を活用した最適化処理に使用されます。これにより、RDBMS とサーバ間の通信量が減少し、応答時間が短縮されます。以前のバージョンでは、定数はデータベースエンジンに転送されなかったため、最適化処理の対象外でした。

RDBMS に渡される定数のフォーマットは、NUMERIC、CHAR、VARCHAR、CHAR/INTEGER の組み合わせ、および DATE です。ただし、一部のフォーマットは RDBMS に渡されません。CHAR と VARCHAR の組み合わせ、TIME、および DATETIME がこれに該当します。

有効な定数が RDBMS エンジンに渡されると、レポートは定義された値に基づいて計算され、計算されたレコードサブセットだけが返されます。アダプタレベルでのデータ処理範囲は限られているため、パフォーマンスが向上します。

数値定数の転送

```
DEFINE FILE SMIX87
INTEGERCONST/I4 = 5
REALCONST/D20.2 = -97995.38
NUMERICCONST/P13.4 = -92999.3647
```

```
TABLE FILE SMIX87
SUM INTEGERCONST REALCONST NUMERICCONST MAX.QUOT.FA01INTEGER
BY QUOT.FA02INTEGER
END
```

以下の SQL が生成されます。

```
SELECT T2."FA02INTEGER", SUM(5), SUM(-97995.38),
SUM(-92999.3647), MAX(T2."FA01INTEGER") FROM TMIX83A T2 GROUP
BY T2."FA02INTEGER" ORDER BY T2."FA02INTEGER";
```

CHAR 定数の転送

```
DEFINE FILE SMIX87
CHARCONST/A10 = '2N'
```

```
TABLE FILE SMIX87
SUM CHARCONST MIN.FA02CHAR_15
BY QUOT.FA02INTEGER
END
```

以下の SQL が生成されます。

```
SELECT T2."FA02INTEGER", MAX('2N'), MIN(T3."FA02CHAR_15") FROM
TMIX83A T2,TMIX86A T3 WHERE (T3."FA01INTEGER" =
T2."FA01INTEGER") GROUP BY T2."FA02INTEGER" ORDER BY
T2."FA02INTEGER";
```

CHAR および INTEGER 定数の転送

```
DEFINE FILE SMIX87
CHARCONST/A10 = '2N'
INTEGERCONST/I4 = 5
```

```
TABLE FILE SMIX87  
SUM CHARCONST INTEGERCONST  
BY QUOT.FA02INTEGER  
END
```

以下の SQL が生成されます。

```
SELECT T2."FA02INTEGER", MAX('2N'), SUM(5) FROM TMIX83A T2  
GROUP BY T2."FA02INTEGER" ORDER BY T2."FA02INTEGER";
```

Amazon Athena アダプタの使用

Amazon Athena アダプタの構成方法について説明します。

Amazon Athena アダプタを使用することで、標準 SQL による Amazon Simply Storage Server (Amazon S3) に格納されたデータへのアクセスおよびデータ分析が可能になります。

Amazon Athena アダプタの概要

Amazon Athena は、インタラクティブなクエリサービスです。このサービスを利用することで、標準 SQL を使用して Amazon S3 のデータを直接分析することが可能になります。Athena で Amazon S3 に格納されたデータを指定し、SQL または adhoc クエリを実行して、結果を迅速に取得することができます。

Amazon Athena アダプタは、ODBC と JDBC の 2 つのタイプで使用できます。構成可能なタイプは、サーバインスタンスごとに 1 つのみです。Amazon は、ODBC (Windows および Linux)、JDBC ドライバ、およびそれぞれに対応するタイプのアダプタ (ODBC または JDBC Athena API) を使用して接続およびレポート作成します。

Amazon Athena 環境の準備

Amazon Athena アダプタを使用するには、次のコンポーネントが必要です。

ODBC ドライバ

- Amazon Athena ODBC ドライバをダウンロードおよびインストールします。
- 使用するオペレーティングシステムの Amazon/Simba の説明に従って、構成手順を実行します。
- AWS リージョン、S3 出力ロケーション、暗号化オプション、AWS アクセスキー、AWS シークレットキーを Amazon Athena の管理者から取得します。

JDBC ドライバ

- Amazon Athena JDBC ドライバをダウンロードおよびインストールします。

- 使用する JDBC ドライバの jar ファイルのパスを取得します。たとえば、Linux で Java 8 に対応する JDBC 4.1 を使用する場合は、次のファイルパスが必要です。

```
/qas/authena/AthenaJDBC41-1.0.1.jar
```

Athena ドライバの .jar ファイルのフルパスを示すこの値が、構成時に IBI_CLASSPATH に追加されます。

- 接続 URL、S3 ステージングディレクトリ、AWS アクセスキー、AWS シークレットキーを Amazon Athena 管理者から取得します。

Amazon Athena アダプタの構成

アダプタを構成するには、確立する各接続の接続情報および認証情報を指定します。

アダプタを構成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定

のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。

4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Amazon Athena ODBC アダプタの構成設定

Amazon Athena ODBC アダプタは、[SQL] グループフォルダ下にあります。このアダプタの構成設定は次のとおりです。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

DSN を使用しない

DSN の代わりに AWS リージョンおよび S3 出力ロケーションを使用する場合は、このオプションを選択します。

DSN

Athena データソース用に構成された DSN です。

セキュリティ

データベースサーバへの接続時に使用する認証方法として [Explicit] を選択します。

AWS アクセスキー ID

Amazon のキーベースのアクセスコントロールで使用する AWS アクセスキーです。

AWS シークレットアクセスキー

Amazon のキーベースのアクセスコントロールで使用する AWS シークレットアクセスキーです。

AWS アカウント ID

AWS アカウントの一意の識別番号です。

AWS ロール名

AWS でのユーザの実行権限を指定する権限ポリシーです。

データロードの許可

指定した接続でアップロード、クイックコピー、カスタムコピーを許可します。デフォルト値は [はい] です。

接続の説明 (オプション)

指定した接続を説明するユーザ定義のメモです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

Amazon Athena JDBC アダプタの構成設定

Amazon Athena アダプタは、[SQL] グループフォルダ下にあります。このアダプタの構成設定は次のとおりです。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

URL

Amazon Athena JDBC データソースのロケーション URL です。

次の例は、Amazon Athena サーバへの接続に使用する URL を示しています。

```
jdbc:awsathena://connection_URL:?s3_staging_directory
```

説明

connection_URL

Amazon Athena の管理者から取得した接続 URL です。

s3_staging_directory

Amazon Athena の管理者から取得したステージングディレクトリです。

セキュリティ

データベースサーバへの接続時に使用する認証方法として [Explicit] を選択します。

AWS アクセスキー ID

Amazon のキーベースのアクセスコントロールで使用される AWS アクセスキーです。

AWS シークレットアクセスキー

Amazon のキーベースのアクセスコントロールで使用される AWS シークレットアクセスキーです。

AWS アカウント ID

AWS アカウントの一意的識別番号です。

AWS ロール名

AWS でのユーザの実行権限を指定する権限ポリシーです。

ドライバ名

JDBC ドライバ名として、com.amazonaws.athena.jdbc.AthenaDriver と入力します。

IBI_CLASSPATH

Java サービスで使用する追加の Java クラスディレクトリ、または jar ファイル名のフルパスを定義します。値を設定するには、通信ファイルを編集するか、WebFOCUS Reporting Server ブラウザインターフェースを使用します。WebFOCUS Reporting

Server ブラウザインタフェースの入力ボックスでは、1 行につき 1 つの参照を入力することができます。すべてのプラットフォームでは、ファイルを保存した際に、複数のエントリがコロン (:) で区切られた単一文字列に変換されます。ファイルを手動で編集する際は、区切り文字のコロン (:) を保持する必要があります。

[Amazon Athena 環境の準備](#)で指定した Athena ドライバの jar ファイルのフルパスを入力します。

データロードの許可

指定した接続でアップロード、クイックコピー、カスタムコピーを許可します。デフォルト値は [はい] です。

接続の説明 (オプション)

指定した接続を説明するユーザ定義のメモです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

Amazon Athena でのシノニムの作成

シノニムを作成して、接続によりアクセス可能な Amazon Athena 外部テーブル (外部テーブルは、Athena で使用可能な唯一のオブジェクトタイプ) の一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、動作を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが指定した Athena 外部テーブルに基づいて生成されます。

シノニムを作成するには

シノニムを作成するには、事前にアダプタおよび接続を構成しておく必要があります。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

5. 必要に応じて、[データタイプマッピングのカスタマイズ] セクションを展開し、[LONGCHAR 長さ] または数値精度の値を入力します。

結果

[ステータス] ウィンドウに、シノニムが正しく作成されたことを示す情報が表示されます。作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

データタイプのサポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[データタイプレポートを表示するには](#)を参照してください。

制限事項

Athena は、スキーマオンリードというアプローチを使用します。これは、クエリの実行時にスキーマをデータに適用する方法です。これにより、データのロードまたは変換の必要がなくなります。Athena は、Amazon S3 のデータを変更しない代わりに、Apache Hive を使用してテーブルを定義し、データベースを作成します。これらは、基本的にはテーブルの論理ネームスペースです。

Athena でテーブルを削除すると、テーブルのメタデータのみが削除され、データは Amazon S3 に保持されます。

Amazon AWS S3 アダプタの使用

Amazon AWS S3 アダプタの構成および使用方法について説明します。

AWS S3 (Amazon Web Services Simple Storage Service) アダプタは、AWS S3 との統合に使用される WebFOCUS アダプタです。WebFOCUS レポートから生成される PDF、JSON、XML などの出力フォーマットは、AWS S3 環境内のバケットに格納することができます。アダプタ (例、JSON、XML) を使用して、AWS S3 に保存された関連するドキュメントに対するレポートが作成できます。

AWS S3 アダプタは、WebFOCUS Reporting Server ブラウザインターフェースを使用して構成することができます。アダプタには、AWS IAM (Identity and Access Management) コンソールから取得したアクセスキー ID およびシークレットアクセスキーを保存する接続が必要です。

AWS IAM および AWS S3 の構成

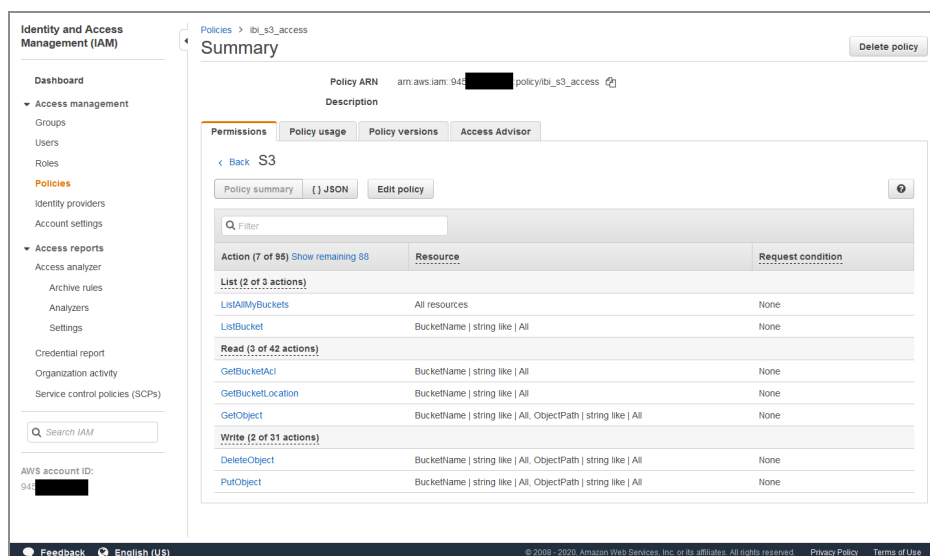
ここでは、AWS S3 アダプタの操作に必要な AWS IAM および AWS S3 の構成方法について説明します。

AWS IAM および AWS S3 を構成するには

手順

1. AWS S3 アダプタと AWS S3 を正常に統合できるよう、AWS にポリシーを作成します。このポリシーは、AWS S3 へのアクセスを要求する AWS ユーザ ID と関連付ける必要があります。

下図のように、S3 で必要最低限の操作権限は、ListAllMyBuckets、ListBucket、GetBucketACL、GetBucketLocation、GetObject、DeleteObject、PutObject です。



2. アクセスキー ID およびシークレットアクセスキーを作成します。

アクセスキー ID とシークレットアクセスキーは、AWS S3 アダプタを構成するために必要です。

次のリンクからアクセス可能な AWS のマニュアルには、これらのキーの作成方法が説明されています。

<https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html#access-keys-and-secret-access-keys>

3. AWS S3 にバケットを作成します。

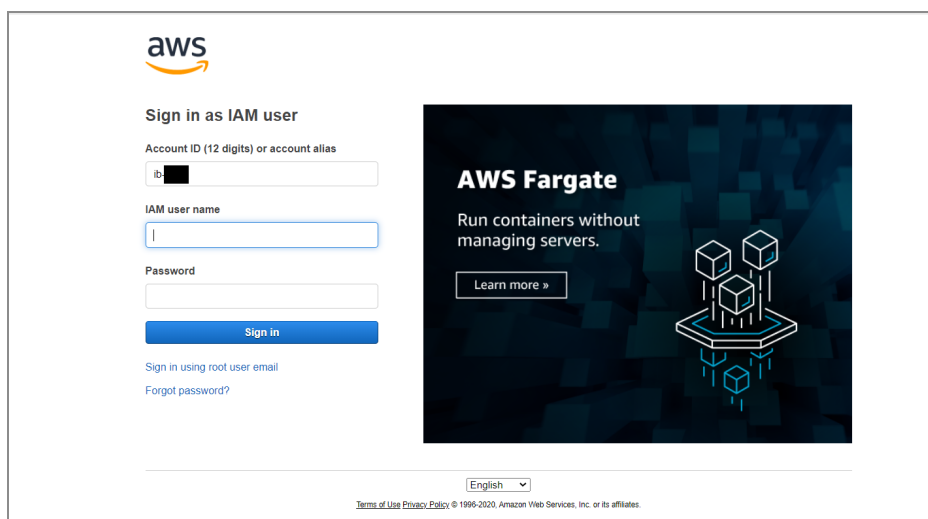
AWS S3 バケットはファイルの格納に使用されます。

AWS S3 の各接続には、ファイルの読み取りと書き込み用の固有のバケットが含まれます。

- a. Web ブラウザのアドレスバーに次の URL を入力します。

<https://s3.console.aws.amazon.com/s3>

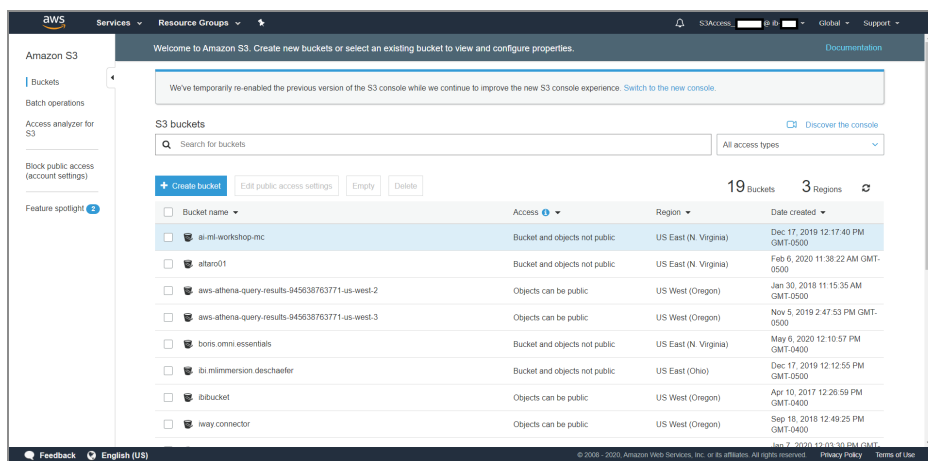
AWS にログインしていない場合は、下図のように、AWS のログインダイアログが表示されます。



The image shows the AWS sign-in interface. On the left, there is a form titled "Sign in as IAM user" with fields for "Account ID (12 digits) or account alias" (containing "ib-"), "IAM user name" (containing "|"), and "Password". Below these fields is a blue "Sign in" button. Links for "Sign in using root user email" and "Forgot password?" are also present. On the right, there is a promotional banner for "AWS Fargate" with the text "Run containers without managing servers." and a "Learn more »" button. At the bottom, there is a language selector set to "English" and a footer with "Terms of Use" and "Privacy Policy" links.

- b. AWS S3 にアクセス可能な有効な AWS アカウント認証情報を入力し、[サインイン] をクリックします。

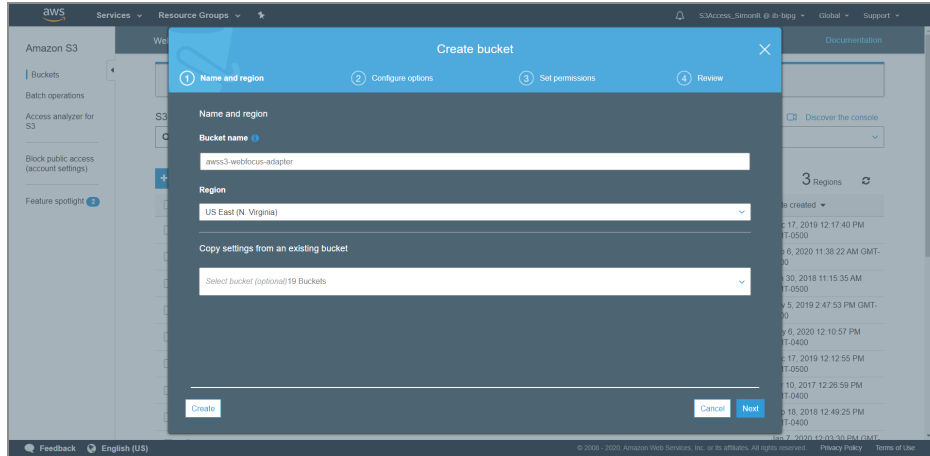
下図のように、AWS S3 の画面が開きます。



The image is a screenshot of the AWS S3 console. The top navigation bar shows "aws" logo, "Services", "Resource Groups", and user information. The main header says "Welcome to Amazon S3. Create new buckets or select an existing bucket to view and configure properties." Below this, there is a message about the previous version of the console being re-enabled. The "S3 buckets" section is active, showing a search bar and a table of buckets. The table has columns for "Bucket name", "Access", "Region", and "Date created". There are 19 buckets listed, including "ai-ml-workshop-mc", "altaro01", "aws-athena-query-results-945638763771-us-west-2", "aws-athena-query-results-945638763771-us-west-3", "boris-omni-essentials", "bi-milimmersion-deschaefer", "ibibucket", and "lway connector". The left sidebar contains links for "Buckets", "Batch operations", "Access analyzer for S3", "Block public access (account settings)", and "Feature spotlight".

- c. [+ バケットを作成する] をクリックします。

下図のように、[バケットの作成] パネルが開きます。



- d. Amazon S3 のすべての既存バケット名で固有のバケット名を入力します。
[リージョン] を選択します (例、米国東部 (バージニア北部))。
- e. [作成] をクリックし、デフォルト構成および権限を使用するバケットを作成します。

AWS S3 アダプタの構成

ここでは、AWS S3 アダプタの構成方法について説明します。

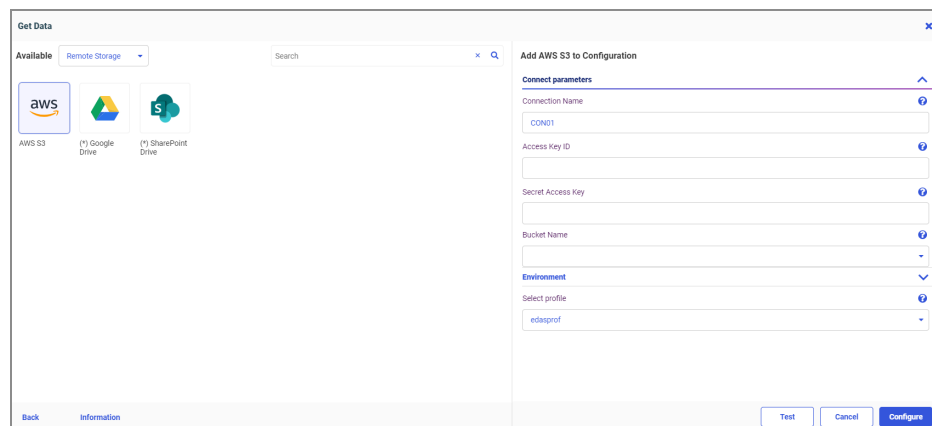
AWS S3 は、リモートストレージアダプタのカテゴリに分類されます。

AWS S3 アダプタを構成するには

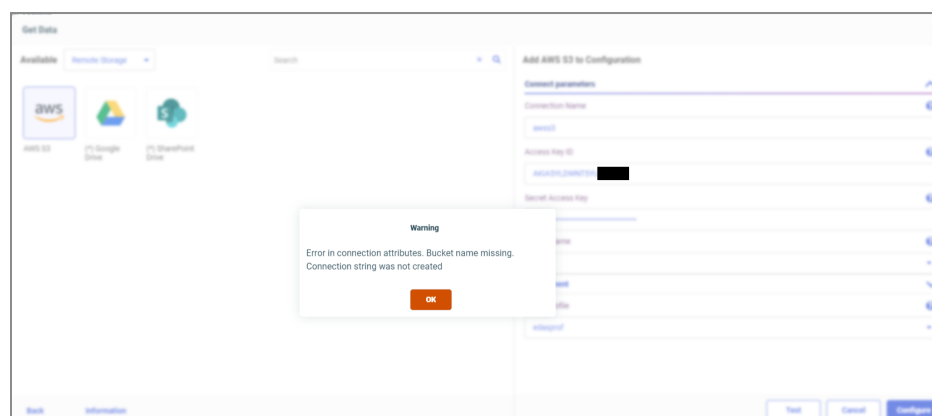
手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
[データの取得] ページが開き、構成済みアダプタが表示されます。
2. プラス (+) ボタン (新規アダプタの追加) をクリックし、アダプタを検索します。
WebFOCUS Reporting Server ブラウザインターフェースでは、ドロップダウンリストからアダプタのカテゴリ (AWS S3 は [リモートストレージ] カテゴリ) を選択することも、検索オプション (拡大鏡) を使用して特定の文字列を検索することもできます。
3. AWS S3 のノードを右クリックし、[構成] を選択します。

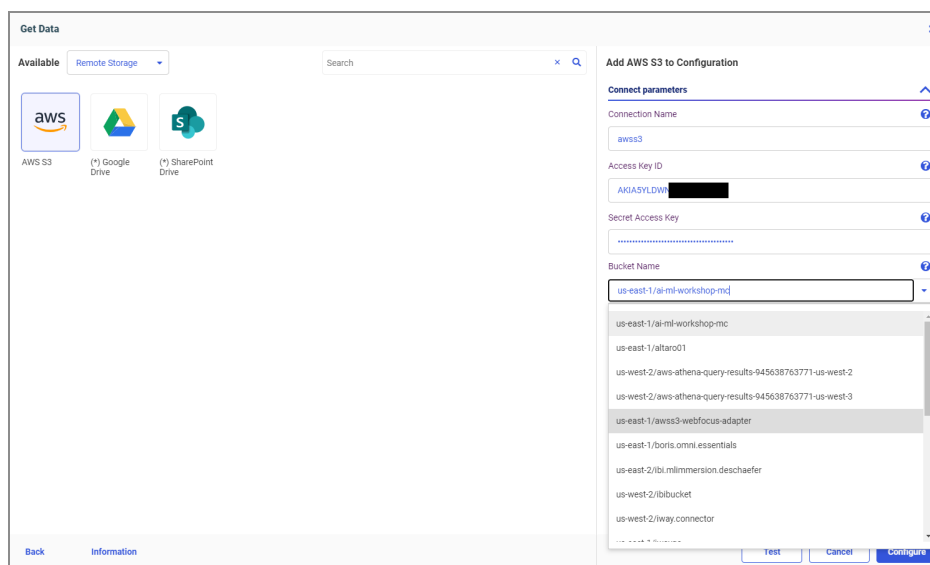
[AWS S3 の接続を追加] ウィンドウが開きます。下図は、[データの取得] の簡略モードを示しています。



4. AWS で構成したアクセスキー ID の値を入力します。
詳細は、[AWS IAM](#) および [AWS S3 の構成](#)を参照してください。
5. AWS で構成したシークレットアクセスキーの値を入力します。
詳細は、[AWS IAM](#) および [AWS S3 の構成](#)を参照してください。
6. AWS S3 アダプタをはじめて構成する場合は、[構成] をクリックします。
下図のように、[バケット名] の値がありませんというエラーメッセージが表示されます。



7. [OK] をクリックします。
8. 下図のように、[バケット名] ドロップダウンリストの下向き矢印をクリックし、リストに表示されるまで待機します。



注意：リストに表示されるまで少し時間がかかる場合があります。

9. バケット名を選択し、[構成] をクリックします。

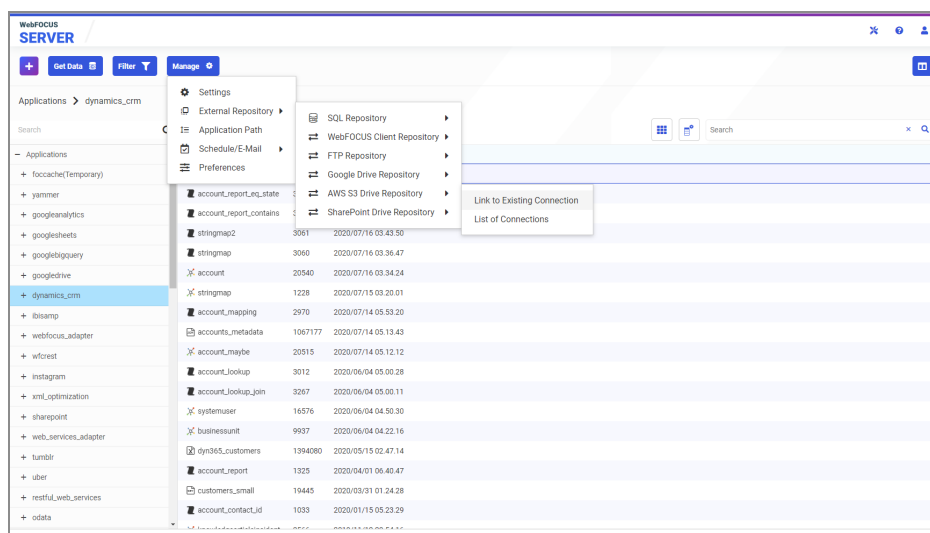
AWS S3 のマッピングの構成

ここでは、AWS S3 リポジトリのマッピングの構成方法について説明します。

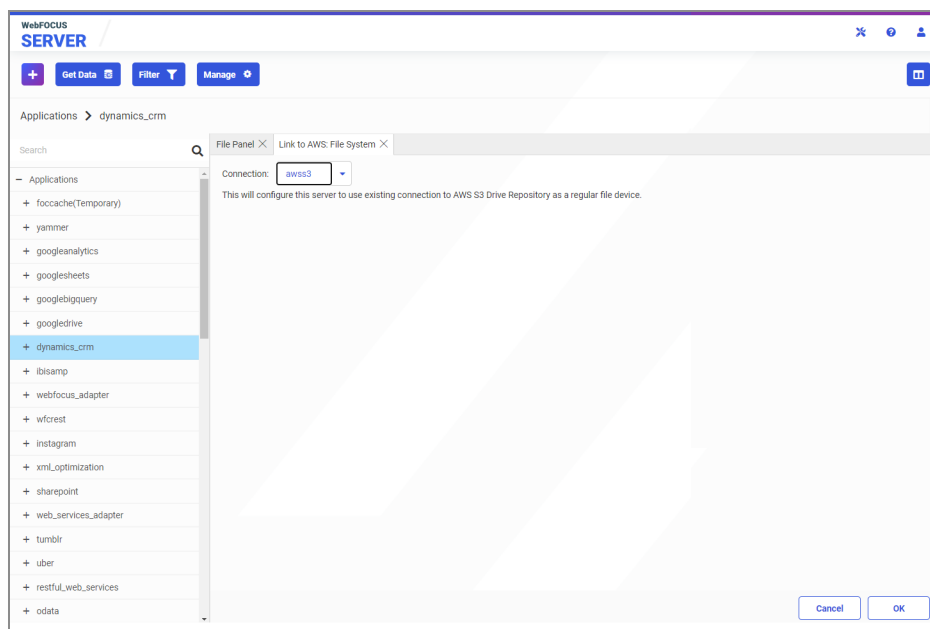
AWS S3 のマッピングを構成するには

手順

1. 下図のように、WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで [管理] をクリックして [外部リポジトリ]、[AWS S3 ドライブリポジトリ]、[既存接続へのリンク] を順に選択します。

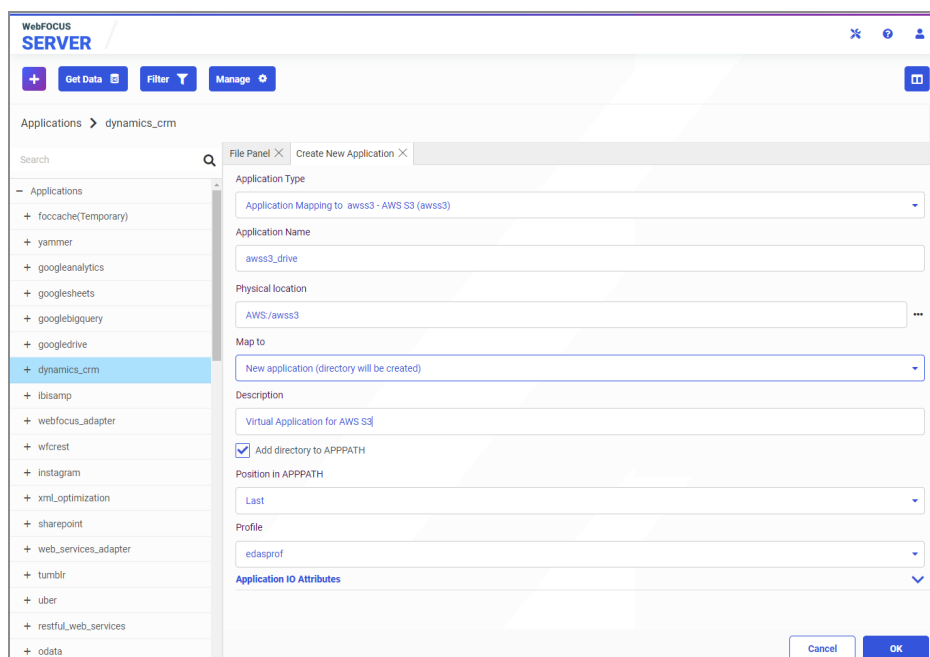


2. 下図のように、AWS S3 ドライブの接続名を選択します。



3. [OK] をクリックします。

下図のように、AWS S3 ドライブのアプリケーション構成画面が表示されます。



4. 次の手順を実行します。

- a. 新しいアプリケーション名を入力し、必要な場合はアプリケーションの説明も入力します。

以下はその例です。

- **アプリケーション名** awss3_drive
- **説明** AWS S3 の仮想アプリケーション

- b. [マッピング先] ドロップダウンリストから [新規アプリケーション (ディレクトリが作成されます)] を選択します。

5. [OK] をクリックします。

インターネットスピードおよび AWS S3 ドライブのファイル数によって、マッピング処理に数分かかる場合があります。

マッピング処理が完了すると、「コマンドを実行しました。」というメッセージが表示されます。

AWS S3 の接続属性

以下は、AWS S3 アダプタの接続属性を示しています。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

アクセスキー ID

Amazon のキーベースのアクセスコントロールで使用される AWS アクセスキーです。

シークレットアクセスキー

Amazon のキーベースのアクセスコントロールで使用される AWS シークレットアクセスキーです。

バケット名

この接続に割り当てられた AWS リージョンおよび Amazon S3 のバケット名です。

バケットは、オブジェクトを格納するコンテナです。

リージョンは、同一地域で指定された一連の AWS リソースです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、ドロップダウンメニューから [新規プロファイル] を選択し、[プロファイル名] フィールドに名前を入力します (拡張子は自動的に付加されます)。

Amazon Redshift アダプタの使用

Amazon Redshift アダプタの構成方法について説明します。

Amazon Redshift アダプタを使用することで、アプリケーションによる Amazon Redshift データソースへのアクセスが可能になります。このアダプタにより、データまたはアプリケーションリクエストが、各構成に基づいてネイティブの Amazon Redshift ODBC コールまたは JDBC コールに変換され、最適化されたアンサーセットがリクエストの送信元プログラムに返されます。

注意：バージョン 9.3.0 以降、Amazon Redshift アダプタは UPSERT サポートを提供します。

Amazon Redshift アダプタの概要

Amazon Redshift は、ODBC と JDBC の 2 つの構成オプションをサポートします。単一のサービンスタンスに実装できるのは、1 つの構成タイプのみです。

- **ODBC ベース** この構成では、Windows および Linux Red Hat (バージョン 7 および 6) 環境にインストール可能な Amazon Redshift ODBC ドライバを使用します。

Linux 環境の Redshift ODBC ドライバでは、ODBC ドライバ以外に ODBC ドライバマネージャ (例、unixODBC または iODBC) のインストールと使用が必要です。使用するドライバマネージャによって、追加の環境変数の設定が必要になります。詳細は、ドライバマネージャのマニュアルを参照してください。

- **JDBC ベース** この構成では、UNIX、Linux、Windows 環境にインストール可能な Amazon Redshift JDBC ドライバを使用します。PowerPC アーキテクチャではサポートされません。

必要なドライバのダウンロードおよびインストールについての詳細は、WebFOCUS Reporting Server ブラウザインターフェースの [データの取得] ページで [Amazon Redshift] アイコンを右クリックし、[要件] を参照してください。

Amazon Redshift ODBC 環境の準備

Windows の場合、Amazon Redshift ODBC 環境は、Redshift ODBC ドライバのインストール時に設定されます。その他の設定の手順は必要ありません。

Linux の場合

- Amazon Redshift ODBC ドライバのインストール後に、Redshift ドライバのインストール先ディレクトリのパスが、LD_LIBRARY_PATH (サーバがセキュリティオンで実行されている場合は IBI_LIBPATH) に含まれていることを確認してください。

注意：IBI_ 変数は、edaenv.cfg ファイルに名前と値の組み合わせ (name=value) を入力することで設定できます。edaenv.cfg ファイルを編集するには、WebFOCUS Reporting Server ブラウザインターフェースの [ワークスペース] ページに移動して [構成ファイル]、[その他] を順に展開し、[環境 - edaenv.cfg] を右クリックして [編集] を選択します。

以下はその例です。

```
IBI_LIBPATH=/opt/amazon/redshiftodbc/lib/64
```

- AMAZONREDSHIFTODBC 変数で Linux ドライバマネージャの初期化ファイルを指定し、この初期化ファイルに DriverManagerEncoding=UTF-16 設定を記述します。

以下はその例です。

```
export AMAZONREDSHIFTODBC=/usr/odbc/ibiodbc/amazon.redshiftodbc.ini
```

- IBI_LOAD_REDSHIFT_ODBC_DRIVER_FROM=**path** を設定し、Redshift ODBC ドライバディレクトリの DLL ロードパスを保存します。以下はその例です。

```
IBI_LOAD_REDSHIFT_ODBC_DRIVER_FROM=path/amazon/redshiftodbc/lib/64
```

Amazon Redshift (ODBC) アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

Amazon Redshift アダプタを構成するには

アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで構成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。

4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。

5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。

6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク

(*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Amazon Redshift アダプタの構成設定 (ODBC)

Amazon Redshift アダプタは、[SQL] グループフォルダ下にあります。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

システム

Amazon Redshift データベースが稼働している Amazon Redshift クラスタのエンドポイントです。基本構文は、**host** です。ポートは、[追加の接続文字列キーワード] テキストボックスで指定できます。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

デフォルトデータベース

接続するデータベースの名前です。

追加の接続文字列キーワード (オプション)

接続文字列のオプションを追加するために使用します (例、'PORT=1563')。

データロードの許可

指定した接続でアップロード、クイックコピー、カスタムコピーを許可します。デフォルト値は [はい] です。

接続の説明 (オプション)

指定した接続を説明するユーザ定義のメモです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

Amazon Redshift (JDBC) アダプタの構成

アダプタを構成するには、確立する各接続の接続情報および認証情報を指定します。

Amazon Redshift アダプタを構成するには

アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで構成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Amazon Redshift アダプタの構成設定 (JDBC)

Amazon Redshift アダプタは、[SQL] グループフォルダ下にあります。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

URL

Amazon Redshift JDBC データソースのロケーション URL です。基本構文は、以下のとおりです。

```
jdbc:redshift://host.redshift.amazonaws.com:1563/dev
```

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

ドライバ名

JDBC ドライバの名前です (例、com.amazon.redshift.jdbc42.Driver)。

IBI_CLASSPATH

Java サービスで使用する追加の Java クラスディレクトリ、または jar ファイル名のフルパスを定義します。入力ウィンドウの 1 行に 1 つ参照を入力します。

データロードの許可

指定した接続でアップロード、クイックコピー、カスタムコピーを許可します。デフォルト値は [はい] です。

接続の説明 (オプション)

指定した接続を説明するユーザ定義のメモです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

Amazon Redshift アダプタの拡張バルクロード構成

拡張バルクロードは、Amazon Redshift アダプタの ODBC および JDBC バージョンで構成することができます。拡張バルクロードでは、ベンダーの指示に従って、サーバノードに AWS CLI (コマンドラインインターフェース) をインストール、構成する必要があります。インストールおよび構成の完了後、次の手順を実行します。

1. Redshift DBA には、AWS S3 のバケットへの読み取り/書き込み権限を所有する Amazon IAM ロールで ID を設定する必要があります。これにより、暗号化されたアカウント ID、ロール名、ロールプロファイルが生成されます。
2. WebFOCUS Reporting Server ブラウザインターフェースの [データの取得] ページの [サーバのデータソース] 下で、構成済みの [Amazon Redshift] アダプタを右クリックし、[バルクロードの構成] を選択します。
3. デフォルト設定の [接続名] を受容します。
4. [プロトコル] では、[AWS (ロールベースアクセス制御)] を選択します。
5. [バケット名] には、AWS S3 のデータのアップロード先を入力します。
6. [アカウント ID]、[ロール名]、[アクセスキー ID]、[シークレットアクセスキー]、[RDS インポートロール名] を各テキストボックスに入力します。
注意：アカウント ID、ロール名、アクセスキー ID およびシークレットアクセスキーは、AWS S3 バケットにステージデータを書き込むために必要です。RDS インポートロール名は、AWS S3 から Redshift テーブルにデータをロードする Redshift COPY SQL コマンド実行の認可に必要です。
7. 固定フォーマットファイルの接続を構成し、テストします。

拡張バルクロードが正しく構成されていることを確認するには、構成済みの [Amazon Redshift] 接続を右クリックし、[バルクロードテスト] を選択します。

Amazon Redshift でのシノニムの作成

シノニムを作成して、サーバからアクセス可能な Amazon Redshift テーブルの一意の名前またはエイリアスを定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、動作を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、マスターファイルとアクセスファイルが指定した Amazon Redshift テーブルに基づいて生成されます。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

データタイプのサポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[データタイプレポートを表示するには](#)を参照してください。

Amazon Redshift でのダイレクトパススルーの使用

テーブル名が SHOW TABLES および DESCRIBE などのダイレクトパススルーコマンドでは、結果が表示されません。

これは、次の記述を追加することで回避できます。

```
SQL SQLRDS /*QUERY*/ show all;  
END
```


Apache Drill アダプタの使用

Apache Drill アダプタの構成方法について説明します。

Apache Drill は、Hadoop、SQL 以外の各種データベース、および HDFS、MapR-FS、HBase、MapR-DB、MongoDB などのファイルシステムに対応するクエリエンジンです。Apache Drill は、Hive メタデータを使用することも、自己記述型データファイルからメタデータを取得することもできます。

Apache Drill 環境の準備

Apache Drill アダプタを使用するには、次のコンポーネントが必要です。

- **Java** この Java ベースのアダプタを使用するには、Java のインストールが必要です。Apache Drill では、バージョン 1.7 以降を使用する必要があります。Java は <http://www.java.com> からダウンロード可能です。

Java インストールのディレクトリパスは、環境変数で指定する必要があります。

Linux を使用する場合、Java インストール先のプロファイルに 1 行を追加します。以下はその例です。

```
export JAVA_HOME=/usr/lib/jvm/jre-1.7.0
```

JDK をインストールした場合は、次のような 1 行を追加します。

```
export JAVA_HOME=/usr/lib/jvm/jdk-1.7.0
```

Windows を使用する場合は、[PC] を右クリックし、[プロパティ] を選択します。次に、[システムの詳細設定] を選択し、[環境変数] をクリックします。PATH 変数にパスを追加します。以下はその例です。

```
C:¥Program Files¥Java¥jdk7¥bin¥server;C:¥Program Files¥Java¥jdk7¥bin;
```

- **JDBC ドライバ**

Apache Drill では、2 つの JDBC ドライバが使用可能です。

オープンソース JDBC ドライバは、Apache Drill に同梱されています。ドライバは単一ファイルで構成され、インストール後に次のディレクトリパスに格納されます。

```
drill_installation/jars/jdbc-driver/drill-jdbc-all-version.jar
```

説明

drill_installation

Apache Drill をインストールした場所です。

version

ドライバのバージョンです。

<https://drill.apache.org/docs/using-the-jdbc-driver> からダウンロードすることもできます。

オープンソース JDBC ドライバを使用し、インストールした Apache Drill と同一システムにサーバをインストールする場合、次のセクションの説明に従って jar ファイルを指定することができます。サーバを別のシステムにインストールする場合は、ファイルを任意の場所にコピーします。

MapR JDBC ドライバは、MapR から取得できます。詳細は、以下の手順 3 を参照してください。このドライバは、MapR SASL 認証もサポートします。

Java CLASSPATH を構成するには

Apache Drill の jar ファイルパスは、サーバに指定する必要があります。サーバが Drillbit と同一システムで稼働している場合は、このパスを指定することができます。サーバが別のシステムで稼働している場合、以下に示したファイルをシステムの任意の場所にコピーし、このパスを指定します。

これは、システムの CLASSPATH またはサーバの IBI_CLASSPATH 変数のいずれかで実行できます。以下はその手順です。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースから [ワークスペース] に移動します。

または

ibi Data Migrator デスクトップインターフェースで、[ワークスペース] フォルダを展

開します。

2. [Java サービス] フォルダを選択し、[DEFAULT] を右クリックしてから [プロパティ] を選択します。

[Java サービスの構成] ページが開きます。

3. [クラスパス] セクションを展開します。

オープンソース JDBC ドライバの場合

[IBI_CLASSPATH] テキストボックスに、jar ファイルのフルパスを入力します。以下はその例です。

```
drill_installation/jars/jdbc-driver/drill-jdbc-all-version.jar
```

説明

drill_installation

Apache Drill をインストールした場所です。

version

ドライバのバージョンです。

HBase または MapR-DB に格納されたデータにアクセスする場合は、jar ファイルを追加します。以下はその例です。

```
drill_installation/jars/3rdparty/hadoop-common-2.7.0.jar
```

注意：ファイル名は、1 行に 1 つ入力する必要があります。

アダプタを Apache Drill のインストール先と異なるシステムにインストールした場合、jar ファイルをそのシステムの任意の場所にコピーします。

注意：Windows で稼働するサーバの場合、ディレクトリ名には Windows 構文を使用します。以下はその例です。

```
C:¥Drill¥jars¥jdbc-driver¥drill-jdbc-all-version.jar
```

MapR (Simba) JDBC ドライバの場合

MapR が提供する Drill JDBC ドライバのリンク

(https://maprdocs.mapr.com/home/Drill/drill_jdbc_connector.html) から MapR JDBC ドライバをダウンロードします。これにより、次の名前のファイルがダウンロード

ドされます。

```
MapRDrillJDBC41-version.zip
```

注意：バージョン番号は、リリースによって変更されます。

このファイルをサーバ上の任意の場所に解凍します。MapR Drill JDBC ドライバは、7 つの jar ファイルで構成されます。

上記の説明に従って、この jar ファイルの名前をすべて CLASSPATH または IBI_CLASSPATH に追加します。

4. 下方向へスクロールし、[Java サービスを保存して再起動] ボタンをクリックします。

Apache Drill アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

Apache Drill アダプタを構成するには

アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで構成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. [URL] テキストボックスに、Drillbit への接続に使用する URL を入力します。詳細は、[MapR JDBC ドライバを使用した Apache Drill URL の構成設定](#)を参照してください。
6. [ドライバ名] テキストボックスに、下表から使用する JDBC ドライバを入力します。

```
org.apache.drill.jdbc.Driver
```

アダプタ	JDBC ドライバ名
Apache Hive	<pre>org.apache.drill.jdbc.Driver</pre>
MapR Drill	<pre>com.mapr.drill.jdbc41.Driver</pre>

7. ユーザとパスワードを入力します。
下図は、構成設定の例を示しています。

Add Apache Drill to Configuration

? Prerequisites

^ **Connect parameters**

? Connection Name

? URL

? Security

? User

? Password

? Driver Name *

? IBI_CLASSPATH **

* - Common for all connections of the adapter
** - Common for all Java-based adapters

^ **Environment**

? Select profile (type in a new one or select one from the list)

8. [テスト] をクリックします。サーバ上のデータソースのリストが表示されます。
9. [構成] をクリックします。

Apache Drill の接続属性

Apache Drill アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

URL

Apache Drill データソースのロケーション URL です。

ドライバ名

Apache Drill ドライバの名前です。

IBI_CLASSPATH

Java サービスで使用する追加の Java クラスディレクトリ、または jar ファイル名のフルパスを定義します。値を設定するには、通信ファイルを編集するか、WebFOCUS Reporting Server ブラウザインターフェースを使用します。WebFOCUS Reporting Server ブラウザインターフェースの入力ボックスでは、1 行につき 1 つの参照を入力することができます。すべてのプラットフォームでは、ファイルを保存した際に、複数のエントリがコロン (:) で区切られた単一文字列に変換されます。ファイルを手動で編集する際は、区切り文字のコロン (:) を保持する必要があります。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 3 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。
- **Trusted** アダプタは、現在のオペレーティングシステムに関連している偽装プロセス用のデータベースの規則を使用してデータベースに接続します。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (括弧は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

オープンソース Drill JDBC ドライバを使用した Apache Drill URL の構成設定

Apache Drill アダプタは、[SQL] グループフォルダ下にあります。

Zookeeper を使用する場合、Apache Drill アダプタの URL 構成は次のとおりです。

```
jdbc:drill:zk=zkname:port/directory/cluster_id[;schema=schema]
```

Drillbit に直接接続する場合、Apache Drill アダプタの URL 構成は次のとおりです。

```
jdbc:drill:drillbit=nodename:port[;schema=schema]
```

説明

zkname

ZooKeeper のホスト名または IP アドレスを指定します。ローカルの Drillbit に接続する場合は、ホスト名または IP アドレスの代わりに local を使用します。

nodename

Apache Drill を実行するクラスタノードの 1 つまたは複数のホスト名または IP アドレスを指定します。

port

ポート番号です。デフォルト値は 2181 です。MapR クラスタのデフォルト値は 5181 です。Drillbit への直接接続のデフォルト値は 31010 です。

注意：複数の zkname:port の値には、カンマ (,) 区切りを使用することができます。

directory

Zookeeper 内の drill ディレクトリの名前です。デフォルト値は drill です。

cluster_id

drill クラスタの名前です。デフォルト値は Drillbit1 です。

schema

使用するストレージプラグインおよびスキーマの名前です (例、hive または dfs.directory)。

MapR JDBC ドライバを使用した Apache Drill URL の構成設定

Apache Drill アダプタは、[SQL] グループフォルダ下にあります。

クライアント Drillbit 接続に使用する URL 構成文字列は、次のとおりです。

```
jdbc:drill:drillbit=host[:port][:auth=method[:delegationuid=id]
```

説明

host

Drillbit が稼動しているマシン名または IP アドレスです。

port

Drillbit のポートです。デフォルト値は 31010 です。

クライアント Zookeeper 接続に使用する URL 構成文字列は、次のとおりです。

```
jdbc:drill:zk=zkname[:port]/drill/[cluster][:auth=method[:delegationuid=id] ]
```

説明

zkname

Zookeeper が稼動しているマシン名または IP アドレスです。

port

Zookeeper のポートです。MapR クラスタのデフォルト値は 5181 です。

cluster

Drillbit クラスタの名前です。

上記のいずれの接続の場合も、以下は共通です。

method

認証方法です。値は、plain、kerberos、maprsasl のいずれかです。

id

認証方法が maprsasl の場合に、接続するユーザ ID を指定します。

トラブルシューティング

サーバが接続を構成できない場合、ダイアログボックスにエラーメッセージが表示されます。以下はメッセージの開始部分です。

```
(FOC1400) SQLCODE IS -1 (HEX:FFFFFFFF) XOPEN: nnnn
```

説明

nnnn

返されるメッセージ番号です。

一般的なエラーの例です。

```
(FOC1500) : (-1) [00000] JDBFOC>> connectx():  
(FOC1500) : java.lang.UnsupportedClassVersionError:  
org/apache/drill/jdbc/Driver :  
(FOC1500) :Unsupported major.minor version 51.0  
(FOC1479) ERROR CONNECTING TO SQL DATABASE
```

上記のメッセージは、Java が 1.7 より前のバージョンで、Drill の要件を満たしていないことを示しています。

Apache Drill でのシノニムの作成

シノニムを作成して、サーバからアクセス可能な Hive テーブルの一意の名前またはエイリアスを定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、動作を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが指定した Hive テーブルに基づいて生成されます。

シノニムを作成するには

シノニムを作成するには、事前にアダプタおよび接続を構成しておく必要があります。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

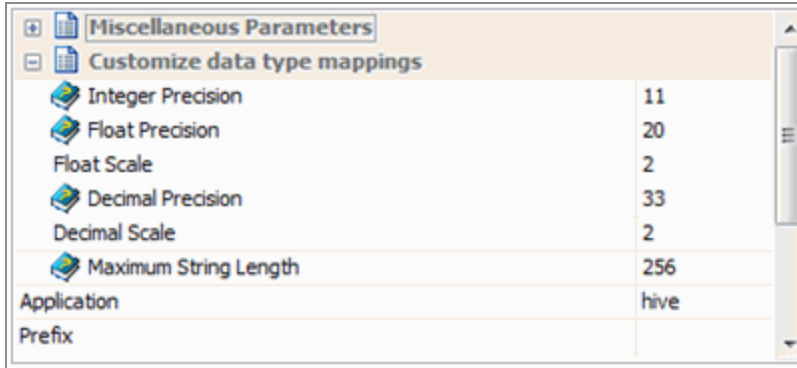
3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

5. 必要に応じて、下図のように、[データタイプマッピングのカスタマイズ] セクションを展開し、文字列の最大長さまたは数値精度の値を入力します。



6. シノニムを作成するオブジェクトのチェックボックスを選択します。シノニムの名前をデフォルト名から変更する場合は、名前をクリックし、必要に応じて編集します。
7. [シノニムの作成] をクリックします。

結果

[ステータス] ウィンドウに、シノニムが正しく作成されたことを示す情報が表示されます。作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

Apache Drill のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

Db2/Db2 Warehouse アダプタの使用

Db2 アダプタの構成方法について説明します。

Db2/Db2 Warehouse アダプタを使用すると、アプリケーションから Db2 データソースへのアクセスが可能になります。このアダプタにより、データまたはアプリケーションリクエストがネイティブの Db2 ステートメントに変換され、最適化されたアンサーセットがリクエストの送信元プログラムに返されます。

このアダプタは、Db2 ストアドプロシジャおよび Db2 Cube Views の実行をサポートします。

Db2 環境の準備

Db2 アダプタの CLI バージョンには、Db2 クライアントのインストールが必要です。CAF バージョンでは、Db2 ロードライブラリが必要です。JDBC バージョンでは、jar ファイルをインストールする必要があります。Db2 アダプタのすべてのバージョンで、ローカルまたはリモートの Db2 サーバへの接続が可能です。

注意：Db2 アダプタの CAF バージョンには DBRM が付属しており、リリース番号と Gen 番号 (例、9.0.2) でバージョン管理されています。Db2 PLAN で DBRM をバインドするために独自の JCL を使用する場合は、それに応じて JCL を調整する必要がある場合があります。

Windows で環境を設定するには

この環境では、Db2 アダプタの 2 つのバージョンが使用できます。WebFOCUS Reporting Server ブラウザインターフェースを使用するには、それらを事前に構成する必要があります。

Db2 CLI

このアダプタで、Unicode データがサポートされます。次のいずれかの方法で、Windows 環境に DB2CODEPAGE=1208 を追加する必要があります。

- Windows で環境変数を設定する。

- edaenv.cfg ファイルで環境変数を設定する。

```
DB2CODEPAGE=1208
```

Db2 JDBC

サーバを起動する前に、JDBC ドライバファイルのパスを CLASSPATH 環境変数に追加することで、これらのファイルの格納先を識別します。たとえば、次のように設定します。

```
CLASSPATH=c:¥usr¥driver_files¥mydriver.jar:%CLASSPATH%
```

UNIX で環境を設定するには

この環境では、Db2 アダプタの 2 つのバージョンが使用できます。WebFOCUS Reporting Server ブラウザインターフェースを使用するには、それらを事前に構成する必要があります。

Db2 CLI

- UNIX の環境変数 \$DB2INSTANCE を使用して、アクセスする Db2 インスタンスを指定します。以下はその例です。

```
export DB2INSTANCE=db2
```

- UNIX の環境変数 \$INSTHOME を使用して、アクセスする Db2 インスタンスのロケーションを指定します。以下はその例です。

```
export INSTHOME=/usr/db2
```

- UNIX の環境変数 \$LD_LIBRARY_PATH を使用して、Db2 の共有ライブラリへのパスを指定します。以下はその例です。

```
export LD_LIBRARY_PATH=$INSTHOME/sql/lib/lib:$LD_LIBRARY_PATH
```

セキュリティを ON に設定してサーバを実行している場合、LD_LIBRARY_PATH 変数は無視されます。この場合は、IBI_LIBPATH を使用する必要があります。

- **Unicode サポート** このアダプタで、Unicode データがサポートされます。edastart ファイルまたは別のシェルフファイルで LANG 環境変数を設定しておく必要があります。

す。たとえば、英語 (米国) の場合は、次の変数をエクスポートします。

```
export LANG=EN_US.UTF-8
```

アダプタの Unicode データサポートを利用するには、WebFOCUS Reporting Server の構成でコードページ 65001 を設定する必要があります。

Db2 JDBC

サーバを起動する前に、JDBC ドライバファイルのパスを CLASSPATH 環境変数に追加することで、これらのファイルの格納先を識別します。たとえば、次のコマンドを発行します。

```
CLASSPATH=/usr/driver_files/mydriver.jar;
$CLASSPATH
export CLASSPATH
...
```

Db2 アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

アダプタは、Db2 データベースサーバに接続する際に、接続情報および認証情報を必要とします。この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。コンソールから、選択したプロファイルにこのコマンドが追加されます。選択可能なプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) があります。
- グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) のいずれかに手動でコマンドを追加します。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の Db2 データベースへの接続を宣言することができます。Db2 データベースサーバへの実際の接続は、そ

の接続を参照する最初のクエリの発行時に確立されます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

接続属性を定義するには

アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで構成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
または
ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。
[アダプタ] フォルダが開きます。
2. [サーバのデータソース] フォルダを展開するか、プラス記号 (+) をクリックします (展開されていない場合)。
3. 適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタの「接続属性」の説明に従って、アダプタで必須のパラメータ値を入力します。
6. [構成] または [追加] をクリックします。構成したアダプタが、ナビゲーションウィンドウの [アダプタ] リストに追加されます。

CLI での Db2 の接続属性

Db2 アダプタは [SQL] グループフォルダ内にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

データソース/DSN

Db2 データソース名。ローカルマシンに登録されたリモート Db2 データベースのエイリアスです。デフォルトのデータソース名はありません。値を入力する必要があります。

IBM i では、リモートデータベースディレクトリエントリまたは *LOCAL (ローカルホストの場合) です。

DSN を使用しない

Linux、UNIX、Windows に適用されます。このチェックをオンにすると、DSN の代わりに次の接続パラメータを指定することができます。このタイプの接続では、バルクロードはサポートされません。

ホスト名

Db2 サーバが稼働しているホスト名です。

ポート番号

Db2 サーバのリスナポート番号です。

データベース名

Db2 のデータベース名です。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 3 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。
- **Trusted** アダプタは、現在のオペレーティングシステムに関連している偽装プロ

セス用のデータベースの規則を使用してデータベースに接続します。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

信頼済みコンテキスト

[有効にする] のチェックをオンにすると、Db2 トラステッドコンテキストが有効になります。Db2 トラステッドコンテキストは、データベースと外部 Application Server との接続の認証関係を定義するデータベースオブジェクトです。認証関係は、トラステッドコンテキストオブジェクトで定義された属性に基づきます。

[有効にする] を選択すると、SET CONNECTION ATTRIBUTES コマンドのパスワードの後に次の記述が追加されます。

```
: 'trusted_context=y'
```

たとえば、WebFOCUS Reporting Server が LDAP セキュリティで構成され、ibi™ WebFOCUS® Client が SECURITY TRUSTED で構成されている場合を想定します。WebFOCUS® Client から、認証情報の要求がされずに WebFOCUS Reporting Server ブラウザインターフェースが開き、サーバへの接続が認証されます。アダプタは、接続文字列のユーザ ID を使用して Db2 データベースに接続します。Trusted コンテキストオブジェクトの確認後に、ユーザを切り替えられます。

追加の接続文字列キーワード (オプション)

追加の接続文字列キーワードは、セミコロン (;) 区切りで入力することができます。各キーワードは、属性と値の組み合わせで、attribute=value 形式で構成されます。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

JDBC での Db2 の接続属性

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

URL

JDBC データソースのロケーション URL です。

ドライバ名

JDBC ドライバ名です。

詳細は、使用しているバージョンのドライバのマニュアルを参照してください。

IBI_CLASSPATH

Java サービスで使用する追加の Java クラスディレクトリ、または jar ファイル名のフルパスを定義します。値を設定するには、通信ファイルを編集するか、WebFOCUS Reporting Server ブラウザインターフェースを使用します。WebFOCUS Reporting Server ブラウザインターフェースの入力ボックスでは、1 行につき 1 つの参照を入力することができます。すべてのプラットフォームでは、ファイルを保存した際に、複数のエントリがコロン (:) で区切られた単一文字列に変換されます。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、`CONNECTION_ATTRIBUTES` コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの `edasprof.prf` です。

新しいユーザプロファイル (`user.prf`) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (`edasprof`) に保存します。

接続属性の手動宣言

Explicit 認証 ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報が Db2 データソースへの接続時に認証情報として渡されます。

```
ENGINE DB2 SET CONNECTION_ATTRIBUTES connection DSN_name/userid,password
```

Password Passthru 認証 ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報が Db2 データソースへの接続時に認証情報として渡されます。

```
ENGINE DB2 SET CONNECTION_ATTRIBUTES connection DSN_name/
```

Trusted 認証 アダプタは、サーバのデータアクセスエージェントによって偽装された Windows ユーザの認証情報を使用して、Windows ログインとして Db2 に接続します。

使用可能なパラメータは次のとおりです。

```
ENGINE DB2 SET CONNECTION_ATTRIBUTES connection DSN_name/,
```

説明

DB2

アダプタを指定します。SET `SQLENGINE` コマンドをすでに発行している場合は、この値を省略することができます。

connection

この接続属性のセットの識別に使用される論理名です。

`connection` と `DSN_name` の間にスペースが 1 つ必要です。

DSN_name

アクセスする Db2 データソース名 (DSN) です。この名前は、odbc.ini ファイルのエントリに一致させる必要があります。

userid

Db2 に登録されたプライマリ認証 ID です。

password

プライマリ認証 ID に関連するパスワードです。

接続属性の宣言

以下の SET CONNECTION_ATTRIBUTES コマンドを実行すると、アプリケーションが明示的なユーザ ID (MYUSER) およびパスワード (PASS) を使用して、SAMPLESERVER という名前の Db2 データベースサーバにアクセスすることができます。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

```
ENGINE DB2 SET CONNECTION_ATTRIBUTES CON01 SAMPLESERVER/MYUSER,PASS
```

以下の SET CONNECTION_ATTRIBUTES コマンドは、Password Passthru 認証を使用して Db2 データベースサーバ SAMPLESERVER に接続します。

```
ENGINE DB2 SET CONNECTION_ATTRIBUTES CON01 SAMPLESERVER/
```

以下の SET CONNECTION_ATTRIBUTES コマンドは、オペレーティングシステムの認証を使用してローカルの Db2 データベースサーバに接続します。

```
ENGINE DB2 SET CONNECTION_ATTRIBUTES CON01 SAMPLESERVER/,
```

接続情報の更新

このアダプタの CONNECTION_ATTRIBUTES コマンドの構文が拡張され、アプリケーションの開発環境から実稼働環境への移植をサポートするよう設計された論理接続名を指定できるようになりました。この拡張された構文では、既存の CONNECTION_ATTRIBUTES コマンド

のマイグレートが必要になる場合があります。

WebFOCUS Reporting Server ブラウザインターフェースの [マイグレート] オプションを使用すると、現在のサーバ設定が新しいバージョンにマイグレートされます。このオプションにアクセスするには、[ワークスペース] ページから、[設定]、[マイグレート] を順に選択します。[マイグレート] ウィンドウで、構成インスタンスディレクトリ (EDACONF) の完全なパスを入力して [続行] ボタンをクリックします。この方法を使用することをお勧めします。

[マイグレート] オプションを使用しない場合、以下の情報に注意してください。

- バージョン 7.6 SP01 より前のバージョンで宣言された接続名は引き続きサポートされます。
- 新規のシノニムを作成する目的で新しい接続を追加する場合、既存の接続は新しいフォーマットで再保存され、また既存のシノニムは変更を行うことなく引き続き機能します。
- 既存のシノニムを使用する目的で新しい接続を追加する場合、デフォルトの論理接続名を、既存のアクセスファイルの CONNECTION=value 属性で指定された値に一致するように変更する必要があります。

たとえば、バージョン 7.6 SP01 より前のバージョンで、以下のように接続が定義されている場合を想定します。

```
ENGINE DB2 SET CONNECTION_ATTRIBUTES DSN_A/uid,pwd
```

この DSN_A に保存されているオブジェクトに基づいてシノニムを作成すると、アクセスファイルには以下の記述が追加されます。

```
CONNECTION=DSN_A
```

この後、新しい接続を追加する際に、この既存のシノニムを再利用するには、接続名をデフォルトの CON01 から DSN_A に変更し、DSN_A として保存する必要があります。この接続は、以下のようにプロファイルに保存されます。

```
ENGINE DB2 SET CONNECTION_ATTRIBUTES DSN_A DSN_A/uid,pwd
```

デフォルト接続の上書き変更

複数の接続を定義した場合、最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続がデフォルト接続として機能します。このデフォルト値は、SET DEFAULT_CONNECTION コマンドを使用して上書きすることができます。

デフォルト接続の変更

```
ENGINE DB2 SET DEFAULT_CONNECTION connection
```

説明

DB2

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

```
ENGINE DB2 SET DEFAULT_CONNECTION SAMPLE
```


接続範囲の制御

アダプタを使用する際に接続の継続を制御するには、SET AUTODISCONNECT コマンドを使用します。

AUTODISCONNECT による接続範囲の制御

AUTODISCONNECT は、ユーザのアドレス空間 (タスク) を Db2 から完全に切断します。DISCONNECT の実行後、タスクはデータソース処理を実行する前に Db2 との接続を再度確立する必要があります。DISCONNECT コマンドを頻繁に実行するタスクは Db2 との接続時間が短くなるため、他のタスクは必要に応じてスレッドに接続することが可能です。ただし、接続と切断を頻繁に繰り返すことで過大なシステムオーバーヘッドが発生したり、タスクが再接続を試行した際にスレッドを即座に使用できない可能性が発生したりします。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

AUTODISCONNECT による接続範囲の制御

```
ENGINE DB2 SET AUTODISCONNECT ON {FIN|COMMIT}
```

説明

DB2

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

Db2 メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および Db2 データタイプのサーバマッピングを定義するシノニムを作成します。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な Db2 テーブルまたはビューの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

ストアドプロシジャのシノニムの作成、およびストアドプロシジャに対するレポートの実行についての詳細は、[ストアドプロシジャのシノニムの作成](#)を参照してください。

シノニムを作成するには

シノニムを作成するには、事前にアダプタを構成しておく必要があります。シノニムは、WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで作成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。
選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオブ

ションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Db2 のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

- フィルタオプションとして [Aliases] を選択し、エイリアスがリモート Db2 システムを指す場合、バージョン 8 以降である必要があります。サポートされるリモート Db2 のレベルは 1 つだけです。
- Db2 がニックネームオブジェクトタイプをサポートするプラットフォーム (Windows および UNIX) で、フィルタオプションとして [Nicknames] を選択すると、このタイプの Db2 オブジェクトに対して基本シノニムが作成され、タイトル、備考、キーの情報は記録されません。シノニムを作成するユーザのユーザ ID には、このオブジェクトタイプに対する SELECT 権限が付与されている必要があります。

重要：オブジェクトタイプにストアードプロシージャを選択した場合、入力パラメータはここで説明するものとは多少異なります。詳細は、[Db2 のストアードプロシージャに対するレポートの実行](#)を参照してください。

オーナー/スキーマ

このオプションを選択すると、所有者およびオブジェクト名のパラメータが画面に追加されます。ドロップダウンリストから [オーナー/スキーマ] を選択するか、選択のフィルタに使用する文字列を入力します。必要に応じて、文字列の先頭や末尾にワイルドカード文字 (%) を入力します。[検索] をクリックします。たとえば、「ABC%」と入力すると、オーナー/スキーマが文字列 ABC で始まるテーブルまたはビューが選択され、「%ABC」と入力すると、オーナー/スキーマが文字列 ABC で終わるテーブルまたはビューが選択されます。「%ABC%」と入力すると、文字列 ABC を先頭、中間、または末尾に含むオーナー/スキーマのテーブルまたはビューが選択されます。

オブジェクト名

このオプションを選択すると、所有者およびオブジェクト名のパラメータが画面に追加されます。オブジェクト名のフィルタに使用する文字列を入力します。必要に応じて、文字列の先頭や末尾にワイルドカード文字 (%) を入力します。[検索] をクリックします。たとえば、「ABC%」と入力して、名前が ABC で始まるオブジェクトをすべて選択します。

「%ABC」と入力して、名前が ABC で終わるオブジェクトをすべて選択します。

「%ABC%」と入力して、名前の先頭、中間、または末尾に ABC が含まれるオブジェクトをすべて選択します。

外部 SQL スクリプトのパス

[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合は、さらに次の項目が表示されます。

次の標準名前規則は UNIX に適用されます。

- [ベースパス] テキストボックスに、SQL クエリが記述されたファイルの物理ディレクトリパスを指定します。ディレクトリ名を直接入力するか、[...] (参照) ボタンをクリックしてファイルの格納先ディレクトリへ移動します。このクリックにより、[ベースパスの選択] ダイアログボックスが表示されます。
- [ドキュメント名] テキストボックスに、ファイル名を入力します。必要に応じて、ワイルドカード文字を使用します。
- [ドキュメント拡張子] テキストボックスに、スクリプトファイルの拡張子を入力して、その拡張子のファイルのみを表示します。

アダプタはシノニムの作成中にネイティブ API コールを発行して選択リスト内の要素のリストを取得し、各要素に対するフィールドを持つマスターファイルを構成します。生成されたアクセスファイルは、DATASET 属性の SQL スクリプトパスを参照します。これには SQL クエリが格納されたファイルの名前と拡張子を含む、フルパスが記述されています。以下はその例です。

```
DATASET=/u1/home2/apps/report3.sql
```

WebFOCUS レポートを作成する際は、SQL クエリを使用してデータにアクセスします。

最大行数

[シノニムの作成] ページに表示するオブジェクト数を指定します。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (:) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[データタイプサポートレポート](#)を参照してください。

クラスタシノニムまたはベースシノニムの作成

作成するシノニムタイプのボタンを選択します。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

オーナー/スキーマ

ユーザが所有する 1 つまたは複数のオブジェクトを作成したユーザアカウントです。

ファクトまたはディメンション

[ファクト] または [ディメンション] のチェックをオンにして、セグメントをファクトテーブルまたはディメンションセグメントとして生成することができます。クラスタシノニムを作成する場合は、選択したファクトテーブルを右クリックし、[関連するディメンションの表示] または [関連するディメンションの追加] を選択して、関連するディメンションのリストを表示したり、関連するディメンションをシノニムに追加したりできます。

テーブル名

基になるオブジェクトの名前です。

タイプ

オブジェクトタイプ (テーブル、ビューなど)。

テーブルの選択

シノニムを作成するテーブルを選択します。

- ベースシノニムを作成する場合、[すべて選択] のチェックをオンにすることで、リスト内のすべてのテーブルを選択することができます。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

シノニムを作成するオブジェクトの選択後、リボンの [ベースシノニムの作成] または [クラスタシノニムの作成] ボタンをクリックします。

生成されたシノニムの例

Db2 アダプタのシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

生成されたマスターファイル nf29004.mas

```
FILE=DIVISION, SUFFIX=DB2 , $
SEGNAME=SEG1_4, SEGTYPE=S0 , $
FIELD=DIVISION4, DIVISION4, I9, I4 ,MISSING=OFF , $
FIELD=DIVISION_NA4, DIVISION_NA4, A25, A25 ,MISSING=ON , $
FIELD=DIVISION_HE4, DIVISION_HE4, I9, I4 ,MISSING=ON , $
```


生成されたアクセスファイル nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=EDAQA.NF29004,
CONNECTION=DB1, KEYS=1, WRITE=YES, $
```

Db2 テーブルとカラムの属性のシノニムへのマッピング

これらのマッピングは OS 固有になります。

Db2 テーブル属性のマッピング

プラットフォーム	Db2 テーブル属性	シノニムでのマッピング	説明
UNIX/Windows	COMMENT	REMARKS	

Db2 カラム属性のマッピング

プラットフォーム	Db2 カラム属性	シノニムでのマッピング	説明
UNIX/Windows	COMMENT	DESCRIPTION	

アクセスファイルのキーワード

下表は、アクセスファイルで使用されるキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABlename	テーブルまたはビューの名前です。この値には、以下のようにロケーションまたはオーナー名を含めることができます。 <pre>TABlename=[location.][owner.]tablename</pre> 注意：ロケーションは、Db2 CAF にのみ有効で、サブシステムのロケーション名を指定します。
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre> CONNECTION=' ' は、ローカルのデータベースサーバへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
DBSPACE	テーブルの記憶域を示すオプションのキーワードです。以下はその例です。 <pre>datasource.tablespace DATABASE datasource</pre>
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。この属性を使用するには、キーを構成するフィールドをマスターファイルの先頭に記述する必要があります。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属

キーワード	説明
	性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。 <pre>KEY=fld1/fld2/.../fldn</pre>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。
KEYFLD IXFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定します。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 - KEYFLD は、親テーブルの共通カラムの FIELDNAME です。 - IXFLD は関連するテーブルの共通カラムの FIELDNAME です。 KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることをお勧めしますが、必須ではありません。 注意：KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。
AUTO INCREMENT	自動インクリメントを有効にするには [Yes] に設定します。

キーワード	説明
START	インクリメントの順序の初期値です。
INCREMENT	インクリメントの間隔です。
INDEX_NAME INDEX_UNIQUE INDEX_COLUMNS INDEX_ORDER	データベース内のインデックス名、およびインデックスフィールドの一意性、名前、順序を示します。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

Db2 データタイプのサポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[アクセスファイル](#)を参照してください。

可変長データタイプのマッピングの制御

SET パラメータ VARCHAR は、Db2 データタイプ VARCHAR のマッピングを制御します。デフォルト設定では、サーバはこのデータタイプを可変長の文字 (AnV) としてマッピングします。

以下の表は、VARCHAR の値に基づくデータタイプのマッピングのリストです。

Db2 データ タイプ	備考	VARCHAR ON		VARCHAR OFF	
		USAGE	ACTUAL	USAGE	ACTUAL
VARCHAR (<i>n</i>)	<i>n</i> は 1 から 32768 までの整数です。	<i>AnV</i>	<i>AnV</i>	<i>An</i>	<i>An</i>

可変長データタイプのマッピングの制御

```
ENGINE DB2 SET VARCHAR {ON|OFF}
```

説明

DB2

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

Db2 データタイプ VARCHAR を可変長文字データタイプ (*AnV*) としてマッピングします。Unicode 環境ではこれは必須です。デフォルト値は ON です。

OFF

Db2 データタイプ VARCHAR を文字データタイプ (A) としてマッピングします。

SQL 式内での末尾ブランク

TABLE アダプタ内の新しい SQL ジェネレータでは、デフォルトで文字列リテラル内の末尾ブランク、および数値リテラル内の小数部と指数の表記を含むリテラルコンテンツを保持します。これにより、生成される SQL に対する制御が向上します。

末尾ブランクが不要なケースでは、次の構文が下位互換性を確保するために使用できます。

```
ENGINE DB2 SET TRIM_LITERALS ON
```

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアードプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

Db2 のストアードプロシジャに対するレポートの実行

SELECT ステートメントや TABLE コマンドなどのレポートツールを使用して、Db2 のストアードプロシジャを実行して、プロシジャの出力パラメータおよびアンサーセットに対するレポートを作成することができます。ストアードプロシジャの実行という方法には、次の利点があります。

- アンサーセット以外に、出力パラメータである OUT パラメータおよび OUT モードの INOUT パラメータを検索することができる。他の呼び出し方法では、アンサーセットのみが検索されます。
- TABLE および他のレポートツールを使用して、出力パラメータおよびアンサーセットの処理、フォーマット設定、表示を簡単に行うことができる。

ストアードプロシジャに対するレポートを実行するには、次の手順を実行します。

1. **シノニムの作成** ストアードプロシジャのアンサーセットのシノニムを作成します。詳細は、[ストアードプロシジャのシノニムの作成](#)を参照してください。
2. **レポートプロシジャの作成** レポートプロシジャを作成します。詳細は、[ストアードプロシジャに対するレポートの実行](#)を参照してください。

3. **レポートの実行** ストアドプロシジャを実行し、出力パラメータ (OUT および OUT モードの INOUT) およびレポートに指定されたアンサーセットフィールドに対するレポートを実行します。

ストアドプロシジャのシノニムの作成

シノニムは、ストアドプロシジャのパラメータとアンサーセットを記述します。

アンサーセットの構造は、プロシジャの実行時に指定する入力パラメータ値により異なります。このため、プロシジャの実行時に指定される入力パラメータ値のセットごとに、個別のシノニムを作成する必要があります。たとえば、3つの異なる入力パラメータ値セットでストアドプロシジャを実行する場合、シノニムを3つ (値セットごとに1つ) 作成する必要があります。なお、特に明記していない限り、「入力パラメータ」は、IN パラメータおよび IN モードの INOUT パラメータを意味します。

例外 ユーザがプロシジャの内部ロジックを理解しており、入力パラメータの値の範囲とプロシジャによって返された各アンサーセットの構造の関係を把握している場合、各アンサーセットの構造に対応する単一のシノニムを作成して、各シノニムにアンサーセットの構造を返すために必要な入力パラメータの値のセットを指定することが可能です。

シノニムには、次のセグメントを含めます。

- **INPUT** - このセグメントは、IN パラメータおよび IN モードの INOUT パラメータを記述します。
IN パラメータおよび IN モードの INOUT パラメータがない場合、このセグメントはダミーフィールドを1つ記述します。
- **OUTPUT** - このセグメントは、OUT パラメータおよび OUT モードの INOUT パラメータを記述します。
OUT パラメータおよび OUT モードの INOUT パラメータがない場合、このセグメントは省略されます。
- **ANSWERSET n** - 各アンサーセットに1つ含まれます。
アンサーセットがない場合、このセグメントは省略されます。

Db2 のストアドプロシジャ CustOrders のシノニム

以下のシノニムは、IN/OUT OUTPUT パラメータを含むストアドプロシジャ DB2SPR04 SP を定義します。

```
FILENAME=SDB2SPR04, SUFFIX=DB2      , $
SEGMENT=INPUT, SEGTYPE=S0, $
  FIELDNAME=PARM1, ALIAS=P0001, USAGE=I11, ACTUAL=I4,
    MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=PARM2, ALIAS=P0002, USAGE=I6, ACTUAL=I4,
    MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=PARM5, ALIAS=P0005, USAGE=YYMD, ACTUAL=DATE,
    MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=PARM6, ALIAS=P0006, USAGE=HHIS, ACTUAL=HHIS,
    MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=PARM7, ALIAS=P0007, USAGE=HYMDm, ACTUAL=HYMDm,
    MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=PARM11, ALIAS=P0011, USAGE=P17.5, ACTUAL=P8,
    MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=PARM13, ALIAS=P0013, USAGE=P20, ACTUAL=P10,
    MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=PARM15, ALIAS=P0015, USAGE=F9.2, ACTUAL=F4,
    MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=PARM16, ALIAS=P0016, USAGE=D20.2, ACTUAL=D8,
    MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
```

```
SEGMENT=OUTPUT, SEGTYPE=S0, PARENT=INPUT, $
  FIELDNAME=RETURN_CODE, ALIAS=P0000, USAGE=I11, ACTUAL=I4,
    MISSING=ON, TITLE='Return Code', $
  FIELDNAME=PARM1, ALIAS=P0001, USAGE=I11, ACTUAL=I4, MISSING=ON, $
  FIELDNAME=PARM2, ALIAS=P0002, USAGE=I6, ACTUAL=I4, MISSING=ON, $
  FIELDNAME=PARM3, ALIAS=P0003, USAGE=I11, ACTUAL=I4, MISSING=ON, $
  FIELDNAME=PARM4, ALIAS=P0004, USAGE=I6, ACTUAL=I4, MISSING=ON, $
  FIELDNAME=PARM5, ALIAS=P0005, USAGE=YYMD, ACTUAL=DATE, MISSING=ON, $
  FIELDNAME=PARM6, ALIAS=P0006, USAGE=HHIS, ACTUAL=HHIS,
    MISSING=ON, $
  FIELDNAME=PARM7, ALIAS=P0007, USAGE=HYMDm, ACTUAL=HYMDm,
    MISSING=ON, $
  FIELDNAME=PARM8, ALIAS=P0008, USAGE=YYMD, ACTUAL=DATE, MISSING=ON, $
  FIELDNAME=PARM9, ALIAS=P0009, USAGE=HHIS, ACTUAL=HHIS, MISSING=ON, $
  FIELDNAME=PARM10, ALIAS=P0010, USAGE=HYMDm, ACTUAL=HYMDm,
```

```
MISSING=ON, $
FIELDNAME=PARM11, ALIAS=P0011, USAGE=P17.5, ACTUAL=P8, MISSING=ON, $
FIELDNAME=PARM12, ALIAS=P0012, USAGE=P17.5, ACTUAL=P8, MISSING=ON, $
FIELDNAME=PARM13, ALIAS=P0013, USAGE=P20, ACTUAL=P10, MISSING=ON, $
FIELDNAME=PARM14, ALIAS=P0014, USAGE=P20, ACTUAL=P10, MISSING=ON, $
FIELDNAME=PARM15, ALIAS=P0015, USAGE=F9.2, ACTUAL=F4, MISSING=ON, $
FIELDNAME=PARM16, ALIAS=P0016, USAGE=D20.2, ACTUAL=D8, MISSING=ON, $
FIELDNAME=PARM17, ALIAS=P0017, USAGE=F9.2, ACTUAL=F4, MISSING=ON, $
FIELDNAME=PARM18, ALIAS=P0018, USAGE=D20.2, ACTUAL=D8, MISSING=ON, $
```

ストアドプロシジャのシノニム作成パラメータ

以下は、ストアドプロシジャのシノニム作成パラメータを示しています。

オブジェクトタイプ

[ストアドプロシジャ] を選択します。

オーナー/スキーマおよびオブジェクト名

このオプションを選択すると、オーナーおよびオブジェクト名のパラメータが画面に追加されます。

- オーナー/スキーマ** 選択のフィルタに使用する文字列を入力します。必要に応じて、文字列の先頭や末尾にワイルドカード文字 (%) を入力します。たとえば、「ABC%」と入力すると、オーナー/スキーマが文字列 ABC で始まるテーブルまたはビューが選択され、「%ABC」と入力すると、オーナー/スキーマが文字列 ABC で終わるテーブルまたはビューが選択されます。「%ABC%」と入力すると、文字列 ABC を先頭、中間、または末尾に含むオーナー/スキーマのテーブルまたはビューが選択されます。
- オブジェクト名** プロシジャ名のフィルタに使用する文字列を入力します。必要に応じて、文字列の先頭や末尾にワイルドカード文字 (%) を入力します。たとえば、「ABC%」と入力して、名前が ABC で始まるプロシジャをすべて選択します。「%ABC」と入力して、名前が ABC で終わるプロシジャをすべて選択します。「%ABC%」と入力して、名前の先頭、中間、または末尾に ABC が含まれるプロシジャをすべて選択します。

選択

プロシジャを選択します。次のシノニム作成ウィンドウの [値] テキストボックスで、プロシジャごとに固有の値を入力する必要があるため、1 回に選択できるプロシジャは、1 つだけです。

シノニム名

シノニム名です。デフォルト設定は、ストアードプロシジャ名です。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のストアードプロシジャが存在する場合は、接頭語または接尾語を割り当てて、対応するシノニムを区別します。なお、シノニム名の最大文字数は 64 バイトです。

すべてのプロシジャに固有の名前が付いている場合は、[接頭語] および [接尾語] のテキストボックスはブランクにしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

ストアードプロシジャに対するレポートの実行

ストアードプロシジャのアンサーセットに対してレポートを実行する場合は、データベーステーブルに対してレポートを実行する際と同一の機能を使用することができます。

- **SQL SELECT ステートメント** 構文についての詳細は、[SELECT を使用したストアードプロシジャに対するレポートの実行](#)を参照してください。
- **TABLE および GRAPH コマンド** 構文についての詳細は、[TABLE コマンドを使用したストアードプロシジャに対するレポートの実行](#)を参照してください。

ストアードプロシジャのアンサーセットとの JOIN には、次の 2 つのファイルを使用します。

- **ホストファイル** OUTPUT および ANSWERSET セグメントのみです。
- **クロスリファレンス** ファイル内の INPUT セグメントのみです。

TABLE コマンドを使用したストアドプロシジャに対するレポートの実行

TABLE コマンドを使用してストアドプロシジャを実行するには、次の構文を使用します。

```
TABLE FILE synonym
PRINT [parameter [parameter] ... | *]
[IF in-parameter EQ value]
.
.
.
END
```

説明

synonym

実行するストアドプロシジャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアドプロシジャにパラメータを必要としない場合は、アスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

★

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

IF

IF または WHERE キーワードです。IN パラメータまたは IN モードの INOUT パラメータに値を渡す際に使用します。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

注意：アダプタが Unicode をサポートするよう構成されている場合は、IN パラメータの長さは 1000 バイト以内にする必要があります。

value

パラメータに渡す値です。

SELECT を使用したストアードプロシジャに対するレポートの実行

```
SQL
SELECT [parameter [,parameter] ... | *] FROM synonym
[WHERE in-parameter = value]
.
.
.
END
```

説明**synonym**

実行するストアードプロシジャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアードプロシジャにパラメータを必要としない場合は、構文にアスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

★

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

WHERE

IN パラメータまたは IN モードの INOUT パラメータに値を指定する場合に使用します。各パラメータの値は、それぞれ異なる行に指定する必要があります。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

value

パラメータに渡す値です。

Db2 環境のカスタマイズ

Db2 アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。ここでは、カスタマイズオプションの概要について説明します。

デフォルトのテーブルスペースの指定

SET DBSPACE コマンドを使用して、作成するテーブルのデフォルトのテーブルスペースを指定することができます。セッション中、アダプタはこれらのテーブルを SET DBSPACE コマンドによって識別される Db2 テーブルスペースに格納します。SET DBSPACE コマンドを使用しない場合、Db2 は接続ユーザのデフォルトのテーブルスペースを使用します。

テーブルのデフォルト記憶域の指定

```
ENGINE DB2 SET DBSPACE {datasource.tablespace|DATABASE datasource}
```

説明**DB2**

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

datasource

データソースの名前です。デフォルト値は、パブリックデータソースの DSNDDB04 です。

tablespace

データソース内の有効なテーブルスペース名です。

注意：このコマンドはテーブルサービスによって実行された CREATE FILE リクエストにのみ影響します。パススルーの CREATE TABLE コマンドには影響しません。

Linux、UNIX、Windows では、ユーザが作成したテーブルスペースに揮発テーブルまたはグローバル一時テーブルを追加する場合は、次のコマンドを発行します。

```
ENGINE DB2 SET DBSPACE usertemp
```

説明

usertemp

ユーザの一時テーブルスペースの名前です。

注意：この設定が有効な場合、セッション中に作成されたテーブルはすべて、このテーブルスペースに自動的に作成されます。ただし、永続テーブルをこのタイプのテーブルスペースに追加することはできません。この場合、エラーが発生します。そのため、この設定はプロファイルに追加しないでください。一時テーブルを作成するプロシジャには追加することができます。プロシジャの実行後、この設定は無効になります。

ロックのタイプの制御

SET ISOLATION コマンドを使用して、アダプタが作成したトランザクションの分離レベルを指定することができます。分離レベルは、トランザクション内で実行されたリクエストが参照するオブジェクトのロックのタイプを制御します。

ロックタイプの制御

```
ENGINE DB2 SET ISOLATION level
```

説明

DB2

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

level

IBM RDBMS の分離レベルにマッピングされる、Db2 の分離レベルを設定します。分離レベルを指定しない場合、アダプタのデフォルト設定にリセットされます。

Db2 の分離レベル (CLI)	IBM RDBMS の分離レベル
RC (SQL_TXN_READ_COMMITTED)	CS (カーソル固定)
SE (SQL_TXN_SERIALIZABLE_READ)	RR (反復可能読み取り)
RR (SQL_TXN_REPEATABLE_READ)	RS (読み取り固定)
RU (SQL_TXN_READ_UNCOMMITTED)	UR (非コミット読み取り)
NC (SQL_TXN_NO_COMMIT)	使用不可

- **RC** - カーソルがテーブル上を移動している間、共有ロックを解除します。読み取り専用のリクエストで使用します。デフォルト値は、RC です。CS (カーソル固定) にマッピングされます。
- **SE** - SQL COMMIT WORK または SQL ROLLBACK WORK ステートメントによって解放されるまで、検索されたデータをロックします。RR (反復可能読み取り) にマッピングされます。
- **RR** - RS (読み取り固定) にマッピングされます。詳細は、該当する Db2 のマニュアルを参照してください。
- **RU** - ロックされている場合でも、レコードの読み取り専用アクセスを提供します。ただし、これらのレコードはデータソースにコミットされません。UR (非コミット読み取り) にマッピングされます。
- **NC** - SQL ステートメントに対して、コミットおよびロールバック処理は影響しません。変更はその処理の正常終了後にコミットされ、即座にアクセス可能となります。

注意

- アダプタは分離レベルの値を検査しません。無効な値を指定して SET ISOLATION コマ

ンドを実行した場合も、アダプタはメッセージを返しません。

- 分離レベルの設定を表示するには、ENGINE [DB2] ?CONNECTINFO コマンドを実行します。

インデックススペースのデフォルトのパラメータの上書き変更

SET IXSPACE コマンドを使用して、CREATE FILE および HOLD FORMAT コマンドによって暗黙的に作成されたインデックススペースのデフォルトのパラメータを上書き変更することができます。

IXSPACE の設定

```
ENGINE DB2 SET IXSPACE [index-spec]
```

説明

DB2

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

index-spec

インデックスのパラメータを定義する DB2 CREATE INDEX ステートメントの一部です。94 バイト以内の有効な Oracle インデックススペースパラメータで構成することができます。インデックススペースパラメータをデフォルト値にリセットするには、パラメータを指定せずに SET IXSPACE コマンドを実行します。

注意：このコマンドの詳細は、Db2 のマニュアルを参照してください。

以下は、複数行にわたるロングフォームの SQL パススルー構文の例です。

```
ENGINE DB2
SET IXSPACE index-spec
END
```

たとえば、DB2 CREATE INDEX ステートメントの NOSORT、NOLOGGING、TABLESPACE の部分を指定するには、以下のコマンドを入力します。

```
ENGINE DB2
SET IXSPACE NOSORT NOLOGGING
TABLESPACE TEMP
END
```

注意：このコマンドは、CREATE FILE および HOLD FORMAT DB2 コマンドによって実行された CREATE INDEX リクエストにのみ影響を与えます。パススルーの CREATE INDEX コマンドには影響しません。

```
ENGINE DB2 SET IXSPACE TABLESPACE tablespace_name
TABLE FILE table_name
PRINT *
ON TABLE HOLD AS file_name FORMAT DB2
END
```

NONBLOCK モードの有効化

Db2 アダプタには、NONBLOCK モードで呼び出しを実行する機能があります。デフォルト設定は、BLOCK モードです。

この機能により、アダプタはクライアントリクエストに応答して、エンジン処理の待ち状態からクエリを取り消すことができます。待機状態は、通常、SQL 解析中、アンサーセットの 1 行目がアダプタへ配信できる状態になる前、または別のアプリケーションによりロックされたオブジェクトへのアクセスの待機中に起こります。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

NONBLOCK モードの有効化

```
ENGINE DB2 SET NONBLOCK {0|n}
```

説明

DB2

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

正の数値です。デフォルト値はゼロ (0) です。これは、アダプタが BLOCK モードで動作することを意味します。数値が 1 以上の場合、NONBLOCK の呼び出しが有効になります。この数値は、アダプタが次の項目を確認する際の待機時間間隔 (秒) を指定します。

- 検索が実行されたか。
- クライアントアプリケーションが検索のキャンセルを要求したか。
- WebFOCUS Reporting Server ブラウザインターフェースの [セッションの終了] ボタンが押されたか。

注意：通常の操作には、値に 1 または 2 を指定します。

カラム名の制御

ストアドプロシジャを実行する際、SET NOCOLUMNTITLE コマンドを使用してレポートのカラム名を制御することができます。

カラム名の制御

```
ENGINE DB2 SET NOCOLUMNTITLE {ON|OFF}
```

説明

DB2

アダプタを指定します。このパラメータは、SET SQLENGINE コマンドを発行済みの場合省略できます。

ON

Db2 が返すカラム名の代わりに、作成したカラム名を使用します (例、E01、E02)。

OFF

Db2 が返すカラム名を使用します。デフォルト値は OFF です。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

```
ENGINE DB2 SET PASSRECS {ON|OFF}
```

説明**DB2**

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

注意：この動作は、すべてのプラットフォームで、Db2 バージョン 9 までのすべてのサポート対象バージョンに適用されます。z/OS の Db2 バージョン 9 では、多量の DELETE 処理 (DELETE ステートメントに WHERE 句が指定されていない) に対しては PASSRECS コマンドは行数を返しません。このような状況では、レポートリクエストに WHERE 句を追加すると

この情報を取得することができます。たとえば、次のリクエストが削除数を生成します。

```
SQL DB2
DELETE FROM DB2TAB WHERE F1=15;
END
(FOC1364) パススルーコマンドで影響を受けた行 : 1/DELETE
```

エンドユーザ情報の設定

UNIX および Windows WebFOCUS Reporting Server 環境では、後続の SQL リクエストが処理された際に、Db2 サーバに渡されるエンドユーザ情報に新たな値を設定することができます。次の情報を含めることができます。

- クライアントユーザ ID
- アプリケーションプログラム名
- ワークステーション名
- アカウンティングストリング

SELECT ステートメントを使用して情報を照会することができます。

エンドユーザ情報の設定

```
ENGINE DB2 SET CLIENT_APPLNAME application_name
ENGINE DB2 SET CLIENT_USERID userid
ENGINE DB2 SET CLIENT_WRKSTNNAME workstation
ENGINE DB2 SET CLIENT_ACCTNG account
```

説明

application_name

アプリケーションプログラム名です。

userid

クライアントのユーザ ID です。

workstation

ユーザのワークステーションに関連付けられた名前です。

account

クライアントユーザに関連付けられたアカウントティングストリングです。

エンドユーザ情報の照会

UNIX および Windows WebFOCUS Reporting Server 環境では、Db2 サーバが稼動している場所に応じて以下の構文を使用してエンドユーザ情報を照会することができます。

Db2 サーバが UNIX または Windows 上にある場合

```
ENGINE DB2
SELECT CLIENT APPLNAME, CLIENT USERID, CLIENT WRKSTNNAME, CLIENT ACCTNG
FROM SYSIBM.SYSDUMMY1;
END
```

命名規則の設定

複合テーブル、ビュー、RPC 名の解釈に使用する区切り文字を制御します。区切り文字としてスラッシュ (/) を使用するシステムで区切り文字としてピリオド (.) が記述されているリクエストを実行できるようになります。

この設定は、CLI として構成されたサーバとともに使用する場合にのみ有効です。その構成では、大部分のプラットフォームに適用されます。サーバで両方の命名規則をサポートする必要がある場合は、必要な命名設定を使用するサービスブロックプロファイルを持つ別のサービスブロックを構成して、対象とするブロックにリクエストを送信することができます。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

命名規則の設定

使用可能なパラメータは次のとおりです。

```
ENGINE DB2 SET NAMING {SQL|SYS}
```

説明

DB2

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

SQL

標準のピリオド (.) が、複合テーブル名の区切り文字として使用されます。

SYS

システムのスラッシュ (/) が、複合テーブル名の区切り文字として使用されます。

HOLD DBMS 作成の制御

HOLD AS *app/name* FORMAT DB 構文が拡張され、HOLD DBMS ファイルの作成に関する、より厳密な制御が可能になりました。

DBMS 作成の制御

```
HOLD AS app/name FORMAT DB2 [TABLENAME dbms_name] [CONNECTION conn_name]
[DROP]
```

説明

dbms_name

作成する DBMS テーブルです。1 部構成、2 部構成、3 部構成のいずれかの名前にします。複数の部分を区切るには、DBMS に適合した区切り記号を使用します。

conn_name

DBMS の接続名です。複数の DBMS 接続を構成して使用している場合は、conn_name で接続名を指定します。

DROP

作成前にテーブルを削除します。このオプションを使用すると、既知のテーブルまたは `persistValue=global_temporary` 設定時に作成され一時スペースに保存されているテーブルのいずれかを削除することができます。この DROP オプションを使用しない場合、既存のテーブルが存在すると、テーブルの作成時にエラーが発生してテーブルのロードに失敗します。

既存のテーブルが存在しない場合、このオプションは無視されます。

Db2 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)および[定数として定義された一時項目 \(DEFINE\) を転送するリクエストの最適化](#)を参照してください。

非等価条件 WHERE ベース LEFT OUTER JOIN の最適化

LEFT OUTER JOIN は、ホストテーブルからすべてのレコードを選択し、それらをクロスリファレンステーブルのレコードと照合します。一致するレコードが存在しない場合、ホストレコードは保持されますが、クロスリファレンスフィールドにはデフォルト値 (ブランクまたはゼロ) が割り当てられます。アダプタは、RDBMS で条件式がサポートされるすべての WHERE ベースの LEFT OUTER JOIN コマンドを最適化することができます。

条件付き LEFT OUTER JOIN の指定

```
JOIN LEFT_OUTER FILE hostfile AT hfld1 [TAG tag1]
  [WITH hfld2]
  TO {UNIQUE|MULTIPLE}
  FILE crfile AT crfld [TAG tag2] [AS joinname]
  [WHERE expression1;
  [WHERE expression2;
  ...]

END
```

説明

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。

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 パラメータにフィールド値を指定しても、この値が接続情報として使用されることはありません。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

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に一意的修飾子として使用するタグ名 (オプション) です。

WITH 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 に基づく条件付き JOIN では、KEEPDEFINES を ON に設定し、一時項目 (DEFINE) を JOIN コマンドを発行する前に作成しておく必要があります。

MULTIPLE

from_file と *to_file* 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。なお、ONE は UNIQUE の同義語です。

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。

ユニーク JOIN は WebFOCUS の概念です。RDBMS はユニークと非ユニークを区別しないため、クロスリファレンスファイルから条件に一致するすべての行を取得します。

たとえば、リクエストでユニーク JOIN として指定したが、実際には条件に一致する行がクロスリファレンスファイルに複数存在する場合、この JOIN の処理を RDBMS が行うと、RDBMS は一致するすべての行を返します。一方、最適化を無効にしてこの JOIN の処理を WebFOCUS が行うと、異なるレポート結果が得られます。WebFOCUS はユニーク JOIN の概念を順守するため、ホストの 1 行に対してクロスリファレンスの行を 1 行だけ返します。

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データソース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。

WHERE ベース OUTER JOIN の最適化条件

- WHERE ベースの LEFT OUTER JOIN を最適化するには、それに関係する RDBMS で式の最適化を行うことができるとともに、次の条件のいずれかを満たす必要があります。
 - JOIN WHERE コマンドに、少なくとも 1 つの「field1 EQ field2」述語が含まれている。ここで、field1 は table1 のフィールド、field2 は table2 のフィールドを表します。
 - または
 - 右側のテーブルに、NULL データを含まないユニークインデックスまたはキーがある。
 - または
 - 右側のテーブルに、長いデータタイプ (例、TEXT、IMAGE) を持たない「NOT NULL」フィールドが少なくとも 1 つ含まれている。
- アダプタの SQLJOIN OUTER 設定は、ON (デフォルト) に設定しておく必要があります。

非等価条件 LEFT OUTER JOIN の最適化

次のリクエストは、2 つの Db2 データソース間の条件付き LEFT OUTER JOIN を作成し、結合後のデータソースを使用してレポートを作成します。このリクエストに対して生成された SQL を表示するため、STMTRACE がオンに設定されています。

```
SET TRACEUSER = ON
SET TRACEOFF = ALL
SET TRACEON = STMTRACE//CLIENT
JOIN LEFT_OUTER FILE baseapp/EQUIP AT CARS
TO ALL FILE baseapp/CARREC AT CARC
WHERE CARS NE CARC;
END
TABLE FILE baseapp/EQUIP
PRINT CARS CARC STANDARD
BY MODEL
END
```

WebFOCUS リクエストは、LEFT OUTER JOIN を組み込んだ単一の DB2 SELECT ステートメントに変換され、非等価条件が ON 句で RDBMS に渡されます。

```
SELECT T1."CARS"(CHAR( 16)),T1."STANDARD"(CHAR( 40)),
T2."CARC"(CHAR( 16)),T2."MODEL"(CHAR( 24)) FROM ( EQUIP T1 LEFT
OUTER JOIN CARREC T2 ON (T1."CARS" <> T2."CARC") ) ORDER BY
T2."MODEL";
```

検索処理のブロックサイズの指定

Db2 アダプタは、SELECT クエリまたはストアードプロシジャの実行結果に対する配列検索をサポートします。この手法により、ネットワークおよび CPU 使用の負荷を大幅に軽減することができます。

大きい値を指定すると仮想記憶域のコストは増えてますが、大量の行を処理するリクエストの効率は改善されます。ただし、100 を超える値を指定しても僅かな効率改善しか期待できないため、100 以内の値を設定することをお勧めします。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

配列検索のブロックサイズの指定

SELECT リクエストのブロックサイズは、TABLE FILE リクエスト、MODIFY リクエスト、MATCH リクエスト、および DIRECT SQL SELECT ステートメントに適用されます。

```
ENGINE DB2 SET FETCHSIZE n
```

説明

DB2

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

CLI アダプタで配列取得手法を使用するか、CAF アダプタで行セット配置カーソルおよび複数行フェッチを使用して同時に取得される行数です。受容値は 1 から 32000 です。デ

フォルト値は 100 です。CLOB または BLOB として処理する必要のあるフィールドが結果セットに含まれている場合は、その結果セットに使用する FETCHSIZE 値は 1 です。

挿入処理のブロックサイズの指定

INSERT のブロックサイズは、LOADONLY と組み合わせて MODIFY INCLUDE リクエストに適用されます。INSERTSIZE は、パラメータ化 DIRECT SQL INSERT ステートメントでもサポートされます。

```
ENGINE DB2 SET INSERTSIZE n
```

説明

DB2

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

配列検索手法を使用して挿入する行数です。受容値は 1 から 5000 です。BLOB として処理する必要のあるフィールドが結果セットに含まれている場合は、その結果セットに使用する INSERTSIZE 値は 1 です。

バルクインサート API の省略

```
ENGINE DB2 SET FASTLOAD [ON|OFF]
```

説明

DB2

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

バルクインサート API を使用します。デフォルト値は ON です。

OFF

バルクインサート API の使用を抑制します。

バルクインサート API の動作

ibi Data Migrator で Db2 のバルクインサート API (Windows および UNIX) を使用できます。

ロード中に発生するエラー (例、重複エラー) により、行のバッチ全体が拒否される場合があります。

集計の最適化による効率の改善

集計の最適化により、クエリの効率が大幅に改善されます。

この機能についての詳細は、[集計の最適化のサポート](#)を参照してください。

集計の最適化の設定

```
SET AGGREGATE_AWARENESS {FRESH_ONLY|OLD_OK|OFF}
```

説明

FRESH_ONLY

各 RDBMS に関連するパラメータに異なる値を設定します。

OLD_OK

各 RDBMS に関連するパラメータに異なる値を設定します。

OFF

オプションを選択しない場合、ターゲット RDBMS の動作はデータベース構成オプションに基づいて決定されます。この設定のデフォルト値はありません。

アダプタ固有の設定の詳細は、[集計最適化の使用時の注意](#)を参照してください。

Db2 Cube Views の使用

Db2 アダプタは、オブジェクトタイプとして Cube Views をサポートします。シノニムを作成するオブジェクトタイプとして Cubes を指定することができます。

Cube Views で Db2 アダプタを使用するには、Db2 cube views を含むデータベースの Db2 接続を定義しなければなりません。

詳細は、[シノニムの作成](#)を参照してください。

Db2 Cubes Views のメタデータのマッピング

Cubes (Cube Models ではなく) のシノニムが生成されます。これらのシノニムは、基本的な Db2 テーブルおよびそのカラムの構造ではなく、キューブとその属性およびメジャーの構造を反映します。

マスターファイルのシノニム構成要素には、複数のセグメントが含まれています。

- ルートセグメントは Cube Fact に対応しています。
- 子セグメントは Cube Hierarchie に対応しています。

ルートセグメントのフィールドは Cube Measure を反映しています。エイリアスはメジャー名に基づいて生成されますが、フィールド名とそのタイトルはメジャービジネス名に基づいて生成されます。

- これらのフィールドの PROPERTY 属性には、MEASURE という用語 (または計算されたメジャーでは Calculated Measure) が含まれています。
- REFERENCE 属性には、簡単なカラム (式ではなく) に基づくメジャーの集計式、および DML で使用可能な集計が含まれています。

Db2 Cube View のマスターファイルの例

```
FILENAME=BASEAPP/COMPLEX_MEASURES__C_, SUFFIX=DB2CV , $  
SEGMENT=SALESCF, SEGTYPE=S0, $  
FIELDNAME=4ADDS, ALIAS=4adds, USAGE=I11, ACTUAL=I4, MISSING=ON,
```

```

    TITLE='4adds', REFERENCE=CNT.4ADDS, PROPERTY=MEASURE, $
FIELDNAME=POWEROFADDS, ALIAS=powerofadds, USAGE=D21.2, ACTUAL=D8,
    MISSING=ON, TITLE='powerofadds', PROPERTY=MEASURE, $
FIELDNAME=PROFIT_CONST, ALIAS=profit*const, USAGE=D21.2, ACTUAL=D8,
    MISSING=ON, TITLE='profit*const', PROPERTY=MEASURE, $
FIELDNAME=RANDOM, ALIAS=random, USAGE=I11, ACTUAL=I4, MISSING=ON,
    TITLE='random', REFERENCE=CNT.RANDOM, PROPERTY=MEASURE, $
FIELDNAME=3_COLS_WITH_MIXTUREOF_FUNC,
    ALIAS='3 cols with mixtureof FUNC', USAGE=D21.2, ACTUAL=D8,
    MISSING=ON, TITLE='3 cols with mixtureof FUNC',
    PROPERTY=MEASURE, $
FIELDNAME=ROUND_OF_ANOTHER_MEAS, ALIAS='round of another meas',
    USAGE=D21.2, ACTUAL=D8, MISSING=ON, TITLE='round of another meas',
    PROPERTY=MEASURE, $

```

```

SEGMENT=PRODUCT__CH_, SEGTYPE=U, PARENT=SALESCF, $
FIELDNAME=PRODUCT_GROUP_ID, ALIAS=product_group_ID, USAGE=I11,
    ACTUAL=I4, MISSING=ON, TITLE='product_group_ID',
    WITHIN='*product (CH)', $
FIELDNAME=PRODUCT_LINE_ID, ALIAS=product_line_ID, USAGE=I11, ACTUAL=I4,
    MISSING=ON, TITLE='product_line_ID', WITHIN=PRODUCT_GROUP_ID, $
FIELDNAME=PRODUCT_LINE_NAME,
    ALIAS='12 product_line_ID product_line_name', USAGE=A25V,
    ACTUAL=A25V, MISSING=ON, TITLE='product_line_name',
    REFERENCE=PRODUCT_LINE_ID, PROPERTY=CAPTION, $

```

```

SEGMENT=STORE__CH_, SEGTYPE=U, PARENT=SALESCF, $
FIELDNAME=STORE_LOCATION_ID, ALIAS='**** store_location_ID_1',
    USAGE=I11, ACTUAL=I4, MISSING=ON, TITLE='store_location_ID',
    WITHIN='*store (CH)', $
FIELDNAME=STORE_ADDRESS,
    ALIAS='store_location_ID_1 _store_address', USAGE=A25V,
    ACTUAL=A25V, MISSING=ON, TITLE='store address',
    REFERENCE=STORE_LOCATION_ID, PROPERTY=UDA, $
FIELDNAME=STORE_CITY, ALIAS='store_location_ID_1 _store_city',
    USAGE=A45, ACTUAL=A45, MISSING=ON, TITLE='store_city',
    REFERENCE=STORE_LOCATION_ID, PROPERTY=UDA, $
FIELDNAME=STORE_ID, ALIAS=store_ID, USAGE=I11, ACTUAL=I4, MISSING=ON,
    TITLE='store_ID', WITHIN=STORE_LOCATION_ID, $

```

SQL パススルーによる Db2 ストアドプロシージャの呼び出し

Db2 のストアドプロシージャは、SQL パススルーの使用をサポートします。これらのプロシージャは、CREATE PROCEDURE コマンドを使用して Db2 内で開発する必要があります。

アダプタは、IN、OUT、INOUT パラメータを含むストアドプロシージャをサポートします。

ストアドプロシージャによって返される出力パラメータ値は結果セットとして利用可能になります。単一行の結果セットのこれらの値は、呼び出したストアドプロシージャによって他のすべての結果セットが返された後で、クライアントに転送されます。(利用可能な場合の) 出力パラメータ名は、結果セットのカラムタイトルになります。

クライアントに返されるのは、呼び出し文字列内で参照されている出力パラメータ (およびその戻り値) だけです。結果として、ユーザは表示する出力パラメータ値を完全に制御することができます。

サーバは、基となる DBMS の規則に従って記述されたストアドプロシージャの呼び出しをサポートします。ここでの例は SQL ベースであることに注意してください。DBMS の規則、言語、および他のプログラミングの例については、DBMS のマニュアルを参照してください。

ストアドプロシージャの呼び出し

```
SQL DB2 EX procname [parameter_specification1]
[,parameter_specification2]...
END
```

説明

DB2

Db2 の ENGINE 接尾語です。

procname

ストアドプロシージャ名です。ネイティブデータベース構文のストアドプロシージャの完全 (または部分) 修飾名です。

SQL または SYS のいずれかの命名規則を使用して、複数構成名の解釈に使用される区切り文字を制御することができます。詳細は、[命名規則の設定](#)を参照してください。

parameter_specification

IN、OUT、INOUT がサポートされています。ストアプロシジャにより異なる適切なパラメータを指定します。

IN

リテラル (例、125、3.14、'abcde') です。入力文字に予約語を使用することも可能です。予約語は文字リテラルとは異なり、引用符で囲む必要はありません (例、NULL)。この値は必須です。

OUT

1 つの疑問符 (?) で表されます。このパラメータの有無により、出力結果をアプリケーションへ渡すかどうかを決定します。省略すると、この項目は (0 バイトの) 空白文字列になります。

INOUT

出力を表す 1 つの疑問符 (?) と入力を表すリテラルで構成されます。出力と入力はスラッシュ (/) で区切ります (例、?/125、?/3.14、?/'abcde')。出力値を (0 バイトの) 空白文字列にすることも可能です。

ストアプロシジャの呼び出し

この例では、SQL の命名規則を使用していることに注意してください。ここでは、ピリオド (.) が複合テーブル名の区切り文字として使用されます。サイトで SYS[TEM] の命名規則を使用している場合、複数構成のテーブル名の区切り文字としてスラッシュ (/) が使用されます。この場合、必要に応じて SYS 命名規則に適合するように区切り文字を調整してください。詳細は、[命名規則の設定](#)を参照してください。

この例では、ユーザが edaqa.test_proc01 ストアドプロシジャを呼び出してパラメータ 1、3、5、7 の入力値を指定することで、このストアプロシジャの戻り値、およびパラメータ 2、3 の出力値を取得します。

パラメータ 4 と 6 が省略されていることに注意してください。ストアプロシジャは、作成時に指定されたデフォルト値を使用します。

```
SQL DB2 EX edaqa.test_proc01 125,?,?/3.14,, 'abc',, 'xyz'
END
```


ストアードプロシージャの例

この例では、SQL の命名規則を使用していることに注意してください。ここでは、ピリオド (.) が複合テーブル名の区切り文字として使用されます。サイトで SYS[TEM] の命名規則を使用している場合、複数構成のテーブル名の区切り文字としてスラッシュ (/) が使用されます。この場合、必要に応じて SYS 命名規則に適合するように区切り文字を調整してください。詳細は、[命名規則の設定](#)を参照してください。

このストアードプロシージャは、OUT および INOUT パラメータを使用します。

```
CREATE PROCEDURE EDAQA.PROCP3 (    OUT chSQLSTATE_OUT    CHAR(5),
                                OUT intSQLCODE_OUT    INT,
                                INOUT l_name char(20),
                                INOUT f_name char(20))

    RESULT SETS 1
    LANGUAGE SQL

-----
-- SQL Stored Procedure
-----

P1:BEGIN
    -- Declare variable
    DECLARE SQLSTATE CHAR(5) DEFAULT '00000';
    DECLARE SQLCODE INT DEFAULT 0;
    -- Declare cursor
    DECLARE cursor1 CURSOR WITH RETURN FOR
        SELECT
            EDAQA.NF29005.SSN5 AS SSN5,
            EDAQA.NF29005.LAST_NAME5 AS LAST_NAME5,
            EDAQA.NF29005.FIRST_NAME5 AS FIRST_NAME5,
            EDAQA.NF29005.BIRTHDATE5 AS BIRTHDATE5,
            EDAQA.NF29005.SEX5 AS SEX5
        FROM
            EDAQA.NF29005
        WHERE
            (
                ( EDAQA.NF29005.LAST_NAME5 = l_name )
            AND
                ( EDAQA.NF29005.FIRST_NAME5 = f_name )
            );
    -- Cursor left open for client application
    OPEN cursor1;
    SET chSQLSTATE_OUT = SQLSTATE;
    SET intSQLCODE_OUT = SQLCODE;
    SET l_name = 'this is first name';
```

```
      SET f_name = 'this is last name';  
END P1    @
```

ESRI ArcGIS アダプタの使用

ESRI ArcGIS アダプタは、ESRI ArcGIS Online 環境またはローカルホスト ArcGIS サーバに存在する情報からレポートを作成する際に使用します。たとえば、特定の郵便番号 (ZIP コード) のデータエンリッチメント情報を分析することができます。

ESRI ArcGIS アダプタを構成するには、WebFOCUS Reporting Server ブラウザインターフェースを使用します。このアダプタには、リフレッシュトークンの格納先への接続が必要です。有効な ESRI ArcGIS Online のリフレッシュトークンは、ESRI ArcGIS Online の AP コールを発行するために必要です。このトークンは、ESRI ArcGIS Online アプリケーションおよび特定の ESRI ArcGIS Online ユーザに関連付けられています。

オンプレミスの ESRI サーバを使用していない場合、マップ機能を使用するためには、WebFOCUS Reporting Server からオンラインリソースにアクセスする必要があります。デフォルト設定で、サイトのセキュリティルールによりアクセスがブロックされる場合は、特定の URL のリストが必要になります。必要な URL は、次の ESRI サイトに記載されています。

https://downloads.esri.com/resources/enterprisegis/AGOL_Domain_Requirements.pdf

注意：WebFOCUS Hub で ESRI ArcGIS アダプタへの定義済み接続を構成するには、WebFOCUS および WebFOCUS Reporting Server が使用する Application Server の両方が同一マシンで稼働している必要があります。同一マシンで稼働していない場合は、アダプタのリフレッシュトークンを取得することができず、[リフレッシュトークン] フィールドには「リフレッシュトークンが取得できません」 (Refresh Token unavailable) というテキストが表示されます。この場合、HTTP リスナでアクセスする WebFOCUS Reporting Server ブラウザインターフェースからアダプタを構成し、トークンを取得します。

ESRI ArcGIS Online アプリケーションの作成

ESRI ArcGIS Online アプリケーションは、ESRI ArcGIS アダプタの構成前に取得しておく必要があります。

ESRI ArcGIS アプリケーションを作成するには

手順

1. Web ブラウザのアドレスバーに次の URL を入力します。

```
https://developers.arcgis.com/applications/#/
```

ESRI にログイン済みでない場合は、下図のように、ArcGIS for Developers サイトのログインダイアログボックスが表示されます。

Sign in with 

ArcGIS login 

 Username

 Password

[Sign In](#) [Cancel](#)

[Forgot username? or Forgot password?](#)

Your ArcGIS organization's URL 

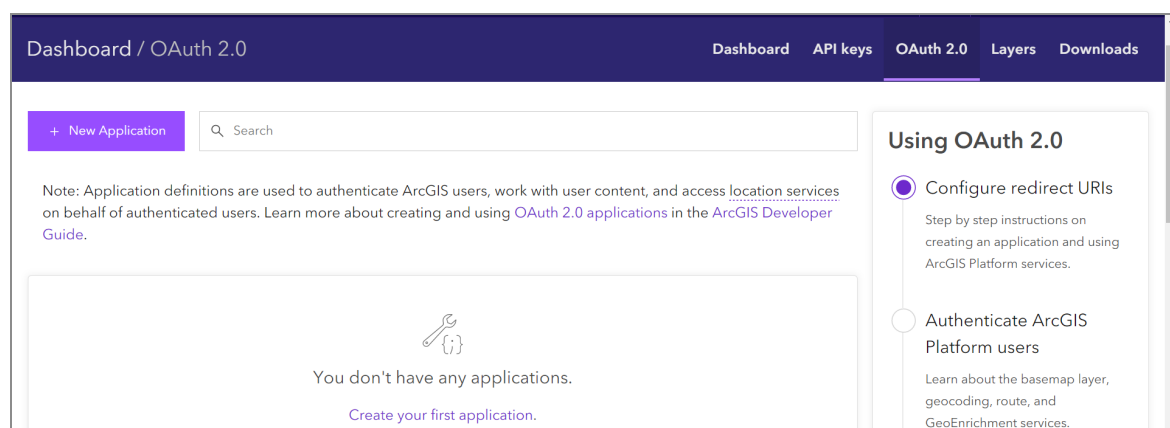
   

[Privacy](#)

2. ESRI ArcGIS Online アプリケーションの作成権限を持つ有効な ESRI ArcGIS Online ア

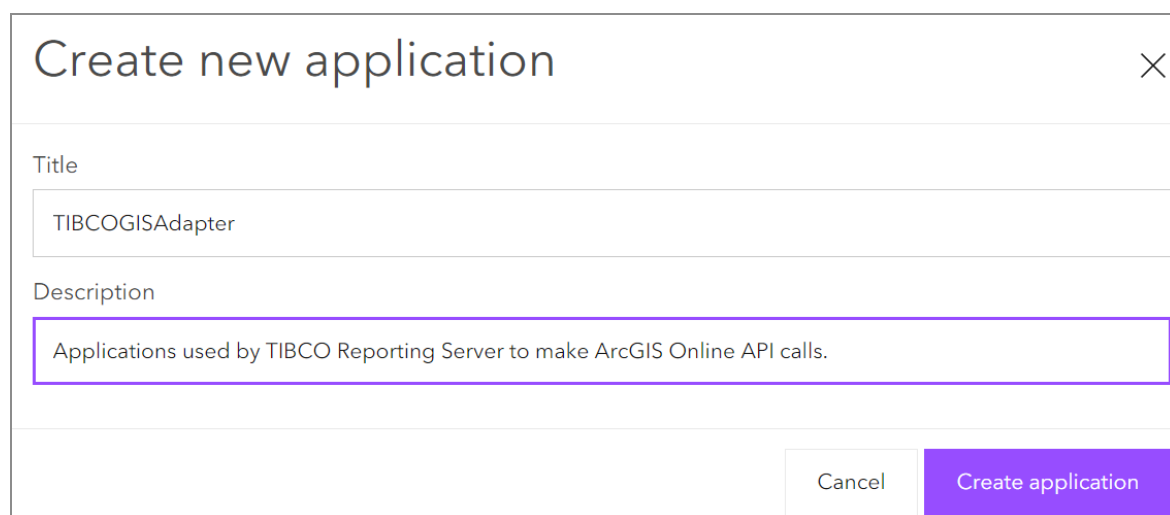
カウントの認証情報を入力し、[Sign In] をクリックします。

下図のように、[Dashboard] が開きます。



3. [+ New Application] をクリックします。

4. 下図のように、新規アプリケーションのタイトルと説明を入力します。



5. [Create application] をクリックします。

下図のように、アプリケーションの概要画面が開き、[Client ID] および [Client Secret] の値が表示されます。

TIBCOGISAadapter

Applications used by TIBCO Reporting Server to make ArcGIS Online API calls.

Created: January 3, 2022 | Modified: January 3, 2022

Overview

Proxies

Usage

Settings

Open in ArcGIS Online

OAuth 2.0 Credentials

OAuth can be used to authenticate ArcGIS users to access content and location-based services on behalf of the authenticated user. [Learn more about OAuth 2.0 in the security and authentication guide.](#)

Client ID

Client Secret

Temporary Token

Short lived token for testing. Will expire Mon, 03 Jan 2022 20:45:50 GMT .

Redirect URLs

+ Add URI

These are the allowed URLs to redirect users to after completing authorization. See [implementing OAuth 2.0](#) for more information.

No Redirect URIs.

6. ページを下方へスクロールして [Redirect URLs] セクションに移動し、[+ Add URI] をクリックします。
7. [Add Allowed URI] フィールドに、WebFOCUS Reporting Server ブラウザインターフェイスへのアクセスに使用するホスト名とポート番号を入力し、末尾に `oauth20.exe` を追加します。

以下はその例です。

http://host.tibco.com:8121/oauth20.exe

WebFOCUS Reporting Server がスタンドアロンサーバとしてインストールされていない場合、[Add Redirect URI] フィールドの値として次の値を指定する必要があります。

```
http://localhost/oauth20.exe
```

8. [Add URI] をクリックします。

ESRI ArcGIS の接続属性

以下は、ESRI ArcGIS アダプタの接続属性を示しています。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

接続タイプ

ESRI に接続する接続タイプです。利用可能な値は、[ArcGIS] と [on premises] です。

- **ArcGIS** ESRI ArcGIS Online 環境を示します。これがデフォルト値です。
- **on premises** ArcGIS サーバが、ArcGIS Online と併用して、または ArcGIS Online の代わりに使用されることを示します。

セキュリティ

データベースサーバへの接続時に、ユーザを認証する方法です。[OAuth] または [トークンの生成] のいずれかの認証方法を選択します。

OAuth

以下は、[OAuth] セキュリティの接続属性です。

認可 URL

ESRI ArcGIS へのアプリケーションの認可に使用する URL です。デフォルト値は次のとおりです。

```
https://www.arcgis.com/sharing/rest/oauth2/authorize
```

トークン URL

ESRI ArcGIS にアクセスするトークンを取得するための URL です。デフォルト値は次のとおりです。

```
https://www.arcgis.com/sharing/rest/oauth2/token
```

クライアント ID/クライアントシークレット

OAuth 認証を使用した場合に、ESRI ArcGIS に対してユーザのアプリケーションを識別する値です。

リフレッシュトークン

実行時に新しいアクセストークンを取得するために使用されます。

トークンの生成

以下は、[トークンの生成] セキュリティの接続属性です。コールバック URL 要件により、クラウド環境ではこのセキュリティ方法を使用する必要があります。

ユーザ

ユーザサインイン用の ESRI ArcGIS 認証 ID です。

パスワード

ユーザサインイン用の ESRI ArcGIS パスワードです。

トークン URL

ESRI ArcGIS にアクセスするトークンを取得するための URL です。デフォルト値は次のとおりです。

```
https://www.arcgis.com/sharing/generateToken
```

WAPPURL

このトークンを使用するクライアントアプリケーションのベース URL です。これは、NAMED 認証と基本的に同じであり、クラウド環境で必要です。デフォルト値は次のとおりです。

```
https://mysrver/mywebapp
```

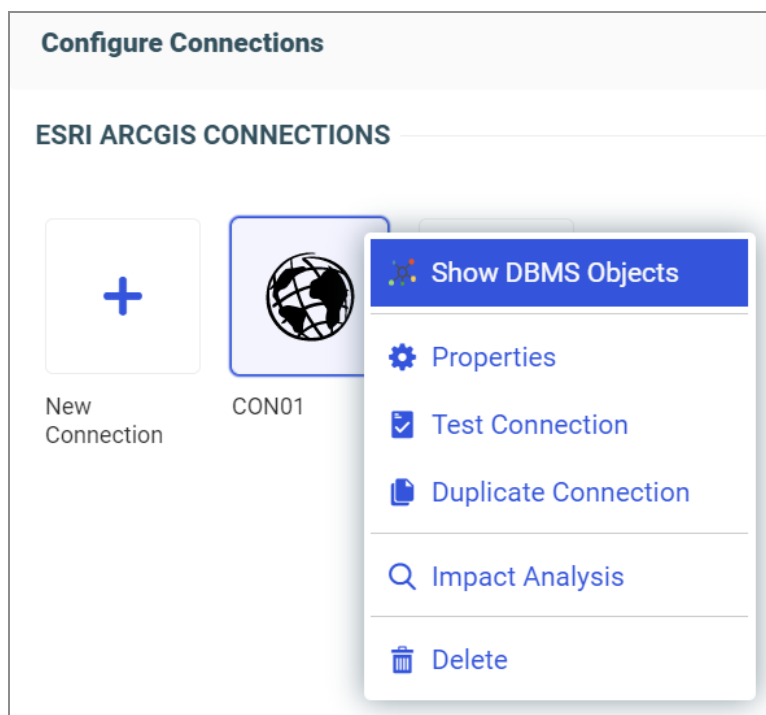

プレミアム API コールを使用した ESRI ArcGIS アダプタでのメタデータおよびサンプルレポートの作成

ESRI ArcGIS アダプタでのシノニムの作成では、プレミアム ESRI ArcGIS Online API コールから返されるデータでの WebFOCUS レポートの作成に使用するメタデータおよびサンプル WebFOCUS レポートを作成することができます。

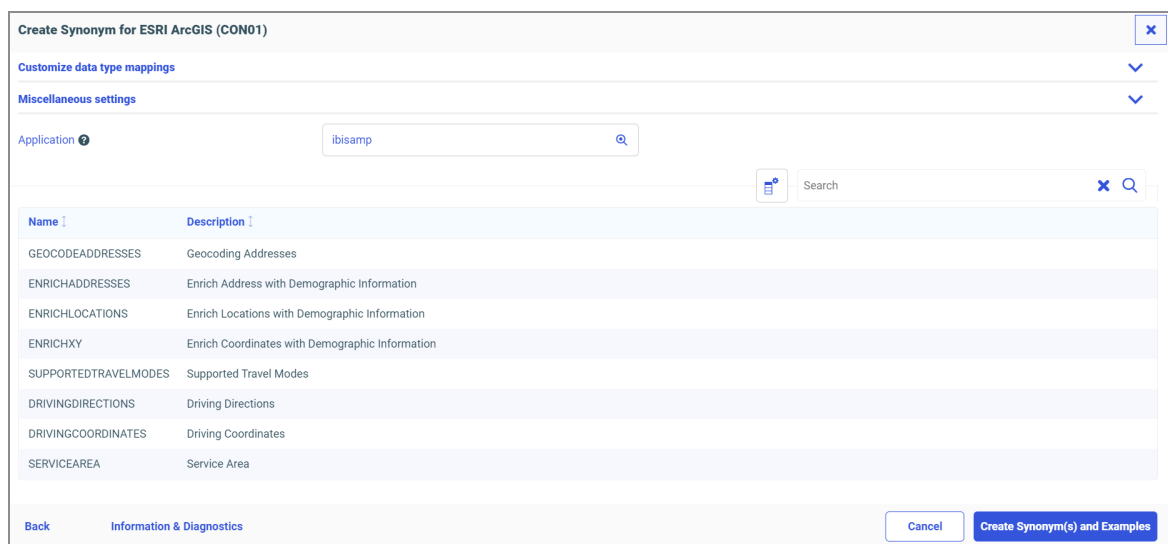
プレミアム API コールによるメタデータおよびサンプルレポートを作成するには

手順

1. プレミアム ArcGIS Online API コールにアクセス可能な構成済み ESRI ArcGIS 接続 (例、CON01) を右クリックし、コンテキストメニューから [DBMS オブジェクトの表示] を選択します。



下図のように、[ESRI ArcGIS シノニムの作成] ウィンドウが開きます。



2. [アプリケーション] テキストボックスにアプリケーションフォルダ名を入力するか、テキストボックス右横の参照ボタンをクリックしてアプリケーションを選択します。このアプリケーションフォルダに、メタデータおよびサンプルレポートが格納されます。
3. [シノニムとサンプルの作成] をクリックします。

[ESRI ArcGIS シノニムの作成ステータス] ページが開き、シノニムが正しく作成されたことを示すメッセージが表示されます。

メタデータおよびサンプルレポート

ここでは、ESRI ArcGIS アダプタのメタデータおよびサンプルレポートについて説明します。

ESRI ArcGIS アダプタのメタデータ

メタデータ	説明
drivingcoordinates	ジオコードされた座標間の走行経路を示す一連のジオコードされた座標
drivingdirections	ジオコードされた座標間の走行経路
enrich_wfretail_us	人口統計情報を含む Retail Sales Demo Data の郵便番号 (ZIP コード) の追加
enrichaddresses	人口統計情報を含む住所を囲むラジアスの追加
enrichlocations	人口統計情報を含むロケーションの追加 (例、郵便番号)
enrichxy	人口統計情報を含むジオコードされた座標を囲むラジアスの追加
geocodeaddresses	特定住所のジオコード
servicearea	特定のロケーション周辺で利用可能なサービスエリアのジオコードされた座標
supportedtravelmodes	有効な移動モード

ESRI ArcGIS アダプタのサンプルレポート

下表は、ESRI ArcGIS アダプタで使用可能なサンプルレポートのリストおよび説明を示しています。

サンプルレポート	説明
esrsampl/drivingcoordinates	ジオコードされた座標間の走行経路を示す一連のジオコードされた座標を返します。 drivingcoordinates シノニムを使用します。
esrsampl/drivingdirections	ジオコードされた座標間の走行経路を返します。 drivingdirections シノニムを使用します。
esrsampl/enrich_wfretail_us	人口統計情報を含む Retail Sales Demo Data の郵便番号 (ZIP コード) の追加 enrichlocations シノニムを使用します。
esrsampl/enrichaddresses	人口統計情報を含む住所を囲むラジアスを追加します。 enrichaddresses シノニムを使用します。
esrsampl/enrichlocations	人口統計情報を含む 1 つまたは複数の郵便番号 (ZIP コード) を追加します。 enrich_wfretail_us シノニムを使用します。
esrsampl/enrichxy	人口統計情報を含むジオコードされた座標を囲むラジアスを追加します。 enrichxy シノニムを使用します。
esrsampl/geocodeaddresses	特定住所のジオコード geocodeaddresses シノニムを使用します。

サンプルレポート	説明
esrsampl/servicearea	す。 特定のロケーション周辺で利用可能なサービスエリアのジオコードされた座標のリストを返します。 servicearea シノニムを使用します。
esrsampl/supportedtravelmodes	サポートされる移動モードのリストを返します。 supportedtravelmodes シノニムを使用します。

Excel アダプタの使用

Excel アダプタの構成方法について説明します。

Excel アダプタを使用すると、アプリケーションから Excel データソースへのアクセスが可能になります。アダプタは、リクエストの送信元アプリケーションにアンサーセットを返します。

Excel アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

アダプタは、Excel データベースサーバに接続する際に、接続情報および認証情報を必要とします。この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。選択したプロファイルにコマンドが追加されます。選択可能なプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) があります。
- グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) のいずれかに手動でコマンドを追加します。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の Excel データベースサーバへの接続を宣言することができます。Excel データベースサーバへの実際の接続は、その接続を参照する最初のクエリの発行時に確立されます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。

4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。

5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。

6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Excel の接続属性

Excel アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

データソース

データソース名 (DSN) です。デフォルトのデータソース名はありません。値を入力する必要があります。

この値は、Microsoft Excel Driver 用に構成された ODBC データソース名にする必要があります。ファイル DSN を使用する場合は、そのファイル名にする必要があります (例、samplefdsn.dsn)。

セキュリティ

データソースへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** Excel には適用されません。今後使用するために予約されています。
- **Trusted** アダプタは、サーバのデータアクセスエージェントによって偽装されたオペレーティングシステムユーザの認証情報を使用して、Windows ログインとして Excel データソースに接続します (デフォルト)。

ユーザ

Excel には適用されません。今後使用するために予約されています。

パスワード

Excel には適用されません。今後使用するために予約されています。

ファイル DSN パラメータ (オプション)

Windows マシンまたはネットワーク上で定義された ODBC ファイル DSN のパスです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロ

ファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

接続属性の手動宣言

ユーザ/システム DSN の場合

```
ENGINE SQLEXCEL SET CONNECTION_ATTRIBUTES connection DSN_name/,
```

ファイル DSN の場合

```
ENGINE [SQLEXCEL] SET CONNECTION_ATTRIBUTES connection  
FileDSN_name/,:'filedsn=FileDSN_path';,
```

説明

SQLEXCEL

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

この接続属性のセットの識別に使用される論理名です。

connection と *DSN_name* の間にスペースが 1 つ必要です。

DSN_name

アクセスする Excel データソース名 (DSN) です。この名前は、odbc.ini ファイルのエントリに一致させる必要があります。

FileDSN_name

アクセスする Excel ファイルデータソース名 (DSN) です。

FileDSN_path

ハードドライブまたはネットワーク上の Excel ファイル DSN のパスです。

接続属性の宣言

ユーザ/システム DSN の場合

次の SET CONNECTION_ATTRIBUTES コマンドは、SAMPLESERVER という名前の Excel DSN への CON1 接続を宣言します。

```
ENGINE SQLEXCEL SET CONNECTION_ATTRIBUTES CON1 SAMPLESERVER/,
```

ファイル DSN の場合

次の SET CONNECTION_ATTRIBUTES コマンドは、SAMPLESERVER.dsn という名前の ODBC ファイル DSN を使用して CON1 接続を宣言します。

```
ENGINE SQLEXCEL SET CONNECTION_ATTRIBUTES CON2  
SAMPLESERVER.dsn /,:'filedsn=C:¥ SAMPLESERVER.dsn;'
```

デフォルト接続の上書き変更

複数の接続を定義した場合、最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続がデフォルト接続として機能します。このデフォルト値は、SET DEFAULT_CONNECTION コマンドを使用して上書きすることができます。

デフォルト接続の変更

```
ENGINE SQLEXCEL SET DEFAULT_CONNECTION connection
```

説明

SQLEXCEL

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

```
ENGINE SQLEXCEL SET DEFAULT_CONNECTION SAMPLE
```

接続範囲の制御

確立する接続ごとにアダプタを使用する場合、SET AUTODISCONNECT コマンドを使用して接続の継続を制御することができます。

接続範囲の制御

```
ENGINE SQLEXCEL SET AUTODISCONNECT ON {FIN|COMMAND|COMMIT}
```

説明

SQLXCEL

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMAND

各リクエストの後に自動的に切断されます。SET AUTODISCONNECT コマンドは、イベントがどの程度の頻度で発生するかによって、大幅なオーバーヘッドをもたらす可能性があります。このオーバーヘッドのほとんどすべてがサーバには関連しません。このオーバーヘッドは、オペレーティングシステムとデータソースに関連します。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

Excel メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および Excel データタイプのサーバマッピングを定義するシノニムを作成します。

アダプタの識別

リクエストの解釈に必要なアダプタを識別するには、マスターファイルの SUFFIX 属性を使用します。Excel アダプタを識別するには、SUFFIX 値として SQLXCEL を使用します。

アダプタの識別

```
FILE[NAME]=file, SUFFIX=SQLXCEL [,,$]
```

説明

file

マスターファイルのファイル名です。このファイル名は文字で開始する必要があります。また、テーブルまたはビューの内容を表す名前にすることをお勧めします。実際のファイルには拡張子 .mas を付ける必要がありますが、この属性の値には拡張子を含めないでください。

SQLEXCEL

アダプタの値です。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な Excel テーブルまたはシートの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ペー

ジが表示されます。

- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Excel のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

Excel ブック

Excel ブックを選択します。デフォルト設定では、ODBC ソースの Excel ブックが選択されています。

必要に応じて、別の Excel ブック名を入力するか、[...] (参照) ボタンをクリックします。このクリックにより、[Excel ブックの選択] ダイアログボックスが開き、使用する Excel ブック名を選択することができます。

オブジェクト名

オブジェクト名のフィルタに使用する文字列を入力します。必要に応じて、文字列の先頭や末尾にワイルドカード文字 (%) を入力します。たとえば、「ABC%」と入力して、名前が ABC で始まるオブジェクトをすべて選択します。「%ABC」と入力して、名前が ABC で終わるオブジェクトをすべて選択します。「%ABC%」と入力して、名前の先頭、中間、または末尾に ABC が含まれるオブジェクトをすべて選択します。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

テーブル名

基になるオブジェクトの名前です。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

テーブルのリストには、次の情報が表示されます。

- Excel ブック内の範囲名
- シート名。これらの名前には、末尾に「\$」が付いています。

生成されたシノニムの例

Excel アダプタのシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

マスターファイル

```
FILENAME=ONHAND, SUFFIX=SQLEXCEL, $
SEGMENT=ONHAND, SEGTYPE=S0, $
  FIELDNAME=PROD_NUM, ALIAS=Prod_Num, USAGE=A255V, ACTUAL=A255V,
  MISSING=ON, $
  FIELDNAME=PRODNAME, ALIAS=Prodname, USAGE=A255V, ACTUAL=A255V,
  MISSING=ON, $
  FIELDNAME=QTY_IN_STOCK, ALIAS=Qty_in_Stock, USAGE=D20.2, ACTUAL=D8,
  MISSING=ON, $
  FIELDNAME=PRICE, ALIAS=Price, USAGE=D20.2, ACTUAL=D8,
  MISSING=ON, $
  FIELDNAME=COST, ALIAS=Cost, USAGE=D20.2, ACTUAL=D8,
  MISSING=ON, $
```

アクセスファイル

```
SEGNAME=SEG1_4, TABLENAME=EDAQA.NF29004,
CONNECTION=DB1, KEYS=1, WRITE=YES, $
```

アクセスファイルのキーワード

下表は、Excel のアクセスファイルのキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABLENAME	Excel テーブルを識別します。この属性に割り当てる値には、次のように Excel ブック名を含めることができます。

キーワード	説明
	<pre>TABLENAME=[filename]table</pre>
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre> CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。 <pre>KEY=fld1/fld2/.../fldn</pre>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

データタイプのサポート

サポートされるデータタイプは、実際のデータソースに応じて異なります。

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアドプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

Excel 環境のカスタマイズ

Excel アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。

タイムアウト制限の指定

Excel に対して SQL リクエストを実行した後、アダプタがその応答を待機する時間 (秒) を指定するには、TIMEOUT コマンドを使用します。

TIMEOUT コマンドの発行

```
ENGINE SQLEXCEL SET TIMEOUT {nn|0}
```

説明

SQLEXCEL

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nn

タイムアウトとなる秒数を指定します。デフォルト値は 30 です。

0

応答の待機時間が無制限であることを示します。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、[アプリケーション] ページで [データの取得] をクリックした後、[サーバのデータソース] 下で構成済みアダプタを右クリックし、メニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

```
ENGINE SQLEXCEL SET PASSRECS {ON|OFF}
```

説明

SQLXCEL

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

Excel 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)を参照してください。

Excel (直接取得) アダプタの使用

Excel (直接取得) アダプタの構成方法について説明します。

Excel (直接取得) アダプタでは、Excel ファイル (.xls および .xlsx ファイル) からの直接取得がサポートされます。このアダプタは、Microsoft ファイル形式にアクセスするための Java API である Apache POI を使用します。

Excel (直接取得) アダプタの構成

Excel (直接取得) アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで構成することができます。

Excel (直接取得) アダプタを構成するには

アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで構成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. [Excel (直接取得)] を右クリックし、[構成] を選択します。

入力パラメータは必要ありません。

構成したアダプタが、ナビゲーションウィンドウの [アダプタ] リストに追加されます。

Excel (直接取得) メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および データタイプのサーバマッピングを定義するシノニムを作成します。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な Excel シートの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

一部の設定は、シノニムの作成に影響する場合があります。アダプタ設定についての詳細は、[アダプタ設定の変更](#)を参照してください。

シノニムを作成するには

シノニムを作成するには、事前にアダプタを構成しておく必要があります。シノニムは、WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで作成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. 構成済みアダプタの接続を右クリックし、[ファイルの表示] を選択します。
ファイル選択のダイアログボックスが表示されます。
3. シノニムを作成するアプリケーションおよび Excel ファイルを特定し、[OK] をクリックします。

下図のように、[Excel ワークシートの選択] ページが開きます。

Select the Excel worksheet(s) to upload by selecting the corresponding checkbox(s).

? Scan All rows ☐ (Full row scan to determine column formats and lengths. May take a long time for large files)

? Row Scan Limit (Enter number of rows to determine column formats and lengths)

? Extend Character Length (Character columns percentage increase)

? Missing Value (Enter a string to indicate a missing value)

? Column format recognition

? Number of header rows (Enter number of topmost rows containing column headers)

? ☐ Add RowID Column

☐ Customize data type mappings

? Application ... ? Prefix ? Suffix

Workbook: doc82/Sales_Data.xlsx

☐	Default Name (editable)	Worksheet Name	Named Range	Rows/Columns
☒	sales_by_product	Sales_by_Product		32284/18

4. 次のパラメータの値を入力または選択します。

すべての行をスキャン

選択した場合、データタイプおよびデータ長を決定する際にすべての行をスキャンします。

行スキャン制限

データタイプおよびデータ長を決定する際に、スキャンする行の最大数を指定します。デフォルト値は 5000 です。

文字列の長さを拡張

部分スキャンを補うために文字列の長さに追加するパーセントです。デフォルト値は 25 です。

ミッシング値

ミッシング値を示す文字列を入力します。

フィールドフォーマット認識

行スキャン制限によりデータタイプとデータ長を決定する [非厳密処理]、または、行スキャン制限により先頭行およびその長さを使用してデータタイプを決定する [厳密処理] のいずれかを指定できます。デフォルト値は [非厳密処理] です。

ヘッダ行数

フィールド名を含むヘッダ行の数を入力します。デフォルト値は 1 です。

行 ID フィールドの追加

選択した場合、各ワークシート行の行番号を追加します。

データタイプマッピングのカスタマイズ

選択した場合、データフィールドの分解、地理的役割の追加、数値フィールドおよび文字フィールドのデータタイプの選択のために追加のフィールドを開きます。

アプリケーション

アプリケーション名を入力するか、[...] (参照) ボタンをクリックしてシノニムのアプリケーションを特定します。

追加コンポーネント

シノニム名に接頭語を割り当てます。

接尾語

シノニム名に接尾語を割り当てます。

Excel ブック

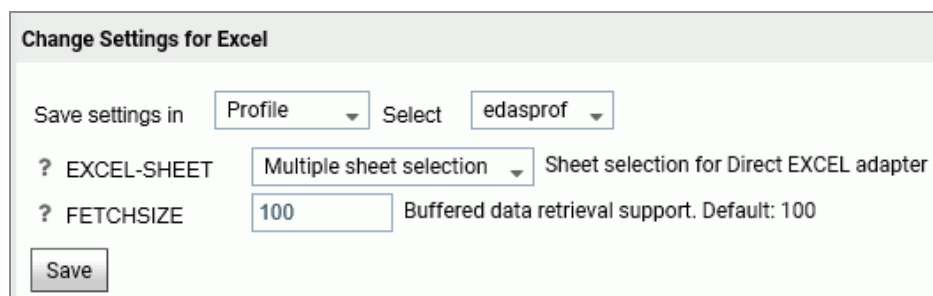
シノニムを作成する Excel ブックを選択します。シノニム名を変更するには、デフォルト名を選択し、新しい名前を入力します。

5. [ベースシノニムの作成] をクリックします。

アダプタ設定の変更

アダプタの設定を変更するには、アダプタを右クリックし、[設定の変更] を選択します。

下図のように、[Excel の設定の変更] ページが開きます。



The screenshot shows a dialog box titled "Change Settings for Excel". It contains the following elements:

- "Save settings in" with a dropdown menu set to "Profile".
- "Select" with a dropdown menu set to "edasprof".
- "? EXCEL-SHEET" with a dropdown menu set to "Multiple sheet selection". To the right of this dropdown is the text "Sheet selection for Direct EXCEL adapter".
- "? FETCHSIZE" with a text input field containing "100". To the right of this field is the text "Buffered data retrieval support. Default: 100".
- A "Save" button at the bottom left.

次の設定を変更することができます。

EXCEL-SHEET

ドロップダウンリストから、[複数シートの選択] (デフォルト) または [単一シートの選択] を選択します。この選択は、アップロードウィザードを使用してファイルをアップロードする際に、複数のワークシートを選択できるか、単一のワークシートのみを選択できるかに影響します。

FETCHSIZE

バッファデータの取得をサポートします。行数を入力するか、デフォルト値 (100) を受容します。

プロファイルの選択

ドロップダウンリストから、これらの設定の格納先として [プロファイル] または [プロシジャ] を選択し、[選択] からプロファイル名またはプロシジャ名を選択します。

Fixed-Format File および Delimited File アダプタの使用

Fixed-Format File および Delimited File アダプタの構成方法について説明します。

Fixed-Format File および Delimited Flat File アダプタは、レコードが入力された順序で保存、検索されるシーケンシャルファイルにアクセスします。シーケンシャルデータソースには、次のものがあります。

- 固定フォーマットのファイル (Fixed-Format File) (以前のバージョンでは、Flat File と記載) - 各フィールド値は、レコード内で事前に定義された位置に格納されます。固定フォーマット構造を保持するには、データの存在しない位置にはブランクを配置します。
- 区切り付きフォーマットのファイル (Delimited File) - 各フィールド値は事前に定義された順序で、特殊文字で区切られて格納されます。各フィールドは、ファイル内で事前に定義された順序を保持する必要がありますが、データの存在しない位置にブランクを配置する必要はありません。

アダプタは、アプリケーションリクエストを I/O 呼び出しに変換し、最適化されたアンサーセットをリクエストの送信元アプリケーションに返します。

このアダプタを使用して、圧縮されたデータセットにアクセスすることもできます。ただし、実際の圧縮処理は ZCOMP イグジットを使用して実行されます。詳細は、[データセット圧縮イグジット - ZCOMP](#)を参照してください。

Fixed-Format File および Delimited File 環境の準備

Fixed-Format File および Delimited File アダプタでは、環境変数を設定する必要はありません。

Fixed-Format File および Delimited File アダプタの構成

Fixed-Format File および Delimited File アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで構成することができます。また、それぞれのファイルタイプで、アクセスするデータファイルのパスを指定する必要があります。ファイルのパスは、次のいずれかの方法で定義します。

- サポート対象のサーバプロファイルに FILEDEF コマンドを含める。この方法では、メタデータに関連付けられた物理ファイル名を迅速に変更できます。
- シノニムを作成する際にパスを指定する。この方法では、データファイルのパスをマスターファイル内で定義できます。

Fixed-Format File および Delimited File アダプタを構成するには

アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで構成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

結果

注意： Fixed-Format File および Delimited File アダプタは、[ファイル] グループフォルダ下にあります。入力パラメータは必要ありません。

WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで Fixed-Format File および Delimited File の FTP サーバ接続を構成するには

FTP サーバへの接続を構成することで、FTP サーバを Flat File および Delimited Flat File のソースとして使用することができます。はじめに、アダプタを構成する必要があります。

FTP サーバへの接続を構成する手順は以下のとおりです。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. [Flat File] フォルダを右クリックし、[接続の追加] を選択するか、[区切り付きファイル (CSV/TAB)] フォルダを右クリックし、[FTP 接続の追加] を選択します。
3. [区切り付きファイル (CSV/TAB)/Flat File の FTP サーバ接続を追加] ウィンドウが開きます。

Add Connection to FTP Server for Delimited Files (CSV/TAB)

^ Connect parameters

? Connection Name

CON01

? Host name

? Initial Directory

? User

? Password

? Protocol

FTP

Environment

? Select profile

edasprof

(type in a new one or select one from the list)

Configure

Test

4. 接続パラメータを入力します。詳細は、以下の「FTP 接続パラメータ」を参照してください。
5. [構成] をクリックします。

FTP 接続パラメータ

FTP 接続パラメータは以下のとおりです。

パラメータ	値
接続名	この接続の名前で、デフォルトは CON01。 注意：接続名を使用する場合は、データアロケーションセクションのシノニムの CONNECTION 属性で接続名を指定する必要があります。
ホスト名	FTP サーバの名前または IP アドレス
初期ディレクトリ	最初の接続での FTP サーバのディレクトリ
ユーザ	FTP サーバへ接続するユーザ ID または匿名
パスワード	FTPへ接続するパスワードまたは Email アドレス
プロファイル	この接続を追加するサーバのプロファイル

SFTP プロトコルによるバルクロードで複数ユーザを構成するには

SFTP を使用してバルクロードを構成する場合、自分自身のユーザ ID (例、user1) で補助接続を作成します。ユーザが別のユーザ ID (例、user2) でサーバに接続し、上記と同一の補助接続を使用してバルクロードを開始する場合は、user2 は、user 2 固有のキーファイルを作成し、このキーを user1 ID に関連付ける必要があります。

この手順では、次の表記が使用されています。

- サーバは、dmhost と表記されます。
- SFTP サーバは、sftphost と表記されます。
- サーバに接続するユーザは、userdm と表記されます。
- SFTP サーバへの接続で構成されたユーザは、usersftp と表記されます。

次の手順を実行して、userdm が SFTP 補助接続によるバルクロードを開始できるよう構成します。

手順

1. sftphost で次のコマンドを発行し、キーを格納するディレクトリを作成します。

```
/home/usersftp/$ mkdir .ssh  
/home/usersftp/$ cd .ssh
```

次のコマンドを発行し、認証済みキーを格納する authorized_keys という名前のファイルを作成します。

```
/home/usersftp/$ .ssh $ touch authorized_keys
```

2. dmhost で次のコマンドを発行し、dmhost から sftphost に接続するための usersftp を有効にします。

```
/home/userdm/$ sftp usersftp@sftphost  
sftp> exit
```

3. dmhost で、必要に応じて known_hosts を検証し、このリスト上に sftphost が存在することを確認します。


```
/home/userdm/$ cd .ssh
/home/userdm/.ssh $ ls -ltr -a
/home/userdm/.ssh $ cat known_hosts
/home/userdm/.ssh $ cd
```

このファイルの先頭にリモートホスト名が表示されます。

4. dmhost にキーを作成します。

```
/home/userdm/$ ssh-keygen -t rs
```

5. dmhost で、userdm のパブリックキーを sftphost の authorized_keys ファイルに関連付けます。

```
/home/userdm/$ append id_rsa.pub authorized_keys
```

または

```
/home/userdm/$ cat ~/.ssh/id_rsa.pub | ssh usersftp@sftphost
"cat >> /home/usersftp/.ssh/authorized_keys"
```

これにより、userdm ID を使用して usersftp による sftphost への接続が可能になります。

6. 次のコマンドを使用して、dmhost からパスワードなしで sftphost に SFTP 接続します。

```
/home/userdm/$ sftp usersftp@sftphost
sftp> ls
sftp> exit
```

7. dmhost でサーバを開始します。dmhost でサーバに接続します。

```
http://dmhost:8117
```

接続時に、OPSYS セキュリティプロバイダおよびユーザ ID userdm を使用します。

8. dmhost のサーバプロファイル edasprof.prf から次のコマンドを発行し、sftphost への固定フォーマットファイル接続を確立します。

```
ENGINE FIX SET CONNECTION_ATTRIBUTES  
conTest 'sftp://sftphost'/usersftp,:'mydir1'
```

9. 接続をテストします。結果は次のとおりです。

```
User has read access to the FTP server directory  
User has write access to the FTP server directory.  
Successful test for the connection conTest.
```

UNIX、Windows ファイルシステムでの Fixed-Format File または Delimited File のパス指定

FILEDEF コマンドを使用して、FIXED データセットを割り当てます。

```
FILEDEF FILE1 DISK filename [(LRECL lrecl RECFM recfm)]
```

説明

filename

データファイルのフルパスおよびファイル名を指定します。

lrecl

レコード長 (バイト単位) です。このパラメータはオプションとして指定します。省略した場合、メタデータ定義に基づいて計算されます。

オプションとして指定するパラメータの前には、左括弧を入れる必要があります。

recfm

レコードのフォーマットです。固定フォーマットには F、可変フォーマットには V を指定します。このパラメータはオプションとして指定します。このパラメータを省略すると、デフォルトの固定フォーマットになります。

Fixed-Format File および Delimited File のメタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造と、Fixed-Format File および Delimited File のデータタイプのサーバマッピングを定義するシノニムを作成します。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な Fixed-Format File および Delimited File の一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

Fixed-Format File のシノニムの作成

Fixed-Format File はシーケンシャルファイルです。各フィールドは、レコード内で事前に定義された位置に格納されます。固定長フォーマットを保持するために、データの存在しない位置に空白を配置する必要があります。

Fixed-Format File のシノニムを作成するには

シノニムを作成するには、事前にアダプタを構成しておく必要があります。シノニムは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスク

トップインターフェースから作成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
 - **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
 - **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
 - **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
 - **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。
3. シノニムの作成パラメータの参照の箇所で説明しているように、アダプタで必要になるパラメータ用の値を入力します。
 - Windows、UNIX の場合、詳細は、[Windows、UNIX ファイルシステムでの Fixed-Format File のシノニム作成パラメータ](#)を参照してください。
 4. パラメータ値を入力後、[シノニムの作成] をクリックします。

[ステータス] ウィンドウに、シノニムが正しく作成されたことを示す情報が表示されます。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Windows、UNIX ファイルシステムでの Fixed-Format File のシノニム作成パラ メータ

以下のリストは、アダプタのシノニムを作成する際に、値の指定が必要なパラメータ、および実行する関連作業を示しています。これらのオプションが複数のウィンドウに表示される場合もあります。ウィンドウに表示されたボタンをクリックして次のウィンドウに進み、最後に [シノニムの作成] ボタンをクリックします。これで、入力データに基づいてシノニムが作成されます。

COBOL FD

複数の COBOL ファイルを含むディレクトリへのパス。

データファイル割り当ての選択

2 つのオプションのいずれかを選択します。

- **単一ファイル** 単一のデータファイルを定義します。このオプションはリモート接続には適用されません。
- **ファイルコレクション** 単一のシノニムによって定義される一連のデータファイルを定義します。

データプロセスモードの選択

データ検索タイプを示します。

- **1 回だけ取得** シーケンシャルファイルのシノニムを作成するためにデータを 1 回だけ検索します。
- **ファイルリスナ** 継続して検索することができます。

ファイルリスナ属性

新しいファイルのリスナディレクトリポーリング間隔です。

ファイルが存在することをリスナが確認する間隔を秒数で指定します。デフォルト値は 10 秒です。

新しいファイルのリスナディレクトリタイムアウト間隔です。

リスナのタイムアウト間隔を秒数で指定します。デフォルト値は 10 秒です。

リクエストで処理するレコードの最大数です。

1 回のリクエストで処理されるレコード数を指定します。このオプションは単一ファイルの処理に関して指定できます。

リクエストで処理するデータファイルの最大数です。

1 回のリクエストで処理されるファイル数を指定します。このオプションは、ファイルのコレクションの処理に使用できます。デフォルト値は 99,999,999 です。

データファイルピックアップ条件

ファイルエージェントがディレクトリからファイルを取得するメカニズムを指定します。

- **即時** ファイルの検出直後に処理します。
- **トリガ** トリガファイルの検出直後に処理します。このオプションを使用するには、トリガファイル拡張子を設定する必要があります。

トリガファイル拡張子

トリガファイルの拡張子を指定します。トリガファイル拡張子は、リスナの対象となるファイルのフルネームに追加されます。

処理済みデータファイル破棄条件

ファイルエージェントによる処理後のファイルの処理方法を指定します。

- **削除** ファイルを削除します。
- **アーカイブ** タイムスタンプ付きファイルを [ファイルリスナアーカイブディレクトリ] オプションで定義された指定ディレクトリへコピーします。
- **保持** ファイルを保持します。

ファイルリスナアーカイブディレクトリ

[処理済みデータファイル破棄条件] を [アーカイブ] に設定した場合に、データファイルのアーカイブ先を定義します。アプリケーションディレクトリ、マッピングされたアプリ

ケーション名、ファイルシステム上のディレクトリのいずれかを指定することができます。

ファイルシステム上のローカルファイルが格納されるディレクトリを使用する場合は、物理ディレクトリの名前を指定することができます。

ファイルシステム上のリモートデータファイルが格納されるディレクトリを使用する場合は、初期ディレクトリの相対ディレクトリを指定することができます。たとえば、リモート初期ディレクトリが /home/user1/apps の場合、「sales」と指定すると、データファイルの格納先ディレクトリは /home/user1/apps/sales に設定されます。

コレクションの定義属性

ディレクトリまたはアプリケーション

ポーリングするデータファイルのディレクトリを定義します。アプリケーションディレクトリ、マッピングされたアプリケーション名、ファイルシステム上のディレクトリのいずれかを指定することができます。

ファイルシステム上のローカルファイルが格納されるディレクトリを使用する場合は、物理ディレクトリの名前を指定することができます。

ファイルシステム上のリモートデータファイルが格納されるディレクトリを使用する場合は、初期ディレクトリの相対ディレクトリを指定することができます。たとえば、リモート初期ディレクトリが /home/user1/apps の場合、「sales」と指定すると、データファイルの格納先ディレクトリは /home/user1/apps/sales に設定されます。

ファイル選択マスク属性

これらの属性によってデータファイルのリストを制限することができます。

名前

ワイルドカード文字 (%) を含めた完全名または部分名を入力します。完全名を入力すると、その項目のみが返されます。ワイルドカード文字を使用すると、多くのファイル名が検索されます。

拡張子

ワイルドカード文字 (%) の有無に関係なく、拡張子を入力します。

シノニムフィールド名の処理オプション

フィールド名の確認

すべての特殊文字をアンダースコア (_) に変換するとともに、予約語の使用を回避するために名前の確認を行う場合は、[フィールド名の確認] のチェックをオンにします。これに

より、名前に番号が付加されます。このパラメータにより、規則に一致した名前が付けられます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

[フィールド名の確認] のチェックをオフにすると、ハイフン (-)、ブランク、円記号 (¥)、フォワードスラッシュ (/)、カンマ (,)、ドル記号 (\$) の特殊文字のみがアンダースコア (_) に変換されます。名前の確認は行われません。

ユニークフィールドにする

フィールド名およびグループ名の範囲をシノニム全体に設定する場合、[ユニークフィールドにする] のチェックをオンにします。これにより、シノニムの異なるセグメントであっても重複名は使用されません。このチェックをオフにすると、範囲はセグメントに設定されます。

シノニム名

シノニム名を示します。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスはブランクにしておきます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

データフォーマット属性詳細

CODEPAGE

このオプションは、異なるコード化システムでロードされたデータ処理を有効にして、サーバ間のデータ移行を可能とします。

[データフォーマット属性詳細] のチェックをオンにすると、[CODEPAGE] オプションが表示されます。

表示された入力ボックスに、保存されているデータのコードページを入力します。生成されたシノニムのマスターファイルに、入力が増加されます。

注意：ここで指定するコードページは、NLS 構成の一部として設定したカスタムコードページの変換テーブルに一致させる必要があります。特定のコードページの変換テーブルを設定していない場合は、次の方法で設定することができます。WebFOCUS Reporting Server ブラウザインターフェースの [ワークスペース] ページで、メニューバーから [設定] をクリックし、[LOCALE]、[構成ウィザード] を順に選択します。[NLS 構成ウィザード] が開きます。[コードページ変換テーブルのカスタマイズ] をクリックして、使用するコードページのチェックボックスを選択します。[NLS ファイルを再作成/保存して再起動] ボタンをクリックして、新しい設定を有効にします。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

Delimited File のシノニムの作成

Delimited File は、各フィールドが特殊文字で区切られ、事前に定義された順序で格納されたシーケンシャルファイルです。各フィールドは、区切り文字を使用して、ファイルで事前に定義された順序どおりに配置する必要がありますが、データの存在しない位置に空白を配置する必要はありません。

以下の 4 つのフィールド区切り文字があります。

- カンマ
- タブ
- | (パイプ文字)
- トークン - データソースのシノニムで定義されている文字を組み合わせたものです。トークンで区切られたファイルは、JOIN には使用できません。

Delimited File アダプタを使用して、さまざまなタイプのシノニムを作成することができます。

- サーバまたはローカルマシン上の単一ファイルのシノニム
- 1 回だけ取得するファイルコレクションのシノニム
- ファイルリスナのファイルコレクションのシノニム
- 複数シノニム

Delimited File のシノニムを作成するには

シノニムを作成するには、事前にアダプタを構成しておく必要があります。シノニムは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースから作成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースのメニューバーで [アプリケーション] をクリックし、[新規] から [シノニム] を選択します。

または

ibi Data Migrator デスクトップインターフェースで、アプリケーションディレクトリを右クリックし、[新規]、[シノニム] を順に選択します。

[Delimited Flat File シノニムデータファイルの選択] ウィンドウが開きます。

2. シノニムの作成パラメータの参照の箇所で説明しているように、アダプタで必要になるパラメータ用の値を入力します。

3. パラメータ値を入力後、[シノニムの作成] をクリックします。

[ステータス] ウィンドウに、シノニムが正しく作成されたことを示す情報が表示されます。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムは、[アダプタ] ページから作成することもできます。その場合は、構成済みアダプタの接続を右クリックし、[シノニムの作成] を選択します。

Delimited File のシノニム作成パラメータ

以下のリストは、アダプタのシノニムを作成する際に、値の指定が必要なパラメータ、および実行する関連作業を示しています。これらのオプションが複数のウィンドウに表示される場合もあります。[次へ] ボタンをクリックして、ウィンドウ間を移動します。最後に [シノニムの作成] ボタンをクリックして、入力した情報に基づいてシノニムを作成します。

複数シノニムの作成

[複数シノニム] のチェックボックスを選択します。

データファイルパス

複数のデータファイルへのパス。

ファイルパス

データファイルの格納先を指定します。

ドロップダウンメニューから以下のいずれかを選択します。

- サーバサイド
- ローカルマシン

データファイル

サーバ上のデータファイルへの絶対パスまたは最初のファイルへの相対ディレクトリを定義します。データファイルの名前を含みます。

パスとファイル名を入力するか、[...] (参照) ボタンをクリックしてファイルを選択します。

ファイルのアップロード

[...] (参照) ボタンをクリックして、アップロードするデータファイルを選択します。

データファイル割り当ての選択

次のいずれかを選択します。

- **単一ファイル** 単一のデータファイルを定義します。このオプションはリモート接続には適用されません。
- **ファイルコレクション** 単一のシノニムによって定義される一連のデータファイルを定義します。

データプロセスモードの選択

以下のデータ検索タイプを指定します。

- **1 回だけ取得** シーケンシャルファイルのシノニムを作成するためにデータを 1 回だけ検索します。
- **ファイルリスナ** 継続して検索することができます。

レコードの指定

区切り文字

ファイルレコードまたはデータ行 (区切られたフィールドのグループなど) の区切りを示す文字または一連の文字を定義します。

この属性は RDELIMITER としてアクセスファイルに反映されます。

非表示のレコード区切り文字

区切り文字フィールド内の文字列が表示可能な文字列であるか表示不可の 10 進文字であるかを定義します。表示不可の 10 進文字はアクセスファイルでは 16 進値として表示 (変換) されることに注意してください。区切り文字が 16 進として入力される場合は、この属性を [No] に設定します。

フィールドの指定

区切り文字

ファイルのフィールド区切りを示す文字または一連の文字を定義します。

複数の文字を使用するには、それらを一緒に入力します。たとえば、2 つのスラッシュ(//)を入力します。

タブ文字を区切り文字として使用するには、予約語 TAB を入力します。

あるいは、16 進のコードを直接入力することもできます。たとえば、ラインフィード (LF) とキャリッジリターン (CR) はそれぞれ 0x0A、0x0D となります。

区切り文字が複数の文字から成る場合、単一の文字列を入力します。たとえば、ラインフィード (LF) とキャリッジリターン (CR) は 0x0A0D と入力します。

非表示の区切り文字

区切り文字フィールド内の文字列が表示可能な文字列であるか表示不可の 10 進文字であるかを定義します。表示不可の 10 進文字はアクセスファイルでは 16 進値として表示 (変換) されることに注意してください。

区切り文字が 16 進値として入力される場合は、この属性を [No] に設定します。

この属性を [Yes] に設定する場合、[区切り文字] 属性の値は 10 進値として入力する必要があります。たとえば、ラインフィードなどの表示不可の区切り文字については、10 進値の 10 を使用します。複数の非表示文字で区切り文字が構成される場合、10 進値の間に空白を挿入します。たとえば、ラインフィードとキャリッジリターンを一緒に指定するには、10 13 と入力します。

ヘッダ行

このチェックボックスを選択すると、シノニムのカラム名にヘッダ行の名前が使用されます。

スキップする先頭の行数

シノニムの作成時およびデータの読み込み時に無視する、ヘッダ行より前の行数を指定します。

囲み文字

文字カラムが引用符または他の記号で囲まれている場合、シノニムを作成する区切り付きファイルの文字カラムフォーマットに囲み文字を入力します。たとえば、ファイル内の文字カラムが二重引用符 (") で囲まれている場合、[囲み文字] テキストボックスに "\"" と入力します。

レコードスキャン制限

カラムの長さを特定するため、DFIX データファイルからスキャンするレコードの件数を指定します。

このパラメータを省略した場合、ファイルのすべてのレコードが処理されます。

フォーマットを保持する

元のデータレイアウト (空のレコードやラインフィードを含む) を保持するかどうかを指定します。

- **No** - 元の文字は表示可能な文字列で置き換えられます。これがデフォルト値です。
- **Yes** - 表示不可の文字やラインフィードなどの元の文字は保持されます。

この属性は PRESERVEFRMT としてアクセスファイルに反映されます。

ファイルリсна

新しいファイルのリスナディレクトリポーリング間隔です。

ファイルが存在することをリスナが確認する間隔を秒数で指定します。デフォルト値は 10 秒です。

新しいファイルのリスナディレクトリタイムアウト間隔です。

リスナのタイムアウト間隔を秒数で指定します。デフォルト値は 10 秒です。

リクエストで処理するデータファイルの最大数です。

1 回のリクエストで処理されるファイル数を指定します。デフォルト値は 99,999,999 です。

データファイルピックアップ条件

ファイルエージェントがディレクトリからファイルを取得するメカニズムを指定します。

- **即時** ファイルの検出直後に処理します。
- **トリガ** トリガファイルの検出直後に処理します。このオプションを使用するには、トリガファイル拡張子を設定する必要があります。

トリガファイル拡張子

トリガファイルの拡張子を指定します。トリガファイル拡張子は、リスナの対象となるファイルのフルネームに追加されます。

処理済みデータファイル破棄条件

ファイルエージェントによる処理後のファイルの処理方法を指定します。

- **削除** ファイルを削除します。
- **アーカイブ** タイムスタンプ付きファイルを [ファイルリスナアーカイブディレクトリ] オプションで定義された指定ディレクトリへコピーします。
- **保持** ファイルを保持します。

ファイルリスナアーカイブディレクトリ

データファイルがアーカイブされるサーバ上のアーカイブディレクトリ。絶対パスまたは最初のファイルへの相対ディレクトリを使用することができます。

コレクションの定義

ディレクトリまたはアプリケーション

データファイルのロケーションタイプを定義します。ローカルファイルの物理ディレクトリまたはリモートデータファイルの初期ディレクトリへの相対ディレクトリを指定できます。たとえば、リモート初期ディレクトリが /home/user1/apps の場合、コレクションの定義で「sales」と指定すると、データファイルの格納先ディレクトリは /home1/user1/apps/sales に設定されます。

ファイル選択マスク

名前

ワイルドカード文字 (%) を含めた完全名または部分名を入力します。完全名を入力すると、その項目のみが返されます。ワイルドカード文字を使用すると、多くのファイル名が検索されます。

拡張子

ワイルドカード文字 (%) の有無に関係なく、拡張子を入力します。

シノニムフィールド名の処理オプション

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有する必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

ファイルを選択

[デフォルトシノニム名] の左のチェックボックスを選択して、シノニムを作成したいファイルを選択します。チェックボックスを選択すると、対応するデータファイルが反転表示されます。

[すべて選択] をクリックすると、そのディレクトリパス内のすべてのデータファイルが選択されます。

データフォーマット属性詳細

CODEPAGE

このオプションは、異なるコード化システムでロードされたデータ処理を有効にして、サーバ間のデータ移行を可能とします。

[データフォーマット属性詳細] チェックボックスを選択して、[CODEPAGE] オプションを表示します。

表示された入力ボックスに、保存されているデータのコードページを入力します。生成されたシノニムのマスターファイルに、入力が追加されます。

注意：ここで指定するコードページは、NLS 構成の一部として設定したカスタムコードページの変換テーブルに一致させる必要があります。特定のコードページの変換テーブルを設定していない場合は、次の方法で設定することができます。WebFOCUS Reporting Server ブラウザインターフェースの [ワークスペース] ページで、[設定] をクリックし、[LOCALE]、[構成ウィザード] を順に選択します。[NLS 構成ウィザード] が開きます。[コードページ変換テーブルのカスタマイズ] をクリックして、使用するコードページのチェックボックスを選択します。[NLS ファイルを再作成/保存して再起動] ボタンをクリックして、新しい設定を有効にします。

DFIX データファイルおよび生成されたシノニムのサンプル

以下は、サーバ上の単一ファイルに対する単一シノニムの例です。

DFIX データファイル (カンマ区切り)

```
ENGLAND,JAGUAR,V12XKE AUTO,CONVERTIBLE,2,7427,8878.0
ENGLAND,JAGUAR,XJ12L AUTO,SEDAN,5,11194,13491,12000
ENGLAND,JENSEN,INTERCEPTOR III,SEDAN,4,14940,17850,0
ENGLAND,TRIUMPH,TR7,HARDTOP,2,4292,5100,0
JAPAN,DATSUN,B210 2 DOOR AUTO,SEDAN,4,2626,3139,4300
```



```
JAPAN,TOYOTA,COROLLA 4 DOOR DIX AUTO,SEDAN,4,2886,3339,35030
ITALY,ALFA ROMEO,2000 4 DOOR BERLINA,SEDAN,4,4915,5925,4800
ITALY,ALFA ROMEO,2000 GT VELOCE,COUPE,2,5660,6820,12400
ITALY,ALFA ROMEO,2000 SPI,2,25000,31500,0,13000
ITALY,MASERATI,DORA 2 DOOR,COUPE,2,25000,31500,0
GERMANY,AUDI,100 LS 2 DOOR AUTO,SEDAN,5,5063,5970,7800
GERMANY,BMW,2002 2 DOOR,SEDAN,5,5800,5940,8950
GERMANY,BMW,2002 2 DOOR AUTO,SEDAN,4,6000,6355,8900
GERMANY,BMW,3.0 SI 4 DOOR,SEDAN,5,10000,13752,14000
GERMANY,BMW,3.0 SI 4 DOOR AUTO,SEDAN,5,11000,14123,1894
GERMANY,BMW,530I 4 DOOR,SEDAN,5,8300,9097,14000
GERMANY,BMW,530I 4 DOOR AUTO,SEDAN,5,8400,9495,1560
FRANCE,PEUGEOT,504 4 DOOR,SEDAN,5,4631,5610,0
```

DFIX カンマ区切りデータファイルのシノニムを作成するには、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの [シノニムの作成] ウィンドウに以下の情報を入力します。

1. ディレクトリパスフィールドに、Delimited Flat File が格納されているディレクトリ名を入力します。
2. ファイル名 (たとえば、dfix_car) を入力します。
3. ファイルの拡張子フィールド (たとえば、.dat) を入力します。
4. [ファイルの選択] をクリックします。
5. シノニム作成の 2 番目のウィンドウで、表示されたデータファイルのリストからファイルを選択し、[区切り文字] テキストボックスに区切り文字としてカンマ (,) を指定します。
6. [シノニムの作成] をクリックします。シノニムが作成され、指定したアプリケーションディレクトリに追加されます。デフォルトのディレクトリは baseapp です。
ステータスウィンドウに「全シノニムを作成しました」というメッセージが表示されます。ステータスウィンドウに「全シノニムを作成しました」というメッセージが表示されます。
7. [メッセージ] ウィンドウで、[アプリケーション] をクリックします。
8. ナビゲーションウィンドウの baseapp アプリケーションフォルダを開いて、シノニム (この例では dfix_car) をクリックします。
9. メニューから [テキストとして編集] を選択して、生成されたマスターファイルを表示し、次に [アクセスファイルをテキストとして編集] を選択して、対応するアクセスファイルを表示します。

作成されたマスターファイル

```

FILENAME=CAR_DFIX, SUFFIX=DFIX      ,
DATASET=directory/car_dfix.dat, $
SEGMENT=CAR_DFIX, SEGTYPE=S0, $
  FIELDNAME=FIELD_1, USAGE=A7, ACTUAL=A7BV, $
  FIELDNAME=FIELD_2, USAGE=A10, ACTUAL=A10BV, $
  FIELDNAME=FIELD_3, USAGE=A23, ACTUAL=A23BV, $
  FIELDNAME=FIELD_4, USAGE=A11, ACTUAL=A11BV, $
  FIELDNAME=FIELD_5, USAGE=I1, ACTUAL=A1, $
  FIELDNAME=FIELD_6, USAGE=I5, ACTUAL=A5V, $
  FIELDNAME=FIELD_7, USAGE=I5, ACTUAL=A5V, $
  FIELDNAME=FIELD_8, USAGE=I5, ACTUAL=A5V, $

```

生成されたマスターファイルの FILE 定義に、以下の属性が含まれていることに注意してください。

```
SUFFIX=DFIX
```

作成されたアクセスファイル

```
SEGNAME=DFIX_CAR, DELIMITER=',', HEADER=NO,$
```

区切り文字には、表示可能な単一文字が指定されています。

ヘッダおよび囲み文字を使用したサンプルの DFIX データファイル

このサンプルファイルにはヘッダ行が指定され、囲み文字として二重引用符 (") が使用されています。

```

"COUNTRY","CAR","MODEL","BODYTYPE","SEATS","DEALER_COST","RETAIL_
COST","SALES"
"ENGLAND","JAGUAR","V12XKE AUTO","CONVERTIBLE",2,7427,8878,0
"ENGLAND","JAGUAR","XJ12L AUTO","SEDAN",5,11194,13491,12000
"ENGLAND","JENSEN","INTERCEPTOR III","SEDAN",4,14940,17850,0
"ENGLAND","TRIUMPH","TR7","HARDTOP",2,4292,5100,0
"FRANCE","PEUGEOT","504 4 DOOR","SEDAN",5,4631,5610,0
"ITALY","ALFA ROMEO","2000 4 DOOR BERLINA","SEDAN",4,4915,5925,4800
"ITALY","ALFA ROMEO","2000 GT VELOCE","COUPE",2,5660,6820,12400
"ITALY","ALFA ROMEO","2000 SPIDER VELOCE","ROADSTER",2,5660,6820,13000
"ITALY","MASERATI","DORA 2 DOOR","COUPE",2,25000,31500,0

```

```
"JAPAN","DATSUN","B210 2 DOOR AUTO","SEDAN",4,2626,3139,43000
"JAPAN","TOYOTA","COROLLA 4 DOOR DIX AUTO","SEDAN",4,2886,3339,3503
"GERMANY","AUDI","100 LS 2 DOOR AUTO","SEDAN",5,5063,5970,7800
"GERMANY","BMW","2002 2 DOOR","SEDAN",5,5800,5940,8950
"GERMANY","BMW","2002 2 DOOR AUTO","SEDAN",4,6000,6355,8900
"GERMANY","BMW","3.0 SI 4 DOOR","SEDAN",5,10000,13752,14000
"GERMANY","BMW","3.0 SI 4 DOOR AUTO","SEDAN",5,11000,14123,18940
"GERMANY","BMW","530I 4 DOOR","SEDAN",5,8300,9097,14000
"GERMANY","BMW","530I 4 DOOR AUTO","SEDAN",5,8400,9495,15600
```

[シノニムの作成] ウィンドウで、区切り文字としてカンマ (,)、囲み文字として二重引用符 (") を指定します。[ヘッダ] のチェックをオンにして、ヘッダ行の名前をマスターファイルのフィールド名として使用することを指定します。

作成されたマスターファイル

```
FILENAME=CAR_CSV, SUFFIX=DFIX      ,
DATASET=directory/car_csv.csv, $
SEGMENT=CAR_CSV, SEGTYPE=S0, $
  FIELDNAME=COUNTRY, ALIAS=COUNTRY, USAGE=A7, ACTUAL=A7BV, $
  FIELDNAME=CAR, ALIAS=CAR, USAGE=A10, ACTUAL=A10BV, $
  FIELDNAME=MODEL, ALIAS=MODEL, USAGE=A23, ACTUAL=A23BV, $
  FIELDNAME=BODYTYPE, ALIAS=BODYTYPE, USAGE=A11, ACTUAL=A11BV, $
  FIELDNAME=SEATS, ALIAS=SEATS, USAGE=I1, ACTUAL=A1, $
  FIELDNAME=DEALER_COST, ALIAS=DEALER_COST, USAGE=I5, ACTUAL=A5V, $
  FIELDNAME=RETAIL_COST, ALIAS=RETAIL_COST, USAGE=I5, ACTUAL=A5V, $
  FIELDNAME=SALES, ALIAS=SALES, USAGE=I5, ACTUAL=A5V, $
```

区切り文字、囲み文字、ヘッダの各項目は、アクセスファイルに反映されます。

作成されたアクセスファイル

```
SEGNAME=CAR_CSV, DELIMITER=',', ENCLOSURE="", HEADER=YES, $
```

区切り文字の USAGE および ACTUAL フォーマット

バージョン 7.6 SP01 より前のバージョンでは、サーバの区切り文字は、アクセスファイルの DELIMITER パラメータではなく、マスターファイルのフィールド定義で指定されていました。

そのため、旧バージョンのシノニムでは、FIELDNAME=DELIMITER、区切り文字自体の ALIAS、USAGE および ACTUAL の値が指定された構文が使用されている場合があります。

以下はその例です。

```
FIELDNAME=DELIMITER, ALIAS=',', USAGE=A1, ACTUAL=A1, $
```

USAGE および ACTUAL フォーマットは、以下のように区切られます。

区切り文字のタイプ	USAGE	ACTUAL
表示可能な文字	An n は文字数。	An n は文字数。
表示不可の文字 (例、Tab)	I4	I1
グループ (表示可能な文字と表示不可の文字の組み合わせ、または複数の表示不可の文字)	すべての USAGE 長さの合計	すべての ACTUAL 長さの合計

Delimited File の更新

MODIFY 機能を使用して Delimited Flat File を更新することができます。以下の区切り文字が使用されます。

TAB

タブ文字です。これがデフォルト値です。

a

たとえば、「~」のような文字列です。

0xnn

16 進数コードです。たとえば、044 はカンマ (,) を、0x0D0A はリターンとラインフィードを表します。16 進数コードは、Windows および UNIX システムでは ASCII を使用しま

す。

ibi Data Migrator フローのターゲットタイプとして、Delimited File が新しく追加されました。

Google Sheets アダプタの使用

Google Sheets アダプタの構成方法について説明します。

ここでは、Google Sheets アダプタの構成方法について説明します。

Google Sheets アダプタの概要

Google Sheets は、オンラインのスプレッドシートプログラムです。Google Sheets アダプタは、Google Sheets 環境に常駐する情報に対するレポートの作成に使用されます。

Google Sheets アダプタおよび Google Drive アダプタの両方が、Google Sheets 環境でのスプレッドシートの作成に使用されます。

これらのアダプタを構成するには、WebFOCUS Reporting Server ブラウザインターフェースを使用します。これらのアダプタには、アクセストークンとリフレッシュトークンを格納する接続が必要です。

有効な Google Sheets のアクセストークンおよびリフレッシュトークンは、Google Sheets および Google Drive の API コールの発行に必要です。これらのトークンは、Google アプリケーションおよび特定の Google Sheets ユーザに関連付けられます。

Google Sheets プロジェクトの作成

Google Sheets アダプタの構成前に、Google プロジェクトを用意する必要があります。

Google プロジェクトを作成するには

手順

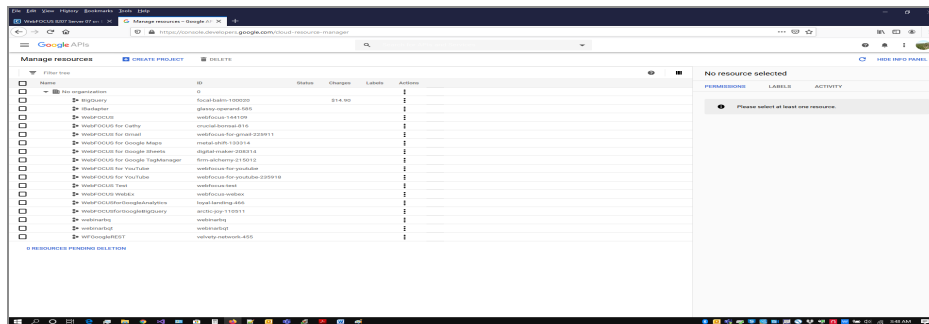
1. Web ブラウザのアドレスバーに次の URL を入力します。

<https://console.developers.google.com/project>

Google アカウントにサインインしていない場合は、Google Developers Console にサインインするためのサインインダイアログが表示されます。

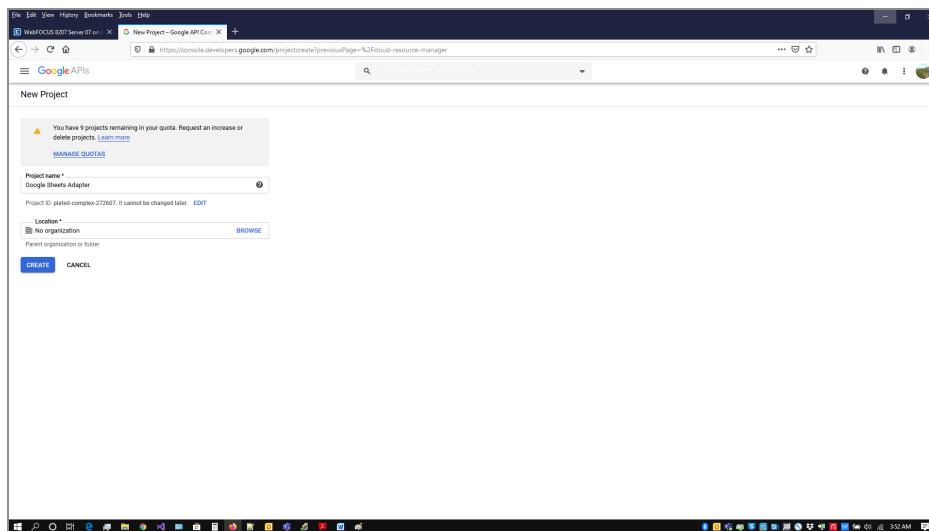
2. Google 認証情報を入力し、サインインします。

下図のように、Google Developers Console の画面が開きます。



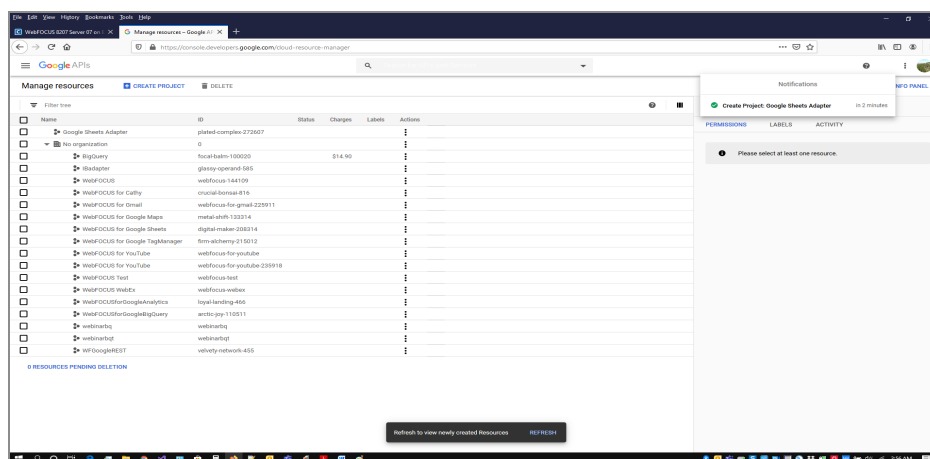
3. [プロジェクトを作成] をクリックします。

下図のように、[新しいプロジェクト] の画面が開きます。

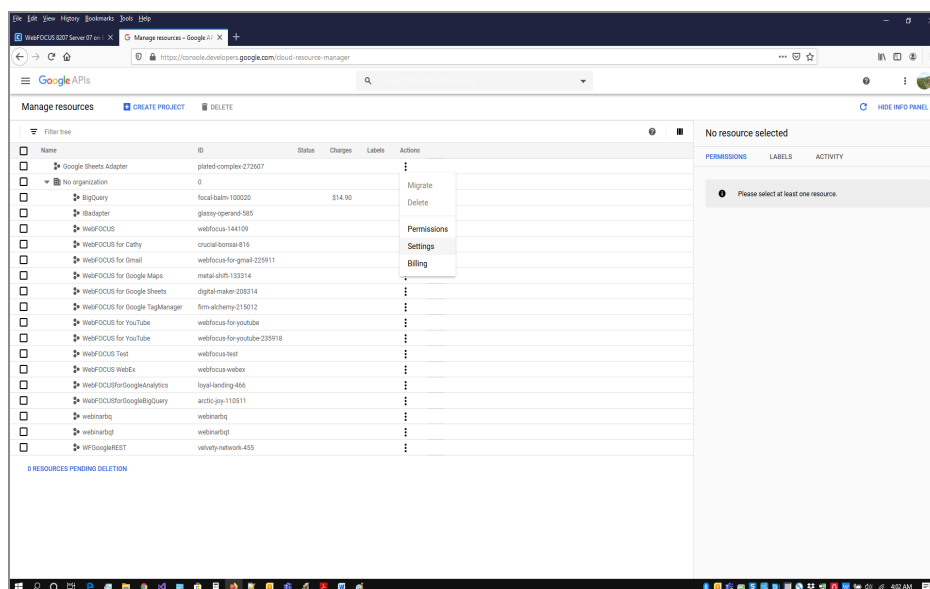


4. プロジェクト名を入力し、[作成] をクリックします。

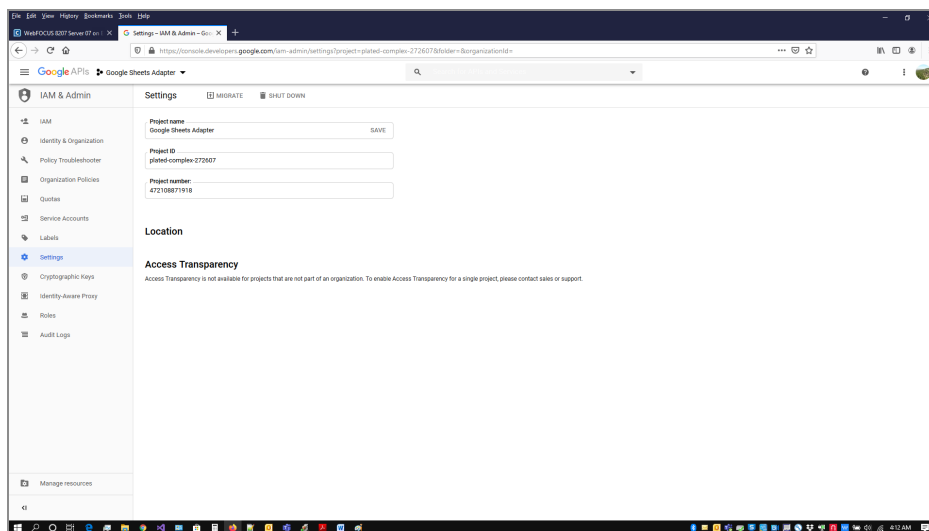
下図のように、プロジェクトが作成されたことを示すページが開きます。



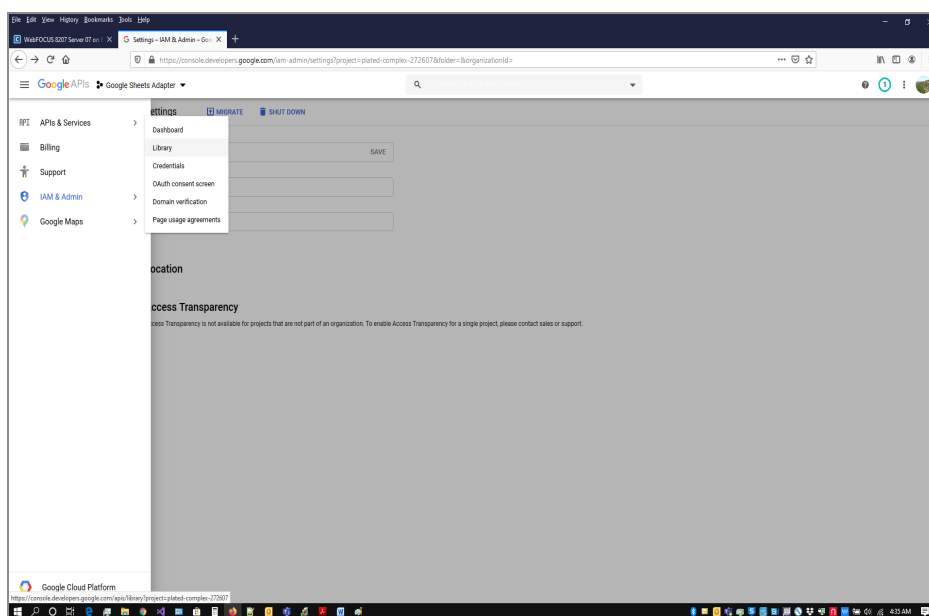
5. 画面下部の [リフレッシュ] をクリックします。
6. 下図のように、新しく作成したプロジェクトの場合、[アクション] 下の [:] (参照メニュー) をクリックし、[設定] を選択します。



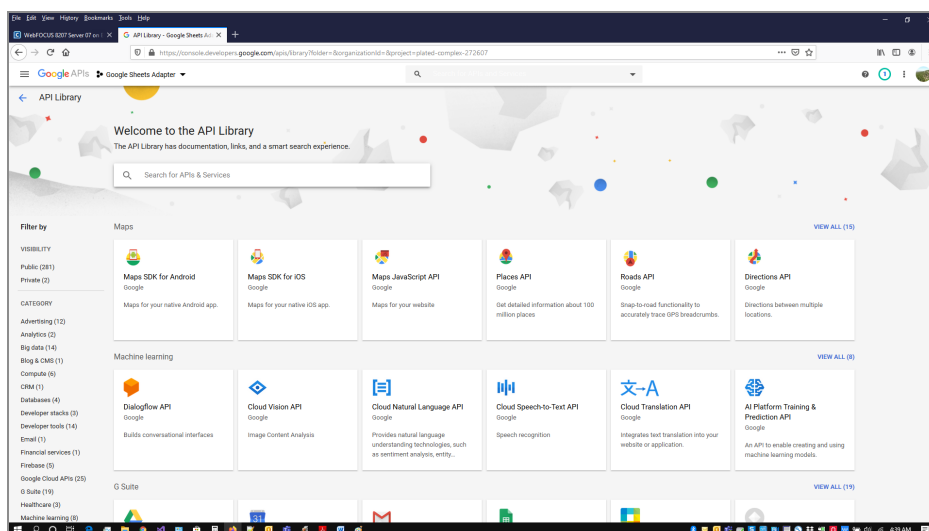
下図のように、Google プロジェクトの [設定] ページが開きます。



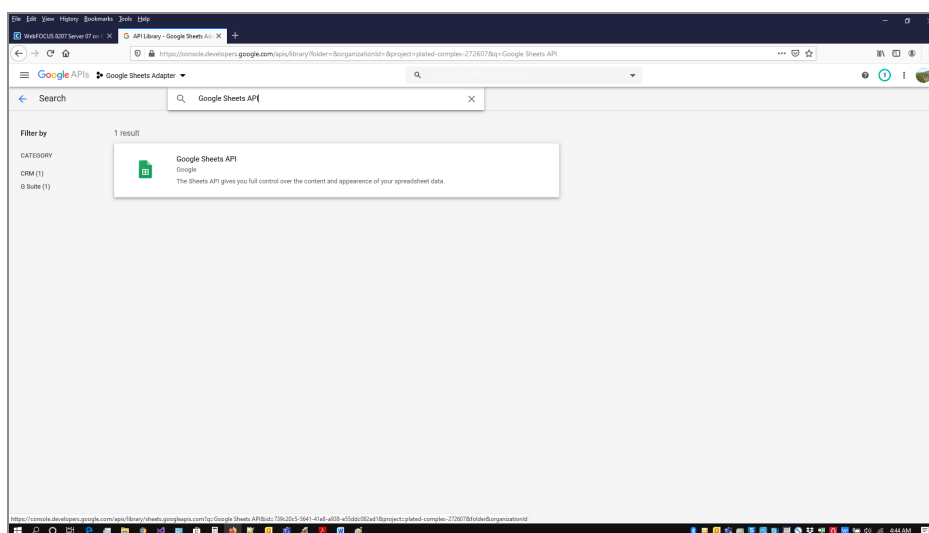
7. 下図のように、画面左上のナビゲーションメニューを示す 3 本の水平バーをクリックし、[APIとサービス]、[ライブラリ] を順に選択します。



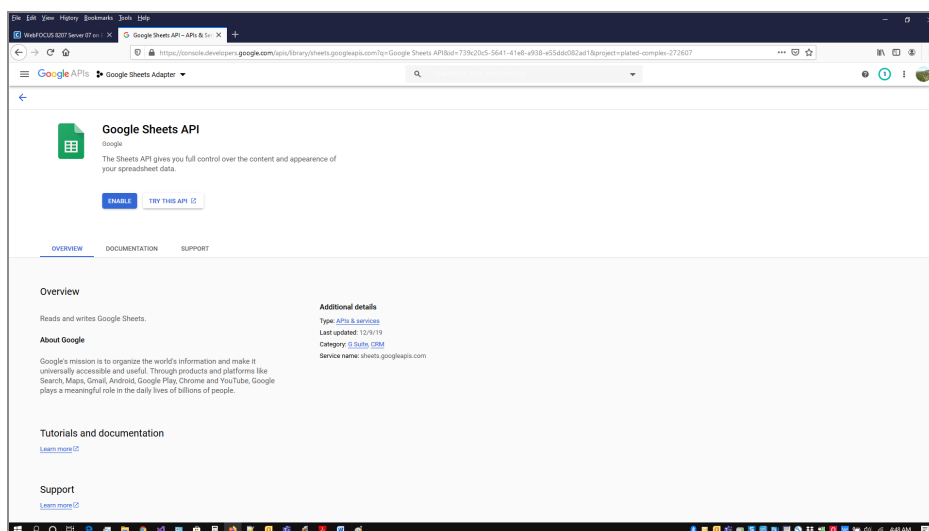
下図のように、API ライブラリの画面が開きます。



8. 下図のように、[検索] テキストボックスに「Google Sheets API」と入力し、検索結果からこれを選択します。

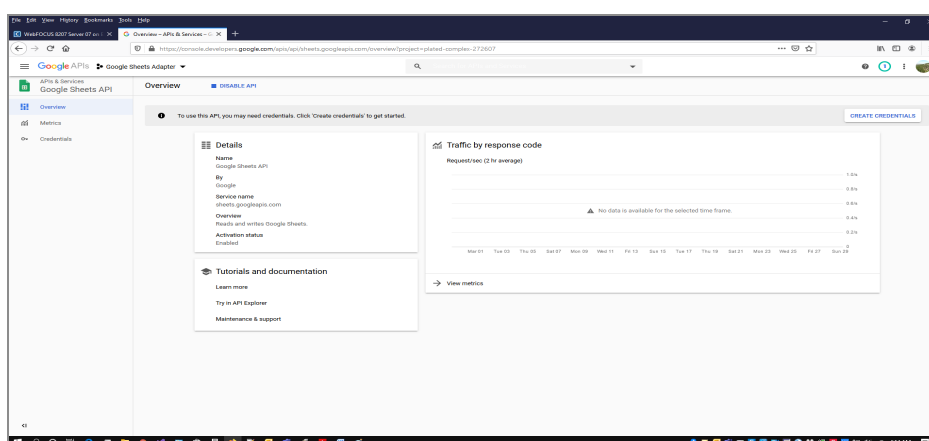


下図のように、Google Sheets API のページが開きます。

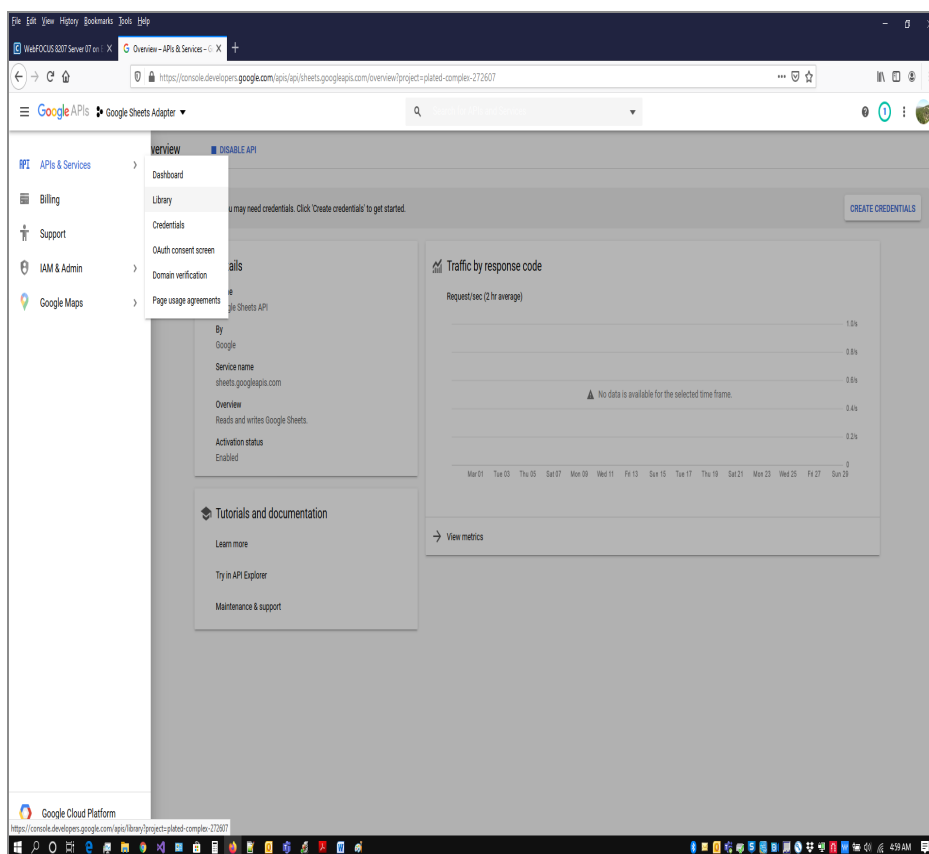


9. [有効にする] をクリックします。

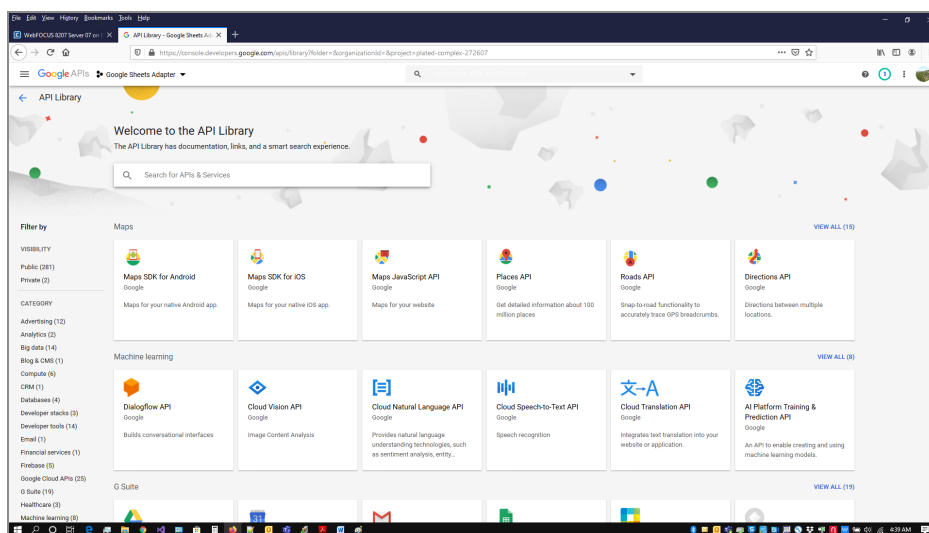
下図のように、Google Sheets API のページが開き、利用情報が表示されます。



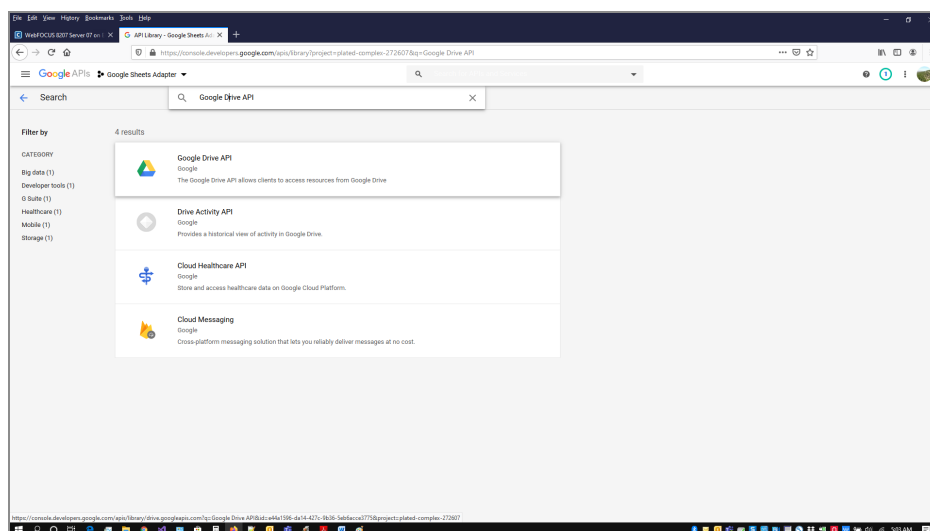
10. 下図のように、画面左上のナビゲーションメニューを示す 3 本の水平バーをクリックし、[APIとサービス]、[ライブラリ] を順に選択します。



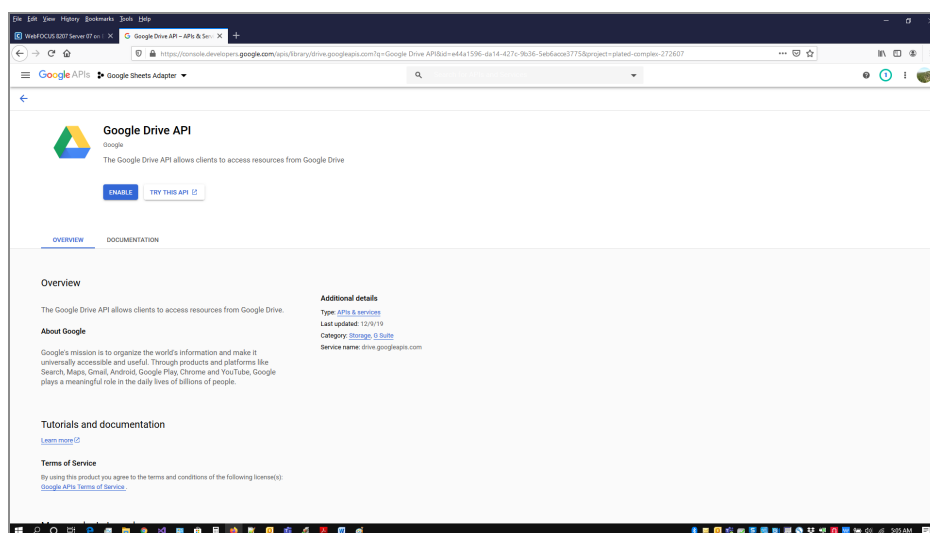
下図のように、API ライブラリのページが開きます。



11. 下図のように、[検索] テキストボックスに「Google Drive API」と入力し、検索結果からこれを選択します。

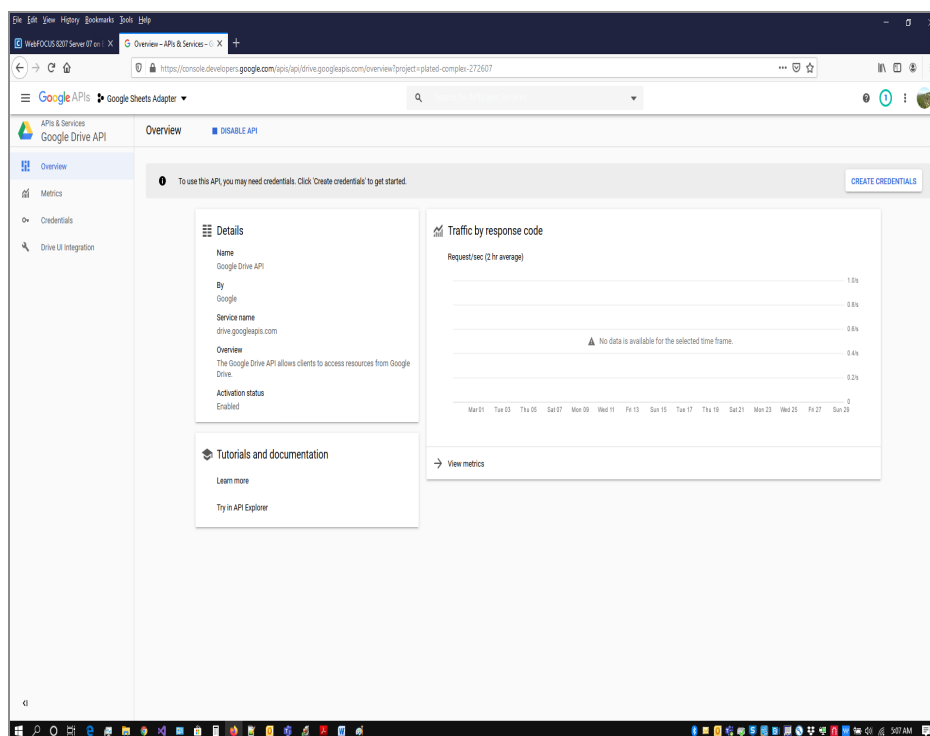


下図のように、Google Drive API のページが開きます。

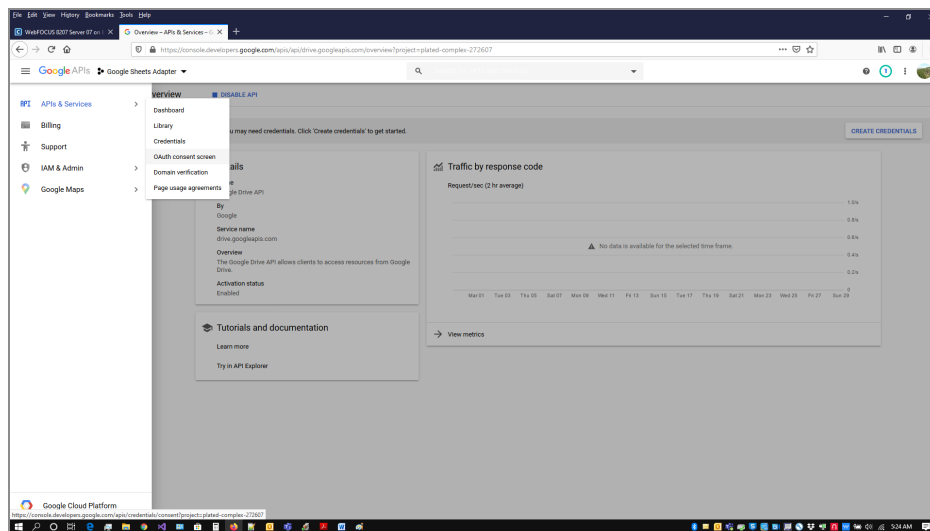


12. [有効にする] をクリックします。

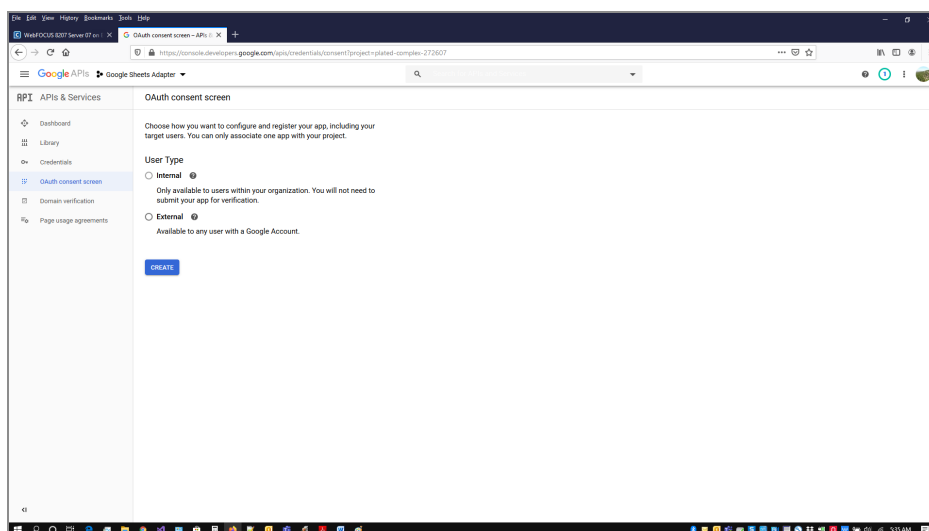
下図のように、Google Drive API のページが開き、利用情報が表示されます。



13. 下図のように、画面左上のナビゲーションメニューを示す 3 本の水平バーをクリックし、[APIとサービス]、[OAuth 同意画面] を順に選択します。



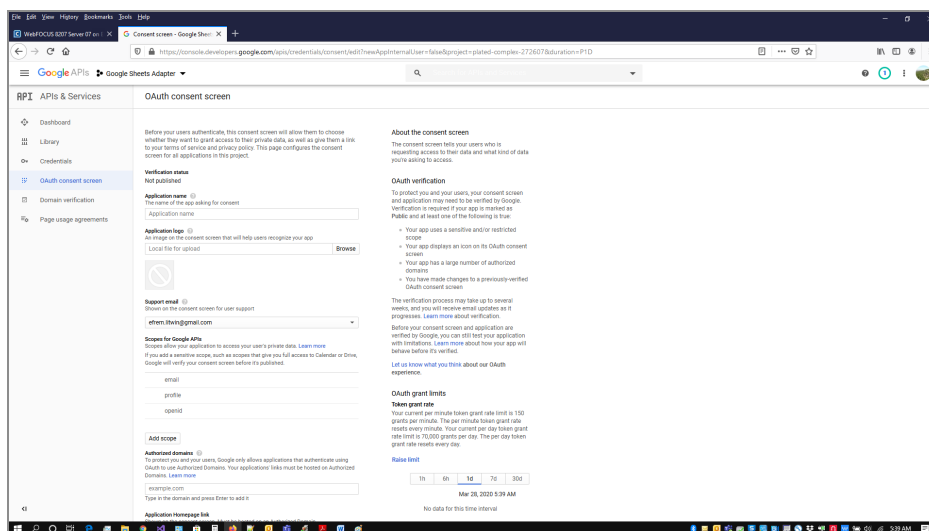
下図のように、OAuth 同意画面のページが開きます。



14. [内部] または [外部] のいずれかを選択し、[作成] をクリックします。

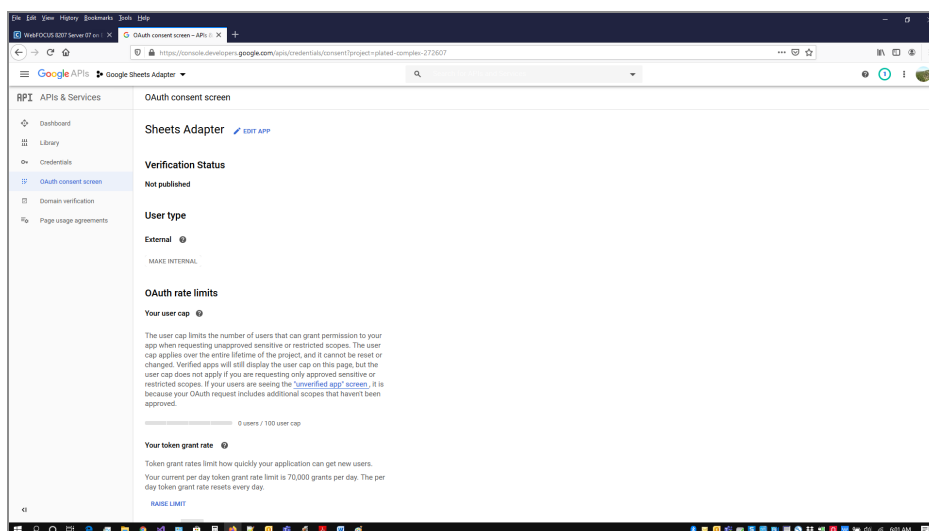
注意：G Suite ユーザのみ [内部] を選択できます。

下図のように、OAuth 同意画面の構成ページが開きます。

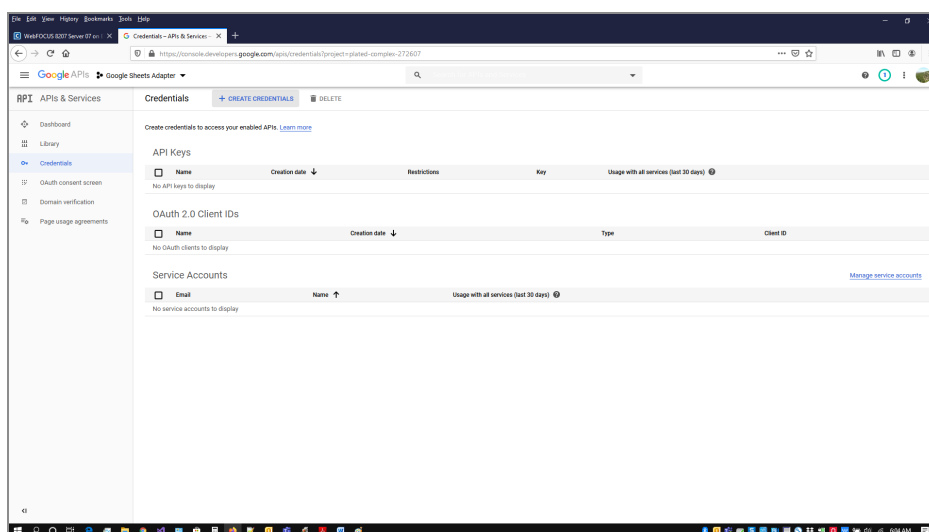


15. [アプリケーション名] を入力し、WebFOCUS Server で使用するドメインを [許可されたドメイン] 下に入力して、ページ下部の [保存] をクリックします。

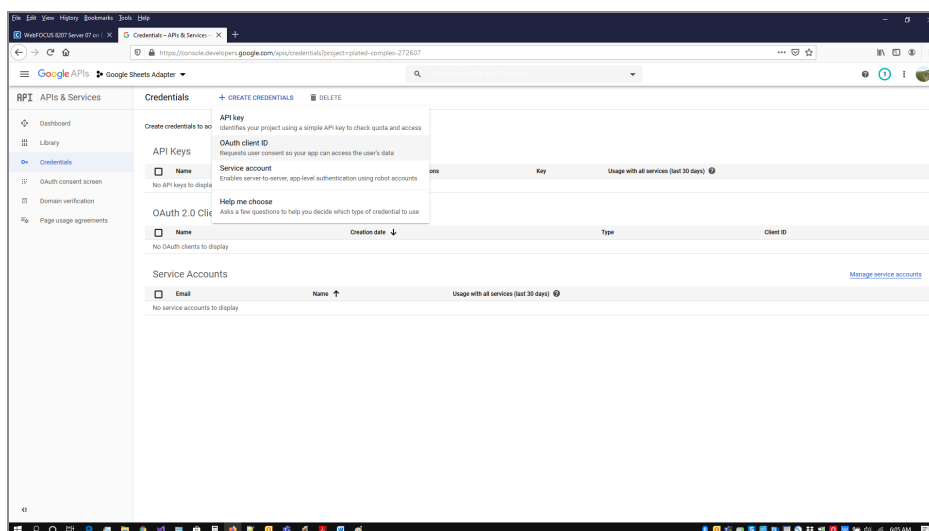
下図のように、OAuth 同意画面のページが開きます。



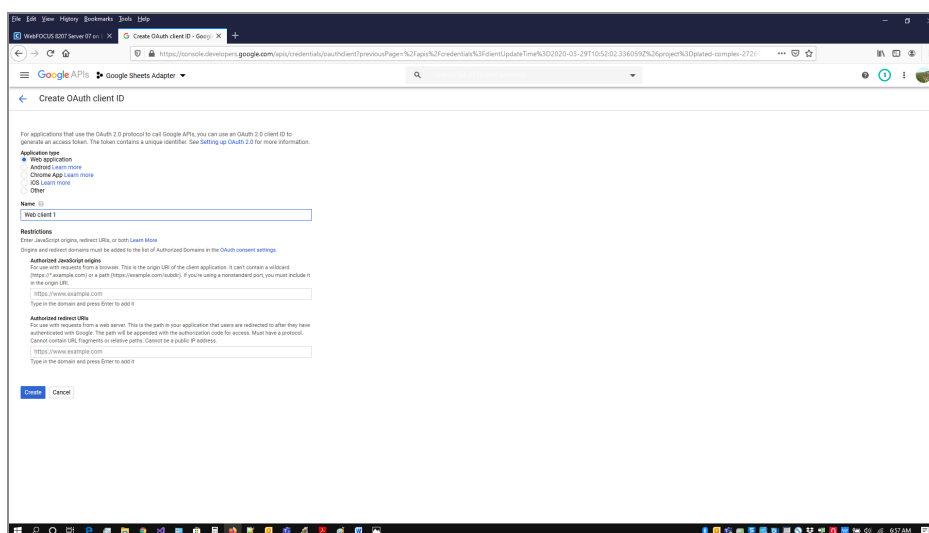
16. 左側ウィンドウで [認証情報] をクリックします。
下図のように、[認証情報] ページが開きます。



17. 下図のように、[認証情報を作成] をクリックし、[OAuth クライアント ID] を選択します。



下図のように、OAuth クライアント ID の作成ページが開きます。



18. 次の手順を実行します。

- a. アプリケーションタイプのリストから、[ウェブアプリケーション] を選択します。
- b. [承認済みの JavaScript 生成元] テキストボックスに、WebFOCUS Reporting Server ブラウザインターフェースへのアクセスに使用するホスト名とポート番号を入力します。

Google プロジェクトでは、HTTPS が要求されます。

以下はその例です。

```
https://host.ibi.com:8121
```

- c. [承認済みのリダイレクト URI] テキストボックスに、WebFOCUS Reporting Server ブラウザインターフェースへのアクセスに使用するホスト名とポート番号を入力し、末尾に `oauth20.exe` を追加します。

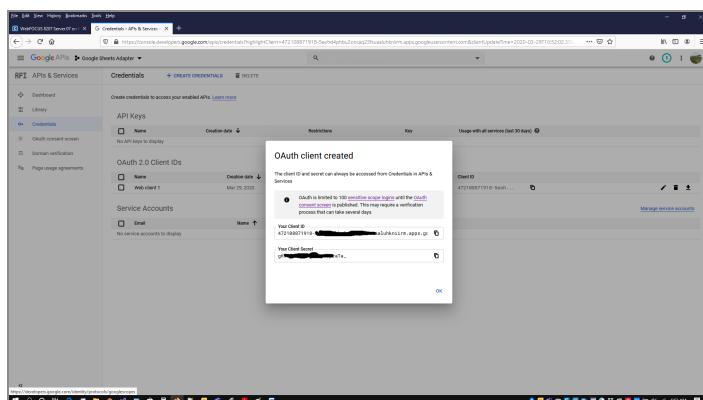
Google プロジェクトでは、HTTPS が要求されます。

以下はその例です。

```
https://host.ibi.com:8121/oauth20.exe
```

19. [作成] をクリックします。

下図のように、OAuth クライアントの画面が開き、クライアント ID とクライアントシークレットが表示されます。



注意：クライアント ID とクライアントシークレットの値は、Google Sheets アダプタの構成に必要です。

20. [OK] をクリックします。

Google Sheets アダプタの構成

ここでは、Google Sheets アダプタの構成方法について説明します。

Google Sheets アダプタを構成するには

手順

1. この構成で使用する Google アカウントからサインアウト済みであることを確認します。
2. Google プロジェクトの [承認済みの JavaScript 生成元] テキストボックスで指定したホスト名とポート番号を使用して、WebFOCUS Reporting Server ブラウザインターフェースにアクセスします。

以下はその例です。

```
https://host.ibi.com:8121
```

詳細は、[Google プロジェクトを作成するには](#)を参照してください。

3. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

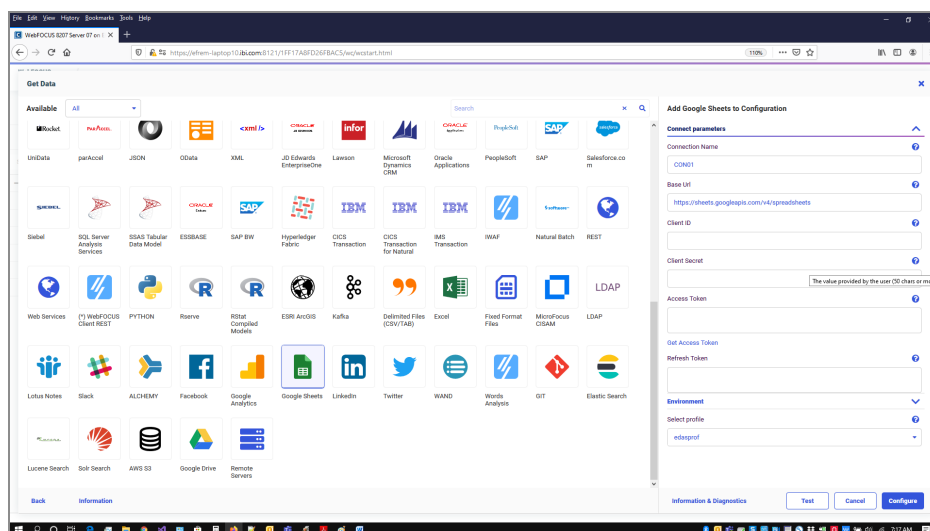
[データの取得] ページが開き、構成済みアダプタが表示されます。

4. プラス (+) ボタンをクリックし、表示されたページでアダプタを検索します。

WebFOCUS Reporting Server ブラウザインターフェースでは、ドロップダウンリストからアダプタのカテゴリ (Google Sheets は [ソーシャルメディア] カテゴリ) を選択することも、検索オプション (拡大鏡) を使用して特定の文字列を検索することもできます。

5. [Google Sheets] ノードを右クリックし、[構成] を選択します。

[Google Sheets の構成の追加] ウィンドウが開きます。下図は、[データの取得] の簡略モードを示しています。



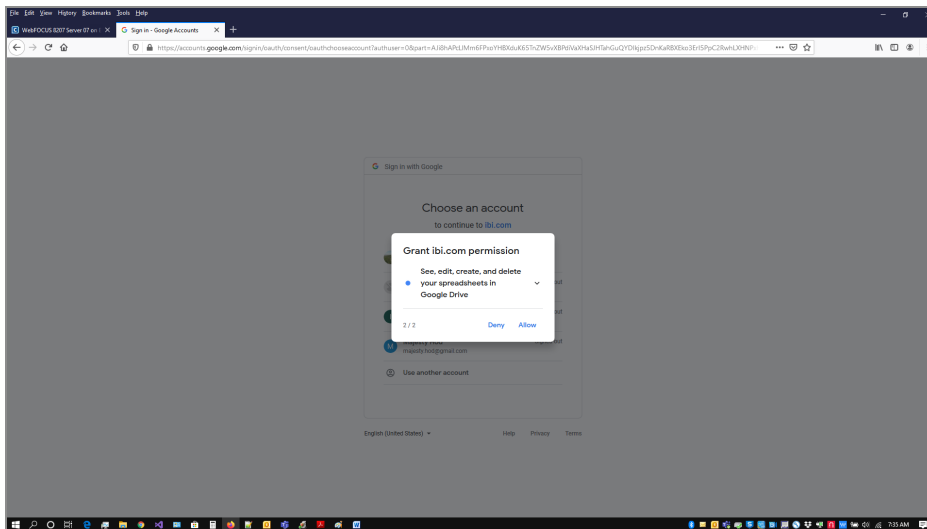
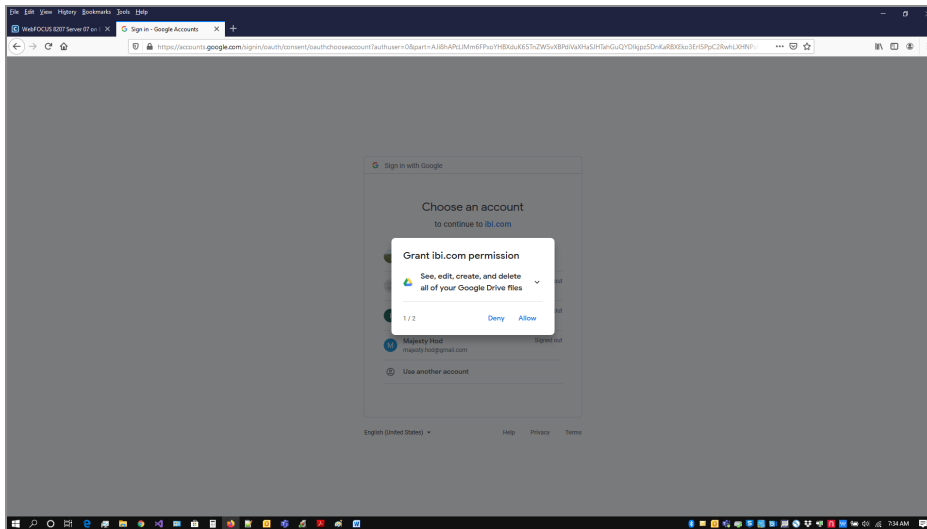
6. [セキュリティ] で認証方法を選択します。
[OAuth] または [サービスアカウント] のいずれかを選択できます。
7. 認証方法に基づいて、次のセキュリティ属性を入力します。
 - OAuth 認証の場合、Google プロジェクトで定義したクライアント ID およびクライアントシークレットの値を入力します。
 - サービスアカウント認証の場合、Google Development 環境で作成されたキーファイルパスとクライアント Email アドレスの値を入力します。

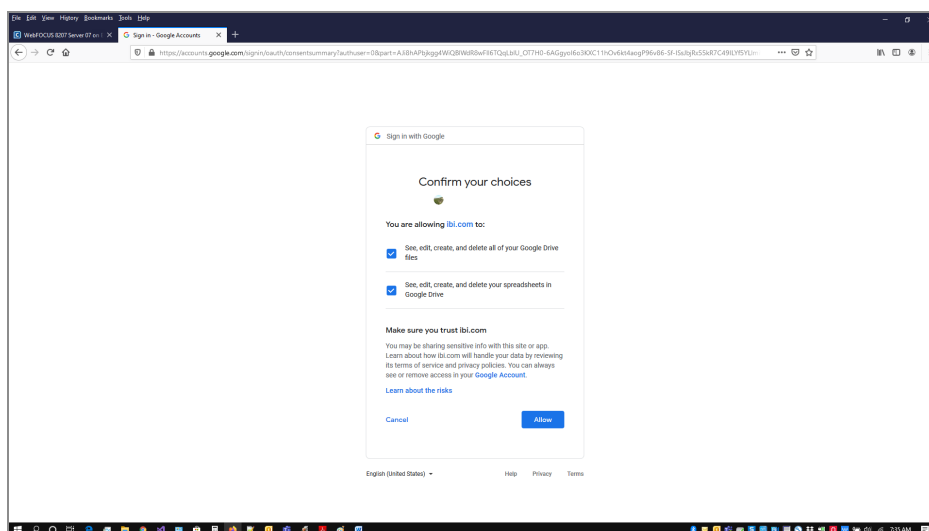
詳細は、[Google プロジェクトを作成するには](#)を参照してください。

8. OAuth 認証の場合、[アクセストークンの取得] をクリックします。

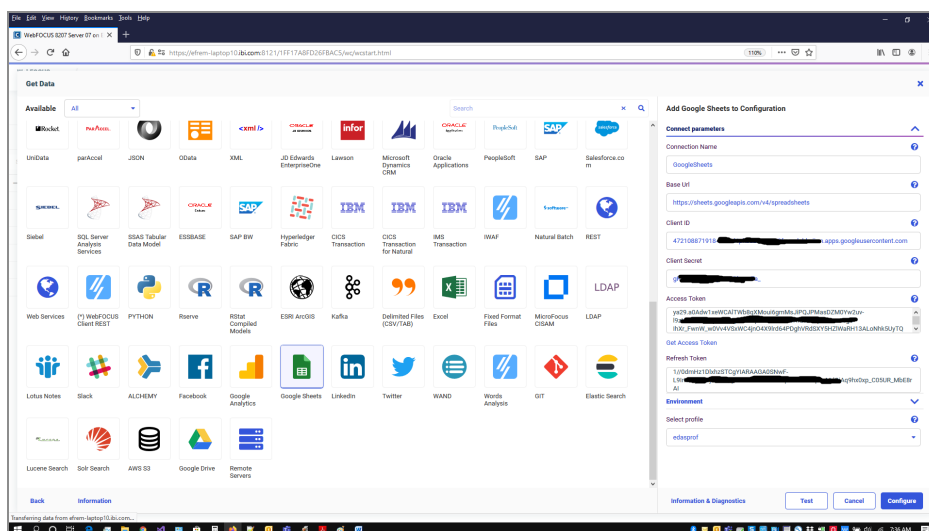
Google アカウントにサインインしていない場合は、Google への認証を行うサインインダイアログが開きます。

下図のように、権限の同意ページが開きます。





9. OAuth 認証の場合、すべての同意に対するプロンプトで [許可] をクリックします。
- 下図のように、[Google Sheets の構成の追加] ウィンドウに戻ると、[アクセストークン] および [トークンのリフレッシュ] のテキストボックスに値が入力されています。



注意：

- [アクセストークン] および [トークンのリフレッシュ] に値が入力されていない場合は、Google アカウントからサインアウトし、ブラウザの Cookie をクリアします。
- [トークンのリフレッシュ] のみ値が入力されていない場合は、Google アカウントの [セキュリティ] セクション下で、[アカウントにアクセスできるサードパー

ティアプリ] から Google Sheets アプリを削除します。

10. [構成] をクリックします。

Google Sheets アダプタが、ナビゲーションウィンドウのアダプタリストに追加されます。

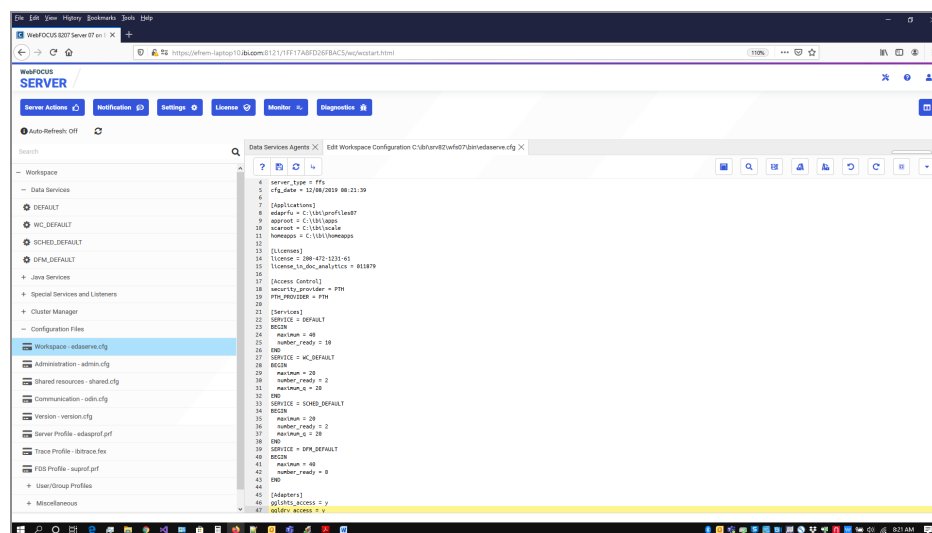
Google Drive アダプタの構成

ここでは、Google Drive アダプタの構成方法について説明します。

Google Drive アダプタを構成するには

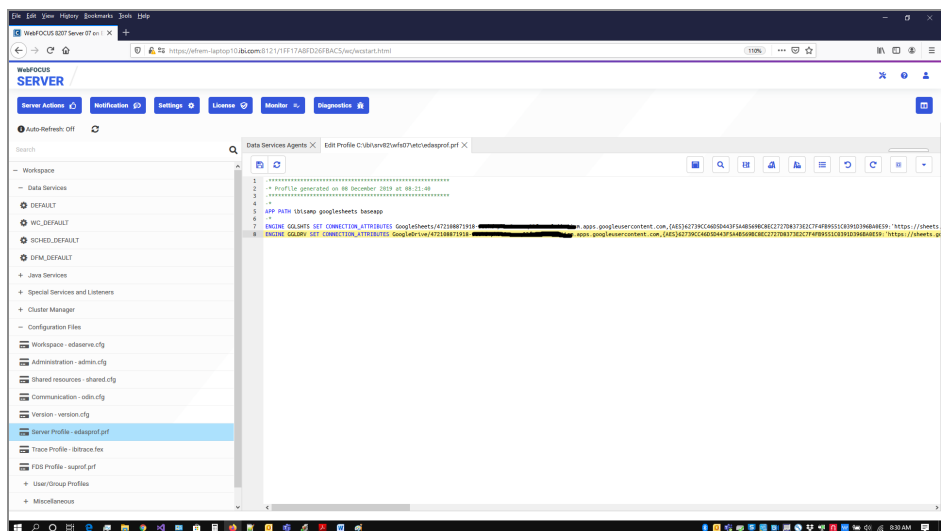
手順

1. 下図のように、WebFOCUS Reporting Server ブラウザインターフェースの [ワークスペース] ページで [構成ファイル] を展開して、[ワークスペース - edaserve.cfg] 構成ファイルを編集し、[Adapters] セクションで `ggldrv_access = y` 属性を追加します。



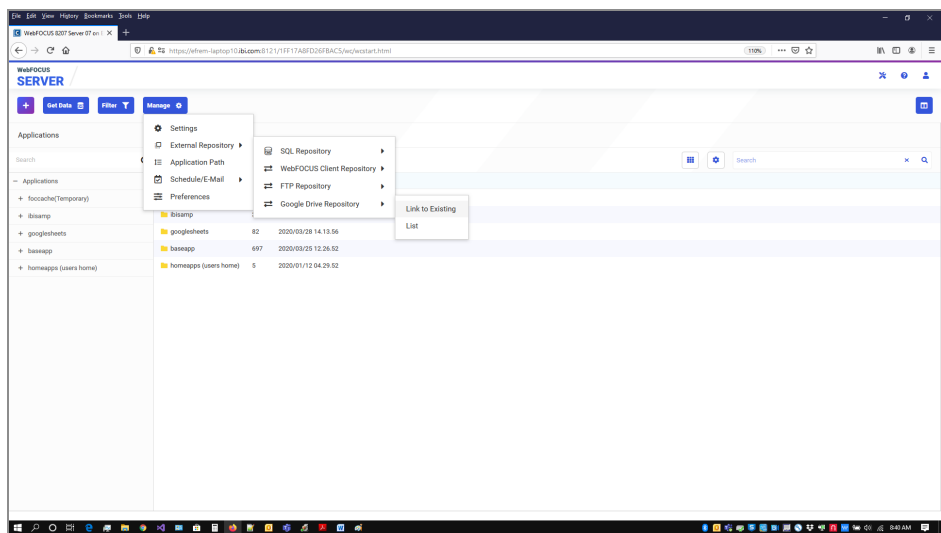
2. [保存してサーバを再起動] ボタンをクリックします。
3. [サーバプロファイル - edasprof.prf] を編集し、Google Sheets の接続コマンドをコピーします。

4. 下図のように、コピーした接続コマンドを edasprof.prf ファイルに貼り付け、コピーした接続のアダプタ名を GGLSHTS から GGLDRV に変更します。

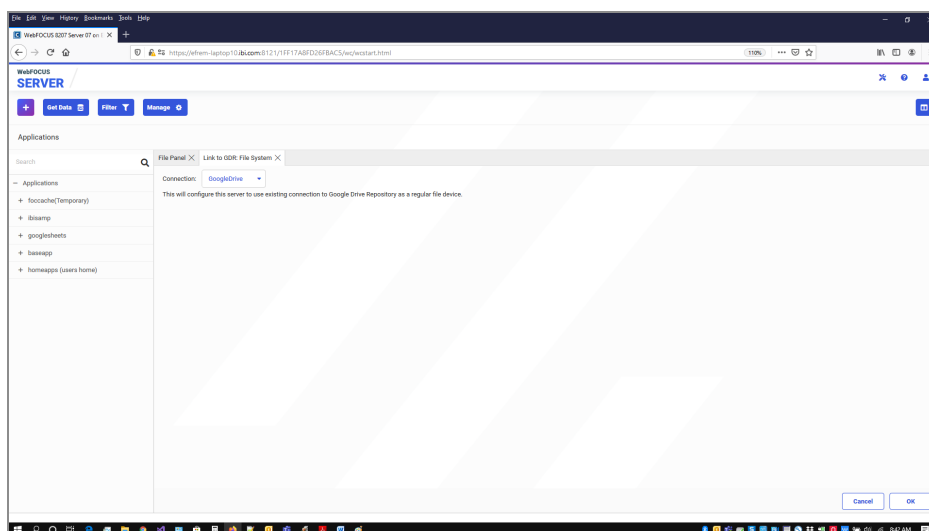


必要に応じて、Google Drive の接続名を変更することができます。

5. [保存] アイコンをクリックします。
6. 下図のように、WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] 表示で [管理] をクリックして [外部リポジトリ]、[Google ドライブリポジトリ]、[既存接続へのリンク] を順に選択します。

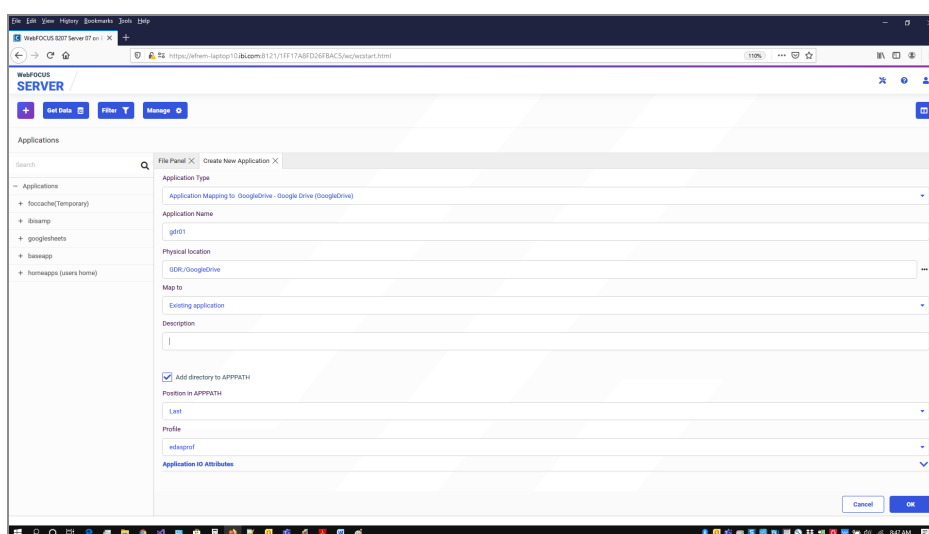


7. 下図のように、Google Drive の接続名を選択します。



8. [OK] をクリックします。

下図のように、Google Drive アプリケーションの構成画面が開きます。



9. 次の手順を実行します。

- a. 新しいアプリケーション名を入力し、必要な場合はアプリケーションの説明も入力します。

以下はその例です。

- **アプリケーション名** googledrive
- **説明** Google Drive

- b. [マッピング先] ドロップダウンリストから [新規アプリケーション (ディレクトリが作成されます)] を選択します。

10. [OK] をクリックします。

インターネットスピードおよび Google Drive のファイル数によって、マッピング処理に数分かかる場合があります。

完了後、「コマンドを実行しました」というメッセージが表示されます。

Google Sheets の接続属性

Google Sheets アダプタの接続属性には、次のものがあります。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

ベース URL

Google Sheets API リクエストの URL です。デフォルト値は次のとおりです。

```
https://sheets.googleapis.com/v4/spreadsheets
```

セキュリティ

[OAuth] または [サービスアカウント] のいずれかの認証方法を選択します。

クライアント ID

OAuth 認証を使用した場合に、Google Sheets に対してユーザのアプリケーションを識別する値です。

次の手順でこの値を取得します。

1. 次のサイトに移動します。

```
https://cloud.google.com/console/project
```

2. 作成済みの Google Sheets アダプタアプリケーションの [プロジェクト名] をクリックします。
3. 左側ウィンドウで [API とサービス] をクリックします。
4. 左側ウィンドウで [認証情報] をクリックします。

5. ウェブアプリケーションセクションの [クライアント ID] に、クライアント ID を使用します。

クライアントシークレット

OAuth 認証を使用した場合に、Google Sheets に対してユーザのアプリケーションを識別する値です。この値は、クライアント ID とともに使用します。

次の手順でこの値を取得します。

1. 次のサイトに移動します。

```
https://cloud.google.com/console/project
```

2. 作成済みの Google Sheets アダプタアプリケーションの [プロジェクト名] をクリックします。
3. 左側ウィンドウで [API とサービス] をクリックします。
4. 左側ウィンドウで [認証情報] をクリックします。
5. OAuth 2.0 クライアント ID の名前 (例、Web client 1) をクリックします。
6. ウェブアプリケーションセクションの [クライアントシークレット] に、クライアントシークレットを使用します。

アクセストークン

OAuth 認証を使用した場合に、アプリケーションが代行するユーザを識別する値です。[アクセストークンの取得] のリンクをクリックし、アクセストークンおよびリフレッシュトークンを取得します。

[アクセストークンの取得] を完了するには、WebFOCUS Reporting Server ブラウザインターフェースへのアクセスに使用するホスト名と、Google Sheets アプリケーションのリダイレクト URI で指定したホスト名が一致する必要があります。

Google アカウントにサインインしていない場合は、Google のサインイン画面が表示されます。

同意画面が開きます。[許可] をクリックします。

アクセストークンおよびリフレッシュトークンの取得で問題が発生した場合、ブラウザのキャッシュを Cookie も含めてクリアします。

リフレッシュトークン

アクセストークンの有効期限は非常に短いです。[リフレッシュトークン] は、実行時に新しいアクセストークンを取得するために使用されます。

キーファイルパス

サービスアカウント認証を使用する場合に、Google Development 環境で作成された JSON キーファイルのパスです。以下はその例です。

```
C:\ibi\apps\googlesheets\webfocus-555555-874feowfwojoe7.json
```

クライアント Email アドレス

サービスアカウント認証を使用する場合に、JSON キーファイルで定義されたクライアント Email アドレスです。以下はその例です。

```
xxxxxxxxxxxx-compute@developer.gserviceaccount.com
```

プロキシサーバ IP アドレス

プロキシサーバの IP アドレスです (例、170.115.249.42)。

プロキシポート番号

プロキシサーバのリスナポート番号です。デフォルトポート番号は 80 です。

PROXY HTTPS 相対パス

このオプションを選択すると、プロキシサーバの構成時に、絶対パスではなく相対パスが REST リクエストにより送信されます。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

Google Sheets アダプタでのメタデータ作成

Google Sheets アダプタのシノニムを作成することで、WebFOCUS レポートで使用するメタデータが作成されます。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Google Sheets のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

フィールド名の確認

すべての特殊文字をアンダースコア (_) に変換するとともに、予約語の使用を回避するために名前の確認を行う場合は、[フィールド名の確認] のチェックをオンにします。これにより、名前に番号が付加されます。このパラメータにより、規則に一致した名前が付けられます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

[フィールド名の確認] のチェックをオフにすると、ハイフン (-)、ブランク、円記号 (¥)、フォワードスラッシュ (/)、カンマ (,)、ドル記号 (\$) の特殊文字のみがアンダースコア (_) に変換されます。名前の確認は行われません。

ユニークフィールドにする

フィールド名およびグループ名の範囲をシノニム全体に設定する場合、[ユニークフィールドにする] のチェックをオンにします。これにより、シノニムの異なるセグメントであっても重複名は使用されません。このチェックをオフにすると、範囲はセグメントに設定されます。

データタイプマッピングのカスタマイズ

選択した場合、データフィールドの分解、地理的役割の追加、数値フィールドおよび文字フィールドのデータタイプの選択のために追加のフィールドを開きます。

追加コンポーネント

シノニム名に接頭語を割り当てます。

接尾語

シノニム名に接尾語を割り当てます。

[追加] をクリックして、シノニムを作成します。

HOLD コマンドを使用した Google スプレッドシートまたは Google スライドの作成

構成済みの Google Drive アダプタを Google Sheets アダプタとともに使用して、WebFOCUS レポートから Google スプレッドシートまたは Google スライドを作成することができます。

WebFOCUS HOLD コマンドを使用した Google スプレッドシートの作成

```
ON TABLE HOLD FORMAT GGLSHTS AS GoogleDriveApplication/Name
```

説明

GoogleDriveApplication

Google Drive にマッピングされた WebFOCUS Server アプリケーションの名前です。

名前

Google Sheets で保存される Google スプレッドシートの名前です。

たとえば、次の HOLD コマンドについて考察します。

```
ON TABLE HOLD FORMAT GGLSHTS AS googledrive/retailsheet
```

Google Drive にマッピングされた WebFOCUS Server アプリケーションは、googledrive です。

このコマンドにより、Google Sheets 環境で「retailsheet」という Google スプレッドシートが作成されます。

HOLD コマンドを使用した Google スライドの作成

```
ON TABLE HOLD AS mappedapp/filename FORMAT PPTX
```

説明

mappedapp

マッピング済み Google Drive アプリケーションです。

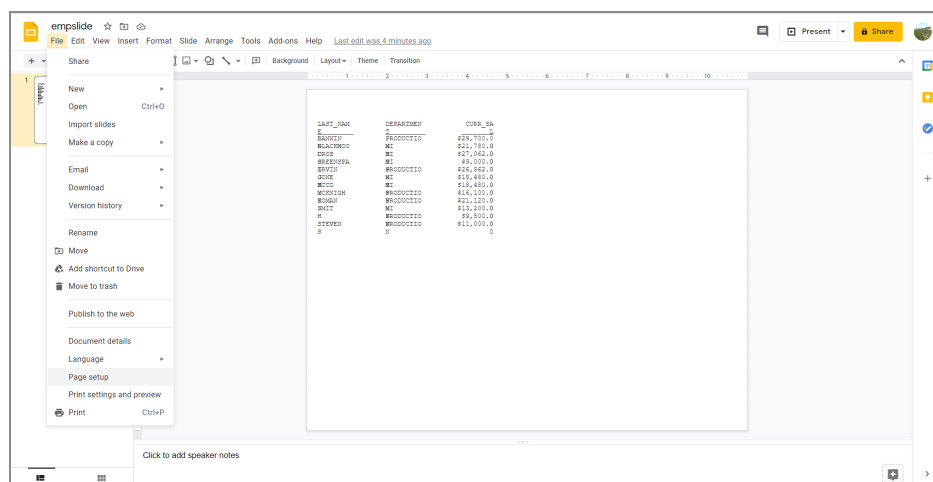
filename

作成する Google スライドのファイル名です。

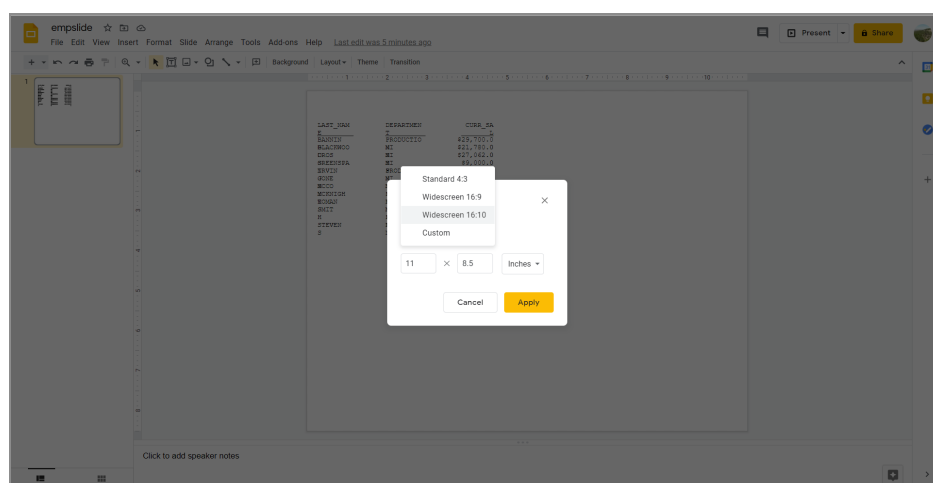
次の例では、アプリケーションディレクトリ「gdr01」が、Google Drive にマッピングされたアプリケーションディレクトリです。次のリクエストは、プロシジャに HOLD FORMAT PPTX コマンドを使用します。

```
TABLE FILE employe2
SUM CURR_SAL
BY LAST_NAME
BY DEPARTMENT
ON TABLE HOLD AS gdr01/empslide FORMAT PPTX
END
```

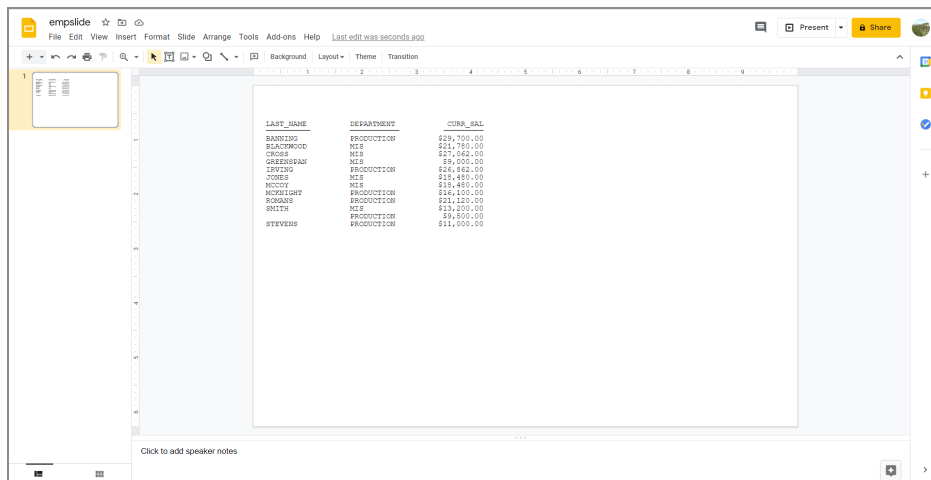
下図のように、Google スライドのページ設定に調整が必要な場合があります。



この場合、下図のように、Google スライドのページ設定を異なる縦横比に変更することができます。



下図は、この Google スライドを変更されたレイアウトで示しています。



The screenshot shows the empside application window. The title bar reads "empside" with standard window controls. The menu bar includes "File", "Edit", "View", "Insert", "Format", "Slide", "Arrange", "Tools", "Add-ons", and "Help". The status bar at the bottom indicates "Last edit was seconds ago". The main content area displays a table with three columns: "LAST_NAME", "DEPARTMENT", and "CURR_BAL". The table contains 12 rows of data. A sidebar on the left shows a list of slides, with the first slide selected. The bottom of the window has a "Click to add speaker notes" area.

LAST_NAME	DEPARTMENT	CURR_BAL
BANDING	PRODUCTION	\$28,700.00
BLATWOOD	NCS	\$21,700.00
CROSS	NCS	\$27,062.00
GREENSPAN	NCS	\$9,000.00
IRVING	PRODUCTION	\$26,862.00
JOES	NCS	\$19,480.00
MCCOY	NCS	\$10,480.00
MONTGOMERY	PRODUCTION	\$26,100.00
MORGAN	PRODUCTION	\$21,100.00
SMITH	NCS	\$13,200.00
STEVENS	PRODUCTION	\$9,500.00
	PRODUCTION	\$11,000.00

HP Vertica アダプタの使用

HP Vertica アダプタの構成方法について説明します。

HP Vertica アダプタを使用すると、アプリケーションから特定の HP Vertica データソースへのアクセスが可能になります。このアダプタにより、アプリケーションリクエストが HP Vertica コールに変換され、最適化されたアンサーセットがリクエストの送信元アプリケーションに返されます。

HP Vertica 環境の準備

HP Vertica アダプタを使用するには、アクセスするデータソースの HP Vertica ドライバをインストールする必要があります。また、サーバを起動する前に CLASSPATH 値を設定するほか、JSCOM3 サービスを実行状態にしておく必要があります。

Windows および UNIX で環境を設定するには

手順

1. サーバを起動する前に、HP Vertica ドライバファイルのパスを CLASSPATH 環境変数に追加することで、これらのファイルの格納先を設定します。

たとえば、UNIX で HP Vertica ドライバファイルのパスを /usr/driver_files/mydriver.jar に設定するには、次のコマンドを実行します。

```
CLASSPATH=/usr/driver_files/mydriver.jar;$CLASSPATH
export CLASSPATH
```

Windows では、以下のコマンドを実行します。

```
set CLASSPATH=c:¥usr¥driver_files¥mydriver.jar:%CLASSPATH%
```

サーバのワークスペースを Windows サービスとして実行する場合、CLASSPATH をシステム全体の環境変数として設定する必要があります。

同様に UNIX では、サーバを起動する前に変数を設定して、UNIX プロファイルに CLASSPATH の設定を追加します。

2. サーバを起動または再起動します。
ドライバを変更した場合にもサーバを再起動する必要があります。
3. サーバのワークスペースで以下のコマンドを実行することにより、JSCOM3 サービスが有効であることを確認します。

```
edastart -show
```

また、[スペシャルサービス] セクションに「JSCOM3 active」と表示されていることを確認します。

JSCOM3 サービスが有効ではない場合、サーバの EDAPRINT ログにメッセージ “fail to start” が出力されます。

HP Vertica アダプタの構成

アダプタを構成するには、確立する各接続の接続情報および認証情報を指定します。

アダプタは、HP Vertica データソースに接続する際に、接続情報および認証情報を必要とします。

この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。選択したサーバプロファイルにコマンドが追加されます。これらのプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用するプラットフォームでサポートされる場合) があります。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の HP Vertica データソースへの接続を宣言することができます。実際の接続は、最初のクエリ発行時に確立されます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスク

トップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

HP Vertica の接続属性

HP Vertica アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

URL

HP Vertica データソースのロケーション URL です。

ドライバ名

HP Vertica ドライバの名前です。

```
com.vertica.jdbc.Driver
```

詳細は、使用しているバージョンのドライバのマニュアルを参照してください。

IBI_CLASSPATH

Java サービスで使用する追加の Java クラスディレクトリ、または jar ファイル名のフルパスを定義します。値を設定するには、通信ファイルを編集するか、WebFOCUS Reporting Server ブラウザインターフェースを使用します。WebFOCUS Reporting Server ブラウザインターフェースの入力ボックスでは、1 行につき 1 つの参照を入力することができます。すべてのプラットフォームでは、ファイルを保存した際に、複数のエントリがコロン (:) で区切られた単一文字列に変換されます。ファイルを手動で編集する際は、区切り文字のコロン (:) を保持する必要があります。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

接続属性の手動宣言

```
ENGINE SQLVRT SET CONNECTION_ATTRIBUTES connection  
'URL'/userid,password
```

説明

SQLVRT

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

接続名です。

URL

HP Vertica データソースのロケーション URL です。

userid

ターゲットデータベースに登録されたプライマリ認証 ID です。

password

プライマリ認証 ID に関連するパスワードです。

接続属性の宣言

以下の SET CONNECTION_ATTRIBUTES コマンドは、明示的なユーザ ID (MYUSER) およびパスワード (PASS) を使用して、HP Vertica ドライバを介してデータソースにアクセスします。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザインターフェースまたは Data Migrator デスクトップインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

```
ENGINE SQLVRT SET CONNECTION_ATTRIBUTES CON1  
'jdbc:vertica://host:port'
```

デフォルト接続の上書き変更

複数の接続を定義している場合、最初の SET CONNECTION_ATTRIBUTES コマンドに指定した接続が、デフォルトの接続として設定されます。

デフォルト接続の変更

```
ENGINE SQLVRT SET DEFAULT_CONNECTION connection
```

説明**SQLVRT**

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

```
ENGINE SQLVRT SET DEFAULT_CONNECTION SAMPLE
```

HP Vertica メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および JDBC データタイプのサーバマッピングを定義するシノニムを作成します。

アダプタの識別

リクエストの解釈に必要なアダプタを識別するには、マスターファイルの SUFFIX 属性を使用します。HP Vertica アダプタを識別するには、SUFFIX 値として SQLVRT を使用します。

アダプタの識別

```
FILE[NAME]=file, SUFFIX=SQLVRT [, $]
```

説明

file

マスターファイルのファイル名です。このファイル名は文字で開始する必要があります。また、テーブルまたはビューの内容を表す名前にすることをお勧めします。実際のファイルには拡張子 .mas を付ける必要がありますが、この属性の値には拡張子を含めないでください。

SQLVRT

アダプタの値です。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な各オブジェクトの一意的名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

HP Vertica のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

外部 SQL スクリプトのパス

[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合は、さらに次の項目が表示されます。

次の標準名前規則は UNIX に適用されます。

- [ベースパス] テキストボックスに、SQL クエリが記述されたファイルの物理ディレクトリパスを指定します。ディレクトリ名を直接入力するか、[...] (参照) ボタンをクリックしてファイルの格納先ディレクトリへ移動します。このクリックにより、[ベースパスの選択] ダイアログボックスが表示されます。
- [ドキュメント名] テキストボックスに、ファイル名を入力します。必要に応じて、ワイルドカード文字を使用します。
- [ドキュメント拡張子] テキストボックスに、スクリプトファイルの拡張子を入力して、その拡張子のファイルのみを表示します。

アダプタはシノニムの作成中にネイティブ API コールを発行して選択リスト内の要素のリストを取得し、各要素に対するフィールドを持つマスターファイルを構成します。生成されたアクセスファイルは、DATASET 属性の SQL スクリプトパスを参照します。これには SQL クエリが格納されたファイルの名前と拡張子を含む、フルパスが記述されています。以下はその例です。

```
DATASET=/ul/home2/apps/report3.sql
```

WebFOCUS レポートを作成する際は、SQL クエリを使用してデータにアクセスします。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

外部キーを使用したクラスタ作成 (非推奨。[シノニムの編集] の使用を推奨)

[外部キーを使用したクラスタ作成] のチェックをオンにすると、現在のテーブルに外部キーで関係付けられているテーブルすべてをこのシノニムに含めることができます。ただし、このオプションは廃止される予定で、クラスタの作成にはシノニムエディタを使用することをお勧めします。生成されるマルチテーブルのシノニムは、このテーブルの外部キーの関係をすべて定義します。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (:) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[データタイプサポートレポート](#)を参照してください。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

オーナー/スキーマ

ユーザが所有する 1 つまたは複数のオブジェクトを作成したユーザアカウントです。

テーブル名

基になるオブジェクトの名前です。

タイプ

オブジェクトタイプ (テーブル、ビューなど)。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

生成されたシノニムの例

HP Vertica のシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

マスターファイル nf29004.mas

```
FILE=DIVISION, SUFFIX=SQLVRT , $
SEGNAME=SEG1_4, SEGTYPE=S0 , $
FIELD=DIVISION4, DIVISION4, I9, I4, MISSING=OFF , $
FIELD=DIVISION_NA4, DIVISION4, I9, I4, MISSING=OFF , $
FIELD=DIVISION_HE4, DIVISION4, I9, I4, MISSING=OFF , $
```

アクセスファイル nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=EDAQA.NF29004,
CONNECTION=CON1, KEYS=1, WRITE=YES, $
```

アクセスファイルのキーワード

下表は、アクセスファイルで使用するキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABLENAME	HP Vertica テーブルを識別します。この属性に割り当てる値には、次のようにオーナー (スキーマとも呼ばれる) の名前とデータベースリンク名を含めることができます。

キーワード	説明
	<pre>TABLENAME=[owner.]table</pre>
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre> CONNECTION=' ' は、ローカルのデータソースへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。 <pre>KEY=fld1/fld2/.../fldn</pre>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。
KEYFLD IXFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定します。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 KEYFLD は、親テーブルの共通カラムの

キーワード	説明
	FIELDNAME です。 IXFLD は関連するテーブルの共通カラムの FIELDNAME です。 KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることをお勧めしますが、必須ではありません。 注意：KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

データタイプサポートレポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[データタイプレポートを表示するには](#)を参照してください。

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアドプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

HP Vertica 環境のカスタマイズ

HP Vertica アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。

タイムアウト制限の指定

HP Vertica に対して SQL リクエストを実行した後、アダプタがその応答を待機する時間 (秒) を指定するには、TIMEOUT コマンドを使用します。

TIMEOUT コマンドの発行

```
ENGINE SQLVRT SET TIMEOUT {nn|0}
```

説明

SQLVRT

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nn

タイムアウトとなる秒数を指定します。デフォルト値は 30 です。

0

応答の待機時間が無制限であることを示します。

長時間実行中リクエストのキャンセル

WebFOCUS Reporting Server ブラウザインターフェースで、長時間実行中のリクエストをキャンセルすることができます。この操作を実行すると、ネイティブ HP Vertica ドライバの機能に応じて、リクエスト全体がキャンセルされるか、フェッチサイクルが中断されます。

長時間実行中のリクエストをキャンセルするには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [ワークスペース] ページで、[Java サービス] 下の [DEFAULT] を選択します。[DEFAULT] を右クリックし、[エージェント] を選択します。
2. [Java サービスエージェント] ウィンドウで、キャンセルする jscomid の行を選択し、右クリックして [終了] を選択します。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

```
ENGINE SQLVRT SET PASSRECS {ON|OFF}
```

説明

SQLVRT

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

HP Vertica 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)を参照してください。

JDBC アダプタの使用

JDBC アダプタの構成方法について説明します。

JDBC アダプタを使用すると、アプリケーションから JDBC データソースへのアクセスが可能になります。このアダプタにより、アプリケーションリクエストが JDBC コールに変換され、最適化されたアンサーセットがリクエストの送信元アプリケーションに返されます。

JDBC 環境の準備

JDBC アダプタを使用するには、アクセスするデータソースの JDBC ドライバをインストールする必要があります。また、サーバを起動する前に CLASSPATH 値を設定するほか、JSCOM3 サービスを実行状態にしておく必要があります。

たとえば、UniData データベースにアクセスするには、UniData の JDBC ドライバをインストールする必要があります。一般的な JDBC アダプタを使用する場合、構成可能なアダプタはサーバインスタンスごとに 1 つです。

注意

- 特定のデータソースタイプのアダプタが使用可能な場合は、そのアダプタを使用して特定のタイプにアクセスする必要があります。JDBC アダプタでは、特定のアダプタが存在するデータソースタイプはサポートされません。たとえば、Db2 データベースにアクセスするために Db2 アダプタが使用可能な場合は、JDBC アダプタを使用して Db2 データベースにアクセスすることはできません。
- 特定のデータソースタイプのアダプタが現在使用可能でない場合は、ベンダーから提供された JDBC ドライバとともに JDBC アダプタを使用することは可能です。ただし、そのアダプタが一般的な JDBC 標準に準拠している必要があります。一方、特定のデータソース要件に適合するよう、これらの標準をベンダー側で解釈したり、拡張したりしている場合があります。その結果として互換性が保持されない場合、そのベンダーのデータソースタイプのアダプタを使用する際に、動作に不整合が発生する可能性があります。この問題が発生した場合は、技術サポートに問い合わせてください。ユーザのデータアクセス要件に応じたサポートを提供します。

Windows および UNIX で環境を設定するには

手順

1. サーバを起動する前に、JDBC ドライバファイルのパスを CLASSPATH 環境変数に追加することで、これらのファイルの格納先を識別します。

たとえば、UNIX で JDBC ドライバファイルのパスを /usr/driver_files/mydriver.jar に設定するには、次のコマンドを発行します。

```
CLASSPATH=/usr/driver_files/mydriver.jar;$CLASSPATH
export CLASSPATH
```

Windows では、以下のコマンドを実行します。

```
set CLASSPATH=c:¥usr¥driver_files¥mydriver.jar:%CLASSPATH%
```

サーバのワークスペースを Windows サービスとして実行する場合、CLASSPATH をシステム全体の環境変数として設定する必要があります。

同様に UNIX では、サーバを起動する前に変数を設定して、UNIX プロファイルに CLASSPATH の設定を追加します。

2. サーバを起動または再起動します。
ドライバを変更した場合にもサーバを再起動する必要があります。
3. サーバのワークスペースで以下のコマンドを実行することにより、JSCOM3 サービスが有効であることを確認します。

```
edastart -show
```

また、[スペシャルサービス] セクションに「JSCOM3 active」と表示されていることを確認します。

JSCOM3 サービスが有効ではない場合、サーバの EDAPRINT ログにメッセージ“fail to start”が出力されます。

JDBC アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

アダプタは、JDBC データソースに接続する際に、接続情報および認証情報を必要とします。

この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。選択したサーバプロファイルにコマンドが追加されます。これらのプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用するプラットフォームでサポートされる場合) があります。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の JDBC データソースへの接続を宣言することができます。実際の接続は、最初のクエリ発行時に確立されます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップイン

ターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

JDBC の接続属性

JDBC アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

URL

JDBC データソースのロケーション URL です。

ドライバ名

JDBC ドライバの名前です。

詳細は、使用しているバージョンのドライバのマニュアルを参照してください。

IBI_CLASSPATH

Java サービスで使用する追加の Java クラスディレクトリ、または jar ファイル名のフルパスを定義します。値を設定するには、通信ファイルを編集するか、WebFOCUS Reporting Server ブラウザインターフェースを使用します。WebFOCUS Reporting Server ブラウザインターフェースの入力ボックスでは、1 行につき 1 つの参照を入力することができます。すべてのプラットフォームでは、ファイルを保存した際に、複数のエントリがコロン (:) で区切られた単一文字列に変換されます。ファイルを手動で編集する際は、区切り文字のコロン (:) を保持する必要があります。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (括弧子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

接続属性の手動宣言

```
ENGINE SQLJDBC SET CONNECTION_ATTRIBUTES connection
'URL'/userid,password
```

説明

SQLJDBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

接続名です。

URL

JDBC データソースのロケーション URL です。

userid

ターゲットデータベースに登録されたプライマリ認証 ID です。

password

プライマリ認証 ID に関連するパスワードです。

接続属性の宣言

以下の SET CONNECTION_ATTRIBUTES コマンドは、明示的なユーザ ID (MYUSER) およびパスワード (PASS) を使用して、JDBC ドライバを介してデータソースにアクセスします。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザインターフェースまたは Data Migrator デスクトップインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

```
ENGINE SQLJDBC SET CONNECTION_ATTRIBUTES CON1
'jdbc:xxxxxxx://hostname:port/datasource
```

デフォルト接続の上書き変更

複数の接続を定義している場合、最初の SET CONNECTION_ATTRIBUTES コマンドに指定した接続が、デフォルトの接続として設定されます。

デフォルト接続の変更

```
ENGINE SQLJDBC SET DEFAULT_CONNECTION connection
```

説明

SQLJDBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

```
ENGINE SQLJDBC SET DEFAULT_CONNECTION SAMPLE
```

接続範囲の制御

確立する接続ごとにアダプタを使用する場合、SET AUTODISCONNECT コマンドを使用して接続の継続を制御することができます。

接続範囲の制御

```
ENGINE SQLJDBC SET AUTODISCONNECT ON {FIN|COMMIT}
```

説明

SQLJDBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

JDBC メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および JDBC データタイプのサーバマッピングを定義するシノニムを作成します。

アダプタの識別

リクエストの解釈に必要なアダプタを識別するには、マスターファイルの SUFFIX 属性を使用します。JDBC アダプタを識別するには、SUFFIX 値として SQLJDBC を使用します。

アダプタの識別

```
FILE[NAME]=file, SUFFIX=SQLJDBC [, $]
```

説明

file

マスターファイルのファイル名です。拡張子.mas を持たないファイルには、8 バイト以内の文字でファイル名を指定することができます。このファイル名は文字で開始する必要があります。また、テーブルまたはビューの内容を表す名前にすることをお勧めします。実際のファイルには拡張子 .mas を付ける必要がありますが、この属性の値には拡張子を含めないでください。

SQLJDBC

アダプタの値です。

データベーステーブルへのアクセス

JDBC を使用してリモートのサードパーティのテーブルにアクセスする場合、RDBMS の JDBC ドライバをローカルにインストールする必要があります。

サーバは、JDBC ネットワーク上のサードパーティのデータベーステーブルにアクセスすることが可能です。データソースの URL を指定する必要があります。また、必要に応じて、アクセスするデータベースのユーザ ID およびパスワードを指定します。これらのパラメータは、サーバのグローバルプロファイルまたはユーザプロファイルで定義することができます。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な各オブジェクトの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

注意：JDBC データソースのシノニムを作成する場合、最初に edasprof.prf ファイルに SET SYNONYM=BASIC 構文を追加する必要があります。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティ

の選択ページが表示されます。

- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

JDBC のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイ

ルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

外部 SQL スクリプトのパス

[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合は、さらに次の項目が表示されます。

次の標準名前規則は UNIX に適用されます。

- [ベースパス] テキストボックスに、SQL クエリが記述されたファイルの物理ディレクトリパスを指定します。ディレクトリ名を直接入力するか、[...] (参照) ボタンをクリックしてファイルの格納先ディレクトリへ移動します。このクリックにより、[ベースパスの選択] ダイアログボックスが表示されます。
- [ドキュメント名] テキストボックスに、ファイル名を入力します。必要に応じて、ワイルドカード文字を使用します。
- [ドキュメント拡張子] テキストボックスに、スクリプトファイルの拡張子を入力して、その拡張子のファイルのみを表示します。

アダプタはシノニムの作成中にネイティブ API コールを発行して選択リスト内の要素のリストを取得し、各要素に対するフィールドを持つマスターファイルを構成します。生成されたアクセスファイルは、DATASET 属性の SQL スクリプトパスを参照します。これには SQL クエリが格納されたファイルの名前と拡張子を含む、フルパスが記述されています。以下はその例です。

```
DATASET=/ul/home2/apps/report3.sql
```

WebFOCUS レポートを作成する際は、SQL クエリを使用してデータにアクセスします。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

外部キーを使用したクラスタ作成 (非推奨。[シノニムの編集] の使用を推奨)

[外部キーを使用したクラスタ作成] のチェックをオンにすると、現在のテーブルに外部キーで関係付けられているテーブルすべてをこのシノニムに含めることができます。ただし、このオプションは廃止される予定で、クラスタの作成にはシノニムエディタを使用することをお勧めします。生成されるマルチテーブルのシノニムは、このテーブルの外部キーの関係をすべて定義します。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (:) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[データタイプサポートレポート](#)を参照してください。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

オーナー/スキーマ

ユーザが所有する 1 つまたは複数のオブジェクトを作成したユーザアカウントです。

テーブル名

基になるオブジェクトの名前です。

タイプ

オブジェクトタイプ (テーブル、ビューなど)。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

生成されたシノニムの例

JDBC アダプタのシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

マスターファイル nf29004.mas

```
FILE=DIVISION, SUFFIX=SQLJDBC , $
SEGNAME=SEG1_4, SEGTYPE=S0 , $
FIELD=DIVISION4, DIVISION4, I9, I4, MISSING=OFF , $
FIELD=DIVISION_NA4, DIVISION4, I9, I4, MISSING=OFF , $
FIELD=DIVISION_HE4, DIVISION4, I9, I4, MISSING=OFF , $
```

アクセスファイル nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=EDAQA.NF29004,
CONNECTION=CON1, KEYS=1, WRITE=YES, $
```

アクセスファイルのキーワード

下表は、アクセスファイルで使用されるキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABLENAME	JDBC テーブルを識別します。この属性に割り当てる値には、次のようにオーナー (スキーマとも呼ばれる) の名前とデータベースリンク名を含めることができます。 TABLENAME=[owner.]table
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。

キーワード	説明
	<code>CONNECTION=connection</code> <code>CONNECTION=' '</code> は、ローカルのデータソースへのアクセスを示します。 <code>CONNECTION</code> 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の <code>KEY</code> 属性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。 <code>KEY=fld1/fld2/.../fldn</code>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。
KEYFLD IXFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定します。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 - KEYFLD は、親テーブルの共通カラムの FIELDNAME です。 - IXFLD は関連するテーブルの共通カラムの FIELDNAME です。 KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることを

キーワード	説明
	お勧めしますが、必須ではありません。 注意：KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

データタイプサポートレポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[データタイプレポートを表示するには](#)を参照してください。

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアードプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

JDBC 環境のカスタマイズ

JDBC アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。

タイムアウト制限の指定

JDBC に対して SQL リクエストを実行した後、アダプタがその応答を待機する時間 (秒) を指定するには、TIMEOUT コマンドを使用します。

TIMEOUT コマンドの発行

```
ENGINE SQLJDBC SET TIMEOUT {nn|0}
```

説明

SQLJDBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nn

タイムアウトとなる秒数を指定します。デフォルト値は 30 です。

0

応答の待機時間が無制限であることを示します。

長時間実行中リクエストのキャンセル

WebFOCUS Reporting Server ブラウザインターフェースで、長時間実行中のリクエストをキャンセルすることができます。この操作を実行すると、ネイティブ JDBC ドライバの機能に応じて、リクエスト全体がキャンセルされるか、フェッチサイクルが中断されます。

長時間実行中のリクエストをキャンセルするには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [ワークスペース] ページで、[Java サービス] 下の [DEFAULT] を選択します。[DEFAULT] を右クリックし、[エージェント] を選択します。
2. [Java サービスエージェント] ウィンドウで、キャンセルする jscomid の行を選択し、右クリックして [終了] を選択します。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、メニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

```
ENGINE SQLJDBC SET PASSRECS {ON|OFF}
```

説明

SQLJDBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

JDBC 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)を参照してください。

JSON アダプタの使用

JSON アダプタの構成方法について説明します。

JSON (JavaScript Object Notation) アダプタを使用すると、アプリケーションから JSON データソースへのアクセスが可能になります。このアダプタは、階層ツリーを読み取り、アンサーセットをリクエストの送信元アプリケーションに返します。

JSON 環境の準備

JSON アダプタでは、環境変数を構成する必要はありません。

JSON アダプタの構成

JSON アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで構成することができます。

アダプタを構成するには

アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで構成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

結果

注意：JSON アダプタは、[XML ベース] ドロップダウンリスト下にあります。入力パラメータは必要ありません。

JSON ドキュメントとマスターファイルの関連付け

マスターファイルを JSON ドキュメントに関連付けるために、FILEDEF コマンドを明示的に発行する必要があります。FILEDEF コマンドを適切に記述するためには、JSON ドキュメン

トの編成方法を決定する必要があります。ドキュメントがすべて単一ファイルに格納されている場合、この編成は COLUMN と呼ばれます。各種ドキュメントが単一のディレクトリに格納されている場合、この編成は COLLECTION と呼ばれます。

ソース	コレクション編成	カラム編成
disk	サポートされています	サポートされています
http	サポートされていません	サポートされています
https	サポートされていません	サポートされています

JSON ドキュメントとマスターファイルの関連付け

```
FILEDEF ddname source filename
```

説明

ddname

必要なマスターファイルと一致する論理参照名です。

source

JSON ドキュメントのロケーションです。有効な値は http、https、disk です。

filename

JSON カラムでは、ソースファイルへのフルパスです。

JSON コレクションでは、ソースディレクトリへのフルパスです。ディレクトリ名の後に、ワイルドカード文字として「*.*」を付加する必要があります。

HTTP または HTTPS セッションを使用して JSON ドキュメントを読み込む場合、ファイル名の先頭を 'http://' または 'https://' で始める必要があります。

JSON カラムに対する FILEDEF コマンドの発行

注意：WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの [シノニムの作成] ウィンドウで作業を行う場合、FILEDEF の作成方法についての詳細は、[JSON のシノニム作成パラメータ](#)を参照してください。

Windows C:¥ に ordercol.JSON ドキュメントが格納されている場合、次の FILEDEF をプロファイルで使します。

```
FILEDEF ordercol disk C:¥ordercol.JSON
```

UNIX /usera/JSON/ に ordercol.JSON ドキュメントが格納されている場合、次の FILEDEF をプロファイルで使します。

```
FILEDEF ordercol DISK /usera/JSON/ordercol.JSON
```

HTTP および HTTPS http://www.test.com/JSONtest/ (または https://www.test.com/JSONtest/) に ordercol.JSON ドキュメントが格納されている場合、次の FILEDEF をプロファイルで使します。

```
FILEDEF order http http://www.test.com/JSONtest/ordercol.JSON
```

または

```
FILEDEF order http https://www.test.com/JSONtest/ordercol.JSON
```

JSON コレクションに対する FILEDEF コマンドの発行

注意：WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの [シノニムの作成] ウィンドウで作業を行う場合、FILEDEF の作成方法についての詳細は、[JSON のシノニム作成パラメータ](#)を参照してください。

Windows C:¥ORDERS に order1.JSON および order2.JSON の各ドキュメントが格納されている場合、次の FILEDEF をプロファイルで使します。

```
FILEDEF order disk C:¥orders¥*.*
```

UNIX /usera/orders に ordercol.JSON ドキュメントが格納されている場合、次の FILEDEF をプロファイルで使します。

```
FILEDEF ordercol DISK //'TEST.JSON'
```

JSON メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および JSON データタイプのサーバマッピングを定義するシノニムを作成します。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な JSON データ構造の一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは一時項目 (DEFINE) や他のセキュリティ機構などの、サーバの拡張メタデータ機能をサポートします。

シノニムを使用すると、クライアントアプリケーションを変更せずに、動作を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが指定した JSON ドキュメントに基づいて生成されます。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。
選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオブ

ションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

JSON のシノニム作成パラメータ

以下のリストは、アダプタのシノニムを作成する際に、値の指定が必要なパラメータ、および実行する関連作業を示しています。これらのオプションが複数のウィンドウに表示される場合もあります。ウィンドウに表示されたボタンをクリックして次のウィンドウに進み、最後に [シノニムの作成] ボタンをクリックします。これで、入力データに基づいてシノニムが作成されます。

シノニムは、JSON ドキュメントに基づいて作成することができます。

- ドキュメントインスタンスを選択し (ステップ 1)、[ドキュメントインスタンスからシノニムを作成] をクリックします。

ドキュメントインスタンスの選択のパラメータ (ステップ 1/2)

HTTP パス

URL を使用してドキュメントインスタンスを選択します。この選択には、[ドキュメントパス]、[ドキュメント名]、および [ドキュメント拡張子] が必要になります。

ドキュメントパス

ドキュメントインスタンスのパスを定義します。

- [HTTP パス] を選択していない場合、物理パスまたはアプリケーションディレクトリ、および JSON ドキュメント名を入力するか、[...] (参照) ボタンをクリックしてドキュメントを選択します。
- [HTTP パス] を選択している場合、シノニムの作成に使用する JSON ドキュメントが格納されたディレクトリの http アドレスを入力します。この機能は、JSON ドキュメントがローカルファイルの場合は使用できません。URL は、http:// または https:// で始める必要があります。

ドキュメント名

JSON ドキュメントの名前を入力します。

ドキュメント拡張子

ドキュメントの拡張子を入力します。デフォルト値は JSON です。

ドキュメントインスタンスからシノニムを作成

このボタンをクリックすると、ドキュメントインスタンスからシノニムを作成することができます。

JSON シノニムの作成 (ステップ 2/2)

フィールド名の確認

すべての特殊文字をアンダースコア (_) に変換するとともに、予約語の使用を回避するために名前の確認を行う場合は、[フィールド名の確認] のチェックをオンにします。これにより、名前に番号が付加されます。このパラメータにより、規則に一致した名前が付けられます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

[フィールド名の確認] のチェックをオフにすると、ハイフン (-)、ブランク、円記号 (¥)、フォワードスラッシュ (/)、カンマ (,)、ドル記号 (\$) の特殊文字のみがアンダースコア (_) に変換されます。名前の確認は行われません。

ユニークフィールドにする

フィールド名およびグループ名の範囲をシノニム全体に設定する場合、[ユニークフィールドにする] のチェックをオンにします。これにより、シノニムの異なるセグメントであっても重複名は使用されません。このチェックをオフにすると、範囲はセグメントに設定されます。

シノニム名

シノニムに割り当てる名前を指定します。異なる名前を指定する場合は、表示されている名前を上書きします。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスはブランクにしておきます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

リレーショナル DBMS の JSON データタイプからの JSON ドキュメントへのアクセス

JSON ドキュメントは、任意のデータソースのフィールドまたはカラムに保存される場合があります。これらのドキュメントからのレポート作成は、元のデータを記述する親セグメントに関連付けられたサブツリーとしてドキュメント構造を定義することでサポートされます。

シノニムの作成プロセスは、DBMS 内のデータに対して実行するか、JSON ドキュメントに対して実行する必要があります。次に、2 つのマスターファイルを組み合わせ、JSON マスターファイルを DBMS に対して作成されたマスターファイルの子としなければなりません。この場合、FILEDEF は必要ありません。

WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのツールを使用して RDBMS から JSON データにアクセスするには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのシノニム作成機能を使用して、JSON データのフィールドを含む RDBMS データソース用シノニムを生成します。JSON データは、ネイティブデータソースに格納するために使用するデータタイプとは関係なく、マスターファイルのシノニムに TX フィールドとしてマッピングされます。RDBMS エンジンでは多くの場合、CLOB データタイプが TX にマッピングされます。
2. 生成されたマスターファイルをシノニムエディタで開きます。マスターファイルは [テキスト表示] の右側ウィンドウに表示されます。たとえば、Db2 データソースのマスターファイルは次のように表示されます。

```
FILENAME=TESTJSON, SUFFIX=DB2      , $  
SEGMENT=TESTJSON, SEGTYPE=S0, $  
FIELDNAME=STORE, ALIAS=STORE, USAGE=A30, ACTUAL=A30, $  
FIELDNAME=ADDRESS, ALIAS=ADDRESS, USAGE=A20, ACTUAL=A20,  
MISSING=ON, $  
FIELDNAME=GLOSS, ALIAS=GLOSS, USAGE=TX50, ACTUAL=TX,  
MISSING=ON, $
```

3. ibi Data Migrator デスクトップインターフェースで、「TX」として記述されている JSON データを格納するフィールドを右クリックします。ポップアップメニューに [ピボット] オプションが表示されます。
4. [ピボット]、[複数値を行に変換] を順に選択します。このオプションは、JSON フィールドを直接読み込み、SEGSUF=JSON で新しいセグメントを作成します。
[マッピングソースを選択] ダイアログボックスが開きます。
 - a. [構造分析サンプリング行] を選択し、[次へ] をクリックします。

[カラムのマッピング] ダイアログボックスが開き、ピボットの対象となるカラムから 2 行が表示されます。

データの外觀が JSON と一致していることを確認します。

- b. [データタイプの選択] ドロップダウンリストから [JSON] を選択し、[適用] をクリックします。

結果のマスターファイルには、「SEGSUF=JSON」という新しい DUMROOT セグメントとして表現される JSON データ定義が格納されます。これは、[テキスト表示] モードで、元の RDBM セグメントの後に表示されます。

[DUMROOT] を展開して、JSON のフィールド名を表示することができます。

[DUMROOT] を右クリックして、これらのフィールドの値を表示することもできます。

マスターファイルは次のように表示されます。DUMROOT セグメントに JSON フィールドのデータが格納されています。

```
FILENAME=testjson, SUFFIX=DB2      , $
  SEGMENT=TESTJSON, SEGTYPE=S0, $
    FIELDNAME=STORE, ALIAS=STORE, USAGE=A30, ACTUAL=A30, $
    FIELDNAME=ADDRESS, ALIAS=ADDRESS, USAGE=A20, ACTUAL=A20,
      MISSING=ON, $
    FIELDNAME=GLOSS, ALIAS=GLOSS, USAGE=TX50, ACTUAL=TX,
      MISSING=ON, ACCESS_PROPERTY=(INTERNAL), $
  SEGMENT=DUMROOT, SEGTYPE=S0, SEGSUF=JSON      ,
    PARENT=TESTJSON, POSITION=GLOSS, $
    FIELDNAME=DUMROOT, ALIAS=JSON_DUMMY_EL, USAGE=A1, ACTUAL=A1,
      ACCESS_PROPERTY=(INTERNAL), $
    FIELDNAME=TITLE, ALIAS=title, USAGE=A55, ACTUAL=A55,
      MISSING=ON,
      REFERENCE=GLOSSARY, PROPERTY=ELEMENT, $
    FIELDNAME=GLOSSDIV, ALIAS=GlossDiv, USAGE=A1, ACTUAL=A1,
      MISSING=ON, ACCESS_PROPERTY=(INTERNAL),
      REFERENCE=GLOSSARY, PROPERTY=ELEMENT, $
    FIELDNAME=TITLE1, ALIAS=title, USAGE=A55, ACTUAL=A55,
      MISSING=ON,
      REFERENCE=GLOSSDIV, PROPERTY=ELEMENT, $
    FIELDNAME=GLOSSLIST, ALIAS=GlossList, USAGE=A1, ACTUAL=A1,
      MISSING=ON, ACCESS_PROPERTY=(INTERNAL),
      REFERENCE=GLOSSDIV, PROPERTY=ELEMENT, $
    FIELDNAME=GLOSSENTRY, ALIAS=GlossEntry, USAGE=A1, ACTUAL=A1,
```

```
MISSING=ON, ACCESS_PROPERTY=(INTERNAL),
REFERENCE=GLOSSLIST, PROPERTY=ELEMENT, $
FIELDNAME=ID, ALIAS=ID, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
FIELDNAME=SORTAS, ALIAS=SortAs, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
FIELDNAME=GLOSSTERM, ALIAS=GlossTerm, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
```

```
FIELDNAME=ACRONYM, ALIAS=Acronym, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
FIELDNAME=ABBREV, ALIAS=Abbrev, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
FIELDNAME=GLOSSDEF, ALIAS=GlossDef, USAGE=A1, ACTUAL=A1,
MISSING=ON, ACCESS_PROPERTY=(INTERNAL),
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
FIELDNAME=PARA, ALIAS=para, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSDEF, PROPERTY=ELEMENT, $
FIELDNAME=GLOSSSEE, ALIAS=GlossSee, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
SEGMENT=GLOSSSEEALSO, SEGTYPE=S0, PARENT=GLOSSARY, $
FIELDNAME=GLOSSSEEALSO, ALIAS=GlossSeeAlso, USAGE=A55,
ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSDEF, PROPERTY=ELEMENT, $
```

5. シノニムエディタの [ファイル] メニューから [保存] を選択し、更新したマスターファイルを保存します。

RDBMS から JSON データに手動でアクセスするには

JSON データを格納する、1 つまたは複数のフィールドを含むテーブルが RDBMS 内に存在すると想定します。この JSON データからレポートを作成するためには、以下の手順を実行

します。

手順

1. 対象の DBMS のフォーマットを使用して、リレーショナルデータソースのマスターファイルを作成します。

```
FILENAME=TESTJSON, SUFFIX=DB2      , $
SEGMENT=TESTJSON, SEGTYPE=S0, $
  FIELDNAME=STORE, ALIAS=STORE, USAGE=A30, ACTUAL=A30, $
  FIELDNAME=ADDRESS, ALIAS=ADDRESS, USAGE=A20, ACTUAL=A20,
    MISSING=ON, $
  FIELDNAME=GLOSS, ALIAS=GLOSS, USAGE=TX50, ACTUAL=TX,
    MISSING=ON, $
```

2. RDBMS テーブルのフィールドに JSON ドキュメント用のマスターファイルを作成します。フォーマットの異なる 2 つの JSON ドキュメントが存在する場合、それぞれにマスターファイルを作成する必要があります。
3. 各マスターファイルを手動で結合します。JSON マスターファイルの各ルートセグメントに、POSITION、PARENT、SEGSUF という 3 つのフィールドを追加します。POSITION キーワードでは、JSON ドキュメントを含むフィールドを特定します。PARENT フィールドでは、元のデータソースを定義します。SEGSUF フィールドでは、サブツリーを表す JSON ドキュメントのルートセグメントを定義します。マスターファイル内のすべてのフィールドの長さの合計は、FOCUS の制限である 32 キロバイトを超えてはなりません。超えた場合、クエリはエラーになります。

```
FILENAME=BASEAPP_GLOSSARY, SUFFIX=JSON      ,
DATASET=c:¥json¥glossary.json, $
SEGMENT=GLOSSARY, SEGTYPE=S0, $
  FIELDNAME=GLOSSARY, ALIAS=glossary, USAGE=A1, ACTUAL=A1,
    MISSING=ON, ACCESS_PROPERTY=(INTERNAL), $
  FIELDNAME=TITLE, ALIAS=title, USAGE=A55, ACTUAL=A55,
    MISSING=ON,
    REFERENCE=GLOSSARY, PROPERTY=ELEMENT, $
  FIELDNAME=GLOSSDIV, ALIAS=GlossDiv, USAGE=A1, ACTUAL=A1,
    MISSING=ON, ACCESS_PROPERTY=(INTERNAL),
    REFERENCE=GLOSSARY, PROPERTY=ELEMENT, $
  FIELDNAME=TITLE1, ALIAS=title, USAGE=A55, ACTUAL=A55,
    MISSING=ON,
    REFERENCE=GLOSSDIV, PROPERTY=ELEMENT, $
  FIELDNAME=GLOSSLIST, ALIAS=GlossList, USAGE=A1, ACTUAL=A1,
    MISSING=ON, ACCESS_PROPERTY=(INTERNAL),
    REFERENCE=GLOSSDIV, PROPERTY=ELEMENT, $
```

```

FIELDNAME=GLOSSENTRY, ALIAS=GlossEntry, USAGE=A1, ACTUAL=A1,
MISSING=ON, ACCESS_PROPERTY=(INTERNAL),
REFERENCE=GLOSSLIST, PROPERTY=ELEMENT, $
FIELDNAME=ID, ALIAS=ID, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
FIELDNAME=SORTAS, ALIAS=SortAs, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
FIELDNAME=GLOSSTERM, ALIAS=GlossTerm, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
FIELDNAME=ACRONYM, ALIAS=Acronym, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $

```

```

FIELDNAME=ABBREV, ALIAS=Abbrev, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
FIELDNAME=GLOSSDEF, ALIAS=GlossDef, USAGE=A1, ACTUAL=A1,
MISSING=ON, ACCESS_PROPERTY=(INTERNAL),
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
FIELDNAME=PARA, ALIAS=para, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSDEF, PROPERTY=ELEMENT, $
FIELDNAME=GLOSSSEE, ALIAS=GlossSee, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
SEGMENT=GLOSSSEEALSO, SEGTYPE=S0, PARENT=GLOSSARY, $
FIELDNAME=GLOSSSEEALSO, ALIAS=GlossSeeAlso, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSDEF, PROPERTY=ELEMENT, $

```

結合されたマスターファイル

```

FILENAME=testjson, SUFFIX=DB2      , $
SEGMENT=TESTJSON, SEGTYPE=S0, $
FIELDNAME=STORE, ALIAS=STORE, USAGE=A30, ACTUAL=A30, $
FIELDNAME=ADDRESS, ALIAS=ADDRESS, USAGE=A20, ACTUAL=A20,
MISSING=ON, $
FIELDNAME=GLOSS, ALIAS=GLOSS, USAGE=TX50, ACTUAL=TX,
MISSING=ON, ACCESS_PROPERTY=(INTERNAL), $

```

```

SEGMENT=GLOSSARY, SEGTYPE=S0, SEGSUF=JSON      , PARENT=TESTJSON,
POSITION=GLOSS, $
  FIELDNAME=GLOSSARY, ALIAS=glossary, USAGE=A1, ACTUAL=A1,
  MISSING=ON, ACCESS_PROPERTY=(INTERNAL), $
  FIELDNAME=TITLE, ALIAS=title, USAGE=A55, ACTUAL=A55,
  MISSING=ON,
  REFERENCE=GLOSSARY, PROPERTY=ELEMENT, $
  FIELDNAME=GLOSSDIV, ALIAS=GlossDiv, USAGE=A1, ACTUAL=A1,
  MISSING=ON, ACCESS_PROPERTY=(INTERNAL),
  REFERENCE=GLOSSARY, PROPERTY=ELEMENT, $
  FIELDNAME=TITLE1, ALIAS=title, USAGE=A55, ACTUAL=A55,
  MISSING=ON,
  REFERENCE=GLOSSDIV, PROPERTY=ELEMENT, $
  FIELDNAME=GLOSSLIST, ALIAS=GlossList, USAGE=A1, ACTUAL=A1,
  MISSING=ON, ACCESS_PROPERTY=(INTERNAL),
  REFERENCE=GLOSSDIV, PROPERTY=ELEMENT, $

```

```

FIELDNAME=GLOSSENTRY, ALIAS=GlossEntry, USAGE=A1, ACTUAL=A1,
  MISSING=ON, ACCESS_PROPERTY=(INTERNAL),
  REFERENCE=GLOSSLIST, PROPERTY=ELEMENT, $
  FIELDNAME=ID, ALIAS=ID, USAGE=A55, ACTUAL=A55,
  MISSING=ON,
  REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
  FIELDNAME=SORTAS, ALIAS=SortAs, USAGE=A55, ACTUAL=A55,
  MISSING=ON,
  REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
  FIELDNAME=GLOSSTERM, ALIAS=GlossTerm, USAGE=A55, ACTUAL=A55,
  MISSING=ON,
  REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
  FIELDNAME=ACRONYM, ALIAS=Acronym, USAGE=A55, ACTUAL=A55,
  MISSING=ON,
  REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
  FIELDNAME=ABBREV, ALIAS=Abbrev, USAGE=A55, ACTUAL=A55,
  MISSING=ON,
  REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
  FIELDNAME=GLOSSDEF, ALIAS=GlossDef, USAGE=A1, ACTUAL=A1,
  MISSING=ON, ACCESS_PROPERTY=(INTERNAL),
  REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $

```

```

FIELDNAME=PARA, ALIAS=para, USAGE=A55, ACTUAL=A55,
  MISSING=ON,

```

```
REFERENCE=GLOSSDEF, PROPERTY=ELEMENT, $
FIELDNAME=GLOSSSEE, ALIAS=GlossSee, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSENTRY, PROPERTY=ELEMENT, $
SEGMENT=GLOSSSEEALSO, SEGTYPE=S0, PARENT=GLOSSARY, $
FIELDNAME=GLOSSSEEALSO, ALIAS=GlossSeeAlso, USAGE=A55, ACTUAL=A55,
MISSING=ON,
REFERENCE=GLOSSDEF, PROPERTY=ELEMENT, $
```

変換

JSON ドキュメント内のデータには、日付または数値が反映される場合がありますが、マスターファイルのシノニム内のフィールドは、すべて ALPHA データタイプに設定されます。

数値

数値フィールドに対する算術演算を有効にするには、数値フィールドの USAGE 属性に指定されたデータタイプを変更する必要があります。この変更では、JSON ドキュメント内のデータに応じて、整数 (Integer) (I)、倍精度浮動小数点数 (Double Float) (D)、またはパック 10 進数 (Decimal) (P) のいずれかを指定します。データタイプを倍精度浮動小数点数 (D) またはパック 10 進数 (P) に変更する場合、必要に応じて小数点以下の桁数および精度を使用して JSON ドキュメント内のデータを記述します。

また、数値フィールドの ACTUAL 属性に指定された ALPHA データタイプの長さを JSON ドキュメント内のデータの最大長に合わせて変更することをお勧めします。

JSON の日付

日付に対する算術演算を有効にするには、日付フィールドの USAGE 属性に指定されたデータタイプを変更する必要があります。この変更では、JSON ドキュメント内で使用されている日付フォーマットに応じて、YYMD、MDYY、または DMY のいずれかを指定します。

また、日付フィールドの ACTUAL 属性に指定された ALPHA データタイプの長さを 10 に設定する必要があります。

注意

- マスターファイルでデータタイプを日付データタイプのいずれかに設定した場合、レポートの作成元となる JSON ドキュメントのそれぞれで、日付フォーマットがマスターファイルで定義した日付フォーマットに一致していることを確認する必要があります。データタイプが異なる場合、エラーが発生します。
- アンサーセット内の日付フォーマットでは、JSON ドキュメント内で使用されている日付フォーマットが使用されず、常に YYYY/MM/DD に設定されます。

JSON での日付の使用

JSON ドキュメント内の日付フォーマット	使用する USAGE=
1996-01-30	YYMD
01-30-1996	MDYY
30-01-1996	DMYY

Microsoft Access アダプタの使用

Microsoft Access アダプタの構成方法について説明します。

Microsoft Access アダプタを使用すると、アプリケーションから Microsoft Access データソースへのアクセスが可能になります。このアダプタにより、アプリケーションリクエストがネイティブの Microsoft Access ステートメントに変換され、最適化されたアンサーセットがリクエストの送信元アプリケーションに返されます。

Microsoft Access 環境の準備

Microsoft Access アダプタを使用するには、少なくとも ODBC 32 ビットのドライバマネージャをインストールする必要があります。システムデータソース名の構成により、ローカルまたはリモートの Microsoft Access データソースに接続することが可能です。

Windows で環境を設定するには

Microsoft Access へのアクセスに必要な環境の設定はありません。

Microsoft Access アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

アダプタは、Microsoft Access データベースサーバに接続する際に、接続情報および認証情報を必要とします。この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。選択したプロファイルにコマンドが追加されます。選択可能なプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) があります。
- グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) のいずれかに手動でコマンドを追加します。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の Microsoft Access データベースサーバへの接続を宣言することができます。Microsoft Access Server への実際の接続は、その接続を参照する最初のクエリ発行時に確立されます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタ

ンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Microsoft Access の接続属性

MS Access アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

データソース

データソース名 (DSN) です。デフォルトのデータソース名はありません。値を入力する必要があります。

パスワード

データソースに関連するパスワードがある場合はそのパスワードです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、`CONNECTION_ATTRIBUTES` コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの `edasprof.prf` です。

新しいユーザプロファイル (`user.prf`) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (`edasprof`) に保存します。

接続属性の手動宣言

Trusted 認証

```
ENGINE SQLMAC SET CONNECTION_ATTRIBUTES connection DSN_name
```

Explicit 認証 パスワードで保護されたデータソース

```
ENGINE SQLMAC SET CONNECTION_ATTRIBUTES connection DSN_name/, [password]
```

説明

SQLMAC

アダプタを指定します。SET `SQLENGINE` コマンドをすでに発行している場合は、この値を省略することができます。

connection

この接続属性のセットの識別に使用される論理名です。

`connection` と `DSN_name` の間にスペースが 1 つ必要です。

DSN_name

アクセスする Microsoft Access データソース名 (DSN) です。この名前は、`odbc.ini` ファイルのエントリに一致させる必要があります。

password

プライマリ認証 ID に関連するパスワードです。

接続属性の宣言

以下の SET CONNECTION_ATTRIBUTES コマンドは、パスワード (PASS) を使用して、SAMPLESERVER という名前の Microsoft Access DSN への接続 CON1 を宣言します。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

```
ENGINE SQLMAC SET CONNECTION_ATTRIBUTES CON1 SAMPLESERVER/,PASS
```

接続情報の更新

このアダプタの CONNECTION_ATTRIBUTES コマンドの構文が拡張され、アプリケーションの開発環境から実稼働環境への移植をサポートするよう設計された論理接続名を指定できるようになりました。この拡張された構文では、既存の CONNECTION_ATTRIBUTES コマンドのマイグレートが必要になる場合があります。

WebFOCUS Reporting Server ブラウザインターフェースの [マイグレート] オプションを使用すると、現在のサーバ設定が新しいバージョンにマイグレートされます。このオプションにアクセスするには、メニューバーから [ツール] を選択し、[ワークスペース] を選択します。ナビゲーションウィンドウで [ワークスペース] フォルダを右クリックし、[マイグレート] を選択します。[マイグレート] ウィンドウで、構成インスタンスディレクトリ (EDACONF) の完全なパスを入力して [続行] ボタンをクリックします。この方法を使用することをお勧めします。

[マイグレート] オプションを使用しない場合、以下の情報に注意してください。

- バージョン 7.6 SP01 より前のバージョンで宣言された接続名は引き続きサポートされます。
- 新規のシノニムを作成する目的で新しい接続を追加する場合、既存の接続は新しいフォーマットで再保存され、また既存のシノニムは変更を行うことなく引き続き機能します。
- 既存のシノニムを使用する目的で新しい接続を追加する場合、デフォルトの論理接続名を、既存のアクセスファイルの CONNECTION=value 属性で指定された値に一致するよう変更する必要があります。

たとえば、バージョン 7.6 SP01 より前のバージョンで、以下のように接続が定義されている場合を想定します。

```
ENGINE SQLMAC SET CONNECTION_ATTRIBUTES DSN_A/uid,pwd
```

この DSN_A に保存されているオブジェクトに基づいてシノニムを作成すると、アクセスファイルには以下の記述が追加されます。

```
CONNECTION=DSN_A
```

この後、新しい接続を追加する際に、この既存のシノニムを再利用するには、接続名をデフォルトの CON01 から DSN_A に変更し、DSN_A として保存する必要があります。この接続は、以下のようにプロファイルに保存されます。

```
ENGINE SQLMAC SET CONNECTION_ATTRIBUTES DSN_A DSN_A/uid,pwd
```

デフォルト接続の上書き変更

複数の接続を定義した場合、最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続がデフォルト接続として機能します。このデフォルト値は、SET DEFAULT_CONNECTION コマンドを使用して上書きすることができます。

デフォルト接続の変更

```
ENGINE SQLMAC SET DEFAULT_CONNECTION connection
```

説明

SQLMAC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

```
(FOC1671) コマンドの順序に誤りがあります。
```

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

```
ENGINE SQLMAC SET DEFAULT_CONNECTION SAMPLE
```

接続範囲の制御

確立する接続ごとにアダプタを使用する場合、SET AUTODISCONNECT コマンドを使用して接続の継続を制御することができます。

接続範囲の制御

```
ENGINE SQLMAC SET AUTODISCONNECT ON {FIN|COMMAND|COMMIT}
```

説明

SQLMAC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMAND

各リクエストの後に自動的に切断されます。SET AUTODISCONNECT コマンドは、イベントがどの程度の頻度で発生するかによって、大幅なオーバーヘッドをもたらす可能性があります。このオーバーヘッドのほとんどすべてがサーバには関連しません。このオーバーヘッドは、オペレーティングシステムとデータソースに関連します。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

Microsoft Access メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および Microsoft Access データタイプのサーバマッピングを定義するシノニムを作成します。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な Microsoft Access テーブルまたはビューの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Microsoft Access のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

オブジェクト名

オブジェクト名のフィルタに使用する文字列を入力します。必要に応じて、文字列の先頭や末尾にワイルドカード文字 (%) を入力します。たとえば、「ABC%」と入力して、名前が ABC で始まるオブジェクトをすべて選択します。「%ABC」と入力して、名前が ABC で終わるオブジェクトをすべて選択します。「%ABC%」と入力して、名前の先頭、中間、または末尾に ABC が含まれるオブジェクトをすべて選択します。

外部 SQL スクリプトのパス

[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合は、さらに次の項目が表示されます。

次の標準名前規則は UNIX に適用されます。

- [ベースパス] テキストボックスに、SQL クエリが記述されたファイルの物理ディレ

クトリパスを指定します。ディレクトリ名を直接入力するか、[...] (参照) ボタンをクリックしてファイルの格納先ディレクトリへ移動します。このクリックにより、[ベースパスの選択] ダイアログボックスが表示されます。

- [ドキュメント名] テキストボックスに、ファイル名を入力します。必要に応じて、ワイルドカード文字を使用します。
- [ドキュメント拡張子] テキストボックスに、スクリプトファイルの拡張子を入力して、その拡張子のファイルのみを表示します。

アダプタはシノニムの作成中にネイティブ API コールを発行して選択リスト内の要素のリストを取得し、各要素に対するフィールドを持つマスターファイルを構成します。生成されたアクセスファイルは、DATASET 属性の SQL スクリプトパスを参照します。これには SQL クエリが格納されたファイルの名前と拡張子を含む、フルパスが記述されています。以下はその例です。

```
DATASET=/u1/home2/apps/report3.sql
```

WebFOCUS レポートを作成する際は、SQL クエリを使用してデータにアクセスします。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (:) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視

され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[Microsoft Access データタイプのサポート](#)を参照してください。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

テーブル名

基になるオブジェクトの名前です。

タイプ

オブジェクトタイプ (テーブル、ビューなど)。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

生成されたシノニムの例

Microsoft Access アダプタのシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

生成されたマスターファイル nf29004.mas

```
FILE=NF29004, SUFFIX=SQLMAC , $  
SEGNAME=NF29004, SEGTYPE=S0 , $  
FIELD=DIVISION4, DIVISION4, I11, I4, MISSING=ON , $  
FIELD=DIVISION_NA4, DIVISION_NA4, A25, A25, MISSING=ON , $  
FIELD=DIVISION_HE4, DIVISION_HE4, I11, I4, MISSING=ON , $
```

生成されたアクセスファイル nf29004.acx

```
SEGNAME=NF29004, TABLENAME=NF29004,  
CONNECTION=MACDSN, KEYS=1, WRITE=YES, $
```

アクセスファイルのキーワード

下表は、アクセスファイルで使用するキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABlename	Microsoft Access のテーブル名を指定します。
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre> CONNECTION=' ' は、ローカルのデータベースサーバへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。 <pre>KEY=fld1/fld2/.../fldn</pre>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

Microsoft Access データタイプのサポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[データタイプレポートを表示するには](#)を参照してください。

国際言語サポートの有効化

SET パラメータ NCHAR は、文字セットが 1 バイト、2 バイト、3 バイトであることを示します。NCHAR の設定は、NCHAR および NVARCHAR データタイプのマッピングに影響します。

下表は、NCHAR 値に基づくデータタイプのマッピングを示しています。

Microsoft Access のデータタイプ	備考	NCHAR SBCS		NCHAR DBCS		NCHAR TBCS	
		USAGE	ACTUAL	USAGE	ACTUAL	USAGE	ACTUAL
NCHAR (n)	n は 1 から 4000 までの整数です。	An	An	Ad	Ad	At	At

Microsoft Access のデータタイプ	備考	NCHAR SBCS		NCHAR DBCS		NCHAR TBCS	
		USAGE	ACTUAL	USAGE	ACTUAL	USAGE	ACTUAL
	$d = 2 * n$ $t = 3 * n$						
NVARCHAR (n)	n は 1 から 4000 までの整数です。 $d = 2 * n$ $t = 3 * n$	An	An	Ad	Ad	At	At

国際言語サポートの有効化

```
ENGINE SQLMAC SET NCHAR {SBCS|DBCS|TBCS}
```

説明

SQLMAC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

SBCS

1 バイトの文字セットを示します。デフォルト値は、SBCS です。

DBCS

2 バイトの文字セットを示します。

TBCS

3 バイトの文字セットを示します。

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアードプロシージャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェイスで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

Microsoft Azure Synapse Analytics アダプタの使用

Microsoft Azure Synapse Analytics アダプタの構成方法について説明します。

Microsoft Azure Synapse Analytics (従来の Microsoft Azure SQL Data Warehouse) アダプタを使用すると、アプリケーションからクラウドベースの Azure Synapse Analytics データソースへのアクセスが可能になります。このアダプタにより、データまたはアプリケーションリクエストがネイティブの T-SQL ステートメントに変換され、最適化されたアンサーセットがリクエストの送信元プログラムに返されます。

注意：バージョン 9.3.0 以降、Microsoft Azure Synapse Analytics アダプタは、UPSERT サポートを提供します。

Microsoft Azure Synapse Analytics 環境 の準備

Azure ポータルへのアクセスは、Microsoft 担当者から取得する必要があります。Azure Synapse Analytics は、このポータルに作成する必要があります(作成済みでない場合)。

このアダプタは、2 つのタイプの Microsoft クライアントによる Azure Synapse Analytics へのアクセスをサポートします。

- Windows および Linux 環境の ODBC
- すべての環境の JDBC

Microsoft Azure Synapse Analytics アダ プタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

アダプタは、Microsoft Azure Synapse Analytics データベースサーバに接続する際に、接続情報および認証情報を必要とします。この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。

この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。コンソールから、選択したプロファイルにこのコマンドが追加されます。選択可能なプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) があります。
- グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) のいずれかに手動でコマンドを追加します。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の Microsoft Azure Synapse Analytics データベースへの接続を宣言することができます。Microsoft Azure Synapse Analytics への実際の接続は、その接続を参照する最初のクエリ発行時に確立されます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Microsoft Azure Synapse Analytics の接続属性

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

サーバ

Microsoft Azure Synapse Analytics が稼働しているマシンの名前です。

セキュリティ

Microsoft Azure Synapse Analytics への接続時にユーザを認証する方法には、次の 3 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報が Microsoft Azure Synapse Analytics データソースへの接続時に標準ログインの認証情報として渡されます。このオプションを使用するには、Azure Synapse Analytics セキュリティとして Azure Synapse Analytics および Windows に設定する必要があります。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID およびパスワードが、Microsoft Azure Synapse Analytics への接続時に標準ログインの認証情報として渡されます。このオプションを使用するには、Azure Synapse Analytics セキュリティとして Azure Synapse Analytics および Windows に設定する必要があります。
- **Trusted** アダプタは、サーバのデータアクセスエージェントによって偽装されたオペレーティングシステムユーザの認証情報を使用して、オペレーティングシステムログインとして Microsoft Azure Synapse Analytics に接続します。このオプションは、いずれの Azure Synapse Analytics セキュリティの設定においても機能します。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

デフォルトデータベース

この接続に使用される Microsoft Azure Synapse Analytics データベースの名前です。パスを含むデータベース名は、一重引用符 (') で囲む必要があります。このパラメータはオプションとして指定します。指定しない場合、認証 ID に関連するデータベースがデフォルト値として設定されます。

追加の接続文字列キーワード (オプション)

ODBC 接続の動作変更には使用される接続文字列キーワードについての詳細は、Microsoft Azure Synapse Analytics のマニュアルを参照してください。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (括弧は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

Windows での接続属性の手動宣言

Explicit 認証 ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報が Microsoft Azure Synapse Analytics への接続時に標準ログインの認証情報として渡されます。

```
ENGINE SQLADW SET CONNECTION_ATTRIBUTES [connection]
server/userid,password [;dbname][;additional_connection_string_keywords]
```

Password Passthru 認証 クライアントアプリケーションから受け取ったユーザ ID およびパスワードが、Microsoft Azure Synapse Analytics への接続時に標準ログインの認証情報として渡されます。

```
ENGINE SQLADW SET CONNECTION_ATTRIBUTES [connection]
server/[;dbname][;additional_connection_string_keywords]
```

Trusted 認証 アダプタは、サーバのデータアクセスエージェントによって偽装された Windows ユーザの認証情報を使用して、Windows ログインとして Microsoft Azure Synapse Analytics に接続します。

```
ENGINE SQLADW SET CONNECTION_ATTRIBUTES [connection]server/, [;dbname]
[;additional_connection_string_keywords]
```

注意：特殊文字を含む値は、一重引用符 (') で囲みます。値に一重引用符 (') が含まれている場合は、その一重引用符の前に別の一重引用符を追加する必要があります。結果的に、一重引用符が 2 つ連続することになります。たとえば、ブランクと一重引用符 (') の両方を含む「Mary O'Brien」をユーザ ID として指定する場合は、「'Mary O' 'Brian'」と入力します。

デフォルト接続の上書き変更

複数の接続を定義した場合、最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続がデフォルト接続として機能します。このデフォルト値は、SET DEFAULT_CONNECTION コマンドを使用して上書きすることができます。

デフォルト接続の変更

```
ENGINE SQLADW SET DEFAULT_CONNECTION connection
```

説明

SQLADW

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

```
(FOC1671) コマンドの順序に誤りがあります。
```

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

```
(FOC1671) コマンドの順序に誤りがあります。
```


デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

```
ENGINE SQLADW SET DEFAULT_CONNECTION SAMPLE
```

接続範囲の制御

確立する接続ごとにアダプタを使用する場合、SET AUTODISCONNECT コマンドを使用して接続の継続を制御することができます。

接続範囲の制御

```
ENGINE SQLADW SET AUTODISCONNECT ON {FIN|COMMAND|COMMIT}
```

説明

SQLADW

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMAND

各リクエストの後に自動的に切断されます。SET AUTODISCONNECT コマンドは、イベントがどの程度の頻度で発生するかによって、大幅なオーバーヘッドをもたらす可能性があります。このオーバーヘッドのほとんどすべてがサーバには関連しません。このオーバーヘッドは、オペレーティングシステムとデータソースに関連します。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

Microsoft Azure Synapse Analytics メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および Microsoft Azure Synapse Analytics データタイプのサーバマッピングを定義するシノニムを作成します。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な Microsoft Azure Synapse Analytics テーブルまたはビューの一意的な名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

このアダプタでは、MS SQL ネイティブシノニムの特定のタイプでのメタデータの作成がサポートされます。この機能は、Microsoft Azure Synapse Analytics バージョン 2012 以降でのみ使用することができます。

ネイティブシノニムのメタデータは、次の条件下で作成することができます。

- ネイティブシノニムは、TABLE または VIEW ベースオブジェクトから作成されている必要があります。
- ベースオブジェクト (TABLE または VIEW) は、アダプタ接続文字列がターゲットとする Microsoft Azure Synapse Analytics 上に存在している必要があります。
- ネイティブシノニムは、最大で 3 部構成の名前で記述されたベースオブジェクトで作成されている必要があります。ターゲットサーバ内でのベースオブジェクトのパスおよびオーナーシップに応じて、受容可能なフォーマットは次のとおりです。

```
object_name, schema_name.object_name,
```

または

```
database_name.schema_name.object_name.
```

次のタイプのネイティブシノニムはサポートされません。

- ストアドプロシジャに基づいているもの。
- リンクされた Microsoft Azure Synapse Analytics 上のオブジェクトに基づいているもの。
- 次のような 4 部構成の名前で記述されたオブジェクトに基づいているもの。

```
server_name.database_name.schema_name.object_name
```

ストアドプロシジャのシノニムの作成、およびストアドプロシジャに対するレポートの実行についての詳細は、[ストアドプロシジャのシノニムの作成](#)を参照してください。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。

適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。

- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Microsoft Azure Synapse Analytics のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

重要：オブジェクトタイプにストアドプロシジャを選択した場合、入力パラメータはここで説明するものとは多少異なります。詳細は、[ストアドプロシジャに対するレポートの実行](#)を参照してください。

データベース

テーブルやオブジェクトの選択元となるデータベースを指定するには、次のいずれかを実行します。

- [デフォルトデータベース] を選択して、デフォルトのデータベースとして設定されているデータベースを使用します。
- [データベース] ドロップダウンリストからデータベースを選択します。ここには、現在の DBMS インスタンスにあるすべてのデータベースが表示されます。
[デフォルトデータベース] が選択されている場合は、データベースを選択する前に、このチェックをオフにします。

目的のデータベースを指定するには、[データベース] ドロップダウンリストからデータベースを選択します。ドロップダウンリストには、Microsoft Azure Synapse Analytics のターゲットインスタンスのすべてのデータベースがリストされます。デフォルトデータベースを選択すると、接続の構成時に設定されたデータベースが保持されます。構成時にデフォルトデータベースが設定されなかった場合、Azure Synapse Analytics 上でアクティブなログインに割り当てられているデータベースがデフォルトとして使用されます。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

外部キーを使用したクラスタ作成 (非推奨。[シノニムの編集] の使用を推奨)

[外部キーを使用したクラスタ作成] のチェックをオンにすると、現在のテーブルに外部キーで関係付けられているテーブルすべてをこのシノニムに含めることができます。ただし、このオプションは廃止される予定で、クラスタの作成にはシノニムエディタを使用することをお勧めします。生成されるマルチテーブルのシノニムは、このテーブルの外部キーの関係をすべて定義します。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (:) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細を表示するには、WebFOCUS Reporting Server ブラウザインターフェースの [データの取得] ページの [オプション] メニューから [データタイプ] を選択してレポートを実行します。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

テーブル名

基になるオブジェクトの名前です。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

生成されたシノニムの例

Microsoft Azure Synapse Analytics アダプタのシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

生成されたマスターファイル nf29004.mas

```
FILE=DIVISION, SUFFIX=SQLADW , $  
SEGNAME=SEG1_4, SEGTYPE=S0 , $
```

```
FIELD=DIVISION4,    DIVISION4,    I9,    I4,    MISSING=OFF ,$,
FIELD=DIVISION_NA4, DIVISION_NA4, A25, A25, MISSING=ON  ,$,
FIELD=DIVISION_HE4, DIVISION_HE4, I9,    I4,    MISSING=ON  ,$,
```

生成されたアクセスファイル nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=edaqa.nf29004,
CONNECTION=connmss, KEYS=1, WRITE=YES, $
```

Microsoft SQL ODBC テーブルコメントの シノニムへのマッピング

Microsoft SQL テーブルまたはビューのシノニムを作成する際に、アダプタは次のようにコメントをマッピングします。

- Microsoft Azure Synapse Analytics テーブルまたはビューのコメント (存在する場合) は、マスターファイルシノニムの REMARKS 属性にマッピングされます。
- Microsoft Azure Synapse Analytics カラムのコメント (存在する場合) は、マスターファイルシノニムの DESCRIPTION 属性にマッピングされます。

Unicode と 非 Unicode 両方のコメントがサポートされます。

また、Microsoft Azure Synapse Analytics カラムのタイトル (存在する場合) は、マスターファイルシノニムの TITLE 属性にマッピングされます。

アクセスファイルのキーワード

下表は、アクセスファイルで使用するキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。

キーワード	説明
TABLENAME	Microsoft Azure Synapse Analytics のテーブルを指定します。以下のように完全修飾のテーブル名を指定することができます。 <pre>TABLENAME=[[database.]owner.]table</pre>
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre> CONNECTION=' ' は、ローカルのデータベースサーバへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。この属性を使用するには、キーを構成するフィールドをマスターファイルの先頭に記述する必要があります。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。 <pre>KEY=fld1/fld2/.../fldn</pre>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。

キーワード	説明
KEYFLD IXFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定します。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 - KEYFLD は、親テーブルの共通カラムの FIELDNAME です。 - IXFLD は関連するテーブルの共通カラムの FIELDNAME です。 KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることをお勧めしますが、必須ではありません。 注意：KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。
AUTO INCREMENT	[Yes] に設定すると自動インクリメント機能が有効になります。
START	インクリメントの順序の初期値です。
INCREMENT	インクリメントの間隔です。
INDEX_NAME INDEX_UNIQUE INDEX_COLUMNS INDEX_ORDER	データベース内のインデックス名、およびインデックスフィールドの一意性、名前、順序を示します。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

Microsoft Azure Synapse Analytics データタイプのサポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[データタイプレポートを表示するには](#)を参照してください。

SQL 式内での末尾ブランク

TABLE アダプタ内の新しい SQL ジェネレータでは、デフォルトで文字列リテラル内の末尾ブランク、および数値リテラル内の小数部と指数の表記を含むリテラルコンテンツを保持します。これにより、生成される SQL に対する制御が向上します。

末尾ブランクが不要なケースでは、次の構文が下位互換性を確保するために使用できます。

```
ENGINE SQLADW SET TRIM_LITERALS ON
```

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成された

フィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

読み取り専用フィールドのサポート

CREATE SYNONYM コマンドは、TIMESTAMP として作成された Microsoft Azure のカラムまたは IDENTITY 属性が定義されたカラムに FIELDTYPE=R のフィールド定義を作成します。これらのフィールドは読み取り専用です。MAINTAIN または MODIFY プロシジャを実行する場合、アダプタはマスターファイルに FIELDTYPE=R が定義されたカラムに対して書き込み処理を実行しません。

読み取り専用フィールドのサポート

以下は、最初のカラムのプロパティが IDENTITY で、2 番目のカラムがタイムスタンプカラムのテーブルを作成する例です。

```
CREATE TABLE TAB1
(idproptab int IDENTITY (1,1), timestmp timestamp)
```

CREATE SYNONYM コマンドによって、このテーブルに以下のマスターファイルが生成されます。

```
FILE=TAB1, SUFFIX=SQLADW , $
SEGNAME=TAB1, SEGTYPE=S0 , $
FIELD=IDPROPTAB, idproptab, I11, I4, MISSING=OFF, FIELDTYPE=R , $
FIELD=TIMSTMP, timestmp, A16, A16, MISSING=ON, FIELDTYPE=R , $
```

Microsoft Azure Synapse Analytics のストアドプロシジャに対するレポートの実行

SELECT ステートメントや TABLE コマンドなどのレポートツールを使用し、Microsoft Azure Synapse Analytics のストアドプロシジャを実行して、プロシジャの出力パラメータおよびアンサーセットに対するレポートを実行することができます。ストアドプロシジャの実行という方法には、次の利点があります。

- アンサーセット以外に、出力パラメータである OUT パラメータおよび OUT モードの INOUT パラメータを検索することができる。他の呼び出し方法では、アンサーセットのみが検索されます。
- TABLE および他のレポートツールを使用して、出力パラメータおよびアンサーセットの処理、フォーマット設定、表示を簡単に行うことができる。

ストアドプロシジャに対するレポートを実行するには、次の手順を実行します。

1. **シノニムの作成** ストアドプロシジャのアンサーセットのシノニムを作成します。詳細は、[ストアドプロシジャのシノニムの作成](#)を参照してください。
2. **レポートプロシジャの作成** レポートプロシジャを作成します。詳細は、[ストアドプロシジャに対するレポートの実行](#)を参照してください。
3. **レポートの実行** ストアドプロシジャを実行し、出力パラメータ (OUT および OUT モードの INOUT) およびレポートに指定されたアンサーセットフィールドに対するレポートを実行します。

ストアドプロシジャのシノニムの作成

シノニムは、ストアドプロシジャのパラメータとアンサーセットを記述します。

アンサーセットの構造は、プロシジャの実行時に指定する入力パラメータ値により異なります。このため、プロシジャの実行時に指定される入力パラメータ値のセットごとに、個別のシノニムを作成する必要があります。たとえば、3つの異なる入力パラメータ値セットでストアドプロシジャを実行する場合、シノニムを3つ (値セットごとに1つ) 作成する必要があります。なお、特に明記していない限り、「入力パラメータ」は、IN パラメータおよび IN モードの INOUT パラメータを意味します。

例外 ユーザがプロシジャの内部ロジックを理解しており、入力パラメータの値の範囲とプロシジャによって返された各アンサーセットの構造の関係を把握している場合、各アンサー

セットの構造に対応する単一のシノニムを作成して、各シノニムにアンサーセットの構造を返すために必要な入力パラメータの値のセットを指定することが可能です。

シノニムには、次のセグメントを含めます。

- **INPUT** - このセグメントは、IN パラメータおよび IN モードの INOUT パラメータを記述します。
IN パラメータおよび IN モードの INOUT パラメータがない場合、このセグメントはダミーフィールドを 1 つ記述します。
- **OUTPUT** - このセグメントは、OUT パラメータおよび OUT モードの INOUT パラメータを記述します。
OUT パラメータおよび OUT モードの INOUT パラメータがない場合、このセグメントは省略されます。
- **ANSWERSET n** - 各アンサーセットに 1 つ含まれます。
アンサーセットがない場合、このセグメントは省略されます。

Microsoft Azure Synapse Analytics ストアド プロシジャ CustOrders のシノニム

次のシノニムは、Microsoft Azure Synapse Analytics のストアドプロシジャを記述します。入力パラメータ、出力パラメータ、アンサーセットがそれぞれ 1 つあり、アンサーセットには 4 つの変数が含まれています。

マスターファイルシノニムは、以下のとおりです。

```
FILENAME=CUSTORDERS, SUFFIX=SQLADW , $
SEGMENT=INPUT, SEGTYPE=S0, $
  FIELDNAME=@CUSTOMERID, ALIAS=P0001, USAGE=A5, ACTUAL=A5,
  MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
SEGMENT=OUTPUT, SEGTYPE=S0, PARENT=INPUT, $
  FIELDNAME=@RETURN_VALUE, ALIAS=P0000, USAGE=I11, ACTUAL=I4, $
SEGMENT=ANSWERSET1, SEGTYPE=S0, PARENT=INPUT, $
  FIELDNAME=ORDERID, ALIAS=OrderID, USAGE=I11, ACTUAL=I4, $
  FIELDNAME=ORDERDATE, ALIAS=OrderDate, USAGE=HYMDs, ACTUAL=HYMDs,
  MISSING=ON, $
  FIELDNAME=REQUIREDDATE, ALIAS=RequiredDate, USAGE=HYMDs,
  ACTUAL=HYMDs, MISSING=ON, $
  FIELDNAME=SHIPPEDDATE, ALIAS=ShippedDate, USAGE=HYMDs,
  ACTUAL=HYMDs, MISSING=ON, $
```

アクセスファイルシノニムは、以下のとおりです。

```
SEGNAME=INPUT, CONNECTION=ITarget, STPNAME=Northwind.dbo.CustOrders, $  
SEGNAME=OUTPUT, STPRESORDER=0, $  
SEGNAME=ANSWERSET1, STPRESORDER=1, $
```

ストアドプロシジャのシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

[ストアドプロシジャ] を選択します。

データベース

テーブルやオブジェクトの選択元となるデータベースを指定するには、次のいずれかを実行します。

- [デフォルトデータベース] を選択して、デフォルトのデータベースとして設定されているデータベースを使用します。
- [データベース] ドロップダウンリストからデータベースを選択します。ここには、現在の DBMS インスタンスにあるすべてのデータベースが表示されます。

[デフォルトデータベース] が選択されている場合は、データベースを選択する前に、このチェックをオフにします。

選択

プロシジャを選択します。次のシノニム作成ウィンドウの [値] テキストボックスで、プロシジャごとに固有の値を入力する必要があるため、1 回に選択できるプロシジャは、1 つだけです。

名前

シノニム名です。デフォルト設定は、ストアドプロシジャ名です。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のストアドプロシジャが存在する場合は、接頭語または接尾語を割り当てて、対応するシノニムを区別します。なお、シノニム名の最大文字数は 64 バイトです。

すべてのプロシジャに固有の名前が付いている場合は、[接頭語] および [接尾語] のテキストボックスは空白にしておきます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[Microsoft SQL Server ODBC データタイプのサポート](#)を参照してください。

値

指定したプロシジャに表示されるパラメータすべてについて、このチェックボックスを選択します。

選択したプロシジャに入力パラメータ (IN パラメータ、IN モードの INOUT パラメータ) がある場合、それらのパラメータの入力が要求されます。ただし、[値] テキストボックスに明示的に入力する必要があるかどうかは、プロシジャのロジックおよび生成されるデータ構造に依存します。このため、パラメータボックスのチェックはオンにする必要はありますが、値の入力は不要な場合もあります。次のガイドラインに従います。

- 入力パラメータ値によりアンサーセットのデータ構造が変化する場合、明示的な入力値 (によりシノニムを区別すること) が必要です。
- プロシジャの内部ロジックが分かっており、常に同一のデータ構造が生成されることが確定している場合は、明示的な入力値は必要ありません。この場合、生成する必要のあるシノニムは 1 つだけで、シノニム作成の目的では、[値] テキストボックスは空白にすることができます。

[値] が必要な場合は、引用符を含めずに値を入力します。日付、日付時間、タイムスタンプのパラメータ値は、ISO の形式に従って入力します。明示的な値が必要なプロシジャの

場合、入力パラメータにはプロシジャが実行される際に入力されるパラメータと同一の値を指定します。

ストアドプロシジャに対するレポートの実行

ストアドプロシジャのアンサーセットに対してレポートを実行する場合は、データベーステーブルに対してレポートを実行する際と同一の機能を使用することができます。

- **SQL SELECT ステートメント** 構文についての詳細は、[SELECT を使用したストアドプロシジャに対するレポートの実行](#)を参照してください。
- **TABLE および GRAPH コマンド** 構文についての詳細は、[TABLE コマンドを使用したストアドプロシジャに対するレポートの実行](#)を参照してください。

ストアドプロシジャのアンサーセットとの JOIN には、次の 2 つのファイルを使用します。

- **ホストファイル** OUTPUT および ANSWERSET セグメントのみです。
- **クロスリファレンス** ファイル内の INPUT セグメントのみです。

TABLE コマンドを使用したストアドプロシジャに対するレポートの実行

TABLE コマンドを使用してストアドプロシジャを実行するには、次の構文を使用します。

```
TABLE FILE synonym
PRINT [parameter [parameter] ... | *]
[IF in-parameter EQ value]
.
.
.
END
```

説明

synonym

実行するストアドプロシジャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアドプロシジャにパラメータを必要としない場合は、アスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

*

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

IF

IF または WHERE キーワードです。IN パラメータまたは IN モードの INOUT パラメータに値を渡す際に使用します。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

注意：アダプタが Unicode をサポートするよう構成されている場合は、IN パラメータの長さは 1000 バイト以内にする必要があります。

value

パラメータに渡す値です。

SELECT を使用したストアドプロシジャに対するレポートの実行

```
SQL
SELECT [parameter [,parameter] ... | *] FROM synonym
[WHERE in-parameter = value]
.
```

```

.
.
END

```

説明

synonym

実行するストアドプロシジャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアドプロシジャにパラメータを必要としない場合は、構文にアスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

★

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

WHERE

IN パラメータまたは IN モードの INOUT パラメータに値を指定する場合に使用します。

各パラメータの値は、それぞれ異なる行に指定する必要があります。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

value

パラメータに渡す値です。

Microsoft Azure Synapse Analytics 環境のカスタマイズ

Microsoft Azure Synapse Analytics アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。ここでは、カスタマイズオプション

の概要について説明します。

タイムアウト制限の指定

Microsoft Azure Synapse Analytics に対して SQL リクエストを実行した後、アダプタがその応答を待機する時間 (秒) を指定するには、TIMEOUT コマンドを使用します。

TIMEOUT コマンドの発行

```
ENGINE SQLADW SET TIMEOUT {nn|0}
```

説明

SQLADW

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nn

タイムアウトとなる秒数を指定します。デフォルト値は 30 です。

0

応答の待機時間が無制限であることを示します。

長時間実行中リクエストのキャンセル

WebFOCUS Reporting Server ブラウザインターフェースで、長時間実行中のリクエストをキャンセルすることができます。この操作を実行すると、ネイティブ JDBC ドライバの機能に応じて、リクエスト全体がキャンセルされるか、フェッチサイクルが中断されます。

長時間実行中のリクエストをキャンセルするには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースで [ワークスペース] ページを開きます。ナビゲーションウィンドウで [Java サービス] を展開します。[DEFAULT] を右クリックし、[エージェント] を選択します。
2. [Java サービスエージェント] ウィンドウで、キャンセルする jscomid の行を選択し、右クリックして [終了] を選択します。

ログイン待機時間の指定

LOGINTIMEOUT コマンドを使用して、アダプタが接続時に Microsoft Azure Synapse Analytics からの応答を待機する時間を秒単位で指定することができます。

注意：アダプタの従来のリリースとの互換性を維持するため、TIMEOUT を LOGINTIMEOUT のシノニムとして使用することが可能です。

ログイン待機時間の指定

```
ENGINE SQLADW SET LOGINTIMEOUT|TIMEOUT {nn|0}
```

説明

SQLADW

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nn

タイムアウトとなる秒数を指定します。デフォルト値は約 15 秒です。

0

ログイン応答の待機時間が無制限であることを示します。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、メニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

使用可能なパラメータは次のとおりです。

```
ENGINE SQLADW SET PASSRECS {ON|OFF}
```

説明

SQLADW

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

Microsoft Azure Synapse Analytics 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)および[定数として定義された一時項目 \(DEFINE\) を転送するリクエストの最適化](#)を参照してください。

一時項目 (DEFINE) に NULL 値が含まれる場合のリクエストの最適化

SET OPTNOAGGR コマンドを使用して、アダプタの最適化に関する動作を微調整することができます。このコマンドは、何らかの理由で RDBMS に集計処理を渡したくない場合に使用します。その理由の一例として、集計データに NULL 値が含まれる場合があります。SET OPTNOAGGR コマンドを使用すると、アダプタは DBMS に集計処理を渡さずに SQL を生成します。結合およびソート処理は RDBMS で実行されますが、集計処理はサーバ側で内部的に実行されます。

DEFINE フィールドに NULL フィールドを使用する計算が含まれている場合は常に NULL が返されるため、これらの処理を SQL に変換して DBMS に渡すことはできません。そのような処理は FOCUS で処理する必要があります。

これには SET OPTIMIZATION OFF を設定します。

ただし、パフォーマンスを考慮した場合、AGGREGATION は FOCUS で実行して、DBMS に対してはオフロードの JOIN および SORT 処理を使用すると良いケースもあります。

強化された集計制御の設定

```
SQL SQLADW SET OPT {AGGR|NOAGGR}
```

説明

AGGR

アダプタは DEFINE フィールドの集計処理を RDBMS にオフロードします。この設定がデフォルト値です。

NOAGGR

アダプタは SQL を生成しますが、集計処理を RDBMS に渡しません。結合およびソート処理は RDBMS で実行されますが、集計処理はサーバ側で内部的に実行されます。この設定は、RDBMS にオフロードされる SQL が少なかった旧バージョンの機能に基づいて記述されたアプリケーションの下位互換性を提供する目的で使用することもできます。たとえば、集計フィールドの計算に NULL データが含まれている場合、その NULL データは RDBMS の NVL() 関数では処理されませんでした。

常に FALSE になる条件を含む IF-THEN-ELSE 最適化の使用

```
SQL SQLADW SET OPTIFTHENELSE ON
DEFINE FILE EMPINFO
DEF3 = IF FIRST_NAME EQ 'RITA' THEN 1 ELSE 0;
END

TABLE FILE EMPINFO
PRINT FIRST_NAME
IF DEF3 EQ 2
END
```

条件 DEF3 EQ 2 が True になることはないため、アダプタは RDBMS に WHERE 条件 1=0 (常に False) を渡します。RDBMS からレコードは返されません。

```
SELECT T1."FN" FROM USER1."EMPINFO" T1 WHERE (1 = 0) FOR FETCH ONLY;
```

DEFINE 式の最適化での SQL 制限

FOCUS レポート言語は固有の SQL より広範囲に及ぶため、データアダプタは特有の DEFINE 式を RDBMS に渡すことができません。DEFINE 式に以下のものが含まれている場合、データアダプタは DEFINE ベースの集計およびレコード選択を RDBMS にオフロードし

ません。

- ユーザ作成サブルーチン。
- 以下のような自己参照式。

```
X=X+1;
```

- 数値から文字フィールドへの変換、または文字から数値フィールドへの変換を実行する EDIT 関数。
- フィールドの値を変換する DECODE 関数。
- 関係演算子の INCLUDES および EXCLUDES。
- FOCUS サブルーチンの ABS、INT、MAX、MIN、LOG、SQRT。
注意： FOCUS サブルーチンの MAX および MIN と、演算接頭語の MAX. および MIN. を混同しないように注意してください。DEFINE フィールドに演算接頭語を含めることはできません。
- ACTUAL=DATE が指定されたフィールドを含む式。ただし、DATE フィールドから別の DATE フィールドを減算する式、および DATE フィールドのすべての論理式は除きます。
- FOCUS の日付時間関数により処理された日付時間は、SQL には変換されません。
- Financial Modeling Language (FML) のセル計算。FML は、拡張マトリクスレポートリング (EMR) と呼ばれます。

注意： FML レポートリクエストは拡張 TABLE リクエストです。FML は、明細レポートの特殊な機能を提供します。

また IF-THEN-ELSE の最適化では、以下の機能はサポートされません。

- すべてのタイプの DECODE 式
- 静的 SQL
- IF/WHERE DDNAME
- 部分的な日付の選択

検索処理でのブロックサイズの指定

Microsoft Azure Synapse Analytics アダプタは、SELECT クエリまたはストアドプロシージャの実行によって生成された、結果セットからの配列検索をサポートします。この手法によ

り、ネットワークおよび CPU 使用の負荷を大幅に軽減することができます。

大きい値を指定すると仮想記憶域のコストは増えますが、大量の行を処理するリクエストの効率性は改善されます。ただし、100 を超える値を指定しても僅かな効率改善しか期待できないため、100 以内の値を設定することをお勧めします。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、メニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

配列検索のブロックサイズの指定

SELECT リクエストのブロックサイズは、TABLE FILE リクエスト、MODIFY リクエスト、MATCH リクエスト、および DIRECT SQL SELECT ステートメントに適用されます。

```
ENGINE SQLADW SET FETCHSIZE n
```

説明

SQLADW

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

配列検索手法を使用して一度に取得する行数です。受容値は 1 から 5000 です。アダプタに応じてデフォルト値は異なります。CLOB または BLOB として処理する必要のあるフィールドが結果セットに含まれている場合は、その結果セットに使用する FETCHSIZE 値は 1 です。

挿入処理のブロックサイズの指定

INSERT のブロックサイズは、LOADONLY と組み合わせて MODIFY INCLUDE リクエストに適用されます。INSERTSIZE は、パラメータ化 DIRECT SQL INSERT ステートメントでもサ

ポートされます。

```
ENGINE SQLADW SET INSERTSIZE n
```

説明

SQLADW

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

配列検索手法を使用して挿入する行数です。受容値は 1 から 5000 です。BLOB として処理する必要のあるフィールドが結果セットに含まれている場合は、その結果セットに使用する INSERTSIZE 値は 1 です。

バルクインサート API の省略

```
ENGINE SQLADW SET BULKLOAD [ON|OFF]
```

説明

SQLADW

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

バルクインサート API を使用します。デフォルト値は ON です。

OFF

バルクインサート API の使用を抑制します。

バルクインサート API の動作

ibi Data Migrator で Microsoft Azure Synapse Analytics のバルクインサート API を使用することができます。

Microsoft Azure Synapse Analytics アダプタでは、LOADONLY モードでバルク API が自動的に使用されます。その値は、中間的なフラッシュがパフォーマンスに影響しないことを示します。このためその動作は INSERTSIZE に依存しません。

ロード中に発生するエラー (例、重複エラー) により、行のバッチ全体が拒否される場合があります。

非等価条件 WHERE ベース LEFT OUTER JOIN の最適化

LEFT OUTER JOIN は、ホストテーブルからすべてのレコードを選択し、それらをクロスリファレンステーブルのレコードと照合します。一致するレコードが存在しない場合、ホストレコードは保持されますが、クロスリファレンスフィールドにはデフォルト値 (空白またはゼロ) が割り当てられます。アダプタは、RDBMS で条件式がサポートされるすべての WHERE ベースの LEFT OUTER JOIN コマンドを最適化することができます。

条件付き LEFT OUTER JOIN の指定

```
JOIN LEFT_OUTER FILE hostfile AT hfld1 [TAG tag1]
  [WITH hfld2]
  TO {UNIQUE|MULTIPLE}
  FILE crfile AT crfld [TAG tag2] [AS joinname]
  [WHERE expression1;
  [WHERE expression2;
  ...]

END
```

説明

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。

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 パラメータにフィールド値を指定しても、この値が接続情報として使用されることはありません。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

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に一意の修飾子として使用するタグ名 (オプション) です。

WITH 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 に基づく条件付き JOIN では、KEEPDEFINES を ON に設定し、一時項目 (DEFINE) を JOIN コマンドを発行する前に作成しておく必要があります。

MULTIPLE

from_file と *to_file* 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。なお、ONE は UNIQUE の同義語です。

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 (文字フィールドは空白、数値フィールドは 0 (ゼロ)) が指定されます。

ユニーク JOIN は WebFOCUS の概念です。RDBMS はユニークと非ユニークを区別しないため、クロスリファレンスファイルから条件に一致するすべての行を取得します。

たとえば、リクエストでユニーク JOIN として指定したが、実際には条件に一致する行がクロスリファレンスファイルに複数存在する場合、この JOIN の処理を RDBMS が行くと、RDBMS は一致するすべての行を返します。一方、最適化を無効にしてこの JOIN の処理を WebFOCUS が行くと、異なるレポート結果が得られます。WebFOCUS はユニーク JOIN の概念を順守するため、ホストの 1 行に対してクロスリファレンスの行を 1 行だけ返します。

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データソース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。

WHERE ベース OUTER JOIN の最適化条件

- WHERE ベースの LEFT OUTER JOIN を最適化するには、それに関係する RDBMS で式の最適化を行うことができるとともに、次の条件のいずれかを満たす必要があります。
 - JOIN WHERE コマンドに、少なくとも 1 つの「field1 EQ field2」述語が含まれている。ここで、field1 は table1 のフィールド、field2 は table2 のフィールドを表します。
または
 - 右側のテーブルに、NULL データを含まないユニークインデックスまたはキーが

ある。

または

- 右側のテーブルに、長いデータタイプ (例、TEXT、IMAGE) を持たない「NOT NULL」フィールドが少なくとも 1 つ含まれている。
- アダプタの SQLJOIN OUTER 設定は、ON (デフォルト) に設定しておく必要があります。

非等価条件 LEFT OUTER JOIN の最適化

次のリクエストは、2 つの MSSQL データソース間の条件付き LEFT OUTER JOIN を作成し、結合後のデータソースを使用してレポートを作成します。このリクエストに対して生成された SQL を表示するため、STMTRACE がオンに設定されています。

```
SET TRACEUSER = ON
SET TRACEOFF = ALL
SET TRACEON = STMTRACE//CLIENT
JOIN LEFT_OUTER FILE employee AT EMPLOYEEID
TO ALL FILE employeepayhistory AT EMPLOYEEID
WHERE RATECHANGEDATE GT HIREDATE;
END
TABLE FILE employee
PRINT RATE
BY EMPLOYEEID
END
```

WebFOCUS リクエストは、LEFT OUTER JOIN を組み込んだ単一の MSSQL SELECT ステートメントに変換され、非等価条件が ON 句で RDBMS に渡されます。

```
SELECT T1."EmployeeID", T1."HireDate", T2."EmployeeID",
T2."RateChangeDate", T2."Rate" FROM
AdventureWorks.HumanResources.Employee T1 LEFT OUTER JOIN
AdventureWorks.HumanResources.EmployeePayHistory T2 ON
(T2."RateChangeDate" > T1."HireDate") ) ORDER BY T1."EmployeeID";
```

SQL パススルーによる Microsoft Azure Synapse Analytics ストアドプロシジャの呼び出し

Microsoft Azure Synapse Analytics のストアドプロシジャは、SQL パススルーの使用をサポートします。これらのプロシジャは、CREATE PROCEDURE コマンドを使用して Microsoft Azure Synapse Analytics 内で作成する必要があります。

アダプタは、IN、OUT、INOUT パラメータを含むストアドプロシジャをサポートします。

ストアドプロシジャによって返される出力パラメータ値は結果セットとして利用可能になります。単一行の結果セットのこれらの値は、呼び出したストアドプロシジャによって他のすべての結果セットが返された後で、クライアントに転送されます。(利用可能な場合の) 出力パラメータ名は、結果セットのカラムタイトルになります。

クライアントに返されるのは、呼び出し文字列内で参照されている出力パラメータ (およびその戻り値) だけです。結果として、ユーザは表示する出力パラメータ値を完全に制御することができます。

サーバは、基となる DBMS の規則に従って記述されたストアドプロシジャの呼び出しをサポートします。ここでの例は SQL ベースであることに注意してください。DBMS の規則、言語、および他のプログラミングの例については、DBMS のマニュアルを参照してください。

ストアドプロシジャの呼び出し

```
SQL SQLADW EX procname [parameter_specification1]
[,parameter_specification2]...
END
```

説明

SQLADW

Microsoft Azure Synapse Analytics の ENGINE 接尾語です。

procname

ストアードプロシージャ名です。ネイティブデータベース構文のストアードプロシージャの完全 (または部分) 修飾名です。

SQL または SYS のいずれかの命名規則を使用して、複数構成名の解釈に使用される区切り文字を制御することができます。詳細は、命名規則の設定を参照してください。

parameter_specification

IN、OUT、INOUT がサポートされています。ストアードプロシージャにより異なる適切なパラメータを指定します。

IN

リテラル (例、125、3.14、'abcde') です。入力文字に予約語を使用することも可能です。予約語は文字リテラルとは異なり、引用符で囲む必要はありません (例、NULL)。この値は必須です。

OUT

1 つの疑問符 (?) で表されます。このパラメータの有無により、出力結果をアプリケーションへ渡すかどうかを決定します。省略すると、この項目は (0 バイトの) 空白文字列になります。

INOUT

出力を表す 1 つの疑問符 (?) と入力を表すリテラルで構成されます。出力と入力はスラッシュ (/) で区切ります (例、?/125、?/3.14、?/'abcde')。出力値を (0 バイトの) 空白文字列にすることも可能です。

ストアードプロシージャの呼び出し

この例では、ユーザが edaqa.test_proc01 ストアードプロシージャを呼び出してパラメータ 1、3、5、7 の入力値を指定することで、このストアードプロシージャの戻り値、およびパラメータ 2、3 の出力値を取得します。

パラメータ 4 と 6 が省略されていることに注意してください。ストアードプロシージャは、作成時に指定されたデフォルト値を使用します。

```
SQL SQLADW EX edaqa.test_proc01 125,?,?,?/3.14,,'abc',,'xyz'
END
```

ストアドプロシジャの例

このストアドプロシジャは、OUT および INOUT パラメータを使用します。

```
CREATE PROCEDURE EDAQA.PROCP3 (    OUT chSQLSTATE_OUT    CHAR(5),
                                OUT intSQLCODE_OUT    INT,
                                INOUT l_name char(20),
                                INOUT f_name char(20))

    RESULT SETS 1
    LANGUAGE SQL

-----
-- SQL Stored Procedure
-----

P1:BEGIN
    -- Declare variable
    DECLARE SQLSTATE CHAR(5) DEFAULT '00000';
    DECLARE SQLCODE INT DEFAULT 0;
    -- Declare cursor
    DECLARE cursor1 CURSOR WITH RETURN FOR
        SELECT
            EDAQA.NF29005.SSN5 AS SSN5,
            EDAQA.NF29005.LAST_NAME5 AS LAST_NAME5,
            EDAQA.NF29005.FIRST_NAME5 AS FIRST_NAME5,
            EDAQA.NF29005.BIRTHDATE5 AS BIRTHDATE5,
            EDAQA.NF29005.SEX5 AS SEX5
        FROM
            EDAQA.NF29005
        WHERE
            (
                ( EDAQA.NF29005.LAST_NAME5 = l_name )
            AND
                ( EDAQA.NF29005.FIRST_NAME5 = f_name )
            );
    -- Cursor left open for client application
    OPEN cursor1;
    SET chSQLSTATE_OUT = SQLSTATE;
    SET intSQLCODE_OUT = SQLCODE;
    SET l_name = 'this is first name';
    SET f_name = 'this is last name';
END P1    @
```

ストアドプロシジャのアプリケーションエラーの収集

RAISERROR メソッドを使用して、アプリケーションエラーを収集することができます。ストアドプロシジャで生じるアプリケーションエラーは、サーバ変数 &ADWMSGTXT で使用することができます。

Microsoft SQL Server アダプタの使用

Microsoft SQL Server アダプタの構成方法について説明します。

Microsoft SQL Server アダプタを使用すると、アプリケーションから Microsoft SQL Server データソースへのアクセスが可能になります。このアダプタにより、データまたはアプリケーションリクエストがネイティブの Microsoft SQL Server ステートメントに変換され、最適化されたアンサーセットがリクエストの送信元プログラムに返されます。

Microsoft SQL Server 環境の準備

Microsoft SQL Server 2012/2014 対応アダプタの機能を十分に活用するには、同等以上のバージョンの Microsoft SQL Server Client を使用することをお勧めします。使用中の MS SQL Server のバージョンを確認するには、<http://support.microsoft.com/kb/321185> の情報を参照してください。

バージョン 2008 では、OLE DB バージョンのアダプタを使用してください (後述の説明を参照)。

Windows で環境を設定するには (OLE DB)

Microsoft SQL Server の環境は、Microsoft SQL Server、Client、MDAC ソフトウェアのインストール時に設定されます。

その他の設定の手順は必要ありません。

UNIX で環境を設定するには

環境変数 \$CLASSPATH を使用して、Microsoft SQL Server の JDBC ドライバファイルのパスを設定します。たとえば、JDBC ドライバファイルのパスを設定するには、次のように指定します。

```
CLASSPATH=/qas/mss/sqljdbc_4.0/enu/sqljdbc4.jar
export CLASSPATH
```

ドライバの sqljdbc4.jar ファイルは、Microsoft ダウンロードページ「Microsoft JDBC Driver 4.0 for SQL Server」の一部として提供される *.tar.gz ファイルに含まれています。必要に応じて、CLASSPATH または IBI_CLASSPATH を設定します。このクラス検索パスは、アダプタ構成手順の一部として設定することも、Java サービスのプロパティで設定することもできます (エクスポートは省略されるが、ディレクトリを指定する必要があります)。

環境変数 \$JDK_HOME を使用して、Java Development Kit のインストールディレクトリを設定します。Java バージョン 1.6 以降が必要です。たとえば、Java Development Kit のロケーションを /usr/java に設定するには、以下のように指定します。

```
JDK_HOME=/usr/java
export JDK_HOME
```

環境変数 \$LD_LIBRARY_PATH を使用して、Java 仮想マシンのインストールディレクトリを設定します。たとえば、Java 仮想マシンのディレクトリを設定するには、次のように指定します。

```
LD_LIBRARY_PATH=/usr/java/jdk1.6.0_38/jre/lib/amd64/server
export LD_LIBRARY_PATH
```

注意：セキュリティを ON に設定してサーバを実行している場合、LD_LIBRARY_PATH 変数は無視されます。この場合は、IBI_LIBPATH を使用する必要があります。

Microsoft SQL Server のリモートアクセス

リモートノード上の Microsoft SQL Server にアクセスすることが可能です。Microsoft SQL Server にリモートアクセスする手順は、以下のとおりです。

- Microsoft SQL Server Native Client の最新バージョンをローカルでインストールします。
- Microsoft SQL Server のリモートインスタンスの名前を確認します。

インストールした Microsoft SQL Server は、すべて一意の NetBEUI 名を使用して定義されます。有効なユーザ ID およびパスワードを定義している限り、Microsoft SQL Server のインスタンス名と同様、サーバはネットワーク上の任意の Microsoft SQL Server にアクセスする

ことができます。これらのパラメータは、サーバのグローバルプロファイルまたはユーザプロファイルに定義することができます。

XA のサポート

Microsoft SQL Server のデータソースにアクセスする読み取り/書き込みアプリケーションは、XA 互換モードで管理されたトランザクションを実行することができます。

XA トランザクション管理機能を有効にするには、WebFOCUS Reporting Server ブラウザインターフェースの構成機能を使用してサーバをトランザクション調整モードで構成する必要があります。トランザクション調整モードを使用することにより、関連するすべての DBMS に対するデータ修正の整合性が保証され、データ修正の一部がある DBMS に対してコミットされ、別の DBMS に対しては終了されることを回避することができます。

XA のサポートに関する詳しい説明は、[XA のサポート](#)を参照してください。

Microsoft SQL Server アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

アダプタは、Microsoft SQL Server データベースサーバに接続する際に、接続情報および認証情報を必要とします。この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。

この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。選択したプロファイルにコマンドが追加されます。選択可能なプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) があります。
- グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) のいずれかに手動でコ

マンドを追加します。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の Microsoft SQL Server データベースへの接続を宣言することができます。Microsoft SQL Server への実際の接続は、その接続を参照する最初のクエリ発行時に確立されます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。

[アダプタの構成を追加] ウィンドウが表示されます。

5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Microsoft SQL Server の接続属性

Microsoft SQL Server アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

サーバ (Windows のみ)

Microsoft SQL Server が稼働しているマシンの名前です。そのマシンに Microsoft SQL Server の複数のインスタンスがインストールされている場合、「server¥instance」のようにサーバ名およびインスタンス名を指定します。

サーバ接続属性により、ユーザは Microsoft SQL Native Client Enumerator でローカルネットワーク内に表示される Microsoft SQL Server のリストから選択できるようになります。

SQL Native Client Enumerator の制限、ローカルネットワークの設定、MS SQL Server の設定により、一部の稼働サーバが表示されない場合があることに注意してください。ターゲットにするサーバの名前が表示されていない場合は、その名前を直接入力することができます。

URL (UNIX のみ)

Microsoft SQL Server データソースのロケーション URL を入力します。

セキュリティ

Microsoft SQL Server への接続時にユーザを認証する方法には、次の 3 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報が Microsoft SQL Server データソースへの接続時に標準ログインの認証情報として渡されます。

このオプションを使用するには、SQL Server セキュリティとして SQL Server および Windows (Windows の場合)、または SQL Server および UNIX に設定する必要があります。

- **Password Passthru** (Windows のみ) クライアントアプリケーションから受け取ったユーザ ID およびパスワードが、Microsoft SQL Server への接続時に標準ログインの認証情報として渡されます。

このオプションを使用するには、SQL Server セキュリティとして SQL Server および Windows に設定する必要があります。

- **Trusted** アダプタは、サーバのデータアクセスエージェントによって偽装されたオペレーティングシステムユーザの認証情報を使用して、オペレーティングシステムログインとして Microsoft SQL Server に接続します。

このオプションは、いずれの SQL Server セキュリティの設定においても機能します。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

デフォルトデータベース (Windows のみ)

接続するデフォルトのデータベース名です。データオブジェクトがデータベース名によって修飾されていない場合、この値が使用されます。

このパラメータはオプションとして指定します。指定しない場合、認証 ID に関連するデータベースがデフォルト値として設定されます。

追加の接続文字列キーワード (Windows のみ)

Microsoft SQL Server マニュアルの Microsoft OLE DB ドライバの項に記述されている追加の接続文字列キーワードです。このパラメータはオプションとして指定します。

ドライバ名 (UNIX のみ)

Microsoft JDBC ドライバの名前を指定します。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

Windows での接続属性の手動宣言

Explicit 認証 ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報が Microsoft SQL Server データソースへの接続時に標準ログインの認証情報として渡されます。

```
ENGINE SQLMSS SET CONNECTION_ATTRIBUTES [connection]
server/userid,password [;dbname][:provider_string]
```

Password Passthru 認証 クライアントアプリケーションから受け取ったユーザ ID およびパスワードが、Microsoft SQL Server への接続時に認証情報として渡されます。

```
ENGINE SQLMSS SET CONNECTION_ATTRIBUTES [connection]
server/[;dbname][:provider_string]
```

Trusted 認証 アダプタは、サーバのデータアクセスエージェントによって偽装された Windows ユーザの認証情報を使用して、Windows ログインとして Microsoft SQL Server に接続します。

```
ENGINE SQLMSS SET CONNECTION_ATTRIBUTES [connection]
server/, [;dbname][:provider_string]
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

この特定の属性のセットを識別する論理名 (またはデータソース名) です。

Microsoft SQL Server への接続が 1 つの場合、このパラメータはオプションです。指定しない場合、ローカルのデータベースサーバがデフォルトの接続として設定されます。

server

Microsoft SQL Server が稼働しているマシンです。そのマシンに Microsoft SQL Server の複数のインスタンスがインストールされている場合、「server¥instance」のようにサーバ名およびインスタンスを指定します。

サーバ名およびインスタンス名を指定する場合、値を一重引用符 (') で囲むことをお勧めします。

userid

Microsoft SQL Server に登録されたプライマリ認証 ID です。

password

プライマリ認証 ID に関連するパスワードです。

dbname

この接続に使用される Microsoft SQL Server データベースの名前です。パスを含むデータベース名は、一重引用符 (') で囲む必要があります。

provider_string

ネットワークライブラリ名などの接続オプションの指定に使用される、Microsoft SQL Server のプロバイダ文字列です。このパラメータの先頭にはコロン (:) を付加し、パラメータ値を一重引用符 (') で囲む必要があります。このパラメータはオプションとして指定します。

注意：特殊文字を含む値は、一重引用符 (') で囲みます。値に一重引用符 (') が含まれている場合は、その一重引用符の前に別の一重引用符を追加する必要があります。結果的に、一重引用符が 2 つ連続することになります。たとえば、ブランクと一重引用符 (') の両方を含む「Mary O'Brien」をユーザ ID として指定する場合は、「'Mary O' 'Brian'」と入力します。

Windows での接続属性の宣言

以下の SET CONNECTION_ATTRIBUTES コマンドを実行すると、アプリケーションが明示的なユーザ ID (MYUSER) およびパスワード (PASS) を使用して、Microsoft SQL Server データベースサーバ SAMPLESERVER にアクセスすることができます。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

```
ENGINE SQLMSS SET CONNECTION_ATTRIBUTES SAMPLESERVER/MYUSER,PASS
```

以下の SET CONNECTION_ATTRIBUTES コマンドは、Password Passthru 認証を使用して Microsoft SQL Server データベースサーバ SAMPLESERVER に接続します。

```
ENGINE SQLMSS SET CONNECTION_ATTRIBUTES SAMPLESERVER/
```

以下の SET CONNECTION_ATTRIBUTES コマンドは、オペレーティングシステムの認証を使用してローカルの Microsoft SQL Server データベースサーバに接続します。

```
ENGINE SQLMSS SET CONNECTION_ATTRIBUTES /,
```

UNIX での接続属性の手動宣言

```
ENGINE SQLMSS SET CONNECTION_ATTRIBUTES CON1 'URL'/userid,password
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

CON1

接続名です。

URL

データソースのロケーション URL です。

userid

ターゲットデータベースに登録されたプライマリ認証 ID です。

password

プライマリ認証 ID に関連するパスワードです。

UNIX での接続属性の宣言

以下の SET CONNECTION_ATTRIBUTES コマンドは、Microsoft JDBC ドライバを使用することを指定します。

```
ENGINE SQLMSS SET JDBCDRIVERNAME  
com.microsoft.jdbc.sqlserver.SQLServerDriver
```

以下の SET CONNECTION_ATTRIBUTES コマンドは、明示的なユーザ ID (MYUSER) およびパスワード (PASS) を使用して、Microsoft SQL Server の JDBC ドライバを介して myServer に接続します。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

```
ENGINE SQLMSS SET CONNECTION_ATTRIBUTES CON1  
'jdbc:microsoft:sqlserver://myServer:1433'MYUSER,PASS
```

デフォルト接続の上書き変更

複数の接続を定義した場合、最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続がデフォルト接続として機能します。このデフォルト値は、SET DEFAULT_CONNECTION コマンドを使用して上書きすることができます。

デフォルト接続の変更

```
ENGINE SQLMSS SET DEFAULT_CONNECTION connection
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

```
ENGINE SQLMSS SET DEFAULT_CONNECTION SAMPLE
```

接続範囲の制御

確立する接続ごとにアダプタを使用する場合、SET AUTODISCONNECT コマンドを使用して接続の継続を制御することができます。

接続範囲の制御

```
ENGINE SQLMSS SET AUTODISCONNECT ON {FIN|COMMAND|COMMIT}
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMAND

各リクエストの後に自動的に切断されます。SET AUTODISCONNECT コマンドは、イベントがどの程度の頻度で発生するかによって、大幅なオーバーヘッドをもたらす可能性があります。このオーバーヘッドのほとんどすべてがサーバには関連しません。このオーバーヘッドは、オペレーティングシステムとデータソースに関連します。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

Microsoft SQL Server メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および Microsoft SQL Server データタイプのサーバマッピングを定義するシノニムを作成します。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な Microsoft SQL Server テーブルまたはビューの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シ

ノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

このアダプタでは、MS SQL ネイティブシノニムの特定のタイプでのメタデータの作成がサポートされます。この機能は、Microsoft SQL Server バージョン 2005 以降でのみ使用することができます。

ネイティブシノニムのメタデータは、次の条件下で作成することができます。

- ネイティブシノニムは、TABLE または VIEW ベースオブジェクトから作成されている必要があります。
- ベースオブジェクト (TABLE または VIEW) は、アダプタ接続文字列がターゲットとする MS SQL Server 上に存在している必要があります。
- ネイティブシノニムは、最大で 3 部構成の名前で記述されたベースオブジェクトで作成されている必要があります。ターゲットサーバ内でのベースオブジェクトのパスおよびオーナーシップに応じて、受容可能なフォーマットは次のとおりです。

```
object_name, schema_name.object_name,
```

または

```
database_name.schema_name.object_name.
```

次のタイプのネイティブシノニムはサポートされません。

- ストアドプロシジャに基づいているもの。
- リンクされた MS SQL Server 上のオブジェクトに基づいているもの。
- 次のような 4 部構成の名前で記述されたオブジェクトに基づいているもの。

```
server_name.database_name.schema_name.object_name
```

ストアドプロシジャのシノニムの作成、およびストアドプロシジャに対するレポートの実行についての詳細は、[ストアドプロシジャのシノニムの作成](#)を参照してください。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Microsoft SQL Server のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

重要：オブジェクトタイプにストアドプロシジャを選択した場合、入力パラメータはここで説明するものとは多少異なります。詳細は、[ストアドプロシジャに対するレポートの実行](#)を参照してください。

データベース

テーブルやオブジェクトの選択元となるデータベースを指定するには、次のいずれかを実行します。

- [デフォルトデータベース] を選択して、デフォルトのデータベースとして設定されているデータベースを使用します。
- [データベース] ドロップダウンリストからデータベースを選択します。ここには、

現在の DBMS インスタンスにあるすべてのデータベースが表示されます。

[デフォルトデータベース] が選択されている場合は、データベースを選択する前に、このチェックをオフにします。

目的のデータベースを指定するには、[データベース] ドロップダウンメニューからデータベースを選択します。ドロップダウンメニューには、Microsoft SQL Server のターゲットインスタンスのすべてのデータベースがリストされます。デフォルトデータベースを選択すると、接続の構成時に設定されたデータベースが保持されます。構成時にデフォルトデータベースが設定されなかった場合、SQL Server 上でアクティブなログインに割り当てられているデータベースがデフォルトとして使用されます。

外部 SQL スクリプトのパス

[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合は、さらに次の項目が表示されます。

次の標準名前規則は UNIX に適用されます。

- [ベースパス] テキストボックスに、SQL クエリが記述されたファイルの物理ディレクトリパスを指定します。ディレクトリ名を直接入力するか、[...] (参照) ボタンをクリックしてファイルの格納先ディレクトリへ移動します。このクリックにより、[ベースパスの選択] ダイアログボックスが表示されます。
- [ドキュメント名] テキストボックスに、ファイル名を入力します。必要に応じて、ワイルドカード文字を使用します。
- [ドキュメント拡張子] テキストボックスに、スクリプトファイルの拡張子を入力して、その拡張子のファイルのみを表示します。

アダプタはシノニムの作成中にネイティブ API コールを発行して選択リスト内の要素のリストを取得し、各要素に対するフィールドを持つマスターファイルを構成します。生成されたアクセスファイルは、DATASET 属性の SQL スクリプトパスを参照します。これには SQL クエリが格納されたファイルの名前と拡張子を含む、フルパスが記述されています。以下はその例です。

```
DATASET=/u1/home2/apps/report3.sql
```

WebFOCUS レポートを作成する際は、SQL クエリを使用してデータにアクセスします。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決

定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

外部キーを使用したクラスタ作成 (非推奨。[シノニムの編集] の使用を推奨)

[外部キーを使用したクラスタ作成] のチェックをオンにすると、現在のテーブルに外部キーで関係付けられているテーブルすべてをこのシノニムに含めることができます。ただし、このオプションは廃止される予定で、クラスタの作成にはシノニムエディタを使用することをお勧めします。生成されるマルチテーブルのシノニムは、このテーブルの外部キーの関係をすべて定義します。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (;) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[Microsoft SQL Server データタイプのサポート](#)を参照してください。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

テーブル名

基になるオブジェクトの名前です。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

生成されたシノニムの例

Microsoft SQL Server アダプタのシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

生成されたマスターファイル nf29004.mas

```
FILE=DIVISION, SUFFIX=SQLMSS , $
SEGNAME=SEG1_4, SEGTYPE=S0 , $
FIELD=DIVISION4, DIVISION4, I9, I4, MISSING=OFF , $
FIELD=DIVISION_NA4, DIVISION_NA4, A25, A25, MISSING=ON , $
FIELD=DIVISION_HE4, DIVISION_HE4, I9, I4, MISSING=ON , $
```

生成されたアクセスファイル nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=edaqa.nf29004,
CONNECTION=connmss, KEYS=1, WRITE=YES, $
```

Microsoft SQL テーブルのコメントのシノニムへのマッピング

Microsoft SQL テーブルまたはビューのシノニムを作成する際に、アダプタは次のようにコメントをマッピングします。

- MS SQL Server テーブルまたはビューのコメント (存在する場合) は、マスターファイルシノニムの REMARK 属性にマッピングされます。
- MS SQL Server カラムのコメント (存在する場合) は、マスターファイルシノニムの DESCRIPTION 属性にマッピングされます。

Unicode と 非 Unicode 両方のコメントがサポートされます。

また、MS SQL Server カラムのタイトル (存在する場合) は、マスターファイルシノニムの TITLE 属性にマッピングされます。

アクセスファイルのキーワード

下表は、アクセスファイルで使用されるキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABlename	Microsoft SQL Server のテーブルを指定します。以下のように完全修飾のテーブル名を指定することができます。 <pre>TABlename=[[database.]owner.]table</pre>
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre> CONNECTION=' ' は、ローカルのデータベースサーバへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。この属性を使用するには、キーを構成するフィールドをマスターファイルの先頭に記述する必要があります。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。

キーワード	説明
	<pre>KEY=fld1/fld2/.../fldn</pre>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。
KEYFLD IXFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定します。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 • KEYFLD は、親テーブルの共通カラムの FIELDNAME です。 • IXFLD は関連するテーブルの共通カラムの FIELDNAME です。 KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることをお勧めしますが、必須ではありません。 注意：KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。
AUTO INCREMENT	[Yes] に設定すると自動インクリメント機能が有効になります。
START	インクリメントの順序の初期値です。
INCREMENT	インクリメントの間隔です。

キーワード	説明
INDEX_NAME INDEX_UNIQUE INDEX_COLUMNS INDEX_ORDER	データベース内のインデックス名、およびインデックスフィールドの一意性、名前、順序を示します。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

Microsoft SQL Server データタイプのサポ

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[データタイプレポートを表示するには](#)を参照してください。

可変長データタイプのマッピングの制御

SET パラメータ VARCHAR は、Microsoft SQL Server のデータタイプ VARCHAR のマッピングを制御します。デフォルト設定では、サーバはこれらのデータタイプを可変長の文字 (AnV) としてマッピングします。

下表は、VARCHAR 値に基づくデータタイプのマッピングを示しています。

Microsoft SQL Server のデータタイプ	備考	VARCHAR ON		VARCHAR OFF	
		USAGE	ACTUAL	USAGE	ACTUAL
VARCHAR (<i>n</i>)	<i>n</i> は 1 から 8000 までの整数です。	<i>AnV</i>	<i>AnV</i>	<i>An</i>	<i>An</i>
NVARCHAR (<i>n</i>)	<i>n</i> は 1 から 4000 までの整数です。	<i>AnV</i>	<i>AnV</i>	<i>An</i>	<i>An</i>
VARBINARY (<i>n</i>)	<i>n</i> は 1 から 8000 までの整数です ($m = 2 * n$)。	<i>AmV</i>	<i>AmV</i>	<i>Am</i>	<i>Am</i>

可変長データタイプのマッピングの制御

```
ENGINE SQLMSS SET VARCHAR {ON|OFF}
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

Microsoft SQL Server のデータタイプ VARCHAR を可変長文字データタイプ (*AnV*) としてマッピングします。Unicode 環境ではこれは必須です。デフォルト値は ON です。

OFF

Microsoft SQL Server のデータタイプ VARCHAR を文字データタイプ (A) としてマッピングします。

国際言語サポートの有効化

SET パラメータ NCHAR は、文字セットが 1 バイト、2 バイト、3 バイトであることを示します。NCHAR の設定は、NCHAR および NVARCHAR データタイプのマッピングに影響します。

下表は、NCHAR 値に基づくデータタイプのマッピングを示しています。

Microsoft SQL Server のデータタイプ	備考	NCHAR SBCS		NCHAR DBCS		NCHAR TBCS	
		USAGE	ACTUAL	USAGE	ACTUAL	USAGE	ACTUAL
NCHAR (<i>n</i>)	<i>n</i> は 1 から 4000 までの整数です。 $d = 2 * n$ $t = 3 * n$	<i>An</i>	<i>An</i>	<i>Ad</i>	<i>Ad</i>	<i>At</i>	<i>At</i>
NVARCHAR (<i>n</i>)	<i>n</i> は 1 から 4000 までの整数です。 $d = 2$	<i>An</i>	<i>An</i>	<i>Ad</i>	<i>Ad</i>	<i>At</i>	<i>At</i>

Microsoft SQL Server のデータタイプ	備考	NCHAR SBCS		NCHAR DBCS		NCHAR TBCS	
		USAGE	ACTUAL	USAGE	ACTUAL	USAGE	ACTUAL
	$\begin{matrix} * \\ n \\ t = 3 * \\ n \end{matrix}$						

国際言語サポートの有効化

使用可能なパラメータは次のとおりです。

```
ENGINE SQLMSS SET NCHAR {SBCS|DBCS|TBCS}
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

SBCS

1 バイトの文字セットを示します。デフォルト値は、SBCS です。

DBCS

2 バイトの文字セットを示します。

TBCS

3 バイトの文字セットを示します。

SQL 式内での末尾ブランク

TABLE アダプタ内の新しい SQL ジェネレータでは、デフォルトで文字列リテラル内の末尾ブランク、および数値リテラル内の小数部と指数の表記を含むリテラルコンテンツを保持します。これにより、生成される SQL に対する制御が向上します。

末尾ブランクが不要なケースでは、次の構文が下位互換性を確保するために使用できます。

```
ENGINE SQLMSS SET TRIM_LITERALS ON
```

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアードプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

読み取り専用フィールドのサポート

CREATE SYNONYM コマンドは、TIMESTAMP として作成された Microsoft SQL Server のカラムまたは IDENTITY 属性が定義されたカラムに FIELDTYPE=R のフィールド定義を作成します。これらのフィールドは読み取り専用です。MAINTAIN または MODIFY プロシジャを実行する場合、アダプタはマスターファイルに FIELDTYPE=R が定義されたカラムに対して書き込み処理を実行しません。

読み取り専用フィールドのサポート

以下は、最初のカラムのプロパティが IDENTITY で、2 番目のカラムがタイムスタンプカラムのテーブルを作成する例です。

```
CREATE TABLE TAB1
(idproptab int IDENTITY (1,1), timestmp timestamp)
```

CREATE SYNONYM コマンドによって、このテーブルに以下のマスターファイルが生成されます。

```
FILE=TAB1, SUFFIX=SQLMSS , $
SEGNAME=TAB1, SEGTYPE=S0 , $
FIELD=IDPROPTAB, idproptab, I11, I4, MISSING=OFF, FIELDTYPE=R , $
FIELD=TIMSTMP, timestmp, A16, A16, MISSING=ON, FIELDTYPE=R , $
```

Microsoft SQL Server のストアプロシジャに対するレポートの実行

SELECT ステートメントや TABLE コマンドなどのレポートツールを使用し、Microsoft SQL Server のストアプロシジャを実行して、プロシジャの出力パラメータおよびアンサーセットに対するレポートを実行することができます。ストアプロシジャの実行という方法には、次の利点があります。

- アンサーセット以外に、出力パラメータである OUT パラメータおよび OUT モードの INOUT パラメータを検索することができる。他の呼び出し方法では、アンサーセットのみが検索されます。
- TABLE および他のレポートツールを使用して、出力パラメータおよびアンサーセットの処理、フォーマット設定、表示を簡単に行うことができる。

ストアプロシジャに対するレポートを実行するには、次の手順を実行します。

1. **シノニムの作成** ストアドプロシジャのアンサーセットのシノニムを作成します。詳細は、[ストアプロシジャのシノニムの作成](#)を参照してください。
2. **レポートプロシジャの作成** レポートプロシジャを作成します。詳細は、[ストアプロシジャに対するレポートの実行](#)を参照してください。
3. **レポートの実行** ストアドプロシジャを実行し、出力パラメータ (OUT および OUT

モードの INOUT) およびレポートに指定されたアンサーセットフィールドに対するレポートを実行します。

ストアドプロシジャのシノニムの作成

シノニムは、ストアドプロシジャのパラメータとアンサーセットを記述します。

アンサーセットの構造は、プロシジャの実行時に指定する入力パラメータ値により異なります。このため、プロシジャの実行時に指定される入力パラメータ値のセットごとに、個別のシノニムを作成する必要があります。たとえば、3つの異なる入力パラメータ値セットでストアドプロシジャを実行する場合、シノニムを3つ(値セットごとに1つ)作成する必要があります。なお、特に明記していない限り、「入力パラメータ」は、IN パラメータおよび IN モードの INOUT パラメータを意味します。

例外 ユーザがプロシジャの内部ロジックを理解しており、入力パラメータの値の範囲とプロシジャによって返された各アンサーセットの構造の関係を把握している場合、各アンサーセットの構造に対応する単一のシノニムを作成して、各シノニムにアンサーセットの構造を返すために必要な入力パラメータの値のセットを指定することが可能です。

シノニムには、次のセグメントを含めます。

- **INPUT** - このセグメントは、IN パラメータおよび IN モードの INOUT パラメータを記述します。
IN パラメータおよび IN モードの INOUT パラメータがない場合、このセグメントはダミーフィールドを1つ記述します。
- **OUTPUT** - このセグメントは、OUT パラメータおよび OUT モードの INOUT パラメータを記述します。
OUT パラメータおよび OUT モードの INOUT パラメータがない場合、このセグメントは省略されます。
- **ANSWERSET n** - 各アンサーセットに1つ含まれます。
アンサーセットがない場合、このセグメントは省略されます。

Microsoft SQL Server ストアドプロシジャ CustOrders のシノニム

次のシノニムは、Microsoft SQL Server のストアドプロシジャを記述します。入力パラメータ、出力パラメータ、アンサーセットがそれぞれ 1 つあり、アンサーセットには 4 つの変数が含まれています。

マスターファイルシノニムは、以下のとおりです。

```
FILENAME=CUSTORDERS, SUFFIX=SQLMSS , $
  SEGMENT=INPUT, SEGTYPE=S0, $
    FIELDNAME=@CUSTOMERID, ALIAS=P00001, USAGE=A5, ACTUAL=A5,
      MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
  SEGMENT=OUTPUT, SEGTYPE=S0, PARENT=INPUT, $
    FIELDNAME=@RETURN_VALUE, ALIAS=P00000, USAGE=I11, ACTUAL=I4, $
  SEGMENT=ANSWERSET1, SEGTYPE=S0, PARENT=INPUT, $
    FIELDNAME=ORDERID, ALIAS=OrderID, USAGE=I11, ACTUAL=I4, $
    FIELDNAME=ORDERDATE, ALIAS=OrderDate, USAGE=HYMDs, ACTUAL=HYMDs,
      MISSING=ON, $
    FIELDNAME=REQUIREDDATE, ALIAS=RequiredDate, USAGE=HYMDs,
      ACTUAL=HYMDs, MISSING=ON, $
    FIELDNAME=SHIPPEDDATE, ALIAS=ShippedDate, USAGE=HYMDs,
      ACTUAL=HYMDs, MISSING=ON, $
```

アクセスファイルシノニムは、以下のとおりです。

```
SEGNAME=INPUT, CONNECTION=ITarget, STPNAME=Northwind.dbo.CustOrders, $
SEGNAME=OUTPUT, STPRESORDER=0, $
SEGNAME=ANSWERSET1, STPRESORDER=1, $
```

ストアドプロシジャのシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

[ストアドプロシジャ] を選択します。

データベース

テーブルやオブジェクトの選択元となるデータベースを指定するには、次のいずれかを実行します。

- [デフォルトデータベース] を選択して、デフォルトのデータベースとして設定されているデータベースを使用します。
- [データベース] ドロップダウンリストからデータベースを選択します。ここには、現在の DBMS インスタンスにあるすべてのデータベースが表示されます。

[デフォルトデータベース] が選択されている場合は、データベースを選択する前に、このチェックをオフにします。

選択

プロシジャを選択します。次のシノニム作成ウィンドウの [値] テキストボックスで、プロシジャごとに固有の値を入力する必要があるため、1 回に選択できるプロシジャは、1 つだけです。

名前

シノニム名です。デフォルト設定は、ストアードプロシジャ名です。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のストアードプロシジャが存在する場合は、接頭語または接尾語を割り当てて、対応するシノニムを区別します。なお、シノニム名の最大文字数は 64 バイトです。

すべてのプロシジャに固有の名前が付いている場合は、[接頭語] および [接尾語] のテキストボックスは空白にしておきます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[Microsoft SQL Server データタイプのサポート](#)を参照してください。

値

指定したプロシジャに表示されるパラメータすべてについて、このチェックボックスを選択します。

選択したプロシジャに入力パラメータ (IN パラメータ、IN モードの INOUT パラメータ) がある場合、それらのパラメータの入力が要求されます。ただし、[値] テキストボックスに明示的に入力する必要があるかどうかは、プロシジャのロジックおよび生成されるデータ構造に依存します。このため、パラメータボックスのチェックはオンにする必要はありませんが、値の入力は不要な場合もあります。次のガイドラインに従います。

- 入力パラメータ値によりアンサーセットのデータ構造が変化する場合は、明示的な入力値 (によりシノニムを区別すること) が必要です。
- プロシジャの内部ロジックが分かっており、常に同一のデータ構造が生成されることが確定している場合は、明示的な入力値は必要ありません。この場合、生成する必要があるシノニムは 1 つだけで、シノニム作成の目的では、[値] テキストボックスはブランクにすることができます。

[値] が必要な場合は、引用符を含めずに値を入力します。日付、日付時間、タイムスタンプのパラメータ値は、ISO の形式に従って入力します。明示的な値が必要なプロシジャの場合、入力パラメータにはプロシジャが実行される際に入力されるパラメータと同一の値を指定します。

ストアドプロシジャに対するレポートの実行

ストアドプロシジャのアンサーセットに対してレポートを実行する場合は、データベーステーブルに対してレポートを実行する際と同一の機能を使用することができます。

- **SQL SELECT ステートメント** 構文についての詳細は、[SELECT を使用したストアドプロシジャに対するレポートの実行](#)を参照してください。
- **TABLE および GRAPH コマンド** 構文についての詳細は、[TABLE コマンドを使用した](#)

[ストアドプロシジャに対するレポートの実行](#)を参照してください。

ストアドプロシジャのアンサーセットとの JOIN には、次の 2 つのファイルを使用します。

- **ホストファイル** OUTPUT および ANSWERSET セグメントのみです。
- **クロスリファレンス** ファイル内の INPUT セグメントのみです。

TABLE コマンドを使用したストアドプロシジャに対するレポートの実行

TABLE コマンドを使用してストアドプロシジャを実行するには、次の構文を使用します。

```
TABLE FILE synonym
PRINT [parameter [parameter] ... | *]
[IF in-parameter EQ value]
.
.
.
END
```

説明

synonym

実行するストアドプロシジャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアドプロシジャにパラメータを必要としない場合は、アスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

★

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

IF

IF または WHERE キーワードです。IN パラメータまたは IN モードの INOUT パラメータに値を渡す際に使用します。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

注意：アダプタが Unicode をサポートするよう構成されている場合は、IN パラメータの長さは 1000 バイト以内にする必要があります。

value

パラメータに渡す値です。

SELECT を使用したストアプロシージャに対するレポートの実行

```
SQL
SELECT [parameter [,parameter] ... | *] FROM synonym
[WHERE in-parameter = value]
.
.
.
END
```

説明**synonym**

実行するストアプロシージャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアプロシージャにパラメータを必要としない場合は、構文にアスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

★

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

WHERE

IN パラメータまたは IN モードの INOUT パラメータに値を指定する場合に使用します。

各パラメータの値は、それぞれ異なる行に指定する必要があります。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

value

パラメータに渡す値です。

Microsoft SQL Server 環境のカスタマイズ

Microsoft SQL Server アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。ここでは、カスタマイズオプションの概要について説明します。

タイムアウト制限の指定

Microsoft SQL Server に対して SQL リクエストを実行した後、アダプタがその応答を待機する時間 (秒) を指定するには、TIMEOUT コマンドを使用します。

TIMEOUT コマンドの発行

```
ENGINE SQLMSS SET TIMEOUT {nn|0}
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nn

タイムアウトとなる秒数を指定します。デフォルト値は 30 です。

0

応答の待機時間が無制限であることを示します。

長時間実行中リクエストのキャンセル

WebFOCUS Reporting Server ブラウザインターフェースで、長時間実行中のリクエストをキャンセルすることができます。この操作を実行すると、ネイティブ JDBC ドライバの機能に応じて、リクエスト全体がキャンセルされるか、フェッチサイクルが中断されます。

長時間実行中のリクエストをキャンセルするには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [ワークスペース] ページで、[Java サービス] 下の [DEFAULT] を選択します。[DEFAULT] を右クリックし、[エージェント] を選択します。
2. [Java サービスエージェント] ウィンドウで、キャンセルする jscomid の行を選択し、右クリックして [終了] を選択します。

カーソルタイプの指定

SET CURSORS コマンドを使用して、検索するカーソルのタイプを指定することができます。

カーソルタイプの指定

使用可能なパラメータは次のとおりです。

```
ENGINE SQLMSS SET CURSORS [CLIENT|SERVER]
```

説明

CLIENT

データの検索に Microsoft SQL Server のクライアントサイドのカーソルを使用します。通常、クライアントサイドのカーソルによってデータの検索効率が改善され、Microsoft SQL Server の処理に役立ちます。TRANSACTIONS AUTOCOMMITTED モードを除いて、クライアントサイドのカーソルを使用することにより、サーバエージェントが Microsoft SQL Server の同じインスタンスから複数のアンサーセットを同時に読み取ることが回避されます。

SERVER

データの検索に Microsoft SQL Server のサーバサイドのカーソルを使用します。サーバサイドのカーソルの効率は、クライアントサイドのカーソルと比較して劣ります。ただし、FETCHSIZE に高い値 (アダプタのデフォルト値は 100) を設定することによりパフォーマンスは大幅に改善され、クライアントサイドのカーソルと同等となります。Microsoft SQL Server 処理の負荷を軽減するには、クライアントサイドのカーソルを使用することを推奨します。

blank

TRANSACTIONS AUTOCOMMITTED モードではクライアントサイドのカーソルを使用し、それ以外の場合はサーバサイドのカーソルを使用します。これがデフォルト値です。

ログイン待機時間の指定

LOGINTIMEOUT コマンドを使用して、アダプタが接続時に Microsoft SQL Server からの応答を待機する時間を秒単位で指定することができます。

注意：アダプタの従来のリリースとの互換性を維持するため、TIMEOUT を LOGINTIMEOUT のシノニムとして使用することが可能です。

ログイン待機時間の指定

```
ENGINE SQLMSS SET LOGINTIMEOUT|TIMEOUT {nn|0}
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nn

タイムアウトとなる秒数を指定します。デフォルト値は約 15 秒です。

0

ログイン応答の待機時間が無制限であることを示します。

NONBLOCK モードの有効化

Microsoft SQL Server アダプタには、NONBLOCK モードで呼び出しを実行する機能があります。デフォルト設定は、BLOCK モードです。

この機能により、アダプタはクライアントリクエストに応答して、エンジン処理の待ち状態からクエリを取り消すことができます。待機状態は、通常、SQL 解析中、アンサーセットの 1 行目がアダプタへ配信できる状態になる前、または別のアプリケーションによりロックされたオブジェクトへのアクセスの待機中に起こります。

NONBLOCK モードの有効化

使用可能なパラメータは次のとおりです。

```
ENGINE SQLMSS SET NONBLOCK {0|n}
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

正の数値です。デフォルト値はゼロ (0) です。これは、アダプタが BLOCK モードで動作することを意味します。数値が 1 以上の場合、NONBLOCK の呼び出しが有効になります。この数値は、アダプタが次の項目を確認する際の待機時間間隔 (秒) を指定します。

- 検索が実行されたか。
- クライアントアプリケーションが検索のキャンセルを要求したか。
- WebFOCUS Reporting Server ブラウザインターフェースの [セッションの終了] ボタンが押されたか。

注意：通常の操作には、値に 1 または 2 を指定します。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

使用可能なパラメータは次のとおりです。

```
ENGINE SQLMSS SET PASSRECS {ON|OFF}
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

トランザクションの制御

SET TRANSACTION コマンドを使用して、アダプタによるトランザクションの処理を制御することができます。

トランザクションの制御

使用可能なパラメータは次のとおりです。

```
ENGINE SQLMSS SET TRANSACTIONS {LOCAL|DISTRIBUTED|AUTOCOMMITTED}
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

LOCAL

処理を実行する接続ごとに、アダプタがローカルトランザクションを暗黙的に開始します。アダプタは COMMIT または ROLLBACK の実行時、あるいはサーバセッションの終了時に、各接続の処理を連続的にコミットするか、処理を中止します。デフォルト値は、LOCAL です。

DISTRIBUTED

アダプタが Microsoft 分散トランザクションコーディネータ (DTC) を暗黙的に呼び出して、すべての接続の処理を一括実行する単一の分散トランザクションを作成します。アダプタは COMMIT または ROLLBACK の実行時、あるいはサーバセッションの終了時に DTC を呼び出して、2 フェーズのコミットまたはロールバックプロトコルを実行します。この処理を実行するには、サーバが稼働しているマシン、および Microsoft SQL Server の呼び出し先インスタンスが存在するすべてのマシンで、DTC サービスを開始しておく必要があります。

このモードは、複数の接続で更新処理を同時に実行する読み取り/書き込みアプリケーションで使用することをお勧めします。

AUTOCOMMITTED

Microsoft SQL Server での個々の処理が、SQL Server によって即座にコミット (正常終了した場合) またはロールバック (エラーの場合) されます。このモードは、読み取り専用アプリケーションでパフォーマンスを考慮する場合に使用することをお勧めします。このモードを読み取り/書き込みアプリケーションに使用することはお勧めしません。このモードでは、複数の処理で構成された論理作業単位をロールバックすることができないためです。

トランザクションの分離レベルの指定

WebFOCUS Reporting Server ブラウザインターフェースから、または SET ISOLATION コマンドを使用して、トランザクションの分離レベルを指定することができます。

SET コマンドによるトランザクション分離レベルの指定

以下のコマンドを実行して、トランザクションの分離レベルを指定することができます。

```
ENGINE SQLMSS SET ISOLATION {RU|RC|RR|SE|CH|CS}
```

説明

RU

トランザクションの分離レベルを非コミット読み取りに設定します。

RC

トランザクションの分離レベルをカーソル固定に設定します。

RR

トランザクションの分離レベルを反復可能読み取りに設定します。

SE

トランザクションの分離レベルを読み取り固定に設定します。

CH

トランザクションの分離レベルを Chaos に設定します。

CS

トランザクションの分離レベルをカーソル固定に設定します (カーソル固定は読み取り固定のシノニムです)。

Microsoft SQL Server 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)および[定数として定義された一時項目 \(DEFINE\) を転送するリクエストの最適化](#)を参照してください。

一時項目 (DEFINE) に NULL 値が含まれる 場合のリクエストの最適化

SET OPTNOAGGR コマンドを使用して、アダプタの最適化に関する動作を微調整することができます。このコマンドは、何らかの理由で RDBMS に集計処理を渡したくない場合に使用します。その理由の一例として、集計データに NULL 値が含まれる場合などがあります。SET OPTNOAGGR コマンドを使用すると、アダプタは DBMS に集計処理を渡さずに SQL を生成します。結合およびソート処理は RDBMS で実行されますが、集計処理はサーバ側で内部的に実行されます。

DEFINE フィールドに NULL フィールドを使用する計算が含まれている場合は常に NULL が返されるため、これらの処理を SQL に変換して DBMS に渡すことはできません。そのような処理は FOCUS で処理する必要があります。

これには SET OPTIMIZATION OFF を設定します。

ただし、パフォーマンスを考慮した場合、集合計は FOCUS で実行して、DBMS に対してはオフロードの JOIN および SORT 処理を使用すると良いケースもあります。

強化された集計制御の設定

```
SQL SQLMSS SET OPT {AGGR|NOAGGR}
```

説明

AGGR

アダプタは DEFINE フィールドの集計処理を RDBMS にオフロードします。この設定がデフォルト値です。

NOAGGR

アダプタは SQL を生成しますが、集計処理を RDBMS に渡しません。結合およびソート処理は RDBMS で実行されますが、集計処理はサーバ側で内部的に実行されます。この設定は、RDBMS にオフロードされる SQL が少なかった旧バージョンの機能に基づいて記述されたアプリケーションの下位互換性を提供する目的で使用することもできます。たとえば、集計フィールドの計算に NULL データが含まれている場合、その NULL データは RDBMS の NVL() 関数では処理されませんでした。

常に FALSE になる条件を含む IF-THEN-ELSE 最適化の使用

```
SQL QLMSS SET OPTIFTHENELSE ON
DEFINE FILE EMPINFO
DEF3 = IF FIRST_NAME EQ 'RITA' THEN 1 ELSE 0;
END

TABLE FILE EMPINFO
PRINT FIRST_NAME
IF DEF3 EQ 2
END
```

条件 DEF3 EQ 2 が True になることはないため、アダプタは RDBMS に WHERE 条件 1=0 (常に False) を渡します。RDBMS からレコードは返されません。

```
SELECT T1."FN" FROM USER1."EMPINFO" T1 WHERE (1 = 0) FOR FETCH ONLY;
```

DEFINE 式の最適化での SQL 制限

FOCUS レポート言語は固有の SQL より広範囲に及ぶため、データアダプタは特有の DEFINE 式を RDBMS に渡すことができません。DEFINE 式に以下のものが含まれている場合、データアダプタは DEFINE ベースの集計およびレコード選択を RDBMS にオフロードしません。

- ユーザ作成サブルーチン。
- 以下のような自己参照式。

```
X=X+1;
```

- 数値から文字フィールドへの変換、または文字から数値フィールドへの変換を実行する EDIT 関数。
- フィールドの値を変換する DECODE 関数。
- 関係演算子の INCLUDES および EXCLUDES。
- FOCUS サブルーチンの ABS、INT、MAX、MIN、LOG、SQRT。

注意： FOCUS サブルーチンの MAX および MIN と、演算接頭語の MAX. および MIN. を混同しないように注意してください。DEFINE フィールドに演算接頭語を含めることはできません。

- ACTUAL=DATE が指定されたフィールドを含む式。ただし、DATE フィールドから別の DATE フィールドを減算する式、および DATE フィールドのすべての論理式は除きます。
- FOCUS の日付時間関数により処理された日付時間は、SQL には変換されません。
- Financial Modeling Language (FML) のセル計算。FML は、拡張マトリクスレポートリング (EMR) とも呼ばれます。

注意： FML レポートリクエストは拡張 TABLE リクエストです。FML は、明細レポートの特殊な機能を提供します。

また IF-THEN-ELSE の最適化では、以下の機能はサポートされません。

- すべてのタイプの DECODE 式
- 静的 SQL
- IF/WHERE DDNAME
- 部分的な日付の選択

検索処理でのブロックサイズの指定

Microsoft SQL Server アダプタは、SELECT クエリまたはストアードプロシージャの実行によって生成された、結果セットからの配列検索をサポートします。この手法により、ネットワークおよび CPU 使用の負荷を大幅に軽減することができます。

大きい値を指定すると仮想記憶域のコストは増えますが、大量の行を処理するリクエストの効率性は改善されます。ただし、100 を超える値を指定しても僅かな効率改善しか期待できないため、100 以内の値を設定することをお勧めします。

ヒント： この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

配列検索のブロックサイズの指定

SELECT リクエストのブロックサイズは、TABLE FILE リクエスト、MODIFY リクエスト、MATCH リクエスト、および DIRECT SQL SELECT ステートメントに適用されます。

```
ENGINE SQLMSS SET FETCHSIZE n
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

配列検索手法を使用して一度に取得する行数です。受容値は 1 から 5000 です。アダプタに応じてデフォルト値は異なります。CLOB または BLOB として処理する必要のあるフィールドが結果セットに含まれている場合は、その結果セットに使用する FETCHSIZE 値は 1 です。

挿入処理のブロックサイズの指定

INSERT のブロックサイズは、LOADONLY と組み合わせて MODIFY INCLUDE リクエストに適用されます。INSERTSIZE は、パラメータ化 DIRECT SQL INSERT ステートメントでもサポートされます。

```
ENGINE SQLMSS SET INSERTSIZE n
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

配列検索手法を使用して挿入する行数です。受容値は 1 から 5000 です。BLOB として処理する必要のあるフィールドが結果セットに含まれている場合は、その結果セットに使用する INSERTSIZE 値は 1 です。

バルクインサート API の省略

```
ENGINE SQLMSS SET FASTLOAD [ON|OFF]
```

説明

SQLMSS

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

バルクインサート API を使用します。デフォルト値は ON です。

OFF

バルクインサート API の使用を抑制します。

バルクインサート API の動作

ibi Data Migrator で Microsoft SQL Server のバルクインサート API を使用することができます。

Microsoft SQL Server アダプタでは、LOADONLY モードでバルク API が自動的に使用されます。その値は、中間的なフラッシュがパフォーマンスに影響しないことを示します。このためその動作は INSERTSIZE に依存しません。

ロード中に発生するエラー (例、重複エラー) により、行のバッチ全体が拒否される場合があります。

非等価条件 WHERE ベース LEFT OUTER JOIN の最適化

LEFT OUTER JOIN は、ホストテーブルからすべてのレコードを選択し、それらをクロスリファレンステーブルのレコードと照合します。一致するレコードが存在しない場合、ホストレコードは保持されますが、クロスリファレンスフィールドにはデフォルト値 (ブランクまたはゼロ) が割り当てられます。アダプタは、RDBMS で条件式がサポートされるすべての WHERE ベースの LEFT OUTER JOIN コマンドを最適化することができます。

条件付き LEFT OUTER JOIN の指定

```
JOIN LEFT_OUTER FILE hostfile AT hfld1 [TAG tag1]
    [WITH hfld2]
    TO {UNIQUE|MULTIPLE}
    FILE crfile AT crfld [TAG tag2] [AS joinname]
    [WHERE expression1;
    [WHERE expression2;
    ...]

END
```

説明

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。

hostfile

ホストマスタファイルです。

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 パラメータにフィールド値を指定しても、この値が接続情報として使用されることはありません。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

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に一意の修飾子として使用するタグ名 (オプション) です。

WITH 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 に基づく条件付き JOIN では、KEEPDEFINES を ON に設定し、一時項目 (DEFINE) を JOIN コマンドを発行する前に作成しておく必要があります。

MULTIPLE

from_file と *to_file* 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。なお、ONE は UNIQUE の同義語です。

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。

ユニーク JOIN は WebFOCUS の概念です。RDBMS はユニークと非ユニークを区別しないため、クロスリファレンスファイルから条件に一致するすべての行を取得します。

たとえば、リクエストでユニーク JOIN として指定したが、実際には条件に一致する行がクロスリファレンスファイルに複数存在する場合、この JOIN の処理を RDBMS が行くと、RDBMS は一致するすべての行を返します。一方、最適化を無効にしてこの JOIN の処理を WebFOCUS が行くと、異なるレポート結果が得られます。WebFOCUS はユニーク JOIN の概念を順守するため、ホストの 1 行に対してクロスリファレンスの行を 1 行だけ返します。

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データソース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。

WHERE ベース OUTER JOIN の最適化条件

- WHERE ベースの LEFT OUTER JOIN を最適化するには、それに関係する RDBMS で式の最適化を行うことができるとともに、次の条件のいずれかを満たす必要があります。
 - JOIN WHERE コマンドに、少なくとも 1 つの「field1 EQ field2」述語が含まれている。ここで、field1 は table1 のフィールド、field2 は table2 のフィールドを表します。
または
 - 右側のテーブルに、NULL データを含まないユニークインデックスまたはキーがある。
または
 - 右側のテーブルに、長いデータタイプ (例、TEXT、IMAGE) を持たない「NOT NULL」フィールドが少なくとも 1 つ含まれている。
- アダプタの SQLJOIN OUTER 設定は、ON (デフォルト) に設定しておく必要があります。

非等価条件 LEFT OUTER JOIN の最適化

次のリクエストは、2 つの MSSQL データソース間の条件付き LEFT OUTER JOIN を作成し、結合後のデータソースを使用してレポートを作成します。このリクエストに対して生成された SQL を表示するため、STMTRACE が ON に設定されています。

```
SET TRACEUSER = ON
SET TRACEOFF = ALL
SET TRACEON = STMTRACE//CLIENT
JOIN LEFT_OUTER FILE employee AT EMPLOYEEID
TO ALL FILE employeepayhistory AT EMPLOYEEID
    WHERE RATECHANGEDATE GT HIREDATE;
END
TABLE FILE employee
PRINT RATE
BY EMPLOYEEID
END
```

WebFOCUS リクエストは、LEFT OUTER JOIN を組み込んだ単一の MSSQL SELECT ステートメントに変換され、非等価条件が ON 句で RDBMS に渡されます。

```
SELECT T1."EmployeeID", T1."HireDate", T2."EmployeeID",
T2."RateChangeDate", T2."Rate" FROM
AdventureWorks.HumanResources.Employee T1 LEFT OUTER JOIN
AdventureWorks.HumanResources.EmployeePayHistory T2 ON
(T2."RateChangeDate" > T1."HireDate") ) ORDER BY T1."EmployeeID";
```

ヒントによるオプティマイザの効率化

DBMS オプティマイザのヒントを使用して、実行プランを変更することができます。アダプタには、TABLE コマンドの実行時に生成される Microsoft SQL Server 用クエリの末尾にヒントを追加する設定があります。

ヒントは、アダプタが単一の SELECT ステートメントを作成する際に追加されます。FOCUS 管理の JOIN で複数の SELECT ステートメントが生成される場合、ヒントは追加されません。

この設定を元に戻すには、hint_text パラメータを省略して SET HINT を使用します。

ヒントの設定

特定のヒントを設定するには、次の構文を使用します。

```
SQL SQLMSS SET HINT OPTION (hint_text)
```

説明

SQLMSS

ターゲットの RDBMS です。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

OPTION (hint_text)

単一ヒントのテキスト、または複数のヒントを組み合わせたテキストです。ユーザは、構文どおりに正確に入力する必要があります。OPTION (*hint_text*) を省略すると、ヒントを使用しない設定に戻ります。

Microsoft SQL ヒントの設定

```
SQL SQLMSS SET HINT OPTION (FAST 2)
TABLE FILE STXT31M
PRINT *
BY F01INT
END
```

上記の WebFOCUS リクエストは、ヒントが追加された SELECT ステートメントに変換されます。

```
SELECT
    T1."F01INT",
    T1."F02CHAR_10",
    T1."F03VARCHAR_10"
FROM
    D999AIXPPC71XX_TT31M T1
ORDER BY
    T1."F01INT"
OPTION (FAST 2);
```

SQL パススルーによる Microsoft SQL Server ストアドプロシジャの呼び出し

Microsoft SQL Server のストアドプロシジャは、SQL パススルーの使用をサポートします。これらのプロシジャは、CREATE PROCEDURE コマンドを使用して Microsoft SQL Server 内で開発する必要があります。

アダプタは、IN、OUT、INOUT パラメータを含むストアドプロシジャをサポートします。

ストアドプロシジャによって返される出力パラメータ値は結果セットとして利用可能になります。単一行の結果セットのこれらの値は、呼び出したストアドプロシジャによって他のすべての結果セットが返された後で、クライアントに転送されます。(利用可能な場合の) 出力パラメータ名は、結果セットのカラムタイトルになります。

クライアントに返されるのは、呼び出し文字列内で参照されている出力パラメータ (およびその戻り値) だけです。結果として、ユーザは表示する出力パラメータ値を完全に制御することができます。

サーバは、基となる DBMS の規則に従って記述されたストアドプロシジャの呼び出しをサポートします。ここでの例は SQL ベースであることに注意してください。DBMS の規則、言語、および他のプログラミングの例については、DBMS のマニュアルを参照してください。

ストアドプロシジャの呼び出し

```
SQL SQLMSS EX procname [parameter_specification1]
[,parameter_specification2]...
END
```

説明

SQLMSS

Microsoft SQL Server の ENGINE 接尾語です。

procname

ストアドプロシジャ名です。ネイティブデータベース構文のストアドプロシジャの完全 (または部分) 修飾名です。

SQL または SYS のいずれかの命名規則を使用して、複数構成名の解釈に使用される区切り文字を制御することができます。詳細は、命名規則の設定を参照してください。

parameter_specification

IN、OUT、INOUT がサポートされています。ストアプロシジャにより異なる適切なパラメータを指定します。

IN

リテラル (例、125、3.14、'abcde') です。入力文字に予約語を使用することも可能です。予約語は文字リテラルとは異なり、引用符で囲む必要はありません (例、NULL)。この値は必須です。

OUT

1 つの疑問符 (?) で表されます。このパラメータの有無により、出力結果をアプリケーションへ渡すかどうかを決定します。省略すると、この項目は (0 バイトの) 空白文字列になります。

INOUT

出力を表す 1 つの疑問符 (?) と入力を表すリテラルで構成されます。出力と入力はスラッシュ (/) で区切ります (例、?/125、?/3.14、?/'abcde')。出力値を (0 バイトの) 空白文字列にすることも可能です。

ストアプロシジャの呼び出し

この例では、ユーザが edaqa.test_proc01 ストアドプロシジャを呼び出してパラメータ 1、3、5、7 の入力値を指定することで、このストアプロシジャの戻り値、およびパラメータ 2、3 の出力値を取得します。

パラメータ 4 と 6 が省略されていることに注意してください。ストアプロシジャは、作成時に指定されたデフォルト値を使用します。

```
SQL  SQLMSS  EX  edaqa.test_proc01  125,?,?/3.14,, 'abc',, 'xyz'  
END
```

ストアプロシジャの例

このストアプロシジャは、OUT および INOUT パラメータを使用します。


```

CREATE PROCEDURE EDAQA.PROCP3 (    OUT chSQLSTATE_OUT    CHAR(5),
                                OUT intSQLCODE_OUT    INT,
                                INOUT l_name char(20),
                                INOUT f_name char(20))

    RESULT SETS 1
    LANGUAGE SQL
-----
-- SQL Stored Procedure
-----
P1:BEGIN
    -- Declare variable
    DECLARE SQLSTATE CHAR(5) DEFAULT '00000';
    DECLARE SQLCODE INT DEFAULT 0;
    -- Declare cursor
    DECLARE cursor1 CURSOR WITH RETURN FOR
        SELECT
            EDAQA.NF29005.SSN5 AS SSN5,
            EDAQA.NF29005.LAST_NAME5 AS LAST_NAME5,
            EDAQA.NF29005.FIRST_NAME5 AS FIRST_NAME5,
            EDAQA.NF29005.BIRTHDATE5 AS BIRTHDATE5,
            EDAQA.NF29005.SEX5 AS SEX5
        FROM
            EDAQA.NF29005
        WHERE
            (
                ( EDAQA.NF29005.LAST_NAME5 = l_name )
            AND
                ( EDAQA.NF29005.FIRST_NAME5 = f_name )
            );
    -- Cursor left open for client application
    OPEN cursor1;
    SET chSQLSTATE_OUT = SQLSTATE;
    SET intSQLCODE_OUT = SQLCODE;
    SET l_name = 'this is first name';
    SET f_name = 'this is last name';
END P1    @

```

ストアドプロシジャのアプリケーションエラーの収集

RAISERROR メソッドを使用して、アプリケーションエラーを収集することができます。ストアドプロシジャから発行されたアプリケーションエラーは、&MSSMSGTXT サーバ変数に

格納されます。

Microsoft SQL Server ODBC の互換性

バージョン 5.2 以降の Microsoft SQL Server アダプタは、OLE DB (Microsoft 社が推奨する高性能コンポーネント開発用 API) に基づいています。このアダプタは、SQL Server TDS プロトコルに直接アクセスするネイティブの高性能プロバイダである、OLE DB Provider for SQL Server (SQLOLEDB) を使用します。

バージョン 5.2 より前のアダプタは、ODBC API に基づいていました。現在のバージョンのアダプタは、従来の ODBC ベースのバージョンの機能をすべてサポートします。さらに、以下の機能が強化されています。

- 分散トランザクションのサポート
- 配列検索
- コマンド実行のタイムアウト
- 高速ロード - この機能は自動的に有効となり、ロード処理のみの ibi Data Migrator アプリケーションおよび MODIFY プロシジャのパフォーマンスを改善します。
- XA トランザクションのサポート - 詳細は、[XA のサポート](#)を参照してください。

OLE DB および ODBC では、次の領域でアダプタの機能が異なります。

- 接続属性
- カーソル
- UNIQUEIDENTIFIER および BIT データタイプのマッピング

ODBC を使用する場合は、これらのトピックを参照してください。

Microsoft SQL Server ODBC の接続属性

ODBC のデータソース宣言には、Microsoft SQL Server のインスタンス、デフォルトのデータベース、ネットワークライブラリなどの情報が含まれています。従来の ODBC ベースのアダプタでは、SET CONNECTION_ATTRIBUTES コマンドには ODBC データソース名 (ユーザまたはシステム DNS) および認証パラメータのみが必要でした。

ODBC データソースとは異なり、OLE DB データソースは Microsoft SQL Server のインスタンスを直接参照します。OLE DB データソースに接続するには、アダプタは関連するパラ

メータを接続時にすべて指定する必要があります。このため、SET CONNECTION_ATTRIBUTES コマンドの構文が拡張され、これらのパラメータを指定できるほか、Microsoft SQL Server の同じインスタンスに複数の接続を宣言できるようになりました。

Microsoft SQL Server ODBC のカーソル

従来の ODBC ベースのアダプタでは、SET パラメータの CONCUR を使用して ODBC カーソルタイプを制御していました。CONCUR パラメータには、RONLY と ROWVER の 2 つの設定があります。RONLY ではクライアントサイドのカーソルが使用され、ROWVER ではサーバサイドのカーソルが使用されます。バージョン 5.2 以降のアダプタでは、OLE DB の機能を利用して、CURSOR パラメータでカーソルタイプを直接制御することができます。SET パラメータの CURSOR には、CLIENT と SERVER の 2 つの設定があります。デフォルト設定では、アダプタは指定された環境で最高の性能を発揮するカーソルプロパティを割り当てます。

UNIQUEIDENTIFIER および BIT データタイプのマッピング

ODBC とは異なり、OLE DB は以下のデータタイプをマッピングします。

- UNIQUEIDENTIFIER は 38 文字にマッピングされます。ODBC では、このデータタイプは 36 文字にマッピングされます。この相違は、OLE DB が値を括弧 ({ }) で囲むことから生じます。
- BIT では、TRUE は -1、FALSE は 0 にマッピングされます。ODBC では、TRUE は 1 にマッピングされます。

Microsoft SQL Server ODBC アダプタの使用

Microsoft SQL Server ODBC アダプタの構成方法について説明します。

Microsoft SQL Server ODBC アダプタを使用すると、アプリケーションから Microsoft SQL Server データソースへのアクセスが可能になります。このアダプタにより、データまたはアプリケーションリクエストがネイティブの Microsoft SQL Server ステートメントに変換され、最適化されたアンサーセットがリクエストの送信元プログラムに返されます。

Microsoft SQL Server ODBC 環境の準備

Microsoft SQL Server 環境は、Microsoft SQL Server 2012 または 2014、および SQL Server 用の ODBC ドライバ 11 以降をインストールする際に設定されます。

その他の設定の手順は必要ありません。

Microsoft SQL Server のリモートアクセス

リモートノード上の Microsoft SQL Server にアクセスすることが可能です。Microsoft SQL Server にリモートアクセスする手順は、以下のとおりです。

- Microsoft SQL Server Native Client の最新バージョン、および SQL Server 用の ODBC ドライバ 11 以降をローカルマシンにインストールします。
- Microsoft SQL Server のリモートインスタンスの名前を確認します。

インストールした Microsoft SQL Server は、すべて一意の NetBEUI 名を使用して定義されます。有効なユーザ ID およびパスワードを定義している限り、Microsoft SQL Server のインスタンス名と同様、サーバはネットワーク上の任意の Microsoft SQL Server にアクセスすることができます。これらのパラメータは、サーバのグローバルプロファイルまたはユーザプロファイルに定義することができます。

Microsoft SQL Server ODBC アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

Microsoft SQL Server のデータベースサーバに接続するには、アダプタは接続および認証情報を必要とします。この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。

この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。選択したプロファイルにコマンドが追加されます。選択可能なプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) があります。
- グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) のいずれかに手動でコマンドを追加します。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の Microsoft SQL Server データベースへの接続を宣言することができます。Microsoft SQL Server への実際の接続は、接続を参照する最初のクエリを実行する際に行われます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。

4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。

[アダプタの構成を追加] ウィンドウが表示されます。

5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。

6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Microsoft SQL Server ODBC の接続属性

Microsoft SQL Server ODBC アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

サーバ

Microsoft SQL Server が稼働しているマシンの名前です。そのマシンに SQL Server の複数のインスタンスがインストールされている場合、「server¥instance」のようにサーバ名およびインスタンス名を指定します。

セキュリティ

Microsoft SQL Server への接続時にユーザを認証する方法には、次の 3 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報が Microsoft SQL Server データソースへの接続時に標準ログインの認証情報として渡されます。このオプションを使用するには、SQL サーバセキュリティとして SQL サーバおよび Windows に設定する必要があります。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID およびパスワードが、Microsoft SQL Server への接続時に標準ログインの認証情報として渡されます。このオプションを使用するには、SQL サーバセキュリティとして SQL サーバおよび Windows に設定する必要があります。
- **Trusted** アダプタは、サーバのデータアクセスエージェントによって偽装されたオペレーティングシステムユーザの認証情報を使用して、オペレーティングシステムログインとして Microsoft SQL Server に接続します。このオプションは、いずれの SQL サーバセキュリティの設定においても機能します。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

デフォルトデータベース

この接続に使用される Microsoft SQL Server データベースの名前です。パスを含むデータベース名は、一重引用符 (') で囲む必要があります。このパラメータはオプションとして指定します。指定しない場合、認証 ID に関連するデータベースがデフォルト値として設定されます。

追加の接続文字列キーワード (オプション)

ODBC 接続の動作変更に使われる接続文字列キーワードについての詳細は、Microsoft SQL Server のマニュアルを参照してください。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

Windows での接続属性の手動宣言

Explicit 認証 ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報が Microsoft SQL Server データソースへの接続時に標準ログインの認証情報として渡されます。

```
ENGINE MSODBC SET CONNECTION_ATTRIBUTES [connection]
server/userid,password [;dbname][;additional_connection_string_keywords]
```

Password Passthru 認証 クライアントアプリケーションから受け取ったユーザ ID およびパスワードが、Microsoft SQL Server への接続時に認証情報として渡されます。

```
ENGINE MSODBC SET CONNECTION_ATTRIBUTES [connection]
server/[;dbname][;additional_connection_string_keywords]
```

Trusted 認証 アダプタは、サーバのデータアクセスエージェントによって偽装された Windows ユーザの認証情報を使用して、Windows ログインとして Microsoft SQL Server に接続します。


```
ENGINE MSODBC SET CONNECTION_ATTRIBUTES [connection]server/, [;dbname]  
[;additional_connection_string_keywords]
```

注意：特殊文字を含む値は、一重引用符 (') で囲みます。値に一重引用符 (') が含まれている場合は、その一重引用符の前に別の一重引用符を追加する必要があります。結果的に、一重引用符が 2 つ連続することになります。たとえば、ブランクと一重引用符 (') の両方を含む「Mary O'Brien」をユーザ ID として指定する場合は、「'Mary O' 'Brian'」と入力します。

デフォルト接続の上書き変更

複数の接続を定義した場合、最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続がデフォルト接続として機能します。このデフォルト値は、SET DEFAULT_CONNECTION コマンドを使用して上書きすることができます。

デフォルト接続の変更

```
ENGINE MSODBC SET DEFAULT_CONNECTION connection
```

説明

MSODBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。

- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

```
ENGINE MSODBC SET DEFAULT_CONNECTION SAMPLE
```

接続範囲の制御

確立する接続ごとにアダプタを使用する場合、SET AUTODISCONNECT コマンドを使用して接続の継続を制御することができます。

接続範囲の制御

```
ENGINE MSODBC SET AUTODISCONNECT ON {FIN|COMMAND|COMMIT}
```

説明

MSODBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMAND

各リクエストの後に自動的に切断されます。SET AUTODISCONNECT コマンドは、イベントがどの程度の頻度で発生するかによって、大幅なオーバーヘッドをもたらす可能性があります。このオーバーヘッドのほとんどすべてがサーバには関連しません。このオーバーヘッドは、オペレーティングシステムとデータソースに関連します。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

Microsoft SQL Server ODBC メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および Microsoft SQL Server ODBC データタイプのサーバマッピングを定義するシノニムを作成します。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な Microsoft SQL Server ODBC テーブルまたはビューの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

このアダプタでは、MS SQL ネイティブシノニムの特定のタイプでのメタデータの作成がサポートされます。この機能は、Microsoft SQL Server バージョン 2012 以降でのみ使用することができます。

ネイティブシノニムのメタデータは、次の条件下で作成することができます。

- ネイティブシノニムは、TABLE または VIEW ベースオブジェクトから作成されている必要があります。
- ベースオブジェクト (TABLE または VIEW) は、アダプタ接続文字列がターゲットとする MS SQL Server 上に存在している必要があります。
- ネイティブシノニムは、最大で 3 部構成の名前で記述されたベースオブジェクトで作成されている必要があります。ターゲットサーバ内でのベースオブジェクトのパスおよびオーナーシップに応じて、受容可能なフォーマットは次のとおりです。

```
object_name, schema_name.object_name,
```

または

```
database_name.schema_name.object_name.
```

次のタイプのネイティブシノニムはサポートされません。

- ストアドプロシジャに基づいているもの。
- リンクされた MS SQL Server 上のオブジェクトに基づいているもの。
- 次のような 4 部構成の名前で記述されたオブジェクトに基づいているもの。

```
server_name.database_name.schema_name.object_name
```

ストアドプロシジャのシノニムの作成、およびストアドプロシジャに対するレポートの実行についての詳細は、[ストアドプロシジャのシノニムの作成](#)を参照してください。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。
選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオブ

ションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Microsoft SQL Server ODBC のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

重要：オブジェクトタイプにストアドプロシジャを選択した場合、入力パラメータはここで説明するものとは多少異なります。詳細は、[ストアドプロシジャに対するレポートの実行](#)を参照してください。

データベース

テーブルやオブジェクトの選択元となるデータベースを指定するには、次のいずれかを実行します。

- [デフォルトデータベース] を選択して、デフォルトのデータベースとして設定されているデータベースを使用します。
- [データベース] ドロップダウンリストからデータベースを選択します。ここには、現在の DBMS インスタンスにあるすべてのデータベースが表示されます。

[デフォルトデータベース] が選択されている場合は、データベースを選択する前に、このチェックをオフにします。

目的のデータベースを指定するには、[データベース] ドロップダウンメニューからデータベースを選択します。ドロップダウンメニューには、Microsoft SQL Server のターゲットインスタンスのすべてのデータベースがリストされます。デフォルトデータベースを選択すると、接続の構成時に設定されたデータベースが保持されます。構成時にデフォルトデータベースが設定されなかった場合、SQL Server 上でアクティブなログインに割り当てられているデータベースがデフォルトとして使用されます。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

外部キーを使用したクラスタ作成 (非推奨。[シノニムの編集] の使用を推奨)

[外部キーを使用したクラスタ作成] のチェックをオンにすると、現在のテーブルに外部キーで関係付けられているテーブルすべてをこのシノニムに含めることができます。ただし、このオプションは廃止される予定で、クラスタの作成にはシノニムエディタを使用することをお勧めします。生成されるマルチテーブルのシノニムは、このテーブルの外部キーの関係をすべて定義します。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (:) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[Microsoft SQL Server ODBC データタイプのサポート](#)を参照してください。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

テーブル名

基になるオブジェクトの名前です。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

生成されたシノニムの例

Microsoft SQL Server ODBC アダプタのシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

生成されたマスターファイル nf29004.mas

```
FILE=DIVISION, SUFFIX=MSODBC , $  
SEGNAME=SEG1_4, SEGTYPE=S0 , $  
FIELD=DIVISION4, DIVISION4, I9, I4, MISSING=OFF , $  
FIELD=DIVISION_NA4, DIVISION_NA4, A25, A25, MISSING=ON , $  
FIELD=DIVISION_HE4, DIVISION_HE4, I9, I4, MISSING=ON , $
```

生成されたアクセスファイル nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=edaqa.nf29004,  
CONNECTION=connmss, KEYS=1, WRITE=YES, $
```

Microsoft SQL ODBC テーブルコメントのシノニムへのマッピング

Microsoft SQL テーブルまたはビューのシノニムを作成する際に、アダプタは次のようにコメントをマッピングします。

- MS SQL Server テーブルまたはビューのコメント (存在する場合) は、マスターファイルシノニムの REMARK 属性にマッピングされます。
- MS SQL Server カラムのコメント (存在する場合) は、マスターファイルシノニムの DESCRIPTION 属性にマッピングされます。

Unicode と 非 Unicode 両方のコメントがサポートされます。

また、MS SQL Server カラムのタイトル (存在する場合) は、マスターファイルシノニムの TITLE 属性にマッピングされます。

アクセスファイルのキーワード

下表は、アクセスファイルで使用されるキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABlename	Microsoft SQL Server のテーブルを指定します。以下のように完全修飾のテーブル名を指定することができます。 <pre>TABlename=[[database.]owner.]table</pre>
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre> CONNECTION=' ' は、ローカルのデータベースサーバへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。この属性を使用するには、キーを構成するフィールドをマスターファイルの先頭に記述する必要があります。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。

キーワード	説明
	KEY=fld1/fld2/.../fldn
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。
KEYFLD IXFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定します。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 - KEYFLD は、親テーブルの共通カラムの FIELDNAME です。 - IXFLD は関連するテーブルの共通カラムの FIELDNAME です。 KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることをお勧めしますが、必須ではありません。 注意：KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。
AUTO INCREMENT	[Yes] に設定すると自動インクリメント機能が有効になります。
START	インクリメントの順序の初期値です。
INCREMENT	インクリメントの間隔です。

キーワード	説明
INDEX_NAME INDEX_UNIQUE INDEX_COLUMNS INDEX_ORDER	データベース内のインデックス名、およびインデックスフィールドの一意性、名前、順序を示します。

常時暗号化のサポート

SQL Server アダプタの Windows ODBC バージョンでは、次の条件下で SQL Server 2017/2016 常時暗号化ネイティブ機能がサポートされます。

- シノニムを作成する前に、SQL Server Management Studio を使用してフィールドを暗号化する必要があります。
- 接続文字列キーワード ColumnEncryption=Enabled を使用してアダプタ接続を構成します。

シノニムを作成すると、アクセスファイル内の暗号化されたフィールドに ENCRYPT_TYPE={DETERMINISTIC|RANDOMIZED} 属性が追加されます。

暗号化されたフィールドに対する一部の処理が SQL で実行できません。その原因はアダプタロジックにあります。これらの処理は WebFOCUS Reporting Server で実行されます。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

Microsoft SQL Server ODBC データタイプのサポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[データタイプレポートを表示するには](#)を参照してください。

SQL 式内での末尾ブランク

TABLE アダプタ内の新しい SQL ジェネレータでは、デフォルトで文字列リテラル内の末尾ブランク、および数値リテラル内の小数部と指数の表記を含むリテラルコンテンツを保持します。これにより、生成される SQL に対する制御が向上します。

末尾ブランクが不要なケースでは、次の構文が下位互換性を確保するために使用できます。

```
ENGINE MSODBC SET TRIM_LITERALS ON
```

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアードプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

読み取り専用フィールドのサポート

CREATE SYNONYM コマンドは、TIMESTAMP として作成された Microsoft SQL Server のカラムまたは IDENTITY 属性が定義されたカラムに FIELDTYPE=R のフィールド定義を作成します。これらのフィールドは読み取り専用です。MAINTAIN または MODIFY プロシジャを実行する場合、アダプタはマスターファイルに FIELDTYPE=R が定義されたカラムに対して書き込み処理を実行しません。

読み取り専用フィールドのサポート

以下は、最初のカラムのプロパティが IDENTITY で、2 番目のカラムがタイムスタンプカラムのテーブルを作成する例です。

```
CREATE TABLE TAB1
(idproptab int IDENTITY (1,1), timestmp timestamp)
```

CREATE SYNONYM コマンドによって、このテーブルに以下のマスターファイルが生成されます。

```
FILE=TAB1, SUFFIX=MSODBC , $
SEGNAME=TAB1, SEGTYPE=S0 , $
FIELD=IDPROPTAB, idproptab, I11, I4, MISSING=OFF, FIELDTYPE=R , $
FIELD=TIMSTMP, timestmp, A16, A16, MISSING=ON, FIELDTYPE=R , $
```

Microsoft SQL Server ODBC のストアド プロシジャに対するレポートの実行

SELECT ステートメントや TABLE コマンドなどのレポートツールを使用し、Microsoft SQL Server ODBC ストアドプロシジャを実行して、プロシジャの出力パラメータおよびアンサーセットに対するレポートを作成することができます。ストアドプロシジャの実行という方法には、次の利点があります。

- アンサーセット以外に、出力パラメータである OUT パラメータおよび OUT モードの INOUT パラメータを検索することができる。他の呼び出し方法では、アンサーセットのみが検索されます。

- TABLE および他のレポートツールを使用して、出力パラメータおよびアンサーセットの処理、フォーマット設定、表示を簡単に行うことができる。

ストアドプロシジャに対するレポートを実行するには、次の手順を実行します。

1. **シノニムの作成** ストアドプロシジャのアンサーセットのシノニムを作成します。詳細は、[ストアドプロシジャのシノニムの作成](#)を参照してください。
2. **レポートプロシジャの作成** レポートプロシジャを作成します。詳細は、[ストアドプロシジャに対するレポートの実行](#)を参照してください。
3. **レポートの実行** ストアドプロシジャを実行し、出力パラメータ (OUT および OUT モードの INOUT) およびレポートに指定されたアンサーセットフィールドに対するレポートを実行します。

ストアドプロシジャのシノニムの作成

シノニムは、ストアドプロシジャのパラメータとアンサーセットを記述します。

アンサーセットの構造は、プロシジャの実行時に指定する入力パラメータ値により異なります。このため、プロシジャの実行時に指定される入力パラメータ値のセットごとに、個別のシノニムを作成する必要があります。たとえば、3つの異なる入力パラメータ値セットでストアドプロシジャを実行する場合、シノニムを3つ (値セットごとに1つ) 作成する必要があります。なお、特に明記していない限り、「入力パラメータ」は、IN パラメータおよび IN モードの INOUT パラメータを意味します。

例外 ユーザがプロシジャの内部ロジックを理解しており、入力パラメータの値の範囲とプロシジャによって返された各アンサーセットの構造の関係を把握している場合、各アンサーセットの構造に対応する単一のシノニムを作成して、各シノニムにアンサーセットの構造を返すために必要な入力パラメータの値のセットを指定することが可能です。

シノニムには、次のセグメントを含めます。

- **INPUT** - このセグメントは、IN パラメータおよび IN モードの INOUT パラメータを記述します。
IN パラメータおよび IN モードの INOUT パラメータがない場合、このセグメントはダミーフィールドを1つ記述します。
- **OUTPUT** - このセグメントは、OUT パラメータおよび OUT モードの INOUT パラメータを記述します。
OUT パラメータおよび OUT モードの INOUT パラメータがない場合、このセグメントは省略されます。

- ANSWERSET*n* - 各アンサーセットに 1 つ含まれます。
アンサーセットがない場合、このセグメントは省略されます。

Microsoft SQL Server ODBC ストアドプロシ ジャ CustOrders のシノニム

次のシノニムは、Microsoft SQL Server ODBC のストアドプロシジャを記述します。入力パラメータ、出力パラメータ、アンサーセットがそれぞれ 1 つあり、アンサーセットには 4 つの変数が含まれています。

マスターファイルシノニムは、以下のとおりです。

```
FILENAME=CUSTORDERS, SUFFIX=MSODBC , $
  SEGMENT=INPUT, SEGTYPE=S0, $
    FIELDNAME=@CUSTOMERID, ALIAS=P00001, USAGE=A5, ACTUAL=A5,
    MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
  SEGMENT=OUTPUT, SEGTYPE=S0, PARENT=INPUT, $
    FIELDNAME=@RETURN_VALUE, ALIAS=P00000, USAGE=I11, ACTUAL=I4, $
  SEGMENT=ANSWERSET1, SEGTYPE=S0, PARENT=INPUT, $
    FIELDNAME=ORDERID, ALIAS=OrderID, USAGE=I11, ACTUAL=I4, $
    FIELDNAME=ORDERDATE, ALIAS=OrderDate, USAGE=HYMDs, ACTUAL=HYMDs,
    MISSING=ON, $
    FIELDNAME=REQUIREDDATE, ALIAS=RequiredDate, USAGE=HYMDs,
    ACTUAL=HYMDs, MISSING=ON, $
    FIELDNAME=SHIPPEDDATE, ALIAS=ShippedDate, USAGE=HYMDs,
    ACTUAL=HYMDs, MISSING=ON, $
```

アクセスファイルシノニムは、以下のとおりです。

```
SEGNAME=INPUT, CONNECTION=ITarget, STPNAME=Northwind.dbo.CustOrders, $
SEGNAME=OUTPUT, STPRESORDER=0, $
SEGNAME=ANSWERSET1, STPRESORDER=1, $
```

ストアドプロシジャのシノニム作成パラ メータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

[ストアードプロシジャ] を選択します。

データベース

テーブルやオブジェクトの選択元となるデータベースを指定するには、次のいずれかを実行します。

- [デフォルトデータベース] を選択して、デフォルトのデータベースとして設定されているデータベースを使用します。
- [データベース] ドロップダウンリストからデータベースを選択します。ここには、現在の DBMS インスタンスにあるすべてのデータベースが表示されます。

[デフォルトデータベース] が選択されている場合は、データベースを選択する前に、このチェックをオフにします。

選択

プロシジャを選択します。次のシノニム作成ウィンドウの [値] テキストボックスで、プロシジャごとに固有の値を入力する必要があるため、1 回に選択できるプロシジャは、1 つだけです。

名前

シノニム名です。デフォルト設定は、ストアードプロシジャ名です。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のストアードプロシジャが存在する場合は、接頭語または接尾語を割り当てて、対応するシノニムを区別します。なお、シノニム名の最大文字数は 64 バイトです。

すべてのプロシジャに固有の名前が付いている場合は、[接頭語] および [接尾語] のテキストボックスは空白にしておきます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[Microsoft SQL Server ODBC データタイプのサポート](#)を参照してください。

値

指定したプロシジャに表示されるパラメータすべてについて、このチェックボックスを選択します。

選択したプロシジャに入力パラメータ (IN パラメータ、IN モードの INOUT パラメータ) がある場合、それらのパラメータの入力が要求されます。ただし、[値] テキストボックスに明示的に入力する必要があるかどうかは、プロシジャのロジックおよび生成されるデータ構造に依存します。このため、パラメータボックスのチェックはオンにする必要はありませんが、値の入力は不要な場合もあります。次のガイドラインに従います。

- 入力パラメータ値によりアンサーセットのデータ構造が変化する場合、明示的な入力値 (によりシノニムを区別すること) が必要です。
- プロシジャの内部ロジックが分かっており、常に同一のデータ構造が生成されることが確定している場合は、明示的な入力値は必要ありません。この場合、生成する必要のあるシノニムは 1 つだけで、シノニム作成の目的では、[値] テキストボックスはブランクにすることができます。

[値] が必要な場合は、引用符を含めずに値を入力します。日付、日付時間、タイムスタンプのパラメータ値は、ISO の形式に従って入力します。明示的な値が必要なプロシジャの場合、入力パラメータにはプロシジャが実行される際に入力されるパラメータと同一の値を指定します。

ストアドプロシジャに対するレポートの実行

ストアドプロシジャのアンサーセットに対してレポートを実行する場合は、データベーステーブルに対してレポートを実行する際と同一の機能を使用することができます。

- **SQL SELECT ステートメント** 構文についての詳細は、[SELECT を使用したストアドプロシジャに対するレポートの実行](#)を参照してください。
- **TABLE および GRAPH コマンド** 構文についての詳細は、[TABLE コマンドを使用した](#)

[ストアドプロシジャに対するレポートの実行](#)を参照してください。

ストアドプロシジャのアンサーセットとの JOIN には、次の 2 つのファイルを使用します。

- **ホストファイル** OUTPUT および ANSWERSET セグメントのみです。
- **クロスリファレンス** ファイル内の INPUT セグメントのみです。

TABLE コマンドを使用したストアドプロシジャに対するレポートの実行

TABLE コマンドを使用してストアドプロシジャを実行するには、次の構文を使用します。

```
TABLE FILE synonym
PRINT [parameter [parameter] ... | *]
[IF in-parameter EQ value]
.
.
.
END
```

説明

synonym

実行するストアドプロシジャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアドプロシジャにパラメータを必要としない場合は、アスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

★

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

IF

IF または WHERE キーワードです。IN パラメータまたは IN モードの INOUT パラメータに値を渡す際に使用します。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

注意：アダプタが Unicode をサポートするよう構成されている場合は、IN パラメータの長さは 1000 バイト以内にする必要があります。

value

パラメータに渡す値です。

SELECT を使用したストアドプロシジャに対するレポートの実行

```
SQL
SELECT [parameter [,parameter] ... | *] FROM synonym
[WHERE in-parameter = value]
.
.
.
END
```

説明**synonym**

実行するストアドプロシジャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアドプロシジャにパラメータを必要としない場合は、構文にアスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

★

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

WHERE

IN パラメータまたは IN モードの INOUT パラメータに値を指定する場合に使用します。

各パラメータの値は、それぞれ異なる行に指定する必要があります。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

value

パラメータに渡す値です。

Microsoft SQL Server ODBC アダプタの設定

Microsoft SQL Server ODBC アダプタでは、[設定の変更] ダイアログボックスから、さまざまな設定を行うことができます。

このダイアログボックスを開くには、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで、Microsoft SQL Server ODBC アダプタを右クリックし、[設定の変更] をクリックします。Microsoft SQL Server ODBC アダプタには、次の設定があります。

バルクロードサービス

[バルクロードサービス] には、次のオプションがあります。

BULKLOAD

データの DBMS へのロードに使用する手法です。[ON] に設定すると、拡張バルクロード機能が有効になります。[OFF] に設定すると、標準 SQL INSERT ステートメントが有効になります。デフォルト値は [OFF] です。

BULKCHECK

データベースに対してトランザクションをコミット (書き込み) する行間隔です。デフォルト値は 1000000 または DBMS 固有のいずれかです。

注意：このオプションは、[BULKLOAD] パラメータが [ON] に設定されている場合にのみ適用されます。

BLK_MAX_SESSIONS

データベースにログオンする際のロードセッションの最大数です。デフォルト値は 10 です。

注意：このオプションは、[BULKLOAD] パラメータが [ON] に設定されている場合にのみ適用されます。

BLK_MAX_RESTARTS

修復可能なロードセッションエラー後に許容するロードセッション再起動の最大試行回数です。デフォルト値は 10 です。

注意：このオプションは、[BULKLOAD] パラメータが [ON] に設定されている場合にのみ適用されます。

BLK_DFIXDELIM

データベースにロードされる中間 Flat File 内のフィールド間の区切り文字です。区切り文字は、DBMS に応じて最大 4 文字で構成できます。

次のオプションがあります。

- **TAB** タブ文字を示します。
- **a** ~ などの文字列を示します。
- **0xNN** 16 進コードを示します。たとえば、カンマ (,) の 16 進コードは 0x44 です。16 進数コードは、Windows および UNIX システムでは ASCII を使用します。16 進数コードは、Windows および UNIX システムでは ASCII を使用します。

デフォルト値は [TAB] です。

BLK_ROWDELIM

データベースにロードされる中間 Flat File 内のレコード間の区切り文字です。行区切り文字は、フィールドの区切り文字と同様に指定することができますが、カンマ (,) を使用することはできません。

次のオプションがあります。

- **TAB** タブ文字を示します。
- **a** ~ などの文字列を示します。
- **0xNN** 16 進コードを示します。たとえば、カンマ (,) の 16 進コードは 0x44 です。16 進数コードは、Windows および UNIX システムでは ASCII を使用します。16 進数コードは、Windows および UNIX システムでは ASCII を使用します。

この設定にはデフォルト値はありません。

DIRECT_BULK_LOAD

データの DBMS へのロードの手法を制御します。[ON] に設定すると、DBMS ユーティリティによって、データが直接ロードされます。[OFF] に設定すると、ソースデータが中間ファイルに抽出されます。デフォルト値は、アダプタによって異なります。

BLK_SAVE_DATA_FILES

プロセスが完了するまで、中間のバルクロードデータファイルを保存することができます。たとえば、Salesforce アダプタの場合、REST API ソースデータファイルを APPDIR/TargetMFD_JOBID_nnnnnnnnnnnn 内のフォルダに保存できます。ここで、nnnnnnnnnnnn はリクエストに割り当てられたジョブ ID です。[YES] に設定すると、BLK_SAVE_REPONSE_LOG が暗黙的に YES に設定されます。デフォルト値は [NO] です。

BLK_PERM_AS_STAGE

バルクロード処理中に、永続データベーステーブルを中間テーブルとして使用する権限を制御します。デフォルト値は [ON] です。

データタイプマッピングのカスタマイズ

[データタイプマッピングのカスタマイズ] には、次のオプションがあります。

CONV_INT_PREC

整数精度列を制御します。デフォルト値は 11 です。

CONV_FLOAT_PREC

倍精度浮動小数点数精度列のマッピングを制御します。デフォルト値は 20 です。

CONV_FLOAT_SCALE

倍精度浮動小数点数小数部列のマッピングを制御します。デフォルト値は 2 です。

CONV_DEC_PREC

パック 10 進数精度列のマッピングを制御します。このデータタイプにデフォルト値はありません。

CONV_DEC_SCALE

倍精度浮動小数点数小数部列のマッピングを制御します。このデータタイプにデフォルト値はありません。

この設定は、非推奨の SET CONVERSION PRECISION p s コマンドに優先します。

CONV_LONGCHAR

LONGCHAR 列のマッピングを制御します。LONGCHAR のマッピングオプションは、[ALPHA]、[TEXT]、または [BLOB] です。デフォルト値は [ALPHA] です。

CONV_LONGCHAR_LEN

この設定は、[CONV_LONGCHAR] の値を [ALPHA] に設定した場合にのみ使用できます。デフォルト値は 32767 です。

EXACT_DATETIME

日付時間列のマッピングを制御します。この設定は、互換性のために用意されています。[ON] に設定すると、日付時間の正確な精度がシノニムフィールドのフォーマットとして保持されます。たとえば、TIMESTAMP(5) は HYYMDs ではなく HYYMD5 にマッピングされるため、小数点以下 5 桁がすべて保持されます。デフォルト値は [OFF] です。

機能診断

[機能診断] には、次のオプションがあります。

PASSRECS

[ON] に設定すると、「(FOC1364) パススルーコマンドで影響を受けた行」というメッセージが表示されます。デフォルト値は [ON] です。このメッセージの表示を停止するに

は、PASSRECS 設定の値を [OFF] に設定します。

LOG_SWITCHED

ログファイル切り替えの動作を制御します。オプションには [STOP] と [PROCEED] があります。デフォルト値は [STOP] です。

SPLIT_MERGE_STATISTICS

INSERT、UPDATE、DELETE などの行に対して実行された動作に基づいて分割される MERGE ステートメント統計を制御します。デフォルト値は [OFF] です。

メタデータ

[メタデータ] には、次のオプションがあります。

ALIAS_CASE

新しいターゲットの ALIAS の大文字と小文字を設定することができます。

次のオプションがあります。

- **変換しない** 新しいターゲットの ALIAS のデフォルトの大文字と小文字は変更されません。
- **小文字** 新しいターゲットの ALIAS を小文字に変更します。
- **大文字** 新しいターゲットの ALIAS を大文字に変更します。

デフォルト値は [変換しない] です。

TABLE_OPTIONS

CREATE FILE コマンドによって生成された CREATE TABLE ステートメントにテーブルオプションパラメータを追加します。この設定にはデフォルト値はありません。

NCHAR

Unicode 以外のサーバ構成で、バイトセマンティクスを使用することにより、Unicode RDBMS フィールドのマスターフィールドフォーマットの拡張を制御します。この設定は、Unicode サーバ構成では必要ありません。

次のオプションがあります。

- **1 バイト文字** 1 文字につき 1 バイトを使用します。たとえば、NCHAR(10) は A10

に対応します。

- **2 バイト文字** 1 文字につき 2 バイトを使用します。たとえば、NCHAR(10) は A20 に対応します。
- **3 バイト文字** 1 文字につき 3 バイトを使用します。たとえば、NCHAR(10) は A30 に対応します。

デフォルト値は、インストールによって異なります。

その他の設定

[その他の設定] には、次のオプションがあります。

JOINTYPE

自動パススルーが実行不可または無効になっている場合 (SQL SET APT=OFF)、FOCUS 生成 SQL 変換での JOIN タイプを制御します。

次のオプションがあります。

- **SORTMERGE** 各テーブルからデータを取得し、ソート、結合、集計します。SORTMERGE に設定すると、FOCUS 生成では常にソート/マージを使用してデータを取得します。
- **NESTEDLOOP** 外部テーブルから、選別条件に一致するすべての行を選択し、それらの行を使用して内部テーブルから修飾データを取得します。FOCUS 生成では、SQL ステートメント内の JOIN 可能なフィールドに equijoin が存在する場合にのみ、この方法が使用されます。このような equijoin が存在しない場合、FOCUS 生成はソート/マージに戻されます。

この設定にはデフォルト値はありません。

読み取り/書き込みの最適化

[読み取り/書き込みの最適化] には、次のオプションがあります。

FETCHSIZE

1 回の処理で DBMS から取得する行数を制御します。デフォルト値は 100 です。

INNERSIZE

1 回の処理で DBMS に渡される行数を制御します。デフォルト値は 1 です。

DBMS セッションパラメータ

[DBMS セッションパラメータ] には、次のオプションがあります。

AUTODISCONNECT

DBMS との接続を切断するタイミングを制御します。

次のオプションがあります。

- **ON-FIN** セッションの終了後に、接続されているすべての DBMS サーバから切断します。
- **ON COMMAND** 最も外側のコマンドブラケットの末尾で、SQL アダプタで接続済みのすべての DBMS サーバとの接続を解除します。
- **ON COMMIT (ROLLBACK)** COMMIT または ROLLBACK 時に、アダプタの接続を解除します。

デフォルト値は [ON-FIN] です。

LOGINTIMEOUT

アダプタがデータソースへのログオンを完了してからエラーを返すまでの時間を指定します。この時間は秒単位で指定します。デフォルト値は、アダプタによって異なります。

COMMANDTIMEOUT

エラーが返されるまでのコマンドのタイムアウトを制御します。値 0 (ゼロ) は、タイムアウトが無期限であることを示します。この値のデフォルト設定はブランクで、設定が有効ではないことを示します。

TRANSACTIONS

トランザクションモードを設定します。[ON] に設定すると、処理を実行する接続ごとに、アダプタがローカルトランザクションを暗黙的に開始します。アダプタは COMMIT または ROLLBACK の実行時、あるいはサーバセッションの終了時に、各接続の処理を連続的にコミットするか、処理を中止します。[OFF] に設定すると、成功した場合は DBMS での個々の処理が直ちにコミットされ、エラーの場合は DBMS によってロールバックされます。このモードは、読み取り専用アプリケーションでパフォーマンスを考慮する場合に使用することをお勧めします。このモードを読み取り/書き込みアプリケーションに使

用することはお勧めしません。このモードでは、複数の処理で構成された論理作業単位をロールバックすることができないためです。デフォルト値は [ON] です。

ISOLATION

トランザクション分離レベルを指定し、データの整合性を向上させることができます。

次のオプションがあります。

- **インストールのデフォルト** DBMS インストールに固有のデフォルト設定のトランザクション分離レベルを指定します。
- **カーソル固定** 読み取りまたは変更されたデータベース情報はロックされます。変更されたオブジェクトは、トランザクションが完了するまでロックされます。一方、データベース情報を読み取るために取得したロックは、読み取り直後に解除されます。
- **非コミット読み取り** データベースへの変更はロックされ、トランザクションが完了するまで保持されます。単なるデータベースからの読み取りでは、ロックされません。データがロックされると、他のユーザはこのデータにアクセスできなくなります。
- **反復可能読み取り** 読み取りまたは変更されたデータベース情報はロックされます。変更されたオブジェクトおよび読み取られたオブジェクトは、トランザクションが完了するまでロックされます。変更されていないアクセス構造 (例、インデックス) に対するロックは、読み取り直後に解除されます。
- **読み取り固定** 読み取りまたは変更された情報は、トランザクションが完了するまでロックされます。変更または照会されたすべてのアクセス構造も、トランザクションが完了するまでロックされます。
- **スナップショット** 他のトランザクションによってデータベースに加えられた変更は、スナップショットトランザクション内で読み取られるデータには反映されません。単なるデータベースからの読み取りでは、ロックされません。したがって、他のユーザは、このトランザクション中にデータベースにデータを書き込むことができます。他のユーザによるデータベースの更新を防止することはできません。

デフォルト値は [インストールのデフォルト] です。

トランザクション分離レベルについての詳細は、Microsoft のマニュアルを参照してください。

SQL の生成

[SQL の生成] には、次のオプションがあります。

OPTIMIZATION

リクエストの最適化を制御します。

次のオプションがあります。

- **最適化 (乗法の影響が検出されない場合)** JOIN および複雑な WHERE 述語が最適化され、集計は SQL に渡されます。ただし、一部のコンストラクトを SQL に変換できない場合、および乗算の影響が検出される場合は除きます。
- **最適化 (乗法の影響を無視)** JOIN および複雑な WHERE 述語が最適化され、集計は SQL に渡されます。ただし、一部のコンストラクトを SQL に変換できない場合は除きます。この場合、乗算の影響による JOIN および集計の無効化は発生しません。
- **JOIN と複合記述子の最適化をオフ** JOIN および複雑な WHERE 述語の最適化を無効にします。
- **OUTER JOIN の最適化をオフ** OUTER JOIN の最適化のみを無効にします。
- **PRINT オブジェクトの演算を DBMS で処理する** PRINT で使用される定義済みフィールドの計算を DBMS に委任するようアダプタに指示します。
- **AGGREGATION をオフ** 集計は無効にしますが、JOIN および複雑な WHERE 述語の最適化は無効化しません。

デフォルト値は [最適化 (乗法の影響が検出されない場合)] です。

Microsoft SQL Server ODBC 環境のカスタマイズ

Microsoft SQL Server ODBC アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。ここでは、カスタマイズオプションの概要について説明します。

タイムアウト制限の指定

Microsoft SQL Server に対して SQL リクエストを実行した後、アダプタがその応答を待機する時間 (秒) を指定するには、TIMEOUT コマンドを使用します。

TIMEOUT コマンドの発行

```
ENGINE MSODBC SET TIMEOUT {nn|0}
```

説明

MSODBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nn

タイムアウトとなる秒数を指定します。デフォルト値は 30 です。

0

応答の待機時間が無制限であることを示します。

長時間実行中リクエストのキャンセル

WebFOCUS Reporting Server ブラウザインターフェースで、長時間実行中のリクエストをキャンセルすることができます。この操作を実行すると、ネイティブ JDBC ドライバの機能に応じて、リクエスト全体がキャンセルされるか、フェッチサイクルが中断されます。

長時間実行中のリクエストをキャンセルするには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [ワークスペース] ページで、[Java サービス] 下の [DEFAULT] を選択します。[DEFAULT] を右クリックし、[エージェント] を選択します。
2. [Java サービスエージェント] ウィンドウで、キャンセルする jscomid の行を選択し、右クリックして [終了] を選択します。

ログイン待機時間の指定

LOGINTIMEOUT コマンドを使用して、アダプタが接続時に Microsoft SQL Server からの応答を待機する時間を秒単位で指定することができます。

注意：アダプタの従来のリリースとの互換性を維持するため、TIMEOUT を LOGINTIMEOUT のシノニムとして使用することが可能です。

ログイン待機時間の指定

```
ENGINE MSODBC SET LOGINTIMEOUT|TIMEOUT {nn|0}
```

説明

MSODBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nn

タイムアウトとなる秒数を指定します。デフォルト値は約 15 秒です。

0

ログイン応答の待機時間が無制限であることを示します。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

使用可能なパラメータは次のとおりです。

```
ENGINE MSODBC SET PASSRECS {ON|OFF}
```

説明

MSODBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

トランザクションの分離レベルの指定

WebFOCUS Reporting Server ブラウザインターフェースから、または SET ISOLATION コマンドを使用して、トランザクションの分離レベルを指定することができます。

SET コマンドによるトランザクション分離レベルの指定

以下のコマンドを実行して、トランザクションの分離レベルを指定することができます。

```
ENGINE MSODBC SET ISOLATION {RU|RC|RR|SE|CH|CS|SS}
```

説明

RU

トランザクションの分離レベルを非コミット読み取りに設定します。

RC

トランザクションの分離レベルをカーソル固定に設定します。

RR

トランザクションの分離レベルを反復可能読み取りに設定します。

SE

トランザクションの分離レベルを読み取り固定に設定します。

CH

トランザクションの分離レベルを Chaos に設定します。

CS

トランザクションの分離レベルをカーソル固定に設定します (カーソル固定は読み取り固定のシノニムです)。

SS

トランザクションの分離レベルを非コミット読み取りに設定します。

Microsoft SQL Server ODBC 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)および[定数として定義された一時項目 \(DEFINE\) を転送するリクエストの最適化](#)を参照してください。

一時項目 (DEFINE) に NULL 値が含まれる 場合のリクエストの最適化

SET OPTNOAGGR コマンドを使用して、アダプタの最適化に関する動作を微調整することができます。このコマンドは、何らかの理由で RDBMS に集計処理を渡したくない場合に使用します。その理由の一例として、集計データに NULL 値が含まれる場合などがあります。SET OPTNOAGGR コマンドを使用すると、アダプタは DBMS に集計処理を渡さずに SQL を生成します。結合およびソート処理は RDBMS で実行されますが、集計処理はサーバ側で内部的に実行されます。

DEFINE フィールドに NULL フィールドを使用する計算が含まれている場合は常に NULL が返されるため、これらの処理を SQL に変換して DBMS に渡すことはできません。そのような処理は FOCUS で処理する必要があります。

これには SET OPTIMIZATION OFF を設定します。

ただし、パフォーマンスを考慮した場合、AGGREGATION は FOCUS で実行して、DBMS に対してはオフロードの JOIN および SORT 処理を使用すると良いケースもあります。

強化された集計制御の設定

```
SQL MSODBC SET OPT {AGGR|NOAGGR}
```

説明

AGGR

アダプタは DEFINE フィールドの集計処理を RDBMS にオフロードします。この設定がデフォルト値です。

NOAGGR

アダプタは SQL を生成しますが、集計処理を RDBMS に渡しません。結合およびソート処理は RDBMS で実行されますが、集計処理はサーバ側で内部的に実行されます。この設定は、RDBMS にオフロードされる SQL が少なかった旧バージョンの機能に基づいて記述されたアプリケーションの下位互換性を提供する目的で使用することもできます。たとえば、集計フィールドの計算に NULL データが含まれている場合、その NULL データは RDBMS の NVL() 関数では処理されませんでした。

常に FALSE になる条件を含む IF-THEN-ELSE 最適化の使用

```
SQL MSODBC SET OPTIFTHENELSE ON
DEFINE FILE EMPINFO
DEF3 = IF FIRST_NAME EQ 'RITA' THEN 1 ELSE 0;
END

TABLE FILE EMPINFO
PRINT FIRST_NAME
IF DEF3 EQ 2
END
```

条件 DEF3 EQ 2 が True になることはないため、アダプタは RDBMS に WHERE 条件 1=0 (常に False) を渡します。RDBMS からレコードは返されません。

```
SELECT T1."FN" FROM USER1."EMPINFO" T1 WHERE (1 = 0) FOR FETCH ONLY;
```

DEFINE 式の最適化での SQL 制限

FOCUS レポート言語は固有の SQL より広範囲に及ぶため、データアダプタは特有の DEFINE 式を RDBMS に渡すことができません。DEFINE 式に以下のものが含まれている場合、データアダプタは DEFINE ベースの集計およびレコード選択を RDBMS にオフロードしません。

- ユーザ作成サブルーチン。
- 以下のような自己参照式。

```
X=X+1;
```

- 数値から文字フィールドへの変換、または文字から数値フィールドへの変換を実行する EDIT 関数。
- フィールドの値を変換する DECODE 関数。
- 関係演算子の INCLUDES および EXCLUDES。
- FOCUS サブルーチンの ABS、INT、MAX、MIN、LOG、SQRT。

注意： FOCUS サブルーチンの MAX および MIN と、演算接頭語の MAX. および MIN. を混同しないように注意してください。DEFINE フィールドに演算接頭語を含めることはできません。

- ACTUAL=DATE が指定されたフィールドを含む式。ただし、DATE フィールドから別の DATE フィールドを減算する式、および DATE フィールドのすべての論理式は除きます。
- FOCUS の日付時間関数により処理された日付時間は、SQL には変換されません。
- Financial Modeling Language (FML) のセル計算。FML は、拡張マトリクスレポート ティング (EMR) と呼ばれます。

注意： FML レポートリクエストは拡張 TABLE リクエストです。FML は、明細レポートの特殊な機能を提供します。

また IF-THEN-ELSE の最適化では、以下の機能はサポートされません。

- すべてのタイプの DECODE 式
- 静的 SQL
- IF/WHERE DDNAME
- 部分的な日付の選択

検索処理でのブロックサイズの指定

Microsoft SQL Server ODBC アダプタは、SELECT クエリまたはストアドプロシージャの実行によって生成された、結果セットからの配列検索をサポートします。この手法により、ネットワークおよび CPU 使用の負荷を大幅に軽減することができます。

大きい値を指定すると仮想記憶域のコストは増えますが、大量の行を処理するリクエストの効率は改善されます。ただし、100 を超える値を指定しても僅かな効率改善しか期待できないため、100 以内の値を設定することをお勧めします。

ヒント： この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

配列検索のブロックサイズの指定

SELECT リクエストのブロックサイズは、TABLE FILE リクエスト、MODIFY リクエスト、MATCH リクエスト、および DIRECT SQL SELECT ステートメントに適用されます。

```
ENGINE MSODBC SET FETCHSIZE n
```

説明

MSODBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

配列検索手法を使用して一度に取得する行数です。受容値は 1 から 5000 です。アダプタに応じてデフォルト値は異なります。CLOB または BLOB として処理する必要のあるフィールドが結果セットに含まれている場合は、その結果セットに使用する FETCHSIZE 値は 1 です。

挿入処理のブロックサイズの指定

INSERT のブロックサイズは、LOADONLY と組み合わせて MODIFY INCLUDE リクエストに適用されます。INSERTSIZE は、パラメータ化 DIRECT SQL INSERT ステートメントでもサポートされます。

```
ENGINE MSODBC SET INSERTSIZE n
```

説明

MSODBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

配列検索手法を使用して挿入する行数です。受容値は 1 から 5000 です。BLOB として処理する必要のあるフィールドが結果セットに含まれている場合は、その結果セットに使用

する INSERTSIZE 値は 1 です。

バルクインサート API の省略

```
ENGINE MSODBC SET FASTLOAD [ON|OFF]
```

説明

MSODBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

バルクインサート API を使用します。デフォルト値は ON です。

OFF

バルクインサート API の使用を抑制します。

バルクインサート API の動作

ibi Data Migrator で Microsoft SQL Server ODBC のバルクインサート API を使用することができます。

Microsoft SQL Server ODBC アダプタでは、LOADONLY モードでバルク API が自動的に使用されます。その値は、中間的なフラッシュがパフォーマンスに影響しないことを示します。このためその動作は INSERTSIZE に依存しません。

ロード中に発生するエラー (例、重複エラー) により、行のバッチ全体が拒否される場合があります。

非等価条件 WHERE ベース LEFT OUTER JOIN の最適化

LEFT OUTER JOIN は、ホストテーブルからすべてのレコードを選択し、それらをクロスリファレンステーブルのレコードと照合します。一致するレコードが存在しない場合、ホストレコードは保持されますが、クロスリファレンスフィールドにはデフォルト値 (ブランクまたはゼロ) が割り当てられます。アダプタは、RDBMS で条件式がサポートされるすべての WHERE ベースの LEFT OUTER JOIN コマンドを最適化することができます。

条件付き LEFT OUTER JOIN の指定

```
JOIN LEFT_OUTER FILE hostfile AT hfld1 [TAG tag1]
    [WITH hfld2]
    TO {UNIQUE|MULTIPLE}
    FILE crfile AT crfld [TAG tag2] [AS joinname]
    [WHERE expression1;
    [WHERE expression2;
    ...]

END
```

説明

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。

hostfile

ホストマスタファイルです。

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 パラメータにフィールド値を指定しても、この値が接続情報として使用されることはありません。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

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に一意の修飾子として使用するタグ名 (オプション) です。

WITH 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 に基づく条件付き JOIN では、KEEPDEFINES を ON に設定し、一時項目 (DEFINE) を JOIN コマンドを発行する前に作成しておく必要があります。

MULTIPLE

from_file と *to_file* 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。なお、ONE は UNIQUE の同義語です。

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。

ユニーク JOIN は WebFOCUS の概念です。RDBMS はユニークと非ユニークを区別しないため、クロスリファレンスファイルから条件に一致するすべての行を取得します。

たとえば、リクエストでユニーク JOIN として指定したが、実際には条件に一致する行がクロスリファレンスファイルに複数存在する場合、この JOIN の処理を RDBMS が行くと、RDBMS は一致するすべての行を返します。一方、最適化を無効にしてこの JOIN の処理を WebFOCUS が行くと、異なるレポート結果が得られます。WebFOCUS はユニーク JOIN の概念を順守するため、ホストの 1 行に対してクロスリファレンスの行を 1 行だけ返します。

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データソース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。

WHERE ベース OUTER JOIN の最適化条件

- WHERE ベースの LEFT OUTER JOIN を最適化するには、それに関係する RDBMS で式の最適化を行うことができるとともに、次の条件のいずれかを満たす必要があります。
 - JOIN WHERE コマンドに、少なくとも 1 つの「field1 EQ field2」述語が含まれている。ここで、field1 は table1 のフィールド、field2 は table2 のフィールドを表します。
または
 - 右側のテーブルに、NULL データを含まないユニークインデックスまたはキーがある。
または
 - 右側のテーブルに、長いデータタイプ (例、TEXT、IMAGE) を持たない「NOT NULL」フィールドが少なくとも 1 つ含まれている。
- アダプタの SQLJOIN OUTER 設定は、ON (デフォルト) に設定しておく必要があります。

非等価条件 LEFT OUTER JOIN の最適化

次のリクエストは、2 つの MSSQL データソース間の条件付き LEFT OUTER JOIN を作成し、結合後のデータソースを使用してレポートを作成します。このリクエストに対して生成された SQL を表示するため、STMTRACE がオンに設定されています。

```
SET TRACEUSER = ON
SET TRACEOFF = ALL
SET TRACEON = STMTRACE//CLIENT
JOIN LEFT_OUTER FILE employee AT EMPLOYEEID
TO ALL FILE employeepayhistory AT EMPLOYEEID
    WHERE RATECHANGEDATE GT HIREDATE;
END
TABLE FILE employee
PRINT RATE
BY EMPLOYEEID
END
```

WebFOCUS リクエストは、LEFT OUTER JOIN を組み込んだ単一の MSSQL SELECT ステートメントに変換され、非等価条件が ON 句で RDBMS に渡されます。

```
SELECT T1."EmployeeID", T1."HireDate", T2."EmployeeID",
T2."RateChangeDate", T2."Rate" FROM
AdventureWorks.HumanResources.Employee T1 LEFT OUTER JOIN
AdventureWorks.HumanResources.EmployeePayHistory T2 ON
(T2."RateChangeDate" > T1."HireDate") ) ORDER BY T1."EmployeeID";
```

ヒントによるオプティマイザの効率化

DBMS オプティマイザのヒントを使用して、実行プランを変更することができます。アダプタには、TABLE コマンドの実行時に生成される Microsoft SQL Server 用クエリの末尾にヒントを追加する設定があります。

ヒントは、アダプタが単一の SELECT ステートメントを作成する際に追加されます。FOCUS 管理の JOIN で複数の SELECT ステートメントが生成される場合、ヒントは追加されません。

この設定を元に戻すには、hint_text パラメータを省略して SET HINT を使用します。

ヒントの設定

特定のヒントを設定するには、次の構文を使用します。

```
SQL MSODBC SET HINT OPTION (hint_text)
```

説明

MSODBC

ターゲットの RDBMS です。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

OPTION (hint_text)

単一ヒントのテキスト、または複数のヒントを組み合わせたテキストです。ユーザは、構文どおりに正確に入力する必要があります。OPTION (*hint_text*) を省略すると、ヒントを使用しない設定に戻ります。

Microsoft SQL ヒントの設定

```
SQL MSODBC SET HINT OPTION (FAST 2)
TABLE FILE STXT31M
PRINT *
BY F01INT
END
```

上記の WebFOCUS リクエストは、ヒントが追加された SELECT ステートメントに変換されます。

```
SELECT
  T1."F01INT",
  T1."F02CHAR_10",
  T1."F03VARCHAR_10"
FROM
  D999AIXPPC71XX_TT31M T1
ORDER BY
  T1."F01INT"
OPTION (FAST 2);
```

SQL パススルーによる Microsoft SQL Server ODBC ストアドプロシージャの呼び出し

Microsoft SQL Server のストアドプロシージャは、SQL パススルーの使用をサポートします。これらのプロシージャは、CREATE PROCEDURE コマンドを使用して Microsoft SQL Server 内で開発する必要があります。

アダプタは、IN、OUT、INOUT パラメータを含むストアドプロシージャをサポートします。

ストアドプロシージャによって返される出力パラメータ値は結果セットとして利用可能になります。単一行の結果セットのこれらの値は、呼び出したストアドプロシージャによって他のすべての結果セットが返された後で、クライアントに転送されます。(利用可能な場合の) 出力パラメータ名は、結果セットのカラムタイトルになります。

クライアントに返されるのは、呼び出し文字列内で参照されている出力パラメータ (およびその戻り値) だけです。結果として、ユーザは表示する出力パラメータ値を完全に制御することができます。

サーバは、基となる DBMS の規則に従って記述されたストアドプロシージャの呼び出しをサポートします。ここでの例は SQL ベースであることに注意してください。DBMS の規則、言語、および他のプログラミングの例については、DBMS のマニュアルを参照してください。

ストアドプロシージャの呼び出し

```
SQL MSODBC EX procname [parameter_specification1]
[,parameter_specification2]...
END
```

説明

MSODBC

Microsoft SQL Server の ENGINE 接尾語です。

procname

ストアードプロシージャ名です。ネイティブデータベース構文のストアードプロシージャの完全 (または部分) 修飾名です。

SQL または SYS のいずれかの命名規則を使用して、複数構成名の解釈に使用される区切り文字を制御することができます。詳細は、命名規則の設定を参照してください。

parameter_specification

IN、OUT、INOUT がサポートされています。ストアードプロシージャにより異なる適切なパラメータを指定します。

IN

リテラル (例、125、3.14、'abcde') です。入力文字に予約語を使用することも可能です。予約語は文字リテラルとは異なり、引用符で囲む必要はありません (例、NULL)。この値は必須です。

OUT

1 つの疑問符 (?) で表されます。このパラメータの有無により、出力結果をアプリケーションへ渡すかどうかを決定します。省略すると、この項目は (0 バイトの) 空白文字列になります。

INOUT

出力を表す 1 つの疑問符 (?) と入力を表すリテラルで構成されます。出力と入力はスラッシュ (/) で区切ります (例、?/125、?/3.14、?/'abcde')。出力値を (0 バイトの) 空白文字列にすることも可能です。

ストアードプロシージャの呼び出し

この例では、ユーザが edaqa.test_proc01 ストアードプロシージャを呼び出してパラメータ 1、3、5、7 の入力値を指定することで、このストアードプロシージャの戻り値、およびパラメータ 2、3 の出力値を取得します。

パラメータ 4 と 6 が省略されていることに注意してください。ストアードプロシージャは、作成時に指定されたデフォルト値を使用します。

```
SQL MSODBC EX edaqa.test_proc01 125,?,?,?/3.14,,'abc',,'xyz'
END
```

ストアドプロシジャの例

このストアドプロシジャは、OUT および INOUT パラメータを使用します。

```
CREATE PROCEDURE EDAQA.PROCP3 (    OUT chSQLSTATE_OUT    CHAR(5),
                                OUT intSQLCODE_OUT    INT,
                                INOUT l_name char(20),
                                INOUT f_name char(20))

    RESULT SETS 1
    LANGUAGE SQL

-----
-- SQL Stored Procedure
-----

P1:BEGIN
    -- Declare variable
    DECLARE SQLSTATE CHAR(5) DEFAULT '00000';
    DECLARE SQLCODE INT DEFAULT 0;
    -- Declare cursor
    DECLARE cursor1 CURSOR WITH RETURN FOR
        SELECT
            EDAQA.NF29005.SSN5 AS SSN5,
            EDAQA.NF29005.LAST_NAME5 AS LAST_NAME5,
            EDAQA.NF29005.FIRST_NAME5 AS FIRST_NAME5,
            EDAQA.NF29005.BIRTHDATE5 AS BIRTHDATE5,
            EDAQA.NF29005.SEX5 AS SEX5
        FROM
            EDAQA.NF29005
        WHERE
            (
                ( EDAQA.NF29005.LAST_NAME5 = l_name )
            AND
                ( EDAQA.NF29005.FIRST_NAME5 = f_name )
            );
    -- Cursor left open for client application
    OPEN cursor1;
    SET chSQLSTATE_OUT = SQLSTATE;
    SET intSQLCODE_OUT = SQLCODE;
    SET l_name = 'this is first name';
    SET f_name = 'this is last name';
END P1    @
```

ストアドプロシジャのアプリケーションエラーの収集

RAISERROR メソッドを使用して、アプリケーションエラーを収集することができます。ストアドプロシジャで生じるアプリケーションエラーは、サーバ変数 &MSOMSGTXT で使用することができます。

MySQL アダプタの使用

MySQL アダプタの構成方法について説明します。

MySQL アダプタを使用すると、アプリケーションから MySQL データソースへのアクセスが可能になります。このアダプタにより、アプリケーションリクエストがネイティブの MySQL ステートメントに変換され、最適化されたアンサーセットがリクエストの送信元アプリケーションに返されます。

注意：バージョン 9.3.0 以降、MySQL アダプタは UPSERT サポートを提供します。

MySQL 環境の準備

MySQL アダプタを使用するには、MySQL Connector/J JDBC ドライバが必要です。リリース 5.x ではバージョン 5.0.4 またはそれ以降が必要です。リリース 3.x ではバージョン 3.1.14 またはそれ以降が必要です。

MySQL で Unicode データからレポートを作成するには、WebFOCUS Reporting Server で Unicode サポートを有効にする必要があります。詳細は、[MySQL と Unicode](#) を参照してください。

Windows で MySQL 環境を準備するには

手順

1. CLASSPATH 環境変数に MySQL コネクタ/JDBC ドライバファイルのパスを指定します。jar ファイルのフルパスと名前を指定する必要があります。たとえば、jar ファイルの格納先が C:\Program Files\MySQL\JDBC Driver の場合は、次のように指定します。

```
CLASSPATH= C:\Program Files\MySQL\JDBC Driver\mysql-connector-java-5.1.17-bin.jar
```

2. JAVA_HOME または JDK_HOME 環境変数に、Java Runtime Environment または

Development Kit のパスを指定します。いずれかのパスを指定することができます。
Java Runtime Environment がインストールされているディレクトリのパスを指定する必要があります。そのディレクトリ下に bin サブディレクトリがあります。たとえば、Java Runtime Environment が C:\Program Files\java\jre6 にインストールされている場合は、次のように指定します。

```
JAVA_HOME=C:\Program Files\java\jre6
```

別の方法として、Java Development Kit が C:\Program Files\java\jdk1.6.0_17 にインストールされている場合は、次のように指定します。

```
JDK_HOME= C:\Program Files\java\jdk1.6.0_17
```

3. サーバを起動または再起動します。

結果

注意：ドライバを変更した場合にも、サーバを再起動する必要があります。

UNIX で MySQL 環境を準備するには

複数のファイルのロケーションを指定します。

手順

1. \$CLASSPATH 環境変数に MySQL コネクタ/JDBC ドライバファイルのパスを指定します。

たとえば、ファイルの格納先が /usr/driver_files の場合、次のステートメントを発行します。

```
CLASSPATH=/usr/driver_files/mydriver.jar;$CLASSPATH
export CLASSPATH
```

サーバを起動する前に変数を設定するには、UNIX プロファイルに CLASSPATH 設定を追加します。

別の方法として、WebFOCUS Reporting Server ブラウザインターフェースでパスを指定することもできます。

- a. [ツール] メニューの [ワークスペース] をクリックします。
 - b. [Java サービス] フォルダを展開し、[DEFAULT] を右クリックしてから [プロパティ] を選択します。
[Java サービスの構成] ウィンドウが開きます。
 - c. [クラスパス] セクションを展開します。
 - d. [IBI_CLASSPATH] テキストボックスで、MySQL Connector/J jar ファイルのフルパスを指定します。
 - e. [Java サービスを保存して再起動] をクリックします。
2. \$JDK_HOME 環境変数に、Java Development Kit のインストールディレクトリのパスを指定します。

たとえば、Java Development Kit のロケーションを /usr/java に設定するには、以下のステートメントを実行します。

```
JDK_HOME=/usr/java
export JDK_HOME
```

3. \$LD_LIBRARY_PATH 環境変数に Java 仮想マシンのインストールディレクトリのロケーションを指定します。

たとえば、JVM のロケーションを /usr/j2sdk1.4.2_01/jre/lib/i386/server に設定するには、以下のステートメントを実行します。

```
LD_LIBRARY_PATH=/usr/j2sdk1.4.2_01/jre/lib/i386/server
export LD_LIBRARY_PATH
```

セキュリティを ON に設定してサーバを実行している場合、LD_LIBRARY_PATH 変数は無視されます。この場合は、IBI_LIBPATH を使用する必要があります。

4. サーバを起動または再起動します。

結果



注意：ドライバを変更した場合にも、サーバを再起動する必要があります。

MySQL と Unicode

MySQL アダプタは、JDBC を使用して実装されます。この実装により、CHARACTER SET を UTF-8 に設定した文字フィールドへの Unicode データの格納が可能になります。

サーバを開始する前に、edastart ファイルまたは別のシェルフファイルで LANG および NLS_LANG 環境変数を設定しておく必要があります。たとえば、英語 (米国) の場合は、次の変数をエクスポートします。

```
export LANG=EN_US.UTF-8
```

詳細は、『[ibi™ WebFOCUS® サーバ管理者ガイド](#)』の「Unicode サポート」を参照してください。

MySQL アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

SET CONNECTION_ATTRIBUTES コマンドを使用して、単一の MySQL データベースサーバへの接続を定義して、サーバへの接続に必要な認証属性を指定することができます。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の MySQL データベースサーバへの接続を宣言することができます。実際の接続は、その接続を参照する最初のクエリ発行時に確立されます。詳細は、[デフォルト接続の上書き変更](#)を参照してください。SET CONNECTION_ATTRIBUTES コマンドは、RPC またはサーバプロファイルに含めることができます。プロファイルは暗号化することができます。

複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドに指定した MySQL データベースサーバが、デフォルトのデータベースサーバとして設定されます。
- 複数の SET CONNECTION_ATTRIBUTES コマンドで宣言された MySQL データベースサーバが同一の場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された認証情報が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。

4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。

[アダプタの構成を追加] ウィンドウが表示されます。

5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。

6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

MySQL の接続属性

MySQL アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

URL

MySQL データソースのロケーション URL を入力します。基本構文は、以下のとおりです。

```
jdbc:mysql://host/database
```

説明

host

MySQL データベースが存在するコンピュータ名または IP アドレスです。

database

データベースの名前です。

これらの例は以下のように 2 つあります。

```
jdbc:mysql://localhost/qatst
```

```
jdbc:mysql://edaaix52/qatst
```

MySQL リリース 4.1 から、別の MySQL 接続プロパティを URL で参照できます。この機能を使用するには、次のガイドラインに従ってください。以下の例に示すように、URL では最初のプロパティの前に文字「?」を指定し?URL で参照される 2 番目および後続のプロパティの前には文字「&」と「|」を続けて指定する必要があります。

追加する接続プロパティは次のものであるとします。

```
sessionVariables=sql_mode=PIPES_AS_CONCAT  
zeroDateTimeBehavior=convertToNull
```

URL を次のように入力します。

```
jdbc:mysql://host/database?sessionVariables=sql_mode=P  
IPES_AS_CONCAT&|  
zeroDateTimeBehavior=convertToNull
```

このアダプタを SSL で使用するには、verifyServerCertificate、requireSSL、useSSL のプロパティを追加する必要があります。次のように、useSSL プロパティは最後に追加します。

```
jdbc:mysql://host:port/server?verifyServerCertificate=false&|  
requireSSL=true&|useSSL=true
```

注意：URL は単一の行に入力し、文字「|」の後にスペースを入れないでください。

ドライバ名

MySQL JDBC ドライバの名前です。

Connector/J 8.0 以降では、次のドライバを使用します。

```
com.mysql.cj.jdbc.Driver
```

Connector/J 8.0 より前のバージョンでは、次のドライバを使用します。

```
com.mysql.jdbc.Driver
```

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、`CONNECTION_ATTRIBUTES` コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの `edasprof.prf` です。

新しいユーザプロファイル (`user.prf`) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (`edasprof`) に保存します。

接続属性の手動宣言

Explicit 認証の場合

```
ENGINE MYSQL SET CONNECTION_ATTRIBUTES [connection]/userid,password
```

Password Passthru 認証の場合

```
ENGINE MYSQL SET CONNECTION_ATTRIBUTES connection/
```

説明

MYSQL

アダプタを指定します。SET `SQLENGINE` コマンドをすでに発行している場合は、この値を省略することができます。

connection

接続名です。任意の名前を指定することができます。

userid

MySQL に登録されたプライマリ認証 ID です。

password

プライマリ認証 ID に関連するパスワードです。

接続属性の宣言

以下の SET CONNECTION_ATTRIBUTES コマンドは、明示的なユーザ ID およびパスワードを使用して、MySQL データベースサーバ TEST に接続します。

```
ENGINE MYSQL SET CONNECTION_ATTRIBUTES TEST/USERA,PWDA
```

以下の SET CONNECTION_ATTRIBUTES コマンドは、Password Passthru 認証を使用して MySQL データベースサーバ TEST に接続します。

```
ENGINE MYSQL SET CONNECTION_ATTRIBUTES TEST/
```

ユーザの認証

MySQL データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードは SET CONNECTION_ATTRIBUTES コマンドで明示的に指定します。これらのコマンドは、すべてのユーザのサーバグローバルプロファイル edasprof.prf に含めることができます。
- **データベースまたは Password Passthru** クライアントアプリケーションから受け取ったユーザ ID およびパスワードが、認証情報として MySQL データベースサーバに渡されます。

クライアントがサーバに接続する際に、ユーザ ID およびパスワードが認証用として MySQL に渡されますが、サーバ側では認証は実行されません。このタイプの認証を行うには、セキュリティを OFF にしてサーバを起動します。サーバはクライアント接続を許可して、クライアントの接続メッセージが暗号化された形式で保存されます。サーバセッション中は、MySQL データベースサーバへの接続に暗号化メッセージを使用することができます。

デフォルト接続の上書き変更

SET CONNECTION_ATTRIBUTES コマンドを使用してアクセスする MySQL 接続をすべて宣言した後、以下の 2 つの方法に従って、宣言されている接続リストから特定の MySQL 接続を選択することができます。

- SET DEFAULT_CONNECTION コマンドを使用して、デフォルトの接続を選択することができます。このコマンドを実行しない場合、最初の SET CONNECTION_ATTRIBUTES コマンドに指定された接続名の値が使用されます。
- 現在の SQL クエリに指定されたテーブルのアクセスファイルに CONNECTION= 属性を含めることができます。WebFOCUS Reporting Server ブラウザインターフェースから CREATE SYNONYM コマンドに接続名を指定した場合、アクセスファイルに CONNECTION= 属性が自動的に含まれます。この属性はデフォルトの接続に優先します。

アクセスする接続の選択

```
ENGINE MYSQL SET DEFAULT_CONNECTION [connection]
```

説明

MYSQL

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続名です。省略した場合、ローカルデータベースサーバがデフォルト値として設定されます。この接続名を定義していない場合、FOC1671 メッセージが出力されます。

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名が有効となります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合、FOC1671 メッセージが出力されます。

アクセスする接続の選択

以下の SET DEFAULT_CONNECTION コマンドを発行すると、デフォルトの MySQL データベースサーバとして TNSNAMEB が選択されます。

```
ENGINE MYSQL SET DEFAULT_CONNECTION datasource_name
```

注意：これ以前に、datasource_name に対して SET CONNECTION_ATTRIBUTES コマンドを実行していなければなりません。

接続範囲の制御

ここでは、アダプタを使用して論理的な処理単位の範囲を設定する方法について説明します。これには SET AUTODISCONNECT コマンドを使用します。

宣言したデータベースサーバとの最初の通信時に接続が確立されます。

接続範囲の制御

```
ENGINE MYSQL SET AUTODISCONNECT ON {FIN|COMMAND|COMMIT}
```

説明

MYSQL

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMAND

各リクエストの後に自動的に切断されます。SET AUTODISCONNECT コマンドは、イベントがどの程度の頻度で発生するかによって、大幅なオーバーヘッドをもたらす可能性があります。このオーバーヘッドのほとんどすべてがサーバには関連しません。このオーバーヘッドは、オペレーティングシステムとデータソースに関連します。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

MySQL メタデータの管理

ここでは、MySQL データソースに対する CREATE SYNONYM の使用方法について説明します。また、MySQL データタイプのサポートについても説明します。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な MySQL テーブルまたはビューの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは一時項目 (DEFINE) や他のセキュリティ機構などの、サーバの拡張メタデータ機能をサポートします。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。

- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
 - **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
 - **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。
3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。
 4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

MySQL のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作

成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

オブジェクト名

オブジェクト名のフィルタに使用する文字列を入力します。必要に応じて、文字列の先頭や末尾にワイルドカード文字 (%) を入力します。たとえば、「ABC%」と入力して、名前が ABC で始まるオブジェクトをすべて選択します。「%ABC」と入力して、名前が ABC で終わるオブジェクトをすべて選択します。「%ABC%」と入力して、名前の先頭、中間、または末尾に ABC が含まれるオブジェクトをすべて選択します。

外部 SQL スクリプトのパス

[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合は、さらに次の項目が表示されます。

次の標準名前規則は UNIX に適用されます。

- [ベースパス] テキストボックスに、SQL クエリが記述されたファイルの物理ディレクトリパスを指定します。ディレクトリ名を直接入力するか、[...] (参照) ボタンをクリックしてファイルの格納先ディレクトリへ移動します。このクリックにより、[ベースパスの選択] ダイアログボックスが表示されます。
- [ドキュメント名] テキストボックスに、ファイル名を入力します。必要に応じて、ワイルドカード文字を使用します。
- [ドキュメント拡張子] テキストボックスに、スクリプトファイルの拡張子を入力して、その拡張子のファイルのみを表示します。

アダプタはシノニムの作成中にネイティブ API コールを発行して選択リスト内の要素のリストを取得し、各要素に対するフィールドを持つマスターファイルを構成します。生成されたアクセスファイルは、DATASET 属性の SQL スクリプトパスを参照します。これには SQL クエリが格納されたファイルの名前と拡張子を含む、フルパスが記述されています。以下はその例です。

```
DATASET=/u1/home2/apps/report3.sql
```

WebFOCUS レポートを作成する際は、SQL クエリを使用してデータにアクセスします。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (:) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[MySQL データタイプのサポート](#)を参照してください。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

テーブル名

基になるオブジェクトの名前です。

タイプ

オブジェクトタイプ (テーブル、ビューなど)。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

生成されたシノニムの例

MySQL アダプタのシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

生成されたマスターファイル nf29004.mas

```
FILE=DIVISION, SUFFIX=MYSQL , $
SEGNAME=SEG1_4, SEGTYPE=S0 , $
FIELD=DIVISION4, DIVISION4, I9, I4, MISSING=OFF , $
FIELD=DIVISION_NA4, DIVISION_NA4, A25, A25, MISSING=ON , $
FIELD=DIVISION_HE4, DIVISION_HE4, I9, I4, MISSING=ON , $
```

生成されたアクセスファイル nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=EDAQA.NF29004,
CONNECTION=TEST, KEYS=1, WRITE=YES, $
```

アクセスファイルのキーワード

下表は、アクセスファイルで使用するキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABLENAME	MySQL のテーブル名を指定します。テーブル名に所有者 (スキーマ) 名を含めることができます。 以下はその例です。 <pre>TABLENAME=[owner.]tablename</pre>
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。

キーワード	説明
	<pre>CONNECTION=connection</pre> CONNECTION=' ' は、ローカルのデータベースサーバへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。この属性を使用するには、キーを構成するフィールドをマスターファイルの先頭に記述する必要があります。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。 <pre>KEY=fld1/fld2/.../fldn</pre>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。
KEYFLD IXFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定します。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 - KEYFLD は、親テーブルの共通カラムの FIELDNAME です。 - IXFLD は関連するテーブルの共通カラムの FIELDNAME です。

キーワード	説明
	KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることをお勧めしますが、必須ではありません。 注意：KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。
AUTO INCREMENT	自動インクリメントを有効にするには [Yes] に設定します。
START	インクリメントの順序の初期値です。
INCREMENT	インクリメントの間隔です。
INDEX_NAME INDEX_UNIQUE INDEX_ COLUMNS INDEX_ORDER	データベース内のインデックス名、およびインデックスフィールドの一意性、名前、順序を示します。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

MySQL データタイプのサポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[アクセスファイル](#)を参照してください。

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアードプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

MySQL アダプタ環境のカスタマイズ

ここでは、MySQL アダプタ環境を設定する手順について説明します。

PASSRECS

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取

得] をクリックした後、構成済みアダプタを右クリックし、メニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

PASSRECS の設定

```
ENGINE MYSQL SET PASSRECS {ON|OFF}
```

説明

MYSQL

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

トランザクションの分離レベルの指定

WebFOCUS Reporting Server ブラウザインターフェースから、または SET ISOLATION コマンドを使用して、トランザクションの分離レベルを指定することができます。

SET コマンドによるトランザクション分離レベルの指定

以下のコマンドを実行して、トランザクションの分離レベルを指定することができます。

```
ENGINE MYSQL SET ISOLATION {RU|RC|RR|SE}
```

説明

RU

トランザクションの分離レベルを非コミット読み取りに設定します。

RC

トランザクションの分離レベルをカーソル固定に設定します。

RR

トランザクションの分離レベルを反復可能読み取りに設定します。

SE

トランザクションの分離レベルを読み取り固定に設定します。

長時間実行中リクエストのキャンセル

WebFOCUS Reporting Server ブラウザインターフェースで、長時間実行中のリクエストをキャンセルすることができます。この操作を実行すると、ネイティブ JDBC ドライバの機能に応じて、リクエスト全体がキャンセルされるか、フェッチサイクルが中断されます。

長時間実行中のリクエストをキャンセルするには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [ワークスペース] ページで、[Java サービス] 下の [DEFAULT] を選択します。[DEFAULT] を右クリックし、[エージェント] を選択します。
2. [Java サービスエージェント] ウィンドウで、キャンセルする jscomid の行を選択し、右クリックして [終了] を選択します。

MySQL 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)および[定数として定義された一時項目 \(DEFINE\) を転送するリクエストの最適化](#)を参照してください。

Netezza アダプタの使用

Netezza アダプタの構成方法について説明します。

Netezza アダプタを使用すると、アプリケーションから Netezza データソースへのアクセスが可能になります。このアダプタにより、アプリケーションリクエストが Netezza コールに変換され、最適化されたアンサーセットがリクエストの送信元アプリケーションに返されます。

実装方法は JDBC のみです。

注意：バージョン 9.3.0 以降、Netezza アダプタは UPSERT サポートを提供します。

Netezza 環境の準備

Netezza アダプタを使用するには、Netezza ドライバをインストールし、サーバの起動前にその CLASSPATH 値を設定するとともに、JSCOM3 サービスを実行状態にしておく必要があります。

Windows および UNIX で環境を設定するには

手順

1. サーバを起動する前に、Netezza ドライバファイルのパスを CLASSPATH 環境変数に追加することで、これらのファイルの格納先を設定します。

注意：Db2 と Netezza が同一マシンにインストールされている場合は、Netezza の jar ファイルを最初に CLASSPATH に追加する必要があります。

たとえば、UNIX で Netezza ドライバファイルのパスを設定するには、次のように指定します。

```
CLASSPATH=/rdbms/netezza_jar/netezza-jdbc.jar:
/rdbms/netezza_jar/bcprov-jdk14-122.jar:$CLASSPATH
export CLASSPATH
```

Windows では、以下のコマンドを実行します。

```
set CLASSPATH=c:\usr\driver_files\<netezza_jdbc_driver_
files>;%CLASSPATH%
```

特定のファイル名およびその格納先についての詳細は、Netezza のマニュアルを参照してください。

サーバのワークスペースを Windows サービスとして実行する場合、CLASSPATH をシステム全体の環境変数として設定する必要があります。

同様に UNIX では、サーバを起動する前に変数を設定して、UNIX プロファイルに CLASSPATH の設定を追加します。

2. サーバを起動または再起動します。

ドライバを変更した場合にもサーバを再起動する必要があります。

3. サーバのワークスペースで以下のコマンドを実行することにより、JSCOM3 サービスが有効であることを確認します。

```
edastart -show
```

また、[スペシャルサービス] セクションに「JSCOM3 active」と表示されていることを確認します。

JSCOM3 サービスが有効ではない場合、サーバの EDAPRINT ログに「fail to start」というメッセージが出力されます。

Netezza の Unicode サポート

NCHAR/NVARCHAR フィールドの Netezza Unicode データは、WebFOCUS Reporting Server が Unicode (コードページ 65001) を使用するよう構成されている場合にサポートされます。

Netezza アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

アダプタは、Netezza データソースに接続する際に、接続情報および認証情報を必要とします。

この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。選択したプロファイルにコマンドが追加されます。選択可能なプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) があります。
- グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) のいずれかに手動でコマンドを追加します。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の Netezza データソースへの接続を宣言することができます。実際の接続は、最初のクエリ発行時に確立されます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Netezza JDBC の接続属性

Netezza アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

URL

Netezza データソースのロケーション URL です。

ドライバ名

JDBC ドライバの名前です。

詳細は、使用しているバージョンのドライバのマニュアルを参照してください。

IBI_CLASSPATH

Java サービスで使用する追加の Java クラスディレクトリ、または jar ファイル名のフルパスを定義します。値を設定するには、通信ファイルを編集するか、WebFOCUS Reporting Server ブラウザインターフェースを使用します。WebFOCUS Reporting Server ブラウザインターフェースの入力ボックスでは、1 行につき 1 つの参照を入力することができます。すべてのプラットフォームでは、ファイルを保存した際に、複数のエントリがコロン (:) で区切られた単一文字列に変換されます。ファイルを手動で編集する際は、区切り文字のコロン (:) を保持する必要があります。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

接続属性の手動宣言

```
ENGINE SQLNEZ SET CONNECTION_ATTRIBUTES connection  
'URL'/userid,password
```

説明

SQLNEZ

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

この特定の属性のセットを識別する論理名です。

connection と URL の間にスペースが 1 つ必要です。

URL

Netezza データソースのロケーション URL です。

userid

ターゲットデータベースに登録されたプライマリ認証 ID です。

password

プライマリ認証 ID に関連するパスワードです。

接続属性の宣言

以下の SET CONNECTION_ATTRIBUTES コマンドは、明示的なユーザ ID (MYUSER) およびパスワード (PASS) を使用して、Netezza ドライバを介してデータソースにアクセスします。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザイン

ターフェースまたは Data Migrator デスクトップインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

```
ENGINE SQLNEZ SET CONNECTION_ATTRIBUTES CON1  
'jdbc:xxxxxxx://hostname:port/datasource'
```

デフォルト接続の上書き変更

複数の接続を定義している場合、最初の SET CONNECTION_ATTRIBUTES コマンドに指定した接続が、デフォルトの接続として設定されます。このデフォルト値は、SET DEFAULT_CONNECTION コマンドを使用して上書きすることができます。

デフォルト接続の変更

```
ENGINE SQLNEZ SET DEFAULT_CONNECTION connection
```

説明

SQLNEZ

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

```
(FOC1671) コマンドの順序に誤りがあります。
```

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

デフォルト接続の選択

以下の SET DEFAULT_CONNECTION コマンドを発行すると、デフォルト接続として CON1 が選択されます。

```
ENGINE SQLNEZ SET DEFAULT_CONNECTION CON1
```

Netezza メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および JDBC データタイプのサーバマッピングを定義するシノニムを作成します。

アダプタの識別

リクエストの解釈に必要なアダプタを識別するには、マスターファイルの SUFFIX 属性を使用します。Netezza アダプタを識別するには、SUFFIX 値として SQLNEZ を使用します。

アダプタの識別

```
FILE[NAME]=file, SUFFIX=SQLNEZ [, $]
```

説明

file

マスターファイルのファイル名です。拡張子.mas を持たないファイルには、8 バイト以内の文字でファイル名を指定することができます。このファイル名は文字で開始する必要

があります。また、テーブルまたはビューの内容を表す名前にすることをお勧めします。実際のファイルには拡張子 .mas を付ける必要がありますが、この属性の値には拡張子を含めないでください。

SQLNEZ

アダプタの値です。

データベーステーブルへのアクセス

Netezza を使用してリモートのサードパーティのテーブルにアクセスする場合、RDBMS の Netezza ドライバをローカルマシンにインストールする必要があります。

サーバは、JDBC ネットワーク上のサードパーティのデータベーステーブルにアクセスすることが可能です。データソースの URL を指定する必要があります。また、必要に応じて、アクセスするデータベースのユーザ ID およびパスワードを指定します。これらのパラメータは、サーバのグローバルプロファイル、またはユーザプロファイルのいずれかで定義できます。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な各オブジェクトの一意的な名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ペー

ジで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Netezza のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

データベース

[データベース] ドロップダウンリストからデータベースを選択します。ここには、現在の DBMS インスタンスにあるすべてのデータベースが表示されます。

外部 SQL スクリプトのパス

[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合は、さらに次の項目が表示されます。

次の標準名前規則は UNIX に適用されます。

- [ベースパス] テキストボックスに、SQL クエリが記述されたファイルの物理ディレクトリパスを指定します。ディレクトリ名を直接入力するか、[...] (参照) ボタンをクリックしてファイルの格納先ディレクトリへ移動します。このクリックにより、[ベースパスの選択] ダイアログボックスが表示されます。
- [ドキュメント名] テキストボックスに、ファイル名を入力します。必要に応じて、ワイルドカード文字を使用します。
- [ドキュメント拡張子] テキストボックスに、スクリプトファイルの拡張子を入力して、その拡張子のファイルのみを表示します。

アダプタはシノニムの作成中にネイティブ API コールを発行して選択リスト内の要素のリストを取得し、各要素に対するフィールドを持つマスターファイルを構成します。生成されたアクセスファイルは、DATASET 属性の SQL スクリプトパスを参照します。これには

SQL クエリが格納されたファイルの名前と拡張子を含む、フルパスが記述されています。以下はその例です。

```
DATASET=/ul/home2/apps/report3.sql
```

WebFOCUS レポートを作成する際は、SQL クエリを使用してデータにアクセスします。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (:) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

オーナー/スキーマ

ユーザが所有する 1 つまたは複数のオブジェクトを作成したユーザアカウントです。

テーブル名

基になるオブジェクトの名前です。

タイプ

オブジェクトタイプ (テーブル、ビューなど)。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにしま

す。

- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

生成されたシノニムの例

Netezza アダプタのシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

注意：サンプルデータの nf29004 テーブルは、製品には同梱されていません。

マスターファイル nf29004.mas

```
FILE=DIVISION, SUFFIX=SQLNEZ , $
SEGNAME=SEG1_4, SEGTYPE=S0 , $
FIELD=DIVISION4, DIVISION4, I9, I4, MISSING=OFF , $
FIELD=DIVISION_NA4, DIVISION4, I9, I4, MISSING=OFF , $
FIELD=DIVISION_HE4, DIVISION4, I9, I4, MISSING=OFF , $
```

アクセスファイル nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=EDAQA.NF29004,
CONNECTION=CON1, KEYS=1, WRITE=YES, $
```

アクセスファイルのキーワード

下表は、アクセスファイルで使用されるキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABLENAME	Netezza テーブルを識別します。この属性に割り当てる値には、次のようにオーナー (スキーマとも呼ばれる) の名前とデータベースリンク名を含めることができます。

キーワード	説明
	<pre>TABLENAME=[owner.]table</pre>
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre> CONNECTION=' ' は、ローカルのデータソースへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。 <pre>KEY=fld1/fld2/.../fldn</pre>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。
KEYFLD IXFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定します。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 KEYFLD は、親テーブルの共通カラムの FIELDNAME

キーワード	説明
	です。 IXFLD は関連するテーブルの共通カラムの FIELDNAME です。 KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることをお勧めしますが、必須ではありません。 注意：KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

Netezza データタイプのサポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[アクセスファイル](#)を参照してください。

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアドプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

Netezza 環境のカスタマイズ

Netezza アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。

タイムアウト制限の指定

Netezza に対して SQL リクエストを実行した後、アダプタがその応答を待機する時間 (秒) を指定するには、TIMEOUT コマンドを使用します。

TIMEOUT コマンドの発行

```
ENGINE SQLNEZ SET TIMEOUT {nn|0}
```

説明

SQLNEZ

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nn

タイムアウトとなる秒数を指定します。デフォルト値は 30 です。

0

応答の待機時間が無制限であることを示します。

長時間実行中リクエストのキャンセル

WebFOCUS Reporting Server ブラウザインターフェースで、長時間実行中のリクエストをキャンセルすることができます。この操作を実行すると、ネイティブ JDBC ドライバの機能に応じて、リクエスト全体がキャンセルされるか、フェッチサイクルが中断されます。

長時間実行中のリクエストをキャンセルするには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [ワークスペース] ページで、[Java サービス] 下の [DEFAULT] を選択します。[DEFAULT] を右クリックし、[エージェント] を選択します。
2. [Java サービスエージェント] ウィンドウで、キャンセルする jscomid の行を選択し、右クリックして [終了] を選択します。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、メニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

```
ENGINE SQLNEZ SET PASSRECS {ON|OFF}
```

説明

SQLNEZ

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

HOLD DBMS 作成の制御

HOLD AS *dbms_name* FORMAT SQLNEZ 構文が拡張され、HOLD DBMS テーブルの作成に関する、より厳密な制御が可能になりました。

DBMS 作成の制御

```
HOLD [AS dbms_name] FORMAT SQLNEZ [CONNECTION conn_name]
```

説明

dbms_name

作成する DBMS テーブルです。1 部構成、2 部構成、または 3 部構成の名前にします。複数の部分を区切るには、DBMS に適合した区切り記号を使用します。通常はピリオド (.) です。

conn_name

DBMS の接続名です。複数の DBMS 接続を構成して使用している場合は、conn_name で接続名を指定します。

注意：同一の名前と接続のテーブルが存在する場合、そのテーブルは削除されます。

Netezza テーブルの作成

次のコマンドは、CON1 接続を使用して、DBCS_NEZ という名前の Netezza テーブルを作成します。

```
ON TABLE HOLD AS DBCS_NEZ FORMAT SQLNEZ CONNECTION CON1
```

Netezza 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)および[定数として定義された一時項目 \(DEFINE\) を転送するリクエストの最適化](#)を参照してください。

非等価条件 WHERE ベース LEFT OUTER JOIN の最適化

LEFT OUTER JOIN は、ホストテーブルからすべてのレコードを選択し、それらをクロスリファレンステーブルのレコードと照合します。一致するレコードが存在しない場合、ホスト

レコードは保持されますが、クロスリファレンスフィールドにはデフォルト値 (ブランクまたはゼロ) が割り当てられます。アダプタは、RDBMS で条件式がサポートされるすべての WHERE ベースの LEFT OUTER JOIN コマンドを最適化することができます。

条件付き LEFT OUTER JOIN の指定

```
JOIN LEFT_OUTER FILE hostfile AT hfld1 [TAG tag1]
  [WITH hfld2]
  TO {UNIQUE|MULTIPLE}
  FILE crfile AT crfld [TAG tag2] [AS joinname]
  [WHERE expression1;
  [WHERE expression2;
  ...]

END
```

説明

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。

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 パラメータにフィールド値を指定しても、この値が接続情報として使用されることはありません。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

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に一意の修飾子として使用するタグ名 (オプション) です。

WITH 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 に基づく条件付き JOIN では、KEEPDEFINES を ON に設定し、一時項目 (DEFINE) を JOIN コマンドを発行する前に作成しておく必要があります。

MULTIPLE

from_file と *to_file* 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。なお、ONE は UNIQUE の同義語です。

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 (文字フィールドは空白、数値フィールドは 0 (ゼロ)) が指定されます。

ユニーク JOIN は WebFOCUS の概念です。RDBMS はユニークと非ユニークを区別しないため、クロスリファレンスファイルから条件に一致するすべての行を取得します。

たとえば、リクエストでユニーク JOIN として指定したが、実際には条件に一致する行がクロスリファレンスファイルに複数存在する場合、この JOIN の処理を RDBMS が行くと、RDBMS は一致するすべての行を返します。一方、最適化を無効にしてこの JOIN の処理を WebFOCUS が行くと、異なるレポート結果が得られます。WebFOCUS はユニーク JOIN の概念を順守するため、ホストの 1 行に対してクロスリファレンスの行を 1 行だけ返します。

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データソース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。

WHERE ベース OUTER JOIN の最適化条件

- WHERE ベースの LEFT OUTER JOIN を最適化するには、それに関係する RDBMS で式の最適化を行うことができるとともに、次の条件のいずれかを満たす必要があります。
 - JOIN WHERE コマンドに、少なくとも 1 つの「field1 EQ field2」述語が含まれている。ここで、field1 は table1 のフィールド、field2 は table2 のフィールドを表します。
 - または
 - 右側のテーブルに、NULL データを含まないユニークインデックスまたはキーがある。
 - または
 - 右側のテーブルに、長いデータタイプ (例、TEXT、IMAGE) を持たない「NOT NULL」フィールドが少なくとも 1 つ含まれている。
- アダプタの SQLJOIN OUTER 設定は、ON (デフォルト) に設定しておく必要があります。

非等価条件 LEFT OUTER JOIN の最適化

次のリクエストは、2 つの Netezza データソース間の条件付き LEFT OUTER JOIN を作成し、結合後のデータソースを使用してレポートを作成します。このリクエストに対して生成された SQL を表示するため、STMTRACE が ON に設定されています。

```
SET TRACEUSER = ON
SET TRACEOFF = ALL
SET TRACEON = STMTRACE//CLIENT
JOIN LEFT_OUTER FILE baseapp/EQUIP AT CARS
TO ALL FILE baseapp/CARREC AT CARC
WHERE CARS NE CARC;
END
```

```
TABLE FILE baseapp/EQUIP  
PRINT CARS CARC STANDARD  
BY MODEL  
END
```

WebFOCUS リクエストは、LEFT OUTER JOIN を組み込んだ単一の Netezza SELECT ステートメントに変換され、非等価条件が ON 句で RDBMS に渡されます。

```
SELECT T1."CARS"(CHAR( 16)),T1."STANDARD"(CHAR( 40)),  
T2."CARC"(CHAR( 16)),T2."MODEL"(CHAR( 24)) FROM ( EQUIP T1 LEFT  
OUTER JOIN CARREC T2 ON (T1."CARS" <> T2."CARC") ) ORDER BY  
T2."MODEL";
```

ODBC アダプタの使用

ODBC アダプタの構成方法について説明します。

ODBC アダプタを使用すると、アプリケーションから ODBC データソースへのアクセスが可能になります。このアダプタにより、アプリケーションリクエストがネイティブの ODBC ステートメントに変換され、最適化されたアンサーセットがリクエストの送信元アプリケーションに返されます。

ODBC 環境の準備

ODBC アダプタを使用するには、Windows システムに Microsoft の 32 ビットの ODBC ドライバマネージャをインストールする必要があります。ODBC アダプタは、32 ビット ODBC ドライバマネージャで定義されたすべてのインストール済み ODBC ドライバにアクセスすることができます。

注意

- 特定のデータソースタイプのアダプタが使用可能な場合は、そのアダプタを使用して特定のタイプにアクセスする必要があります。ODBC アダプタでは、特定のアダプタが存在するデータソースタイプはサポートされません。たとえば、Db2 データベースにアクセスするために Db2 アダプタが使用可能な場合は、ODBC アダプタを使用して Db2 データベースにアクセスすることはできません。
- 特定のデータソースタイプのアダプタが現在使用可能でない場合は、ベンダーから提供された ODBC ドライバとともに ODBC アダプタを使用することは可能です。ただし、そのアダプタが一般的な ODBC 標準に準拠する必要があります。一方、特定のデータソース要件に適合するよう、これらの標準をベンダー側で解釈したり、拡張したりしている場合があります。その結果として互換性が保持されない場合、そのベンダーのデータソースタイプのアダプタを使用する際に、動作に不整合が発生する可能性があります。この問題が発生した場合は、技術サポートに問い合わせてください。ユーザのデータアクセス要件に応じたサポートを提供します。

ODBC アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

アダプタは、ODBC データベースサーバに接続する際に、接続情報および認証情報を必要とします。この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。選択したプロファイルにコマンドが追加されます。選択可能なプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) があります。
- グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) のいずれかに手動でコマンドを追加します。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の ODBC データベースサーバへの接続を宣言することができます。ODBC Server への実際の接続は、その接続を参照する最初のクエリ発行時に確立されます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

ODBC の接続属性

ODBC アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

データソース

データソース名 (DSN) です。デフォルトのデータソース名はありません。値を入力する必要があります。

ファイル DSN を使用する場合は、そのファイル名にする必要があります (例、samplefdsn.dsn)。

セキュリティ

ODBC データソースへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報が ODBC データソースへの接続時に認証情報として渡されます。
- **Trusted** アダプタは、サーバのデータアクセスエージェントによって偽装されたオペレーティングシステムユーザの認証情報を使用して、Windows ログインとして ODBC データソースに接続します。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

ファイル DSN パラメータ (オプション)

使用中の Windows マシンまたはネットワークで定義された ODBC ファイルの DSN へのパスです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (括弧は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

接続属性の手動宣言

Explicit 認証 ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報が ODBC データソースへの接続時に認証情報として渡されます。

ユーザ/システム DSN の場合

```
ENGINE SQLODBC SET CONNECTION_ATTRIBUTES connection DSN_name/userid,password
```

ファイル DSN の場合

```
ENGINE SQLODBC SET CONNECTION_ATTRIBUTES connection  
FileDSN_name/userid,password:'FileDSN_path';'
```

Trusted 認証 アダプタは、サーバのデータアクセスエージェントによって偽装された Windows ユーザの認証情報を使用して、Windows ログインとして ODBC に接続します。

ユーザ/システム DSN の場合

```
ENGINE SQLODBC SET CONNECTION_ATTRIBUTES connection DSN_name/,
```

ファイル DSN の場合

```
ENGINE SQLODBC SET CONNECTION_ATTRIBUTES connection  
FileDSN_name/,:'filedsn=FileDSN_path';',
```

説明

SQLODBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

この接続属性のセットの識別に使用される論理名です。

connection と *DSN_name* の間にスペースが 1 つ必要です。

DSN_name

アクセスする ODBC データソース名 (DSN) です。この名前は、odbc.ini ファイルのエントリに一致させる必要があります。

FileDSN_name

アクセスする ODBC ファイルデータソース名 (DSN) です。

userid

ODBC に登録されたプライマリ認証 ID です。

password

プライマリ認証 ID に関連するパスワードです。

FileDSN_path

ハードドライブ上またはネットワーク上の ODBC ファイル DSN へのパスです。

接続属性の宣言

ユーザ/システム DSN の場合

以下の SET CONNECTION_ATTRIBUTES コマンドは、明示的なユーザ ID (MYUSER) およびパスワード (PASS) を使用して、SAMPLESERVER という名前の ODBC DSN への接続 CON1 を宣言します。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

```
ENGINE SQLODBC SET CONNECTION_ATTRIBUTES CON1 SAMPLESERVER/MYUSER,PASS
```

ファイル DSN の場合

以下の SET CONNECTION_ATTRIBUTES コマンドは、明示的なユーザ ID (MYUSER) およびパスワード (PASS) を使用して、SAMPLESERVER.dsn という名前の ODBC ファイル DSN を介した接続 CON1 を宣言します。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

```
ENGINE SQLODBC SET CONNECTION_ATTRIBUTES CON2  
SAMPLESERVER.dsn /R729999B,1525B6F3331C2FB3:'filedsn=C:¥ SAMPLESERVER.dsn;'
```

接続情報の更新

このアダプタの CONNECTION_ATTRIBUTES コマンドの構文が拡張され、アプリケーションの開発環境から実稼働環境への移植をサポートするよう設計された論理接続名を指定できるようになりました。この拡張された構文では、既存の CONNECTION_ATTRIBUTES コマンドのマイグレートが必要になる場合があります。

WebFOCUS Reporting Server ブラウザインターフェースの [マイグレート] オプションを使用すると、現在のサーバ設定が新しいバージョンにマイグレートされます。このオプションにアクセスするには、[ツール] メニューの [ワークスペース] を選択します。リボンの [設定] から [マイグレート] を選択します。[マイグレート] ウィンドウで、構成インスタンスディレクトリ (EDACONF) の完全なパスを入力して [続行] ボタンをクリックします。この方法を使用することをお勧めします。

[マイグレート] オプションを使用しない場合、以下の情報に注意してください。

- バージョン 7.6 SP01 より前のバージョンで宣言された接続名は引き続きサポートされます。
- 新規のシノニムを作成する目的で新しい接続を追加する場合、既存の接続は新しいフォーマットで再保存され、また既存のシノニムは変更を行うことなく引き続き機能します。
- 既存のシノニムを使用する目的で新しい接続を追加する場合、デフォルトの論理接続名を、既存のアクセスファイルの CONNECTION=value 属性で指定された値に一致するように変更する必要があります。

たとえば、バージョン 7.6 SP01 より前のバージョンで、以下のように接続が定義されている場合を想定します。

```
ENGINE SQLODBC SET CONNECTION_ATTRIBUTES DSN_A/uid,pwd
```

この DSN_A に保存されているオブジェクトに基づいてシノニムを作成すると、アクセスファイルには以下の記述が追加されます。

```
CONNECTION=DSN_A
```

この後、新しい接続を追加する際に、この既存のシノニムを再利用するには、接続名をデフォルトの CON01 から DSN_A に変更し、DSN_A として保存する必要があります。この接続は、以下のようにプロファイルに保存されます。

```
ENGINE SQLODBC SET CONNECTION_ATTRIBUTES DSN_A DSN_A/uid,pwd
```

デフォルト接続の上書き変更

複数の接続を定義した場合、最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続がデフォルト接続として機能します。このデフォルト値は、SET DEFAULT_CONNECTION コマンドを使用して上書きすることができます。

デフォルト接続の変更

```
ENGINE SQLODBC SET DEFAULT_CONNECTION connection
```

説明

SQLODBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

```
ENGINE SQLODBC SET DEFAULT_CONNECTION SAMPLE
```

接続範囲の制御

確立する接続ごとにアダプタを使用する場合、SET AUTODISCONNECT コマンドを使用して接続の継続を制御することができます。

接続範囲の制御

```
ENGINE SQLODBC SET AUTODISCONNECT ON {FIN|COMMAND|COMMIT}
```

説明

SQLODBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMAND

各リクエストの後に自動的に切断されます。SET AUTODISCONNECT コマンドは、イベントがどの程度の頻度で発生するかによって、大幅なオーバーヘッドをもたらす可能性があります。このオーバーヘッドのほとんどすべてがサーバには関連しません。このオーバーヘッドは、オペレーティングシステムとデータソースに関連します。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

ODBC メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および ODBC データタイプのサーバマッピングを定義するシノニムを作成します。

アダプタの識別

リクエストの解釈に必要なアダプタを識別するには、マスターファイルの SUFFIX 属性を使用します。ODBC アダプタを識別するには、SUFFIX 値として SQLODBC を使用します。

アダプタの識別

```
FILE[NAME]=file, SUFFIX=SQLODBC [,,$]
```

説明

file

マスターファイルのファイル名です。このファイル名は文字で開始する必要があります。また、テーブルまたはビューの内容を表す名前にすることをお勧めします。実際のファイルには拡張子 .mas を付ける必要がありますが、この属性の値には拡張子を含めないでください。

SQLODBC

アダプタの値です。

データベーステーブルへのアクセス

ODBC を使用してリモートのサードパーティのテーブルにアクセスする場合、RDBMS の ODBC ドライバをローカルにインストールする必要があります。

サーバは、ネットワーク ODBC 上のサードパーティのデータベーステーブルにアクセスすることが可能です。システムデータソース名を指定する必要があります。また、必要に応じ

て、アクセスするデータベースのユーザ ID およびパスワードを指定します。これらのパラメータは、サーバのグローバルプロファイル、またはユーザプロファイルのいずれかで定義できます。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な ODBC テーブルまたはビューの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタ

イプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。

- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

ODBC のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

外部 SQL スクリプトのパス

[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合は、さらに次の項目が表示されます。

次の標準名前規則は UNIX に適用されます。

- [ベースパス] テキストボックスに、SQL クエリが記述されたファイルの物理ディレクトリパスを指定します。ディレクトリ名を直接入力するか、[...] (参照) ボタンをクリックしてファイルの格納先ディレクトリへ移動します。このクリックにより、[ベースパスの選択] ダイアログボックスが表示されます。
- [ドキュメント名] テキストボックスに、ファイル名を入力します。必要に応じて、ワイルドカード文字を使用します。
- [ドキュメント拡張子] テキストボックスに、スクリプトファイルの拡張子を入力して、その拡張子のファイルのみを表示します。

アダプタはシノニムの作成中にネイティブ API コールを発行して選択リスト内の要素のリストを取得し、各要素に対するフィールドを持つマスターファイルを構成します。生成されたアクセスファイルは、DATASET 属性の SQL スクリプトパスを参照します。これには SQL クエリが格納されたファイルの名前と拡張子を含む、フルパスが記述されています。以下はその例です。

```
DATASET=/u1/home2/apps/report3.sql
```

WebFOCUS レポートを作成する際は、SQL クエリを使用してデータにアクセスします。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

外部キーを使用したクラスタ作成 (非推奨。[シノニムの編集] の使用を推奨)

[外部キーを使用したクラスタ作成] のチェックをオンにすると、現在のテーブルに外部キーで関係付けられているテーブルすべてをこのシノニムに含めることができます。ただし、このオプションは廃止される予定で、クラスタの作成にはシノニムエディタを使用することをお勧めします。生成されるマルチテーブルのシノニムは、このテーブルの外部キーの関係をすべて定義します。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロンの (;) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスはブランクにしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[データタイプのサポート](#)を参照してください。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

オーナー/スキーマ

ユーザが所有する 1 つまたは複数のオブジェクトを作成したユーザアカウントです。

テーブル名

基になるオブジェクトの名前です。

タイプ

オブジェクトタイプ (テーブル、ビューなど)。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューの

チェックをオンにします。

生成されたシノニムの例

ODBC アダプタのシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

マスターファイル nf29004.mas

```
FILE=DIVISION, SUFFIX=ODB , $
SEGNAME=SEG1_4, SEGTYPE=S0 , $
FIELD=DIVISION4, DIVISION4, I9, I4, MISSING=OFF, $
FIELD=DIVISION_NA4, DIVISION4, I9, I4, MISSING=OFF, $
FIELD=DIVISION_HE4, DIVISION4, I9, I4, MISSING=OFF, $
```

アクセスファイル nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=EDAQA.NF29004,
CONNECTION=DB1, KEYS=1, WRITE=YES, $
```

アクセスファイルのキーワード

下表は、アクセスファイルで使用されるキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABlename	ODBC テーブルを識別します。この属性に割り当てる値には、次のようにオーナー (スキーマとも呼ばれる) の名前とデータベースリンク名を含めることができます。 TABlename=[owner.]table

キーワード	説明
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre> CONNECTION=' ' は、ローカルの ODBC データベースサーバへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
DBSPACE	テーブルの記憶域を示すオプションのキーワードです。以下はその例です。 <pre>datasource.tablespace DATABASE datasource</pre>
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。 <pre>KEY=fld1/fld2/.../fldn</pre>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。
KEYFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定しま

キーワード	説明
IXFLD	す。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 - KEYFLD は、親テーブルの共通カラムの FIELDNAME です。 - IXFLD は関連するテーブルの共通カラムの FIELDNAME です。 KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることをお勧めしますが、必須ではありません。 注意：KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

データタイプのサポート

サポートされるデータタイプは、実際のデータソースに応じて異なります。

ODBC アダプタを使用してデータソースにアクセスする場合は、製品にそのタイプのアダプタが提供されているかどうかを確認してください。

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアードプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

ODBC 環境のカスタマイズ

ODBC アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。

タイムアウト制限の指定

ODBC に対して SQL リクエストを実行した後、アダプタがその応答を待機する時間 (秒) を指定するには、TIMEOUT コマンドを使用します。

TIMEOUT コマンドの発行

```
ENGINE SQLODBC SET TIMEOUT {nn|0}
```

説明

SQLODBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nn

タイムアウトとなる秒数を指定します。デフォルト値は 30 です。

0

応答の待機時間が無制限であることを示します。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

```
ENGINE SQLODBC SET PASSRECS {ON|OFF}
```

説明

SQLODBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

トランザクションの分離レベルの指定

WebFOCUS Reporting Server ブラウザインターフェースから、または SET ISOLATION コマンドを使用して、トランザクションの分離レベルを指定することができます。

SET コマンドによるトランザクション分離レベルの指定

以下のコマンドを実行して、トランザクションの分離レベルを指定することができます。

```
ENGINE SQLODBC SET ISOLATION {RU|RC|RR|SE}
```

説明**RU**

トランザクションの分離レベルを非コミット読み取りに設定します。

RC

トランザクションの分離レベルをカーソル固定に設定します。

RR

トランザクションの分離レベルを反復可能読み取りに設定します。

SE

トランザクションの分離レベルを読み取り固定に設定します。

ODBC 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)を参照してください。

Oracle アダプタの使用

Oracle アダプタの構成方法について説明します。

Oracle アダプタを使用すると、アプリケーションから Oracle データソースへのアクセスが可能になります。このアダプタにより、データまたはアプリケーションリクエストがネイティブの Oracle ステートメントに変換され、最適化されたアンサーセットがリクエストの送信元プログラムに返されます。

注意：バージョン 9.3.0 以降、Oracle アダプタは、更新、挿入、削除された行についての MERGE 統計を提供します。

Oracle 環境の準備

Oracle アダプタを使用するには、少なくとも Oracle データベースクライアントをインストールする必要があります。Oracle クライアントによって、ローカルまたはリモートの Oracle データベースサーバに接続することが可能となります。

UNIX では、クライアント共有ライブラリ libclntch を生成する必要があります。

アダプタの Unicode のサポートを利用する場合、[Unicode のための Oracle の構成](#)を参照してください。

Windows で Oracle 環境を準備するには

Microsoft Windows では、Oracle 環境は Oracle のインストール時に設定されます。

UNIX で Oracle 環境を準備するには

edastart ファイルで以下のエクスポートコマンドを発行することができます。別の方法として、これらのコマンドを、別のシェルフファイル、データベースプロファイル、ユーザプロフィールで発行することもできます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [ワークスペース] ページを開きます。ナビゲーションウィンドウで [構成ファイル] フォルダを展開し、[サーバプロファイル - edastart.prfl] をダブルクリックして、edastart ファイルを編集します。シェル設定の直後にエクスポートコマンドを指定します。以下はその例です。

```
#!/bin/ksh
export ORACLE_SID=orac
export ORACLE_HOME=/usr/oracle/orac
. . .
```

2. UNIX の環境変数 \$ORACLE_SID を使用して、アクセスする Oracle データベースインスタンスを指定します。以下はその例です。

```
ORACLE_SID=orac
export ORACLE_SID
```

3. UNIX の環境変数 \$ORACLE_HOME を使用して、アクセスする Oracle データベースインスタンスのロケーションを指定します。たとえば、Oracle ソフトウェアのホームディレクトリを /usr/orac に設定するには、以下のように指定します。

```
ORACLE_HOME=/usr/oracle/orac
export ORACLE_HOME
```

4. Sun Sparc では \$LD_LIBRARY_PATH、AIX では \$LIBPATH などの、プラットフォームに応じて異なる UNIX 環境変数を使用して、Oracle 共有ライブラリのパスを指定します。

これはセキュリティが設定されていないサーバにのみ必要です。以下はその例です。

```
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

注意：セキュリティを ON に設定してサーバを実行している場合、LD_LIBRARY_PATH 変数は無視されます。この場合は、LIBPATH を使用する必要があります。

リモートの Oracle データベースサーバへの接続

Oracle クライアントの配信の標準規則を使用して、サーバは以下の接続をサポートします。

- ローカルの Oracle データベースサーバ。
- リモートの Oracle データベースサーバ。リモートの Oracle データベースサーバに接続するには、ソースマシン上の Oracle の tnsnames.ora ファイルにターゲットマシンを指示するエントリが含まれており、ターゲットマシン上でリスナ処理が実行されていなければなりません。

Oracle データベースサーバに接続した場合、サーバは他の Oracle データベースサーバ上の Oracle テーブルへのアクセスに使用できる Oracle DATABASE LINK を定義します。

XA のサポート

Oracle データソースにアクセスする読み取り/書き込みアプリケーションは、XA 互換モードで管理されているトランザクションを実行することができます。

XA トランザクション管理機能を有効にするには、Web コンソールの構成機能を使用してサーバをトランザクション調整モードで構成する必要があります。トランザクション調整モードを使用することにより、関連するすべての DBMS に対するデータ修正の整合性が保証され、データ修正の一部がある DBMS に対してコミットされ、別の DBMS に対しては終了されることを回避することができます。

XA のサポートについての詳細は、[XA のサポート](#)を参照してください。

Unicode のための Oracle の構成

アダプタが Unicode をサポートするのは、Oracle バージョン 10g 以降のデータベースで、NLS_CHARACTERSET パラメータが UTF8 に設定されている場合です。edastart ファイルまたは別のシェルフファイルで NLS_LANG 環境変数を設定しておく必要があります。

注意

- NCHAR/NVARCHAR フィールド の Oracle Unicode データは、WebFOCUS Reporting Server が Unicode 用に構成されている場合にサポートされます。

- Oracle 環境変数 `ORA_NCHAR_LITERAL_REPLACE` により、`NCHAR/NVARCHAR` フィールドへの透過的なデータ変換が制御されます。パフォーマンスの向上のため、この変数は通常は未設定になっています。

ただし、使用頻度は低いですが、以下のような場合には、`ORA_NCHAR_LITERAL_REPLACE` を `TRUE` に設定する必要があります。

- SQL パススルーを使用して Oracle `NCHAR/NVARCHAR` フィールドへの入力を行う場合 (`INSERT/UPDATE/WHERE` 句)。さらにリテラルが `N'<string>'` という表記である必要があります。
- `TABLE FILE` に `NCHAR/NVARCHAR` フィールドを持つ `IF/WHERE` 句がある場合。

NLS_LANG 環境変数の設定

`NLS_LANG` 環境変数を Unicode をサポートするように設定するための構文は、以下のとおりです。

```
NLS_LANG = language_territory.characterset
```

説明

language

選択した言語です。

territory

選択した言語に関連付けられた国名です。

characterset

Oracle データベースで設定する `NLS_CHARACTERSET` 変数の値です。Unicode では、この値は常に `UTF8` に設定します。

NLS_LANG 変数に英語 (米国) UTF-8 を設定する例

英語 (米国) UTF-8 の場合、以下を設定する必要があります。


```
NLS_LANG=American_America.UTF8
```

Oracle アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

アダプタは、Oracle データベースサーバに接続する際に、接続情報および認証情報を必要とします。この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。選択したプロファイルにコマンドが追加されます。選択可能なプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) があります。
- グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) のいずれかに手動でコマンドを追加します。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の Oracle データベースサーバへの接続を宣言することができます。Oracle サーバへの実際の接続は、その接続を参照する最初のクエリ発行時に確立されます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。

6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Oracle の接続属性

Oracle アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルトの接続名はありません。値を入力する必要があります。

TNS 名

ネットワーク上の Oracle データベースサーバの接続記述子として使用されるサービス (TNS) 名です。この名前は、tnsnames ファイル内の有効なエントリを指定する必要があります。

ローカルデータベースサーバへの接続を <local> として入力することができます。<local> は、tnsnames.ora ファイルと一致しないこともあることに注意してください。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 3 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。

注意：使用する Oracle インスタンスに Oracle Proxy Authentication が設定されている場合は、Oracle ベンダーのマニュアルの説明に従って、Oracle Proxy Authentication による接続の構成を Explicit 認証を使用して行えます。このタイプの認証を構成するには、[ユーザ] フィールドに次の構文を入力します。

```
srv_userid[proxy_userid]
```

説明

srv_userid は、Oracle へのサーバ接続に使用するユーザ ID です。

proxy_userid は、プロキシサーバのユーザ ID です。

- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。
- **Trusted** アダプタは、現在のオペレーティングシステムに関連している偽装プロセス用のデータベースの規則を使用してデータベースに接続します。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

接続属性の手動宣言

Explicit 認証 ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報が Oracle への接続時に認証情報として渡されます。

```
ENGINE SQLORA SET CONNECTION_ATTRIBUTES connection TNS_name/userid,password
```

Password Passthru 認証 ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報が Oracle への接続時に認証情報として渡されます。

```
ENGINE SQLORA SET CONNECTION_ATTRIBUTES connection TNS_name/
```

Trusted 認証 アダプタは、サーバのデータアクセスエージェントによって偽装されたオペレーティングシステムユーザの認証情報を使用して、オペレーティングシステムログインとして Oracle に接続します。

```
ENGINE SQLORA SET CONNECTION_ATTRIBUTES connection TNS_name/,
```

説明

SQLORA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

この特定の属性のセットを識別する論理名です。

connection と *TNS_name* の間にスペースが 1 つ必要です。

TNS_name

ネットワーク上の Oracle データベースサーバの接続記述子として使用されるサービス (TNS) 名です。この名前は、tnsnames ファイル内の有効なエントリを指定する必要があります。ローカルデータベースサーバへの接続を <local> として入力することができます。<local> は、tnsnames.ora ファイルと一致しないこともあることに注意してください。

userid

Oracle に登録されたプライマリ認証 ID です。

password

プライマリ認証 ID に関連するパスワードです。

接続属性の宣言

以下の SET CONNECTION_ATTRIBUTES コマンドを実行すると、アプリケーションが明示的なユーザ ID (MYUSER) およびパスワード (PASS) を使用して、SAMPLESERVER (tnsnames.ora ファイルで定義された名前) という名前の Oracle データベースサーバにアクセスすることができます。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

```
ENGINE SQLORA SET CONNECTION_ATTRIBUTES SAMPLESERVER SAMPLESERVER/MYUSER,PASS
```

以下の SET CONNECTION_ATTRIBUTES コマンドは、tnsnames を使用せずにローカルの Oracle データベースサーバに接続します。

```
ENGINE SQLORA SET CONNECTION_ATTRIBUTES <Local> <Local>/MYUSER,PASS
```

以下の SET CONNECTION_ATTRIBUTES コマンドは、Password Passthru 認証を使用して Oracle データベースサーバ SAMPLESERVER に接続します。

```
ENGINE SQLORA SET CONNECTION_ATTRIBUTES SAMPLESERVER SAMPLESERVER/
```

以下の SET CONNECTION_ATTRIBUTES コマンドは、オペレーティングシステムの認証を使用してローカルの Oracle データベースサーバに接続します。

```
ENGINE SQLORA SET CONNECTION_ATTRIBUTES SAMPLESERVER SAMPLESERVER/,
```

接続情報の更新

このアダプタの SET CONNECTION_ATTRIBUTES コマンドの構文が拡張され、アプリケーションの開発環境から実稼働環境への移植をサポートするよう設計された論理接続名を指定できるようになりました。この拡張された構文では、既存の CONNECTION_ATTRIBUTES コマンドのマイグレートが必要になる場合があります。

WebFOCUS Reporting Server ブラウザインターフェースの [マイグレート] オプションを使用すると、現在のサーバ設定が新しいバージョンにマイグレートされます。このオプションにアクセスするには、[ツール] メニューの [ワークスペース] を選択し、[設定] から [マイグレート] を選択します。[マイグレート] ウィンドウで、構成インスタンスディレクトリ (EDACONF) の完全なパスを入力して [続行] ボタンをクリックします。この方法を使用することをお勧めします。

[マイグレート] オプションを使用しない場合、以下の情報に注意してください。

- バージョン 7.6 SP01 より前のバージョンで宣言された接続名は引き続きサポートされます。
- 新規のシノニムを作成する目的で新しい接続を追加する場合、既存の接続は新しいフォーマットで再保存され、また既存のシノニムは変更を行うことなく引き続き機能します。
- 既存のシノニムを使用する目的で新しい接続を追加する場合、デフォルトの論理接続名を、既存のアクセスファイルの CONNECTION=value 属性で指定された値に一致するよう変更する必要があります。

たとえば、バージョン 7.6 SP01 より前のバージョンで、以下のように接続が定義されている場合を想定します。

```
ENGINE SQLORA SET CONNECTION_ATTRIBUTES DSN_A/uid,pwd
```

この DSN_A に保存されているオブジェクトに基づいてシノニムを作成すると、アクセスファイルには以下の記述が追加されます。

```
CONNECTION=DSN_A
```

この後、新しい接続を追加する際に、この既存のシノニムを再利用するには、接続名をデフォルトの CON01 から DSN_A に変更し、DSN_A として保存する必要があります。この接続は、以下のようにプロファイルに保存されます。

```
ENGINE SQLORA SET CONNECTION_ATTRIBUTES DSN_A DSN_A/uid,pwd
```

デフォルト接続の上書き変更

複数の接続を定義した場合、最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続がデフォルト接続として機能します。このデフォルト値は、SET DEFAULT_CONNECTION コマンドを使用して上書きすることができます。

デフォルト接続の変更

```
ENGINE SQLORA SET DEFAULT_CONNECTION connection
```

説明

SQLORA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

```
(FOC1671) コマンドの順序に誤りがあります。
```

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

```
ENGINE SQLORA SET DEFAULT_CONNECTION SAMPLE
```

接続範囲の制御

確立する接続ごとにアダプタを使用する場合、SET AUTODISCONNECT コマンドを使用して接続の継続を制御することができます。

接続範囲の制御

```
ENGINE SQLORA SET AUTODISCONNECT ON {FIN|COMMAND|COMMIT}
```

説明

SQLORA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMAND

各リクエストの後に自動的に切断されます。SET AUTODISCONNECT コマンドは、イベントがどの程度の頻度で発生するかによって、大幅なオーバーヘッドをもたらす可能性があります。このオーバーヘッドのほとんどすべてがサーバには関連しません。このオーバーヘッドは、オペレーティングシステムとデータソースに関連します。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

Oracle メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および Oracle データタイプのサーバマッピングを定義するシノニムを作成します。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な Oracle テーブルまたはビューの一意的な名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

ストアドプロシジャのシノニムの作成、およびストアドプロシジャに対するレポートの実行についての詳細は、[ストアドプロシジャのシノニムの作成](#)を参照してください。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Oracle のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

外部 SQL スクリプトのパス

[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合は、さらに次の項目が表示されます。

次の標準名前規則は UNIX に適用されます。

- [ベースパス] テキストボックスに、SQL クエリが記述されたファイルの物理ディレクトリパスを指定します。ディレクトリ名を直接入力するか、[...] (参照) ボタンをクリックしてファイルの格納先ディレクトリへ移動します。このクリックにより、[ベースパスの選択] ダイアログボックスが表示されます。
- [ドキュメント名] テキストボックスに、ファイル名を入力します。必要に応じて、ワイルドカード文字を使用します。
- [ドキュメント拡張子] テキストボックスに、スクリプトファイルの拡張子を入力し

て、その拡張子のファイルのみを表示します。

アダプタはシノニムの作成中にネイティブ API コールを発行して選択リスト内の要素のリストを取得し、各要素に対するフィールドを持つマスターファイルを構成します。生成されたアクセスファイルは、DATASET 属性の SQL スクリプトパスを参照します。これには SQL クエリが格納されたファイルの名前と拡張子を含む、フルパスが記述されています。以下はその例です。

```
DATASET=/ul/home2/apps/report3.sql
```

WebFOCUS レポートを作成する際は、SQL クエリを使用してデータにアクセスします。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

外部キーを使用したクラスタ作成 (非推奨。[シノニムの編集] の使用を推奨)

[外部キーを使用したクラスタ作成] のチェックをオンにすると、現在のテーブルに外部キーで関係付けられているテーブルすべてをこのシノニムに含めることができます。ただし、このオプションは廃止される予定で、クラスタの作成にはシノニムエディタを使用することをお勧めします。生成されるマルチテーブルのシノニムは、このテーブルの外部キーの関係をすべて定義します。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (:) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視

され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[Oracle データタイプのサポート](#)を参照してください。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

オーナー/スキーマ

ユーザが所有する 1 つまたは複数のオブジェクトを作成したユーザアカウントです。

テーブル名

基になるオブジェクトの名前です。

タイプ

オブジェクトタイプ (テーブル、ビューなど)。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

生成されたシノニムの例

Oracle アダプタのシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

生成されたマスターファイル nf29004.mas

```
FILE=DIVISION, SUFFIX=SQLORA , $
SEGNAME=SEG1_4, SEGTYPE=S0 , $
FIELD=DIVISION4, DIVISION4, I9, I4, MISSING=OFF , $
FIELD=DIVISION_NA4, DIVISION_NA4, A25, A25, MISSING=ON , $
FIELD=DIVISION_HE4, DIVISION_HE4, I9, I4, MISSING=ON , $
```

生成されたアクセスファイル nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=EDAQA.NF29004,
CONNECTION=ORA901, KEYS=1, WRITE=YES, $
```

Oracle のコメントのシノニムへのマッピング

Oracle テーブルのシノニムを作成する場合、アダプタは次のようにマッピングします。

- Oracle データディクショナリからシノニムのマスターファイルの REMARK 属性への Oracle テーブルのコメントのマッピング。
- Oracle データディクショナリからシノニムのマスターファイルの DESCRIPTION 属性への Oracle カラムのコメントのマッピング。

アクセスファイルのキーワード

下表は、アクセスファイルで使用するキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABlename	Oracle テーブルを識別します。この属性に割り当てる値には、次のようにオーナー (スキーマとも呼ばれる) の名前とデータベースリンク名を含めることができます。 <pre>TABlename=[owner.]table[@databaselink]</pre>
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre> CONNECTION=' ' は、ローカルの Oracle データベースサーバへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトの

キーワード	説明
	データベースサーバへのアクセスを示します。
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。この属性を使用するには、キーを構成するフィールドをマスターファイルの先頭に記述する必要があります。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。 注意：テーブルに主キーがない場合、最初のユニークインデックスが代わりに使用されます。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。 <pre>KEY=fld1/fld2/.../fldn</pre>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。
KEYFLD IXFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定します。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 - KEYFLD は、親テーブルの共通カラムの FIELDNAME です。 - IXFLD は関連するテーブルの共通カラムの FIELDNAME です。 KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることをお勧めしますが、必須ではありません。

キーワード	説明
	注意： KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。
INDEX_NAME INDEX_UNIQUE INDEX_ COLUMNS INDEX_ORDER	データベース内のインデックス名、およびインデックスフィールドの一意性、名前、順序を示します。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

単一の SQL リクエストでの複数のデータベースサーバへのアクセス

DATABASE LINK を使用してリモートの Oracle テーブルにアクセスするには、以下の要件があります。

- 接続している Oracle データベースサーバに、有効な DATABASE LINK が定義されている必要があります。
- 照会する Oracle テーブルのアクセスファイルの TABLENAME 属性は、以下のフォーマットで記述する必要があります。

```
TABLENAME=[owner.]table@databaseLink
```

説明

owner

デフォルト設定では、ユーザ ID です。30 バイト以内で記述することができます。Oracle では、値は大文字で指定します。

table

テーブルまたはビューの名前です。30 バイト以内で記述することができます。

databaseLink

現在接続されている Oracle データベースサーバに定義されている、有効な DATABASE LINK 名です。

この TABLENAME のフォーマットは、手動または CREATE SYNONYM コマンドを使用してアクセスファイルに定義することができます。以下はその例です。

```
CREATE SYNONYM filename FOR owner.table@databaseLink DBMS SQLORA
```

上記の要件を満たす場合、テーブルのすべてのリクエストは DATABASE LINK 名を使用して、指定されたリモートの Oracle データベースサーバで実行されます。この技法は、単一の SQL リクエストで複数のリモートサーバにアクセスする別の方法です。

Oracle データタイプのサポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[アクセスファイル](#)を参照してください。

可変長データタイプのマッピングの制御

SET パラメータ VARCHAR は、Oracle データタイプ VARCHAR、VARCHAR2、NVARCHAR2 のマッピングを制御します。デフォルト設定では、サーバはこれらのデータタイプを可変長の文字 (AnV) としてマッピングします。

下表は、VARCHAR 値に基づくデータタイプのマッピングを示しています。

Oracle データ タイプ	備考	VARCHAR ON		VARCHAR OFF	
		USAGE	ACTUAL	USAGE	ACTUAL
VARCHAR (<i>n</i>)	<i>n</i> は 1 から 4000 までの整数で す。	<i>AnV</i>	<i>AnV</i>	<i>An</i>	<i>An</i>
VARCHAR2 (<i>n</i>)	<i>n</i> は 1 から 4000 までの整数で す。	<i>AnV</i>	<i>AnV</i>	<i>An</i>	<i>An</i>
NVARCHAR2 (<i>n</i>)	<i>n</i> は 1 から 4000 までの整数で す。	<i>AmV</i>	<i>AmV</i>	<i>Am</i>	<i>Am</i>

可変長データタイプのマッピングの制御

```
ENGINE SQLORA SET VARCHAR {ON|OFF}
```

説明

SQLORA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

Oracle データタイプ VARCHAR、VARCHAR2、NVARCHAR2 を可変長文字データタイプ (*AnV*) としてマッピングします。Unicode 環境ではこれは必須です。デフォルト値は ON です。

OFF

Oracle データタイプ VARCHAR、VARCHAR2、NVARCHAR2 を文字データタイプ (A) としてマッピングします。デフォルト値は ON です。

CHAR および VARCHAR2 データタイプについて

CHAR および VARCHAR2 データタイプには、特別な注意を払う必要があります。CHAR データタイプカラムと VARCHAR2 データタイプカラムを比較した場合、相違点は CHAR データタイプカラムの末尾の空白だけです。Oracle では、これらは異なるカラム値として処理されます。

SET パラメータでは、データの挿入、更新、検索に 2 つのデータタイプのどちらを使用するかを指定することができます。

サーバの外部でテーブルを作成する場合、CHAR または VARCHAR2 データタイプのどちらかを使用して、両方を併用しないことを推奨します。両方のデータタイプを併用してテーブルを作成した場合、Oracle の比較メカニズムに起因して、挿入したデータを検索できないことがあります。サーバの外部の VARCHAR2 カラムにデータを挿入する場合、末尾に空白を挿入してはいけません。

サーバを使用して Oracle テーブルの生成およびデータの検索を行う場合、ORACHAR の設定に基づいてデータタイプが CHAR または VARCHAR2 のどちらかになるため、このような問題は生じません。

ORACHAR の設定

```
ENGINE SQLORA SET ORACHAR {FIX|VAR}
```

説明

SQLORA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIX

CHAR データタイプを使用します。これがデフォルト値です。

VAR

VARCHAR2 データタイプを使用します。

SQL 式内での末尾ブランク

TABLE アダプタ内の新しい SQL ジェネレータでは、デフォルトで文字列リテラル内の末尾ブランク、および数値リテラル内の小数部と指数の表記を含むリテラルコンテンツを保持します。これにより、生成される SQL に対する制御が向上します。

末尾ブランクが不要なケースでは、次の構文が下位互換性を確保するために使用できます。

```
ENGINE SQLORA SET TRIM_LITERALS ON
```

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアードプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

NUMBER データタイプについて

NUMBER データタイプを使用する際に、精度が 1 から 31 の範囲内にあり、小数部桁数が精度を超えない正の整数の場合、そのデータタイプはサーバのデータタイプのパック 10 進数 (P) にマッピングされ、元の精度と小数部桁数が保持されます。結果の表示長さは、符号および小数点の桁数を含める必要があるため、表示長さの精度が 1 桁または 2 桁だけ超過します。

NUMBER データタイプの精度が 38、小数部桁数が 0 (ゼロ) の場合、デフォルト設定で、そのデータタイプはサーバのデータタイプの整数 (I) にマッピングされ、表示長は 11 になります。

それ以外の場合、デフォルト設定で、NUMBER データタイプはサーバのデータタイプの倍精度浮動小数点数 (D) にマッピングされ、精度が 20、小数部桁数が 2 になります。

NUMBER データタイプの精度が 32 から 38 の範囲にあり、小数部桁数が正の整数の場合、このデータタイプのデフォルトマッピングを上書きするには、ORANUMBER 設定を使用します。

ORANUMBER の設定

```
ENGINE SQLORA SET ORANUMBER {COMPAT|DECIMAL}
```

説明

SQLORA

アダプタを指定します。SET SQLENGINE=SQLORA コマンドをすでに発行している場合は、この値を省略することができます。

COMPAT

NUMBER データタイプは、上記のデフォルト動作に基づいてマッピングされます。

DECIMAL

NUMBER データタイプの精度が 32 から 38 の範囲にあり、小数部桁数が正の整数の場合、マッピングの前に NUMBER データタイプの精度 (場合によっては小数部桁数も含め) が 31 に減算された上で、上記のようにサーバのデータタイプのパック 10 進数 (P) にマッピングされます。

Oracle アダプタの設定

Oracle アダプタを使用する場合、[設定の変更] ダイアログボックスから設定可能なさまざまな設定があります。

このダイアログボックスを開くには、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで、Oracle アダプタを右クリックし、[設定の変更] をクリックします。Oracle アダプタには、次の設定があります。

バルクロードサービス

[バルクロードサービス] には、次のオプションがあります。

BULKLOAD

データの DBMS へのロードに使用する手法です。[ON] に設定すると、拡張バルクロード機能が有効になります。[OFF] に設定すると、標準 SQL INSERT ステートメントが有効になります。デフォルト値は [OFF] です。

BULKCHECK

データベースに対してトランザクションをコミット (書き込み) する行間隔です。デフォルト値は 1000000 または DBMS 固有のいずれかです。

注意：このオプションは、[BULKLOAD] パラメータが [ON] に設定されている場合にのみ適用されます。

BLK_MAX_SESSIONS

データベースにログオンする際のロードセッションの最大数です。デフォルト値は 10 です。

注意：このオプションは、[BULKLOAD] パラメータが [ON] に設定されている場合にのみ適用されます。

BLK_MAX_RESTARTS

修復可能なロードセッションエラー後に許容するロードセッション再起動の最大試行回数です。デフォルト値は 10 です。

注意：このオプションは、[BULKLOAD] パラメータが [ON] に設定されている場合にのみ適用されます。

BLK_PERM_AS_STAGE

バルクロード処理中に、永続データベーステーブルを中間テーブルとして使用する権限を制御します。デフォルト値は [ON] です。

データタイプマッピングのカスタマイズ

[データタイプマッピングのカスタマイズ] には、次のオプションがあります。

CONV_INT_PREC

整数精度列を制御します。デフォルト値は 11 です。

CONV_FLOAT_PREC

倍精度浮動小数点数精度列のマッピングを制御します。デフォルト値は 20 です。

CONV_FLOAT_SCALE

倍精度浮動小数点数小数部列のマッピングを制御します。デフォルト値は 2 です。

CONV_DEC_PREC

パック 10 進数精度列のマッピングを制御します。このデータタイプにデフォルト値はありません。

CONV_DEC_SCALE

倍精度浮動小数点数小数部列のマッピングを制御します。このデータタイプにデフォルト値はありません。

この設定は、非推奨の SET CONVERSION PRECISION p s コマンドに優先します。

CONV_LONGCHAR

LONGCHAR 列のマッピングを制御します。LONGCHAR のマッピングオプションは、[ALPHA]、[TEXT]、または [BLOB] です。デフォルト値は [ALPHA] です。

CONV_LONGCHAR_LEN

この設定は、[CONV_LONGCHAR] の値を [ALPHA] に設定した場合にのみ使用できます。デフォルト値は 256 です。

EXACT_DATETIME

日付時間列のマッピングを制御します。この設定は、互換性のために用意されています。[ON] に設定すると、日付時間の正確な精度がシノニムフィールドのフォーマットとして保持されます。たとえば、TIMESTAMP(5) は HYMDs ではなく HYMD5 にマッピングされるため、小数点以下 5 桁がすべて保持されます。デフォルト値は [OFF] です。

機能診断

[機能診断] には、次のオプションがあります。

PASSRECS

[ON] に設定すると、「(FOC1364) パススルーコマンドで影響を受けた行」というメッセージが表示されます。デフォルト値は [ON] です。このメッセージの表示を停止するには、PASSRECS 設定の値を [OFF] に設定します。

LOG_SWITCHED

ログファイル切り替えの動作を制御します。オプションには [STOP] と [PROCEED] があります。デフォルト値は [STOP] です。

CDC_LOG_SOURCE

CDC が使用するログとして、現在のログ、アーカイブログのどちらを使用するかを制御します。この値は、プロファイルまたはプロシジャの設定で指定することもできます。以下はその例です。

```
ENGINE SQLORA SET CDC_LOG_SOURCE {USE_CURRENT|USE_ARCHIVED}
```

デフォルト値は [USE_CURRENT] です。

ORACLE_RAC_NODES

ORACLE RAC ノードの数を制御します。デフォルト値は 0 (ゼロ) です。

ORACLE_SCNBATCHSIZE

ORACLE CDC バッチの SCN の数を制御します。デフォルト値は 100000 です。

SPLIT_MERGE_STATISTICS

INSERT、UPDATE、DELETE などの行に対して実行された動作に基づいて分割される MERGE ステートメント統計を制御します。デフォルト値は [ON] です。

メタデータ

メタデータのオプションは、次のとおりです。

ALIAS_CASE

新しいターゲットの ALIAS の大文字と小文字を設定することができます。

次のオプションがあります。

- **変換しない** 新しいターゲットの ALIAS のデフォルトの大文字と小文字は変更されません。
- **小文字** 新しいターゲットの ALIAS を小文字に変更します。
- **大文字** 新しいターゲットの ALIAS を大文字に変更します。

デフォルト値は [変換しない] です。

TABLE_OPTIONS

CREATE FILE コマンドによって生成された CREATE TABLE ステートメントにテーブルオプションパラメータを追加します。この設定にはデフォルト値はありません。

その他の設定

[その他の設定] には、次のオプションがあります。

JOINTYPE

自動パススルーが実行不可または無効になっている場合 (SQL SET APT=OFF)、FOCUS 生成 SQL 変換での JOIN タイプを制御します。

次のオプションがあります。

- **SORTMERGE** 各テーブルからデータを取得し、ソート、結合、集計します。SORTMERGE に設定すると、FOCUS 生成では常にソート/マージを使用してデータを取得します。
- **NESTEDLOOP** 外部テーブルから、選別条件に一致するすべての行を選択し、それらの行を使用して内部テーブルから修飾データを取得します。FOCUS 生成では、SQL ステートメント内の JOIN 可能なフィールドに equijoin が存在する場合にのみ、この方法が使用されます。このような equijoin が存在しない場合、FOCUS 生成はソート/マージに戻されます。

この設定にはデフォルト値はありません。

読み取り/書き込みの最適化

[読み取り/書き込みの最適化] には、次のオプションがあります。

FETCHSIZE

1 回の処理で DBMS から取得する行数を制御します。デフォルト値は 100 です。

INNERSIZE

1 回の処理で DBMS に渡される行数を制御します。デフォルト値は 1 です。

DBMS セッションパラメータ

[DBMS セッションパラメータ] には、次のオプションがあります。

AUTODISCONNECT

DBMS との接続を切断するタイミングを制御します。

次のオプションがあります。

- **ON-FIN** セッションの終了後に、接続されているすべての DBMS サーバから切断します。
- **ON COMMAND** 最も外側のコマンドブラケットの末尾で、SQL アダプタで接続済みのすべての DBMS サーバとの接続を解除します。
- **ON COMMIT (ROLLBACK)** COMMIT または ROLLBACK 時に、アダプタの接続を解除します。

デフォルト値は [ON-FIN] です。

NONBLOCK

コマンドの完了を連続して確認する間隔です。この時間は秒単位で指定します。これにより、物理アダプタは DBMS API への非ブロッキング呼び出しを使用するようになります。このオプションにより、暴走コマンドを Web コンソールからキャンセルできるようになります。デフォルト値は 0 (ゼロ) です。

SQL の生成

[SQL の生成] には、次のオプションがあります。

OPTIMIZATION

リクエストの最適化を制御します。

次のオプションがあります。

- **最適化 (乗法の影響が検出されない場合)** JOIN および複雑な WHERE 述語が最適化され、集計は SQL に渡されます。ただし、一部のコンストラクトを SQL に変換できない場合、および乗算の影響が検出される場合は除きます。
- **最適化 (乗法の影響を無視)** JOIN および複雑な WHERE 述語が最適化され、集計は SQL に渡されます。ただし、一部のコンストラクトを SQL に変換できない場合は除きます。この場合、乗算の影響による JOIN および集計の無効化は発生しません。
- **JOIN と複合記述子の最適化をオフ** JOIN および複雑な WHERE 述語の最適化を無効にします。
- **OUTER JOIN の最適化をオフ** OUTER JOIN の最適化のみを無効にします。
- **PRINT オブジェクトの演算を DBMS で処理する** PRINT で使用される定義済み

フィールドの計算を DBMS に委任するようアダプタに指示します。

- **AGGREGATION をオフ** 集計は無効にしますが、JOIN および複雑な WHERE 述語の最適化は無効化しません。

デフォルト値は [最適化 (乗法の影響が検出されない場合)] です。

AGGREGATE_AWARENESS

DBMS 集計認識メカニズムにより、TABLE アダプタが生成したクエリの最適化を制御します。

次のオプションがあります。

- **インストールのデフォルト** DBMS でのデフォルト設定を自動的に発行します。
- **新しく具体化されたビューのみを使用してクエリを再書き込み** DBMS 設定が自動的に発行され、新しいマテリアライズドビューのみを使用したクエリの再書き込みを有効にします。
- **具体化済みビューを使用してクエリを再書き込み** DBMS 設定が自動的に発行され、すべてのマテリアライズドビューを使用したクエリの再書き込みを有効にします。
- **クエリの再書き込みを無効化** DBMS 設定が自動的に発行され、クエリの再書き込みを明示的に無効にします。

デフォルト値は [インストールのデフォルト] です。

Oracle ストアドプロシジャに対するレポートの実行

SELECT ステートメントや TABLE コマンドなどのレポートツールを使用し、Oracle ストアドプロシジャを実行して、プロシジャの出力パラメータおよびアンサーセットに対するレポートを実行することができます。ストアドプロシジャの実行という方法には、次の利点があります。

- アンサーセット以外に、出力パラメータである OUT パラメータおよび OUT モードの INOUT パラメータを検索することができる。他の呼び出し方法では、アンサーセットのみが検索されます。
- TABLE および他のレポートツールを使用して、出力パラメータおよびアンサーセットの処理、フォーマット設定、表示を簡単に行うことができる。

ストアドプロシジャに対するレポートを実行するには、次の手順を実行します。

1. **シノニムの作成** ストアドプロシジャのアンサーセットのシノニムを作成します。詳細は、[ストアドプロシジャのシノニムの作成](#)を参照してください。
2. **レポートプロシジャの作成** レポートプロシジャを作成します。詳細は、[ストアドプロシジャに対するレポートの実行](#)を参照してください。
3. **レポートの実行** ストアドプロシジャを実行し、出力パラメータ (OUT および OUT モードの INOUT) およびレポートに指定されたアンサーセットフィールドに対するレポートを実行します。

ストアドプロシジャのシノニムの作成

シノニムは、ストアドプロシジャのパラメータとアンサーセットを記述します。

アンサーセットの構造は、プロシジャの実行時に指定される入力パラメータ値により異なります。このため、プロシジャの実行時に指定される入力パラメータ値のセットごとに、個別のシノニムを作成する必要があります。たとえば、3つの異なる入力パラメータ値セットでストアドプロシジャを実行する場合、シノニムを3つ (値セットごとに1つ) 作成する必要があります。なお、特に明記していない限り、「入力パラメータ」は、IN パラメータおよび IN モードの INOUT パラメータを意味します。

例外 プロシジャの内部ロジックが分かっており、プロシジャから返される各アンサーセット構造を生成する入力パラメータ値の範囲に確信がある場合、アンサーセット構造ごとにシノニムを1つ作成し、個々のシノニムにアンサーセット構造を返すために必要な入力パラメータ値の見本のセットを指定します。

シノニムには、次のセグメントを含めます。

- **INPUT** - このセグメントは、IN パラメータおよび IN モードの INOUT パラメータを記述します。
IN パラメータおよび IN モードの INOUT パラメータがない場合、このセグメントはダミーフィールドを1つ記述します。
- **OUTPUT** - このセグメントは、OUT パラメータおよび OUT モードの INOUT パラメータを記述します。
OUT パラメータおよび OUT モードの INOUT パラメータがない場合、このセグメントは省略されます。
- **ANSWERSET n** - 各アンサーセットに1つ含まれます。
アンサーセットがない場合、このセグメントは省略されます。

ストアドプロシジャのシノニム作成パラメータ

パラメータ/作業	説明
オブジェクトタイプ	[ストアドプロシジャ] を選択します。
選択	プロシジャを選択します。次のシノニム作成ウィンドウの [値] テキストボックスで、プロシジャごとに固有の値を入力する必要があるため、1 回に選択できるプロシジャは、1 つだけです。
名前	シノニム名です。デフォルト値はストアドプロシジャの名前です。
アプリケーション	アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。
接頭語/接尾語	同一名のストアドプロシジャが存在する場合は、接頭語または接尾語を割り当てて、対応するシノニムを区別します。なお、シノニム名の最大文字数は 64 バイトです。 すべてのプロシジャに固有の名前が付いている場合は、[接頭語] および [接尾語] のテキストボックスは空白にしておきます。
既存のシノニムを上書きする	作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。
データタイプマッピングのカスタマイズ	データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。詳細は、 Oracle データタイプのサポート を参照してください。

パラメータ/作業	説明
値	指定したプロシジャに表示されるパラメータすべてについて、このチェックボックスを選択します。 選択したプロシジャに入力パラメータ (IN パラメータ、IN モードの INOUT パラメータ) がある場合、それらのパラメータの入力が要求されます。ただし、[値] テキストボックスに明示的に入力する必要があるかどうかは、プロシジャのロジックおよび生成されるデータ構造に依存します。このため、パラメータボックスのチェックはオンにする必要がありますが、値の入力は不要な場合もあります。次のガイドラインに従います。 • 入力パラメータ値によりアンサーセットのデータ構造が変化する場合は、明示的な入力値 (によりシノニムを区別すること) が必要です。 • プロシジャの内部ロジックが分かっており、常に同一のデータ構造が生成されることが確定している場合は、明示的な入力値は必要ありません。この場合、生成する必要のあるシノニムは 1 つだけで、シノニム作成の目的では、[値] テキストボックスはブランクにすることができます。 [値] が必要な場合は、引用符を含めずに値を入力します。日付、日付時間、タイムスタンプのパラメータ値は、ISO の形式に従って入力します。明示的な値が必要なプロシジャの場合、入力パラメータにはプロシジャが実行される際に入力されるパラメータと同一の値を指定します。

ストアドプロシジャに対するレポートの実行

ストアドプロシジャのアンサーセットに対してレポートを実行する場合は、データベーステーブルに対してレポートを実行する際と同一の機能を使用することができます。

- **SQL SELECT ステートメント** 構文についての詳細は、[SELECT を使用したストアドプロシジャに対するレポートの実行](#)を参照してください。
- **TABLE および GRAPH コマンド** 構文についての詳細は、[TABLE コマンドを使用したストアドプロシジャに対するレポートの実行](#)を参照してください。

ストアドプロシジャのアンサーセットとの JOIN には、次の 2 つのファイルを使用します。

- **ホストファイル** OUTPUT および ANSWERSET セグメントのみです。
- **クロスリファレンス** ファイル内の INPUT セグメントのみです。

TABLE コマンドを使用したストアドプロシジャに対するレポートの実行

TABLE コマンドを使用してストアドプロシジャを実行するには、次の構文を使用します。

```
TABLE FILE synonym
PRINT [parameter [parameter] ... | *]
[IF in-parameter EQ value]
.
.
.
END
```

説明

synonym

実行するストアドプロシジャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアドプロシジャにパラメータを必要としない場合は、アスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

★

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

IF

IF または WHERE キーワードです。IN パラメータまたは IN モードの INOUT パラメータに値を渡す際に使用します。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

注意：アダプタが Unicode をサポートするよう構成されている場合は、IN パラメータの長さは 1000 バイト以内にする必要があります。

value

パラメータに渡す値です。

SELECT を使用したストアドプロシジャに対するレポートの実行

```
SQL
SELECT [parameter [,parameter] ... | *] FROM synonym
[WHERE in-parameter = value]
.
.
.
END
```

説明

synonym

実行するストアドプロシジャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアドプロシジャにパラメータを必要としない場合は、構文にアスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

★

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

WHERE

IN パラメータまたは IN モードの INOUT パラメータに値を指定する場合に使用します。

各パラメータの値は、それぞれ異なる行に指定する必要があります。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

value

パラメータに渡す値です。

Oracle 環境のカスタマイズ

Oracle アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。ここでは、カスタマイズオプションの概要について説明します。

システム日付および時刻のソースの選択

SQL SELECT ステートメントが CURRENT_DATE、CURRENT_TIME、または CURRENT_TIMESTAMP を参照する際、Server (デフォルト設定) または Oracle DBMS から日付または

時刻の値を取得します。新しい SET DATETIME_PROCESS オプションにより、デフォルト設定を上書きすることができます。これは、自動パススルーを使用したレポートでのみ有効です。

システム日時のソースの選択

システム日時のソースを選択するには、次の構文を使用します。

```
SQL SET DATETIME_PROCESS=SERVER|DBMS  
END
```

説明

SERVER

日付および時刻の値は、Server から取得されます。これがデフォルト値です。

DBMS

日付および時刻の値は、Oracle DBMS から取得されます。

デフォルトのテーブルスペースの指定

SET DBSPACE コマンドを使用して、作成するテーブルのデフォルトのテーブルスペースを指定することができます。セッション中、アダプタはこれらのテーブルを SET DBSPACE コマンドによって識別される Oracle テーブルスペースに格納します。SET DBSPACE コマンドを使用しない場合、Oracle は接続ユーザのデフォルトのテーブルスペースを使用します。

DBSPACE の設定

```
ENGINE SQLORA SET DBSPACE tablespace
```

説明

SQLORA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

tablespace

データベースの有効なテーブルスペースです。

注意：このコマンドはテーブルサービスによって実行された CREATE FILE および HOLD FORMAT SQLORA リクエストにのみ影響します。パススルーの CREATE TABLE コマンドには影響しません。

インデックススペースのデフォルトのパラメータの上書き変更

SET IXSPACE コマンドを使用して、CREATE FILE および HOLD FORMAT コマンドによって暗黙的に作成されたインデックススペースのデフォルトのパラメータを上書き変更することができます。

IXSPACE の設定

```
ENGINE SQLORA SET IXSPACE
```

説明

SQLORA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

index-spec

インデックスのパラメータを定義する Oracle CREATE INDEX ステートメントの一部です。94 バイト以内の有効な Oracle インデックススペースパラメータで構成することができます。インデックススペースパラメータをデフォルト値にリセットするには、パラメータを指定せずに SET IXSPACE コマンドを実行します。

注意：このコマンドの詳しい説明は、Oracle のマニュアルを参照してください。

以下は、複数行にわたるロングフォームの SQL パススルー構文の例です。

```
ENGINE SQLORA
SET IXSPACE index-spec
END
```

たとえば、Oracle CREATE INDEX ステートメントの NOSORT、NOLOGGING、TABLESPACE の部分を指定するには、以下のコマンドを入力します。

```
ENGINE SQLORA
SET IXSPACE NOSORT NOLOGGING
TABLESPACE TEMP
END
```

注意：このコマンドは、CREATE FILE および HOLD FORMAT SQLORA コマンドによって実行された CREATE INDEX リクエストにのみ影響を与えます。パススルーの CREATE INDEX コマンドには影響しません。

```
ENGINE SQLORA SET IXSPACE TABLESPACE tablespace_name
TABLE FILE table_name
PRINT *
ON TABLE HOLD AS file_name FORMAT SQLORA
END
```

NONBLOCK モードの有効化

Oracle アダプタには、NONBLOCK モードで呼び出しを実行する機能があります。デフォルト設定は、BLOCK モードです。

この機能により、アダプタはクライアントリクエストに応答して、エンジン処理の待ち状態からクエリを取り消すことができます。待機状態は、通常、SQL 解析中、アンサーセットの 1 行目がアダプタへ配信できる状態になる前、または別のアプリケーションによりロックされたオブジェクトへのアクセスの待機中に起こります。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

NONBLOCK モードの有効化

```
ENGINE SQLORA SET NONBLOCK {0|n}
```

説明

SQLORA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

正の数値です。デフォルト値はゼロ (0) です。これは、アダプタが BLOCK モードで動作することを意味します。数値が 1 以上の場合、NONBLOCK の呼び出しが有効になります。この数値は、アダプタが次の項目を確認する際の待機時間間隔 (秒) を指定します。

- 検索が実行されたか。
- クライアントアプリケーションが検索のキャンセルを要求したか。
- WebFOCUS Reporting Server ブラウザインターフェースの [セッションの終了] ボタンが押されたか。

注意：通常の操作には、値に 1 または 2 を指定します。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、メニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

```
ENGINE SQLORA SET PASSRECS {ON|OFF}
```

説明

SQLORA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

ストアドプロシージャのパラメータの最大数の指定

SET SPMAXPRM コマンドを使用して、任意の Oracle ストアドプロシージャに関連付ける入力パラメータの最大数を指定することができます。

SPMAXPRM の設定

```
ENGINE SQLORA SET SPMAXPRM nnn
```

説明

SQLORA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nnn

このクライアントセッションで実行可能なストアードプロシジャに渡すことができるパラメータの最大数です。デフォルト値は 256 です。

Oracle 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)および[定数として定義された一時項目 \(DEFINE\) を転送するリクエストの最適化](#)を参照してください。

一時項目 (DEFINE) に NULL 値が含まれる場合のリクエストの最適化

SET OPTNOAGGR コマンドを使用して、アダプタの最適化に関する動作を微調整することができます。このコマンドは、何らかの理由で RDBMS に集計処理を渡したくない場合に使用します。その理由の一例として、集計データに NULL 値が含まれる場合があります。SET OPTNOAGGR コマンドを使用すると、アダプタは DBMS に集計処理を渡さずに SQL を生成します。結合およびソート処理は RDBMS で実行されますが、集計処理はサーバ側で内部的に実行されます。

DEFINE フィールドに NULL フィールドを使用する計算が含まれている場合は常に NULL が返されるため、これらの処理を SQL に変換して DBMS に渡すことはできません。そのような処理は FOCUS で処理する必要があります。

これには SET OPTIMIZATION OFF を設定します。

ただし、パフォーマンスを考慮した場合、AGGREGATION は FOCUS で実行して、DBMS に対してはオフロードの JOIN および SORT 処理を使用すると良いケースもあります。

強化された集計制御の設定

```
SQL SQLORA SET OPT {AGGR|NOAGGR}
```

説明

AGGR

アダプタは DEFINE フィールドの集計処理を RDBMS にオフロードします。これがデフォルト設定です。

NOAGGR

アダプタは SQL を生成しますが、集計処理を RDBMS に渡しません。結合およびソート処理は RDBMS で実行されますが、集計処理はサーバ側で内部的に実行されます。この設定は、RDBMS にオフロードされる SQL が少なかった旧バージョンの機能に基づいて記述されたアプリケーションの下位互換性を提供する目的で使用することもできます。たとえば、集計フィールドの計算に NULL データが含まれている場合、その NULL データは RDBMS の NVL() 関数では処理されませんでした。

集計リクエストの最適化

以下の例は、Oracle デモテーブル SCOTT.EMP SCOTT.DEPT に基づいています。

```
JOIN CLEAR
JOIN DEPTNO IN EMP TO DEPTNO IN DEPT TAG D AS J1
DEFINE FILE EMP
LOC_FLAG/I2 = IF LOC EQ 'CHICAGO' THEN 1 ELSE 0 ;
BONUS_NEW/D15.2 = ((SAL + COMM)/2) * .2 ;
BONUS_BASE/D15.2 = (SAL + COMM) ;
END
TABLE FILE EMP
SUM COMM SAL BONUS_NEW MAX.LOC
FST.LOC_FLAG BONUS_BASE
-*
BY DNAME
BY ENAME
-*
ON TABLE HOLD
END
-RUN
```

```
TABLE FILE HOLD
PRINT *
END
```

OPTIMIZATION ON の場合、デフォルトの OPT AGGR アダプタは SQL を生成します。COMM が NULL の場合、dbms において計算を実行して、レポートに 0 を返します。

```
18.01.35 AE SELECT T2."DNAME",T1."ENAME", SUM(T1."COMM"), SUM(T1."SAL"),
18.01.35 AE SUM((((T1."SAL" + T1."COMM") / 2) * .2)), MAX(T2."LOC"),
18.01.35 AE MIN((CASE (T2."LOC") WHEN 'CHICAGO' THEN 1 ELSE 0 END)),
18.01.35 AE SUM((T1."SAL" + T1."COMM")) FROM EMP T1,DEPT T2 WHERE
18.01.35 AE (T2."DEPTNO" = T1."DEPTNO") GROUP BY T2."DNAME",T1."ENAME"
18.01.35 AE ORDER BY T2."DNAME",T1."ENAME";
```

この動作は、以下の 2 つの方法で変更することができます。

- SET OPTIMIZATION OFF を使用して、DBMS では基本的な SQL を使用して、FOCUS では最大限の処理を実行するようにアダプタに指示します。

または

- SET OPT NOAGGR を使用して、NULL の計算を避けるため、DBMS に JOIN を渡して FOCUS に集計を実行させます。生成された SQL では、パフォーマンスを改善するため Oracle に JOIN を渡します。

```
18.02.36 AE SELECT T1."EMPNO",T1."ENAME",T1."SAL",T1."COMM",T1."DEPTNO",
18.02.36 AE T2."DEPTNO",T2."DNAME",T2."LOC" FROM EMP T1,DEPT T2 WHERE
18.02.36 AE (T2."DEPTNO" = T1."DEPTNO") ORDER BY T1."EMPNO",T2."DEPTNO";
```

検索処理でのブロックサイズの指定

Oracle アダプタは、SELECT クエリまたはストアドプロシジャの実行によって生成された、結果セットからの配列検索をサポートします。この手法により、ネットワークおよび CPU 使用の負荷を大幅に軽減することができます。

大きい値を指定すると仮想記憶域のコストは増えますが、大量の行を処理するリクエストの効率は改善されます。ただし、100 を超える値を指定しても僅かな効率改善しか期待できないため、100 以内の値を設定することをお勧めします。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取

得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

配列検索のブロックサイズの指定

SELECT リクエストのブロックサイズは、TABLE FILE リクエスト、MODIFY リクエスト、MATCH リクエスト、および DIRECT SQL SELECT ステートメントに適用されます。

```
ENGINE SQLORA SET FETCHSIZE n
```

説明

SQLORA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

配列検索手法を使用して一度に取得する行数です。受容値は 1 から 5000 です。アダプタに応じてデフォルト値は異なります。CLOB または BLOB として処理する必要のあるフィールドが結果セットに含まれている場合は、その結果セットに使用する FETCHSIZE 値は 1 です。

挿入処理のブロックサイズの指定

INSERT のブロックサイズは、LOADONLY と組み合わせて MODIFY INCLUDE リクエストに適用されます。INSERTSIZE は、パラメータ化 DIRECT SQL INSERT ステートメントでもサポートされます。

```
ENGINE SQLORA SET INSERTSIZE n
```

説明

SQLORA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

配列検索手法を使用して挿入する行数です。受容値は 1 から 5000 です。デフォルト値は 1 です。BLOB として処理する必要のあるフィールドが結果セットに含まれている場合は、その結果セットに使用する INSERTSIZE 値は 1 です。

集計の最適化による効率の改善

集計の最適化により、クエリの効率が大幅に改善されます。

この機能についての詳細は、[集計の最適化のサポート](#)を参照してください。

集計の最適化の設定

```
SET AGGREGATE_AWARENESS {FRESH_ONLY|OLD_OK|OFF}
```

説明

FRESH_ONLY

各 RDBMS に関連するパラメータに異なる値を設定します。

OLD_OK

各 RDBMS に関連するパラメータに異なる値を設定します。

OFF

オプションを選択しない場合、ターゲット RDBMS の動作はデータベース構成オプションに基づいて決定されます。この設定のデフォルト値はありません。

アダプタ固有の設定の詳細は、[集計最適化の使用時の注意](#)を参照してください。

非等価条件 WHERE ベース LEFT OUTER JOIN の最適化

LEFT OUTER JOIN は、ホストテーブルからすべてのレコードを選択し、それらをクロスリファレンステーブルのレコードと照合します。一致するレコードが存在しない場合、ホストレコードは保持されますが、クロスリファレンスフィールドにはデフォルト値 (ブランクまたはゼロ) が割り当てられます。アダプタは、RDBMS で条件式がサポートされるすべての WHERE ベースの LEFT OUTER JOIN コマンドを最適化することができます。

条件付き LEFT OUTER JOIN の指定

```
JOIN LEFT_OUTER FILE hostfile AT hfld1 [TAG tag1]
    [WITH hfld2]
    TO {UNIQUE|MULTIPLE}
    FILE crfile AT crfld [TAG tag2] [AS joinname]
    [WHERE expression1;
    [WHERE expression2;
    ...]

END
```

説明

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。

hostfile

ホストマスタファイルです。

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 パラメータにフィールド値を指定しても、この値が接続情報として使用されることはありません。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

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に一意の修飾子として使用するタグ名 (オプション) です。

WITH 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 に基づく条件付き JOIN では、KEEPDEFINES を ON に設定し、一時項目 (DEFINE) を JOIN コマンドを発行する前に作成しておく必要があります。

MULTIPLE

from_file と *to_file* 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。なお、ONE は UNIQUE の同義語です。

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。

ユニーク JOIN は WebFOCUS の概念です。RDBMS はユニークと非ユニークを区別しないため、クロスリファレンスファイルから条件に一致するすべての行を取得します。

たとえば、リクエストでユニーク JOIN として指定したが、実際には条件に一致する行がクロスリファレンスファイルに複数存在する場合、この JOIN の処理を RDBMS が行くと、RDBMS は一致するすべての行を返します。一方、最適化を無効にしてこの JOIN の処理を WebFOCUS が行くと、異なるレポート結果が得られます。WebFOCUS はユニーク JOIN の概念を順守するため、ホストの 1 行に対してクロスリファレンスの行を 1 行だけ返します。

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データソース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。

WHERE ベース OUTER JOIN の最適化条件

- WHERE ベースの LEFT OUTER JOIN を最適化するには、それに関係する RDBMS で式の最適化を行うことができるとともに、次の条件のいずれかを満たす必要があります。
 - JOIN WHERE コマンドに、少なくとも 1 つの「field1 EQ field2」述語が含まれている。ここで、field1 は table1 のフィールド、field2 は table2 のフィールドを表します。
または
 - 右側のテーブルに、NULL データを含まないユニークインデックスまたはキーがある。
または
 - 右側のテーブルに、長いデータタイプ (例、TEXT、IMAGE) を持たない「NOT NULL」フィールドが少なくとも 1 つ含まれている。
- アダプタの SQLJOIN OUTER 設定は、ON (デフォルト) に設定しておく必要があります。

非等価条件 LEFT OUTER JOIN の最適化

次のリクエストは、2つの Oracle データソース間の条件付き LEFT OUTER JOIN を作成し、結合後のデータソースを使用してレポートを作成します。このリクエストに対して生成された SQL を表示するため、STMTRACE がオンに設定されています。

```
SET TRACEUSER = ON
SET TRACEOFF = ALL
SET TRACEON = STMTRACE//CLIENT
JOIN LEFT_OUTER FILE baseapp/EQUIP AT CARS
TO ALL FILE baseapp/CARREC AT CARC
WHERE CARS NE CARC;
END
TABLE FILE baseapp/EQUIP
PRINT CARS CARC STANDARD
BY MODEL
END
```

WebFOCUS リクエストは、LEFT OUTER JOIN を組み込んだ単一の Oracle SELECT ステートメントに変換され、非等価条件が ON 句で RDBMS に渡されます。

```
SELECT T1."CARS"(CHAR( 16)),T1."STANDARD"(CHAR( 40)),
T2."CARC"(CHAR( 16)),T2."MODEL"(CHAR( 24)) FROM ( EQUIP T1 LEFT
OUTER JOIN CARREC T2 ON (T1."CARS" <> T2."CARC") ) ORDER BY
T2."MODEL";
```

ヒントによるオプティマイザ効率の改善

DBMS オプティマイザのヒントを使用して、実行プランを変更することができます。アダプタには、TABLE コマンドの実行時に Oracle に送信される SELECT キーワードの後にヒントを追加する設定があります。

ヒントは、アダプタが単一の SELECT ステートメントを作成する際に追加されます。FOCUS 管理の JOIN で複数の SELECT ステートメントが生成される場合、ヒントは追加されません。

この設定を元に戻すには、hint_text パラメータを省略して SET HINT を使用します。

ヒントの設定

特定のヒントを設定するには、次の構文を使用します。

```
SQL SQLORA SET HINT /* +hint_text */
```

説明

SQLORA

ターゲットの RDBMS です。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

+hint_text

単一ヒントのテキスト、または複数のヒントを組み合わせたテキストです。構文が無効な場合に Oracle からエラーメッセージは返されません。オブティマイザは、単純にヒントを無視します。ユーザは、構文を正確に入力する必要があります。+*hint_text* を省略すると、ヒントを追加しない設定に戻ります。

Oracle ヒントの組み合わせ設定

```
SQL SQLORA SET HINT /* +USE_HASH PARALLEL(EMPNO) INDEX_ASC */
TABLE FILE EMP
  PRINT EMPNO SAL BY DEPTNO
  IF DEPTNO GE 5
END
```

上記の WebFOCUS リクエストは、ヒントの組み合わせが追加された Oracle SELECT ステートメントに変換されます。

```
SELECT
  /* +USE_HASH PARALLEL(EMPNO) INDEX_ASC */
  T1."EMPNO",
  T1."SAL",
  T1."DEPTNO"
FROM
  SCOTT.EMP T1
WHERE
  (T1."DEPTNO" >= 5)
```

```
ORDER BY  
T1."DEPTNO";
```

SQL パススルーによる Oracle ストアドプロシージャの呼び出し

Oracle ストアドプロシージャは、SQL パススルーの使用をサポートします。これらのプロシージャは、CREATE PROCEDURE コマンドを使用して Oracle 内で開発する必要があります。

注意：SQL パススルーを使用したストアドプロシージャの呼び出しでは、1 回の呼び出しで処理可能なアンサーセットは 1 つだけです。複数のアンサーセットが预期される場合は、CREATE SYNONYM による方法をお勧めします。

ストアドプロシージャの呼び出し

```
SQL SQLORA EX procname [parameter_specification1]  
[,parameter_specification2]...  
END
```

説明

SQLORA

Oracle の ENGINE 接尾語です。

procname

ストアドプロシージャ名です。ネイティブデータベース構文のストアドプロシージャの完全 (または部分) 修飾名です。

parameter_specification

IN、OUT、INOUT がサポートされています。ストアプロシージャにより異なる適切なパラメータを指定します。

IN

リテラル (例、125、3.14、'abcde') です。入力文字に予約語を使用することも可能です。予約語は文字リテラルとは異なり、引用符で囲む必要はありません (例、NULL)。この値は必須です。

OUT

1 つの疑問符 (?) で表されます。このパラメータの有無により、出力結果をアプリケーションへ渡すかどうかを決定します。省略すると、この項目は (0 バイトの) 空白文字列になります。

INOUT

出力を表す 1 つの疑問符 (?) と入力を表すリテラルで構成されます。出力と入力はスラッシュ (/) で区切ります (例、?/125、?/3.14、?/'abcde')。出力値を (0 バイトの) 空白文字列にすることも可能です。

アダプタは、以下の規則に従うストアードプロシージャの呼び出しをサポートします。

- すべてのスカラパラメータ (IN、OUT、INOUT) がサポートされます。
- 以下においてカーソルを定義しなければなりません。
 - PACKAGE または PROCEDURE の TYPE ステートメント。
 - 返されるアンサーセットに関連するレコードレイアウト。
- プロシージャでカーソルを表示しなければなりません。
- ストアドプロシージャでのフェッチは許可されません。アダプタはアンサーセットをフェッチします。
- RAISE APPLICATION ERROR を使用して、メッセージを出力する必要があります。
- パラメータが RECORD として宣言されており、それらのパラメータが REFCURSOR に関連する場合、SQL パススルーを使用してストアードプロシージャを呼び出す際にこれらのパラメータを指定してはいけません。詳細は、[REFCURSOR を使用する Oracle ストアドプロシージャ](#)を参照してください。

ストアードプロシージャで生じるアプリケーションエラーは、サーバ変数 &ORAMSGTXT で使用することができます。

REFCURSOR を使用する Oracle ストアドプロシジャ

以下の Oracle ストアドプロシジャは、REFCURSOR を使用します。PL/SQL を使用して、SQL*Plus に REFCURSOR を作成しなければなりません。

```
sqlplus scott/tiger
set serveroutput ON
```

```
CREATE OR REPLACE PACKAGE pack1 AS
TYPE nfrectype IS RECORD (
employee NF29005.EMPLOYEE_ID5%TYPE,
ssn5      NF29005.SSN5%TYPE,
l_name    NF29005.LAST_NAME5%TYPE,
f_name    NF29005.FIRST_NAME5%TYPE,
birthday  NF29005.BIRTHDATE5%TYPE,
salary    NF29005.SALARY5%TYPE,
joblevel  NF29005.JOB_LEVEL5%TYPE);
TYPE nfcurtype IS REF CURSOR RETURN nfrectype ;
PROCEDURE proc1(c_saltable IN OUT nfcurtype);
END pack1 ;
/
sho error
```

```
CREATE OR REPLACE PACKAGE BODY pack1 AS
PROCEDURE proc1 (c_saltable IN OUT nfcurtype)
IS
BEGIN
OPEN c_saltable FOR SELECT EMPLOYEE_ID5,SSN5,LAST_NAME5,FIRST_NAME5,
BIRTHDAT
E5,SALARY5,JOB_LEVEL5 FROM NF29005;
END proc1 ; -- end of procedure
END pack1; -- end of package body
/
sho error
```

```
/*
Invocation using SQL*Plus:
VARIABLE c1 REFCURSOR
```

```
EXEC scott.pack1.proc1 (:c1)
PRINT c1

Invocation using SQL Passthru:
SQL SQLORA EX scott.pack1.proc1
TABLE FILE SQLOUT
END
*/
```

REFCURSOR を使用しない Oracle ストアドプロシジャ

以下の Oracle ストアドプロシジャは、REFCURSOR を使用しません。IN および OUT パラメータが含まれています。

```
sqlplus scott/tiger
set serveroutput ON
```

```
CREATE OR REPLACE PACKAGE pk1 AS
PROCEDURE ibi_pr1 (v_num IN NUMBER,v_rec IN OUT VARCHAR2);
END;
/
sho error
```

```
CREATE OR REPLACE PACKAGE BODY pk1 AS
PROCEDURE ibi_pr1 (v_num IN NUMBER,v_rec IN OUT VARCHAR2)
IS
    v_msg    VARCHAR2(25);
    v_to_chr VARCHAR2(5);
BEGIN
    v_to_chr := TO_CHAR(v_num);
    SELECT v_to_chr||' '|v_rec INTO v_msg FROM dual;
    v_rec := v_msg;
END ibi_pr1;
END pk1;
/v
sho error
```

```
/*  
Invocation using SQL*Plus:  
  VARIABLE v_rec VARCHAR2(20)  
  -- assign IN value to IN OUT parameter:  
  -- BEGIN :v_rec := 'Your message!...'; END;  
  BEGIN :v_rec := &1; END;  
  /  
  EXEC scott.pk1.ibi_pr1 (&2,:v_rec)  
  PRINT v_rec  
  --
```

```
Invocation using SQL Passthru:  
  SQL SQLORA EX scott.pk1.ibi_pr1 12345, ?/'AbCdEf';  
  END  
*/
```

PostgreSQL アダプタの使用

PostgreSQL アダプタの構成方法について説明します。

PostgreSQL アダプタを使用すると、アプリケーションから PostgreSQL データソースへのアクセスが可能になります。このアダプタにより、アプリケーションリクエストが PostgreSQL コールに変換され、最適化されたアンサーセットがリクエストの送信元アプリケーションに返されます。

注意

- バージョン 9.3.0 以降、PostgreSQL アダプタは PostgreSQL バージョン 15 の MERGE サポートを提供します。
- バージョン 9.3.0 以降、PostgreSQL アダプタは PostgreSQL バージョン 14 以前の UPSERT サポートを提供します。

PostgreSQL 環境の準備

PostgreSQL アダプタを使用するには、PostgreSQL ドライバをインストールし、サーバの起動前にその CLASSPATH 値を設定するとともに、JSCOM3 サービスを実行状態にしておく必要があります。

Windows および UNIX で環境を設定するには

手順

1. サーバを起動する前に、PostgreSQL ドライバファイルのパスを CLASSPATH 環境変数に追加することで、これらのファイルの格納先を設定します。
たとえば、UNIX で JDBC ドライバファイルのパスを /usr/driver_files/mydriver.jar に

設定するには、次のコマンドを発行します。

```
CLASSPATH=/usr/driver_files/mydriver.jar;$CLASSPATH
export CLASSPATH
```

Windows では、以下のコマンドを実行します。

```
set CLASSPATH=c:¥usr¥driver_files¥mydriver.jar:%CLASSPATH%
```

サーバのワークスペースを Windows サービスとして実行する場合、CLASSPATH をシステム全体の環境変数として設定する必要があります。

同様に UNIX では、サーバを起動する前に変数を設定して、UNIX プロファイルに CLASSPATH の設定を追加します。

2. サーバを起動または再起動します。
ドライバを変更した場合にもサーバを再起動する必要があります。
3. サーバのワークスペースで以下のコマンドを実行することにより、JSCOM3 サービスが有効であることを確認します。

```
edastart -show
```

また、[スペシャルサービス] セクションに「JSCOM3 active」と表示されていることを確認します。

JSCOM3 サービスが有効ではない場合、サーバの EDAPRINT ログにメッセージ “fail to start” が出力されます。

PostgreSQL アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

アダプタは、PostgreSQL データソースに接続する際に、接続情報および認証情報を必要とします。

この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。
WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスク

トップインターフェースの構成画面で、接続および認証情報を入力します。選択したサーバプロファイルにコマンドが追加されます。これらのプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用するプラットフォームでサポートされる場合) があります。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の PostgreSQL データソースへの接続を宣言することができます。実際の接続は、最初のクエリ発行時に確立されます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの

「接続属性」に記載されています。

4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

PostgreSQL の接続属性

PostgreSQL アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

URL

PostgreSQL データソースのロケーション URL です。

ドライバ名

PostgreSQL JDBC ドライバの名前です。

たとえば、「org.postgresql.Driver」のように入力します。

PostgreSQL のマニュアルを参照して、使用するドライバのバージョンを確認してください。

IBI_CLASSPATH

Java サービスで使用する追加の Java クラスディレクトリ、または jar ファイル名のフルパスを定義します。値を設定するには、通信ファイルを編集するか、WebFOCUS

Reporting Server ブラウザインターフェースを使用します。WebFOCUS Reporting Server ブラウザインターフェースの入力ボックスでは、1 行につき 1 つの参照を入力することができます。すべてのプラットフォームでは、ファイルを保存した際に、複数のエントリがコロン (:) で区切られた単一文字列に変換されます。ファイルを手動で編集する際は、区切り文字のコロン (:) を保持する必要があります。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

接続属性の手動宣言

```
ENGINE SQLPSTGR SET CONNECTION_ATTRIBUTES connection
'URL'/userid,password
```

説明

SQLPSTGR

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

接続名です。

URL

PostgreSQL データソースのロケーション URL です。

userid

ターゲットデータベースに登録されたプライマリ認証 ID です。

password

プライマリ認証 ID に関連するパスワードです。

接続属性の宣言

以下の SET CONNECTION_ATTRIBUTES コマンドは、明示的なユーザ ID (MYUSER) およびパスワード (PASS) を使用して、PostgreSQL ドライバを介してデータソースにアクセスします。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

注意：正確な名前、ポート番号、パスについての詳細は、ベンダーのマニュアルを参照してください。

```
ENGINE SQLPSTGR SET CONNECTION_ATTRIBUTES CON1  
'jdbc:postgresql://hostname:5432/qatst'
```

デフォルト接続の上書き変更

複数の接続を定義している場合、最初の SET CONNECTION_ATTRIBUTES コマンドに指定した接続が、デフォルトの接続として設定されます。

デフォルト接続の変更

```
ENGINE SQLPSTGR SET DEFAULT_CONNECTION connection
```

説明

SQLPSTGR

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

```
ENGINE SQLPSTGR SET DEFAULT_CONNECTION SAMPLE
```

接続範囲の制御

確立する接続ごとにアダプタを使用する場合、SET AUTODISCONNECT コマンドを使用して接続の継続を制御することができます。

接続範囲の制御

```
ENGINE SQLPSTGR SET AUTODISCONNECT ON {FIN|COMMIT}
```

説明

SQLPSTGR

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

PostgreSQL メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および JDBC データタイプのサーバマッピングを定義するシノニムを作成します。

アダプタの識別

リクエストの解釈に必要なアダプタを識別するには、マスターファイルの SUFFIX 属性を使用します。PostgreSQL アダプタを識別するには、SUFFIX 値として SQLPSTGR を使用します。

アダプタの識別

```
FILE[NAME]=file, SUFFIX=SQLPSTGR [, $]
```

説明

file

マスターファイルのファイル名です。このファイル名は文字で開始する必要があります。また、テーブルまたはビューの内容を表す名前にするをお勧めします。実際のファイルには拡張子 .mas を付ける必要がありますが、この属性の値には拡張子を含めないでください。

SQLPSTGR

アダプタの値です。

データベーステーブルへのアクセス

Postgre SQL を使用してリモートのサードパーティのテーブルにアクセスする場合、RDBMS の PostgreSQL ドライバをローカルマシンにインストールする必要があります。

サーバは、PostgreSQL ネットワーク上のサードパーティのデータベーステーブルにアクセスすることが可能です。データソースの URL を指定する必要があります。また、必要に応じて、アクセスするデータベースのユーザ ID およびパスワードを指定します。これらのパラメータは、サーバのグローバルプロファイル、またはユーザプロファイルのいずれかで定義できます。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な各オブジェクトの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの

再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタ

では、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

PostgreSQL のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

外部 SQL スクリプトのパス

[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合は、さらに次の項目が表示されます。

次の標準名前規則は UNIX に適用されます。

- [ベースパス] テキストボックスに、SQL クエリが記述されたファイルの物理ディレクトリパスを指定します。ディレクトリ名を直接入力するか、[...] (参照) ボタンをクリックしてファイルの格納先ディレクトリへ移動します。このクリックにより、

[ベースパスの選択] ダイアログボックスが表示されます。

- [ドキュメント名] テキストボックスに、ファイル名を入力します。必要に応じて、ワイルドカード文字を使用します。
- [ドキュメント拡張子] テキストボックスに、スクリプトファイルの拡張子を入力して、その拡張子のファイルのみを表示します。

アダプタはシノニムの作成中にネイティブ API コールを発行して選択リスト内の要素のリストを取得し、各要素に対するフィールドを持つマスターファイルを構成します。生成されたアクセスファイルは、DATASET 属性の SQL スクリプトパスを参照します。これには SQL クエリが格納されたファイルの名前と拡張子を含む、フルパスが記述されています。以下はその例です。

```
DATASET=/u1/home2/apps/report3.sql
```

WebFOCUS レポートを作成する際は、SQL クエリを使用してデータにアクセスします。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

外部キーを使用したクラスタ作成 (非推奨。[シノニムの編集] の使用を推奨)

[外部キーを使用したクラスタ作成] のチェックをオンにすると、現在のテーブルに外部キーで関係付けられているテーブルすべてをこのシノニムに含めることができます。ただし、このオプションは廃止される予定で、クラスタの作成にはシノニムエディタを使用することをお勧めします。生成されるマルチテーブルのシノニムは、このテーブルの外部キーの関係をすべて定義します。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニム

を JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (:) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[PostgreSQL データタイプのサポート](#)を参照してください。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

オーナー/スキーマ

ユーザが所有する 1 つまたは複数のオブジェクトを作成したユーザアカウントです。

テーブル名

基になるオブジェクトの名前です。

タイプ

オブジェクトタイプ (テーブル、ビューなど)。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

生成されたシノニムの例

PostgreSQL アダプタのシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

マスターファイル nf29004.mas

```
FILE=DIVISION, SUFFIX=SQLPSTGR , $
SEGNAME=SEG1_4, SEGTYPE=S0 , $
FIELD=DIVISION4, DIVISION4, I9, I4, MISSING=OFF , $
FIELD=DIVISION_NA4, DIVISION4, I9, I4, MISSING=OFF , $
FIELD=DIVISION_HE4, DIVISION4, I9, I4, MISSING=OFF , $
```

アクセスファイル nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=EDAQA.NF29004,
CONNECTION=CON1, KEYS=1, WRITE=YES, $
```

アクセスファイルのキーワード

下表は、アクセスファイルで使用されるキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABlename	PostgreSQL テーブルを識別します。この属性に割り当てる値には、次のようにオーナー (スキーマとも呼ばれる) の名前とデータベースリンク名を含めることができます。 TABlename=[owner.]table
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 CONNECTION=connection CONNECTION=' ' は、ローカルのデータソースへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。

キーワード	説明
	<pre>KEY=fld1/fld2/.../fldn</pre>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。
KEYFLD IXFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定します。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 • KEYFLD は、親テーブルの共通カラムの FIELDNAME です。 • IXFLD は関連するテーブルの共通カラムの FIELDNAME です。 KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることをお勧めしますが、必須ではありません。 注意：KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

PostgreSQL データタイプのサポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[アクセスファイル](#)を参照してください。

A256V のデータタイプマッピング

PostgreSQL アダプタは、次の設定を使用してデータタイプマッピングをサポートします。

```
ENGINE SQLPSTGR SET CONVERSION LONGCHAR ALPHA n
```

この設定は、データタイプが [VAR]CHAR(32767) または TEXT の PostgreSQL ネイティブフィールドすべてのマッピングに反映されます。n は、1 から 32767 までの値です。デフォルト値は 256 です。必要に応じて、ALPHA の代わりに、TEXT データタイプ (長さ n の指定なし) を設定することもできます。

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアドプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

PostgreSQL 環境のカスタマイズ

PostgreSQL アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。

タイムアウト制限の指定

PostgreSQL に対して SQL リクエストを実行した後、アダプタがその応答を待機する時間(秒)を指定するには、TIMEOUT コマンドを使用します。

TIMEOUT コマンドの発行

```
ENGINE SQLPSTGR SET TIMEOUT {nn|0}
```

説明

SQLPSTGR

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nn

タイムアウトとなる秒数を指定します。デフォルト値は 30 です。

0

応答の待機時間が無制限であることを示します。

長時間実行中リクエストのキャンセル

WebFOCUS Reporting Server ブラウザインターフェースで、長時間実行中のリクエストをキャンセルすることができます。この操作を実行すると、ネイティブ JDBC ドライバの機能に応じて、リクエスト全体がキャンセルされるか、フェッチサイクルが中断されます。

長時間実行中のリクエストをキャンセルするには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [ワークスペース] ページで、[Java サービス] 下の [DEFAULT] を選択します。[DEFAULT] を右クリックし、[エージェント] を選択します。
2. [Java サービスエージェント] ウィンドウで、キャンセルする jscomid の行を選択し、右クリックして [終了] を選択します。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

```
ENGINE SQLPSTGR SET PASSRECS {ON|OFF}
```

説明

SQLPSTGR

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

PostgreSQL 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)を参照してください。

Python アダプタの使用

Python アダプタの構成方法および Python スクリプトの実行方法について説明します。

Python は、高度で使いやすく、強力なインタープリタ型プログラム言語で、スクリプトおよび複雑なプログラムに適しています。

幅広いモジュールを含む Python 標準ライブラリは、「バッテリーを含んだ」アプリケーションを実装しており、プログラマはインストール直後から洗練された強力な機能にアクセスし、WebFOCUS で使用する独自の Python 関数を簡単に記述することができます。

Python アダプタは、一時項目 (COMPUTE) (WebFOCUS COMPUTE) を作成するユーザ定義の Python スクリプトを実行するため、Python インタープリタへの接続を定義します。これらのフィールドは、WebFOCUS ブック、グラフ、レポート、ダッシュボード、および WebFOCUS ポータルで 사용할 ことができます。

WebFOCUS で使用する Python スクリプト記述に関するガイドライン

次の arithmetic_exapmle.py と名付けられた Python スクリプトには、2 つの数値を加算する ADDER という名前の関数が含まれ、このセクションで説明する要件に適合します。

```
# arithmetic_example.py

import csv

def adder(csvin, csvout):

    with open(csvin, 'r', newline='') as file_in, \
        open(csvout, 'w', newline='') as file_out:

        fieldnames = ['addition']

        reader = csv.DictReader(file_in, quoting=csv.QUOTE_NONNUMERIC)

        writer = csv.DictWriter(file_out, quoting=csv.QUOTE_NONNUMERIC,
```

```

                                fieldnames=fieldnames)
writer.writeheader()

for row in reader:
    addition = row['a_number'] + row['another_number']

    writer.writerow({'addition': addition})

```

Python スクリプトは、WebFOCUS で使用するためには次の要件に適合する必要があります。

- **csv モジュール要件** スクリプトは csv モジュールをインポートする必要があります。これは、自動的に生成された一時ファイル .csv を使用して WebFOCUS が Python スクリプトにデータを送信するためです。ファイル名は、csvin という名前の Python グローバル変数に保存されます。Python グローバル変数 csvout には、WebFOCUS に返される一時ファイル .csv の結果ファイルの名前が含まれます。WebFOCUS は、csvin および csvout の値を設定しますが、プログラマはこれらの値を変更するべきではありません。一時ファイルはすぐに削除されるため、ユーザは参照することができません。どちらのファイルもフォーマットは同じです。デフォルト設定の区切り文字はカンマ (,) で、変更することはできません。数値以外のフィールドはすべて、二重引用符 (") で囲みます。

グローバル変数 csvin および csvout は、.csv ファイルの読み取りおよび書き込みのために WebFOCUS で定義されます。

- **csvin および csvout のフォーマットパラメータ** スクリプトは、次のフォーマットパラメータで csvin および csvout を開く必要があります。

```
newline=''
```

newline="" が指定されていない場合は、引用符付きフィールド内に埋め込まれた改行が正確に解釈されず、書き込み時に行末に ¥r¥n を使用するプラットフォームでは、¥r が余分に 1 つ追加されます。

- **リーダおよびライタのフォーマットパラメータ** 次のフォーマットパラメータは、リーダおよびライタのいずれを使用するか、また DictReader および DictWriter のいずれを使用するかにかかわらず、リーダおよびライタのステートメントに含める必要があります。この場合、数値以外の値は二重引用符 (") で囲み、数値を明確に示す必要があります (これらは単精度浮動小数点値に変換されます)。

```
quoting=csv.QUOTE_NONNUMERIC
```

つまり、数値以外の値 (文字、テキスト、文字列、日付) は二重引用符 (") で囲まれ、数

値は囲まれません。csvin の読み取り時に、WebFOCUS の数値はすべて Python の単精度浮動小数点値に自動的に変換されます。WebFOCUS の一時項目 (COMPUTE) が、返されるフィールドを整数として定義する場合、小数部区切り文字および小数部は切り捨てられます。csvin ファイルでは、数値以外のフィールドに二重引用符 (") が 1 つだけ含まれる場合は、WebFOCUS がこれを 2 つにします。Python がこれを正確に解析するのは、Python の csv モジュールの Dialect.doublequote フォーマットパラメータが True にデフォルト設定されているためです。

- **出力** 関数から複数の出力フィールドを取得することができます。ただし、各 WebFOCUS COMPUTE コマンドが取得できるのは、単一の出力フィールドのみです。複数の出力フィールドを取得するには、複数の COMPUTE コマンドを発行します。Python 関数の呼び出しでは、出力引数 (最後の引数) が、Python スクリプトで生成されたシノニムの OUTPUT_DATA セグメントのフィールド名と一致します。

たとえば、次のマスターファイルでは、出力引数は ADDITION です。

```
SEGMENT=OUTPUT_DATA, SEGTYPE=U, PARENT=INPUT_DATA, $
  FIELDNAME=ADDITION, ALIAS=addition, USAGE=D7.1,
    ACTUAL=STRING,      MISSING=ON,      TITLE='Addition', $
```

したがって、Python の呼び出しで使用する最後の (出力) 引数は、必ず ADDITION になります。

```
COMPUTE Anyname/I5 = PYTHON(synonym_name, anyarg1, anyarg2, ADDITION);
```

csvout に書き込まれる出力は、単一フィールドの場合でもシーケンスになります (例、リスト)。

単一フィールドを含むリストの場合、正確な構文は次のようになります。

```
writer.writerow([result])
```

次の構文は不正確なため、文字列では不正確な値が返され、数値フィールドでは例外が発生します。

```
writer.writerow(result)
```

- **関数** Python スクリプトには、1 つまたは複数のユーザ定義の関数を含めることができます。Python スクリプトのメタデータオブジェクト (シノニム) を作成する場合、これらの関数のいずれか 1 つは、スクリプト実行時の開始点として選択する必要があります。開始点として使用されるユーザ作成関数の定義には、先頭位置の引数として csvin および csvout が必ず含まれます。

注意：Python スクリプトはインポートされるため、次の Python プログラム言語のイディオムは無視されます。

```
if __name__ == '__main__':
```

ただし、WebFOCUS 以外のテストでは、上記のイディオムを含めることが有効な場合もあります。

- **ヘッダ** Python スクリプトのシノニム作成時、または Python スクリプトへのデータ送信時および Python スクリプトからのデータ取得時のサンプル入力データファイルでは、フィールド名を示すヘッダレコードの使用 (位置を示すインデックス番号を使用する代わり) は必要ありません。ただし、Python スクリプトにヘッダレコードつまりフィールド名を使用した方が理解しやすくなります。次の構文は、ヘッダを使用した例です。csv.DictReader ステートメントで、ヘッダレコードの使用が暗示されています。

```
fieldnames = ['addition']
reader = csv.DictReader(file_in, quoting=csv.QUOTE_NONNUMERIC)
writer = csv.DictWriter(file_out, quoting=csv.QUOTE_NONNUMERIC,
                        fieldnames=fieldnames)
writer.writeheader()
```

入力および出力 .csv ファイルでは、ヘッダレコードを使用することをお勧めします。ヘッダなしでサンプルデータを使用した場合、生成されたメタデータのフィールド名は、FIELD_1 から FIELD_n の形式になります。

Python アダプタには定義済みの Python 統計関数も実装されており、WebFOCUS で簡単に呼び出すことができます。

Python アダプタの使用要件

要件の完全なリストにアクセスするには、[データの取得] ページで Python アダプタのアイコンを右クリックし、[要件] を選択します。

注意：バージョン 9.3.0 以降、Python アダプタは z/OS プラットフォームではサポート対象外になります。このアダプタは、Windows x64 および Linux Intel x64 プラットフォームのみでサポートされます。また、インストール済みの Python リリースは提供されなくなりました。そのため、これらのプラットフォームに Python を別途インストールする必要があります。Python でサポートされるリリースは、3.9.x のみです。Python をインストールする方法については、[要件] ページに記載されています。

Python アダプタを構成する前に、次の操作を実行する必要があります。

- WebFOCUS Reporting Server と同じマシンに、同じビットサイズの Python をインストールします。
- Python をインストール後、次のパッケージをインストールまたは更新します (詳細は [要件] ページを参照)。
 - cyclcr
 - fonttools
 - joblib
 - kiwisolver
 - matplotlib
 - memory-profiler
 - numpy
 - packaging
 - pandas
 - patsy
 - Pillow
 - pip
 - psutil
 - pyparsing
 - python-dateutil
 - pytz
 - scikit-learn
 - scipy
 - seaborn
 - setuptools
 - six
 - statsmodels
 - threadpoolctl

- wheel
- xgboost

注意：これらのパッケージの中には、追加のパッケージが自動的にインストールされるものがあります。

- 必須のシステム変数を設定します。これらの変数についての情報は、このアダプタの [要件] ページに記載されています。

Python アダプタの構成

アダプタの構成では、Python インストールディレクトリを特定します。

Python アダプタを構成するには

アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで構成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックするか、ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、または ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。

- アダプタ名前を右クリックして、[構成 Python 3.9] を選択します。
[PYTHON Python 3.9 の構成の追加] ウィンドウが開きます。
- Python インストールディレクトリにパスを入力し、[テスト] をクリックします。
構成が完了すると、次のメッセージが表示されます。

```
Successful test for the Python environment
```

- [構成] をクリックします。
- [ワークスペース] の [サーバアクション] から、WebFOCUS Reporting Server を再起動します。

構成したアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

- アダプタの構成後、プロパティを確認、変更するには、このアダプタを右クリックして [プロパティ] を選択します。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。

Python 関数のシノニムの作成

Python アダプタで使用する Python スクリプトにはそれぞれ、スクリプトの入力フィールドおよび出力フィールドを記述するシノニム (メタデータオブジェクト) が含まれます。Python スクリプトに複数のユーザ作成関数が含まれ、スクリプト内で複数の関数を開始点として使用できるようにしたい場合は、スクリプト内の関数ごとに別個のシノニムを作成する必要があります。

シノニムは、スクリプトの入力パラメータとなるフィールドのみを含むサンプルファイルを使用して作成されます。数行のサンプルデータがあれば、Python アダプタは適切なデータタイプおよびパラメータの長さを指定することができます。サンプルファイルは常に .csv ファイルです。このファイルのデータに実データは必要ありませんが、実データに表示される数値フィールドには最大値、文字値には最大長を示す必要があります。マスターファイルには、入力フィールドと出力フィールドのリストが含まれます。アクセスファイルには、スクリプトファイルおよびサンプル入力ファイルに関する情報が含まれます。

Python 関数のシノニムを作成するには

手順

1. 構成済みアダプタのリストで [PYTHON] を右クリックし、コンテキストメニューから [メタデータオブジェクトの作成] を選択します。

下図のように、[Python シノニムの作成] ウィンドウが開きます。

メタデータオブジェクトとシノニムは同義語です。Python 関数のシノニムは、マスターファイル (関数の実行に必要な入力フィールドと出力フィールドを記述) およびアクセスファイル (サンプルデータファイルおよびスクリプトファイルに関する情報を格納) で構成されます。

2. 次のパラメータの値を入力または選択します。

Python スクリプト

アプリケーションディレクトリ名およびスクリプト名を入力するか、[...] (参照) ボタンをクリックしてアプリケーションディレクトリを指定し、スクリプトを選択後 [OK] をクリックします。Python スクリプトの拡張子は .py です。

関数名

シノニムを作成するスクリプトファイルの (開始点の) 関数の名前を選択します。

たとえば、arithmetic_example.py という名前の Python スクリプトには、ADDER 関数の定義が含まれます。

```
def adder(csvin, csvout):
```

Python スクリプトのサンプル入力データファイル

[...] (参照) ボタンをクリックし、シノニム作成のためのサンプルデータを含むアプリケーションディレクトリおよびファイルを選択します。[OK] をクリックします。

このファイルは、フィールド名、データタイプ、および csvin で Python スクリプトに送られるデータの長さを指定するために使用します。サンプルファイルにヘッダレコードが含まれない場合は、フィールド名が FIELD_1 から FIELD_n で表示されます。

ヘッダ付き CSV ファイル

入力 .csv ファイルにヘッダ行が含まれない場合は、[入力] のチェックをオフにします。出力 .csv ファイルにヘッダ行が含まれない場合は、[出力] のチェックをオフにします。ヘッダ要件は、関数コードに含まれます。

たとえば、次のサンプルコードでは、入力ファイルにヘッダレコード (フィールド名) が含まれず、フィールド名オブジェクトは、リーダのフィールド名引数を使用して実行時に定義されます。出力ファイルには、ライタのフィールド名引数で定義されたヘッダレコードが含まれ、writer.writeheader() ステートメントを使用してファイルに書き込まれます。

```
with open(csvin, 'r', newline='') as file_in, \
    open(csvout, 'w', newline='') as file_out:
    reader = csv.DictReader(file_in, fieldnames=['input_field'],
                           quoting=csv.QUOTE_NONNUMERIC)
    writer = csv.DictWriter(file_out, fieldnames=['output_field'],
                           quoting=csv.QUOTE_NONNUMERIC)
    writer.writeheader()
```

アプリケーション

シノニムを作成するアプリケーションディレクトリの名前を入力するか、[...] (参照) ボタンをクリックしてアプリケーションディレクトリを指定し、[OK] をクリックします。

シノニム名

作成するシノニムの名前を入力するか、デフォルト名を受容します。

次のステップ

注意： WebFOCUS Reporting Server ブラウザインターフェースでもサンプルの Python スクリプトおよびデータファイルを作成することができます。サンプルファイルを格納するアプリケーションディレクトリを作成し、アプリケーションフォルダを右クリックして [新規] を選択し、[チュートリアル] をクリックします。[チュートリアル] ページで、[WebFOCUS - Retail デモ] チュートリアルを選択し、[Python サンプルの作成] のチェックがオンになっていること、および [チュートリアルデータのデータ量制限] が [大] または [中] に設定されていることを確認します。[作成] をクリックしてデモファイルを作成します。

ユーザ定義の Python スクリプトの実行

Python スクリプトを実行するには、WebFOCUS PYTHON 関数を呼び出します。PYTHON 関数に追加する引数には、Python スクリプトのシノニム、入力フィールド、出力フィールドがあります。

ユーザ定義の Python 関数の実行

```
PYTHON([app/]synonym, input1 [, input2 ...], output)
```

説明

[app/]synonym

Python スクリプトのアプリケーションおよびシノニム名です。

input1 [, input2 ...]

入力引数です。

output

出力引数です。この引数は、マスターファイルの OUTPUT_DATA セグメントのフィールド名と一致する必要があります。

Python スクリプトの実行

次の Python スクリプトは、ARITHMETIC という名前の関数で ADDITION、SUBTRACTION、MULTIPLICATION、DIVISION の4 つの出力フィールドを演算する、arithmetic_example_multiple_computes.py スクリプトです。この例で使用するファイルはすべて、python という名前のアプリケーションディレクトリに格納されます。

```
# arithmetic_example_multiple_computes.py

import csv
import time

def arithmetic(csvin, csvout):
```

```

with open(csvin, 'r', newline='') as file_in,¥
    open(csvout, 'w', newline='') as file_out:

    fieldnames = ['addition', 'subtraction',
                  'multiplication', 'division']

    reader = csv.DictReader(file_in, quoting=csv.QUOTE_NONNUMERIC)

    writer = csv.DictWriter(file_out, quoting=csv.QUOTE_NONNUMERIC,
                           fieldnames=fieldnames)
    writer.writeheader()

    for row in reader:
        addition      = row['a_number'] + row['another_number']
        subtraction   = row['a_number'] - row['another_number']
        multiplication = row['a_number'] * row['another_number']
        division      = row['a_number'] / row['another_number']

        writer.writerow({'addition':      addition,
                        'subtraction':     subtraction,
                        'multiplication':   multiplication,
                        'division':        division})

```

サンプルデータを格納する .csv ファイル arithmetic_sample_input.csv には、ヘッダレコードおよびデータタイプと入力引数の長さの指定に使用される 2 つのデータレコードが含まれます。

```

"a_number", "another_number"
1.1
100,000.100000

```

下図は、このスクリプトのシノニム作成画面を示しています。

The screenshot shows a 'Create Synonym for PYTHON' dialog box. It is divided into two main sections: 'Create Synonym options' and 'Customize data type mappings'. In the first section, the 'PYTHON Script' is set to 'python/arithmetic_example_multiple_computes.py', the 'Function Name' is 'arithmetic', and the 'File with sample input data for the PYTHON Script' is 'python/arithmetic_sample_input.csv'. There are checkboxes for 'CSV files with header' for both 'Input' and 'Output', both of which are checked. The second section, 'Customize data type mappings', has 'Application' set to '/bisamp', 'Prefix' and 'Suffix' are empty, and 'Synonym Name' is 'arithmetic_example_syn'.

生成されたマスターファイルは次のとおりです (arithmetic_example_syn.mas)。

```

FILENAME=ARITHMETIC_EXAMPLE_SYN, SUFFIX=PYTHON , $
SEGMENT=INPUT_DATA, SEGTYPE=S0, $
  FIELDNAME=A_NUMBER, ALIAS=a_number, USAGE=I11, ACTUAL=STRING,
  MISSING=ON,
  TITLE='a_number', $
  FIELDNAME=ANOTHER_NUMBER, ALIAS=another_number, USAGE=I11, ACTUAL=STRING,
  MISSING=ON,
  TITLE='another_number', $
SEGMENT=OUTPUT_DATA, SEGTYPE=U, PARENT=INPUT_DATA, $
  FIELDNAME=ADDITION, ALIAS=addition, USAGE=D10.1, ACTUAL=STRING,
  MISSING=ON,
  TITLE='addition', $
  FIELDNAME=SUBTRACTION, ALIAS=subtraction, USAGE=D5.1, ACTUAL=STRING,
  MISSING=ON,
  TITLE='subtraction', $
  FIELDNAME=MULTIPLICATION, ALIAS=multiplication, USAGE=D15.1,
ACTUAL=STRING,
  MISSING=ON,
  TITLE='multiplication', $
  FIELDNAME=DIVISION, ALIAS=division, USAGE=D5.1, ACTUAL=STRING,
  MISSING=ON,
  TITLE='division', $

```

生成されたアクセスファイルは次のとおりです (arithmetic_example_syn.acx)。

```

SEGNAME=INPUT_DATA,
MODNAME=python/arithmetic_example_multiple_computes.py,
FUNCTION=arithmetic,
PYTHON_INPUT_SAMPL=python/arithmetic_sample_input.csv,
INPUT_HEADER=YES,
OUTPUT_HEADER=YES, $

```

次の WebFOCUS プロシジャ (sales_multiple_computes.fex) は、4 つの出力それぞれに対して値が返されるように、ARITHMETIC 関数を 4 回呼び出します。

```

-* sales_multiple_computes.fex

TABLE FILE GGSales
SUM
  DOLLARS
  UNITS

COMPUTE Addition/D7      = PYTHON(python/arithmetic_example_syn,
                                DOLLARS, UNITS, ADDITION);
COMPUTE Subtraction/D7   = PYTHON(python/arithmetic_example_syn,

```

```

                                DOLLARS, UNITS, SUBTRACTION);
COMPUTE Multiplication/D16 = PYTHON(python/arithmetic_example_syn,
                                DOLLARS, UNITS, MULTIPLICATION);
COMPUTE Division/D7.2      = PYTHON(python/arithmetic_example_syn,
                                DOLLARS, UNITS, DIVISION);

WHERE RECORDLIMIT EQ 100
HEADING
    "Arithmetic Example, Multiple Computes"
    ""

ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END

```

下図は、出力結果を示しています。

Arithmetic Example, Multiple Computes					
<u>Dollar Sales</u>	<u>Unit Sales</u>	<u>Addition</u>	<u>Subtraction</u>	<u>Multiplication</u>	<u>Division</u>
1343927	105860	1,449,787	1,238,067	142,268,112,220	12.70

WebFOCUS Reporting Server リモートサーバアダプタの使用

WebFOCUS Reporting Server リモートサーバアダプタの構成方法について説明します。

WebFOCUS Reporting Server リモートサーバアダプタを使用すると、アプリケーションからリモートサーバ上に存在するデータソースへのアクセスが可能になります。

「プラットフォームの透過性」と呼ばれるメカニズムを使用することで、異なるプラットフォームに存在するデータソースを結合できます。サーバへの接続に使用される通信プロトコルは、同一である場合も、異なる場合もあります。

リモート ibi WebFOCUS Reporting Server の構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインター

フェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

接続属性

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

基本

NODE

8 バイトの文字列

ノードブロックの論理名を定義します。これは、ファイル内で一意である必要があります。論理名は最大 8 バイトで、文字で始まる必要があります、セミコロン (;) と等号 (=) を除く任意の文字を含めることができます。ノードブロックにはキーワードと値の組み合わせ

が含まれ、それらは BEGIN-END ステートメントで囲まれています。以下は、ノードブロックの例を示しています。

```
NODE = ABC
BEGIN
  DESCRIPTION = client node block
  CLASS = CLIENT
  PROTOCOL = TCP
  HOST = localhost
  PORT = 8100
END
```

HOST

文字列

クライアントが接続するホスト、またはリスナが待ち受ける IP アドレスを定義します。

PORT

正の整数

クライアントが接続またはリスナが待ち受ける TCP ポート番号を定義します。

HTTP_PORT

正の整数

リモートサーバの WebFOCUS Reporting Server ブラウザインターフェースリスナが待ち受ける TCP ポート番号を定義します。リモートシノニムの目的で通信ブロックを宣言する際にも使用されます。

CLASS

SECURITY

アプリケーションに資格情報が提供されていない場合に、リモートサーバとの通信に使用する認証方法を定義します。

- **IWA** - Windows クライアントと Windows サーバ間の統合 Windows 認証に使用します。
- **PWP** - パスワードパススルー認証に使用します。
- **TRUSTED** - トラステッド接続に使用します。この接続では、現在のログオンユーザのユーザ名をサーバに渡します。

- **userid,password** - IWA、PWP、TRUSTED 以外の値は、EXPLICIT 認証のユーザ ID とパスワード文字列と見なされます。

この情報は、クリアテキストにすることも、暗号化することもできます。
WebFOCUS Reporting Server ブラウザインターフェースを使用して値を設定する場合、暗号化は自動的に行われます。

ENGINE EDA SET CONNECTION_ATTRIBUTES server/userid,password コマンドがプロファイル (サーバ、グループ、ユーザ) またはプロシジャで実行された場合、SECURITY = (IWA、PWP、TRUSTED、その他) で設定された値は上書きされます。

DESCRIPTION

文字列

Reporting Server ノードの説明です。

詳細 これらのオプションを使用すると、特定のサービスの接続情報を提供できます。

SERVICE NAME

CLIENT (servicename)

リモートサーバのアウトバウンド通信方法を定義します。

servicename はオプションです。servicename を指定する場合、サーバのサービスブロックの [SERVICE] の値に一致させる必要があります。

HTTP_SSL

0 または 1

リモートサーバの WebFOCUS Reporting Server ブラウザインターフェースリスナで、セキュアソケットレイヤプロトコルを使用するかどうかを定義します。

- **0** - WebFOCUS Reporting Server ブラウザインターフェースへの接続に SSL を使用しません。
- **1** - WebFOCUS Reporting Server ブラウザインターフェースへの接続に SSL を使用します。

COMPRESSION

0 または 1

クライアントとサーバ間のデータ転送でデータ圧縮を有効にします。

- **0** - 圧縮しません。
- **1** - 圧縮をオンにします。

ENCRYPTION

0、DES、詳細、または IBCRYPT

クライアントとサーバ間のデータ転送で使用する暗号化方法を定義します。

- **0** - 暗号化しません。
- **DES** - 56 ビット固定キーデータ暗号化標準を使用します。
- **詳細** - 暗号、モード、RSA キー長の選択と組み合わせが、簡単に行えます。このオプションでは、3DES、AES 128、AES192、AES 256 の各暗号と、ECB と CBC のモードが提供されます。
- **IBCRYPT** - ユーザ定義のアルゴリズムを使用します。キーは 512 ビットの RSA 暗号化です。

注意：HTTP プロトコルの場合は、[0] と [DES] のみがサポートされています。

CONNECT_LIMIT

秒数

クライアントがサーバからの TCP 接続レスポンスを待機する最大時間を定義します。

- -1 - 待機時間は無制限になります。
- 0 - 待機しません。

MAXWAIT

クライアントがサーバからのレスポンスを待機する時間を秒数で定義します。

「-1」と入力すると、タイムアウトは無制限になります。

メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および データタイプのサーバマッピングを定義するシノニムを作成します。

シノニムの作成

シノニムは、SUB Server によって管理されるメタデータごとに、固有の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

リモート WebFOCUS Reporting Server アダプタを使用して、SUB Server 上に存在するリモートデータソースのシノニムを作成することができます。データソースがリモートサーバに存在する場合、マスターファイルには SUFFIX=EDA 属性が追加されます。

注意：リモート WebFOCUS Reporting Server アダプタはリレーショナルデータソースを定義するため、マスターファイルは単一セグメントです。シノニムが CAR ファイルなどの複数セグメントの階層テーブルから作成された場合、そのすべてのセグメントは、結果のマスターファイルで 1 つのセグメントにフラット化されます。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。

- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

シノニム作成パラメータ

以下のリストは、アダプタのシノニムを作成する際に、値の指定が必要なパラメータ、および実行する関連作業を示しています。これらのオプションが複数のウィンドウに表示される場合もあります。ウィンドウに表示されたボタンをクリックして次のウィンドウに進み、最後に [シノニムの作成] ボタンをクリックします。これで、入力データに基づいてシノニムが作成されます。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにしま

す。

- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

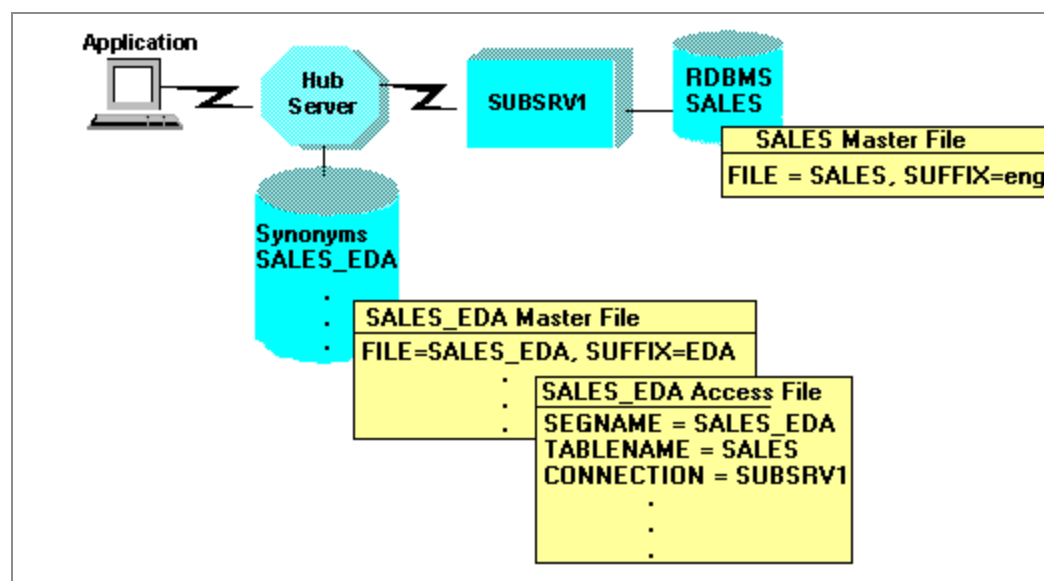
リモートデータソースのシノニム

下図は、データソースのマスターファイルとアクセスファイルの使用方法を示しています。SALES_EDA というシノニムが、SUB Server 上の SALES ファイルを使用して HUB Server 上に作成されています。

- SALES ファイルの複数のセグメントは、SALES_EDA マスターファイルではフラット化され、単一セグメントテーブルになっています。
- SALES_EDA マスターファイルには、データソースがリモートであることを示す SUFFIX=EDA が含まれています。
- SALES_EDA アクセスファイルには、データソースのシノニム、データソースの名前、およびデータソースのパスが含まれています。

```
SEGNAME = SALES_EDA
TABLENAME = SALES
CONNECTION = SUBSRV1
```

HUB Server に接続すると、SALES_EDA マスターファイルとアクセスファイルを使用して、SUBSRV1 上のリモートデータソース SALES にアクセスします。



上記のアクセスファイルの例では、CONNECTION パラメータの値 SUBSRV1 が HUB Server の通信ファイル内のアウトバウンドノードブロックと一致する必要があります。

SUB Server 上の元のシノニムは、次のとおりです。

Sales.mas

```
FILENAME=KSALES, SUFFIX=FOC      , $
  SEGMENT=STOR_SEG, SEGTYPE=S1, $
    FIELDNAME=STORE_CODE, ALIAS=SNO, USAGE=A3, $
    FIELDNAME=CITY, ALIAS=CTY, USAGE=A15, $
    FIELDNAME=AREA, ALIAS=LOC, USAGE=A1, $
$ $
  SEGMENT=DATE_SEG, SEGTYPE=SH1, PARENT=STOR_SEG, $
    FIELDNAME=DATE, ALIAS=DTE, USAGE=A4MD, $
$ $
  SEGMENT=PRODUCT, SEGTYPE=S1, PARENT=DATE_SEG, $
    FIELDNAME=PROD_CODE, ALIAS=PCODE, USAGE=A3, FIELDTYPE=I, $
    FIELDNAME=UNIT_SOLD, ALIAS=SOLD, USAGE=I5, $
    FIELDNAME=RETAIL_PRICE, ALIAS=RP, USAGE=D5.2M, $
    FIELDNAME=DELIVER_AMT, ALIAS=SHIP, USAGE=I5, $
    FIELDNAME=OPENING_AMT, ALIAS=INV, USAGE=I5, $
    FIELDNAME=RETURNS, ALIAS=RTN, USAGE=I3, MISSING=ON, $
    FIELDNAME=DAMAGED, ALIAS=BAD, USAGE=I3, MISSING=ON, $
```

シノニム作成ユーティリティによって作成されたシノニムは、次のとおりです。

Sales_eda.mas

```
FILENAME=SALES_EDA, SUFFIX=EDA      , $
  SEGMENT=SALES_EDA, SEGTYPE=S0, $
    FIELDNAME=STORE_CODE, ALIAS=STORE_CODE, USAGE=A3, ACTUAL=A3, $
    FIELDNAME=DATE, ALIAS=DATE, USAGE=A4MD, ACTUAL=A4MD, $
    FIELDNAME=PROD_CODE, ALIAS=PROD_CODE, USAGE=A3, ACTUAL=A3, $
    FIELDNAME=CITY, ALIAS=CITY, USAGE=A15, ACTUAL=A15, $|
    FIELDNAME=AREA, ALIAS=AREA, USAGE=A1, ACTUAL=A1, $
    FIELDNAME=UNIT_SOLD, ALIAS=UNIT_SOLD, USAGE=I5, ACTUAL=I4, $
    FIELDNAME=RETAIL_PRICE, ALIAS=RETAIL_PRICE, USAGE=D5.2M, ACTUAL=D8, $
    FIELDNAME=DELIVER_AMT, ALIAS=DELIVER_AMT, USAGE=I5, ACTUAL=I4, $
    FIELDNAME=OPENING_AMT, ALIAS=OPENING_AMT, USAGE=I5, ACTUAL=I4, $
    FIELDNAME=RETURNS, ALIAS=RETURNS, USAGE=I3, ACTUAL=I4, MISSING=ON, $
    FIELDNAME=DAMAGED, ALIAS=DAMAGED, USAGE=I3, ACTUAL=I4, MISSING=ON, $
```

Sales_eda.acx

```
SEGNAME=SALES_EDA, TABLENAME=EDADBA.sales, CONNECTION=server, KEYS=3, $
```

構造がフラット化されているだけでなく、3つのキー (STORE_CODE、DATE、PROD_CODE) が最上部に移動されていることに注意してください。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

アクセスファイルのキーワード

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABlename	テーブルまたはビューの名前です。この値には、以下のよう にロケーションまたはオーナー名を含めることができます。 <pre>TABlename=[location.][owner.]tablename</pre>
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre> CONNECTION=' ' は、ローカルのデータベースサーバへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。

キーワード	説明
KEYS	テーブルの主キーを構成するカラムの数を指定します。範囲は 0 から 64 です。マスターファイルセグメントの最初の n フィールドに対応しています。
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。

アクセスファイルへのサーバ名の保存

次の例は、サーバ名 IBMSERVE をアクセスファイルに保存する方法を示しています。

```
SEGNAME=ONE, TABLENAME=CAR, KEYS=1, WRITE=YES, SERVER=IBMSERVE, $
```

注意：アクセスするデータソースのタイプに関係なく、アクセスファイルには以下の属性が含まれている必要があります。

- **SEGNAME** アクセスファイルのセグメント名は、マスターファイルのセグメント名と一致させる必要があります。
- **TABLENAME** この属性は、サーバ上のマスターファイルの名前を指定します。
- **SERVER** この属性は、サーバの名前を指定します。

SUFFIX=EDA シノニムの使用

サーバ上のシノニム (DIGIT) のマスターファイルは次のとおりです。

```
FILENAME=DIGIT, SUFFIX=FOC, $
SEGNAME=DIGIT, SEGTYPE=S0, $
FIELD=THIS_DIGIT ,THIS_DIGIT ,I9 ,I4 ,MISSING=ON, $
FIELD=SSN ,SSN ,A9 ,A9 ,MISSING=ON, $
FIELD=AMOUNT1 ,AMOUNT1 ,P16.0 ,P8 ,MISSING=ON, $
FIELD=AMOUNT2 ,AMOUNT2 ,P16.0 ,P8 ,MISSING=ON, $
```

クライアント上のシノニム (DIGITEDA) のマスターファイルは次のとおりです。

```

FILENAME=DIGITEDA,SUFFIX=EDA,$
SEGNAME=DIGIT,SEGTYPE=S0,$
  FIELD=THIS_DIGIT ,THIS_DIGIT ,I6 ,I4 ,,$
  FIELD=SSN ,SSN ,A9 ,A9 ,MISSING=OFF,$
  FIELD=AMOUNT1 ,AMOUNT1 ,P8 ,P8 ,MISSING=ON ,,$
  FIELD=AMOUNT2 ,AMOUNT2 ,P9.0 ,P8 ,MISSING=ON ,,$

```

クライアント上のシノニム (DIGITEDA) のアクセスファイルは、サーバ上のファイルを指定しています。

```

SEGNAME=DIGIT,TABLENAME=DIGIT,KEYS=1,SERVER=PMSEDA,$

```

説明

- SEGNAME=DIGIT は、サーバマスターファイル内のセグメント名と一致します。
- TABLENAME=DIGIT は、サーバ上のマスターファイルの名前を指定します。
- SERVER=PMSEDA は、サーバの名前を指定します。

EDACS3 クライアント通信構成ファイルには、サーバを指定する NODE ブロックが含まれています。以下はその例です。

```

NAME = EDA CLIENT USING CS/3 TCP/IP
NODE = PMSEDA
BEGIN
; TRACE = 31
  PROTOCOL = TCP
  CLASS = CLIENT
  HOST = IBIMVS ; DNS NAME (PNO 28109)
  SERVICE = 2386 ; TCP/IP PORT FOR SERVER
END

```

次のリクエスト (SQL および TABLE) は、DIGITEDA という名前のローカルマスターファイルを参照します。上記のように、対応するアクセスファイルには、サーバで指定したとおりのサーバ名とマスターファイル名が含まれています。この場合、サーバ PMSEDA には DIGIT というマスターファイルが含まれています。サーバでは、DIGIT マスターファイルには FOCUS データソースが記述されています。通信構成ファイルには、FOCUS クライアントがサーバとの通信を確立できるように、サーバ名 PMSEDA の項目が含まれています。

SQL リクエスト	TABLE リクエスト
SQL SELECT * FROM DIGITEDA END	TABLE FILE DIGITEDA PRINT * END

注意：SUFFIX=EDA のアクセスファイルで指定する元の物理ファイルの名前を、数字で開始することはできません。ファイル名の先頭に数字を使用すると、次の問題が発生します。

- 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。
- 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。
- 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他社製アプリケーションでは、問題が発生することがあります。

リモートの複数セグメントマスターファイルおよびアクセスファイル

以下は、複数セグメントのシノニムを示しています。SUB Server 上の 2 つのテーブル (JOBHIST および JOBLIST) が記述され、これらがフィールド JOBCCLASS で結合されています。

Jobs_eda.mas

```
FILENAME=jobs_eda, SUFFIX=EDA      , $
SEGMENT=JOBHIST, SEGTYPE=S0, $
  FIELDNAME=PIN, ALIAS=PIN, USAGE=A9, ACTUAL=A9, $
  FIELDNAME=JOBSTART, ALIAS=JOBSTART, USAGE=YMD, ACTUAL=DATE, $
  FIELDNAME=JOBCCLASS, ALIAS=JOBCCLASS, USAGE=A8, ACTUAL=A8, $
  FIELDNAME=FUNCTITLE, ALIAS=FUNCTITLE, USAGE=A20, ACTUAL=A20, $
SEGMENT=JOBLIST, SEGTYPE=S0, PARENT=JOBHIST,
  JOIN_WHERE=JOBHIST.JOBCCLASS EQ JOBLIST.JOBCCLASS;, $
  FIELDNAME=JOBCCLASS, ALIAS=JOBCCLASS, USAGE=A8, ACTUAL=A8, $
  FIELDNAME=CATEGORY, ALIAS=CATEGORY, USAGE=A25, ACTUAL=A25, $
  FIELDNAME=JOBDESC, ALIAS=JOBDESC, USAGE=A40, ACTUAL=A40, $
```

```

FIELDNAME=LOWSAL, ALIAS=LOWSAL, USAGE=D12.2M, ACTUAL=D8, $
FIELDNAME=HIGHSAL, ALIAS=HIGHSAL, USAGE=D12.2M, ACTUAL=D8, $
FIELDNAME=GRADE, ALIAS=GRADE, USAGE=A2, ACTUAL=A2, $
FIELDNAME=LEVEL, ALIAS=LEVEL, USAGE=A25, ACTUAL=A25, $

```

対応する複数セグメントのアクセスファイルは次のとおりです。

Jobs_eda.acx

```

SEGNAME=JOBHIST, TABLENAME=EDADBA.jobhist, CONNECTION=server, $
SEGNAME=JOBLIST, TABLENAME=EDADBA.joblist, CONNECTION=server, $

```

注意：このシノニムでは、2つの異なるテーブルが指定されているため、WebFOCUS Reporting Server ブラウザインターフェースの [シノニムの作成] 機能を使用して作成することはできませんが、ibi Data Migrator デスクトップインターフェースのシノニムエディタを使用して作成することができます。新しいシノニムから開始し、各テーブルは、リモートサーバから [挿入]、[セグメント]、[インポートしたメタデータからセグメントを挿入] を使用して追加されます。JOIN_WHERE 句は自動的に追加され、テーブルが JOBCCLASS で結合されることを示します。

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアードプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

ストアドプロシジャの実行

メタデータを使用してデータを使用可能にすることに加えて、サーバはリモートサーバ上で実行されるコマンドを実行することができます。一般的なコマンドは、リモートサーバ上のプロシジャを実行しますが、CREATE コマンドおよび DROP コマンドも実行できます。

リモートサーバの元の構成では、選択したプロファイルに明示的な ENGINE EDA SET CONNECTION_ATTRIBUTES ノード行が追加されますが、一方では、サーバは最初に通信ファイルを参照し、使用可能なターゲットノードのリストを作成します。その結果 (他のアダプタエンジンとは異なり)、デフォルトのリモートノードは、通常、プロファイル内で最初に検出される ENGINE EDA SET CONNECTION_ATTRIBUTES ノード行にはなりません。

使用する接続を設定するための 2 つのコマンドオプションに加えて、接続ステータスをテストして代替の ID およびパスワードの情報を渡すためのコマンドが用意されています。さらに、変数 &EDASERVER および &EDAUSER には、現在の接続ターゲットノードと ID が入力されます。

次のリクエストへの接続の設定

次のコマンドを発行します。

```
SQL EDA SET SERVER node
```

説明

node

有効な構成済みノードです。

セッションの以降の接続を設定

次のコマンドを発行します。

```
SQL EDA SET DEFAULT_CONNECTION node
```

説明

node

有効な構成済みノードです。

または、次の構文を使用することもできます。

```
REMOTE DEST = node
```

接続ステータスのテスト

次のコマンドを発行します。

```
SQL EDA PING node
```

説明**node**

有効な構成済みノードです。

「準備完了」状態ではない接続では、適切な FOC エラーメッセージが表示された上で、&RETCODE と &FOCERRNUM が設定されます。さらに、これらをダイアログマネージャ構文で使用することで、アプリケーションの必要に応じて実行を指示することができます (つまり、-GOTO EXIT または代替ノードを使用)。

SET CONNECTION_ATTRIBUTES による代替 ID とパスワードの組み合わせの使用

次のコマンドを発行します。

```
SQL EDA SET CONNECTION_ATTRIBUTES node/id,password
```

説明**node**

有効な構成済みノードです。

id

リモートサーバ上の有効な ID です。

password

ID の有効なパスワードです。

接続を設定してもデフォルトの接続にはならないため、通常、このコマンドは SET SERVER または SET DEFAULT_CONNECTION コマンドとともに使用する必要があることを、覚えておくことが重要です。

REMOTE USER および REMOTE PASS での代替 ID とパスワードの組み合わせの使用

次のコマンドを入力して、ID とパスワードを現在のデフォルトリモート接続に適用します。

```
REMOTE USER = id  
REMOTE PASS = password
```

説明**id**

リモートサーバで有効なユーザ ID です。

password

ID の有効なパスワードです。

使用可能なサーバの確認

```
SQL EDA ?SERVERS
```

リモート実行を使用したストアードプロシジャの実行

ストアードプロシジャを実行するには、次のコマンドを発行します。

```
SQL EDA EX rpcnameparm1, parm2, ...  
END
```

説明

rpcname

サーバ上のプロシジャです。

parm1, parm2,...

サーバに送信される文字列です (FOCEXEC の実行行で渡すことができるパラメータと同じ)。

リモート接続でのローカル保存コマンドの実行

ローカルに保存されたコマンドは、次の構文を使用してリモート接続で実行できます。

```
-REMOTE BEGIN  
command1  
command2  
command3  
...  
-REMOTE END
```

説明

command1, command2, command3 ...

プロシジャ内の任意の数のコマンドまたは行 (例、TABLE FILE ... END リクエスト内の行)。

デフォルト設定のリモート接続は、コードが実行されるターゲットノードです。SET DEFAULT_CONNECTION または REMOTE DEST を使用して、実行ターゲットを変更します。

コマンド内の変数は、実行のためにリモート接続に送信される前に、ローカルで解決されます。

ローカルマシン上のダイアログマネージャによって最初に解決されるため、リモートマシン上で解決させるダイアログマネージャの直接ダッシュコマンド (例、-SET、-IF、-GOTO) および変数は許可されません。次の例のように、コマンドをローカル変数内に隠すという手法を使用して、解決を遅延させることで、ダイアログマネージャのコマンドと変数を間接的に使用することができます。

```
-SET &GOWHERE = 'STEP1' ;
-SET &DASHGOTO = '-GOTO' ;
-SET &DASHSTEP1 = '-STEP' ;
-SET &DASHSTEP2 = '-STEP2' ;
-SET &DASHSTART = '-START' ;
-SET &DASHEND = '-END' ;
-REMOTE BEGIN
&DASHSTART
&DASHGOTO &GOWHERE
&DASHSTEP1
...
&DASHGOTO END
&DASHSTEP2
&DASHEND
-REMOTE END
```

クエリアダプタの設定

現在のアダプタパラメータ設定を表示するには、次のコマンドを発行します。

```
SQL EDA ?
```

出力結果は次のとおりです。

```
(FOC1450) CURRENT EDA INTERFACE SETTINGS ARE :
(FOC1446) DEFAULT DBSPACE IS                - :IBIEDA
(FOC1449) CURRENT SQLID IS                   - :USER1
(FOC1444) AUTOCLOSE OPTION IS                 - :ON FIN
(FOC1496) AUTODISCONNECT OPTION IS           - :ON FIN
```

(FOC1499) AUTOCOMMIT OPTION IS	- :ON COMMAND
(FOC1441) WRITE FUNCTIONALITY IS	- :OFF
(FOC1445) OPTIMIZATION OPTION IS	- :ON
(FOC1484) SQL ERROR MESSAGE TYPE IS	- :DBMS
(FOC1552) INTERFACE DEFAULT DATE TYPE	- :NEW

REST アダプタの使用

REST アダプタの構成方法について説明します。

REST は、REpresentational State Transfer の略語です。REST スタイルのアーキテクチャは、クライアントとサーバで構成されます。クライアントはリクエストをサーバに送信し、サーバはリクエストを処理して適切なレスポンスを返します。REST は当初、HTTP のコンテンツで記述されていました。REST は、GET、POST、PUT、DELETE など一般的で定義済みのメソッドを利用します。

RESTful Web サービスは、HTTP の使用と REST の原理に基づいて実装される Web サービスです。

REST アダプタは、RESTful Web サービスをサポートし、XML または JSON (JavaScript Object Notation) のいずれかのレスポンスが使用されます。詳細は、[JSON アダプタの使用](#) および [XML アダプタの使用](#) を参照してください。

REST アダプタの構成

REST アダプタの構成では、接続情報を指定するほか、必要に応じて、少なくとも 1 つの接続に対して認証情報を指定します。

接続属性の宣言

ターゲット Web サービスをホストする RESTful Web サービスプロバイダにアクセスするには、アダプタに接続情報を指定する必要があります。

この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。接続および認証情報は、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で入力することができます。選択したプロファイルにコマンドが追加されます。選択可能なプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) があります。

複数の接続を追加することで、複数の Web サービスプロバイダへの接続を宣言することができます。この場合、複数の SET CONNECTION_ATTRIBUTES コマンドを発行します。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。

6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

REST アダプタの接続属性

REST アダプタは、[プロシジャ] グループフォルダ下にあります。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

ベース URL

特定の RESTful Web サービスのすべての機能に共通の URL の一部です。アドレスは、http:// または https:// で開始されます (例、http://api.geonames.org)。

セキュリティ

データベースサーバへの接続時に、ユーザを認証する 4 つの方法があります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。
- **Trusted** アダプタは、現在のオペレーティングシステムに関連している偽装プロセス用のデータベースの規則を使用してデータベースに接続します。
- **OAuth** アダプタは、REST API へのアクセス許可に OAuth 2.0 認証プロトコルを使用してデータベースに接続します。

KERBEROS SPN

クライアントが、サービスのインスタンスを一意に識別する名前です (例、mydaemon/foo:4761)。

KERBEROS SPN は、Trusted セキュリティを選択した場合に表示されます。

OAuth 権限付与タイプ

構成可能な OAuth 権限付与タイプには、次の 3 つがあります。

- **認可コード** 認可リクエストから返される認可コードを使用し、[リフレッシュトークン] で更新可能な長期アクセストークンを取得することができます。
- **パスワード** OAuth 実装でユーザ ID とパスワードのみを必要とする場合に使用します。
- **クライアント認証情報** OAuth 実装でクライアント ID とクライアントシークレットのみを必要とする場合に使用します。

このフィールドは、OAuth セキュリティを選択した場合のみ表示されます。

サービスプロバイダ

特定のサービスプロバイダの OAuth 構成です。このフィールドは、OAuth セキュリティおよび認可コード権限付与タイプを選択した場合に表示されます。

サービス URL

REST または OData アダプタ接続として構成されていない場合、Web サービスへの URL です。

ユーザ

REST サービスに登録されたプライマリ認証 ID です。このフィールドは、Explicit セキュリティを選択した場合、または OAuth セキュリティおよびパスワード権限付与タイプを選択した場合に表示されます。

パスワード

プライマリ認証 ID に関連するパスワードです。このフィールドは、Explicit セキュリティを選択した場合、または OAuth セキュリティおよびパスワード権限付与タイプを選択した場合に表示されます。

クライアント認証情報 (body)

このフィールドは、OAuth セキュリティおよびパスワード権限付与タイプを選択した場合に表示されます。このオプションを選択すると、接続文字列に `bdyclt:true` 属性が追加され、認証情報が HTTP ヘッダではなく POST リクエストの本文内に送られます。

クライアント ID

REST アプリケーションで定義されたクライアント ID です。このフィールドは、OAuth セキュリティを選択した場合のみ表示されます。

クライアントシークレット

REST アプリケーションで定義されたクライアントシークレットです。このフィールドは、OAuth セキュリティを選択した場合のみ表示されます。

認可 URL

特定のアプリケーションへの OAuth 認可に使用する URL です。

たとえば、Google API セットの認可 URL は「<https://accounts.google.com/o/oauth2/auth>」です。

トークン URL

特定のアプリケーションへのアクセストークンの取得に使用する URL です。このフィールドは、OAuth セキュリティを選択した場合のみ表示されます。

たとえば、Google API セットのトークン URL は
「https://accounts.google.com/o/oauth2/token」です。

追加の認証パラメータ

このフィールドは、OAuth セキュリティで認可コードまたはクライアント認証情報の権限付与タイプを選択した場合に表示されます。認証に必要な追加のパラメータを入力します。

アクセストークン

特定のユーザを識別する値で、REST アプリケーションが動作する際に、そのユーザの代わりとして使用されます。このフィールドは、OAuth セキュリティおよび認可コード権限付与タイプを選択した場合に表示されます。このトークンを取得するには、[アクセストークンの取得] ボタンをクリックします。

[アクセストークンの取得] を完了するには、WebFOCUS Reporting Server ブラウザインターフェースを起動するために使用したホスト名と、REST アプリケーションの [リダイレクト URI] で設定したホスト名が一致する必要があります。

リフレッシュトークンを保存しない

このフィールドは、OAuth セキュリティおよび認可コード権限付与タイプを選択した場合に表示されます。このオプションは、リクエストトークン機能をサポートしていない Adobe API にのみ使用されます。このチェックボックスをオンにすると、OAuth トークン要求から返される更新トークンは格納されません。

リフレッシュトークン

OAuth トークンリクエストから返されるリフレッシュトークンです。このフィールドは、OAuth セキュリティおよび認可コード権限付与タイプを選択した場合に表示されます。このトークンは、特定のアプリケーションの REST サービスにアクセスしてレポートを実行する際に、新しいアクセストークンを取得するために使用されます。

連鎖認証

このオプションを選択すると、Web サービス認証リクエストが実行され、Cookie またはトークンを含むレスポンスが後続の Web サービスリクエストに渡されます。

認証用シノニム

このオプションは、連鎖認証にのみ使用します。アプリケーションを選択すると、SUFFIX=REST を含むシノニムのリストが表示されます。選択したシノニムは、Web サービス認証を実行し、セキュリティコンテキストを返すために使用されます。

この処理についての詳細は、[連鎖実行用のアダプタを構成するには](#)を参照してください。

HTTP 認証値

HTTP ヘッダ内の認証の一部として渡されるトークン値の名前です。

認可行キー

認可リクエストのレスポンスからトークンを選択するために使用するフィールド名および値です。これは、HTTP ヘッダ内の認証の一部として渡されます (例、INSTALLATIONID="1")。

認証行の選択

このオプションを選択すると、認可行キーの値を入力するために、認証用シノニムの各フィールド値が表示されます。値のクリック時に、[RowKey の表示] を選択します。

[認可行キー] には、FIELDNAME および対応する値が入力されます (例、INSTALLATIONID="1")。

HTTP 接続詳細オプション

カスタムヘッダの追加

カスタムヘッダを追加するためのテキストボックスが表示されます。

カスタムヘッダ

カスタムヘッダの値は二重引用符 (") で囲み、各カスタムヘッダはセミコロン (;) で区切る必要があります。以下はその例です。

```
Content-Type="CDF";PARAM1="ABC"
```

接続文字列でカスタムヘッダを定義すると、カスタムヘッダが REST サービス全体に適用されます。カスタムヘッダは、メタデータレベルでも定義することができます。メタデータレベルでカスタムヘッダを定義すると、カスタムヘッダは特定の REST コールに適用され、WHERE または IF ステートメントを使用してレポート実行時にカスタムヘッダの値を変更することができます。

プロキシサーバ IP アドレス

プロキシサーバの IP アドレスです。このサーバがリクエストを代行受信して、実サーバに転送します。

プロキシポート番号

プロキシサーバのリスナポート番号です。デフォルトポート番号は 80 です。

PROXY HTTPS 相対パス

このオプションを選択すると、プロキシサーバの構成時に、絶対パスではなく相対パスが REST リクエストにより送信されます。

SSL 証明書

ローカルに格納されるユーザ指定の SSL 認証用のサーバの x.509 証明書の場所です。証明書ファイルは、アダプタが接続するサーバの認証に使用されます。

SSL 相互認証

このオプションを選択すると、相互認証が有効になります。

SSL 証明書タイプ

証明書のタイプです。次のオプションがあります。

- **Trusted** 認証済み証明書ファイル (trustedcertfile) には、次のように PEM フォーマットの CA 証明書のファイルが指定されます。

```
-----BEGIN CERTIFICATE-----  
...(CA certificate in base64 encoding) ...  
-----END CERTIFICATE-----
```

認証済み証明書ファイルには、複数の CA 証明書を格納することができます。証明書の前後または証明書の間に、テキストを追加することができます。たとえば、証明書の説明を追加することができます。

- **Trusted 以外** このオプションを選択すると、構成ウィンドウに、[キーファイル]、[パスフレーズ]、[ラベル] が追加されます。

SSL 証明書キーファイル

クライアントの X.509 証明書を PEM フォーマットで作成するために使用されるプライベートキーです。このオプションは、Trusted 以外の接続用の証明書と

もに使用されます。

SSL 証明書パスフレーズ

キーファイルのロックを解除するために使用されるパスワードです。この値は、キーファイルが暗号化される場合にのみ必要です。

SSL 証明書ラベル

ファイルに複数の証明書が含まれている場合に、ファイル内の証明書を識別します。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

RESTful Web サービスのセキュリティ

認証作業の連鎖実行は、Web サーバが Cookie またはトークンを返し、後続の処理作業にその Cookie またはトークンを使用する場合に使用します。連鎖実行は、リクエストされた処理に関連する接続情報で定義されます。Cookie を使用する場合は、認証および実行の両方の接続情報に同一のベース URL を指定する必要があります。

RESTful Web サービスアダプタは、次のセキュリティ機能をサポートします。

- **Cookie** ユーザの認証情報を格納します。この情報は TSCOM エージェントセッションの期間 (ユーザが接続してから切断するまでの間) に限り有効です。Cookie の有効範囲は単一の TSCOM エージェントです。受信した Cookie は、後続の REST リクエストとともに送信元の IP へ返信されます。
- **認証トークン** 認証作業から返される値です。この値は、関連する実行作業に使用されます。

連鎖認証では、Explicit および Password Passthru (PING 機能の使用) セキュリティモデルがサポートされます。

連鎖認証を使用するには、認証接続を宣言する必要があります。次に、認証情報を記述したマスターファイルを作成し、そのファイルをアプリケーションに保存します。続いて、認証シノニムを使用して、関連する 1 つまたは複数の実行接続情報を作成します。

注意：接続情報にユーザ ID とパスワードの情報は含まれていても、連鎖認証が定義されていない場合は、ベーシック HTTP 認証が使用されます。この場合、ユーザ ID とパスワードは x64 アルゴリズムで暗号化された後、HTTP 接続の確立に使用されます。

連鎖実行用のアダプタを構成するには

連鎖認証用のアダプタを構成するには、次の手順を実行します。

手順

1. REST コールによる認証に必要なベース URL を使用して、接続情報を作成します。この接続情報は、認証作業で、URL 拡張子の前に追加する REST URL の一部を指定するために必要です。

WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの [接続パラメータ] ウィンドウで [ベース URL] に値を入力し、[構成] をクリックします。

なお、後続の処理作業にトークンではなく Cookie を使用する場合は、この認証接続情報と手順 3 で指定する実行接続情報には同一のベース URL を指定する必要があります。

2. 関連する認証作業のシノニムを作成します。

WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースからシノニムを作成後、シノニムエディタを開き、次のガイドラインに従ってシノニムを編集します。

認証シノニムには、次の入力項目を含める必要があります。

USERNAME

ユーザ ID の値は、実行作業を定義した接続情報から取得されます。

PASSWORD

パスワードの値は、実行作業を定義した接続情報から取得されます。

認証シノニムには、次のアクセスプロパティの項目を含めることもできます。次の項目は、REST リクエストを記述するセグメントに含める必要があります。

ACCESS_PROPERTY=NEED_VALUE

認証作業に使用する追加パラメータの項目を定義します。デフォルト値は、XDEFAULT パラメータで指定することができます。関連する接続情報に値を指定した場合、デフォルト値はその値で上書きされます。

複数の項目を定義することができます。

次の項目は、REST レスポンスを記述するセグメントに含める必要があります。

ACCESS_PROPERTY=AUTHRESP

認証作業の結果を記述する項目を定義します。正しいレスポンス値を ACCEPT 属性に入力します。複数值を許可する場合は、OR により記述します。複数の項目を定義することができます。ただし、これらの項目のいずれかに受容不可の値が含まれている場合、この認証作業は無効と見なされます。

ACCESS_PROPERTY=AUTHTOKEN

レスポンストークンを格納する項目を定義します。このトークンは、入力値として実行作業に渡されます。指定可能な項目は 1 つです。この項目を定義しない場合、この認証作業には Cookie が返されるものと見なされます。

3. 実行リクエストおよび作成済みの認証作業シノニムの呼び出しに使用するベース URL から接続情報を作成します。

WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの [接続パラメータ] ウィンドウで [ベース URL] に値を入力し、[連鎖認証] のチェックをオンにします。

[認証シノニム] テキストボックス横の [...] (参照) ボタンをクリックし、アプリケーションおよび関連する認証作業を記述したサーバ上のマスターファイルを選択します。

[OK] をクリックします。

4. 関連する実行作業のシノニムを作成するには、プロシジャ内でこれを実行する必要があります。CREATE SYNONYM では、返されるレスポンスを指定するために REST リクエストが実行されるためです。メタデータの作成で使用する有効なレスポンスを生成するためには、このリクエストが正しく認証される必要があります。

以下はその例です。

```

-*Authentication Request
TABLE FILE LOGON
PRINT
    ENTRY.VALUE AS 'CSRF Token'

```

```

RETURNCODE
RETURNDESC
IF USERNAME EQ 'admin'
IF PASSWORD EQ 'admin'
IF ENTRY.KEY EQ 'IBI_CSRF-Token_Value'
ON TABLE HOLD AS CSRFTOKEN
END
-RUN
-READ CSRFTOKEN &&CSRFTOKEN.A100
-*Create Synonym which passes Token
-*Since the Authentication request and Create Synonym run in the same
Agent, Cookies are automatically passed
CREATE SYNONYM restful_web_services/folderlist AT ibfs FOR rs PARMS
'HTTPMETHOD=GET PARAMSURL="IBIRS_action=get&|IBIRS_service=ibfs&|IBIWF_
SES_AUTH_TOKEN=&&CSRFTOKEN"'
CHECKNAMES UNIQUENAMES DBMS REST DROP
-RUN

```

5. 実行作業シノニムには、ACCESS_PROPERTY=AUTHTOKEN を使用して認証トークンを定義した入力項目を含めます (トークンを使用する場合)。

WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースでシノニムエディタを表示し、シノニムを編集します。

関連情報については、[シノニムの作成](#)を参照してください。

連鎖認証の構成

接続文字列

```
ENGINE REST SET CONNECTION_ATTRIBUTES ibfs/,:'http://localhost:8080/ibi_
apps/rs auth:rest/logon'
```

認証作業を記述したマスターファイルの抜粋

```

FILENAME=GETTOKEN, SUFFIX=REST , $
SEGMENT=GETTOKEN, SEGTYPE=S0, $
GROUP=HEADER, ALIAS=Header, ELEMENTS=2, $
FIELDNAME=USERNAME, ALIAS=username, USAGE=A30, ACTUAL=A30,
ACCESS_PROPERTY=(NEED_VALUE), $
FIELDNAME=PASSWORD, ALIAS=password, USAGE=A30, ACTUAL=A30,
ACCESS_PROPERTY=(NEED_VALUE), $
FIELDNAME=__RESPONSE, USAGE=TX80L, ACTUAL=TX,

```

```

ACCESS_PROPERTY=(INTERNAL), $
SEGMENT=RESPONSE, SEGTYPE=S0, SEGSUF=XML , PARENT=GETTOKEN,
POSITION=__RESPONSE, $
FIELDNAME=RESPONSE, ALIAS=getToken00out, USAGE=A1, ACTUAL=A1,
ACCESS_PROPERTY=(INTERNAL), $
FIELDNAME=RESULT, ALIAS=Result, USAGE=A120, ACTUAL=A120,
ACCESS_PROPERTY=(AUTHTOKEN),
REFERENCE=RESPONSE, PROPERTY=ELEMENT, $

```

データ抽出作業を記述したマスターファイルの抜粋

```

FILENAME=FINDDADDRESS, SUFFIX=REST , $
SEGMENT=FINDDADDRESS, SEGTYPE=S0, $
GROUP=HEADER, ALIAS=Header, ELEMENTS=8, $
FIELDNAME=HOUSENUMBER, ALIAS=houseNumber, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='2815', $
FIELDNAME=STREET, ALIAS=street, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='PRAIRIE AVE.', $
FIELDNAME=INTERSECTION, ALIAS=intersection, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
FIELDNAME=CITY, ALIAS=city, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='MIAMI BEACH', $
FIELDNAME=STATE_PROV, ALIAS=state_prov, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='FL', $
FIELDNAME=ZONE, ALIAS=zone, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='33140', $
FIELDNAME=COUNTRY, ALIAS=country, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='US', $
FIELDNAME=TOKEN, ALIAS=token, USAGE=A120, ACTUAL=A120,
ACCESS_PROPERTY=(NEED_VALUE, AUTHTOKEN), $
FIELDNAME=__RESPONSE, USAGE=TX80L, ACTUAL=TX,
ACCESS_PROPERTY=(INTERNAL), $
SEGMENT=RESPONSE, SEGTYPE=S0, SEGSUF=XML , PARENT=FINDDADDRESS,
POSITION=__RESPONSE, $
FIELDNAME=RESPONSE, ALIAS=findAddressResponse, USAGE=A1, ACTUAL=A1,
ACCESS_PROPERTY=(INTERNAL), $

```

認証作業への追加パラメータの受け渡し

認証作業に一連の追加パラメータを渡すには、マスターファイルに XDEFAULT パラメータを追加して、デフォルト値を設定します。

ヒント：WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのシノニムエディタを使用してこのタスクを実行することができます。

この構文例は、次の例を参照してください。

認証作業への追加パラメータの受け渡し

```
ENGINE REST SET CONNECTION_ATTRIBUTES belgR/:'https://secure.securex.be/
HRAWebservice/webservices AUTH:belgium/
directauthenticate DBID="HRADemo01" LANGID="1"'
```

マスターファイルの抜粋

```
FILENAME=DIRECTAUTHENTICATE, SUFFIX=REST , $
SEGMENT=DIRECTAUTHENTICATE, SEGTYPE=S0, $
GROUP=HEADER, ALIAS=Header, ELEMENTS=4, $
FIELDNAME=USERNAME, ALIAS=usrname, USAGE=A30, ACTUAL=A30,
    ACCESS_PROPERTY=(NEED_VALUE), $
FIELDNAME=PASSWORD, ALIAS=pwd, USAGE=A30, ACTUAL=A30,
    ACCESS_PROPERTY=(NEED_VALUE), $
FIELDNAME=DBID, ALIAS=dbId, USAGE=A30, ACTUAL=A30, ACCESS_PROPERTY=(NEED_
VALUE), $
    XDEFAULT='Test',
$
FIELDNAME=LANGID, ALIAS=langId, USAGE=A30, ACTUAL=A30, ACCESS_PROPERTY=(NEED_
VALUE), $
    XDEFAULT='E',
$
FIELDNAME=__RESPONSE, USAGE=TX80L, ACTUAL=TX,
    ACCESS_PROPERTY=(INTERNAL), $
SEGMENT=RESPONSE, SEGTYPE=S0, SEGSUF=XML , PARENT=DIRECTAUTHENTICATE,
    POSITION=__RESPONSE, $
FIELDNAME=RESPONSE, ALIAS=DirectAuthenticateResponse, USAGE=A1,
    ACTUAL=A1, ACCESS_PROPERTY=(INTERNAL), $
FIELDNAME=DIRECTAUTHENTICATERESULT, ALIAS=DirectAuthenticateResult,
    USAGE=A5, ACTUAL=A5, ACCESS_PROPERTY=(AUTHRESP), ACCEPT='true',
    REFERENCE=RESPONSE, PROPERTY=ELEMENT, $
```

RESTful Web サービスのメタデータの管理

サーバが RESTful Web サービスプロバイダにアクセスする場合、そのサーバは各 REST リクエストについて、パラメータを渡す方法およびレスポンスを受信する方法を認識しておく必要があります。ほとんどの RESTful Web サービスでは、Web サービスプロバイダのマニュアルを使用して、送信するパラメータおよび予期されるレスポンスを指定することができます。このため、シノニムの作成プロセスで、パラメータを含む REST リクエストを送信し、Web サービスからのレスポンスを指定することができます。

シノニムの作成

シノニムは、Web サービス処理ごとに一意の論理名（「エイリアス」とも呼ばれる）を定義します。シノニムの使用には次の利点があります。

- シノニムでは、クライアントアプリケーションをリクエストのロケーションと ID に対する変更から分離します。リクエストは、それを使用するクライアントアプリケーションを変更せずに、移動したり名前を変更したりできます。行う必要があるのは 1 つの変更だけです。つまり、サーバ上でリクエストのシノニムを再定義するだけで済みます。
- シノニムは、一時項目 (DEFINE) やセキュリティ機構などのサーバの拡張メタデータ機能をサポートします。

シノニムを作成すると、マスターファイルおよびアクセスファイルが生成されます。これらのファイルは、サーバへの Web サービスリクエストを記述したメタデータファイルです。

1 つのシノニムが 1 つの処理を表します。

注意：各シノニムに、それぞれ特定のプロバイダの Web サービス処理を記述します。処理パラメータを変更した場合は、シノニムを再作成する必要があります。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

RESTful Web サービスのシノニム作成パラメータ

以下のリストは、アダプタのシノニムを作成する際に、値の指定が必要なパラメータ、および実行する関連作業を示しています。これらのオプションが複数のウィンドウに表示される場合もあります。ウィンドウに表示されたボタンをクリックして次のウィンドウに進み、最後に [シノニムの作成] ボタンをクリックします。これで、入力データに基づいてシノニムが作成されます。

REST 処理の選択

次のいずれかの REST 処理を選択します。

- **GET** GET 処理を行います。
- **POST** POST 処理を行います。

サービス URL 拡張子

RESTful Web サービスの特定の機能を実行するために使用するベース URL の拡張子です。

ベース URL は、REST リクエストの実行時にサービス URL 拡張子と連結されます。このパラメータには値が必要です。以下はその例です。

```
postalCodeSearch
```

パラメータ化

選択した場合、サービス URL 拡張子をパラメータ化します。

サービス URL 拡張子のパスの要素ごとに、マスターファイルにフィールドが追加されます。作成されたこれらのフィールド名には、先頭に「ID」の文字、末尾にパラメータ化されるパスの要素に対応する数値が使用されます。

たとえば、サービス URL 拡張子が a/b/c の場合、次のフィールドがマスターファイルに追加されます。

```
FIELDNAME=ID1, ALIAS=id, USAGE=A1, ACTUAL=A1,  
ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='a', $  
FIELDNAME=ID2, ALIAS=id, USAGE=A1, ACTUAL=A1,  
ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='b', $
```

```
FIELDNAME=ID3, ALIAS=id, USAGE=A1, ACTUAL=A1,
ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='c', $
```

アクセスファイルの OBJECT 属性は、マスターファイルに追加され、アンパサンド文字 (&) で囲まれたフィールドを含むパスになります。たとえば、サービス URL 拡張子が a/b/c の場合、次の OBJECT 属性がアクセスファイルに追加されます。

```
OBJECT=&ID1&/&ID2&/&ID3&
```

サービス URL パラメータ

パラメータ名と値の組み合わせです。レスポンスからメタデータを作成するための REST リクエストの作成に使用されます。

入力パラメータ名もメタデータに含まれます。

特定の RESTful Web サービスでは、パラメータおよびパラメータ値によって返されるレスポンスの構造が異なることに注意してください。このため、構造の異なる各レスポンスで異なるシノニムが作成されます。以下はその例です。

```
postalcode=90111&maxRows=10&username=demo
```

ドキュメントサンプルを入力

このパラメータは、POST 処理が選択された場合のみ表示されます。

メタデータを作成するために、REST リクエストの本文を渡すことで、サービス URL パラメータフィールドとともに使用されます。リクエスト本文は、名前と値の組み合わせ、XML、JSON のいずれかの組み合わせで構成されます。以下はその例です。

```
IBIRS_action=put&object=<object _jt="IBFSUserObject" description="Rest
Userid" email="restid@informationbuilders.com" password="rest"
type="User"> <status _jt="IBSSUserStatus" name="ACTIVE"/> </object>&IBIRS_
replace=true
```

カスタムヘッダ

カスタムヘッダの値は二重引用符 (") で囲み、各カスタムヘッダはセミコロン (;) で区切る必要があります。以下はその例です。

```
Content-Type="CDF";PARAM1="ABC"
```

接続文字列でカスタムヘッダを定義すると、カスタムヘッダが REST サービス全体に適用されます。カスタムヘッダは、メタデータレベルでも定義することができます。メタデータレベルでカスタムヘッダを定義すると、カスタムヘッダは特定の REST コールに適用され、WHERE 句または IF 句を使用してレポート実行時にカスタムヘッダの値を変更することができます。

フィールド名の確認

すべての特殊文字をアンダースコア (_) に変換するとともに、予約語の使用を回避するために名前の確認を行う場合は、[フィールド名の確認] のチェックをオンにします。これにより、名前に番号が付加されます。このパラメータにより、規則に一致した名前が付けられます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

[フィールド名の確認] のチェックをオフにすると、ハイフン (-)、ブランク、円記号 (¥)、フォワードスラッシュ (/)、カンマ (,)、ドル記号 (\$) の特殊文字のみがアンダースコア (_) に変換されます。名前の確認は行われません。

ユニークフィールドにする

フィールド名およびグループ名の範囲をシノニム全体に設定する場合、[ユニークフィールドにする] のチェックをオンにします。これにより、シノニムの異なるセグメントであっても重複名は使用されません。このチェックをオフにすると、範囲はセグメントに設定されます。

シノニム名

シノニムに割り当てる名前を入力します。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスはブランクにしておきます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

マスターファイルの属性

属性	説明
PROPERTY	フィールドが XML/JSON 属性または XML/JSON 要素に対応しているかどうかを示します。
REFERENCE	XML/JSON 階層内のフィールドの親要素を識別します。 フォーマットは、次のとおりです。 <i>segmentname.fieldname</i> 説明 segmentname このフィールドが存在するマスターファイルのセグメント名です。 fieldname 親要素に対応するフィールド名です。

アクセスファイルの属性

属性	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 CONNECTION=connection
OBJECT	シノニムの作成プロセスで定義されたサービス URL 拡張子です。
HEADER	REST GET リクエストのパラメータを定義するマスターファイルの GROUP 名です。
SERVICE NAME	サービスタイプは常に REST になります。
HTTPMETHOD	REST リクエストで使用する HTTP メソッドです (例、GET または POST)。
RESTRESPONSE	REST リクエストから返されるレスポンスのタイプです。有効な値は、XML または JSON です。
HTTPBODY	HTTPMETHOD=POST の場合、HTTPBODY=ENCODE に設定すると、リクエストの本文がエンコードされます。
TIMEOUT	Web サービスのタイムアウトを秒単位で示します。

文字フィールドのデフォルトフィールド長の設定

REST SET FIELDLENGTH コマンドに対応するアダプタは、シノニム作成時に使用される文字フィールドのデフォルト USAGE および ACTUAL フィールド長を設定します。

構文は次のとおりです。

```
ENGINE REST SET FIELDLENGTH nnn
```

説明

nnn

デフォルト長さです。

たとえば、次のコマンドは、文字フィールドの USAGE および ACTUAL のデフォルトフィールド長を 100 に設定します。

```
ENGINE REST SET FIELDLENGTH 100
```

POST リクエストの JSON シノニム での XDEFAULT 属性自動挿入

POST シノニムの作成リクエストのドキュメントサンプルに JSON が使用される場合、XDEFAULT 属性がメタデータに自動的に挿入されます。

サンプル REST シノニム

次のシノニムは、以下の接続パラメータおよびシノニム作成パラメータにより生成されたものです。

ベース URL

```
http://api.geonames.org
```

ベース URL は、[接続の追加] ページに入力されます。

それ以外のエントリは、[シノニムの作成] ページに表示されます。

REST 処理の選択

[GET] を選択します。

サービス URL パラメータ

次のパラメータを入力します。

```
postalcode=10121&country=US&maxRows=10&username=demo
```

シノニム名

「postalCodeSearch」と入力します。

次に、[フィールド名の確認] および [ユニークフィールドにする] のチェックをオンにし、[シノニムの作成] をクリックします。

postalCodeSearch マスターファイル

```
FILENAME=M6ILO, SUFFIX=REST      , $
  SEGMENT=M6ILO, SEGTYPE=S0, $
    GROUP=HEADER, ALIAS=Header, ELEMENTS=4, $
      FIELDNAME=POSTALCODE, ALIAS=postalcode, USAGE=A30, ACTUAL=A30, ACCESS_
PROPERTY=(NEED_VALUE),
      XDEFAULT='10121', $
      FIELDNAME=COUNTRY, ALIAS=country, USAGE=A30, ACTUAL=A30, ACCESS_PROPERTY=
(NEED_VALUE),
      XDEFAULT='US', $
      FIELDNAME=MAXROWS, ALIAS=maxRows, USAGE=A30, ACTUAL=A30, ACCESS_PROPERTY=
(NEED_VALUE),
      XDEFAULT='10', $
      FIELDNAME=USERNAME, ALIAS=username, USAGE=A30, ACTUAL=A30, ACCESS_
PROPERTY=(NEED_VALUE),
      XDEFAULT='demo', $
      FIELDNAME=__RESPONSE, USAGE=TX80L, ACTUAL=TX, ACCESS_PROPERTY=(INTERNAL),
$,
  SEGMENT=RESPONSE, SEGTYPE=S0, SEGSUF=XML      , PARENT=M6ILO, POSITION=__
RESPONSE, $
```

```

FIELDNAME=GEONAMES, ALIAS=geonames, USAGE=A1, ACTUAL=A1, ACCESS_PROPERTY=
(INTERNAL),
    PROPERTY=ELEMENT, $
FIELDNAME=TOTALRESULTSCOUNT, ALIAS=totalResultsCount, USAGE=P33,
ACTUAL=A33,
    REFERENCE=GEONAMES, PROPERTY=ELEMENT, $
FIELDNAME=CODE, ALIAS=code, USAGE=A1, ACTUAL=A1, ACCESS_PROPERTY=
(INTERNAL),
    REFERENCE=GEONAMES, PROPERTY=ELEMENT, $
FIELDNAME=POSTALCODE1, ALIAS=postalcode, USAGE=P33, ACTUAL=A33,
    REFERENCE=CODE, PROPERTY=ELEMENT, $
FIELDNAME=NAME, ALIAS=name, USAGE=A55, ACTUAL=A55,
    REFERENCE=CODE, PROPERTY=ELEMENT, $
FIELDNAME=COUNTRYCODE, ALIAS=countryCode, USAGE=A55, ACTUAL=A55,
    REFERENCE=CODE, PROPERTY=ELEMENT, $
FIELDNAME=LAT, ALIAS=lat, USAGE=P20.3, ACTUAL=A20,
    REFERENCE=CODE, PROPERTY=ELEMENT, $
FIELDNAME=LNG, ALIAS=lng, USAGE=P20.3, ACTUAL=A20,
    REFERENCE=CODE, PROPERTY=ELEMENT, $

```

```

FIELDNAME=ADMINCODE1, ALIAS=adminCode1, USAGE=A55, ACTUAL=A55,
    REFERENCE=CODE, PROPERTY=ELEMENT, $
FIELDNAME=ADMINNAME1, ALIAS=adminName1, USAGE=A55, ACTUAL=A55,
    REFERENCE=CODE, PROPERTY=ELEMENT, $
FIELDNAME=ADMINCODE2, ALIAS=adminCode2, USAGE=P33, ACTUAL=A33,
    REFERENCE=CODE, PROPERTY=ELEMENT, $
FIELDNAME=ADMINNAME2, ALIAS=adminName2, USAGE=A55, ACTUAL=A55,
    REFERENCE=CODE, PROPERTY=ELEMENT, $
FIELDNAME=ADMINCODE3, ALIAS=adminCode3, USAGE=A55, ACTUAL=A55,
    REFERENCE=CODE, PROPERTY=ELEMENT, $
FIELDNAME=ADMINNAME3, ALIAS=adminName3, USAGE=A55, ACTUAL=A55,
    REFERENCE=CODE, PROPERTY=ELEMENT, $

```

postalCodeSearch アクセスファイル

```

SEGNAME=M6ILO,
CONNECTION=geonames,
OBJECT=postalCodeSearch,
HEADER=HEADER,
SERVICETYPE=REST,
HTTPMETHOD=GET,
RESTRESPONSE=XML, $

```

REST POST リクエストでのテキストデータおよびバイナリデータの送信

REST アダプタは、ファイルからテキストデータおよびバイナリデータを読み取り、これらを POST リクエストの本文に渡すことができます。

次のアクセスファイル属性は、テキストデータを別の REST リクエストに渡すために使用されます。

```
HTTPBODY=TEXT
```

次のアクセスファイル属性は、バイナリデータを別の REST リクエストに渡すために使用されます。

```
HTTPBODY=BINARY
```

次のアクセスファイル属性は、入力時に読み取られるファイルパスをREST コールへ渡すために使用するフィールドの定義に使用されます。

```
FIELD=DATAFILE, INPUT=FILENAME
```

REST Web サービスリクエストからのバイナリレスポンスの読み取りおよび出力ファイルの保存もサポート対象に追加されました。バイナリ出力を含むフィールドの USAGE および ACTUAL フォーマットは、STRING として定義する必要があります。

次のコマンドは、HOLD バイナリ出力に使用されます。

```
ON TABLE HOLD FORMAT BINFILE DATASET filename
```

次のアクセスファイル属性は、REST リクエストからのバイナリデータの読み取りに使用されます。

```
RESTRESPONSE=BINARY
```

説明

filename

出力を格納するファイルの名前です。

以下はその例です。

```
ON TABLE HOLD FORMAT BINFILE DATASET baseapp/wfretail.xlsx
```

1 つの REST Web サービスリクエストからのバイナリレスポンスを、後続の REST Web サービスリクエストの入力に結合することもできます。バイナリレスポンスを含む fieldname および バイナリ入力の fieldname はともに、USAGE=STRING および ACTUAL=STRING で定義する必要があります。

REST POST リクエストからのバイナリデータの読み取りおよび出力のファイル保存

次のマスターファイルは、GET_LIBRARYVERSION REST コールを記述します。__RESPONSE フィールドは、STRING フォーマットの USAGE および ACTUAL で記述されます。

```
FILENAME=M6ILO, SUFFIX=WFCREST      , $
SEGMENT=M6ILO, SEGTYPE=S0, $
  FIELDNAME=IBIRS_ACTION, ALIAS=IBIRS_action, USAGE=A30, ACTUAL=A30,
  ACCESS_PROPERTY=(NEED_VALUE),
  XDEFAULT='run', $
  FIELDNAME=IBIRS_SERVICE, ALIAS=IBIRS_service, USAGE=A30, ACTUAL=A30,
  ACCESS_PROPERTY=(NEED_VALUE),
  XDEFAULT='ibfs', $
  FIELDNAME=IBIRS_PATH, ALIAS=IBIRS_path, USAGE=A100, ACTUAL=A100,
  ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=__RESPONSE, USAGE=STRING, ACTUAL=STRING,
  ACCESS_PROPERTY=(INTERNAL), $
```

以下は、関連するアクセスファイルです。

```
SEGNAME=M6ILO,
CONNECTION=WIN10HYPER1,
OBJECT=rs,
SERVICETYPE=REST,
TIMEOUT=60,
HTTPMETHOD=POST,
HTTPBODY=ENCODE, $
```

次のリクエストは REST コールを発行し、結果を baseapp アプリケーション内の holdversion.xlsx という名前のファイルに保存します。

```

SET EQTEST=EXACT
TABLE FILE GET_LIBRARYVERSION
PRINT
    __RESPONSE
WHERE IBIRS_PATH EQ
    'IBFS:/WFC/Repository/Binary_Data_TSCQ/L1e7us4uhe01.lib$(1)'
ON TABLE HOLD FORMAT BINFILE DATASET baseapp/holdversion.xlsx
END

```

POST リクエストでのマルチパート/フォームデータの送信

REST アダプタを使用して、POST リクエストでマルチパート/フォームデータのコンテンツタイプの一部を送信することができます。

この場合、シノニム内に次の宣言が必要です。

- マルチパート/フォームデータとして送信する POST リクエストの本文を定義します。

アクセスファイル宣言

```
HTTPBODY=MULTIPART, $
```

結果 - REST リクエストの HTTP ヘッダに「Content-Type: multipart/form-data」を追加します。

- パートヘッダ内の Content-Disposition の値を含むフィールド名を定義します。

例

```
Content-Disposition: form-data
```

アクセスファイル宣言

```
FIELD=part1_content_disposition, FORMAT=PARTHEADER, $
```

マスターファイル宣言

```
FIELDNAME=part1_content_disposition, ALIAS=Content-Disposition,
USAGE=A30, ACTUAL=A30, XDEFAULT='form-data', $
```

- パートヘッダ内の name の値を含むフィールド名を定義します。

例

```
name="file"
```

アクセスファイル宣言

```
FIELD=part1_name, FORMAT=PARTHEADER, $
```

マスターファイル宣言

```
FIELDNAME=part1_name, ALIAS=name,  
        USAGE=A30, ACTUAL=A30, XDEFAULT='file', $
```

- パートヘッダ内の filename の値を含むフィールド名を定義します。

例

```
filename="reportcaster_version.xlsx"
```

アクセスファイル宣言

```
FIELD=part1_filename, FORMAT=PARTHEADER, $
```

マスターファイル宣言

```
FIELDNAME=part1_filename, ALIAS=filename, USAGE=A100, ACTUAL=A100, $
```

- パートヘッダ内の Content-Type の値を含むフィールド名を定義します。
この値を指定しない場合、Content-Type は filename の値の拡張子に基づいて設定されます。

例

```
Content-Type: application/vnd.openxmlformats-  
officedocument.spreadsheetml.sheet
```

アクセスファイル宣言

```
FIELD=part1_content_type, FORMAT=PARTHEADER, $
```

マスターファイル宣言

```
FIELDNAME=part1_content_type, ALIAS=Content-Type,
USAGE=A200, ACTUAL=A200, $
```

REST レスポンスでの CSV フォーマットのサポート

REST アダプタは、XML または JSON Web サービスレスポンスのメタデータを作成することができます。

Web サービスレスポンスが CSV フォーマットの場合、次の手順でレスポンスのメタデータを作成します。

1. REST メタデータの初期アダプタを作成するには、[RESTful Web サービスのシノニム作成パラメータ](#)の説明に従ってパラメータを入力します。
2. 最初のフィールドにドル記号 (\$) を追加して、マスターファイルの SEGMENT=RESPONSE 行をコメントアウトします。以下はその例です。

```
$ SEGMENT=RESPONSE, SEGTYPE=S0, PARENT=M6ILO, POSITION=__RESPONSE, $
```

3. プロシジャを作成、実行し、Web サービスコールのレスポンス全体をファイルに保存します。必須のパラメータ値は、WHERE ステートメントを使用して渡します。

次のサンプルプロシジャは、Web サービスリクエストの CSV レスポンスを、baseapp アプリケーション内の outfile.csv ファイルに保存します。

```
SET END_OF_TEXT=' '
FILEDEF OUTFILE DISK baseapp/outfile.csv
TABLE FILE CSV_RESPONSE
PRINT
    __RESPONSE
ON TABLE HOLD AS OUTFILE
END
```

4. 区切り付きファイル (CSV/TAB) アダプタを使用し、手順 3 で作成した outfile.csv ファイルに対してメタデータを作成します。

区切りファイルに対するメタデータの作成についての詳細は、[Fixed-Format File](#) およ

び **Delimited File のメタデータの管理**を参照してください。

下図は、区切り付きファイルの [シノニムの作成] 画面を示しています。

5. REST マスターファイルに次の変更を加えます。

- a. SEGMENT=RESPONSE ステートメントのコメントを解除し、SEGSUF=DFIX を追加します。
- b. 手順 4 で作成した区切りファイルのマスターファイルから FIELDNAME メタデータをすべてコピーし、これを REST マスターファイルの SEGMENT=RESPONSE 行以下に貼り付けます。

次の例は、これらの編集を加えたマスターファイルのサンプルです。強調表示された SEGMENT=RESPONSE 宣言には SEGSUF=DFIX が追加され、その後に、手順 4 で作成された outfile.csv に対するメタデータから FIELD 定義が追加されています。

```
FILENAME=M6ILO, SUFFIX=REST, $
  SEGMENT=M6ILO, SEGTYPE=S0, $
    GROUP=HEADER, ALIAS=Header, ELEMENTS=1, $
      FIELDNAME=FORMAT, ALIAS=format, USAGE=A30, ACTUAL=A30, ACCESS_
PROPERTY=(NEED_VALUE),
      XDEFAULT='csv', $
        FIELDNAME=__RESPONSE, USAGE=TX80L, ACTUAL=TX, ACCESS_PROPERTY=
(INTERNAL), $
          SEGMENT=RESPONSE, SEGTYPE=S0, PARENT=M6ILO, SEGSUF=DFIX,
POSITION=__RESPONSE,
$
            FIELDNAME=YEAR1, ALIAS=year, USAGE=A152V, ACTUAL=A152VB,
MISSING=ON,
```

```

    TITLE='year', $
FIELDNAME=WEIGHT, ALIAS=weight, USAGE=D33.5, ACTUAL=A64V,
MISSING=ON,
    TITLE='weight', $
FIELDNAME=GENDER, ALIAS=gender, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
    TITLE='gender', $
FIELDNAME=RACE1R, ALIAS=race1R, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
    TITLE='race1R', $
FIELDNAME=HISPANIC, ALIAS=hispanic, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
    TITLE='hispanic', $
FIELDNAME=ETHNIC1R, ALIAS=ethnic1R, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
    TITLE='ethnic1R', $

```

```

FIELDNAME=AGER, ALIAS=ager, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
    TITLE='ager', $
FIELDNAME=MARITAL2, ALIAS=marital2, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
    TITLE='marital2', $
FIELDNAME=HINCOME, ALIAS=hincome, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
    TITLE='hincome', $
FIELDNAME=POPSIZE, ALIAS=popsiz, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
    TITLE='popsiz', $
FIELDNAME=REGION, ALIAS=region, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
    TITLE='region', $
FIELDNAME=MSA, ALIAS=msa, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
    TITLE='msa', $
FIELDNAME=DIREL, ALIAS=direl, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
    TITLE='direl', $

```

```

FIELDNAME=NOTIFY, ALIAS=notify, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
    TITLE='notify', $

```

```

FIELDNAME=WEAPON, ALIAS=weapon, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
TITLE='weapon', $
FIELDNAME=WEAPCAT, ALIAS=weapcat, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
TITLE='weapcat', $
FIELDNAME=NEWCRIME, ALIAS=newcrime, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
TITLE='newcrime', $
FIELDNAME=NEWOFF, ALIAS=newoff, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
TITLE='newoff', $
FIELDNAME=SERIOUSVIOLENT, ALIAS=seriousviolent,
USAGE=I11, ACTUAL=A11V, MISSING=ON,
TITLE='seriousviolent', $
FIELDNAME=INJURY, ALIAS=injury, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
TITLE='injury', $
FIELDNAME=TREATMENT, ALIAS=treatment, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
TITLE='treatment', $
FIELDNAME=VICSERVICES, ALIAS=vicservices, USAGE=I11,
ACTUAL=A11V,
MISSING=ON,
TITLE='vicservices', $
FIELDNAME=LOCATIONNR, ALIAS=locationnr, USAGE=I11, ACTUAL=A11V,
MISSING=ON,
TITLE='locationnr', $

```

6. REST アクセスファイルに次の変更を加えます。

- a. ドル記号 (\$) の前に、RESTRESPONSE=DFIX を挿入します。
- b. SEGNAME=RESPONSE 行を追加します。
- c. 手順 4 で作成した区切りファイル outfile.csv に対するアクセスファイルから、SEGMENT 以下の行をすべて (CONNECTION 以外) をコピーし、REST アクセスファイルに追加した SEGMENT=RESPONSE 行以下に貼り付けます。
- d. RDELIMITER 行を追加します。

この属性の値は、Windows または Linux などの ASCII プラットフォームでは 0x0A になります (RDELIMITER=0x0A)。

この属性の値は、OS400 プラットフォームでは 0x25 になります (RDELIMITER=0x25)。

この属性の値は、z/OS プラットフォームでは 0x15 になります

(RDELIMITER=0x15)。

- e. RDELIMITER 値の後に、カンマとドル記号 (,\$) を追加します。

次の例は、Windows および Linux のプラットフォームでこれらの編集を加えたアクセスファイルのサンプルです。

```
SEGNAME=M6ILO,
CONNECTION=bjsncvs,
OBJECT=personal/2019,
HEADER=HEADER,
SERVICETYPE=REST,
HTTPMETHOD=GET,
RESTRESPONSE=DFIX,$SEGNAME=RESPONSE,
DELIMITER=',',
ENCLOSURE=",
HEADER=YES,
CDN=COMMAS_DOT,
RDELIMITER=0x0A,$
```

JSON 本文での値が存在しないラベルのサポート

REST アダプタは、POST リクエストの JSON 本文内に値が存在しないラベルを渡すことができます。

たとえば、WHERE COUNTRY EQ " は、JSON 本文内で "country": "" と表示されます。

アクセス (.acx) ファイルに OMIT_MISSING=NO を追加すると、JSON 本文に次のプロパティが生成されます。

- マスターファイルで MISSING=ON 定義されたフィールドは、そのフィールドに対して WHERE 句が定義されていない場合、NULL 値を送信します。
- マスターファイルで MISSING=OFF 定義されたフィールドは、そのフィールドに対して WHERE 句が定義されていない場合、数値フィールドではプロパティ "value":0、数値以外のフィールドではプロパティ "value":"" を送信します。

JSON 本文での値が存在しないラベルの送信

次のアクセスファイルは、OMIT_MISSING=NO が設定された JSON POST レスポンスを指定します。

```
SEGNAME=M6ILO,
CONNECTION=geonames,
OBJECT=postalCodeSearchJSON,
SERVICETYPE=REST,
HTTPMETHOD=POST,
RESTRESPONSE=JSON,
OMIT_MISSING=NO,
HTTPBODY=JSON, $
```

マスターファイルには、次のフィールド定義が含まれます。

```
FIELDNAME=ADMINCODE1, ALIAS=adminCode1, USAGE=A55, ACTUAL=A55,
MISSING=OFF, ACCESS_PROPERTY=(NEED_VALUE), $
```

これらの属性を定義することで、JSON 本文には、WebFOCUS レポートリクエストで定義された ADMINCODE1 に対して WHERE 句または IF 句が存在しない場合に、"adminCode1":"" が自動的に含まれます。

WHERE 句でのパラメータ化された URL への複数値の送信

WHERE 句で、REST URL のパラメータとして定義されたフィールドに複数値が含まれる場合、REST サービスへの呼び出しが値ごとに複数回実行されます。

たとえば、次のマスターファイル postalcodesearch1.mas は、デフォルト郵便番号 10024 に関する情報を返します。

```
FILENAME=M6ILO, SUFFIX=REST , $
SEGMENT=M6ILO, SEGTYPE=S0, $
GROUP=HEADER, ALIAS=Header, ELEMENTS=1, $
FIELDNAME=ID1, ALIAS=ID1, USAGE=A30, ACTUAL=A30, ACCESS_PROPERTY=(NEED_VALUE), $
FIELDNAME=POSTALCODE, ALIAS=postalcode, USAGE=A30, ACTUAL=A30,
ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='10024', $
```

```

FIELDNAME=COUNTRY, ALIAS=country, USAGE=A30, ACTUAL=A30,
  ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='us', $
FIELDNAME=MAXROWS, ALIAS=maxRows, USAGE=A30, ACTUAL=A30,
  ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='10', $
FIELDNAME=USERNAME, ALIAS=username, USAGE=A30,
  ACTUAL=A30, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='xxxxxx', $
FIELDNAME=__RESPONSE, USAGE=TX80L, ACTUAL=TX, ACCESS_PROPERTY=(INTERNAL),
$

```

対応するアクセスファイル postalcodeSearch1.acx には、1 つのパラメータが含まれます。

```

SEGNAME=M6ILO,
CONNECTION=geonames,
OBJECT=&ID1&,
HEADER=HEADER,
SERVICETYPE=REST,
HTTPMETHOD=POST,
RESTRESPONSE=JSON,
HTTPBODY=ENCODE, $

```

次のプロシジャは、REST URL の `http://api.geonames.org` に接続し、WHERE 句および 2 つの値が含まれます。

```

SET END_OF_TEXT=''
ENGINE REST SET CONNECTION_ATTRIBUTES
  geonames_ged/,:'http://api.geonames.org'
TABLE FILE baseapp/POSTALCODESEARCH1
PRINT
  __RESPONSE
WHERE ID1 EQ 'postalCodeSearch' OR 'postalCodeSearchJSON'
ON TABLE HOLD AS BASEAPP/OUTPUT
END

```

次の出力が生成されます。出力には、WHERE 句で指定された 2 つの値「postalCodeSearch」および「postalCodeSearchJSON」のレスポンスが含まれています。

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<geonames>
  <totalResultsCount>1</totalResultsCount>
  <code>
    <postalcode>10024</postalcode>
    <name>New York</name>
    <countryCode>US</countryCode>
  </code>
</geonames>

```

```
<lat>40.78645</lat>
<lng>-73.97638</lng>
<adminCode1 ISO3166-2="NY">NY</adminCode1>
<adminName1>New York</adminName1>
<adminCode2>061</adminCode2>
<adminName2>New York</adminName2>
<adminCode3/>
<adminName3/>
</code>
</geonames>

{"postalCodes":[
  {
    "adminCode2":"061",
    "adminCode1":"NY",
    "adminName2":"New York",
    "lng":-73.976385,
    "countryCode":"US",
    "postalCode":"10024",
    "adminName1":"New York",
    "ISO3166-2":"NY",
    "placeName":"New York",
    "lat":40.786446
  }
]}
```

Salesforce.com アダプタの使用

Salesforce.com アダプタの構成方法について説明します。

Salesforce.com アダプタは、Salesforce Object Query Language (SOQL) を使用して、Salesforce オブジェクトへのクエリアクセスを提供します。

Salesforce.com アダプタの構成

アダプタを構成するには、確立する各接続の接続情報および認証情報を指定します。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Salesforce.com の接続パラメータ

以下のリストは、[Salesforce.com の構成の追加] ページに表示される情報の説明です。表示されるオプションは、アプリケーションに応じて異なります。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

SALESFORCE URL

SOAP を使用して Salesforce.com にログインするための URL。技術サポートから別途の指示がない限り、デフォルト値を使用します。

このフィールドは、Explicit セキュリティを選択した場合に表示されます。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **OAuth** HTTP サービスでユーザアカウントへの制限付きアクセスを許可します。

Explicit

以下は、[Explicit] セキュリティの接続属性です。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

OAuth

OAuth セキュリティを使用して Salesforce.com 用アダプタを設定する前に、接続するアプリケーションを登録して構成する必要があります。

接続アプリケーションを認証コードフロー用に設定するには、API (OAuth 設定の有効化) セクションで、次の設定を有効にします。

- 次の [OAuth 範囲] を選択します。
 - いつでも要求を実行 (refresh_token、offline_access)
 - フルアクセス
- [更新トークンフローの秘密が必要] 設定を選択します。
- [Web サーバフローの秘密が必要] 設定を選択します。
- コールバック URL を入力します。

下図は、接続アプリケーションの認証コード設定を示しています。

Manage Connected Apps

Connected App Name
OAuthApp

« Back to List: Custom Apps

Edit Delete Manage

Version 1.0

API Name OAuthApp

Created Date 1/13/2023 6:22 PM

Contact Email [Redacted]

Contact Phone [Redacted]

Last Modified Date 1/13/2023 6:24 PM

Description [Redacted]

Info URL

▼ API (Enable OAuth Settings)

Consumer Key and Secret [Manage Consumer Details](#)

Selected OAuth Scopes Full access (full)
Perform requests at any time (refresh_token, offline_access)

Callback URL
https://login.salesforce.com/
https://na20dev00000124821/oauth20.exe
https://na20dev00000124821/oauth20.exe

Enable for Device Flow ☐

Require Secret for Web Server Flow ☒

Require Secret for Refresh Token Flow ☒

Enable Client Credentials Flow ☐

Introspect All Tokens ☐

Token Valid for 0 Hour(s)

Include Custom Attributes ☐

Include Custom Permissions ☐

Enable Single Logout Single Logout disabled

▼ Initial Access Token for Dynamic Client Registration

Initial Access Token [Generate](#)

以下は、[OAuth] セキュリティの接続属性です。

OAuth 権限付与タイプ

構成可能な OAuth 権限付与タイプには、次の 2 つがあります。

- **認可コード** 認可リクエストから返される認可コードを使用し、[リフレッシュトークン] で更新可能な長期アクセストークンを取得することができます。
- **クライアント認証情報** OAuth 実装でクライアント ID とクライアントシークレットのみを必要とする場合に使用します。

このフィールドは、OAuth セキュリティを選択した場合のみ表示されます。

認可コード

以下は、[認可コード] セキュリティの接続属性です。

クライアント ID

OAuth 認証に使用するクライアント ID です。このフィールドは、OAuth セキュリティを選択した場合のみ表示されます。

クライアントシークレット

OAuth 認証に使用するクライアントシークレットです。このフィールドは、OAuth セキュリティを選択した場合のみ表示されます。

認可 URL

特定のアプリケーションへの OAuth 認可に使用する URL です。

たとえば、Salesforce の API セットの認可 URL は
「<https://login.salesforce.com/services/oauth2/authorize>」です。

トークン URL

特定のアプリケーションへのアクセストークンの取得に使用する URL です。このフィールドは、OAuth セキュリティを選択した場合のみ表示されます。

たとえば、Salesforce の API セットのトークン URL は
「<https://login.salesforce.com/services/oauth2/token>」です。

リフレッシュトークン

OAuth トークンリクエストから返されるリフレッシュトークンです。このフィールドは、OAuth セキュリティおよび認可コード権限付与タイプを選択した場合に表示されます。このトークンは、特定のアプリケーションの REST サービスにアクセスしてレポートを実行する際に、新しいアクセストークンを取得するために使用されます。このトークンを取得するには、[アクセストークンの取得] ボタンをクリックします。

クライアントの資格情報

以下は、[クライアントの資格情報] セキュリティの接続属性です。

クライアント ID

OAuth 認証に使用するクライアント ID です。このフィールドは、OAuth セキュリティを選択した場合のみ表示されます。

クライアントシークレット

OAuth 認証に使用するクライアントシークレットです。このフィールドは、OAuth セキュリティを選択した場合のみ表示されます。

トークン URL

特定のアプリケーションへのアクセストークンの取得に使用する URL です。このフィールドは、OAuth セキュリティを選択した場合のみ表示されます。

HTTP 接続詳細オプション

プロキシサーバ IP アドレス

プロキシサーバの IP アドレスです。このサーバがリクエストを代行受信して、実サーバに転送します。

プロキシポート番号

プロキシサーバのリスナポート番号です。デフォルトポート番号は 80 です。

PROXY HTTPS 相対パス

このチェックをオンにすると、プロキシサーバの構成時に、絶対パスではなくプロキシ HTTPS 相対パスがリクエストにより送信されます。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

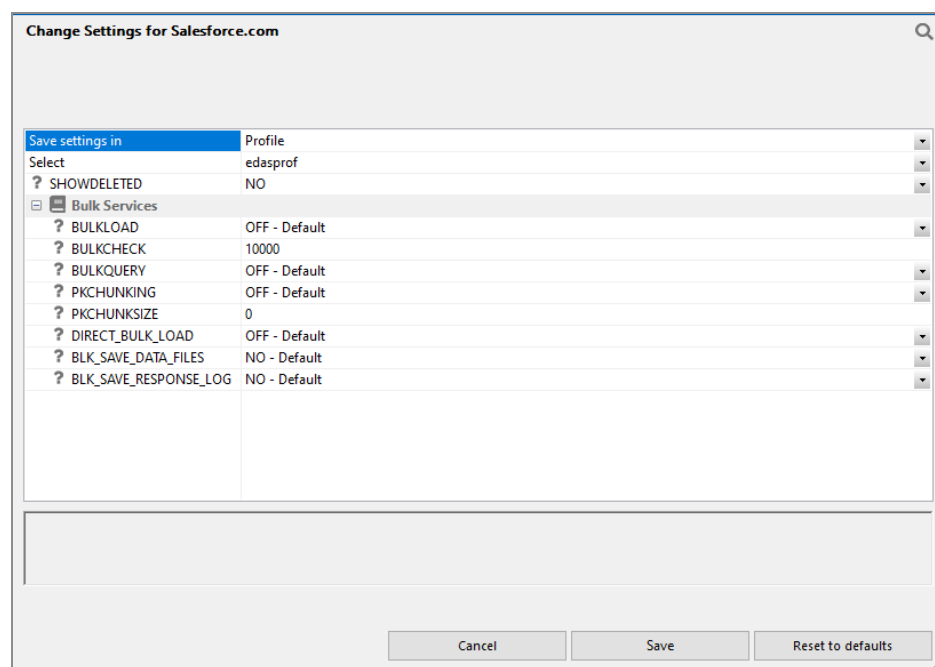
接続属性をサーバプロファイル (edasprof) に保存します。

Salesforce.com による一括取得

Salesforce.com では、大規模データセットのクエリ実行および API リクエスト数の削減を可能にするさまざまな設定が利用できます。

これらの設定は、[Salesforce.com の設定の変更] ダイアログボックスからアクセスすることができます。このダイアログボックスを開くには、ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダを展開し、[Salesforce.com] を右クリックして [設定の変更] を選択します。

下図のように、[設定の変更] ダイアログボックスが開きます。



[バルクサービス] には、次のオプションがあります。

BULKLOAD

拡張バルクロードユーティリティの使用を有効にします。デフォルト値は [OFF] です。

BULKCHECK

データベースに対してトランザクションをコミット (書き込み) する行間隔です。デフォルト値は 10000 です。値が 0 (ゼロ) に設定されている場合、ジョブの完了時に COMMIT が発行されます。

BULKQUERY

データ取得時のバルククエリを有効にします。デフォルト値は [OFF] です。このオプションを選択すると、Salesforce.com の要件 (例、集計なし) を満たすクエリで、バルククエリが使用されます。

次のコマンドを使用して、バルククエリを有効または無効にすることもできます。

```
ENGINE SFDC SET BULKQUERY {ON|_OFF_}
```

PKCHUNKING

データ取得時のバルククエリ PK チャンクを有効にします。デフォルト値は [OFF] です。このオプションを有効にすると、非常に大規模なデータセットのデータが抽出できます。オブジェクトの主キーに基づいてデータがチャンク単位で取得されるため、大規模データの取得が可能になります。

次のコマンドを使用して、大規模データセットの抽出を実行することもできます。

```
ENGINE SFDC SET PKCHUNKING {ON|_OFF_}
```

PKCHUNKSIZE

データ取得時のバルククエリ PK チャンクサイズを有効にします。デフォルト値は 0 (ゼロ) です。この値が設定されていない場合、または 0 (ゼロ) に設定されている場合、Salesforce.com は、デフォルトチャンクサイズの 100000 を使用します。最大チャンクサイズは 250000 です。

次のコマンドを使用して、バルククエリチャンクサイズを有効にすることもできます。

```
ENGINE SFDC SET PKCHUNKSIZE {0..250000}
```

DIRECT_BULK_LOAD

有効にすると、ソースファイルがロードされるデータベースの一括ロードプログラムで期待されるフォーマットになっている場合、ロード時間が短縮されます。ロード時間の短縮は、中間ファイルを作成せず、ファイルを直接ロードすることで実現されます。ダイレクトバルクロードは、アダプタの設定として有効化でき、このアダプタを使用するすべてのフローに適用されます。

BLK_SAVE_DATA_FILES

[Yes] に設定すると、ログ (レスポンス) ファイルとともに、作成される中間ファイルが保持されます。デフォルト値は [No] です。

ファイルは、フローと同一のアプリケーションディレクトリに保存されます。各ファイルは、レポート作成に使用可能なシノニムで生成されます。中間 HTML ファイルのファイルタイプは .ftm、レスポンスファイルは .log です。これらのファイルをバッチの実行結果とともに使用して、ロードが成功したレコード (これらのレコードの内部 ID を含む) およびロードが失敗したレコード (これらのレコードのエラーメッセージを含む) を確認することができます。

BLK_SAVE_RESPONSE_LOG

[Yes] に設定した場合、レスポンスログファイルが保存されます。[BLK_SAVE_DATA_FILES] が [Yes] に設定されている場合、ログファイルは自動的に保存されます。ファイルは、フローと同一のアプリケーションディレクトリに保存されます。ファイルタイプは、.log です。

Salesforce.com でのシノニムの作成

シノニムを作成して、サーバからアクセス可能な各オブジェクトの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

Salesforce.com オブジェクトのシノニムを作成するには

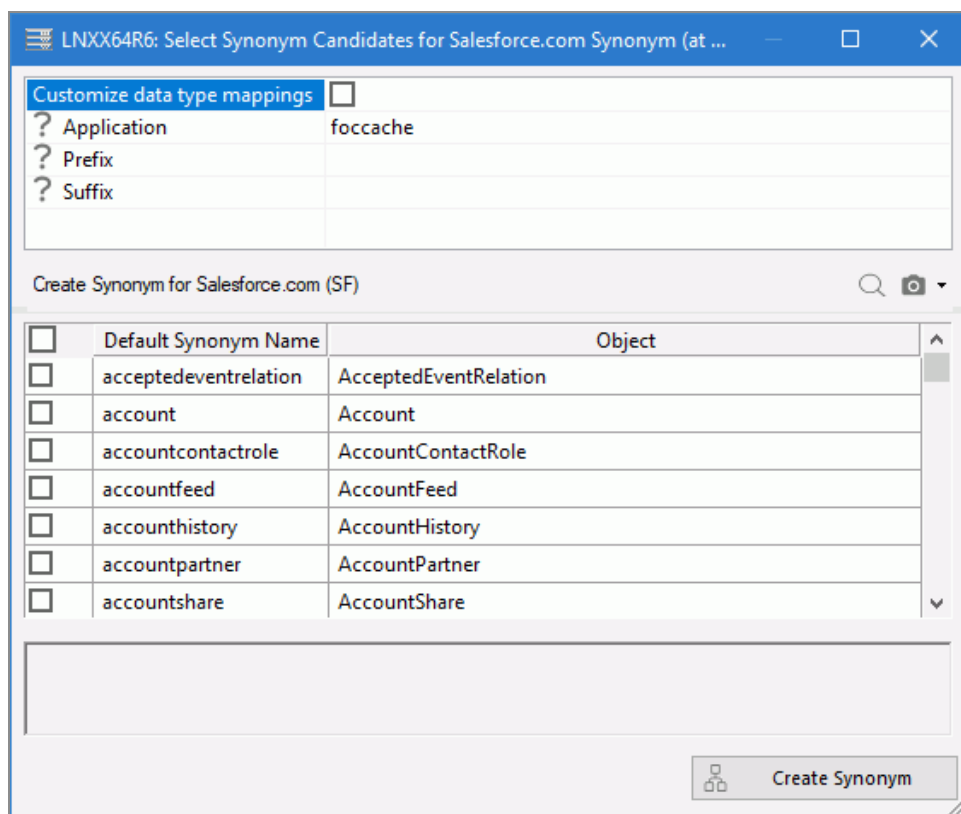
手順

1. ibi Data Migrator デスクトップインターフェースで [アプリケーションディレクトリ] を右クリックし、[新規]、[シノニム (作成または更新)] を順に選択します。

[構成するアダプタを選択するか、シノニムを作成する接続を選択] ダイアログボックスが開きます。

2. 構成済みアダプタの接続を選択し、[OK] をクリックします。

下図のように、一連のシノニムの作成ウィンドウの最初のページが開き、使用可能なすべての Salesforce.com オブジェクトのリストが表示されます。



3. シノニムを作成するオブジェクトのチェックボックスを選択します。
シノニムの名前をデフォルト名から変更する場合は、名前をクリックし、必要に応じて編集します。
4. [シノニムの作成] をクリックします。
[ステータス] ウィンドウに、シノニムが正しく作成されたことを示す情報が表示されます。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムは、[アダプタ] ページで、構成済みアダプタ接続を右クリックし、[シノニムの作成] を選択して作成することもできます。

Salesforce.com オブジェクトのクラスタ JOIN を作成するには

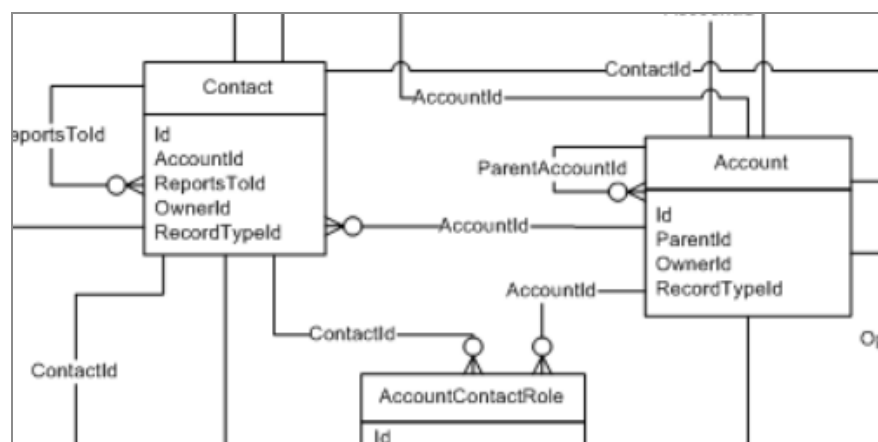
クラスタ JOIN を使用したシノニムでは、複数の Salesforce.com オブジェクトに対するレポートの作成が容易になります。

Salesforce.com のオブジェクト間の関係は、同社 Web サイトの各データモデルに記述されています。使用可能なエンティティリレーションダイアグラム (ERD) リストのデータモデルで検索します。たとえば、Salesforce オブジェクトのモデルには、account、contact、opportunity、およびその他の関連オブジェクトが含まれています。Salesforce オブジェクトモデルの例については、以下の Web サイトを参照してください。

http://www.salesforce.com/us/developer/docs/api/Content/sforce_api_erd_majors.htm

Salesforce.com オブジェクトのクラスタ JOIN を作成する場合は、これをデータモデルのサブセットにする必要があります。これにより、オブジェクト間の JOIN が Salesforce.com で実行され、Salesforce.com のサーバに送信するクエリ数を最小限にすることができます。

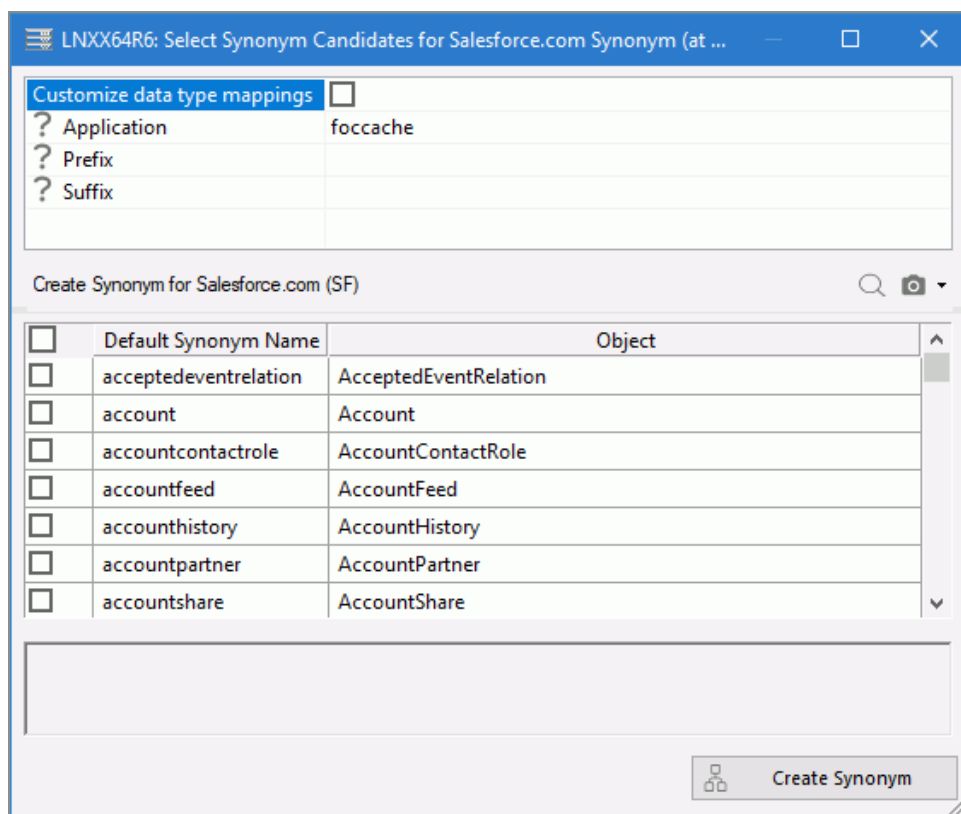
たとえば、連絡先の名前 (ファーストネームとラストネーム) を取得するコンタクトオブジェクトおよび会社名を取得するアカウントオブジェクトからレポートの作成が可能なシノニムを作成する場合を想定します。下図のようなモデルから、Account と Contact との関連性が確認できます。



この関係によるクラスタ JOIN を使用したシノニムの作成手順は次のとおりです。

1. Data Migrator デスクトップインターフェースでアプリケーションディレクトリを右クリックし、[新規]、[シノニム (作成または更新)] を順に選択します。
[シノニムを作成する接続の選択] ダイアログボックスが開きます。
2. Salesforce.com への接続を選択し、[OK] をクリックします。

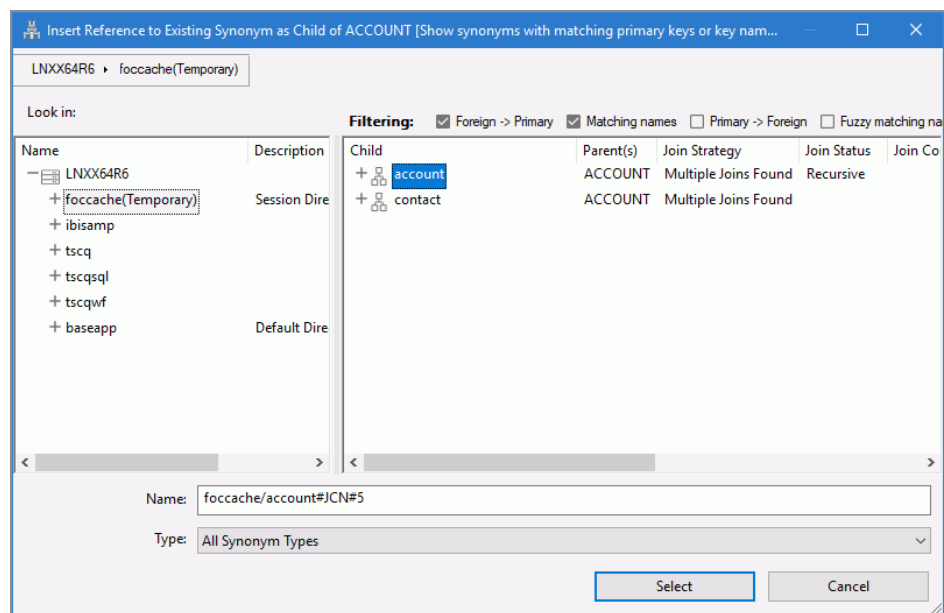
下図のように、[シノニム候補の選択] ダイアログボックスが表示されます。



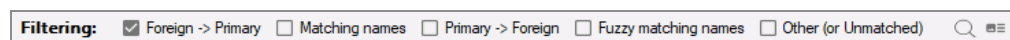
- Account および Contact のチェックボックス、またその他必要なオブジェクトのチェックボックスを選択します。
- [シノニムの作成] をクリックします。
[ステータス] ダイアログボックスが開きます。
- 選択した各シノニムについて、「作成しました。」というメッセージが表示されていることを確認します。
[閉じる] をクリックします。
- アプリケーションディレクトリを右クリックし、[新規]、[シノニムエディタでシノニムを編集] を順に選択します。
ブランクのシノニムが表示されます。
- [モデル表示] タブをクリックします。ワークスペースを右クリックし、[挿入]、[既存シノニムの参照をルートとして挿入] を順に選択します。
- [account] を選択し、[選択] をクリックします。ACCOUNT オブジェクトがワークフローに追加されます。

9. [account] を右クリックし、[挿入]、[既存シノニムの参照を子として挿入] を順に選択します。

下図のように、[既存シノニムの参照を ACCOUNT の子として挿入] ダイアログボックスが表示されます。

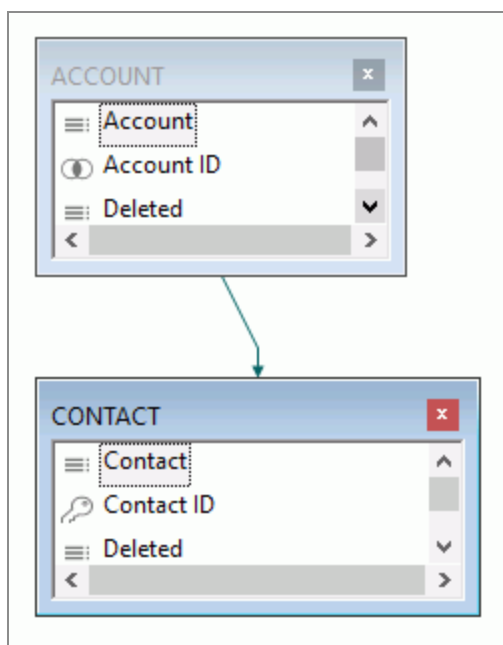


10. 下図のように、[外部 -> 主] 以外の [フィルタ] のチェックをすべてオフにします。

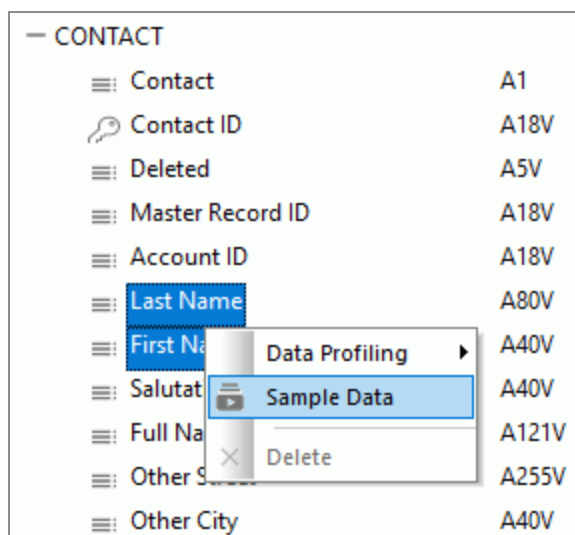


11. [contact] を選択し、[選択] をクリックします。

モデル表示にこの 2 つのオブジェクトが表示されます。



12. ツールバーの [保存] ボタンをクリックします。
[名前を付けて保存] ダイアログボックスが開きます。
シノニムの名前に「account_contact」と入力し、[保存] をクリックします。
13. ワークスペース下部の [フィールド表示] タブをクリックし、表示を切り替えます。
14. [ACCOUNT] セグメント下で [Account Name] をクリックします。次に、下方方向へスクロールして [CONTACT] セグメントを特定し、Ctrl キーを押しながら [Last Name] と [First Name] をクリックします。
15. 下図のように、選択後に右クリックして [サンプルデータ] を選択します。



下図のように、[Account Name]、連絡先の [Last Name] および [First Name] のデータを示すレポートが表示されます。

LNXX64R6: Data for "Account Name"/"Last Name"/"First Name" (foccache...			
	Account Name	Last Name	First Name
1	GenePoint	Frank	Edna
2	United Oil & Gas, UK	James	Ashley
3	United Oil & Gas, Singapore	D'Cruz	Liz
4	Edge Communications	Forbes	Sean
5	Burlington Textiles Corp of America	Rogers	Jack
6	Grand Hotels & Resorts Ltd	Bond	John
7	Express Logistics and Transport	Davis	Josh
8	University of Arizona	Grey	Jane
9	United Oil & Gas Corp.	Green	Avi
10	blue ribbon		.
11	Pyramid Construction Inc.		.
12	New Test Account		.

削除済みレコードの取得の制御

削除済みおよびアーカイブ済みレコードを Salesforce.com から取得するかどうかを制御することができます。

ENGINE SFDC SET SHOWDELETED {YES|NO}

説明

SFDC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

YES

削除済みおよびアーカイブ済みのレコードが、表示レコードに含まれます。

NO

削除済みおよびアーカイブ済みのレコードは取得されません。これがデフォルト値です。

ブロックサイズの制御

1 つのブロックで Salesforce.com から取得するレコード数を制御することができます。

ENGINE SFDC SET BATCHSIZE nnnn

説明

SFDC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nnnn

1 ブロックで取得するレコード数です。デフォルト値は 200 です。最大数は 9999 です。

Salesforce.com の日付リテラルおよび日付定数の使用

SOQL (Salesforce Object Query Language) は、SQL にはない便利な日付コンストラクトを提供します。これらは、SELECT ステートメントのフィルタまたは DB_EXPR 関数を使用する TABLE リクエストで使用できます。

以下の例は、DB_EXPR 関数による TABLE リクエストで使用される日付コンストラクトを示しています。

```
WHERE datefield GT DB_EXPR(LAST_YEAR)
WHERE datefield GT DB_EXPR(LAST_N_YEARS:3)
WHERE (DB_EXPR(CALENDAR_MONTH("datefield")) EQ 1)
```

これらの日付コンストラクトについての詳細は、Salesforce.com マニュアルの「Date Formats and Date Literals」を参照してください。

Salesforce.com による集計の無効化

Salesforce.com アダプタで、日付フィールドに WHERE 句を使用してデータを取得する際に、TABLE リクエストに、Salesforce.com でグループ化が許可されない日付フィールドまたはその他のフィールドタイプの SUM および (GROUP) BY が含まれる場合、データの取得に失敗します。この場合、SOQL エラーが発生します。このエラーを回避するため、集計が Salesforce に送信されないようにすることができます。未集計データが取得され、FOCUS で集計が実行されます。

必要に応じて、Salesforce による集計を無効化するには、次の SET コマンドを使用します。

```
ENGINE SFDC SET OPTIMIZATION {RESET|NOAGGR}
```

説明

RESET

集計を無効化しません。これがデフォルト値です。

NOAGGR

Salesforce で集計が無効化されます。

SAP BW アダプタの使用

SAP BW (Business Intelligence Warehouse) アダプタの構成方法について説明します。

SAP BW アダプタを使用することにより、SAP BW のマルチディメンション (OLAP) モデルを通じて SAP BW のデータソースにアクセスすることが可能となります。このアダプタにより、データまたはアプリケーションリクエストがネイティブの SAP BW ステートメントに変換され、最適化されたアンサーセットがリクエストの送信元プログラムに返されます。

SAP BW 環境の準備

SAP BW のリモート通信には、以下のものが必要となります。

- SAP BW BASIS 6.4 以降、および SAP BW 3.1C 以降。
- SAP RFC/SDK 4.6 以降のサーバシステムへのインストール。

Server では、SAP Application Server と通信を行うために、SAP の RFC SDK が必要になります。RFC SDK の場所は、プラットフォーム固有の検索パスに応じて異なります。

プラットフォームに依存する共有ライブラリ (RFC SDK) 用の検索パスは、以下のとおりです。

- **Windows**
 - 実行可能ファイルの存在するディレクトリ。
 - 現在のディレクトリ。
 - Windows システムディレクトリと Windows ディレクトリ。
 - 環境変数の PATH。
- **AIX**

環境変数の LIBPATH。
- **他のすべての UNIX プラットフォーム**

環境変数の LD_LIBRARY_PATH。

サーバが開始せず、一部の NLS ライブラリが見つからないことを示す場合は、

次の行を試みます。

```
EXPORT NLSUI_7BIT_FALLBACK=YES
```

- **SAP BW 3.5 ユーザ** SAP ノート 11682 は実装されています。必要な環境設定には、以下のものがあります。

すべてのプラットフォームで、次のように設定します (RFCSDK で SAP RFC SDK 共有ライブラリを指定する)。

```
SAP_HOME=$RFCSDK/lib
```

Solaris では、次のように設定します (その他のプラットフォームについては、パスの定義に関する OSS ノート 11682 を参照してください)。

```
LD_LIBRARY_PATH=$SAP_HOME
```

サーバが開始せず、一部の NLS ライブラリが見つからないことを示す場合は、次の行を試みます。

```
EXPORT NLSUI_7BIT_FALLBACK=YES
```

- SAP BW のソースシステムへの TCP/IP 接続。

以下の SAP 権限プロファイルが最低限必要になります。このリストは、ユーザの SAP BW リリースやサイトに固有の権限プロファイルに応じて変わります。

権限オブジェクト、権限フィールド、および必須値のリスト

S_RFC - S_RFC - RFC アクセスの権限チェック

項目	値
ACTVT (アクティビティ)	16 (実行)
RFC_NAME (保護する RFC の名前)	RSAB、RSOB、SYST、BAPI_

項目	値
	IOBJ_GETDETAIL、RSBAPI_IOBJ
RFS_TYPE (保護する RFC オブジェクトのタイプ)	FUGR、FUNC (関数グループ)

S_RS_ADMWB - アドミニストレータワークベンチ - オブジェクト

項目	値
ACTVT (アクティビティ)	03 (照会)、16 (実行)
RSADMWBOB (アドミニストレータワークベンチオブジェクト)	INFOOBJECT

S_RS_COMP - ビジネスエクスプローラ - コンポーネント

項目	値
ACTVT (アクティビティ)	03 (照会)、16 (実行)
RSINFOAREA (インフォエリア)	* (All) または特定のインフォエリア
RSINFOCUBE (インフォキューブ)	* (All) または特定のインフォキューブ
RSZCOMPID (レポートコンポーネントの ID)	* (すべて)
RSZCOMPTP (コンポーネントのタイプ)	CKF、REP、RKF、STR、VAR

S_RS_COMP1 - ビジネスエクスプローラ - コンポーネント - 拡張

項目	値
ACTVT (アクティビティ)	03 (照会)、16 (実行)
RSZCOMPID (レポートコンポーネントの名前 (ID))	* (すべて)
RSZCOMPTP (レポートコンポーネントのタイプ)	CKF、REP、RKF、STR、VAR
RSZOWNER (レポートコンポーネントの所有者 (責任者))	* (すべて)

S_RS_ICUBE - アドミニストレータワークベンチ - インフォキューブ

項目	値
ACTVT (アクティビティ)	03 (照会)
RSICUBEOBJ (インフォキューブのサブオブジェクト)	DATA
RSINFOAREA (インフォエリア)	* (All) または特定のインフォエリア
RSINFOCUBE (インフォキューブ)	* (All) または特定のインフォキューブ

S_RS_ISET - アドミニストレータワークベンチ - インフォセット

項目	値
ACTVT (アクティビティ)	03 (照会)
RSINFOAREA (インフォエリア)	* (All) または特定のインフォエリア

項目	値
RSINFOSET (インフォセット)	* (All) または特定のインフォセット
RSISETOBJ (インフォセットのサブオブジェクト)	DATA

S_RS_ODSO - アドミニストレータワークベンチ - ODS オブジェクト

項目	値
ACTVT (アクティビティ)	03 (照会)
RSINFOAREA (インフォエリア)	* (All) または特定のインフォエリア
RSODSOBJ (ODS オブジェクト)	* (All) または特定の ODS オブジェクト
RSODSPART (ODS オブジェクトのサブオブジェクト)	DATA

複数のシステムへのアクセス

クエリアダプタは、複数の BW システムにまたがる処理が可能です。クエリアダプタでは、システムごとに、クライアント、ユーザ、パスワードから成る BW ログオンが必要となります。このログオンには、以下の要件があります。

- RFC 対応であること。
- アクセスする可能性があるクライアントすべてで権限を付与されていること。

SAP BW での権限は、ローカルのセキュリティポリシーに固有のものです。ほとんどの場合、このマニュアルで記述されている権限のほかに、追加の権限が必要になります。詳細は、SAP BW 管理者に問い合わせてください。

SAP BW アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

アダプタは、SAP BW Application Server に接続する際に、接続情報および認証情報を必要とします。この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。

接続と認証の情報は、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で入力することができます。グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、またはグループプロファイルがアダプタの接続属性で更新されます。

複数の接続を組み込むことにより、複数の SAP BW データソースへの接続を宣言することができます。SAP BW への実際の接続は、その接続を参照する最初のクエリ発行時に確立されます。複数の接続を宣言した場合、最後の接続で指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

SAP BW の接続属性

SAP BW アダプタは、[OLAP] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

クエリアダプタの主な機能は、ユーザリクエストを SAP BW で理解できるコードに変換することです。

システム

12 バイト以内の接続名です。

接続パラメータ

[ロードバランシング] チェックボックスにより、以下のオプションのいずれが表示されるかが決まります。

GROUP

アプリケーショングループの名前です。アプリケーショングループでは、RFC アプリケーションを実行できる Application Server のリストを定義します。R/3 トランザクション SMLG を使用して、アプリケーショングループを参照または変更することができます。

入力文字の大文字と小文字は区別され、またブランクは有意義であることに注意してください。

[ロードバランシング] のチェックがオンの場合にのみ表示されます。

MSGHOST

SAP システム (メッセージサーバ) のホスト名です。

[ロードバランシング] のチェックがオンの場合にのみ表示されます。

R3NAME

SAP システムのシステム ID です。

[ロードバランシング] のチェックがオンの場合にのみ表示されます。

HOST

SAP Application Server のホスト名です。

GWHOST

SAP ゲートウェイプロセスが稼働しているマシンのホスト名です。SAP Application Server が 1 つしかない場合、GWHOST と HOST は同じになります。

SYSNR

SAP システム番号です。この番号は 2 桁の数値です。この番号は SAP 管理者から入手します。

CONNECTION LANGUAGE

SAP BW の Unicode 構成の場合、サーバにログオンする際に使用した言語とは異なる言語を選択することができます (NLS 構成で決定されます)。

[SAP BW Non Unicode の構成の追加] ウィンドウの [CONNECTION LANGUAGE] ドロップダウンリストから、SAP BW への接続に使用する言語を選択します。

注意

- [CONNECTION LANGUAGE] オプションを使用して、サーバへのログインに使用す

る言語以外の言語を指定する場合、[多言語のシノニム生成] オプションは使用できません。

- このオプションが機能するためには、WebFOCUS Reporting Server のコードページが SAP BW サーバのコードページと一致する必要があります。たとえば、WebFOCUS Reporting Server がコードページ 942 (SJIS) で構成されている場合、SAP サーバはコードページ 8000 で構成されている必要があります。
- 要求された接続言語を WebFOCUS Reporting Server のコードページ (表示またはバイト数に関して) または SAP BW サーバのいずれかで使用できるかどうかを検査する方法はありません。

SAP セキュリティモード

[SNC] のチェックをオンにした場合、以下の追加パラメータに値を指定する必要があります。

SNC_LIB

SNC_LIB には、外部セキュリティ製品のライブラリへのパスが含まれます。外部セキュリティ製品のライブラリ、外部ライブラリ、SNC_LIB、または gssapi ライブラリには、SAP が認定した外部セキュリティ製品によって提供される機能が含まれています。

環境変数 SNC_LIB を以下のようにセキュリティ製品のライブラリへのパスを含むように設定します。

```
<drive>:¥path¥to¥your¥snclib.dll
```

SNC_PARTNERNAME

SAP システムの外部名です。これは、SNC 名と呼ばれる外部名称の拡張バージョンです。SNC 名は、次のように入力される名称タイプを指定する外部ユーザ名を接頭辞に提供して作成します。

```
<SNC-name_of_SAP_AppServer>
```

以下はその例です。

```
p/secude:CN=miller, OU=ADMIN, O=SAP, C=DE
```

```
p/krb5:miller@WDF.SAP-AG.DE
```

SNC_QOP

保護レベルを指定します。

SNC_MYNAME

own_snc_name で定義された、イニシエータの SNC 名です。

注意：SNC とロードバランシング ロードバランシングまたはグループログオンでは、宛先の SNC 名をメッセージサーバから動的に取得します。[SNC with load balancing] を使用する場合、以下の追加パラメータを指定する必要があります。

```
SNC_PARTNERNAME=p:unused  
SNC_LIB=<drive>:%path%to%your%sncLib.dll
```

[構成] をクリックします。

次のステップは接続の追加です。[次へ] をクリックして次のステップへ進みます。

一般ユーザログインパラメータ

セキュリティ

SAP BW インスタンスへの接続時にユーザを認証する方法には、次の 3 つがあります。

- **Explicit** ユーザ ID およびパスワードは接続ごとに指定され、認証およびリクエストの実行のために SAP BW に渡されます ([Explicit] を選択すると、USER と PASSWORD のオプションが表示されます)。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID およびパスワードは、認証およびユーザリクエストの実行のために SAP BW に渡されます。
- **Trusted** [SNC] が選択されると自動的に設定されます。

CLIENT

ユーザログオン用の SAP BW Client です。最大 3 バイトです。

USER

ユーザログオン用の SAP BW ユーザ ID です。

PASSWORD

ユーザログオン用の SAP BW パスワードです。最大 40 バイトです。

詳細は、[SAP BW 環境の準備](#)を参照してください。

SAP BW アダプタの混合コードページ環境のサポート

Unicode をサポートする SAP サーバを使用する場合、アダプタを Unicode 用に構成する必要があります。この構成を使用すれば、文字変換テーブルは必要ありません。

ただし、Unicode をサポートしない旧バージョンの SAP サーバを使用する場合、以下の情報が適用されます。SAP BW およびアダプタは、各種プラットフォームにインストールすることができます。SAP BW をアダプタのプラットフォームとは異なるコードページまたは文字セットを使用するプラットフォームにインストールする場合、異なる文字セットを変換するために、データフローの方向ごとに 1 つずつ、合計 2 つの文字変換テーブルを作成する必要があります。たとえば、SAP BW が IBM i の EBCDIC マシンにインストールされ、SAP BW アダプタが Intel Windows の ASCII マシンにインストールされている場合を想定します。この場合、リクエストをサーバから SAP BW に送信するには、ASCII から EBCDIC への変換が必要になります。データを SAP BW から SAP BW アダプタに返送するには、EBCDIC から ASCII への変換が必要になります。ここでは、変換テーブルの作成方法、およびそれらの格納場所について説明します。

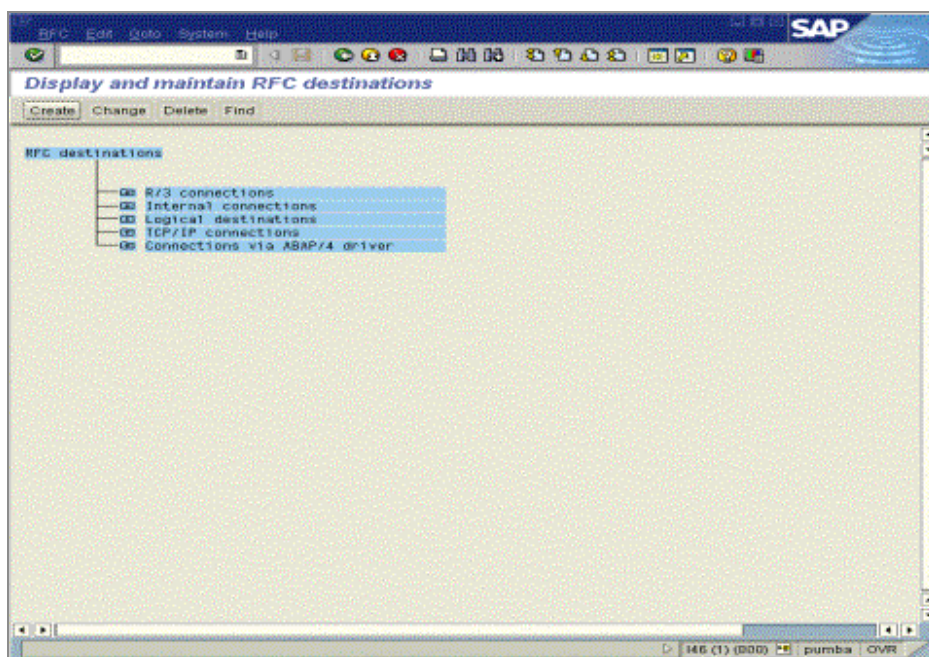
文字変換テーブル

SAP BW には、SAP BW アダプタなどの RFC ベースのアプリケーション用の変換テーブルを生成するためのトランザクション SM59 が用意されています。これらのテーブルを作成した後、アダプタにコピーする必要があります。

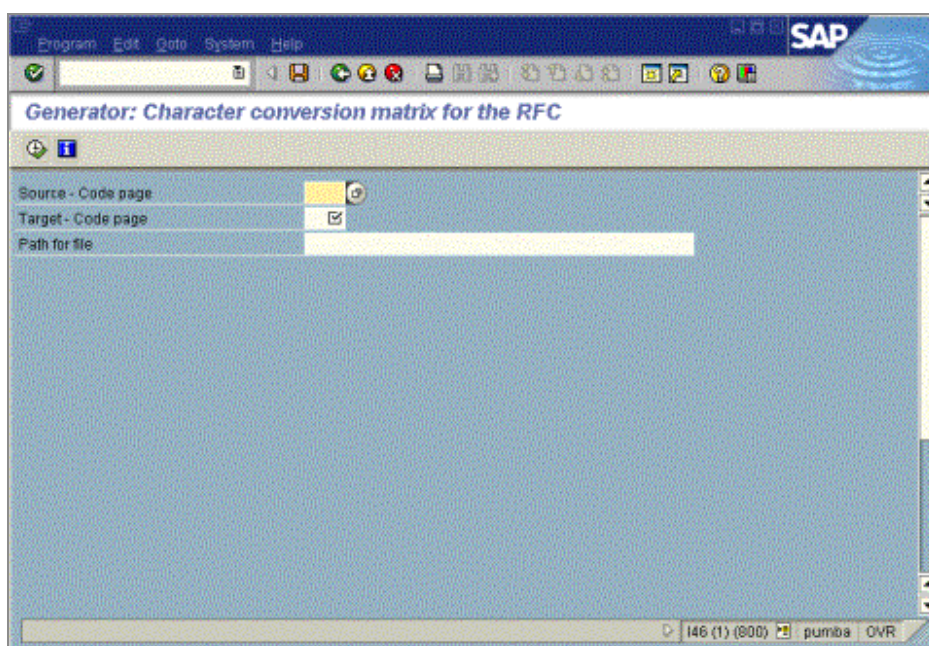
変換テーブルを生成するには

手順

1. トランザクションの SM59 を SAP GUI のコマンド行に入力します。
下図のようなウィンドウが開きます。



2. RFC プルダウンメニューから [変換テーブル生成] タブオプションを選択します。
下図のようなウィンドウが開きます。



3. ソースコードページ、対象コードページ、変換テーブルファイルを作成した Application Server の有効なパスを入力します。オプションで、有効なパスおよび一意のファイル名を指定することができます。

4. [実行]  をクリックします。

一意のファイル名を指定しなかった場合、新たに作成した変換テーブルのデフォルトのファイル名には、ソースおよび対象のコードページの値が使用されます。たとえば、ソースコードページに 0102 (一部の CUA 用の IBM i)、対象コードページに 1101 (7 ビットの USA ASCII pur) を指定した場合、デフォルトのファイル名は 01021101.cdp となります。これは IBM i (EBCDIC) から ASCII への変換テーブルファイルです。

次に、コードページを逆方向に変換するための、もう 1 つの変換テーブルファイルを作成する必要があります。

5. ソースと対象のコードページを入れ替えて、変換テーブルを生成します。

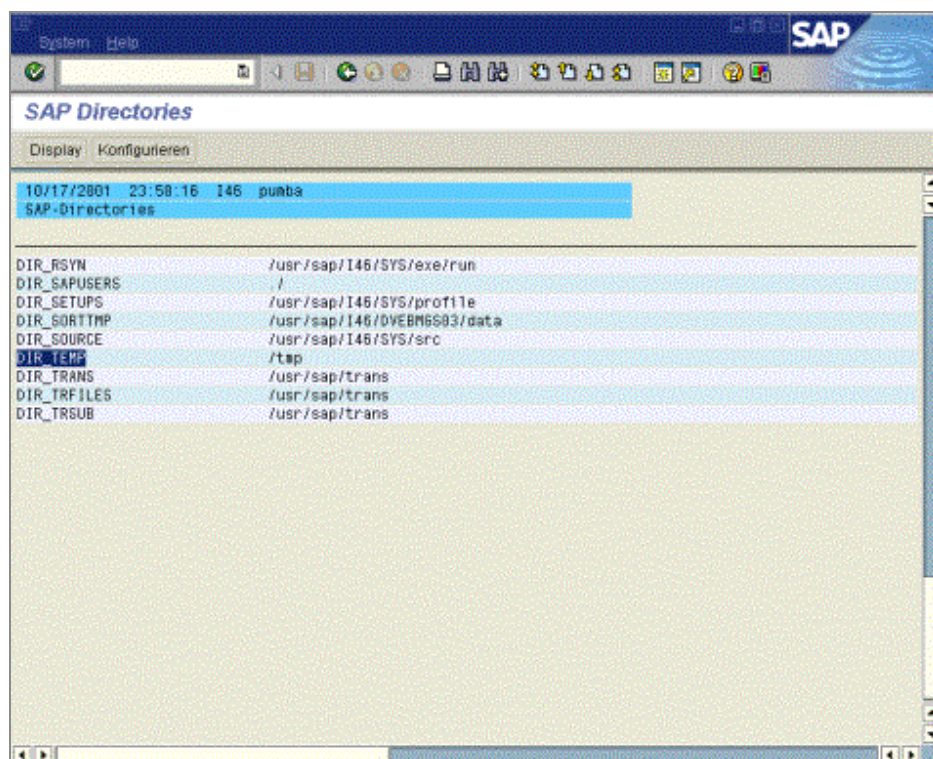
上の例で続けると、ソースコードページに 1101、対象コードページに 0102 を指定した場合、デフォルトのファイル名は 11010102.cdp になります。これは ASCII から IBM i (EBCDIC) への変換テーブルファイルです。

結果

注意：最初の変換テーブルで一意のファイル名を使用した場合、2 番目の変換テーブルにも一意のファイル名を必ず指定してください。

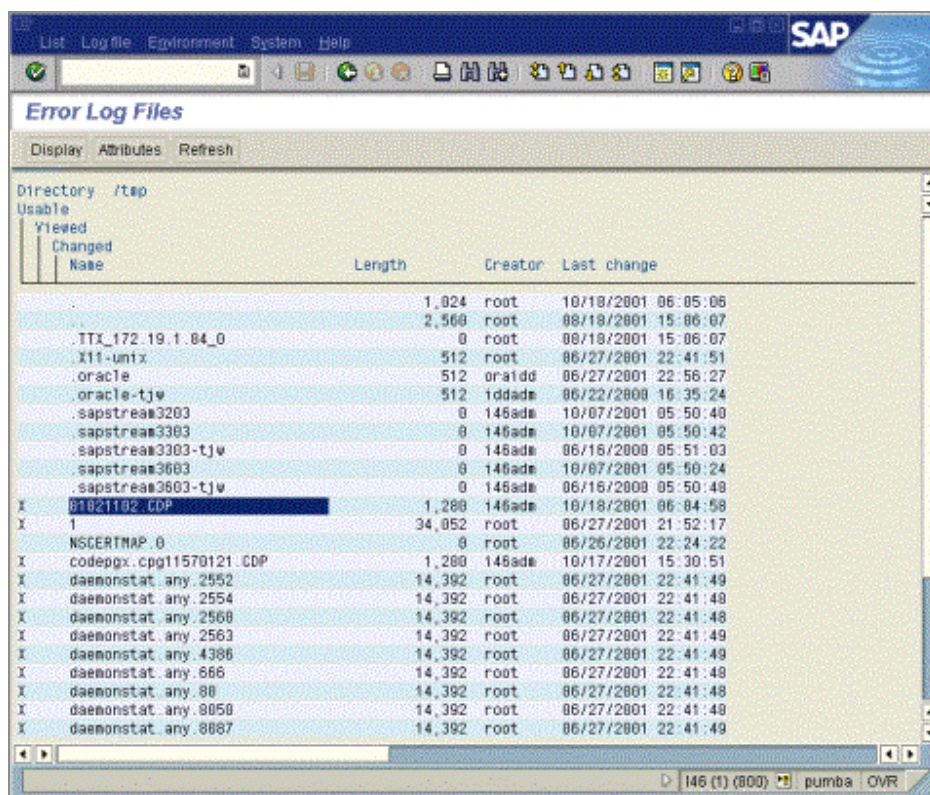
変換テーブルをサーバにダウンロードするには

これらの変換テーブルファイルを作成すると、それらを SAP Application Server のディレクトリからサーバ構成の ETC ディレクトリにダウンロードすることができます。SAP BW のトランザクション AL11 を使用して、2 つの変換テーブルファイルを参照し、サーバがインストールされているデスクトップまたは Windows マシン上のローカルファイルにダウンロードします。変換テーブルファイルをダウンロードするマシンには、SAP GUI が必要になることに注意してください。次のウィンドウは、トランザクション AL11 を示しています。

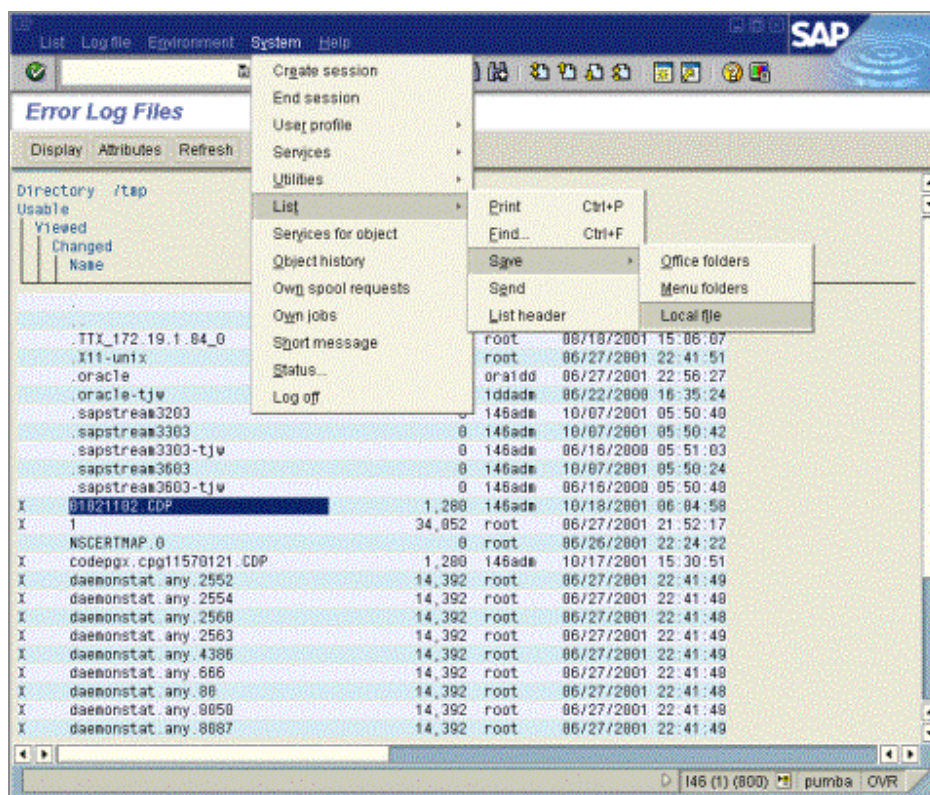


手順

1. 変換テーブルファイルを作成したディレクトリをダブルクリックします。
下図のようなウィンドウが開きます。



2. 変換テーブルファイルをダブルクリックして、内容を表示します。
3. 下図のように、[システム] プルダウンメニューから [一覧] オプションを選択して、[保存]、[ローカルファイル] を選択します。



転送ファイルの形式の入力を要求するダイアログボックスが表示されます。

4. 変換なしの形式を選択して、[続行] をクリックします。
5. 有効なファイル名を入力して、[転送] をクリックします。これにより、ファイルがローカルデスクトップに転送されます。
6. デスクトップで利用できるエディタを使用して、変換テーブルファイルから最初の 3 行を削除します。以下は、削除する必要があるテキストの例です。

```
Directory /tmp
Name:01021102.CDP
-----
```

結果

この編集を行った後、SAP BW アダプタを使用してサーバを構成します。アダプタの構成後、変換テーブルファイルをサーバの EDHOME ETC ディレクトリに移動する必要があります。両方の変換テーブルファイルを必ず移動してください。

BEx クエリの作成

アダプタでは、数種類の SAP BW オブジェクトに対するレポートを作成することができます。それらの一部は、実際のデータのビューまたはサブセットを表しています。SAP のレポート作成および分析ツールとして提供されているビジネスエクスペローラ (BEx) を使用して、クエリ (データ抽出の技法) を作成して、WebFOCUS のレポートリクエストで参照することができます。

クエリとは、特性とキー数値を選択することにより作成されるサブキューブです。クエリを使用すれば、データの対象化と評価を迅速に行うことが可能になります。クエリをより正確に定義するほどサブキューブのサイズは小さくなり、クエリのナビゲートやリフレッシュが高速化されます。一部の特性を選択し他の特性を指定しないと、その一部の特性のより綿密な分析が可能になります。結果のキー数値は、指定されなかった特性の特性値すべてにわたって集約されます。

特性とキー数値をクエリの行と列に配置すると、デフォルトのナビゲーションステータスがクエリ定義に設定されます。

現在要求されているデータだけがクエリに転送されます。OLAP プロセッサでは、データからクエリが作成され、複数のディメンション内のデータをナビゲートする方法が提供されます。クエリでは事前に情報を選択するため、同じデータでもその内容を表示するクエリによって結果が大きく異なる可能性があります。

SAP BW の用語

ビジネスエクスペローラ (BEx) とは、レポートツールと分析ツールを提供する SAP BW のコンポーネントです。BEx クエリでは、データをフィルタして「インフォプロバイダ」と呼ばれるレポートオブジェクトまたはビューを作成します。インフォプロバイダには、実際のデータが含まれているものもあれば、別の場所に保存されているデータのビューを提供するものもあります。

インフォプロバイダの定義に使用される基本構成要素を「インフォオブジェクト」といいます。1 つのインフォオブジェクトは、「得意先」や「売上高」などの業績の分析に関連する 1 つの評価オブジェクトです。インフォオブジェクトとして、特性 (ディメンション)、キー数値 (メジャーまたはファクト)、単位、時間特性、技術特性 (たとえば、依頼番号) があります。

WebFOCUS レポートでは、以下のインフォプロバイダを使用することができます。

- **インフォキューブ** インフォキューブとは自己完結型の業務データセットです。物理的なデータストアは「基本キューブ」といいます。基本キューブはインフォオブジェク

トから構成され、スタースキーマに構造化されます。これは単一のサイズの大きいファクトテーブルに複数のキー数値 (数値データの値) が含まれていることを意味します。このファクトテーブルは、特性 (データのカテゴリ) を格納するより小さいサイズのディメンションテーブルで囲まれています。他のタイプのインフォキューブでは、データセットの論理ビューが表示されます。例をいくつか以下に示します。

- **リモートキューブ** リモートキューブとは、トランザクションデータが BW ではなく外部から管理されるインフォキューブのことです。
- **SAP リモートキューブ** SAP リモートキューブとは、他の SAP システム内のトランザクションデータにアクセスすることにより、クエリを定義できるリモートキューブのことです。
- **サービス付き仮想インフォキューブ** サービス付き仮想インフォキューブのデータは、BW には格納されず、このデータの検索には、ユーザ定義の汎用モジュールが使用されます。
- **ODS オブジェクト** ODS オブジェクトとは、クリーンアップされたトランザクション (詳細レベル) のデータセットです。このオブジェクトには、キーフィールド (たとえば、明細番号) およびデータフィールド (発注状況または得意先など) が含まれています。
- **インフォセット** インフォセットとは、ODS オブジェクトとインフォオブジェクトとの間の結合です。インフォセットとは、オブジェクトの結合方法を決定する条件の定義です。
- **マルチプロバイダ** マルチプロバイダは、異なるインフォプロバイダのデータの組み合わせです。マルチプロバイダにデータは含まれておらず、個々のインフォプロバイダから取得されます。

インフォオブジェクトカタログ アプリケーション固有の編成のインフォオブジェクトで、特性またはキー数値です。

キューブに対してレポートを作成するには、そのキューブをアダプタに定義する BW シノニムを作成しなければなりません。クエリキューブのシノニムを作成するには、最初に BEx クエリを作成します。クエリのプロパティで、[Extended Query Properties] を [Allow external access to this Query] に設定します。

BEx クエリ用語

クエリキューブの基本構成要素は、以下のとおりです。

- **特性** 「ディメンション」とも呼ばれ、「地域」または「時間」などの分類で、ビジネ

パフォーマンスの分析や比較に使用されます。

- **属性** 「プロパティ」とも呼ばれ、特性に関するその他の情報です。たとえば、品目特性には「色」や「重さ」などの属性があります。
- **キー数値** 「メジャー」または「ファクト」とも呼ばれ、業績の評価に使用される定量化可能な値です。キー数値の例としては「売上高」があります。
 - **演算キー数値** これらのキー数値はデータに保存されません。これらは BEx クエリで定義され、実行時に評価されます。
 - **制限キー数値** これらのキー数値は、1 つまたは複数の特性を選択することによってフィルタされます。これらのキー数値は、基本キー数値、演算キー数値、または他の制限キー数値のフィルタされたバージョンにできます。
- **階層** 階層とは、個々の評価基準に従った特性のグループのことです。たとえば、「得意先」の特性は地理的な場所、業種、またはその他の基準に基づいて分類することができます。特性には複数の階層を含めることができます。
- **変数** 変数はクエリのパラメータです。これらはクエリ定義で設定され、値を異なる方法で取得できます。たとえば、デフォルト値が設定される変数もあれば、ユーザが値を入力する必要がある変数もあります。変数についての詳細は、[変数のタイプ](#)および[変数を使用したレポート](#)を参照してください。
- **フィルタ** フィルタは、リクエストの出力を制限するために使用されます。WebFOCUS のシノニムには含まれていませんが、WebFOCUS リクエストの結果に影響を及ぼします。

新規ビジネスエクスプローラクエリの定義

SAP BW インフォプロバイダからレポートを作成するには、SAP ビジネスエクスプローラクエリを定義する必要があります。このクエリは、キューブからのデータ抽出用のテンプレートとして機能します。

注意：これ以降の情報は、SAP BW ビジネスエクスプローラの資料に基づいています。SAP BW BEx の資料は、<http://help.sap.com> から入手することができます。

クエリに対するインフォプロバイダを選択するには

手順

1. 新規クエリを定義するには、ビジネスインフォメーションウェアハウスにインフォプロバイダが少なくとも 1 つ含まれている必要があります。ビジネスエクスプローラアナライザを起動します。
2. BEx ツールバーから [開く] を選択します。
既存のすべてのワークブックの選択画面が表示されます。
3. [クエリ] を選択します。
選択画面に使用可能なすべてのクエリが表示されます。
4. [新規] を選択します。
1 つ以上の新規クエリを定義可能なインフォプロバイダのすべての選択画面が表示されます。
5. クエリで基礎とするデータが格納されているインフォプロバイダを選択します。インフォプロバイダの技術名称を表示するには、[技術名称オン/オフ] アイコンを [オン] に設定します。
選択したインフォプロバイダ内に使用できるオブジェクトが、画面の左部分にディレクトリツリーとして表示されます。

結果

次に、クエリのオブジェクトを選択して、クエリを作成するための適切なボックスにそれらをドラッグします。

クエリを作成するには

BEx 画面の右側には、フィルタ選択、行、列、クエリの自由特性の選択ボックスが表示されます。

以下の手順を実行して、クエリを作成します。

手順

1. クエリを実行するディメンションまたはキー数値の、左側のプラスまたはマイナスの記号をクリックします。
オブジェクトリストが展開され、使用できるキー数値または特性のリストが表示されます。
2. インフォプロバイダの特性とキー数値をクエリ定義の選択ボックスにドラッグアンドドロップします。
これらはフィルタ、行、列、および自由特性のいずれかです。

クエリをフィルタするには

クエリをフィルタして制限することができます。フィルタ選択では、クエリ全体が制限されます。クエリのフィルタに使用するフィールドを選択するには、以下の手順を実行します。

手順

1. インフォプロバイダのオブジェクトリストから、クエリで基礎とする特性またはキー数値を選択します。
注意：フィルタとして選択されたフィールドは定義に使用されるため、SAP BW アダプタのメタデータには表示されません。これらはデータの選別に使用されるため、レポートに出力される情報は含まれません。データを選別してそのデータでレポートを作成したい場合は、[クエリの特性の制限](#)または[キー数値の制限と計算](#)を参照してください。
2. オブジェクトを [フィルタ] ボックスにドラッグします。
3. [フィルタ] ボックス内のオブジェクトを右クリックします。ダイアログボックスが開いて、このオブジェクトに可能なフィルタ定義が表示されます。
4. 単一のメンバー、メンバーの範囲、またはフィルタ用の変数を選択します。

クエリの特性の制限

クエリを定義する際、特性を単一特性値、値の間隔、階層ノード、または特性値変数に制限することができます。

特性を制限するには

手順

1. 値の範囲を選択する特性をインフォプロバイダから選択します。
2. クエリ定義 (行または自由特性) の適切な選択ボックスに特性をドラッグします。
3. 制限する (またはフィルタする) 特性を選択します。マウスの右ボタンを使用して、コンテキストメニューから [制限] を選択します。
4. 特性を単一値、値の間隔、階層ノードに制限するかどうかを選択します。

ヒント：使用する特性値または階層ノードを入力できます。また入力フィールドの右側の拡大鏡をクリックして、指定可能なすべての値のリストを表示することもできます。

5. [OK] をクリックして、入力を確定します。

キー数値の制限と計算

キー数値を特性値、特性値の間隔、階層ノードに制限することができます。たとえば、制限キー数値を第 1 四半期の売上高にすることができます。

クエリ定義のインフォプロバイダのキー数値を制限することもできます。また論理式を使用して、(基本) キー数値から新規キー数値を計算することもできます。

- **制限キー数値** 1 つまたは複数の特性の選択により、制限 (フィルタ) されたインフォプロバイダの (基本) キー数値です。
- **演算キー数値** インフォプロバイダの (基本) キー数値、または作成済みの演算キー数値から成る論理式です。

キー数値を制限するには

手順

1. (基本) キー数値をキー数値選択ボックスにドラッグします。別の方法として、行または列の選択ボックスのヘッダを選択し、マウスの右ボタンを使用してコンテキストメ

ニューから [新規構造] を選択することもできます。

2. [構造] ディレクトリを選択し、マウスの右ボタンを使用してコンテキストメニューから [新規選択] を選択します。[新規選択] 画面が表示されます。
3. 画面の上部のテキストフィールドに、制限キー数値の説明を入力します。
4. 左側のテキストフィールドの下には、インフォプロバイダで使用するすべてのオブジェクトのディレクトリがあります。画面右側の空のフィールドを使用して新規選択を定義します。
5. ドラッグアンドドロップを使用してインフォプロバイダからキー数値を選択し、1 つまたは複数の特性値でそのキー数値を制限します。
6. [OK] を選択します。新規の制限キー数値が構造内に定義されます。

キー数値を計算するには

手順

1. 行または列のディレクトリをハイライト表示し、マウスの右ボタンを使用してコンテキストメニューの [新規構造] を選択して、クエリ定義の行または列に新規構造を作成します。
2. インフォプロバイダの (基本) キー数値を、新規構造のディレクトリにドラッグします。
3. [構造] ディレクトリを選択して、[新規論理式] を選択します。[論理式定義] 画面が表示されます。
4. 画面の上部のテキストフィールドに、論理式の説明を入力します。

結果

注意：論理式の入力フィールドは、テキストフィールドの下にあります。論理式の定義に使用できるオペランドは、すべて画面の左下部にあります。これらは、構造にすでに定義しているキー数値、および変数更新で生成されている、[変数] ディレクトリにあるすべての論理式変数です。

画面の右側には、演算子として使用できる機能があります。これらは、基本的な四則演算、および演算関数 (パーセント関数や三角関数など) を持つディレクトリの記号です。演算子の右側には、数値ブロックがあります。

論理式を定義するには

手順

1. 使用するオペランドを選択して、ダブルクリックまたはドラッグアンドドロップすることにより、それらを論理式の入力フィールドに挿入します。
2. 使用する機能を選択します。この選択は、基本的な四則演算の記号をクリックするか、個々の値をダブルクリックして選択するか、キー数値全体を論理式ボックスにドラッグして行います。
3. 論理式の数値を数値ブロックをクリックして選択します。
4. 論理式を利用できるオペランドおよび演算子を使用して定義します。
オペランドに含まれていない変数を使用する場合は、まずその変数を作成する必要があります。
5. 論理式定義の妥当性を天秤のアイコンを押してチェックします。
6. 説明ボックスに論理式列の名前を入力します。
7. [OK] を選択します。新規の演算キー数値が構造内に定義されます。

クエリプロパティの表示および OLAP へのリリース

- クエリのプロパティを表示するには、ツールバーの [クエリのプロパティ] アイコンをクリックします。[クエリのプロパティ] ダイアログボックスが表示されます。
- クエリを OLAP にリリースするには、ツールバーの [クエリのプロパティ] アイコンをクリックして、[クエリのプロパティ] ダイアログボックスの以下の項目をチェックします。
 - **外部アクセスを許可** (SAP BW リリース 3.x 以降)。
 - **OLE DB for OLAP にリリース** (SAP BW リリース 3.x よりも前)。

これにより、クエリをレポートの目的で QUERY_CUBE として表示することが可能になります。クエリの各要素 (階層レベル、メジャー、変数、およびプロパティ) は、シノニムを作成するために、対応する OLAP の各要素にマッピングされます。

WebFOCUS による SAP BW レポートの作成

ここでは、SAP BW および SAP BW アダプタのレポート作成の概念について説明します。

SAP BW レポートの概念の概要

マルチディメンションのデータソース (キューブ) では、地域や時間などのデータカテゴリを「ディメンション」(SAP BW では「特性」) として表現し、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。階層構造では、各メンバーはノードとして表現され、階層全体がツリー形式で表示されます。

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」(SAP BW では「キー数値」) と呼ばれ、このメジャーを使用してビジネスパフォーマンスを数値で表します。

マルチディメンションのキューブは、「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。

たとえば、次の組み合わせ (タプル) には、2005 年のイギリス (England) の全店舗におけるポンプ (Pumps) の売上高 (Sales Volume) のデータが含まれます。

```
{Sales Volume, Pumps, England, 2005}
```

キューブ内で、この集計値が格納されている点を「セル」と呼びます。各セルには、1 つのメジャーと各種ディメンションのメンバーの組み合わせに基づいてアドレスが割り当てられ

ます。この例では、Sales Volume がメジャーで、「Product」、「Region」、「Time」というディメンションのメンバーがそれぞれ Pumps、England、2005 です。

通常、個々のファクトレコードは、キューブの各階層のリーフノードに関連付けられます。ファクト値は、個々のリーフノードに基づいてアドレスが割り当てられたセルに格納されるとともに、キューブの各階層のリーフノードより上位にあるすべての組み合わせに基づいてアドレスが割り当てられたすべてのセルに挿入されます。

メジャーのファクト集計演算には、単純集計または平均などの複雑な集計関数を使用することができます。

一部の階層ノードの組み合わせに、ファクトレコードが関連付けられていないことがよくあります。これらの組み合わせでアドレスが割り当てられたセルは、ブランクのセルになります。

前述のように、タプルはキューブ内の各種ディメンションに属するメンバーの組み合わせです。前述のタプルにはすべてのディメンションのメンバーが格納されているため、アドレスはセル単体に割り当てられます。いくつかのディメンションのメンバーのみがタプルに格納されている場合、アドレスはセル単体ではなく、キューブ内のスライス全体に割り当てられます。たとえば、次のタプルには、Region および Time ディメンションのメンバーはいずれも含まれていません。

```
{Sales Volume, Pumps}
```

このタプルでは、キューブ内の Region ディメンションのメンバー数と Time ディメンションのメンバー数の積と同数のセルにアドレスが割り当てられます。

キューブ内で、1つのディメンションメンバーに基づいてアドレスが割り当てられるセル数は、他のすべてのディメンションのカーディナリティの積に等しくなります。

1つのメンバーに基づいてアドレスが割り当てられたセルがすべてブランクの場合、このメンバーは「ブランクのメンバー」と呼ばれます。1つのタプルに基づいてアドレスが割り当てられたセルがすべてブランクの場合、このタプルは「ブランクのタプル」と呼ばれます。

デフォルト設定では、ブランクのセルはレポート出力に表示されません。ブランクのセルをレポート出力に表示するには、次のコマンドを発行します。

```
ENGINE BWBAPI SET EMPTY ON
```

BW キューブには、変数を含めることもできます。変数はデータの選択時のパラメータです。変数を表示したりソートに使用したりすることはできません。変数は選択条件でのみ使用できます。変数は、必須値またはオプション値として定義することができます。変数を必須値として定義する場合は、この変数を使用した WHERE 条件をリクエストに含める必要があります。WHERE 条件には、変数の選択タイプに応じた制限を定義します。これには、等価条件のテスト値や、FROM/TO または GE/LE による間隔などがあります。

アダプタがマルチディメンションのキューブにアクセスする場合、そのアダプタが使用するディメンションのメタデータ要素には次の 2 種類があります。

- **階層フィールド** このフィールドには、特定の階層に適用するデータが格納され、このデータに基づいて各メンバーの階層内での位置が定義されます。たとえば、これらのフィールドはメンバーの親、フィールド名、キャプション、ユニーク ID、レベル番号を識別します。
- **ディメンションプロパティ** このフィールドには、ディメンションのすべてのメンバーに適用するデータが格納されます。たとえば、Region ディメンションの「GEOGRAPHICAL_HEIGHT」というプロパティは、Region ディメンションの各メンバーの標高を表します。

アダプタは、階層フィールドを使用して階層を再作成し、リクエストで要求された階層の必要部分を検出することができます。

カラムレポートおよび階層レポートの理解

シノニムには、レベル階層および親子階層という 2 つのタイプの階層を表示できます。

シノニムは、各レベルに異なるフィールドを使用してレベル階層を定義します。レベル階層でレポートを作成するには、カラムレポートを使用します。このレポートでは、表示する各レベルのフィールド名を指定します。

シノニムは、階層構造を定義するフィールド群および階層メンバー間の関係を使用して親子階層を定義します。アダプタには、親子階層でレポートを作成するための特別な階層レポート構文が用意されています。

注意：シノニムを作成する際、親子階層をこのような方法で記述するか、レベルごとに別のフィールドを使用してレベル階層として記述するかを選択できます。

階層レポートを使用すると、レベル番号を特定することなく親子階層のメンバーのソートや選択が行えます。

階層レポートリクエストは、いくつかの段階を経由して出力結果を表示します。

階層のソートおよびメンバーの選択

第 1 段階として、表示する階層メンバーを選択します。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。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。

ディメンションプロパティとメンバーは、ディメンションのリーフノードに関連付けられています。このため、表示する階層レベルの選択に使用することはできません。

ディメンションデータの選別

階層メンバーの選択後、選択したメンバーに WHERE テストを適用することで、検索されたディメンションデータを選択することができます。

WHERE 条件はリーフノードに適用され、階層メンバーを選択する段階が終了した後で処理されます。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。

集計値による選別

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。WHERE TOTAL で選別すると、レポート処理の集計句が終了するため、選別されたデータを計上するためにレポートの合計は再計算されず、単に行が削除されます。

階層レポートの要件

階層レポートでは、特別なメタデータ属性およびレポート構文を使用します。その要件は次のとおりです。

- 親子関係として階層を定義するシノニムを作成する。
- リクエストに階層レポート構文を使用して、レポート出力に表示する階層構造の作成とフォーマット設定を自動的に実行する。

シノニムの作成についての詳細は、[シノニムの作成](#)を参照してください。階層レポートについての詳細は、[階層レポート](#)を参照してください。

シノニムの階層表現

ディメンションは、階層構造を形成します。たとえば、「時間」というディメンションは、「年」、「四半期」、「月」という階層 (親子) 関係を形成することができます。この場合、

月ごとのメジャーが四半期の値として集計され、四半期の値が年の値として集計されます。階層の各点は「ノード」と呼ばれます。子を持たない最下位のノードは「リーフノード」と呼ばれます。

OLAP モデルに準じて、各ディメンションには「フラット階層」と呼ばれる階層が 1 つ存在します。フラット階層には、「ALL」という名前のルートノードの子として、ディメンションのすべてのリーフノードが格納されています。なお、シノニムでは「ALL」というフィールド名は割り当てられていません。シノニムでは、ディメンションと同一の名前がフラット階層に割り当てられます。

シノニムでは、次のいずれかの方法で階層を定義します。

- **レベル階層** 階層レベルは、それぞれ異なるフィールド名で定義されます。レポート出力に階層レベルを表示するには、レポートリクエストでフィールド名を参照する必要があります。

レベル階層を使用するシノニムを作成する場合、シノニムにはレベルごとに 1 つのフィールド宣言が記述されます。この宣言は、フィールドの親も特定します。新しいレベルを追加するためにデータを変更した場合は、シノニムを再作成して、追加したフィールドと親の参照を含める必要があります。

- **親子階層** この階層を定義する一連のフィールドには、階層内の各メンバーの位置を定義するプロパティ値が格納されます。たとえば、メンバーのユニーク ID、レベル番号、親、親メンバーのレベル番号情報を持つフィールドがあります。レポートリクエストを発行する場合は、階層フィールドのいずれか 1 つを指定します。

親子階層では、シノニム内の 1 組のフィールド群で階層を定義します。レベル番号が変更されてもシノニムに変更を加える必要はありません。

さらに、親子階層では、アダプタが階層の一部を再作成し、フォーマットを設定できるという利点もあります。リクエストでレベル番号を指定する必要はありません。

注意：フラット階層は、常に単一レベルのレベル階層として定義されます。

ディメンションのプロパティは、ディメンション内のすべての階層に適用され、シノニム内ではディメンションの全階層の後に表示されます。以下のほとんどの例では、OSD_C01/ZTSCQ31CQ1 キューブを使用して作成された ZOPT および ZLEVEL という名前のシノニムを使用します。

ディメンション宣言

各ディメンションは、それぞれのディメンションおよびその階層を定義するディメンションレコードから開始します。ディメンション自体のレベルはゼロ (0) です。次の例では、シノニムの一部として Material Class 階層のみが選択されています。ただし、ディメンションと同名のフラット階層が常に存在します。フラット階層には、リーフノードのすべてのメン

バーリストが格納され、常にレベル階層として定義されます。そのため、ここでは [0MATERIAL] ディメンションに 2 つの階層が存在することになります。

ディメンション

```
DIMENSION=[0MATERIAL], CAPTION='Material', $
```

ディメンションと同一キャプションのフラット階層

```
HIERARCHY=[0MATERIAL], CAPTION='Material'... $
```

Material Class 階層

```
HIERARCHY=[0MATERIAL                                001], CAPTION='Material class' ...
```

レベル階層の定義

階層各レベルには MATERIAL_CLASS などの階層名とレベル番号で構成されるフィールド名が割り当てられます。また、各フィールド宣言は、WITHIN 属性を使用して親フィールド名を指定します。このフィールドに格納される値は、メンバーのキャプション (TITLE) です。

レベル 1 フィールドの MATERIAL_CLASS_LEVEL_01 では、親は MATERIAL_CLASS ディメンションになります。

レベル 2 フィールドの MATERIAL_CLASS_LEVEL_02 では、親は MATERIAL_CLASS_LEVEL_01 ディメンションになります。

データに新しいレベルを追加する場合は、シノニムを再作成し、このレベルを定義する必要があります。各レベルのカーディナリティは、メンバー数です。

次のフィールドはレベル 3 を記述します。この親はレベル 2 です。

```
$ [0MATERIAL                                001].[LEVEL03] Cardinality 86
  FIELDNAME=MATERIAL_CLASS_LEVEL_03, ALIAS=LEVEL03, USAGE=A40, ACTUAL=A40,
  MISSING=ON,
  TITLE='Material class',
  WITHIN=MATERIAL_CLASS_LEVEL_02,
  PROPERTY=CAPTION, $
```

次のフィールドは、フラット階層を表します。この階層には、2 つのフィールドが関連付けられています。最初のフィールドにはメンバーのキャプションが含まれ、2 番目のフィールドにはメンバー名が含まれています。

```

FIELDNAME=MATERIAL_LEVEL_01, ALIAS=LEVEL01, USAGE=A40, ACTUAL=A40,
MISSING=ON,
TITLE='Material Member Caption',
WITHIN='*[0MATERIAL]',
PROPERTY=CAPTION, $
FIELDNAME=MATERIAL_NAME, ALIAS=MEMBER_NAME, USAGE=A18, ACTUAL=A18,
MISSING=ON,
TITLE='Material Member Name',
REFERENCE=MATERIAL_LEVEL_01, PROPERTY=NAME, $

```

親子階層の定義

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。階層フィールドの名前は、階層名およびそれに付加した接尾語 (SUFFIX) で構成されます。

たとえば、「MATERIAL_CLASS」という階層のキャプションがフィールドに格納され、フィールド名が MATERIAL_CLASS_CAPTION、プロパティ属性が PROPERTY=CAPTION になります。

下表は、階層フィールドの説明です。

データの説明	PROPERTY=	フィールド接尾語 (SUFFIX)
メンバーのユニーク ID (キューブ内で一意)	UID	なし
メンバー名 (階層内で一意)	NAME	_NAME
メンバーのレベル番号	LEVEL_NUMBER	_LVLNO
メンバーの親	PARENT_OF	_PARENT
親のレベル番号	PARENT_LEVEL_NUMBER	_PARENT_LVLNO
子の数	CHILDREN_	_CHILDREN_CARD

データの説明	PROPERTY=	フィールド接尾語 (SUFFIX)
	CARDINALITY	
メンバーのキャプション (レポートのタイトル)	CAPTION	_CAPTION

MATERIAL_CLASS 階層に対する次の宣言では、メンバーのユニーク ID (PROPERTY=UID) を格納するフィールドを定義しています。

```
FIELDNAME=MATERIAL_CLASS, USAGE=A143, ACTUAL=A143,
MISSING=ON,
TITLE='Material class',
WITHIN='*[@MATERIAL                                001]',
REFERENCE=[@MATERIAL], PROPERTY=UID, $
```

MATERIAL_CLASS 階層に対する次の宣言では、メンバーのタイトル (PROPERTY=CAPTION) を格納するフィールドを定義しています。

```
FIELDNAME=MATERIAL_CLASS_CAPTION, USAGE=A60, ACTUAL=A60,
MISSING=ON,
TITLE='Material class CAPTION',
REFERENCE=MATERIAL_CLASS, PROPERTY=CAPTION, $
```

各親子階層には、レポートの可読性を向上させる目的で、上記以外に 3 つのフィールドが定義されます。

- キー (フィールド名の接尾語 _KEY)。このキーは、メンバー名の第 1 トークンです。
- 1 つの空白で区切られたキーおよびキャプション (フィールド名の接尾語 _KEY_CAP)
- 1 つの空白で区切られたキャプションおよびキー (フィールド名の接尾語 _CAP_KEY)

たとえば、MATERIAL_CLASS 階層では、一時項目 (DEFINE) に次の名前が付けられます。

- MATERIAL_CLASS_KEY
- MATERIAL_CLASS_KEY_CAP
- MATERIAL_CLASS_CAP_KEY

ディメンションプロパティ

ディメンションプロパティ (SAP BW の「属性」) は、すべてのディメンション階層の後に記述されます。ディメンションプロパティは、シノニムで PROPERTY=UDA (ユーザ定義属性) と記述されます。たとえば、次のフィールドは、メンバーの重量プロパティを示しています。

```
FIELDNAME=NET_WEIGHT__KEY_, ALIAS='20NET_WEIGHT', USAGE=A9, ACTUAL=A9,
MISSING=ON,
TITLE='Net weight (Key)',
REFERENCE=[0MATERIAL], PROPERTY=UDA, $
```

レベル階層を使用したリクエストのサンプル

レベル階層に対するレポートリクエストでは、レポートで使用する階層の各レベルのフィールド名を指定する必要があります。たとえば、次のリクエストは、Material Class 階層のレベル 1 からレベル 3 に対する売上高のメジャー (Sales Volume) を表示します。

```
TABLE FILE ZLEVEL
SUM SALES_VOLUME
BY MATERIAL_CLASS_LEVEL_01 AS 'LEVEL1'
BY MATERIAL_CLASS_LEVEL_02 AS 'LEVEL2'
BY MATERIAL_CLASS_LEVEL_03 AS 'LEVEL3'
ON TABLE COLUMN-TOTAL
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
GRID=OFF, $
END
```

出力結果は次のとおりです。

<u>LEVEL1</u>	<u>LEVEL2</u>	<u>LEVEL3</u>	<u>Sales Volume</u>
Products	Computer systems	Computer accessories	67,967,525.71
		Computer hardware	17,874,768.15
	Paints / aux. and operating supplies	Paints	.00
	Pumps	Pumps (complete)	59,367,205.93
TOTAL			145,209,499.79

階層全体の合計を計算するには、レポートで ON TABLE COLUMN-TOTAL コマンドを使用するか、BY 句を指定せずに SUM 句を追加します。これにより、レポート出力にカラムが追加されます。

親子階層を使用したリクエストのサンプル

親子階層に対するレポートリクエストでは、BY HIERARCHY 句を使用して階層全体をレポートの対象にすることができます。出力結果には適切なインデントが自動的に設定され、階層レベルと関係が示されます。たとえば、次のリクエストは、Material Class 階層の 3 つのレベルに対する売上高のメジャー (Sales Volume) を表示します。

```
TABLE FILE ZOPT
SUM SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
SHOW TO DOWN 3
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
GRID=OFF, $
END
```

出力結果は次のとおりです。

<u>Material class Member Caption</u>	<u>Sales Volume</u>
Material class	145,209,499.79
Products	145,209,499.79
Computer systems	85,842,293.86
Computer hardware	17,874,768.15
Computer accessories	67,967,525.71
Paints / aux. and operating supplies	.00
Paints	.00
Pumps	59,367,205.93
Pumps (complete)	59,367,205.93

レポートリクエストで特定の階層レベルを参照する必要はありません。BY HIERARCHY 句により、表示する階層が再作成され、フォーマットが再設定されます。また、WHEN 句を使用して階層の一部を選択したり、SHOW 句を使用して選択した階層部分の上下レベル数を指定したりすることもできます。この表示フォーマットにより、階層メンバー間の親子関係が明示されます。

また、各親に対するメジャー値の集計を、すべての子 (完全合計) または選択項目のみ (表示合計) から選択することができます。

詳細は、[階層レポート](#)を参照してください。

階層レポート

キューブに対してリクエストを発行する場合、一部の要件と利用できる機能がシノニムで定義されている階層のタイプに応じて異なります。

親子階層では、各階層のメジャー値に下位すべての合計 (完全合計) を表示するか、表示した下位レベルの合計 (表示合計) を表示するかを指定することができます。レベル階層では、レポートは常に完全合計を表示します。これは、キューブに実際に格納されている値です。

シノニムに親子階層が定義されているときは、BY HIERARCHY 句による階層のソートおよびフォーマットの適用が可能です。また、WHEN 句を使用して、階層の一部のみを選択して表示することもできます。

階層リクエストが処理される場合、はじめに階層が構築され、レポートに表示するノード、レポートから除外するノード、コンテキストに表示するノードのそれぞれが特定されます。

次に、階層にメジャー値が挿入されます。ここでは、リーフノードに WHERE 条件を適用して、レポートに表示するメンバーを限定します。ディメンションプロパティは、リクエストの 1 つ目の選択に使用することはできませんが、ディメンションデータに基づく選択行の選別時に使用することができます。

レポートの階層レベルを選択するために、メジャー値を使用することもできません。階層の行を選択、選別、集計した後、WHERE TOTAL テストを使用して、メジャー値に基づいて表示する行を制限することができます。

親子階層の表示

シノニムに親子階層が定義されている場合は、次の構文を使用して階層レポートを作成することができます。

```
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] ...]]

```

説明

FROLL

メジャーのフルロールアップを指定します。完全集約を指定した場合、表示する値には、キューブ内に存在する値が使用されます。WHEN および SHOW オプションを使用していくつかの下位レベルを出力結果から除外した場合、この値には表示した下位レベルの合計が反映されない場合があります。FROLL を指定しない場合、表示する値には表示合計が使用されます。つまり、表示した下位レベルの値の合計に一致します。

注意：リクエストに WHERE 条件を使用して特定のデータを選択した場合、FROLL はキューブに存在する値を表示しません。この場合、FROLL は、選択したデータの集約を表示します。

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。

ここでは、BY HIERARCHY 句を使用して階層レポートを作成するいくつかの例を紹介します。マスターファイル zopt の親子階層は、0SD_C01/ZTSCQ31CQ1 キューブに基づいて、シノニムの作成時に階層タイプとして [最適化] を選択して作成されています。

階層全体に対するレポート

次のリクエストは表示合計 (SALES_VOLUME) および完全合計 (FROLL.SALES_VOLUME) を生成します。階層レポートは、BY HIERARCHY 句で指定します。階層の一部のみを表示するための WHEN 句は使用していません。なお、完全集約を表示するフィールドには、フィールド名に接頭語の FSUM を付加しています。この完全合計は、階層全体でのレポートの表示合計と一致しています。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
ON TABLE SET PAGE NOPAGE
ON TABLE SET SCREEN PAPER
ON TABLE SET LINES 88
ON TABLE SUBHEAD
"Reporting on a Whole Hierarchy"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

出力結果は次のとおりです。

Reporting on a Whole Hierarchy

<u>Material class Member Caption</u>	<u>Sales Volume</u>	<u>FSUM Sales Volume</u>
Material class	145,209,499.79	145,209,499.79
Products	145,209,499.79	145,209,499.79
Computer systems	85,842,293.86	85,842,293.86
Computer hardware	17,874,768.15	17,874,768.15
Computer	1,948,298.70	1,948,298.70
Desktop computer	1,948,298.70	1,948,298.70
Maxitec-R 375 Personal computer	99,456.50	99,456.50
Maxitec-R 3100 Personal computer	1,848,842.20	1,848,842.20
Maxitec-R 3300 Professional PC	.00	.00
Computer components	15,926,469.45	15,926,469.45
RAM	545,330.61	545,330.61
SIM-Module 8M x 32, PS/2-72 Pin EDO-RAM	545,330.61	545,330.61
Hard disks	13,865,934.28	13,865,934.28
Harddisk 10.80 GB / SCSI-2-Fast	4,176,103.06	4,176,103.06
Harddisk 21.49 GB / SCSI-2-Fast	4,681,064.20	4,681,064.20
Harddisk 42.94 GB / SCSI-2-Fast	5,008,767.02	5,008,767.02
Processors	1,495,035.56	1,495,035.56
Processor 700 MHz	1,072,938.99	1,072,938.99
3Processor 500 MHz	422,096.57	422,096.57
Drives	20,169.00	20,169.00
CD-ROM drives	20,169.00	20,169.00
CD-ROM Drive	20,169.00	20,169.00
Computer accessories	67,967,525.71	67,967,525.71
Keyboards	3,285,675.73	3,285,675.73
Standard Keyboard - EURO Model	614,686.79	614,686.79
Standard Keyboard - EURO-Special Model	619,987.95	619,987.95
Professional keyboard - PROFITEC Model	575,433.19	575,433.19
Professional keyboard - MAXITEC Model	689,355.86	689,355.86
Professional keyboard - NATURAL Model	786,211.94	786,211.94
Printer	.00	.00
Other printers	.00	.00
High Speed Printer	.00	.00

Monitors	64,681,849.98	64,681,849.98
Sunny Sunny 01	2,325,600.85	2,325,600.85
Sunny Xal	2,928,315.00	2,928,315.00
Sunny Tetral3	3,050,854.49	3,050,854.49
Sunny Extreme	3,292,012.09	3,292,012.09
Flatscreen LE 50 P	1,174,441.69	1,174,441.69
Flatscreen MS 1460 P	2,643,840.60	2,643,840.60
Flatscreen LE 64P	2,313,955.17	2,313,955.17
Flatscreen MS 1575P	3,316,991.41	3,316,991.41
Flatscreen MS 1585	3,611,043.69	3,611,043.69
Flatscreen MS 1775P	3,939,392.43	3,939,392.43
Flatscreen MS 1785P	5,042,300.06	5,042,300.06
MAG DX 15F/Fe	2,386,685.30	2,386,685.30
MAG DX 17F	2,688,350.99	2,688,350.99
MAG PA/DX 175	3,059,434.33	3,059,434.33
SEC Multisync XV15	3,129,661.92	3,129,661.92
SEC Multisync XV 17	3,686,004.42	3,686,004.42
Jotachi SN4000	4,908,515.84	4,908,515.84
Jotachi SN4500	2,571,753.78	2,571,753.78
Jotachi SN5000	3,085,110.71	3,085,110.71
Jotachi SN 7000	3,005,405.91	3,005,405.91
TFT Monitor, 17"	871,063.60	871,063.60
PAQ Monitor, 20", Color	1,651,115.70	1,651,115.70
Paints / aux. and operating supplies	.00	.00
Paints	.00	.00
Coating Matt Green RAL 6014/10 Liter	.00	.00
Pumps	59,367,205.93	59,367,205.93
Pumps (complete)	59,367,205.93	59,367,205.93
Pump PRECISION 100	2,964,954.28	2,964,954.28
Pump PRECISION 101	6,407,899.03	6,407,899.03
Pump PRECISION 102	11,213,145.65	11,213,145.65
Pump PRECISION 103	10,406,910.38	10,406,910.38
Pump PRECISION 104	14,328,130.56	14,328,130.56
Pump cast steel IDESNORM 170-230	4,937,452.91	4,937,452.91
Pump standard IDESNORM 100-402	9,108,713.12	9,108,713.12

階層メンバーの選択

次のリクエストは MATERIAL ディメンションの MATERIAL_CLASS 階層の表示合計および完全合計を生成します。ただし WHEN 句により表示に選択されるメンバーは制限されます。なお、選択した階層メンバーから階層のルートに至るまでのすべての上位レベルのメンバーがコンテキストに表示されます。ただし、レポートのリクエストで上位レベルのメンバーを指定していないため、上位レベルにはミッシングデータ記号が表示されます。レポート出力には、選択したメンバーの下位レベルがすべて表示されます。これは、デフォルトの SHOW オプションが BOTTOM に設定されているためです。


```

TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_CAPTION EQ 'Computer hardware';
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Selecting a Hierarchy Member"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END

```

出力結果は次のとおりです。

Selecting a Hierarchy Member		
<u>Material class Member Caption</u>	<u>Sales Volume</u>	<u>FSUM Sales Volume</u>
Material class	.	.
Products	.	.
Computer systems	.	.
Computer hardware	17,874,768.15	17,874,768.15
Computer	1,948,298.70	1,948,298.70
Desktop computer	1,948,298.70	1,948,298.70
Macintec-R 375 Personal computer	99,456.50	99,456.50
Macintec-R 3100 Personal computer	1,848,842.20	1,848,842.20
Macintec-R 3300 Professional PC	.00	.00
Computer components	15,926,469.45	15,926,469.45
RAM	545,330.61	545,330.61
SIM-Module 8M x 32, PS/2-72 Pin EDO-RAM	545,330.61	545,330.61
Hard disks	13,865,934.28	13,865,934.28
Harddisk 10.80 GB / SCSI-2-Fast	4,176,103.06	4,176,103.06
Harddisk 21.49 GB / SCSI-2-Fast	4,681,064.20	4,681,064.20
Harddisk 42.94 GB / SCSI-2-Fast	5,008,767.02	5,008,767.02
Processors	1,495,035.56	1,495,035.56
Processor 700 MHz	1,072,938.99	1,072,938.99
3Processor 500 MHz	422,096.57	422,096.57
Drives	20,169.00	20,169.00
CD-ROM drives	20,169.00	20,169.00
CD ROM Drive	20,169.00	20,169.00

メンバーの選択と親の追加

次のリクエストでは、WHEN 句に追加された SHOW オプション UP 1 TO DOWN 0 により、選択したメンバー (Computer hardware) の親 (Computer systems) が追加されています。この親には、ミッシングデータ記号ではなく、メジャー値が表示されます。ただし、親の完

全合計カラムには、選択した Computer hardware メンバーのみではなく、下位レベルすべての合計が含まれます。これに対して、表示合計には Computer hardware メンバーの合計のみが表示されます。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_CAPTION EQ 'Computer hardware';
SHOW UP 1 TO DOWN 0
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Selecting a Member and Adding a Parent"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

出力結果は次のとおりです。

Selecting a Member and Adding a Parent			
			FSUM
<u>Material class</u>	<u>Member Caption</u>	<u>Sales Volume</u>	<u>Sales Volume</u>
Material class		.	.
Products		.	.
Computer systems		17,874,768.15	85,842,293.86
Computer hardware		17,874,768.15	17,874,768.15

メンバーの選択と子の追加

次のリクエストでは、WHERE 句に追加された SHOW オプションの UP 0 TO DOWN 1 が選択したメンバー (Computer hardware) の子 (Computer と Computer components) を追加します。ここでは、選択したメンバーの子の中から特定の子を除外せず、また SHOW オプションで上位メンバーを追加していないため、完全合計と表示合計は一致しています。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_CAPTION EQ 'Computer hardware';
SHOW UP 0 TO DOWN 1
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Selecting a Member and Adding Children"
```

```

ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END

```

出力結果は次のとおりです。

Selecting a Member and Adding Children			
		FSUM	
<u>Material class</u>	<u>Member Caption</u>	<u>Sales Volume</u>	<u>Sales Volume</u>
Material class		.	.
Products		.	.
Computer systems		.	.
Computer hardware		17,874,768.15	17,874,768.15
Computer		1,948,298.70	1,948,298.70
Computer components		15,926,469.45	15,926,469.45

メンバーの選択と上位レベルの一括表示

次のリクエストでは、WHEN 句に追加した SHOW オプションの TOP TO DOWN 0 によりすべての上位レベルを追加しますが、選択したメンバー (Computer hardware) の下位レベルは追加されません。この親には、ミッシングデータ記号ではなく、メジャー値が表示されます。ただし、上位レベルの完全合計カラムには、選択した Computer hardware メンバーのみではなく、下位レベルすべての合計が含まれます。これに対して、表示合計は Computer hardware メンバーの合計のみが表示されます。

```

TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_CAPTION EQ 'Computer hardware';
SHOW TOP TO DOWN 0
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Selecting a Member and Adding All Ascendants"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END

```

出力結果は次のとおりです。

Selecting a Member and Adding All Ascendants

<u>Material class Member Caption</u>	<u>Sales Volume</u>	<u>FSUM Sales Volume</u>
Material class	17,874,768.15	145,209,499.79
Products	17,874,768.15	145,209,499.79
Computer systems	17,874,768.15	85,842,293.86
Computer hardware	17,874,768.15	17,874,768.15

メンバーの選択と下位レベルの表示

次のリクエストには、SHOW 句がありません。これは、SHOW UP 0 TO BOTTOM と同一です。WHEN 句により、Computer hardware のメンバーが選択されます。選択したメンバーの子が除外されておらず、SHOW オプションに上位メンバーが含まれないため、完全合計と表示合計は同一であり、上位レベルにはミッシングデータ記号が表示されます。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_CAPTION EQ 'Computer hardware';
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Selecting a Member and Showing All Descendants"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

出力結果は次のとおりです。

Selecting a Member and Showing All Descendants

<u>Material class Member Caption</u>	<u>Sales Volume</u>	<u>FSUM Sales Volume</u>
Material class	.	.
Products	.	.
Computer systems	.	.
Computer hardware	17,874,768.15	17,874,768.15
Computer	1,948,298.70	1,948,298.70
Desktop computer	1,948,298.70	1,948,298.70
Macintec-R 375 Personal computer	99,456.50	99,456.50
Macintec-R 3100 Personal computer	1,848,842.20	1,848,842.20
Macintec-R 3300 Professional PC	.00	.00
Computer components	15,926,469.45	15,926,469.45
RAM	545,330.61	545,330.61
SIM-Module 8M x 32, PS/2-72 Pin EDO-RAM	545,330.61	545,330.61
Hard disks	13,865,934.28	13,865,934.28
Harddisk 10.80 GB / SCSI-2-Fast	4,176,103.06	4,176,103.06
Harddisk 21.49 GB / SCSI-2-Fast	4,681,064.20	4,681,064.20
Harddisk 42.94 GB / SCSI-2-Fast	5,008,767.02	5,008,767.02
Processors	1,495,035.56	1,495,035.56
Processor 700 MHz	1,072,938.99	1,072,938.99
3Processor 500 MHz	422,096.57	422,096.57
Drives	20,169.00	20,169.00
CD-ROM drives	20,169.00	20,169.00
CD ROM Drive	20,169.00	20,169.00

メンバーの選択と上位および下位レベルの一括表示

次のリクエストでは、WHEN 句に追加された SHOW オプションの TOP により、選択したメンバー (Computer hardware) のすべての上位レベルと下位レベル (TO BOTTOM がデフォルト) が追加されます。これらの親は SHOW オプションで指定されているため、ミッシングデータ記号ではなく、メジャー値が表示されます。ただし、上位レベルの完全合計カラムには、指定した Computer hardware メンバーのみではなく、下位レベルすべての合計が含まれます。これに対して、表示合計は Computer hardware メンバーの合計のみが表示されます。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_CAPTION EQ 'Computer hardware';
SHOW TOP
ON TABLE SET PAGE NOPAGE
```

```

ON TABLE SUBHEAD
"Selecting a Member and Adding All Ascendants and Descendants"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END

```

出力結果は次のとおりです。

Selecting a Member and Adding All Ascendants and Descendants		
	FSUM	
<u>Material class Member Caption</u>	<u>Sales Volume</u>	<u>Sales Volume</u>
Material class	17,874,768.15	145,209,499.79
Products	17,874,768.15	145,209,499.79
Computer systems	17,874,768.15	85,842,293.86
Computer hardware	17,874,768.15	17,874,768.15
Computer	1,948,298.70	1,948,298.70
Desktop computer	1,948,298.70	1,948,298.70
Maxitec-R 375 Personal computer	99,456.50	99,456.50
Maxitec-R 3100 Personal computer	1,848,842.20	1,848,842.20
Maxitec-R 3300 Professional PC	.00	.00
Computer components	15,926,469.45	15,926,469.45
RAM	545,330.61	545,330.61
SIM-Module 8M x 32, PS/2-72 Pin EDO-RAM	545,330.61	545,330.61
Hard disks	13,865,934.28	13,865,934.28
Harddisk 10.80 GB / SCSI-2-Fast	4,176,103.06	4,176,103.06
Harddisk 21.49 GB / SCSI-2-Fast	4,681,064.20	4,681,064.20
Harddisk 42.94 GB / SCSI-2-Fast	5,008,767.02	5,008,767.02
Processors	1,495,035.56	1,495,035.56
Processor 700 MHz	1,072,938.99	1,072,938.99
3Processor 500 MHz	422,096.57	422,096.57
Drives	20,169.00	20,169.00
CD-ROM drives	20,169.00	20,169.00
CD ROM Drive	20,169.00	20,169.00

階層レベル範囲のメンバーの表示

次のリクエストは MATERIAL_CLASS_LVLNO フィールドに WHEN 句を使用し、階層レベル 0 から 3 のメンバーを表示します。SHOW オプションの TO DOWN 0 により、WHEN 句で選択されていない下位レベルは表示されません。表示合計の計算時に子が除外されないため、完全合計と表示合計は同一になります。

```

TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_LVLNO LE 3;
SHOW TO DOWN 0
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Displaying Members of a Range of Hierarchy Levels"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END

```

出力結果は次のとおりです。

Displaying Members of a Range of Hierarchy Levels		
<u>Material class Member Caption</u>	<u>Sales Volume</u>	<u>FSUM</u> <u>Sales Volume</u>
Material class	145,209,499.79	145,209,499.79
Products	145,209,499.79	145,209,499.79
Computer systems	85,842,293.86	85,842,293.86
Computer hardware	17,874,768.15	17,874,768.15
Computer accessories	67,967,525.71	67,967,525.71
Paints / aux. and operating supplies	.00	.00
Paints	.00	.00
Pumps	59,367,205.93	59,367,205.93
Pumps (complete)	59,367,205.93	59,367,205.93

表示メンバーの選択とメンバー値による選別

レポートに表示する階層メンバーを選択するには、WHEN 句を使用します。選択したメンバーは、それぞれコンテキスト内に表示されます。選択したメンバーの上位レベルには、ミッシングデータ記号が表示されます。リクエストを処理する場合、はじめに階層が構築され、WHEN 句に基づいてレポートに表示するノード、レポートから除外するノード、コンテキストには表示するがミッシングデータ記号を表示するノードのそれぞれが特定されます。

次に、階層にメジャー値が挿入されます。ここでは、WHERE 条件を適用します。この条件では、階層の最下位レベルを選択する必要があります。これを選択しないと、レポートはブランクになります。

WHERE 句を WHEN 条件とともに指定しない場合は、レポートから除外するノードがないため、いずれのノードにもミッシング値は表示されません。

次のリクエストは Computer systems または Pumps の値を含むメンバーを持つ階層の一部を選択します。これらのメンバーの親とすべての下位レベルが表示されます。リクエスト内にディメンションプロパティ MATERIAL_TYPE__MEDIUM_NAME_ が表示され、このプロパティのリーフメンバーのみが値を持つことに注意してください。

```
TABLE FILE ZOPT
WRITE MATERIAL_TYPE__MEDIUM_NAME_ AS 'MEDIUM NAME' SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_CAPTION EQ 'Computer systems' OR 'Pumps';
SHOW UP 1
ON TABLE SET PAGE NOPAGE
ON TABLE SET SCREEN PAPER
ON TABLE SET LINES 88
ON TABLE SUBHEAD
"Selecting Hierarchy Members"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

出力結果は次のとおりです。

Selecting Hierarchy Members

<u>Material class</u>	<u>Member Caption</u>	<u>MEDIUM NAME</u>	<u>Sales Volume</u>
Material class			
Product			145,209,499.79
Computer systems			85,842,295.84
Computer hardware			17,874,748.15
Computer			1,948,298.70
Desktop computer			1,948,298.70
Maximo-K.375 Personal computer	Finished product		99,454.50
Maximo-K.3100 Personal computer	Finished product		1,848,842.20
Maximo-K.3300 Professional PC	Finished product		.00
Computer components			15,924,449.45
RAM			545,330.41
SIM-Module 8M x 32, P&G-72 Pin EDO-RAM	Trading goods		545,330.41
Hard disk			13,845,934.28
Hard disk 110.80 GB / SCSI-2-Fast	Trading goods		4,174,103.04
Hard disk 21.49 GB / SCSI-2-Fast	Trading goods		4,481,044.20
Hard disk 42.94 GB / SCSI-2-Fast	Trading goods		5,008,747.02
Processor			1,495,035.54
Processor 700 MHz	Trading goods		1,072,938.99
Processor 500 MHz	Trading goods		422,096.57
Drive			20,149.00
CD-ROM drive			20,149.00
CD-ROM Drive	Trading goods		20,149.00
Computer accessories			67,947,525.71
Keyboards			3,285,475.73
Standard Keyboard - EURO Model	Trading goods		614,484.79
Standard Keyboard - EURO Special Model	Trading goods		619,987.95
Professional keyboard - PROFITEC Model	Trading goods		575,433.19
Professional keyboard - MAXITEC Model	Trading goods		689,355.84
Professional keyboard - NATURAL Model	Trading goods		784,211.94
Printer			.00
Other printers			.00
High Speed Printer	Trading goods		.00

Monitor		64,681,849.98
Sunny Sunny 01	Trading goods	2,325,600.85
Sunny Hal	Trading goods	2,928,315.00
Sunny IstraB	Trading goods	3,050,834.49
Sunny Extreme	Trading goods	3,292,012.09
Flatmen LE 50 P	Trading goods	1,174,441.69
Flatmen ME 1440 P	Trading goods	2,443,840.40
Flatmen LE 64P	Trading goods	2,313,933.17
Flatmen ME 1575P	Trading goods	3,316,991.41
Flatmen ME 1585	Trading goods	3,411,043.69
Flatmen ME 1775P	Trading goods	3,939,392.43
Flatmen ME 1785P	Trading goods	5,042,300.04
MAG DX 15FW	Trading goods	2,384,683.30
MAG DX 17F	Trading goods	2,688,330.99
MAG PA/DX 175	Trading goods	3,039,434.33
SEC Multisync KV15	Trading goods	3,129,441.92
SEC Multisync KV17	Trading goods	3,684,004.42
Jonchi SN4000	Trading goods	4,908,515.84
Jonchi SN4500	Trading goods	2,571,733.78
Jonchi SN5000	Trading goods	3,085,110.71
Jonchi SN 7000	Trading goods	3,005,405.91
IFT Monitor 17"	Trading goods	871,043.60
PAQ Monitor 20", Color	Trading goods	1,431,115.70
Pumps		59,347,205.93
Pumps (complete)		59,347,205.93
PumpPRECISION 100	Finished product	2,944,934.28
PumpPRECISION 101	Finished product	4,407,899.03
PumpPRECISION 102	Finished product	11,213,145.45
PumpPRECISION 103	Finished product	10,404,910.38
PumpPRECISION 104	Finished product	14,328,130.54
PumpcaststeelDESNCERM 170-230	Finished product	4,937,432.91
Pump standard DESNCERM 100-402	Finished product	9,108,713.12

次のリクエストは Computer systems または Pumps の値を含むメンバーを持つ階層の一部を選択します。次にディメンションプロパティ MATERIAL_TYPE__MEDIUM_NAME_ が Finished product と同一であるものを選別します。この WHERE 選別条件は、WHEN 句で選択した階層部分には影響しませんが、WHERE 条件に一致するデータがない行の合計の表示や選択は行われません。

```
TABLE FILE ZOPT
WRITE MATERIAL_TYPE__MEDIUM_NAME_ AS 'MEDIUM NAME' SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_CAPTION EQ 'Computer systems' OR 'Pumps';
SHOW UP 1
WHERE MATERIAL_TYPE__MEDIUM_NAME_ EQ 'Finished product';
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Selecting Members and Screening Data Values"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
```

```
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

出力結果は次のとおりです。

Selecting Members and Screening Data Values		
<u>Material class Member Caption</u>	<u>MEDIUM NAME</u>	<u>Sales Volume</u>
Material class	.	.
Products		61,315,504.63
Computer systems		1,948,298.70
Computer hardware		1,948,298.70
Computer		1,948,298.70
Desktop computer		1,948,298.70
Macittec-R 375 Personal computer	Finished product	99,456.50
Macittec-R 3100 Personal computer	Finished product	1,848,842.20
Macittec-R 3300 Professional PC	Finished product	.00
Pumps		59,367,205.93
Pumps (complete)		59,367,205.93
Pump PRECISION 100	Finished product	2,964,954.28
Pump PRECISION 101	Finished product	6,407,899.03
Pump PRECISION 102	Finished product	11,213,145.65
Pump PRECISION 103	Finished product	10,406,910.38
Pump PRECISION 104	Finished product	14,328,130.56
Pump cast steel IDESNORM 170-230	Finished product	4,937,452.91
Pump standard IDESNORM 100-402	Finished product	9,108,713.12

メンバー選択とデータ選別の比較

次のリクエストは第3位以下の階層レベルを選択し、最下位階層レベルのキャプションに Flatscreen または Harddisk を含む値を選別します。SHOW オプションの TO DOWN 0 により、選択したメンバーの下位レベルは表示されません。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_LVLNO LE 3;
SHOW TO DOWN 0
WHERE MATERIAL_CLASS_CAPTION CONTAINS 'Flatscreen' OR 'Harddisk';
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Comparing Member Selection With Data Screening"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
```

```
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

出力結果は次のとおりです。

Comparing Member Selection With Data Screening		
		FSUM
<u>Material class Member Caption</u>	<u>Sales Volume</u>	<u>Sales Volume</u>
Material class	35,907,899.33	35,907,899.33
Products	35,907,899.33	35,907,899.33
Computer systems	35,907,899.33	35,907,899.33
Computer hardware	13,865,934.28	13,865,934.28
Computer accessories	22,041,965.05	22,041,965.05

WHEN を伴わない SHOW の使用

次のリクエストでは、WHEN 句を使用していないため、SHOW オプションの DOWN TO 0 がルートノードに適用されます (ルートノードのみを選択する WHEN 句のように機能します)。このため、レポート出力にはルートレベルのみが表示されます。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
SHOW TO DOWN 0
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"SHOW Without WHEN"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

出力結果は次のとおりです。

SHOW Without WHEN		
		FSUM
<u>Material class Member Caption</u>	<u>Sales Volume</u>	<u>Sales Volume</u>
Material class	145,209,499.79	145,209,499.79

SKIP-LINE の使用

次のリクエストでは、BY HIERARCHY 句に BY オプションの SKIP-LINE を追加しています。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_CAPTION EQ 'Computer hardware';
SHOW UP 2 TO DOWN 2 SKIP-LINE
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Using SKIP-LINE"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

出力結果は次のとおりです。

Using SKIP-LINE

<u>Material class</u>	<u>Member Caption</u>	<u>Sales Volume</u>	<u>FSUM Sales Volume</u>
Material class	.	.	.
Products		17,874,768.15	145,209,499.79
Computer systems		17,874,768.15	85,842,293.86
Computer hardware		17,874,768.15	17,874,768.15
Computer		1,948,298.70	1,948,298.70
Desktop computer		1,948,298.70	1,948,298.70
Computer components		15,926,469.45	15,926,469.45
RAM		545,330.61	545,330.61
Hard disks		13,865,934.28	13,865,934.28
Processors		1,495,035.56	1,495,035.56
Drives		20,169.00	20,169.00

メジャーでの条件ソートオプションの使用

次のリクエストでは、BY HIERARCHY コマンドに WHEN 句を使用してメンバーを選択するとともに、別の WHEN 句を使用して UNDER-LINE オプションを制御します。

SUBFOOT および PAGE-BREAK オプションは 2 つの ON 句にあり、同一の階層フィールドを参照します (すべての BY オプションを BY HIERARCHY 句に追加することも可能です)。

それぞれの BY オプションに対応する WHEN 句があります。BY オプションの WHEN 句の式でメジャーが使用されていることに注意してください。この例では、すべての BY オプションは売上高 (SALES_VOLUME) の値がゼロ (0) のときに有効になります。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
```

```

WHEN MATERIAL_CLASS_CAPTION EQ 'Computer hardware';
SHOW TOP UNDER-LINE WHEN SALES_VOLUME EQ 0;
ON MATERIAL_CLASS_CAPTION SUBFOOT
" "
"The Sum is zero"
" "
WHEN SALES_VOLUME EQ 0;
ON MATERIAL_CLASS_CAPTION PAGE-BREAK WHEN SALES_VOLUME EQ 0;
ON TABLE SUBHEAD
"Using BY Options With WHEN on a Measure"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END

```

出力結果は次のとおりです。

Using BY Options With WHEN on a Measure		
<u>Materialclass Member Caption</u>	<u>Sales Volume</u>	<u>Sales Volume</u>
Materialclass	17,874,748.15	145,209,499.79
Product	17,874,748.15	145,209,499.79
Computer systems	17,874,748.15	85,842,298.84
Computer hardware	17,874,748.15	17,874,748.15
Computer	1,948,298.70	1,948,298.70
Desktop computer	1,948,298.70	1,948,298.70
Maximo-E.375 Personal computer	99,454.50	99,454.50
Maximo-E.3100 Personal computer	1,848,842.20	1,848,842.20
Maximo-E.3300 Professional PC	.00	.00
The sum is zero		
PAGE 2		
<u>Materialclass Member Caption</u>	<u>Sales Volume</u>	<u>Sales Volume</u>
Computer components	15,924,449.45	15,924,449.45
RAM	545,330.41	545,330.41
SIM-Module 8M x 32, PS2-72 Pin EDO-RAM	545,330.41	545,330.41
Hard disk	13,845,984.28	13,845,984.28
Harddisk 10.80 GB / SCSI-2-Fast	4,174,103.04	4,174,103.04
Harddisk 21.49 GB / SCSI-2-Fast	4,681,044.20	4,681,044.20
Harddisk 42.94 GB / SCSI-2-Fast	5,008,777.02	5,008,777.02
Processors	1,495,035.54	1,495,035.54
Processor 700 MHz	1,072,998.99	1,072,998.99
3Processor 500 MHz	422,094.57	422,094.57
Drives	20,149.00	20,149.00
CD-ROM drives	20,149.00	20,149.00
CDROM Drive	20,149.00	20,149.00

ディメンションプロパティでの条件ソートオプションの使用

次のリクエストは、WHEN 句付きの BY HIERARCHY コマンドにより、メンバーを選択します。ON 句には、PAGE-BREAK オプションがあり、これは MATERIAL_TYPE__MEDIUM_NAME_ が 'Finished product' と同一のときに有効になります。

```
TABLE FILE ZOPT
WRITE MATERIAL_TYPE__MEDIUM_NAME_ AS 'MEDIUM NAME'
  SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_CAPTION EQ 'Computer hardware';
SHOW UP 2
ON MATERIAL_CLASS_CAPTION PAGE-BREAK
WHEN MATERIAL_TYPE__MEDIUM_NAME_ EQ 'Finished product';
ON TABLE SUBHEAD
"Using PAGE-BREAK With WHEN on a Dimension Property"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

出力結果は次のとおりです。

PAGE 1

Using PAGE-BREAK With WHEN on a Dimension Property

<u>MATERIAL CLASS CAPTION</u>	<u>MEDIUM NAME</u>	<u>Sales Volume</u>
MATERIAL CLASS		
Product		12,824,768.15
Computer system		12,824,768.15
Computer hardware		12,824,768.15
Computer		1,948,298.20
Desktop computer		1,948,298.20
Minivac-R 375 Personal computer	Finished product	99,486.50

PAGE 2

<u>MATERIAL CLASS CAPTION</u>	<u>MEDIUM NAME</u>	<u>Sales Volume</u>
Minivac-R 3100 Personal computer	Finished product	1,248,842.20

PAGE 3

<u>MATERIAL CLASS CAPTION</u>	<u>MEDIUM NAME</u>	<u>Sales Volume</u>
Minivac-R 3300 Professional PC	Finished product	00

PAGE 4

<u>MATERIAL CLASS CAPTION</u>	<u>MEDIUM NAME</u>	<u>Sales Volume</u>
Computer components		15,926,469.45
RAM		545,330.61
SIM-Module SIMx 32, PS/2-72 Pin EDO-RAM	Trading goods	545,330.61
Hard disk		13,865,934.08
Harddisk 10.80 GB + SCSI-2-Fast	Trading goods	4,176,103.06
Harddisk 21.49 GB + SCSI-2-Fast	Trading goods	4,681,064.20
Harddisk 42.94 GB + SCSI-2-Fast	Trading goods	5,008,767.02
Processor		1,495,635.36
Processor 700 MHz	Trading goods	1,072,938.99
Processor 500 MHz	Trading goods	422,696.37
Drive		20,169.00
CDROM drive		20,169.00
CDROM Drive	Trading goods	20,169.00

HIERARCHY フィールドでの条件ソートオプションの使用

次のリクエストは、WHEN 句付きの BY HIERARCHY コマンドにより、メンバーを選択します。ON 句には、PAGE-BREAK オプションがあり、これは MATERIAL_CLASS_CAPTION に 'Harddisk' が含まれる場合に有効になります。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_CAPTION EQ 'Computer hardware';
SHOW UP 2
ON MATERIAL_CLASS_CAPTION PAGE-BREAK
WHEN MATERIAL_CLASS_CAPTION CONTAINS 'Harddisk';
ON TABLE SUBHEAD
"Using PAGE-BREAK With WHEN on a Hierarchy Field"
ON TABLE SET STYLE *
```

```

TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END

```

出力結果は次のとおりです。

PAGE 1		
Using PAGE-BREAK With WHEN on a Hierarchy Field		
<u>Material class Member Caption</u>	<u>Sales Volume</u>	<u>Sales Volume</u> FSGM4
Material class		
Products	17,874,768.15	145,209,499.79
Computer systems	17,874,768.15	25,842,293.86
Computer hardware	17,874,768.15	17,874,768.15
Computers	1,942,292.70	1,942,292.70
Desktop computers	1,942,292.70	1,942,292.70
Mercury-R 375 Personal computers	99,436.50	99,436.50
Mercury-R 3100 Personal computers	1,242,842.20	1,242,842.20
Mercury-R 3300 Professional PC	00	00
Computer components	15,932,469.45	15,932,469.45
RAM	545,330.61	545,330.61
SR4-Memory SR4x 32, PS/2-72 Pin EDO-RAM	545,330.61	545,330.61
Hard disks	13,265,934.25	13,265,934.25
Harddisk 10 30 GB + SCSI-2-Fast	4,176,103.06	4,176,103.06
PAGE 2		
<u>Material class Member Caption</u>	<u>Sales Volume</u>	<u>Sales Volume</u> FSGM4
Harddisk 21 40 GB + SCSI-2-Fast	4,681,064.20	4,681,064.20
PAGE 3		
<u>Material class Member Caption</u>	<u>Sales Volume</u>	<u>Sales Volume</u> FSGM4
Harddisk 42 94 GB + SCSI-2-Fast	5,005,767.02	5,005,767.02
PAGE 4		
<u>Material class Member Caption</u>	<u>Sales Volume</u>	<u>Sales Volume</u> FSGM4
Processors	1,495,033.99	1,495,033.99
Processor 700 MHz	1,072,932.99	1,072,932.99
Processor 500 MHz	422,096.97	422,096.97
Drives	20,169.00	20,169.00
CDROM drives	20,169.00	20,169.00
CDROM Drive	20,169.00	20,169.00

WHERE TOTAL によるメジャーの選別

次のリクエストは、WHEN 句付きの BY HIERARCHY 句により、メンバーを選択します。
WHERE TOTAL 句により、売上高 (SALES_VOLUME) がゼロ (0) ではない行が選択されます。

```

TABLE FILE ZOPT
WRITE SALES_VOLUME FROLL.SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY

```

```

WHEN MATERIAL_CLASS_CAPTION EQ 'Computer hardware';
SHOW TOP
WHERE TOTAL SALES_VOLUME NE 0
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Using WHERE TOTAL to Screen on a Measure"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END

```

出力結果は次のとおりです。

Using WHERE TOTAL to Screen on a Measure		
	FSUM	
<u>Material class Member Caption</u>	<u>Sales Volume</u>	<u>Sales Volume</u>
Material class	17,874,768.15	145,209,499.79
Products	17,874,768.15	145,209,499.79
Computer systems	17,874,768.15	85,842,293.86
Computer hardware	17,874,768.15	17,874,768.15
Computer	1,948,298.70	1,948,298.70
Desktop computer	1,948,298.70	1,948,298.70
Macintec-R 375 Personal computer	99,456.50	99,456.50
Macintec-R 3100 Personal computer	1,848,842.20	1,848,842.20
Computer components	15,926,469.45	15,926,469.45
RAM	545,330.61	545,330.61
SIM-Module 8M x 32,PS/2-72 Pin ED O-RAM	545,330.61	545,330.61
Hard disks	13,865,934.28	13,865,934.28
Harddisk 10.80 GB / SCSI-2-Fast	4,176,103.06	4,176,103.06
Harddisk 21.49 GB / SCSI-2-Fast	4,681,064.20	4,681,064.20
Harddisk 42.94 GB / SCSI-2-Fast	5,008,767.02	5,008,767.02
Processors	1,495,035.56	1,495,035.56
Processor 700 MHz	1,072,938.99	1,072,938.99
3Processor 500 MHz	422,096.57	422,096.57
Drives	20,169.00	20,169.00
CD-ROM drives	20,169.00	20,169.00
CD ROM Drive	20,169.00	20,169.00

2 つの BY HIERARCHY 句の使用

次のリクエストには Material Class 階層のための BY HIERARCHY 句と World/Continents/Countries 階層のための BY HIERARCHY 句があります。World/Continents/Countries 階層は異なるディメンションに属します。World/Continents/Countries 階層から選択されたすべてのメンバーは Material Class 階層に

選択されたメンバーそれぞれに対して繰り返されます。WHERE TOTAL 句により、売上高 (SALES_VOLUME) がゼロ (0) ではない行が除外されます。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
SHOW TO DOWN 2 AS 'Material Class'
BY WORLD___CONTINENTS___COUNTRIES_CAPTION HIERARCHY
WHEN WORLD___CONTINENTS___COUNTRIES_CAPTION OMITTS 'TEST'
AND WORLD___CONTINENTS___COUNTRIES_LVLNO LE 2;
SHOW UP 0 TO DOWN 0 AS 'Region'
WHERE TOTAL SALES_VOLUME NE 0
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Using Two BY HIERARCHY Phrases"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

出力結果は次のとおりです。

Using Two BY HIERARCHY Phrases		
<u>Material Class</u>	<u>Region</u>	<u>Sales Volume</u>
Material class	World / Continent	14,591,918.04
	Europe	14,582,447.78
	Norway	4,743,800.00
	Italy	3,871,450.00
	Great Britain	4,881,550.00
	France	1,043,847.78
	North America	9,250.24
	US	9,250.24
Product	World / Continent	14,591,918.04
	Europe	14,582,447.78
	Norway	4,743,800.00
	Italy	3,871,450.00
	Great Britain	4,881,550.00
	France	1,043,847.78
	North America	9,250.24
	US	9,250.24
Computer systems	World / Continent	1,043,514.04
	Europe	1,043,847.78
	France	1,043,847.78
	North America	1,448.24
	US	1,448.24
Pumps	World / Continent	13,324,402.00
	Europe	13,318,800.00
	Norway	4,743,800.00
	Italy	3,871,450.00
	Great Britain	4,881,550.00
	North America	7,402.00
	US	7,402.00

BY HIERARCHY 句と BY 句の使用

次のリクエストでは、World/Continents/Countries 階層に BY 句、Material Class 階層に BY HIERARCHY 句が指定されています。Material Class 階層で選択されたすべてのメンバーは World/Continents/Countries 階層で選択されたメンバーのそれぞれに対して繰り返されます (BY 句で参照されるときは、階層インデントは設定されません)。ユニークフィールドには BY を使用する必要があります。これは、World/Continents/Countries 階層に対してキャプションが一意ではないことにより、1 つのユニークフィールドには NOPRINT オプション付きの 1 つの BY フィールドがあり、もう 1 つの BY がキャプションフィールドにあるためです。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME
BY WORLD___CONTINENTS___COUNTRIES NOPRINT
BY WORLD___CONTINENTS___COUNTRIES_CAPTION
BY MATERIAL_CLASS_CAPTION HIERARCHY
SHOW TO DOWN 2
WHERE WORLD___CONTINENTS___COUNTRIES_CAPTION OMITTS 'TEST'
AND WORLD___CONTINENTS___COUNTRIES_LVLNO EQ 2;
WHERE TOTAL SALES_VOLUME NE 0
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Using BY HIERARCHY and BY Phrases"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

出力結果は次のとおりです。

Using BY and BY HIERARCHY Phrases		
<u>World / Continents / Countries</u>	<u>Member Caption</u>	<u>Sales Volume</u>
France	Material class	1,063,867.78
	Products	1,063,867.78
	Computer systems	1,063,867.78
Great Britain	Material class	4,881,550.00
	Products	4,881,550.00
	Pumps	4,881,550.00
Italy	Material class	3,871,450.00
	Products	3,871,450.00
	Pumps	3,871,450.00
Norway	Material class	6,765,800.00
	Products	6,765,800.00
	Pumps	6,765,800.00
US	Material class	9,250.26
	Products	9,250.26
	Computer systems	1,648.26
	Pumps	7,602.00

次は、同一のリクエストであり、1 つ目に BY HIERARCHY 句を、2 つ目に BY 句を使用しています。World/Continents/Countries 階層から選択されたすべてのメンバーは Material Class 階層に選択されたメンバーそれぞれに対して繰り返されます。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
SHOW TO DOWN 2
BY WORLD___CONTINENTS___COUNTRIES NOPRINT
BY WORLD___CONTINENTS___COUNTRIES_CAPTION
WHERE WORLD___CONTINENTS___COUNTRIES_CAPTION OMITTS 'TEST'
AND WORLD___CONTINENTS___COUNTRIES_LVLNO EQ 2;
WHERE TOTAL SALES_VOLUME NE 0
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Using BY HIERARCHY and BY Phrases"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

出力結果は次のとおりです。

Using BY HIERARCHY and BY Phrases			
Material class	Member Caption	World / Continents / Countries	Member Caption Sales Volume
Material class	France		1,063,867.78
	Great Britain		4,881,550.00
	Italy		3,871,450.00
	Norway		6,765,800.00
	US		9,250.26
Products	France		1,063,867.78
	Great Britain		4,881,550.00
	Italy		3,871,450.00
	Norway		6,765,800.00
	US		9,250.26
Computer systems	France		1,063,867.78
	US		1,648.26
Pumps	Great Britain		4,881,550.00
	Italy		3,871,450.00
	Norway		6,765,800.00
	US		7,602.00

BY HIERARCHY を伴わない ON HIERARCHY の使用

次のリクエストでは、Material Class 階層に対して BY 句と ON HIERARCHY 句が使用されています。ON HIERARCHY 句では、すべての階層オプションと BY オプションがサポートされます。このリクエストにも、WHERE 句が使用されており、階層フィールド (MATERIAL_CLASS_CAPTION) の値に基づいて行を選択します。WHERE テストはリーフノードの値 (Harddisk) に基づいて行を選択します。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME
BY MATERIAL_CLASS_CAPTION
ON MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_CAPTION CONTAINS 'Computer';
WHERE MATERIAL_CLASS_CAPTION CONTAINS 'Harddisk'
ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Using BY and ON HIERARCHY"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
```

```
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

出力結果は次のとおりです。

Using BY and ON HIERARCHY	
<u>Material class Member Caption</u>	<u>Sales Volume</u>
Material class	.
Products	.
Computer systems	13,865,934.28
Computer hardware	13,865,934.28
Computer components	13,865,934.28
Hard disks	13,865,934.28
Harddisk 10.80 GB / SCSI-2-Fast	4,176,103.06
Harddisk 21.49 GB / SCSI-2-Fast	4,681,064.20
Harddisk 42.94 GB / SCSI-2-Fast	5,008,767.02

複数の表示コマンドを伴う階層レポート

複数の表示コマンドを伴うリクエストでは、コマンドごとに 1 つ前のコマンドで使用したすべてのソート句を同一順序で繰り返す必要があります。その後、別のソート句を追加することができます。階層レポートリクエストでは、次の規則に従う必要があります。

- 表示コマンドには、すべて集計コマンドを使用します。階層レポートリクエストでは、PRINT はサポートされません。
- BY および BY HIERARCHY ソート句の両方を使用することができます。非階層リクエストと同様に、表示コマンドが後に続く場合、その表示コマンドですべてのソート句を同一順序で繰り返す必要があります。
- リクエストで階層を参照する場合、各階層に対して指定できる WHEN および SHOW 句は、それぞれ 1 つに限定されます。これらの句は、リクエストの任意のレベルで指定することができますが、リクエスト内のすべての表示コマンドに適用されます。

階層レポートでの複数表示コマンドの使用

次のリクエストでは、2 つの WRITE コマンドを使用します。1 つ目の WRITE コマンドには、Material Class 階層に対する BY HIERARCHY ソート句と WORLD_CONTINENTS_

COUNTRIES 階層に対する BY HIERARCHY 句があります。2 つ目の WRITE コマンドはこれらの句を繰り返し、SALESORG ディメンションに対する BY 句が追加されています。

```
TABLE FILE ZOPT
WRITE SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
WHEN MATERIAL_CLASS_CAPTION EQ 'Computer hardware'; AS 'Material Class'
SHOW TO DOWN 2

BY WORLD___CONTINENTS___COUNTRIES_CAPTION HIERARCHY
WHEN WORLD___CONTINENTS___COUNTRIES_CAPTION OMITS 'TEST'
AND WORLD___CONTINENTS___COUNTRIES_LVLNO EQ 2; AS 'Region'

WRITE SALES_VOLUME SALES_ORDER_ITEM
BY MATERIAL_CLASS_CAPTION HIERARCHY
BY WORLD___CONTINENTS___COUNTRIES_CAPTION HIERARCHY
BY SALES_ORGANIZATION_LEVEL_01 AS 'Sales Organization'

ON TABLE SET PAGE NOPAGE
ON TABLE SUBHEAD
"Using Multiple Display Commands"
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=8,$
TYPE=REPORT, GRID=OFF, WRAP=OFF, $
TYPE=TABHEADING, SIZE=10, COLOR=RED, JUSTIFY=CENTER, $
END
```

Material Class 階層の BY HIERARCHY 句は、WHEN 句によりキャプションが Computer hardware であるメンバーを選択します。SHOW オプション (TO DOWN 2) により Computer hardware レベルの下での 2 つのレベルが表示されます。選択したメンバーの上位レベルはコンテキストとして表示されますが、これらには、値ではなくミッシングデータ記号が表示されます。

WORLD_CONTINENTS_COUNTRIES 階層で選択されたメンバーはレベル番号 2 を持ち、キャプションに「TEST」という文字列を持ちません。SHOW オプションはレポート出力に 2 つの下位レベルを表示します。上位レベルはコンテキストとしてミッシングデータ記号とともに表示されます。World/Continents/Countries 階層から選択されたすべてのメンバーは Material Class 階層に選択されたメンバーそれぞれに対して繰り返されます。

出力結果は次のとおりです。

Using Multiple Display Commands

<u>Material Class</u>	<u>Region</u>	<u>Sales Volume</u>	<u>Sales Organization</u>	<u>Sales Volume</u>	<u>Sales Order Item</u>
Material class	.	.	.	.	.
Products	.	.	.	.	.
Computer systems	.	.	.	.	.
Computer hardware	World / Continents	.	.	.	.
	Europe	.	.	.	.
Computer hardware	France	1,061,431.86	.	1,061,431.86	1.00
	IDES France SA	7,925.02	.	7,925.02	.00
	Adecom SA	1,053,506.84	.	1,053,506.84	1.00
Computer	World / Continents	.	.	.	.
	Europe	.	.	.	.
Computer	France	1,061,431.86	.	1,061,431.86	1.00
	IDES France SA	7,925.02	.	7,925.02	.00
	Adecom SA	1,053,506.84	.	1,053,506.84	1.00
Desktop computer	World / Continents	.	.	.	.
	Europe	.	.	.	.
Desktop computer	France	1,061,431.86	France, Paris	1,053,506.84	1.00
			Germany Frankfurt	7,925.02	.00
	IDES France SA	7,925.02	Germany Frankfurt	7,925.02	.00
	Adecom SA	1,053,506.84	France, Paris	1,053,506.84	1.00

カラムレポート

キューブに対して標準の TABLE または FML レポートを実行する場合、スライスレポートと呼ばれるレポートを実行します。この 2 次元レポートは、より大きいマルチディメンションのキューブのスライスに類似しています。スライスレポートでは、キューブのマルチディメンション構造は平面化され、特定の角度から見た情報が表示されます。

有効なスライスレポートを作成するには、キューブデータの構造、ロジック、および SAP BW アダプタでキューブをスライスする際の前提を理解する必要があります。

有効または無効なスライス - キューブのスライスロジック

SAP BW アダプタには、各ロールアップから抽出する行およびセルを特定してスライスレポートを提供するロジックが組み込まれています。TABLE では、ディメンションごとに最下位レベルの BY 句を使用して集計用の参照を見つけるためのロールアップを判別します。次に、そのロールアップから結果をロールアップし、また計算結果の整合性を確保するために

必要に応じてそのデータの集計を実行します。

TABLE リクエストに BY フィールドを指定しない場合、無効なデータまたはメッセージが返されます。これは、データ検索を実行するロールアップのレベルを特定する BY フィールドが指定されておらず、TABLE で結果を返す場所を認識していないためです。

スライスレポートでの単一のディメンションの表示

```
TABLE FILE MYCUBE  
SUM STORE_COST STORE_SALES  
BY COUNTRY_LEVEL_01 ON TABLE COLUMN_TOTAL  
END
```

このレポートでは、ソースの単一のディメンションに基づいてソートされた集計データが表示されます。STORE_COST や STORE_SALES などの標準のメジャーが、SUM によって参照されます。レポート内に列合計が要求されています。

スライスレポートでの複数のディメンションの表示

```
TABLE FILE MYCUBE  
SUM STORE_COST STORE_SALES  
PRINT PROFIT  
BY COUNTRY_LEVEL_01 BY STATE_PROVINCE_LEVEL BY CITY_LEVEL_01  
END
```

このレポートでは、キューブの両方のディメンション階層に基づいてソートされる点を除いて、上記のレポートと同じデータが表示されます。ソートフィールドの順序は、ディメンション階層の論理順序に従います。

変数を使用したレポート

BW クエリでは、変数を使用して実行時に値を提供します。変数の Entry_Type には、必須またはオプションがあります。必須の変数の値は、TABLE または SQL リクエストで指定しなければなりません。リクエスト内の WHERE または IF ステートメントでは、実行時の値を提供します。たとえば、次のように変数の COMPANY_CODE が必須の場合、実行時に会社コード 430 が提供されます。

```
WHERE COMPANY_CODE_N EQ '430'
```

BW に渡される変数のフォーマットは、124 バイトの alpha です。マスターファイルのシノニムでは、変数を variable_C と variable_N という 2 つの部分に分割します。

- キャプション (_C) が先頭から 60 バイトで、BW 用の内部参照です。
- 名前 (_N) が残りの 64 バイトで、その中で実際のキー値がクエリに送信されます。

SAP BW では、SAP VARIABLES 句内のリテラルにメンバー名の複合でない部分のみが含まれていると想定します。SAP BW アダプタではメンバー名とキャプション (MEMBER タイプの変数用の) によってユーザと通信しますが、それは、自動的に複合を検出して、提供されたメンバー名の複合でない部分に基づくリテラルを生成します。複合変数の使用時に分かりやすいレポートを得るためには、ユーザは SAP OSS ノート 605208 について理解し、その指示に従う必要があります。このノートでは、複合特性に対して個別の変数を作成するか、単一値のハードフィルタを作成することを実際上要求しています。詳細は、OSS ノート 605208 を参照してください。

上の例では、名前 (_N) で変数の値を提供します。キャプションまたは名前を選別に使用できますが、次のように適切な値をそれぞれ指定する必要があります。

```
company_code_n :WHERE COMPANY_CODE_N EQ 430
company_code_c :WHERE COMPANY_CODE_N EQ 'IDES USA'
```

ヒント：この変数に関連するディメンションをクエリで 사용할 ことができる場合、そのフィールドに対して選択を作成するのではなく名前を使用します。つまり、COMPANY_CODE_N は変数に代入されますが、WHERE COMPANY_CODE_LEVEL_01 EQ 430 は変数に代入されません。その他の非変数を選択することは可能ですが、効率的ではありません。非変数の条件は、BW のアンサーセットが抽出された後で適用されるためです。

変数はリクエスト内で表示することはできません。変数は選択条件を適用する目的でのみ使用する必要があります。関連付けられたディメンションまたはメジャーを表示することができます。

変数を BY フィールドとして表示することはできません。変数の選択タイプとして、単一、間隔、複合があります。

- 単一タイプの変数には、選択条件に単一の値のみを指定できます。
- 間隔タイプの変数には、単一の値またはある範囲の値を指定できます。
- 複合タイプの変数には、制限はありません。

値の範囲を指定するには、2 つの WHERE ステートメントを使用するか、AND を使用して 2 つを 1 つのステートメントにまとめます。たとえば、次のレポートでは 2 つの年を指定する必要があります。

```
WHERE FISCAL_YEAR_N EQ '2001'  
      OR FISCAL_YEAR_N EQ '2002'  
WHERE FISCAL_YEAR_N GE '2001'  
WHERE FISCAL_YEAR_N LE '2002'  
WHERE FISCAL_YEAR_N GE '2001'  
      AND FISCAL_YEAR_N LE '2002'  
WHERE FISCAL_YEAR_N IN ('2001', '2002')  
WHERE FISCAL_YEAR_N FROM '2001' TO '2002'
```

上の 1 番目の例のように OR を使用する選択では、同じフィールドまたは関連する階層に対してのみテストを行うことができます。同じ WHERE ステートメント (選別の述語) で、変数の選択と非変数の選択を組み合わせることはできません。

完全集計および部分集計

キューブリクエストは、1 つ以上のメジャーと、場合によっては複数の階層を参照します。変数を参照することもできます。

キューブリクエストでは、アダプタが完全集計または部分集計データをキューブから読み取り、TABLE リクエストに特定の制限を適用して、レポートの整合性を保持します。

キューブリクエストを処理する場合、「完全集計」と「部分集計」という 2 つのモードがあります。

- 完全集計モードでは、アダプタはレポートに表示されるとおりにキューブから集計値を抽出します。
- 部分集計モードでは、アダプタが部分的に集計データを抽出した後、WebFOCUS がさらに集計を行います。

完全集計モードでは、キューブに格納されている集計以外を必要としないため、より効率的です。

完全集計モードのサポート

下表は、完全集計モードのサポートについての説明です。

コマンド	ディメンションプロパティ	メジャー	変数
WRITE/SUM/ADD	サポートされています。FST に変換されます。	サポートされています。メジャー集計演算子に一致する演算に変換されます。	サポートされていません。
COMPUTE	サポートされています。	サポートされています。	サポートされていません。
PRINT	サポートされています。	互換性のためにサポートされています。BY HIERARCHY との使用はサポートされていません。	サポートされていません。
WHERE TOTAL	サポートされています。	サポートされています。	サポートされていません。
WHERE	サポートされています。	サポートされていません。	必須変数では省略できません。変数の選択タイプに応じた制限を指定する必要があります。単一変数を表すフィールドグループに制限されます。
WHEN	サポートされています。	サポートされて	サポートされてい

コマンド	ディメンションプロパティ	メジャー	変数
	ます。	いません。	ません。
DEFINE	サポートされています。	サポートされていません。	サポートされていません。
BY	サポートされています。親子階層のユニークフィールドには BY を使用する必要があります。ただし、WHERE でのみ参照する場合は例外です。BY HIERARCHY は、BY を内部的に追加します。	サポートされていません。	サポートされていません。
BY HIERARCHY	サポートされています。	サポートされていません。	サポートされていません。
ON HIERARCHY	サポートされています。対応する BY 句または BY HIERARCHY 句を使用する必要があります。	サポートされていません。	サポートされていません。

部分集計モードのサポート

部分集計モードでは、メジャーを使用する際に、上記以外の制限事項が適用されます。なお、複数の表示コマンドを使用するリクエストでは、部分集計の制限事項がすべての表示コマンドに適用されます。ただし、最下位の BY 句に関連する表示コマンドは対象外です。

- メジャー集計演算子に一致する加算集計演算子 (SUM.、CNT.、MAX.、MIN. などの部分合計からさらに集計が可能な集計演算子)

- 加算以外の集計演算子 (AVE. など) はサポートされていません。

次の要因により、部分集計モードが有効になります。

- 複数の表示コマンド。
- 親子階層 - 階層に WHERE を伴う完全合計または表示合計。
- レベル階層 - リクエストで参照する階層レベルに、BY 句を伴わないレベルが存在する場合。
- すべての階層 - 複数のメンバーを選択する WHERE 句のみで階層が参照される場合。

時間依存階層サポート

階層のコンテンツは、時間の経過とともに変更される場合があります。変更が発生すると、異なる期間の複数の階層バージョンが累積されます。キューブから作成されるレポート結果は、レポート実行時にどの階層バージョンを選択したかによって異なります。

複数の時間依存階層で構成されたキューブを作成することができます。ここで、各階層には、期間ごとに異なるバージョンが含まれます。

このキューブに対してレポートを作成する場合、時間依存階層のバージョンを任意に選択することはできません。それは、特定のバージョンの組み合わせのみがビジネス上で意味をなすためです。時間依存階層のバージョンを選択するには、「キー日付」と呼ばれる日付を指定します。キー日付を指定すると、その有効な日付のキューブのコンテンツが表示されます。BW システムは、そのキー日付で有効な時間依存階層のバージョンの適切な組み合わせを自動的に選択します。

時間依存階層には、同一階層の同時バージョンが複数存在する場合もあります。これらのバージョンの数は、時間の経過とともに変化し、キューブのメタデータに反映されます。BW システムは、キー日付に基づいて、これらの変動する数の同時バージョンをそれぞれに固有の名前が付けられた階層として表示します。これにより、階層のコンテンツが時間とともに変化するだけでなく、その階層の同時バージョン (同時に存在する) も変化することができます。

時間依存階層を含むキューブに対してシノニムを作成する際に、キー日付を指定し、そのキー日付に関連する期間に存在していたすべてのバージョンの階層 (時間依存および非時間依存) を含むマスターファイルを生成することができます。シノニムでは、同一階層のさまざまな同時バージョンがそれぞれ異なるフィールド名を持つため、バージョンの違いを区別することができます。複数の同時バージョンが存在する、または存在していた時間依存階層がある場合、シノニムの作成時にキー日付を指定することはほぼ理にかなっています。キー日付は、階層の同時バージョンの数が最大になったときの期間に対して指定する必要があります。

シノニムの作成時にキー日付を指定しない場合は、現在の日付がキー日付として使用されます。キューブ内のすべての時間依存階層に、時間変化するコンテンツのみが存在し、数が変動する同時バージョンが存在しない場合、シノニムの作成時にキー日付を指定することは、メンバーのサンプリングを同時にリクエストする場合に限って意味をなします。それ以外の場合、指定したキー日付に関係なく、結果のシノニムはすべて同一になります。

時間依存階層に対してレポートを作成するときは、ENGINE BWBAPI SET KEY_DATE コマンドを使用して、次に実行する TABLE リクエストのキー日付を指定します。キー日付を指定しない場合は、現在の日付がキー日付として使用されます。キューブ内のすべての時間依存階層のコンテンツは、指定したキー日付の時点でのコンテンツとしてレポートに表示されます。階層の特定の同時バージョンに対してレポートを実行するには、リクエスト内でその階層バージョンに関連付けられたフィールド名を参照します。この参照は、指定するキー日付に合わせる必要があります。正しいキー日付 (指定した階層の同時バージョンに関連付けられた期間内のキー日付) を指定しなかった場合は、レポートが失敗し、指定した階層が存在しないことを告げるエラーメッセージが表示されます。

マスターファイル内の時間依存階層

あるマスターファイルの次の部分は、\$0SD_C01 キューブに対してキー日付の 20081031 を使用して生成されました。時間依存階層は、0MATERIAL ディメンションに対して定義されます。時間依存階層の同時バージョンは太字で示されています。

```
DIMENSION=[0MATERIAL], CAPTION='Material', $
  HIERARCHY=[0MATERIAL], CAPTION='Material', HRY_DIMENSION=[0MATERIAL],
  HRY_STRUCTURE=STANDARD, $
    HIERARCHY=[0MATERIAL                                001], CAPTION='Material
class', HRY_DIMENSION=[0MATERIAL], HRY_STRUCTURE=RECURSIVE, $
      HIERARCHY=[0MATERIAL                                MAT_CLASS_TD
V1], CAPTION='Material Class Time Dep', HRY_DIMENSION=[0MATERIAL],
HRY_STRUCTURE=RECURSIVE, $
        HIERARCHY=[0MATERIAL                                MAT_CLASS_TD
V2], CAPTION='Material Class Time Dep', HRY_DIMENSION=[0MATERIAL],
HRY_STRUCTURE=RECURSIVE, $
          HIERARCHY=[0MATERIAL                                PRDHA], CAPTION='Product
Hierarchy for material MARA', HRY_DIMENSION=[0MATERIAL],
  HRY_STRUCTURE=RECURSIVE, $
    HIERARCHY=[0MATERIAL                                RCV], CAPTION='Retail category
view', HRY_DIMENSION=[0MATERIAL], HRY_STRUCTURE=RECURSIVE, $
```

マスターファイルの以下の部分は、2 つの同時バージョンの時間依存階層にリストされるフィールドを示しています。最初のバージョンにリストされるフィールドには、それらの名前に文字列 TIME_DEP が付いており、2 番目のバージョンにリストされるフィールドには、それらの名前に文字列 TIME_DEP1 が付いています。

バージョン 1 の時間依存階層のフィールド

```

FIELDNAME=MATERIAL_CLASS_TIME_DEP, USAGE=A143, ACTUAL=A143,
MISSING=ON,
TITLE='Material Class Time Dep Member Unique Name',
WITHIN='*[@MATERIAL          MAT_CLASS_TD          V1]',
PROPERTY=UID, $

```

```

FIELDNAME=MATERIAL_CLASS_TIME_DEP_NAME, USAGE=A64, ACTUAL=A64,
MISSING=ON,
TITLE='Material Class Time Dep Member Name',
REFERENCE=MATERIAL_CLASS_TIME_DEP, PROPERTY=NAME, $

```

```

FIELDNAME=MATERIAL_CLASS_TIME_DEP_LVLNO, USAGE=I2L, ACTUAL=I4,
MISSING=ON,
TITLE='Material Class Time Dep Member Level Number',
REFERENCE=MATERIAL_CLASS_TIME_DEP, PROPERTY=LEVEL_NUMBER, $

```

```

FIELDNAME=MATERIAL_CLASS_TIME_DEP_PARENT, USAGE=A143, ACTUAL=A143,
MISSING=ON,
TITLE='Material Class Time Dep Parent Unique Name',
REFERENCE=MATERIAL_CLASS_TIME_DEP, PROPERTY=PARENT_OF, $

```

```

FIELDNAME=MATERIAL_CLASS_TIME_DEP_PARENT_LVLNO, USAGE=I2L, ACTUAL=I4,
MISSING=ON,
TITLE='Material Class Time Dep Parent Level',
REFERENCE=MATERIAL_CLASS_TIME_DEP, PROPERTY=PARENT_LEVEL_NUMBER, $

```

```

FIELDNAME=MATERIAL_CLASS_TIME_DEP_CHILDREN_CARD, USAGE=I9, ACTUAL=I4,
MISSING=ON,
TITLE='Material Class Time Dep Member Children Cardinality',
REFERENCE=MATERIAL_CLASS_TIME_DEP, PROPERTY=CHILDREN_CARDINALITY, $
FIELDNAME=MATERIAL_CLASS_TIME_DEP_CAPTION, USAGE=A60, ACTUAL=A60,

```

```

MISSING=ON,
TITLE='Material Class Time Dep Member Caption',
REFERENCE=MATERIAL_CLASS_TIME_DEP, PROPERTY=CAPTION, $

```

バージョン 2 の時間依存階層のフィールド

```

FIELDNAME=MATERIAL_CLASS_TIME_DEP1, USAGE=A143, ACTUAL=A143,
MISSING=ON,
TITLE='Material Class Time Dep Member Unique Name',
WITHIN='*[0MATERIAL          MAT_CLASS_TD          V2] ',
PROPERTY=UID, $
FIELDNAME=MATERIAL_CLASS_TIME_DEP1_NAME, USAGE=A64, ACTUAL=A64,
MISSING=ON,
TITLE='Material Class Time Dep Member Name',
REFERENCE=MATERIAL_CLASS_TIME_DEP1, PROPERTY=NAME, $
FIELDNAME=MATERIAL_CLASS_TIME_DEP1_LVLNO, USAGE=I2L, ACTUAL=I4,
MISSING=ON,
TITLE='Material Class Time Dep Member Level Number',
REFERENCE=MATERIAL_CLASS_TIME_DEP1, PROPERTY=LEVEL_NUMBER, $
FIELDNAME=MATERIAL_CLASS_TIME_DEP1_PARENT, USAGE=A143, ACTUAL=A143,
MISSING=ON,
TITLE='Material Class Time Dep Parent Unique Name',
REFERENCE=MATERIAL_CLASS_TIME_DEP1, PROPERTY=PARENT_OF, $
FIELDNAME=MATERIAL_CLASS_TIME_DEP1_PARENT_LVLNO, USAGE=I2L, ACTUAL=I4,
MISSING=ON,
TITLE='Material Class Time Dep Parent Level',
REFERENCE=MATERIAL_CLASS_TIME_DEP1, PROPERTY=PARENT_LEVEL_NUMBER, $
FIELDNAME=MATERIAL_CLASS_TIME_DEP1_CHILDREN_CARD, USAGE=I9, ACTUAL=I4,
MISSING=ON,
TITLE='Material Class Time Dep Member Children Cardinality',
REFERENCE=MATERIAL_CLASS_TIME_DEP1, PROPERTY=CHILDREN_CARDINALITY, $
FIELDNAME=MATERIAL_CLASS_TIME_DEP1_CAPTION, USAGE=A60, ACTUAL=A60,
MISSING=ON,
TITLE='Material Class Time Dep Member Caption',
REFERENCE=MATERIAL_CLASS_TIME_DEP1, PROPERTY=CAPTION, $

```

時間依存階層に対するレポートの作成

次のリクエストでは、キー日付の 20081031 を使用して作成された TIMEDEP という名前のシノニムに対してレポートを作成します。このリクエストには、最初のバージョンの時間依存 0MATERIAL 階層の売上高の正味価値がリストされています。このリクエストでは、以下のように、階層の 3 つのレベルと空白以外のセルのみが表示されます。

```

ENGINE BWBAPI SET KEY_DATE 20081031
TABLE FILE TIMEDEP
SUM NET_VALUE_OF_SALES_VOLUME
BY MATERIAL_CLASS_TIME_DEP_CAPTION HIERARCHY
SHOW TO DOWN 3
ON TABLE SET PAGE NOPAGE

```

```
ON TABLE SET STYLE *
GRID=OFF, $
END
```

出力結果は次のとおりです。

<u>Material Class Time Dep Member Caption</u>	<u>Net Value of Sales Volume</u>
ACCESSORIES	31,926,127.92
Parts	31,926,127.92
Deluxe Headlight	1,770,180.46
Deluxe Taillight	566,089.48
SunFun / 1200 cm3	29,589,857.98

次のリクエストでは、2 番目のバージョンの時間依存階層が使用されます。

```
ENGINE BWBAPI SET KEY_DATE 20081031
TABLE FILE TIMEDEP
SUM NET_VALUE_OF_SALES_VOLUME
BY MATERIAL_CLASS_TIME_DEP1_CAPTION HIERARCHY
SHOW TO DOWN 3
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
GRID=OFF, $
END
```

出力結果は次のとおりです。

<u>Material Class Time Dep Member Caption</u>	<u>Net Value of Sales Volume</u>
ACCESSORIES	1,770,180.46
Parts	1,770,180.46
Deluxe Headlight	1,770,180.46

以下は、時間依存でない階層のバージョンに対する同じリクエストです。このバージョンの階層を使用した場合、結果はいずれのキー日付でも同じになります。

```
TABLE FILE TIMEDEP
SUM NET_VALUE_OF_SALES_VOLUME
BY MATERIAL_CLASS_CAPTION HIERARCHY
SHOW TO DOWN 3
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
```

```
GRID=OFF, $
END
```

出力結果は次のとおりです。

<u>Material class Member Caption</u>	<u>Net Value of Sales Volume</u>
Material class	145,209,499.79
Products	145,209,499.79
Computer systems	85,842,293.86
Computer hardware	17,874,768.15
Computer accessories	67,967,525.71
Paints / aux. and operating supplies	.00
Paints	.00
Pumps	59,367,205.93
Pumps (complete)	59,367,205.93

逆符号でのメジャーの表示

BEx クエリでは、メジャーのオプションの 1 つを使用して、メジャー値のプラス符号 (+) とマイナス符号 (-) を反転させて表示することができます。BEx クエリでメジャーを逆符号で表示するよう指定されていることが分かっている場合、データ管理コンソールを使用して、WebFOCUS レポートでそのメジャーを逆符号で表示するシノニムを作成することができます。これを行うには、そのメジャーのフィールド宣言に REVERSE_SIGN=ON 属性を追加します。

この属性を適用した場合、BEx クエリで逆符号が指定されているかどうかに関係なく、正負の符号は常に反転されます。そのため、この属性を適用する前に、データおよびクエリ定義を確認する必要があります。この属性は、格納されているメジャー値には反映されず、表示のみに影響します。

逆符号でのメジャーの表示

メジャーを逆符号 (正負の符号を反転) で表示するには、次の属性を FIELD 宣言に追加します。

```
REVERSE_SIGN=ON
```

逆符号でのメジャーの表示

次のリストは ZMINUS シノニムの一部ですが、ここでは CM_1 メジャーに REVERSE_SIGN=ON 属性が追加されています。この属性が追加されているのは、BEx クエリで CM_1 メジャーに対して逆符号が指定されているためです。

```
FILENAME=WAREHOUS, SUFFIX=BWBAPI,$
SEGMENT=ZMINUS, SEGTYPE=S0, $
$ MEASURES FOR CUBE 0SD_C01/ZMINUS
FIELDNAME=CM_1, ALIAS=4KN324NNXDL1055YDE017L0L8, USAGE=D18.2,
ACTUAL=D8, MISSING=ON, TITLE='CM 1',
REFERENCE=SUM, PROPERTY=MEASURE, REVERSE_SIGN=ON,
$
FIELDNAME=COST_OF_SALES____, ALIAS=4KN324VCGC6QIRPEJ8QDHMB0,
USAGE=D18.5, ACTUAL=D8, MISSING=ON, TITLE='Cost of Sales (%)',
REFERENCE=UNKNOWN, PROPERTY=MEASURE, $
DIMENSION=[0MATERIAL], CAPTION='Material', $
HIERARCHY=[0MATERIAL], CAPTION='Material', HRY_DIMENSION=[0MATERIAL],
HRY_STRUCTURE=STANDARD, $
HIERARCHY=[0MATERIAL 001], CAPTION='Material class',
HRY_DIMENSION=[0MATERIAL], HRY_STRUCTURE=RECURSIVE, $
```

次のリクエストは、ZMINUS シノニムに対して実行され、CM_1 および Cost_of_Sales メジャーを表示します。

```
TABLE FILE ZMINUS
PRINT CM_1 COST_OF_SALES
BY MATERIAL_NAME
END
```

以下は、出力の一部を示しています。CM_1 メジャーは、逆符号で表示されています。

PAGE 1		
Material Member Name	CM 1	Cost of Sales (%)
100-400	.00	.
1400-100	-486,395.92	72.52281
1400-200	-177,820.37	68.58794
1400-300	-15,185,915.01	48.67865
1400-310	491,315.31	103.81913
1400-400	-3,389,899.27	15.21928
1400-750	1,159,102.91	264.66759
AM2-GT	-76,574.25	3.77397
AZ2-600	2,550.45	104.82289
AZ2-730	-3,376.03	33.86726
BP-100	.00	.
C-100	.00	.
C-1100	.00	.
C-202	.00	.

シノニムから REVERSE_SIGN=ON 属性を削除すると、同一のリクエストで次の出力が生成されます。

PAGE 1		
Material Member Name	CM 1	Cost of Sales (%)
100-400	.00	.
1400-100	486,395.92	72.52281
1400-200	177,820.37	68.58794
1400-300	15,185,915.01	48.67865
1400-310	-491,315.31	103.81913
1400-400	3,389,899.27	15.21928
1400-750	-1,159,102.91	264.66759
AM2-GT	76,574.25	3.77397
AZ2-600	-2,550.45	104.82289
AZ2-730	3,376.03	33.86726
BP-100	.00	.
C-100	.00	.
C-1100	.00	.
C-202	.00	.

レポートの規則

レポートの規則および機能は、シノニムで定義した階層タイプにより異なります。これらの項目は、次のセクションで要約しています。

表示コマンドおよび演算接頭語のサポート

リクエストで BY HIERARCHY を使用しない場合は、集計データに対して表示コマンド PRINT を使用することもできます。これは、内部で最適なコマンドに変換されます。

注意：集計値がキューブ内に存在し、それ以外の集計が必要でないことを強調するために、この例では WRITE コマンドを使用します。これは、SUM コマンドと同義です。

また、メジャー演算は内部で自動的に提供されるため、リクエストに演算接頭語を追加する必要はありません。ただし、演算接頭語を追加する場合は、メジャー演算子と一致させる必要があります。

たとえば、「SALES」と「ASALES」という2つのメジャーが存在し、SALES が単純集計、ASALES が平均である場合を想定します。SALES と ASALES を参照するリクエストでは、次の表示コマンドを使用することができます。

```
WRITE SALES ASALES
```

メジャー ASALES の演算は、内部で自動的に AVE.ASALES に変換されます。

集計演算は、次のコマンドで指定することもできます。

```
WRITE SALES AVE.ASALES
```

シノニムでは、REFERENCE= 属性でメジャー集計を指定します。たとえば、REFERENCE=SUM のように指定します。

リクエストに演算接頭語を指定する場合、次のコマンドを発行することにより、アダプタによる演算のメジャー集計演算子との一致の確認を回避することができます。

```
ENGINE BWBABI SET CHECKAGGR OFF
```

すべての階層に対するレポートの規則

変数 SAP BW では、クエリに変数値を含める必要があります。すべての必須変数に対する値を提供する必要があります。変数値を指定するには、TABLE リクエストで WHERE 句または IF 句を使用します。

たとえば、次の WHERE 句では変数の COMPANY_CODE に値 430 を提供します。

```
WHERE COMPANY_CODE_N EQ '430'
```

BY 任意のメンバープロパティおよびプロパティに基づく式 (DEFINE) に辞書ソートを使用することができますが、リクエストでは、階層メンバーを一意的に識別するプロパティである PROPERTY=UID または PROPERTY=NAME に (NOPRINT オプション付きの) ソートを含める必要があります。BY HIERARCHY は、これらのプロパティに自動的に BY を追加します。レベル階層を参照するレポートでは、これはリクエストで指定する必要があります。

階層 同一ディメンション内の1つの階層のみをレポートに使用することができます。

カーディナリティ ディメンション、階層、階層レベルのメンバー数です。SAP BW はマルチディメンションデータベースであるため、返されるディメンションメンバー数を WHERE

テストにより制限しないと、参照するディメンションのカーディナリティの積に応じてレポートの大きさが増大します。

メジャー 合計に適用するフィルタの設定は、WHERE TOTAL テストを使用します。合計の計算は COMPUTE により実行されます。

選別 メンバープロパティによる選別は、ソート付きまたはソートなしで使用できます。

親子階層に対するレポートの規則

BY HIERARCHY SUM、WRITE、ADD コマンドのいずれかを使用する必要があります。PRINT はサポートされていません。

階層コンテキスト レポートに対して WHEN テストを使用してメンバーを選択する場合、選択されたメンバー数が増大しても、階層の各メンバーが同一の階層コンテキスト内に表示されます。階層メンバーのサブツリーは、階層全体のルートを通じたルートとする連続したツリーとして表示されます。追加したコンテキストメンバーには合計は表示されません。

レポートの圧縮 デフォルト設定では、ファクトデータを持たない階層メンバーはレポートに表示されません。また、選別条件に一致するデータを持たない階層メンバーはレポートに表示されません。マルチ階層レポートでは、選択したメンバーのコンテキストのみを表示するため、レポートに出力する外部階層メンバーの内部階層は表示されません。

完全合計および表示合計 各ノードの表示合計は、同一レポートに表示される階層の子の合計で構成されます。これらは、複数動詞のレベルレポート内の同一メンバーの合計と互換性があります。デフォルト設定は表示合計です。メジャーフィールドの FROLL 演算接頭語により、レポートに完全合計を指定することができます。完全合計は、表示される階層メンバー別の累計です。これらは、親子階層上の階層以外のレポートの同一メンバーに対する合計と互換性があります。

レポート作成の一般的なヒント

レポートリクエストを作成する際には、次のヒントを参照してください。

- レベル階層を使用する場合は、PRINT と SUM を相互に代用することができます。

オペレーショナルデータストア (ODS) またはインフォセットクエリ (ODS 結合) からレポートを作成する場合、キーフィールドの合計の設定に応じて PRINT 動詞を使用します。ODS オブジェクトに対して PRINT を使用する場合は、BY フィールド (キー数値

フィールドではなく) によって一意の値の合計レベルが決定されます。

- レポートを最も効率的に実行するには、常にキューブの階層と同一方向にソートします。

たとえば、次のようなディメンションの階層を想定します。

```
Country > State_Province > City
```

この場合、最初に City を、次に State_Province をソートする方法は、効率的なリクエストにはなりません。また、OLAP の階層構造により、レポートの SUM または PRINT の各フィールドに対して検索する BY フィールドは 1 つのみです。

SAP BW の階層のサポート

SAP BW リリース 3.0A 以前では、クエリの単一の特性を展開して階層表示することができません。

動的階層の作成

SAP BW アダプタでは、階層の個々のレベルをその各レベルに対応するフィールド名とともに生成します。Business Number というディメンションおよび States という階層では、レベル 1 は States、レベル 2 は Business Number になります。階層のすべてのフィールドを一度に参照するレポートを作成でき、また階層の最上位レベルを参照してから対話的なドリルダウンを行うことが可能な OLAP トリガレポートとして作成することもできます。

SAP BW リリース 3.0B 以降では、クエリの複数の特性を展開して階層表示することができません。

動的ディメンションの作成

SAP BW では、以下のことが可能です。

- Create New Field (Define) 関数による、動的ディメンションの作成。
- クエリキューブに含まれていないが、実行時に定義または計算される階層レベルとディメンションの使用。

ディメンションの作成

以下のリクエストでは、会社名、現在の日付 (&DATE 変数から取得)、およびその日付の Sales (売上) 金額を連結する Daily Sales (日次売上) というディメンションを作成します。

```
DEFINE Daily Sales/A124= company|&date|'Sales'
END
```

```
TABLE FILE MYQUERY
SUM Retail_Amount
BY Daily Sales
END
```

出力結果は次のとおりです。

Daily Sales		Retail_Amount
ACME	11/15/2002	10,000
SMCE	11/15/2002	15,000

SAP BW メタデータの管理

SAP BW のフィールド名は、BW のマスターデータにロードされたレベルキャプションから生成され、BW のクエリキューブから取得されます。これらのキャプションでは、意味の分かりやすいフィールド名を提供します。

注意

- インフォプロバイダに対するレポートを作成するには、SAP BEx クエリを生成して OLAP にこのクエリをリリースした後、シノニムを作成します。これらの処理を実行するには、SAP ビジネスエクスペローラ (レポートリクエストで参照する SAP BW クエリを作成するツール) を使用します。これらの処理をまだ実行していない場合は、[BEx クエリの作成](#)を確認してから、メタデータの生成に戻ります。
- シノニムを作成するためには、汎用グループに対する権限の SU_USER を必要とします。
- SAP BW のマスターデータテーブルへのアクセス。**多くの組織では、以下のような場合に SAP BW 内にあるトランザクションデータと組み合わせたマスターデータの分析が必要になります。

- トランザクションデータが頻繁に変化する場合 (たとえば、会計データ、販売と購買の注文、在庫など)。このデータはインフォキューブ内に格納され、SAP BW アダプタを介して直接アクセスできます。このデータは最新の分析に使用できます。
- マスターデータは長期間にわたって変化せず、常に同じように必要です (たとえば、アクセス権限、標準のプリンタ情報、組織構造など)。ただし、このデータはキューブベースの SAP BW 環境で初めはアクセスできない可能性がある SAP BW マスターデータテーブル内に格納されます。

この複雑な分析はマスターデータへのアクセスに基づくため、ユーザはマスターデータテーブルを SAP BW 環境内でアクセス可能にできる必要があります。これは、テーブルベースの SAP ECC アダプタを使用して、簡単な構成手順を完了することによって行えます。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な SAP BW テーブルまたはビューの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは一時項目 (DEFINE) や他のセキュリティ機構などの、サーバの拡張メタデータ機能をサポートします。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。

注意：シノニムを作成するためには、汎用グループに対する権限の SU_USER を必要とします。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

SAP BW の拡張シノニム作成パラメータ

以下のリストは、アダプタのシノニムを作成する際に、値の指定が必要なパラメータ、および実行する関連作業を示しています。これらのオプションが複数のウィンドウに表示される場合もあります。ウィンドウに表示されたボタンをクリックして次のウィンドウに進み、最後に [シノニムの作成] ボタンをクリックします。これで、入力データに基づいてシノニムが作成されます。

カタログの選択

ドロップダウンリストからカタログを選択することができます。デフォルト設定では、[すべてのカタログ] が選択されます。

カタログおよびキューブでフィルタ

利用可能なすべてのクエリを表示するには、[カタログおよびキューブでフィルタ] チェックボックスをオフにします。

カタログ名、キューブ名、または説明によるフィルタを設定するには、このチェックボックスをオンにします。3つの入力ボックスが表示されます。適切なボックスに、カタログ (インフォキューブ)、キューブ (クエリキューブ)、または説明の名前を入力します。必要に応じて、ワイルドカード文字 (*) を使用することもできます。以下はその例です。

0CCA*

クエリキューブに対してリクエストを発行すると、アダプタでは、コンポーネント (完全カタログ名、フォワードスラッシュ (/)、クエリキューブ名の順序で構成されます) からキューブ名を作成します。

説明を入力する場合、この説明はインフォキューブまたはクエリキューブからの SAP BW の説明の連続した部分である必要があります。

複数言語

シノニムは、1つまたは複数の言語で作成することができます。複数の言語を選択した場合、選択したそれぞれの言語に対してタイトルと説明の属性を指定することができます。複数言語のタイトルおよび説明についての詳細は、[複数言語のタイトルと説明](#)を参照してください。

1. [複数言語] のチェックをオンにして、デフォルトのログイン言語 (サーバのスタートアップ時の SET LANGUAGE コマンドの値) 以外の追加言語を選択します。使用可能な言語のリストが表示されます (デフォルトの言語はリスト上に表示されません)。
2. 1つまたは複数の追加言語を選択することができます。

SAP BW では、アクセスする言語ごとに権限が必要になります。ユーザ ID およびパスワードは sapserv.cfg に格納されます。言語は、このリストで選択したものから取得されます。

ただし、選択した言語を実装するには、その言語をインストールしておく必要があることに注意してください。また、NLS を有効にし、NLS 構成ウィザードで設定したコードページ設定と選択した言語を一致させる必要があります。WebFOCUS Reporting Server ブラウザインターフェースからこのウィザードにアクセスするには、[ワークスペース] ページのメニューバーから [設定]、[LOCALE]、[構成ウィザード] を順に選択します。詳細は、[コードページおよび複数言語のシノニム](#)を参照してください。

また、[フィールド名] オプションも表示されます (詳細は、次の項目を参照)。

フィールド名

複数言語のシノニムを選択した場合、[テクニカル] または [言語] ベースのフィールド名の使用を選択することができます。

- テクニカル名は、SAP BW オブジェクトのグローバル ID に基づいた名前です。
- [言語] 名では、BW オブジェクトのキャプション (説明) が使用されます。[言語] ベースの名前がデフォルト設定です。

[言語] ベースの名前を選択した場合、フィールド名の大文字と小文字オプションが使用可能となります (詳細は、次の項目を参照)。

フィールド名の大文字と小文字

[言語] ベースのフィールド名では、次の [大文字] または [混在] を選択します。

- [大文字] では、シノニムのマスターファイルの構成要素のフィールド名がすべて大文字で表示されます。
- [混在] では、マスターファイルのフィールド名の大文字小文字は、SAP BW で使用されている大文字小文字と同じように表示されます。デフォルト設定は [混在] です。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には

「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

階層のタイプ

シノニムに作成する階層のタイプを選択します。[HType] ドロップダウンリストから次のいずれかのオプションを選択します。

- 最適化済み** このオプションでは、階層が親子関係 (フラット階層を除く) で表されるシノニムが作成されます。フラット階層 (ルートノードを親とするすべてのリーフノードのリスト) は 2 つのフィールドを持つ 1 つのレベル階層として表現されます。このフィールドの 1 つはメンバーのキャプションを持つ *hierarchy_name_LEVEL_01* という形式であり、もう 1 つはメンバー名を持つ *hierarchy_name_NAME* という形式です。
- レベル** このオプションでは、以前のバージョンと同様、階層ごとに異なるフィールドを作成します。

注意：以前のバージョンのアダプタで作成された既存のシノニムをリクエストで使用する場合、使用前にこれらのシノニムのマイグレートが必要になる場合があります。シノニムのマイグレートを実行するには、[メタデータ] ウィンドウでシノニム名をクリックし、メニューから [マイグレート] オプションを選択します。

メンバーのサンプリングを実行する

このチェックボックスをオンにすると、アダプタにより階層レベルのすべてのメンバーが選択されて、フィールドの最適な長さおよびカーディナリティの決定が行われます。

注意：データ量によっては、この処理に時間を要する場合があります。シノニムの作成に時間がかかる場合、[メンバーのサンプリングを実行する] チェックボックスをオフにすることができます。

このチェックボックスをオフにすると、階層レベルのすべてのフィールドの長さは 60 バイトに設定され、カーディナリティ情報は空白になります。

構造化ディメンション

構造は特性とキー数値 (いずれかまたは両方) で構成されます。これらを保存し、他のクエリで再利用することができます。SAP BW 構造はディメンションとして表すことができ、また構造内のフィールドの直積を作成することもできます。構造をディメンションとしてシノニムに含めるには、このチェックボックスをオンにします。

このチェックボックスをオフにすると、作成されたマスターファイルには、構造内のすべてのフィールドの直積を表すフィールドが含まれます。これらのすべてのフィールド

は、マスターファイルの MEASURES セクションの下に配置され、個別のキー数値として処理されます。

Level0 の追加

レベル階層マスターファイルを使用した場合、ほとんどの最上位階層ノード (レベル 0) には、「すべてのメンバー」(All member) または「未割り当て」(Not assigned) ノードのみが含まれます。レポートの作成では、ユーザが階層ノード認証 TYPE 1 で制限され、メタデータで LEVEL HIERARCHIES を使用する場合、これらはほとんど使用されません。

キー日付

指定されたキー日付に関連付けられている期間の間に存在した、すべてのバージョンの階層 (時間依存および非時間依存) を含むマスターファイルを生成します。複数の同時バージョンが存在する、または存在していた時間依存階層がある場合、シノニムの作成時にキー日付を指定することはほぼ理にかなっています。キー日付は、階層の同時バージョンの数が最大になったときの期間に対して指定する必要があります。

キー日付が指定されない場合、現在の日付が使用されます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

生成するシノニムの選択

シノニムを作成するキューブ名の横のラジオボタンをクリックします。リスト上のキューブすべてにシノニムを作成するには、キューブのリストの上のラジオボタンをクリックします。[デフォルトシノニム名] カラムには、キューブを選択した場合にデフォルトで生成されるシノニム名が指定されています。生成されたシノニムにデフォルト以外の名前を指定するには、デフォルト名を上書きします。

メジャー、ディメンション、または階層をシノニムから削除するには、シノニムの作成後、ibi Data Migrator デスクトップインターフェースを使用します。

複数言語のタイトルと説明

複数言語のシノニムを作成すると、そのシノニムには選択した各言語の TITLE および DESCRIPTION 属性が追加されます。たとえば、フランス語を選択した場合、シノニムには

次のように TITLE および DESCRIPTION 属性が追加されます。

```
TITLE_FR='french title'  
DESC_FR='french description'
```

デフォルトのログイン言語は、TITLE および DESCRIPTION 属性を使用してそれぞれのタイトルと説明を生成します。

- TITLE および DESCRIPTION 属性を追加した言語に LANGUAGE パラメータを設定すると、タイトルと説明はそれぞれ TITLE_In と DESC_In 属性として生成されます (ここで、In は EN、DE、JA などの 2 文字の ISO コード)。
- タイトルと説明を追加していない言語に LANGUAGE パラメータを設定すると、TITLE および DESCRIPTION 属性を使用してタイトルと属性が生成されます。

シノニム言語のフィールド名を他と重複しない名前にするために、フィールド名はそのフィールドのユニーク名 (技術名と呼ばれる) に基づいて作成されます。デフォルト言語のみで作成されたシノニムでは、フィールドのキャプションに基づいてフィールド名が作成されます。

コードページおよび複数言語のシノニム

複数言語のシノニム用に選択した言語により、構成時に使用するコードページが決定されます。たとえば、ISO 8859-1 は、ほとんどの西ヨーロッパ言語に対応します。そのため、このコードページを使用すれば、英語、ドイツ語、フランス語、スペイン語で記述したリクエストを作成することができます。

Unicode UTF-8 のコードページ (65001) は、すべての主要言語をサポートするため、あらゆる言語の組み合わせに使用することができます。ただし、UTF-8 文字エンコードスキームは、ASCII 環境では 3 バイトを使用して文字を表示するため、より大きい文字データの格納領域が必要となります。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

メタデータのマッピング

OLAP (OLE DB 標準) では、行セットに対して、階層、レベル、メジャー、メンバー、およびディメンションプロパティという各用語を使用します。対応する用語が BW と SAP BW アダプタでマッピングされます。

データがキューブにロードされた場合は必ず、アダプタのメタデータを再生成することをお勧めします。再生成によりデータが最新状態に保たれ、またアダプタで各ディメンション階層が拡張する範囲を正確に記録することができます。

BW インフォプロバイダの OLE/DB キューブへのマッピング

ビジネスインフォメーションウェアハウス (BW) のインフォプロバイダは、OLE/DB のキューブに類似しています。ともに中核となるメタデータオブジェクトであり、レポート作成に使用されるデータの格納場所です。インフォプロバイダには、キー数値と特性という 2 つのタイプのデータが格納されます。関連情報については、[BEx クエリの作成](#)を参照してください。これらのマッピングの例については、[マッピングの例](#)を参照してください。

OLAP	BW の接続属性	FOCUS
キューブ	クエリ	マスターファイル
ディメンション	特性	内部的な使用法がマスターファイルの注釈として示されます (上記参照)。
階層	階層	ディメンション (SAP BW アダプタのディメンションは、WITHIN キーワードによって関連付けられたフィールドのグループです。WITHIN キーワードによって、集計のために内部の値が外部の値と関連付けられます。)
レベル	レベル	フィールド (親フィールドを指示する WITHIN キーワード)。

OLAP	BW の接続属性	FOCUS
メンバー	特性値	フィールドの値。
ディメンションプロパティ (特殊なプロパティキーを含む)	属性	フィールド (参照するディメンション階層が BY フィールドとして使用されている場合に限り、表示可能です)。
OLAP マッピングなし	SAP BW の変数	オプションまたは必須の入力をとるフィールド (特にマスターファイルにマッピングされる)。
メジャー	キー数値 (メジャー)	FOCUS の動詞 (PRINT/SUM) で処理を実行できる数値フィールド。

マッピングの例

以下の各行の構文は、SAP BW アダプタのメタデータの構成の例です。それぞれ対応する定義とともにリストされています。

FILENAME=WAREHOUS, SUFFIX=BWBAPI,\$

キューブの記述子を示します。FILENAME=値は、クエリキューブ名から生成されます。

SEGNAME=WAREHOUS, SEGTYPE=S0,\$

セグメント名 (FOCUS でのみ内部的に使用) です。FILENAME=値と同様に、SEGNAME=値が生成されます。

\$ DIMENSION -MEASURES-

マスターファイルのメジャーセクションの開始を示します。ファクトテーブルは論理的にディメンションであり、その結果マスターファイル内に表示されます。

\$ CARDINALITY=n

ディメンションのカーディナリティでは、レポートで取得可能なメンバーの数が定義されます。ディメンションのカーディナリティが大きい場合、返される値を限定するために

WHERE 条件の使用を検討してください。

FIELD=STORE_INVOICE, ALIAS='STORE INVOICE', USAGE=D10.2, ACTUAL=D8.2, MISSING=OFF,\$

メジャーを定義します。マスターファイルの最初のセクションでは、キューブのファクトテーブルを構成する各フィールドが定義されます。

\$ DIMENSION: dimension_name

ディメンションセクションの注釈です。SAP BW アダプタのシノニムのロジックによって、対応する階層の説明の前の注釈にディメンション名が挿入され、マスターファイルの可読性が向上します。ディメンションの注釈の後ろに続く列は、すべてディメンションのメンバーです。

注意：同一ディメンションに対する複数の階層は、個々のディメンション項目として表示されます。

FIELD= PRODUCT_FAMILY, ALIAS='PRODUCT FAMILY', WITHIN= *PRODUCT, USAGE=A20, ACTUAL=A20, MISSING=OFF,\$

ディメンション階層の最上位レベルです。階層のそれ以降の個々のレベルには、対応するフィールド名があります。階層の各レベルには、レベルで一意的なカーディナリティ (そのレベルのメンバーのカウント) を含む次の注釈行が含まれています。

注意：WITHIN=値の文字は、66 バイト以内で指定します。階層の固有の名前が 65 バイト (および * 用の文字を加えた 66 バイト) より長い場合、[シノニムの作成] でメッセージが出力されます。OLAP では集計レベル (階層のルートレベル) をサポートしますが、これらはマスターファイル内には表示されません。

FIELD=STORE_MANAGER, ALIAS='STORE MANAGER', USAGE=A20, ACTUAL=A20, MISSING=ON,\$

ディメンションのプロパティを記述しています。割り当てられた階層を持たない (例、WITHIN の指定がない)、マスターファイルのディメンションセクションにリストされたすべてのフィールドは、ディメンションのプロパティを表します。この例では、プロパティはマネージャ名をロールアップに格納します。

マスターファイルの構文規則

SAP BW で使用するマスターファイルでは、以下の規則に従う必要があります。

- **JOIN の制限** JOIN は、BW の構文に関する規則ではサポートされません。ただし、

MATCH FILE を使用して同じ結果を得ることができます。

- **引用符が必要** ブランクを含むフィールド名または階層名は、一重引用符 (') で囲む必要があります。
- **階層レベル** 階層レベルには、明確に定義されたデータタイプがありません。このため、レベルは常に alpha で表されます。
- **特別なプロパティ - キー** BW のディメンションには、level_01 が付加されたディメンションの名前によって示されるデフォルトの階層があります。これらのデフォルトの階層フィールドには、BW のフィールドの内部参照コードが含まれた「キー」と呼ばれる特別なプロパティがあります。キーはデフォルトの階層に対してのみ出力され、それを参照する場合にのみ使用できます。また、ディメンションフィールドを参照する場合、キーフィールドを WHERE または DEFINE ステートメントで使用することもできます。シノニムでは、このフィールドのフィールド名に接尾語の _NAME が付加されます。関連情報については、[すべての階層に対するレポートの規則](#)を参照してください。

ディメンション - 0MATERIAL

次の例では、レポートでフィールドを選択した場合に予想できる結果を示しています。同じデータ値に対して 2 つのフィールドが返されていることに注目することが重要です。

```
Field:0MATERIAL_LEVEL01
Field:0MATERIAL_NAME
```

レポートでは、以下のように表示されます。

0MATERIAL_LEVEL_01	0MATERIAL_NAME	TOTAL SALES
Fitdrink 2000(CAN)	R100032	\$1000.00

両方のフィールドで同じ特性値が表されています。

0MATERIAL_LEVEL_01 は、ディメンションのすべてのメンバーのキャプションが含まれたディメンションレベルです。

0MATERIAL_NAME には、ディメンションのメンバーの SAP BW 技術名称が含まれています。

変数のタイプ

SAP BW では、以下の 4 つのタイプの変数がサポートされます。

- **メンバー** 個別の TABLE リクエストで表示することができます。TABLE クエリでの使用時に、変数のタイプが必須の場合 (およびデフォルト値がない場合)、検索条件を指定する必要があります。検索条件では、変数に対応する変数の値のセット、単一の値、間隔、または複合 (複合には組み合わせに関する制限はありません) を提供する必要があります。各検索述語 (AND 論理式) では、1 つの変数のみを参照でき、他のキューブの要素は参照できません。
- **ノード** 個別の TABLE リクエストで表示することができます。ノードタイプの変数には、単一の値の選択タイプがあります。
- **階層** 個別の TABLE リクエストで表示することができます。階層タイプの変数には、単一の値の選択タイプがあります。
- **数値** 表示することはできません。

関連情報については、[変数を使用したレポート](#)を参照してください。

カスタマイズの設定

以下の各トピックでは、SAP BW アダプタをカスタマイズできる高度な設定コマンドについて説明します。

空のレポートのセル設定の指定

```
ENGINE BWBAPI SET EMPTY {ON|OFF}
```

説明

ON

空のセルがレポートに含まれること、およびデータがディメンションのメンバーに関連付けられているかどうかに関係なく、すべてのデータが表示されることを示します。

OFF

空のセルがレポートから除外されること (メジャーを参照する場合)、およびファクトデータがディメンションのメンバーに関連付けられていない場合、ディメンションのデータは表示されない (メジャーを参照しない場合) を示します。デフォルト値は OFF です。

ヒント：この設定は、WebFOCUS Reporting Server ブラウザインターフェースのメニューバーで [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択して変更できます。

メンバーのサンプリング設定の指定

シノニムの作成時に、メンバーのサンプリングを使用して、ディメンションのメンバーのキャプションに正確に一致するディメンションレベルに対するマスターファイルのフィールドが作成されます。

```
ENGINE BWBAPI SET MEMBER_SAMPLING {ON|OFF}
```

説明

ON

アダプタにより、階層レベルのすべてのメンバーが選択されて、フィールドの最適な長さとカーディナリティが決定されることを示します。デフォルト値は [ON] です。

注意：データ量によっては、この処理に時間を要する場合があります。シノニムの作成に時間がかかる場合、MEMBER_SAMPLING を OFF にすることができます。

OFF

階層レベルのすべてのフィールドの長さが 60 バイトに設定され、カーディナリティ情報は空白になります。

ヒント：この設定は、WebFOCUS Reporting Server ブラウザインターフェースのメニューバーで [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択して変更できます。

ブロックサイズの指定

SAP では、SAP サーバのクラッシュを避けるために、サイズの大きい行およびセルのセットを部分ごとにフェッチすることを推奨しています。BLOCK_SIZE 設定で一部分のサイズを設定します。

```
ENGINE BWBAPI SET BLOCK_SIZE n
```

説明

n

一部分のサイズを設定します。デフォルト値の 0 は、結果セットのすべてのメンバーが一部分で返されることを意味します。

この設定値は、タイムアウトが存在する場合を除き、変更しないでください。このオプションの使用法の詳細は、技術サポート担当者にお問い合わせください。

ヒント：この設定は、WebFOCUS Reporting Server ブラウザインターフェースのメニューバーで [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択して変更できます。[設定の変更] ウィンドウが表示されます。

プロパティの制限の指定

SAP BW 3.0B では、MEMBERS 行セット内の属性値に対するクエリの実行機能が使用できるようになりました。(詳細は、SAP OSS ノート 609314 を参照してください。)

```
ENGINE BWBAPI SET PROPERTY_RESTRICTIONS {ON|OFF}
```

説明

ON

ディメンションの属性に対するテストが SAP に送信されます。

使用する SAP BW のリリースおよびパッチレベルで、ユーザ定義プロパティを選択テーブルとして制限することをサポートする場合、PROPERTY_RESTRICTIONS を ON に設定することをお勧めします。(リリースおよびパッチレベルの正確なリストについては、SAP サポートにお問い合わせください。)

OFF

WebFOCUS サーバでフィルタを実行することを示します。デフォルト値は OFF です。

ヒント：この設定は、WebFOCUS Reporting Server ブラウザインターフェースのメニューバーで [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択して変更できます。[設定の変更] ウィンドウが表示されます。

未割り当てメンバーの設定の指定

ディメンションのメンバーが、階層に明示的に配置されないことがあります。SAP BW では、割り当てられたメンバーは All というルートノードの子と見なされ、割り当てられていないメンバーは All Not Assigned という 2 番目のルートノードの子と見なされます。デフォルト設定では、割り当てられていないメンバーはレポートに含められません。それらは、INCLUDE_UNASSIGNED 設定を使用して含めることができます。

```
ENGINE BWBAPI SET INCLUDE_UNASSIGNED {OFF|ACTIVE|ALL}
```

説明

OFF

アクティブ階層に属する割り当てなしのメンバーは、ターゲットレポートに表示されません。デフォルト値は OFF です。

ALL

アクティブ階層に属している割り当てられていないメンバーすべてを、ターゲットレポートに含めます。

ACTIVE

最下位レベルが参照される、アクティブ階層に属している割り当てられていないメンバーを含めます。割り当てられていない個々のメンバーは、階層の最下位レベルにリストされます。この値は、レベル階層のみでサポートされます。

クエリに最下位レベルが含まれていない場合、特定の WHERE/IF ステートメントを最下位レベルに指定していない限り、割り当てられていないノードの集計は表示されません。

ヒント：この設定は、WebFOCUS Reporting Server ブラウザインターフェースのメニューバーで [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキスト

メニューから [設定の変更] を選択して変更できます。[設定の変更] ウィンドウが表示されます。

集計演算子の確認の無効化

リクエストに演算接頭語を指定する場合、デフォルト設定では、アダプタによってこの演算子が BW データ内のメジャー集計演算子と互換性があるかが検査されます。互換性がない場合メッセージが返され、リクエストは処理されません。演算接頭語がメジャー集計演算子と一致するかどうかのアダプタによるこのチェックは、CHECKAGGR OFF コマンドを発行することにより回避できます。

```
ENGINE BWBAPI SET CHECKAGGR {ON|OFF}
```

説明

ON

個々の演算接頭語を BW 内の対応するメジャー集計演算子と照合して検査します。演算接頭語が対応するメジャー集計演算子と一致しない場合メッセージが表示され、リクエストは処理されません。デフォルト値は ON です。

OFF

演算接頭語をメジャー集計演算子と照合して検査しません。

ヒント：この設定は、WebFOCUS Reporting Server ブラウザインターフェースのメニューバーで [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択して変更できます。[設定の変更] ウィンドウが表示されます。

時間依存階層に対するレポート作成

時間依存階層を持つシノニムに対してレポートを作成する場合は、ENGINE BWBAPI SET KEY_DATE コマンドを使用して、後続の TABLE リクエストのキー日付を設定します。

```
ENGINE BWBAPI SET KEY_DATE yyyyymmdd
```

説明

yyyymmdd

後続の TABLE リクエストのキー日付です。キー日付を指定しない場合は、現在の日付がキー日付として使用されます。正しいキー日付 (指定した階層の同時バージョンに関連付けられた期間内のキー日付) を指定しなかった場合は、レポートが失敗し、指定した階層が存在しないことを示すエラーメッセージが表示されます。

リクエストでの大規模インスタンスのサポート

SAP から新しく提供された一連の OLAP BAPI では、軸あたり 100 万件を超えるインスタンスのリクエストがサポートされます。SET USE_RSR_MDX コマンドを使用すると、SAP BW アダプタが、この新しい BAPI を利用できるようになります。

```
ENGINE BWBAPI SET USE_RSR_MDX {OFF|ON}
```

説明

OFF

軸あたり 100 万件を超えるインスタンスのリクエストをサポートしません。デフォルト値は OFF です。

ON

軸あたり 100 万件を超えるインスタンスのリクエストをサポートします。

リテラル文字列による SAP VARIABLES 句の作成

アダプタで SAP VARIABLES 句を生成するには、次の 2 つの異なる方法があります。

- リテラル文字列の使用。
- メンバーの固有の名前の使用。

この場合、リテラル文字列の使用がより効率的であると考えられています。これは、階層のメンバーを読み取って階層のメンバーの固有の名前を取得するために、アダプタで BW サーバと通信する必要がないためです。ただし、リテラル文字列の生成は、変数フィールドに対するテストのタイプ、または関連付けられた階層と変数の選択タイプのタイプによって可能である場合も可能でない場合もあります。また、リテラルを使用した MDX リクエストの成功は、SAP によって明確に記述されていない BW の内部的な側面によって左右されます。

以下の設定により、アダプタでメンバーの固有の名前を使用した技法を使用する前に、リテラルを使用して SAP VARIABLES 句の作成を試みるかどうかは制御されます。

```
ENGINE BWBAPI SET USE_SAP_VARIABLE_LITERALS {ON|OFF}
```

説明

ON

アダプタにリテラルを使用して SAP VARIABLES 句の作成を試みるように指示します。

OFF

アダプタにメンバーの固有の名前を使用して SAP VARIABLES 句の作成を試みるように指示します。デフォルト値は OFF です。

BEx 構造のサポート

構造は特性とキー数値 (いずれかまたは両方) で構成されます。これらを保存し、他のクエリで再利用することができます。たとえば、計画/実績の比較が広範に使用されている場合、計画/実績の比較用に並べられた 10 のキー数値から成るキー数値グループを保存することができます。同じキー数値一式を別のクエリ内で使用できるように、10 のキー数値を個々に選択するのではなく、この構造を BEx クエリに追加します。これにより、保存したこれらのキー数値がその別のクエリに自動的に追加されます。

SAP では BEx クエリで最大 2 つの構造がサポートされますが、それらの構造のうちの 1 つしかキー数値タイプにできないことに注意してください。

構造化ディメンションを持つシノニムは、シノニムの作成時に [構造化ディメンション] のチェックをオンにすることで作成することができます。

構造をディメンションまたは直積として表す

構造はディメンションとして表すことができ、また構造内のフィールドの直積を作成することもできます。以下の 2 つの構造を使用する BEx クエリについて考えてみます。

- 1 つ目の構造は、**Curr.Year vs. Prev. year Cumulated** と呼ばれ、Year と Prev.Year の 2 つのディメンションが含まれています。
- 2 つ目の構造は、**Orders-Sales>Returns** と呼ばれ、Incoming Orders、Open Orders、Sales Volume、および Returns という 4 つのキー数値が含まれています。

構造化ディメンションがない場合、生成されるマスターファイルには 2 つの構造の直積を表す 8 つのフィールドが含まれています。これらのすべてのフィールドは、マスターファイルの MEASURES セクションの下に配置され、個別のキー数値として処理されます。マスターファイルは、以下ようになります。

```
FILENAME=BW31C/Z2STRUCTURES_1KF_ROW_1COL, SUFFIX=BWBAPI  , $
SEGMENT=Z2STRUCT, SEGTYPE=S0, $
$ -MEASURES-
  FIELDNAME=CURRENT_YEAR_CUMULATED_INCOMING_ORDERS,
    ALIAS=3466HXATG6RG1BG9GSTCN65IR3RV6E, USAGE=D18.2,
    ACTUAL=D8, MISSING=ON,
    TITLE='Current year cumulated Incoming Orders', $
```

```
FIELDNAME=CURRENT_YEAR_CUMULATED_OPEN_ORDERS,
  ALIAS=3466HXATG6RG1BG9GSTCN65IR3RV6E, USAGE=D18.2,
  ACTUAL=D8, MISSING=ON,
  TITLE='Current year cumulated Open orders', $
```

```
FIELDNAME=CURRENT_YEAR_CUMULATED_SALES_VOLUME,
  ALIAS=3466HXATG6RG1BG9GSTCN65IR3RV6E, USAGE=D18.2,
  ACTUAL=D8, MISSING=ON,
  TITLE='Current year cumulated Sales Volume', $
```

```
FIELDNAME=CURRENT_YEAR_CUMULATED_RETURNS,
  ALIAS=3466HXATG6RG1BG9GSTCN65IR3RV6E,
  USAGE=D18.2, ACTUAL=D8, MISSING=ON,
  TITLE='Current year cumulated Returns', $
```

```
FIELDNAME=PREV__YEAR_CUMULATED_INCOMING_ORDERS,
  ALIAS=36AWN99ETKI46HMTOGIVHHSOJ3RV6E,
  USAGE=D18.2, ACTUAL=D8, MISSING=ON,
  TITLE='Prev. year cumulated Incoming Orders', $
```

```
FIELDNAME=PREV__YEAR_CUMULATED_OPEN_ORDERS,
  ALIAS=36AWN99ETKI46HMTOGIVHHSOJ3RV6E,
  USAGE=D18.2, ACTUAL=D8, MISSING=ON,
  TITLE='Prev. year cumulated Open orders', $
```

```
FIELDNAME=PREV__YEAR_CUMULATED_SALES_VOLUME,
  ALIAS=36AWN99ETKI46HMTOGIVHHSOJ3RV6E,
  USAGE=D18.2, ACTUAL=D8, MISSING=ON,
  TITLE='Prev. year cumulated Sales Volume', $
FIELDNAME=PREV__YEAR_CUMULATED_RETURNS,
  ALIAS=36AWN99ETKI46HMTOGIVHHSOJ3RV6E,
  USAGE=D18.2, ACTUAL=D8, MISSING=ON,
  TITLE='Prev. year cumulated Returns', $
$ DIMENSION @DISTR_CHAN (Distribution Channel):
$ HIERARCHY (Distribution Channel):
...
...
```

以下は、このマスターファイルに対する WebFOCUS リクエストの例です。

```
TABLE FILE Z2STRUCTURES_1KF_ROW_1COL
SUM CURRENT_YEAR_CUMULATED_INCOMING_ORDERS
  CURRENT_YEAR_CUMULATED_OPEN_ORDERS
  CURRENT_YEAR_CUMULATED_SALES_VOLUME
  CURRENT_YEAR_CUMULATED_RETURNS
  PREV__YEAR_CUMULATED_INCOMING_ORDERS
  PREV__YEAR_CUMULATED_OPEN_ORDERS
  PREV__YEAR_CUMULATED_SALES_VOLUME
  PREV__YEAR_CUMULATED_RETURNS
BY @DISTR_CHAN_KEY
END
```

構造化ディメンションがある場合、生成されるマスターファイルには、マスターファイルの MEASURES セクションの下に配置された 4 つのキー数値が含まれています。**Curr.Year vs. prev. year Cumulated** 構造は、ディメンションとして処理されます。マスターファイルは、以下ようになります。


```

FILENAME=BW31C/Z2STRUCTURES_1KF_ROW_1COL_STON, SUFFIX=BWBAPI , $
SEGMENT=Z2STRUCT, SEGTYPE=S0, $
$ -MEASURES-
  FIELDNAME=INCOMING_ORDERS, ALIAS=[3RV6ENA572BRXFKFCO2M2ZUWS],
  USAGE=D18.2, ACTUAL=D8, MISSING=ON, TITLE='Incoming Orders', $
  FIELDNAME=OPEN_ORDERS, ALIAS=[3RV6ENHTQ0XHGG23VII4YD1TMK],
  USAGE=D18.2, ACTUAL=D8, MISSING=ON, TITLE='Open orders', $
  FIELDNAME=SALES_VOLUME, ALIAS=[3RV6ENPI8ZJ6YONBOC7AN3SCC],
  USAGE=D18.2, ACTUAL=D8, MISSING=ON, TITLE='Sales Volume', $
  FIELDNAME=RETURNS, ALIAS=[3RV6ENX6RY4WHB6RU69MX5R24], USAGE=D18.2,
  ACTUAL=D8, MISSING=ON, TITLE='Returns', $
$ DIMENSION 0DISTR_CHAN (Distribution Channel):
$ HIERARCHY (Distribution Channel):
...
...
$ DIMENSION EAZ5PAJIWG37DI674JZI2C3UB (Curr. year vs. prev. year
(cumulated to last period)):
$ HIERARCHY (Curr. year vs. prev. year (cumulated to last period)):
  FIELDNAME=CURR__YEAR_VS__PREV__YEAR__CUMULATED_TO_LAST_PERIOD_,
  USAGE=A22, ACTUAL=A22, MISSING=ON,
  TITLE='Curr. year vs. prev. year (cumulated to last period)',
  WITHIN='*[EAZ5PAJIWG37DI674JZI2C3UB]', $
$ Cardinality:2
$ DIMENSION EAZ5PAJIWG37DI674JZI2C3UB, PROPERTIES:
  FIELDNAME=EAZ5PAJIWG37DI674JZI2C3UB_KEY, ALIAS=MEMBER_NAME,
  USAGE=A64, ACTUAL=A64, MISSING=ON,
  TITLE='Curr. year vs. prev. year (cumulated to last period) Key', $

```

以下は、このマスターファイルに対する WebFOCUS リクエストの例です。

```

TABLE FILE Z2STRUCTURES_1KF_ROW_1COL_STON
PRINT INCOMING_ORDERS
      OPEN_ORDERS
      SALES_VOLUME
      RETURNS
BY CURR__YEAR_VS__PREV__YEAR__CUMULATED_TO_LAST_PERIOD_
END

```

重要： BEx クエリに構造が 2 つしか含まれておらず、自由特性のディメンションが存在しないか、行および列にディメンションが含まれていない場合、構造化ディメンションを使用する必要があります。構造化ディメンションがない場合、生成されるシノニムにはメジャーのみ含まれ、ディメンションはありません。このようなシノニムに対して作成した WebFOCUS リクエストでは、以下のメッセージが出力されます。

(FOC11205) マスターファイルにエラーがあります:

(FOC11229) MDX セグメントの記述に階層がありません。

SQL を使用した SAP BW リクエストの生成

以下のリクエストと出力結果は、標準の ANSI SQL コードを使用した SAP BW アダプタの機能の例です。

これらの例では、WebFOCUS Reporting Server ブラウザインターフェースから関連するテーブルに対してシノニムを作成しておく必要があります。WebFOCUS Reporting Server ブラウザインターフェースを使用して、関連するテーブルを検索するか、名前による単純な検索を実行して、SAP アプリケーションの階層を移動することができます。

注意：SQL では、階層レポート (BY HIERARCHY) はサポートされません。

SAP BW からの値の取得

以下の構文は、SAP BW データベースから値を取得する構文です。

```
SELECT SUM(DISBURSEMENT),ADDRESS1_LEVEL_01 FROM ZPAY4
GROUP BY ADDRESS1_LEVEL_01
ORDER BY ADDRESS1_LEVEL_01;
```

出力結果は次のとおりです。

address1 Level 01	disbursement
1106 BROADWAY	132,242.33
11607 PACIFIC BLVD	9,174.96
11607 TREMONT AVENUE	14,849.56

address1 Level 01	disbursement
1200 RIVERDALE	9,983.99
187 WEST 39TH ST	9,209.97
2601 8TH AVENUE	26,274.65
31-18 SOUTHERN BLVD	1,165.64
719 SEVENTH STREET	10,800.00
DRAWER 4, COOPER STA	21,406.32
P.O. BOX 1206	53,636.74
P.O. BOX 17616	6,703.90

階層からの項目の選択

以下の構文は、MARA 内の階層から項目を選択する構文です。

```
SELECT SUM(SALES_VOLUME), SUM(INCOMING_ORDERS),
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_01,
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_02 FROM ZBIGQ
GROUP BY PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_01,
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_02
ORDER BY PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_01,
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_02
```

出力結果は次のとおりです。

Product Hierarchy for material MARA Level 01	Product Hierarchy for material MARA Level 02	Sales Volume	Incoming Orders
Foods	Bakery product;	.00	400.00

Product Hierarchy for material MARA Level 01	Product Hierarchy for material MARA Level 02	Sales Volume	Incoming Orders
	configured to order		
Hardware	PCs	85,159,860.98	86,754,997.71
	Printer	.00	60,011.10
Lighting	Bulbs	62,235,898.00	62,674,320.00
Machines	Pumps	44,328,485.55	46,231,307.55
Paints	Gloss paints	.00	125.03
Services	Maintenance	291,562.54	485,315.83
Vehicles	Cars	155,640.00	155,640.00
	Motorcycles	41,761,619.76	42,494,811.00

階層の完全なパスの選択

次の構文は、MARA から完全な階層パスを取得します。

```
SELECT SUM(SALES_VOLUME), SUM(INCOMING_ORDERS),
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_01,
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_02,
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_03,
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_04
FROM ZBIGQ
GROUP BY
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_01,
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_02,
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_03,
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_04
ORDER BY
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_01,
```

```
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_02,  
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_03,  
PRODUCT_HIERARCHY_FOR_MATERIAL_MARA_LEVEL_04;
```

出力結果は次のとおりです。

Product Hierarchy for material MARA Level 01	Product Hierarchy for material MARA Level 02	Product Hierarchy for material MARA Level 03	Product Hierarchy for material MARA Level 04	Sales Volume	Incoming Orders
Foods	Bakery product; configured to order			.00	400.00
Hardware	PCs	Drive	Harddisk 1080 MB/SCSI-2-Fast	4,234,580.64	4,094,612.64
			Harddisk 2113 MB/ATA-2	5,216,171.71	5,216,171.71
			Harddisk 2149 MB/SCSI-2-Fast	4,727,005.36	4,713,724.40
			Harddisk 4294 MB/SCSI-2-Fast	5,132,812.80	5,132,812.80
		Input device	Professional keyboard -	720,022.65	714,750.89

Product Hierarchy for material MARA Level 01	Product Hierarchy for material MARA Level 02	Product Hierarchy for material MARA Level 03	Product Hierarchy for material MARA Level 04	Sales Volume	Incoming Orders
			MAXITEC Model		
			Professional keyboard - NATURAL Model	813,765.24	788,838.81
			Professional keyboard - PROFITEC Model	576,079.36	576,079.36
			Standard Keyboard - EURO Model	620,294.21	600,911.21
			Standard Keyboard - EURO-Special Model	633,094.19	631,433.36
		Memory	SIM-Module 16M x 32, 70 ns	808,663.10	808,663.10
			SIM-Module 4M x 36, 70 ns	348,434.93	337,138.13
			SIM-Module 8M x 32,	562,619.24	561,208.39

Product Hierarchy for material MARA Level 01	Product Hierarchy for material MARA Level 02	Product Hierarchy for material MARA Level 03	Product Hierarchy for material MARA Level 04	Sales Volume	Incoming Orders
			PS/2-72 Pin EDO-RAM		
			SIM-Module 8M x 36, 70 ns	373,794.70	373,794.70
		Monitor	Flatscreen LE 50 P	1,116,349.10	1,155,586.30
			Flatscreen LE 64 P	1,749,563.60	1,749,563.60
			Flatscreen MS 1460 P	2,525,626.85	2,704,385.95
			Flatscreen MS 1575 P	2,375,663.66	2,469,023.66
			Flatscreen MS 1585	3,365,928.24	3,487,309.32
			Flatscreen MS 1775 P	3,809,366.74	4,074,466.82
			Flatscreen MS 1785 P	4,722,113.10	4,722,113.10
			Jotachi SN4000	4,396,902.60	4,572,792.60
			Jotachi SN4500	2,473,317.57	2,647,587.95

Product Hierarchy for material MARA Level 01	Product Hierarchy for material MARA Level 02	Product Hierarchy for material MARA Level 03	Product Hierarchy for material MARA Level 04	Sales Volume	Incoming Orders
			Jotachi SN5000	2,949,233.80	2,949,233.80
			MAG DX 15F/Fe	2,437,207.85	2,521,555.85
			MAG DX 17F	2,567,506.69	2,655,601.89
			MAG PA/DX 175	2,976,662.21	3,181,471.75
			PAQ MONITOR, 17", Color	683,666.40	685,463.40
			PAQ MONITOR, 20", Color	1,371,234.90	1,371,234.90
			SEC Multisync XV 17	3,557,594.77	3,690,080.79
			SEC Multisync XV 15	2,961,072.00	2,961,072.00
			Sunny Extreme	3,215,185.42	3,339,353.42
			Sunny Tetral3	2,996,901.84	2,996,901.84

Product Hierarchy for material MARA Level 01	Product Hierarchy for material MARA Level 02	Product Hierarchy for material MARA Level 03	Product Hierarchy for material MARA Level 04	Sales Volume	Incoming Orders
			Sunny Xa1	2,830,258.78	3,026,866.52
		PC ensemble	Maxite c-R 3100 Personal Computer	3,616,021.08	3,600,521.08
			Maxite c-R 375 personal computer	194,520.00	194,520.00
		Processor	Motherboard 3100	.00	18,558.62
			Process or 100 MHz	401,608.95	401,608.95
			Process or 133 MHz	15,949.50	11,574.50
			Process or 166 MHz	1,083,067.20	1,016,409.60

Product Hierarchy for material MARA Level 01	Product Hierarchy for material MARA Level 02	Product Hierarchy for material MARA Level 03	Product Hierarchy for material MARA Level 04	Sales Volume	Incoming Orders
	Printer	Laser printer	High Speed Printer	.00	60,011.10
Lighting	Bulbs	Light Bulb 40 Watt clear 220/235V		15,351,358.00	15,351,358.00
		Light Bulb 40 Watt frosted 220/235V		5,432,920.00	5,256,060.00
		Light Bulb 40 Watt red 220/235V		2,976,965.00	3,197,425.00
		Light Bulb 40 Watt yellow 220/235V		3,032,064.00	2,917,008.00
		Light Bulb 60 Watt clear 220/235V		6,312,877.00	6,731,493.00
		Light Bulb 60 Watt frosted 220/235V		6,115,488.00	5,902,704.00
		Light Bulb		2,775,919.00	2,975,767.00

Product Hierarchy for material MARA Level 01	Product Hierarchy for material MARA Level 02	Product Hierarchy for material MARA Level 03	Product Hierarchy for material MARA Level 04	Sales Volume	Incoming Orders
		60 Watt red 220/235V			
		Light Bulb 60 Watt yellow 220/235V		2,958,303.00	2,850,624.00
		Light Bulb 80 Watt clear 220/235V		7,095,740.00	7,589,438.00
		Light Bulb 80 Watt frosted 220/235V		5,120,425.00	4,935,200.00
		Light Bulb 80 Watt red 220/235V		2,416,380.00	2,416,380.00
		Light Bulb 80 Watt yellow 220/235V		2,647,459.00	2,550,863.00
Machines	Pumps	Special pump	Pump GG IDESNORM 100-200	5,798,946.55	6,191,646.55
			Pump cast steel	6,721,540.00	6,962,580.00

Product Hierarchy for material MARA Level 01	Product Hierarchy for material MARA Level 02	Product Hierarchy for material MARA Level 03	Product Hierarchy for material MARA Level 04	Sales Volume	Incoming Orders
			IDESNORM 150-200		
			Pump cast steel IDESNORM 170-230	9,558,837.00	9,565,269.00
			Pump chrome-steel IDESNORM 150-200	475,602.00	495,702.00
			Pump standard IDESNORM 100-402	6,893,740.00	7,336,070.00
			pump CR IDESNORM 150-200 ATO	7,822,280.00	8,343,320.00
			pump sphere-cast IDESNORM 150-200	7,057,540.00	7,336,720.00
Paints	Gloss Paints	Opaque	Coating Matt Green RAL 6014/10 Liter	.00	125.03
Services	Maintenance	HiTech maintenance	PC Service (Configurable)	171,992.54	285,995.83

Product Hierarchy for material MARA Level 01	Product Hierarchy for material MARA Level 02	Product Hierarchy for material MARA Level 03	Product Hierarchy for material MARA Level 04	Sales Volume	Incoming Orders
		e	e)		
			PC Service Plus	119,570.00	199,320.00
Vehicles	Cars	Car (complete)	SAPSOTA FUN DRIVE 2000GT	155,640.00	155,640.00
	Motorcycles	Accessories	Motorcycle Helmet - Standard	3,651,535.18	3,800,706.24
		Components	Deluxe Gas Tank Striping Decals	658,897.52	703,373.20
			Deluxe Headlight	1,609,718.67	1,675,515.07
			Deluxe Taillight	546,588.47	579,120.40
		Motor-cycle (compl.)	CrossFun/350 cm3	10,842,211.75	11,243,347.19
			IDES Glad Boy configurable	929,800.00	953,830.00
			SunFun/1200 cm3	23,522,868.17	23,538,918.90

SAP ERP アダプタの使用

SAP アダプタの構成方法について説明します。

SAP アダプタを使用すると、ibi Data Migrator、WebFOCUS、およびその他の有効なクライアントアプリケーションで、SAP のデータソースにアクセスすることが可能になります。クライアントリクエストは、ネイティブの SAP ABAP ルーチンに動的に変換され、最適化されたアンサーセットがリクエストの送信元に返されます。サンプルのリクエストと出力については、[SAP リクエストの生成](#)を参照してください。

注意：「SAP R/3」という名称は、SAP によってリリース 4.6 まで使用されました。SAP で使用される他の名前の変種には、My SAP、SAP ECC、SAP S/4HANA があります。WebFOCUS アダプタでは、これらの変種のすべてに対応します。

SAP 環境の準備

Server では、SAP Application Server と通信を行うために、SAP の RFC SDK が必要になります。RFC SDK の場所は、プラットフォーム固有の検索パスに応じて異なります。

プラットフォームに依存する共有ライブラリ (RFC SDK) 用の検索パスは、以下のとおりです。

- **Windows**

環境変数の PATH。

- **AIX**

環境変数の LIBPATH。

- **他のすべての UNIX プラットフォーム**

環境変数の LD_LIBRARY_PATH。

サーバが開始せず、一部の NLS ライブラリが見つからないことを示す場合は、次の行を試みます。

```
EXPORT NLSUI_7BIT_FALLBACK=YES
```

SAP の GUI を使用して SAP にログオンし、環境を準備します。

開発者キーが管理者の SAP ユーザ ID が必要です。(システムにより開発者キーを要求するプロンプトが表示されます (このキーをそれまでに入力したことがない場合)。)

USER ID が許可のために必要になります。許可される権限は、ユーザの任務の性質に基づく権限の割り当てによって異なります。プロシジャの実行のみが必要なユーザには、基本的な権限が割り当てられます。管理者権限は、SAP アダプタの各種管理機能を実行する必要があるユーザに割り当てられます。たとえば、SAP アダプタでは、ABAP/4 プログラムを実行時に動的に生成します。これは、アダプタの構成時に (管理者権限を必要とするタスク)、SAP システムにアップロードされたアダプタコンポーネントを使用して実行されます。

関連情報については、処理オブジェクトに関するリストの [USER 用の権限オブジェクト](#)、[権限フィールド](#)、[および必須値のリスト](#) および [バックグラウンド処理用の権限オブジェクトのリスト](#) を参照してください。

USER 用の権限オブジェクト、権限フィールド、および必須値のリスト

以下の各表には、基本ユーザとアダプタの管理者にそれぞれ適用される、権限のすべての違いが反映されています。

S_RFC - RFC アクセスの権限チェック

項目	値
ACTVT (アクティビティ)	16 (実行)
RFC_NAME (保護する RFC の名前)	管理者の場合 AQRC、SDTX、SLST、SUTL、SYST、Z*** 基本ユーザの場合 SYST、Z***
RFC_TYPE (保護する RFC オブジェクトのタイプ)	FUGR (汎用グループ)

S_TCODEv - トランザクション開始の権限チェック

項目	値
TCD (トランザクションコード)	管理者の場合 S001、SE09、SE11、SE11_OLD、SE13、SE37、SE38、SE80、SU53 基本ユーザの場合 S001、SE37、SE38、SU53

S_ADMI_FCD - システム権限

項目	値
S_ADMI_FCD (システム管理機能)	管理者の場合 MEMO

S_C_FUNCT - ABAP プログラム内での C 呼び出し

項目	値
ACTVT (アクティビティ)	管理者の場合 16 (実行)
CFUNCNAME (呼び出し可能 C ルーチンの名前)	管理者の場合 SYSTEM
PROGRAM (ABAP プログラム名)	管理者の場合 SAPLSTRF、SAPLSTRI

S_DATASET - ファイルアクセスの権限

項目	値
ACTVT (アクティビティ)	管理者の場合 33、34、A6
FILENAME (物理ファイル名)	管理者の場合 *
PROGRAM (ABAP プログラム名)	管理者の場合 *

S_TABU_DIS - テーブル更新 (SM30 などの標準ツールを使用)

項目	値
ACTVT (アクティビティ)	管理者の場合 03
DICBERCLS (権限グループ)	管理者の場合 SS、&NC&

S_DEVELOP - ABAP ワークベンチ

項目	値
ACTVT (アクティビティ)	管理者の場合 01、02、03、07、16、40
ABAP ワークベンチ DEVCLASS - 移送システムの開発クラス	管理者の場合 *
OBJNAME - オブジェクト名	管理者の場合 *
OBJTYPE - オブジェクトタイプ	管理者の場合 DEVC、FUGR、PROG、TABL、TABT
P_GROUP - ABAP プログラムの権限グループ	管理者の場合 *

S_TRANSPORT - 移送オーガナイザ

項目	値
ACTVT (アクティビティ)	管理者の場合 01
TTYTYPE (リクエストタイプ (移送/修正システム))	管理者の場合 DTRA、TASK

バックグラウンド処理用の権限オブジェクトのリスト

項目	値
S_BTCH_ADM	N (ユーザにバックグラウンドジョブの管理権限が必要でない場合)
S_BTCH_JOB	RELE (リリース)

アダプタコンポーネント用の SAP 環境の準備

SAP アダプタでは、ABAP/4 プログラムを実行時に動的に生成します。これは、アダプタの構成時に SAP システムにアップロードされたアダプタコンポーネントを使用して実行されます。このコンポーネントのアップロードの前に、SAP システムを準備する必要があります。

SAP 環境をアダプタコンポーネント用に準備するには、以下の手順を実行します。

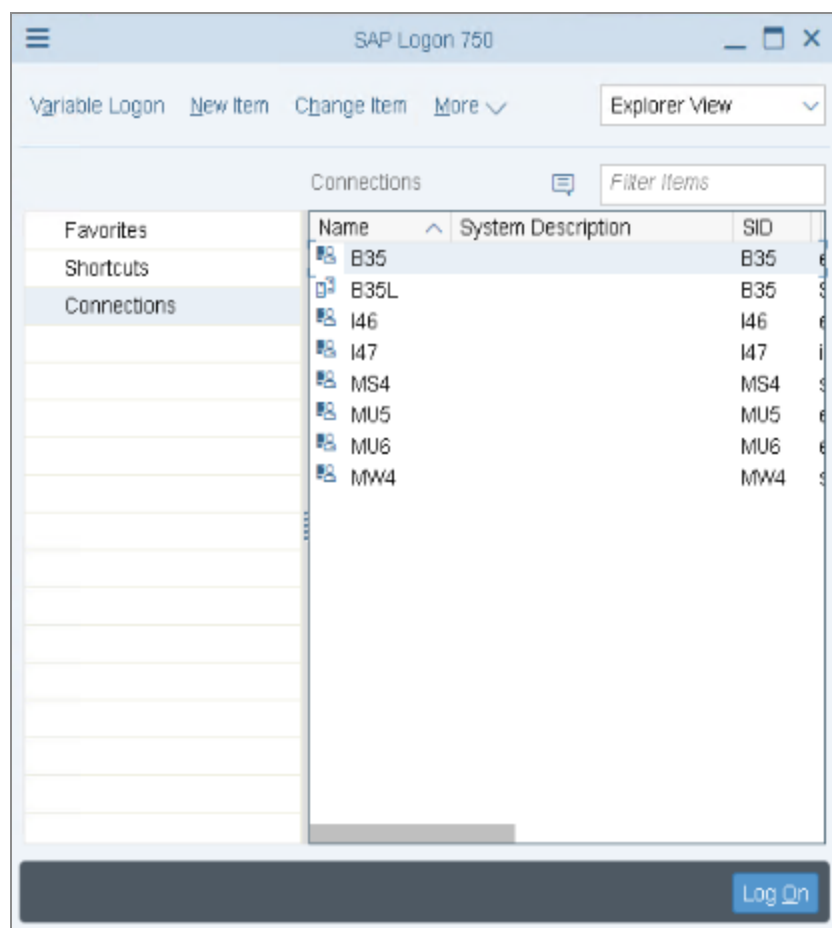
1. SAP にログインします。
2. 開発クラスまたはパッケージを作成します。
3. 汎用グループを作成します。
4. 汎用グループを有効にします。
5. 汎用グループを確認します。

SAP にログインするには

手順

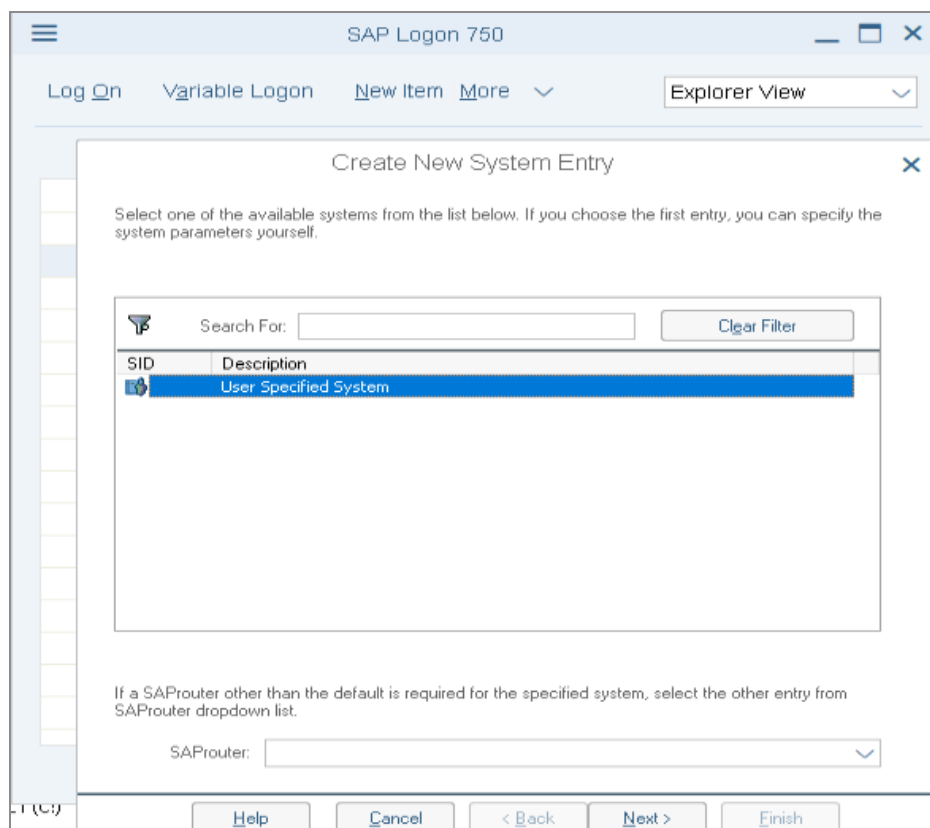
1. SAP ログインを起動します。標準の SAP GUI インストールでは、この実行可能ファイル

ルは ¥Program Files¥SAP¥FrontEnd¥SAPgui¥saplogon.exe にあります。[SAP ログオン] ダイアログボックスが表示されます。



リストが空の場合、新規エントリを作成します。

- [新規作成] をクリックします。[新規エントリ] ダイアログボックスが表示されます。



- b. [内容説明] と [アプリケーションサーバ] を入力し、[システム ID] と [インスタンス番号] を選択します。多くの場合、[SAP Louter スtring] は必要ありませんが、割り当てられている SAP の技術の連絡先にお問い合わせください。

[アプリケーションサーバ] テキストボックスに、SAP Application Server のネットワーク名または IP アドレスのいずれかを入力します。

[OK] をクリックして、[ログオン] のリスト上に新規エントリを作成します。

2. リスト上のエントリを選択して、[ログオン] をクリックします。

ユーザ ID とパスワードの入力を要求するログオンページが表示されます。システムにログオンします。はじめて使用するユーザ ID の場合、システムからパスワードを変更するよう要求されることに注意してください。

開発クラスまたはパッケージを作成するには

注意：リリース 4.7 を使用している場合は、開発クラスの代わりに、パッケージを作成します。

手順

1. トランザクションの SE80 を実行して、オブジェクトナビゲータを開始します。
2. SAP アダプタのインストール用の開発クラスまたはパッケージの名前 (たとえば ZIBI) を入力します。
注意：開発クラスおよびパッケージの名前の最大長は 14 バイトで、SAP のカスタマオブジェクトの命名規則に従う必要があります。
3. [照会]  をクリックします。
システムにより「開発クラスまたはパッケージ ZIBI がありません」という応答が返されます。
4. [はい] をクリックしてオブジェクトを作成します。
5. 開発クラスまたはパッケージの簡単な説明 (たとえば「IBI-Created」) を入力します。
6. 表示されたすべてのデフォルト値を受け入れて、[保存] をクリックします。
システムにより、移送/修正システム (CTS) の固有の依頼番号が生成されます。
7. この CTS の依頼番号を他の SAP システムへのアダプタのデプロイメントのために保存します。
8. [続行] をクリックします。

汎用グループを作成するには

手順

1. トランザクションの SE80 を実行して、オブジェクトナビゲータを開始します。
2. プルダウンメニューから、[汎用グループ] を選択します。
3. [開発クラスまたはパッケージを作成するには](#)で指定した、SAP アダプタのインストー

ル用の開発クラス名またはパッケージ名 (例、ZIBI) と同一の名前を入力します。

4. [照会] をクリックします。

システムにより「汎用グループ ZIBI がありません」という応答が返されます。

5. [はい] をクリックしてオブジェクトを作成します。

6. 汎用グループの説明を入力します。

7. 表示されたすべてのデフォルト値を受け入れて、[保存] をクリックします。

8. [オブジェクトディレクトリエントリ登録] の下で、開発クラスに割り当てられている文字列と同じ文字列を入力して、[保存] をクリックします。

9. [トランスポート可能なリクエストのプロンプト] に応じて、開発クラスまたはパッケージの作成時に割り当てられた CTS の依頼番号を入力します。

10. [続行] をクリックします。

Unicode チェックを無効にするには

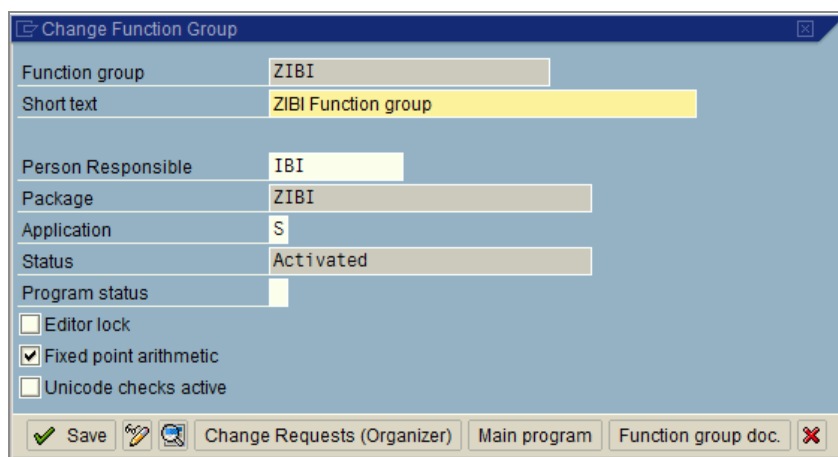
この手順は、SAP リリース 4.7 にのみ適用されます。

手順

1. トランザクションの SE80 を実行して、オブジェクトナビゲータを開始します。
2. プルダウンメニューから、[汎用グループ] を選択します。
3. 汎用グループの値に、作成済みの汎用グループの名前 (たとえば、ZIBI) を入力します。
4. [照会] をクリックします。

オブジェクトウィンドウに、汎用グループの名前を含む開いたフォルダおよび INCLUDE という名前の閉じたフォルダが表示されます。

5. 汎用グループ名 (たとえば、ZIBI) を右クリックして、ポップアップメニューから [変更] を選択します。



6. [ユニコードチェック有効] チェックボックスの選択を解除して、[保存] をクリックします。

汎用グループを有効にするには

手順

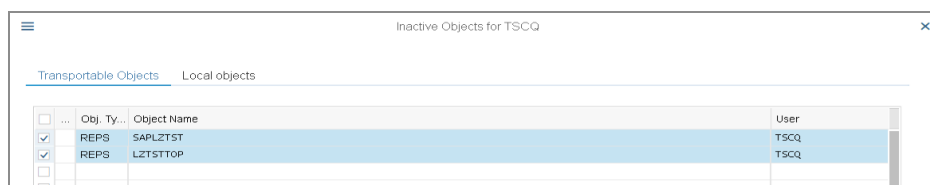
1. トランザクションの SE80 を実行して、オブジェクトナビゲータを開始します。
2. プルダウンメニューから、[汎用グループ] を選択します。
3. 汎用グループの値に、作成済みの汎用グループの名前 (たとえば、ZIBI) を入力します。
4. [照会] をクリックします。

オブジェクトウィンドウに、汎用グループの名前を含む開いたフォルダおよび INCLUDE という名前の閉じたフォルダが表示されます。

5. 汎用グループ名 (この例では、ZTST) を右クリックして、ポップアップメニューから [有効化] を選択します。

新たなウィンドウが開いて、無効なオブジェクトがすべてリストされます。オブジェクト名が SAPL で始まることに注意してください。

汎用グループのオブジェクト (この例では、ZTST) が選択されていることを確認します。



	...	Obj. Ty...	Object Name	User
☒	REPS	SAPLZTST		TSCQ
☒	REPS	LZTSTTOP		TSCQ
☐				
☐				

6. [続行] をクリックします。

汎用グループを確認するには

手順

1. トランザクションの SE80 を実行して、オブジェクトナビゲータを開始します。
2. プルダウンメニューから、[汎用グループ] を選択します。
3. 汎用グループの値に、作成済みの汎用グループの名前 (たとえば、ZIBI) を入力します。
4. [照会] をクリックします。
5. ZTST フォルダを右クリックして、[チェック - 拡張チェック] を選択します。
6. [チェック実行] を選択します。
次のようなレポートが表示されます。

SAP			
Display Results Display All Results Display Single Test			
Check for Program SAPLZTST	Error	Warnings	Messages
Test Environment	0	0	1
PERFORM/FORM Interfaces	0	0	0
CALL FUNCTION/METHOD Interfaces	0	0	0
External Program Interfaces	0	0	0
Authorizations	0	0	0
GUI Status and TITLEBAR	0	0	0
SET/GET Parameter IDs	0	0	0
MESSAGE Statements	0	0	0
Character Strings	0	0	0
Show CURR/QUAN Fields	0	0	0
Field Attributes	0	0	0
Superfluous Statements	0	0	0
Syntax Check Warnings	0	0	0
Check on Load Sizes	0	0	0
Internationalization	0	0	0
Problematic Statements	0	0	0
Structure Enhancements	0	0	0
Hidden Errors and Warnings	0	0	0

注意：この手順でエラーまたは警告が出力された場合、SAP アダプタのインストールを続行する前に、SAP レイヤでそれらの問題を解決します。

7. エラーの詳細情報については、エラーの行をダブルクリックします。

結果

これで、WebFOCUS Reporting Server ブラウザインターフェースから汎用モジュールをアップロードする次の手順のために、SAP 環境の準備が整いました。

ZXXX* 一時オブジェクトの移送/移動のカスタマイズ

SAP アダプタのインストール時に、汎用モジュール一式 (ZXXXBTCH と ZXXXREPTS の2つの一時オブジェクトを含む) が、SAP リポジトリにアップロードされて登録されます。QA から DEV や PROD の Application Server への移送に備えて、ZXXXBTCH オブジェクトを

ZXXX の開発クラスに割り当てる必要があります。この割り当てを行わないと、ZXXXBTCH は開発クラスの \$TMP に割り当てられるか、未割り当てのままになり、これらのいずれかが原因で移送が失敗する場合があります。

移送制御

SAP では、SAP システム間でのリリースのアップグレードと移送を処理するために、移送制御プログラム (tp) を提供しています。移送制御プログラムでは、以下のことが実行されます。

- 移送が追跡される。
- オブジェクトのエクスポートとインポートが正しい順序で実行される。
- 対象システムへのインポートが、ソースシステムからのエクスポートと必ず同じ順序で実行される。(誤った順序でインポートを処理すると、対象システムで重大な不整合が発生するおそれがあり、この場合診断は困難です。)
- エクスポートとインポートを別々に行うことが可能になる。

エクスポート時には、移送されるオブジェクトがソースシステムのデータベースから抽出され、オペレーティングシステムのファイル内に保存されます。

インポート時には、オブジェクトは (タスク内に記録された移送機能に従って) 対象システムのデータベースに追加されます。

注意：技術的な情報についての詳細は、<http://help.sap.com> を参照してください。

複数の SAP システムへのアクセス

アダプタでは、複数の SAP システムにわたる処理が可能です。SAP では、複数の企業からのデータで同じメタデータを共有でき、またこれらのデータはクライアントの値によって識別されます。通常、システムには複数のクライアントがあり、ユーザの権限に応じて 1 つまたは複数のクライアントにアクセスすることが可能です。アダプタでは、システムごとに、クライアント、ユーザ、およびパスワードから成る 1 つの SAP ログオンが必要です。このログオンには、以下の要件があります。

- RFC 対応であること。
- アクセスする可能性があるクライアントすべてで権限を付与されていること。

アクセスする SAP データに応じて、追加の権限が必要になる場合があります。詳細は、SAP 管理者に問い合わせてください。

SAP アダプタの構成

SAP アダプタの構成では、接続と認証の情報を指定します。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタに

についての説明を参照してください。

6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

SAP の接続属性

SAP アダプタは、[ERP] グループフォルダ下にあります。

下図のように、利用可能アダプタページで [SAP] を右クリックし、構成オプションのいずれかを選択します (例、SAP S/4Hana/NetWeaver 7.x Unicode)。

Configure NetWeaver 7.x Non Unicode
Configure SAP S4 Hana/NetWeaver 7.x Unicode
Configure ECC 5.0, 6.x Unicode
Configure 4.6C - ECC 5.0, 6.x Non Unicode

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

ユーザ認証パラメータ

アダプタの主な機能は、ユーザリクエストを SAP で理解できるコードに変換することです。このためには、特定の権限一式を持つ SAP ユーザ ID が必要になります。以下のリストでは、このユーザ ID は IBI_USER と記載されています。

System

12 バイト以内の接続名です。

パッケージパラメータ

FPOOL

SAP アダプタの静的な汎用モジュールをカタログ可能な汎用グループ。これは、[SAP 環境の準備](#)で作成した汎用グループです。

RANGEBEG

今後使用するために予約されています。

RANGEEND

今後使用するために予約されています。

接続パラメータ

[ロードバランシング] チェックボックスにより、以下のオプションのいずれが表示されるかが決まります。

GROUP

アプリケーショングループの名前です。アプリケーショングループでは、RFC アプリケーションを実行できる Application Server のリストを定義します。SAP トランザクション SMLG を使用して、アプリケーショングループを参照または変更することができます。

入力文字の大文字と小文字は区別され、またブランクは有意義であることに注意してください。

[ロードバランシング] のチェックがオンの場合にのみ表示されます。

MSGHOST

SAP システム (メッセージサーバ) のホスト名です。

[ロードバランシング] のチェックがオンの場合にのみ表示されます。

R3NAME

SAP システムのシステム ID です。

[ロードバランシング] のチェックがオンの場合にのみ表示されます。

HOST

SAP Application Server のホスト名です。

GWHOST

SAP ゲートウェイプロセスが稼働しているマシンのホスト名です。SAP Application Server が 1 つしかない場合、GWHOST と HOST は同じになります。

SYSNR

SAP システム番号です。この番号は 2 桁の数値です。この番号は SAP 管理者から入手します。

CONNECTION LANGUAGE

SAP ECC の Unicode 構成の場合、サーバにログオンする際に使用した言語とは異なる言語を選択することができます (NLS 構成で決定されます)。

[SAP Non Unicode の構成の追加] ウィンドウの [CONNECTION LANGUAGE] ドロップダウンリストから、SAP ECC への接続に使用する言語を選択します。

注意

- [CONNECTION LANGUAGE] オプションを使用して、サーバへのログインに使用する言語以外の言語を指定する場合、[多言語のシノニム生成] オプションは使用できません。
- このオプションが機能するためには、WebFOCUS Reporting Server のコードページが SAP ECC サーバのコードページと一致している必要があります。たとえば、WebFOCUS Reporting Server がコードページ 942 (SJIS) で構成されている場合、SAP サーバはコードページ 8000 で構成されている必要があります。
- 要求された接続言語を WebFOCUS Reporting Server のコードページ (表示またはバイト数に関して) または SAP ECC サーバのいずれかで使用できるかどうかを検査する方法はありません。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

一般ユーザログインパラメータ

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報

がデータベースへの接続時に認証情報として渡されます。

- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。

CLIENT

ユーザログイン用の SAP Client で最大 3 文字です。

USER

ユーザログイン用の SAP ユーザ ID です。

PASSWORD

ユーザログイン用の SAP パスワードで最大 8 文字です。

必要に応じて、以下のチェックボックスを選択します。

- **汎用モジュールのインストール** このオプションは、開発クラスと汎用グループの作成など、[SAP 環境の準備](#)の各手順が完了済みであることが前提となります。アダプタの汎用モジュールをインストールするには、このチェックボックスを選択します。汎用モジュールのインストールは、通常一度だけです。再インストールは、SAP システムが大幅に変更されたり、Full Function Server または WebFOCUS Reporting Server がアップグレードされた場合に限り必要になります。
 - **既存の汎用モジュールの上書き (同一の完全修飾名)** このオプションは、[汎用モジュールのインストール] を選択した場合に表示されます。新たにインストールした汎用モジュールで、名前が同じ既存のバージョンを置換する場合に使用します。
- **メタデータの初期化** SAP シノニムを作成する前に、SAP シノニムの候補の形で基本メタデータを提供する必要があります。[メタデータの初期化] オプションでは、基本メタデータが自動的に作成されます。メタデータの初期化は通常一度だけ必要です。メタデータの再初期化が必要になるのは、汎用モジュールを再インストールした場合のみです。

SNC セキュリティモード

[SNC] にチェックを付けた場合、以下の追加パラメータに値を指定する必要があります。

SNC_LIB

SNC_LIB には、外部セキュリティ製品のライブラリへのパスが含まれます。外部セキュリティ製品のライブラリ、外部ライブラリ、SNC_LIB、または gssapi ライブラリには、SAP が認定した外部セキュリティ製品によって提供される機能が含まれています。

環境変数 SNC_LIB を以下のようにセキュリティ製品のライブラリへのパスを含むように設定します。

```
<drive>:¥path¥to¥your¥snclib.dll
```

SNC_PARTNERNAME

SAP システムの外部名です。これは、SNC 名と呼ばれる外部名称の拡張バージョンです。SNC 名は、次のように入力される名称タイプを指定する外部ユーザ名を接頭辞に提供して作成します。

```
<SNC-name_of_SAP_AppServer>
```

以下はその例です。

```
p/secude:CN=miller, OU=ADMIN, O=SAP, C=DE
```

```
p/krb5:miller@WDF.SAP-AG.DE
```

SNC_QOP

保護レベルを指定します。

SNC_MYNAME

own_snc_name で定義された、イニシエータの SNC 名です。

SNC with load-balancing についての注意 ロードバランシングまたは「グループログオン」では、宛先の SNC 名をメッセージサーバから動的に取得します。[SNC with load balancing] を使用する場合、以下の追加パラメータを指定する必要があります。

```
SNC_PARTNERNAME=p:unused
```

```
SNC_LIB=<drive>:¥path¥to¥your¥snclib.dll
```

混合文字コードセット用のパラメータ

以下のセクションは、サーバプラットフォームと SAP インスタンスプラットフォームで、同じ文字コードセット (ASCII または EBCDIC) を使用していない場合にのみ適用されます。

SAP ECC または SAP S/4HANA アダプタを構成するには

手順

1. SAP ECC または SAP S/4HANA アダプタが構成済みでない場合は、[新規データソース] をクリックします。[ERP] グループを選択後、[SAP] を右クリックして SAP のバージョンを選択し、[構成] をクリックします。[SAP の構成の追加] ウィンドウが開きます。

ヒント：SAP ECC または SAP S/4HANA アダプタを構成済みで、ナビゲーションウィンドウのアダプタリスト内に表示されている場合は、現在のシステム接続を右クリックして [プロパティ] を選択することで、この構成をカスタマイズできます。[接続パラメータの編集] ウィンドウが開き、[構成の追加] ウィンドウに表示されるパラメータと同じパラメータが表示されます。

2. SAP ECC または SAP S/4HANA アダプタの接続を確立するには、次の構成パラメータに値を入力します。

SYSTEM

最大 12 文字の分かりやすい名前を入力して、SAP BW システムへの接続を指定します。

HOST

SAP BW システムのサーバのホスト名を入力します。

重要：この値は、GWHOST と SYSNR 用の値とともに、SAP BW の [System Entry Properties] の下にある SAP GUI 上の対応する値と一致している必要があります。

GWHOST

SAP のゲートウェイプロセスが実行されているサーバ (この場合、SAP BW システム) のホスト名を入力します。

SYSNR

SAP システム番号です。この番号は 2 桁の数値です。

3. [構成] をクリックします。「構成に接続を追加しました」というメッセージが表示されます。[次へ] をクリックすると、接続の構成が要求されます。
4. 次の接続属性を入力または選択し、[構成] をクリックします。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。

Client

ユーザログオン用の SAP BW Client です。

User

ユーザログオン用の認証 ID です。

Password

ユーザログオン用の SAP BW パスワードです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

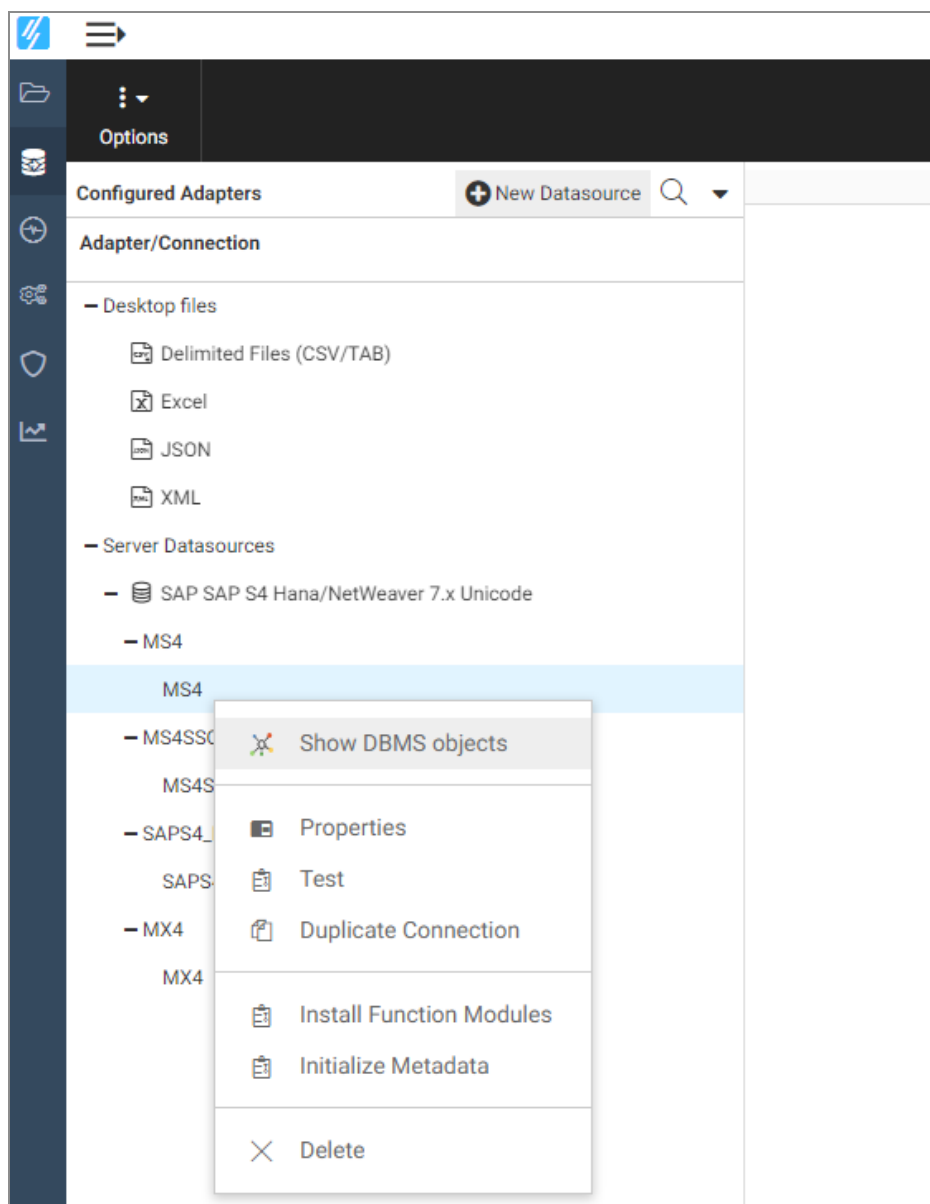
汎用モジュールがインストールされていない場合は、インストールが要求されます。

5. [次へ] をクリックします。

構成されたアダプタおよび接続が、アダプタリストに追加されます。

次に、[テスト] をクリックして接続を確認します。SAP ECC または SAP S/4HANA アダプタと SAP BW 環境との間の接続が確立されている場合、サンプルデータが右側のウィンドウ内に返されます。また、結果が良好な場合、必要な SAP BW マスターデータテーブルが SAP BW 環境内でアクセス可能になることも確認されます。

下図のように、接続を右クリックして [DBMS オブジェクトの表示] を選択することで、SAP テーブルのシノニムを作成することができます。



[利用可能なオブジェクト - SAP] ウィンドウが表示されます。シノニムを作成する TABLE オブジェクトを選択できます。

SAP アダプタを引き続き使用して、TABLE オブジェクトに対するシノニムの作成のための標準の手順に従います。詳細は、[テーブルシノニムの作成パラメータ](#)を参照してください。

接続範囲の制御

確立する接続ごとにアダプタを使用する場合、SET AUTODISCONNECT コマンドを使用して接続の継続を制御することができます。この設定を使用して、rfc api 通信レイヤをどの時点で終了するかを制御することができます。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

接続範囲の制御

```
ENGINE SQLSAP SET AUTODISCONNECT ON {FIN|COMMAND|COMMIT}
```

説明

SQLSAP

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMAND

各リクエストの後に自動的に切断されます。SET AUTODISCONNECT コマンドは、イベントがどの程度の頻度で発生するかによって、大幅なオーバーヘッドをもたらす可能性があります。このオーバーヘッドのほとんどすべてがサーバには関連しません。このオーバーヘッドは、オペレーティングシステムとデータソースに関連します。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

SAP ECC への COMMIT リクエストの送信の制御

COMMIT リクエストが SAP ECC に送信される際は、AUTOCOMMIT コマンドにより制御されます。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

SAP ECC への COMMIT リクエストの送信の制御

```
ENGINE SQLSAP SET AUTOCOMMIT ON {FIN|COMMAND}
```

説明

SQLSAP

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションの終了後にのみ、COMMIT リクエストを SAP に送信します。

COMMAND

コマンドを発行するたびに、COMMIT リクエストを SAP に送信します。COMMAND がデフォルト値です。

SAP コンポーネントのインストールと検査

SAP の接続属性を宣言した後、構成を完了させるために SAP コンポーネントをインストールして検査する必要があります。

SAP コンポーネントをインストールするには

SAP コンポーネントをインストールする手順は、以下のとおりです。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースを起動します。
2. ナビゲーションウィンドウ内の構成済みのアダプタのリストにアクセスします。
3. [SAP] フォルダを展開してから、ユーザ認証パラメータの設定時に作成したシステムをクリックします。詳細は、[SAP の接続属性](#)を参照してください。
4. メニューから [プロパティ] を選択します。
SAP 接続パラメータの編集画面が表示されます。画面の下部にコンポーネントのインストールボタンが表示されます。
5. 再インストールを行っている場合は、[既存の汎用グループを上書きする] のチェックをオンにします。汎用グループに関連する汎用モジュールとともに上書きしようとしている旨を通知する警告メッセージが表示されます。
エラーが発生しない場合、このボックスをチェックしないで手順 6 に進みます。
6. [コンポーネントのインストール] をクリックします。この処理には、数分かかることがあります。
インストールされた汎用モジュールのリストが表示されます。
7. インストールを確認するには、[汎用グループを確認するには](#)を参照してください。

SAP コンポーネントのインストールを確認するには

[SAP コンポーネントをインストールするには](#)の手順に従って必要な汎用モジュールをインストールした後、すべてのモジュールが正常にインストールされたことを確認するには、以下の手順を実行します。

手順

1. SAP で、トランザクション SE80 を実行します。
2. 最初のドロップダウンリストから [汎用モジュール] を選択します。
3. 接続属性を宣言した際に作成した汎用グループの名前を入力します。詳細は、[SAP の接続属性](#)の「パッケージのパラメータ」を参照してください。
4. テキストボックスの横の [表示] をクリックします。
汎用グループ名の下に、汎用モジュールおよび INCLUDE フォルダが表示されなければなりません。
5. Function Modules および Includes 各フォルダを展開して、以下のものが表示されていることを確認します。

```
ZIBI_REP_CREATE  
ZIBI_REP_GET_B_D  
ZIBI_REP_ABORT_B  
ZIBI_REP_DELETE  
ZIBI_REP_RUN  
ZIBI_TEST_REVERSE  
ZIBI_DYNAMIC_RUN  
ZIBI_GET_APPL_TREE  
ZIBI_LDB_GET  
ZIBI_SE80  
ZIBI_REP_RUN_IN_B  
ZIBI_REP_GET_B_S  
ZIBI_DXOBJ_INFO  
ZIBI_DD_FKEYS_GET
```

注意：Z*** は汎用グループの名前を表しています。開発クラスと汎用グループ名が ZIBI の場合、汎用モジュール名は ZIBI で始まります。

SAP 環境での構成後の作業

SAP アダプタのインストール時に、開発クラス (ZXXX) の汎用モジュール一式が SAP リポジトリに追加されます。

詳細は、[ZXXXREPTS 一時オブジェクトの移送/移動のカスタマイズ](#)を参照してください。

ZXXXREPTS 一時オブジェクトの移送/移動のカスタマイズ

SAP では、SAP システム間でのリリースのアップグレードと移送を制御するために、移送制御プログラム (tp) を提供しています。移送制御プログラムでは、以下のことが実行されます。

- 移送が追跡される。
- オブジェクトのエクスポートとインポートが正しい順序で実行される。
- 対象システムへのインポートが、ソースシステムからのエクスポートと必ず同じ順序で実行される。(誤った順序でインポートを処理すると、対象システムで重大な不整合が発生するおそれがあり、この場合診断は困難です。)
- エクスポートとインポートを別々に行うことが可能になる。

エクスポート時には、移送されるオブジェクトがソースシステムのデータベースから抽出され、オペレーティングシステムのファイル内に保存されます。

インポート時には、オブジェクトは (タスク内に記録された移送機能に従って) 対象システムのデータベースに追加されます。

ヒント：技術的な情報についての詳細は、<http://help.sap.com> (バージョン 4.6C) にアクセスし、メニューバーから、SAP Library、Basis Components、Change and Transport System (BC-CTS)、Transport Tools (BC-CTS-TLS)、Transport Control Program tp を順に選択します。このページで、役割に従って次の各セクションを参照してください。

- ユーザは、「Preparing Operating System Users」というタイトルのセクションに進んでください。
- システム管理者は、「Preparing the SAP System」というタイトルのセクションに進んでください。

移送用のデータファイル名

すべてのデータファイルは、移送ディレクトリの data に格納されます。データファイルの名前は、次に示した変更依頼の名前とコード文字から構成されます。このコード文字により、R3trans によって生成されたデータファイルとアプリケーション固有の開発環境オブジェクト (ADO) 用にアプリケーションプログラムによって生成されたデータファイルが区別されます。

- R <6 桁>.<ソースシステム> R3trans
- D <6 桁>.<ソースシステム> アプリケーションプログラム

変更依頼情報ファイル

変更依頼情報ファイルには、移送タイプ、移送するオブジェクトのクラスなど、変更依頼に関する情報が保存されます。また、変更依頼に必要な手順、終了コード、および実行時刻に関する情報も含まれています。

この情報ファイルは、ディレクトリの cofiles に格納されます。その名前は、変更依頼の名前を基に付けられます。

K <6 桁>.<ソースシステム>

依頼の移送

この例では、P46 に対するサンプルの依頼 K900022 を使用しています。この依頼は、依頼の移送のためのテンプレートとして使用できます。

以下のタスクを SAP システムごとに適切な順序で完了させます。

- 送信側システムから依頼をリリースします。
- /usr/sap/transport/cofiles へ移動し、K900022.P46 を対象システムの cofile ディレクトリへコピーします。
- /usr/sap/transport/data へ移動し、R900022.P46 を対象システムの data ディレクトリへコピーします。
- SAP システムの移送システムの状態を検査するには、次のコマンドを入力します。

```
tp checkimpdp <sapsid>
```

説明

sapsid

システム ID です。たとえば、P46 です (これは、SAP システムのインストール処理中に定義されます)。

- 次のコマンドを使用して、tp のインポートバッファの状態を検査します。

```
tp showbuffer <sapsid>
```

ヒント：この手順では、いくつかの tp コマンドを使用します。tp の手順とコマンドに関する詳細は、<http://help.sap.com> の SAP の tp の資料を参照してください。

- 次のコマンドを使用してバッファをクリアします。

```
tp cleanbuffer <sapsid>
```

- すべての項目がクリアされたことを確認するために、次のコマンドを使用して状態をチェックします。

```
tp showbuffer <sapsid>
```

- 次のコマンドを使用して移送を追加します。

```
tp add tobuffer P46K900022 <sapsid>
```

- 次のコマンドを使用して、移送がバッファに追加されたことを確認します。

```
tp showbuffer <sapsid>
```

- 次のコマンドを使用して移送をインポートします。

```
tp import P46K900022 <sapsid> u1
```

上書きのために他のオプションを追加することが必要になる場合があることに注意してください。詳細は、<http://help.sap.com> にある tp のオプションの資料を参照してください。

- システム画面に表示されたように、ZXXX 開発クラスがインポートされ、アクティブ化されていることを検査します。

SAP の予約済みの名前

SAP アダプタは、SAP Basis 4.6C バージョン (アダプタでの最低の認定レベル) との互換性があります。SAP アダプタでは、以下の名前が予約されています。

- 開発クラス ZXXX

- 汎用グループ ZXXX、および ZXXX* の汎用モジュールの名前

説明

XXX

クライアントに依存しないテーブルオブジェクトおよび汎用モジュールを一意に識別するために、認定済み SAP 管理者によって設定される 3 文字を表します。既存の固有の名前との競合が原因で、移送が失敗するおそれがあることに注意してください。

依頼の移送

この例では、P46 に対するサンプルの依頼 K900022 を使用しています。この依頼は、依頼の移送のためのテンプレートとして使用できます。

以下のタスクを SAP システムごとに適切な順序で完了させます。

- 送信側システムから依頼をリリースします。
- /usr/sap/transport/cofiles へ移動し、K900022.P46 を対象システムの cofile ディレクトリへコピーします。
- /usr/sap/transport/data へ移動し、R900022.P46 を対象システムの data ディレクトリへコピーします。
- SAP システムの移送システムの状態を検査するには、次のコマンドを入力します。

```
tp checkimpdp <sapsid>
```

説明

sapsid

システム ID です。たとえば、P46 です (これは、SAP システムのインストール処理中に定義されます)。

- 次のコマンドを使用して、tp のインポートバッファの状態を検査します。

```
tp showbuffer <sapsid>
```

ヒント：この手順では、いくつかの tp コマンドを使用します。tp の手順とコマンドに関する詳細は、<http://help.sap.com> の SAP の tp の資料を参照してください。

- 次のコマンドを使用してバッファをクリアします。

```
tp cleanbuffer <sapsid>
```

- すべての項目がクリアされたことを確認するために、次のコマンドを使用して状態をチェックします。

```
tp showbuffer <sapsid>
```

- 次のコマンドを使用して移送を追加します。

```
tp add tobuffer P46K900022 <sapsid>
```

- 次のコマンドを使用して、移送がバッファに追加されたことを確認します。

```
tp showbuffer <sapsid>
```

- 次のコマンドを使用して移送をインポートします。

```
tp import P46K900022 <sapsid> u1
```

上書きのために他のオプションを追加することが必要になる場合があることに注意してください。詳細は、<http://help.sap.com> にある tp のオプションの資料を参照してください。

- システム画面に表示されたように、ZXXX 開発クラスがインポートされ、アクティブ化されていることを検査します。

SAP メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとにシノニムを作成します。シノニムではデータソースの構造を定義して、サーバのデータタイプを SAP のデータタイプにマッピングします。

シノニムの作成

シノニムでは、サーバからアクセス可能な SAP データオブジェクトごとに固有の名前またはエイリアスを定義します。シノニムは、実際のデータソースの格納先や ID をクライアントア

アプリケーションから隠すことができるという点で便利です。また、シノニムは一時項目 (DEFINE) や他のセキュリティ機構などの、サーバの拡張メタデータ機能をサポートします。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

SAP シノニムのタイプ

SAP アダプタを使用すれば、以下のものに対してシノニムを作成することができます。

- **テーブル**
- **論理データベース (LDB) およびネイティブ LDB**
LDB の各タイプの使用を判断するための詳細は、[論理データベースの制限事項](#)を参照してください。
- **ビジネスアプリケーションプログラミングインターフェース (BAPI)**
要件および制限事項の詳細は、[汎用モジュールおよび BAPI の制限事項](#)を参照してください。
- **汎用モジュール**
要件および制限事項の詳細は、[汎用モジュールおよび BAPI の制限事項](#)を参照してください。
- **SAP クエリ**
SAP クエリとは、SAP のプログラミング言語 ABAP に関する知識のないユーザ向けに設計された、事前定義のレポートです。SAP クエリでは、ユーザグループで構成されるインフォセット (データソース) から、SAP システムにまだ含まれていない情報のリストが作成されます。このレポートには、クエリバリエントがグローバル領域内に含まれている場合があります。バリエントは、クエリまたはクエリによって生成されるレポートに対する、保存された選択基準のセットです。クエリを開始する際にバリエントを指定した場合、システムではクエリの選択基準としてバリエントの値を使用します。

シノニムを作成するには

シノニムを作成するには、事前にアダプタを構成しておく必要があります。シノニムは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースから作成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
 - **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
 - **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
 - **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
 - **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。
3. シノニムの作成パラメータの参照の箇所で説明しているように、アダプタで必要になるパラメータ用の値を入力します。
 - [テーブルシノニムの作成パラメータ](#)
 - [LDB シノニムの作成パラメータ](#)
 - [BAPI シノニムの作成パラメータ](#)
 - [汎用モジュールシノニムの作成パラメータ](#)
 - [QUERY シノニムの作成パラメータ](#)

4. パラメータ値を入力後、[シノニムの作成] をクリックします。

[ステータス] ウィンドウに、シノニムが正しく作成されたことを示す情報が表示されます。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

テーブルシノニムの作成パラメータ

以下は、SAP シノニムを作成する際に、値の指定が必要となるパラメータおよびそれに関連する作業の一覧です。これらのオプションは、さまざまなウィンドウで表示されます。ウィンドウに表示されたボタンをクリックして次のウィンドウに進み、最後に [シノニムの作成] ボタンをクリックします。これで、入力データに基づいてシノニムが作成されます。

オブジェクトタイプ

ドロップダウンリストから [TABLE] を選択します。

複数言語

シノニムは、1 つまたは複数の言語で作成することができます。複数の言語を選択した場合、選択したそれぞれの言語に対してタイトルと説明の属性を指定することができます。複数言語のタイトルおよび説明についての詳細は、[複数言語のタイトルと説明](#)を参照してください。

1. [複数言語] のチェックをオンにして、デフォルトのログイン言語 (サーバのスタートアップ時の SET LANGUAGE コマンドの値) 以外の追加言語を選択します。利用可能な言語の一覧が表示されます (デフォルトの言語は表示されません)。
2. 1 つまたは複数の追加言語を選択します。SAP では、WebFOCUS Reporting Server のログイン言語を使用して、すべての言語を取得します。

注意：選択した言語を実装するには、その言語をインストールしておく必要があります。また、NLS を有効にし、NLS 構成ウィザードで設定したコードページ設定と一致させる必要があります。WebFOCUS Reporting Server ブラウザインターフェースのメニューバーからこのウィザードにアクセスするには、[ワークスペース] ページで、[設定]、[LOCALE]、[構成ウィザード] を順に選択します。詳細は、[コードページおよび複数言語のシノニム](#)を参照してください。

次に表示されたウィンドウで、以下の手順を実行します。

- 「検索」のオプションとして、[アプリケーションツリー]、[開発クラス]、または [名前] のいずれかを選択します。なお、これらのオプションは、SAP がデータをグループ化し、選別する方法と類似しています。

次に表示される選択肢は、下で説明する [検索] オプションにより異なります。

アプリケーションツリー - このオプションを選択してから [次へ] をクリックして、アプリケーションのリストを取得します。

次に表示されたウィンドウで、リストからアプリケーションを 1 つ選択し、[次へ] をクリックします。

開発クラス - このオプションを選択して、入力ボックスに開発クラス名を入力するか、必要に応じてワイルドカードを使用して文字列を入力します。たとえば、「abc*」または「abc%」と入力して、abc という文字で始まる開発クラスを選択します。また、「*abc」または「%abc」と入力して、abc という文字で終わる開発クラスを選択します。さらに、「*abc*」または「%abc%」と入力して、文字列の任意の場所に abc という文字を含む開発クラスを選択します。

[次へ] をクリックして、開発クラスのオプションを表示します。

次に表示されたウィンドウで、開発クラスの 1 つを選択し、[次へ] をクリックします。

名前 - このオプションを選択して、入力ボックスにテーブルの名前を入力するか、必要に応じてワイルドカード (* または %) を使用して文字列を入力します。

[制限値] 入力ボックスに値を入力します。デフォルト値は、2000 テーブルです。利用可能なテーブルリストが大きすぎる場合は、制限値の入力を要求するメッセージが表示されます。

[次へ] をクリックします。

必要に応じて、次のオプションを選択します。

ウィンドウ下部のテーブルリストから、シノニムを作成するテーブルを選択します。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

シノニムオブジェクトの選択

[デフォルトシノニム名] の横に表示されたチェックをオンにして、リスト内のすべてのテーブルを対象にシノニムを生成するか、同一のカラムで特定の行のチェックをオンにして、選択したモジュールのシノニムを作成します。

複数言語のタイトルと説明

複数言語のシノニムを作成すると、そのシノニムには選択した各言語の TITLE および DESCRIPTION 属性が追加されます。たとえば、フランス語を選択した場合、シノニムには次のように TITLE および DESCRIPTION 属性が追加されます。

```
TITLE_FR='french title'  
DESC_FR='french description'
```

デフォルトのログイン言語は、TITLE および DESCRIPTION 属性を使用してそれぞれのタイトルと説明を生成します。

- TITLE および DESCRIPTION 属性を追加した言語に LANGUAGE パラメータを設定すると、タイトルと説明はそれぞれ TITLE_In と DESC_In 属性として生成されます (ここで、In は EN、DE、JA などの 2 文字の ISO コード)。
- タイトルと説明を追加していない言語に LANGUAGE パラメータを設定すると、TITLE および DESCRIPTION 属性を使用してタイトルと属性が生成されます。

シノニム言語のフィールド名を他と重複しない名前にするために、フィールド名はそのフィールドのユニーク名に基づいて作成されます。これらは技術名称と呼ばれます。デフォルト言語のみで作成されたシノニムでは、フィールドのキャプションに基づいてフィールド名が作成されます。

コードページおよび複数言語のシノニム

複数言語のシノニム用に選択した言語により、構成時に使用するコードページが決定されます。たとえば、ISO 8859-1 は、ほとんどの西ヨーロッパ言語に対応します。そのため、このコードページを使用すれば、英語、ドイツ語、フランス語、スペイン語で記述したリクエストを作成することができます。

Unicode UTF-8 のコードページ (65001) は、すべての主要言語をサポートするため、あらゆる言語の組み合わせに使用することができます。ただし、UTF-8 文字エンコードスキームは、ASCII 環境では 3 バイトを使用して文字を表示するため、より大きい文字データの格納領域が必要となります。

LDB シノニムの作成パラメータ

以下は、SAP シノニムを作成する際に、値の指定が必要となるパラメータおよびそれに関連する作業の一覧です。これらのオプションは、さまざまなウィンドウで表示されます。ウィンドウに表示されたボタンをクリックして次のウィンドウに進み、最後に [シノニムの作成] ボタンをクリックします。これで、入力データに基づいてシノニムが作成されます。

オブジェクトタイプ

ドロップダウンリストから [LDB] を選択します。

複数言語

シノニムは、1 つまたは複数の言語で作成することができます。複数の言語を選択した場合、選択したそれぞれの言語に対してタイトルと説明の属性を指定することができます。複数言語のタイトルおよび説明についての詳細は、[複数言語のタイトルと説明](#)を参照してください。

1. [複数言語] のチェックをオンにして、デフォルトのログイン言語 (サーバのスタートアップ時の SET LANGUAGE コマンドの値) 以外の追加言語を選択します。利用可能な言語の一覧が表示されます (デフォルトの言語は表示されません)。
2. 1 つまたは複数の追加言語を選択することができます。SAP では、WebFOCUS Reporting Server のログイン言語を使用して、すべての言語を取得します。

注意：選択した言語を実装するには、その言語をインストールしておく必要があります。また、NLS を有効にし、NLS 構成ウィザードで設定したコードページ設定と選択した言語を一致させる必要があります。WebFOCUS Reporting Server ブラウザインターフェースからこのウィザードにアクセスするには、[ワークスペース] ページのメニューバーから [設定]、[LOCALE]、[構成ウィザード] を順に選択します。詳細は、[コードページおよび複数言語のシノニム](#)を参照してください。

ネイティブまたはネイティブダイナミックで検索

次に表示されたウィンドウで、以下の手順を実行します。

- 「検索」のオプションとして、[アプリケーションツリー]、[開発クラス]、または [名前] のいずれかを選択します。「検索」のオプションとして、[アプリケーションツリー]、[開発クラス]、または [名前] のいずれかを選択します。なお、これらのオプションは、SAP がデータをグループ化し、選別する方法と類似しています。

次に表示される選択肢は、下で説明する [検索] オプションにより異なります。

- 取得するオブジェクトが、ネイティブ LDB またはネイティブダイナミック LDB と関連付けられているかどうかを指定します。詳細は、[論理データベースの制限事項](#)

を参照してください。

ネイティブモード 静的レポートがカタログに追加され、実行されます。

ネイティブダイナミックモード 特定の BAPI が呼び出されて、LDB クエリに結果が表示されます。

アプリケーションツリー - このオプションを選択して、[次へ] をクリックします。

次に表示されたウィンドウで、リストからアプリケーションを 1 つ選択し、[次へ] をクリックします。

開発クラス - このオプションを選択して、入力ボックスに開発クラス名を入力するか、必要に応じてワイルドカードを使用して文字列を入力します。たとえば、「abc*」（または「abc%」）と入力して、abc という文字で始まる開発クラスを選択します。また、「*abc」（または「%abc」）と入力して、abc という文字で終わる開発クラスを選択します。さらに、「*abc*」（または「%abc%」）と入力して、文字列の任意の場所に abc という文字を含む開発クラスを選択します。

[次へ] をクリックして、開発クラスを表示します。

次に表示されたウィンドウで、開発クラスの 1 つを選択し、[次へ] をクリックします。

名前 - このオプションを選択して、入力ボックスにテーブルの名前を入力するか、必要に応じてワイルドカード (* または %) を使用して文字列を入力します。

[制限値] 入力ボックスに値を入力します。デフォルト値は、2000 テーブルです。利用可能なテーブルリストが大きすぎる場合は、制限値の入力を要求するメッセージが表示されます。

必要に応じて、次のオプションを選択します。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスはブランクにしておきます。

ウィンドウ下部の LDB のリストから、シノニムを作成する LDB を選択します。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

シノニムオブジェクトの選択

シノニムを作成するオブジェクトを選択します。

- すべてのオブジェクトを選択するには、[デフォルトシノニム名] カラムの横に表示されたチェックをオンにします。
- 特定のオブジェクトを選択するには、そのオブジェクトのチェックをオンにします。

論理データベースの制限事項

SAP 論理データベース (LDB) 用のシノニムは、「SQL モード」(結合された SQL テーブルのセット) または「ネイティブモード」(基となる SAP 論理データベースプログラムを使用) という 2 つの異なる形式で生成することができます。

いずれのモードでも、現在 VAV、SDF、KDF、CEK、CFK、CIK、CPK、CRK の LDB しか認定されていないことに注意してください。その他のものについては、フィールド認定を必要とします。

- **SQL モード** JOIN の数またはシノニム内のフィールド数に関する標準の制限事項が適用されます。

エンドユーザの観点からは、通常 SQL モードの方がより高速に実行されるため、好ましい方法です。ネイティブモードは、たとえば SDF のセグメント SKC1A、SKC1C などのいわゆる「構造」にアクセスする要件がある場合にのみ使用してください。

- **ネイティブモード** ネイティブモードのほとんどの LDB では、実行時に選択画面が表示されます。この画面では、いくつかのパラメータが必須となります。SAP/R3-ECC アダプタでは、RFC を通じて処理を行うため、この画面に関する認識はありません。

このモードには、以下の 2 つのバリエーションがあります。

- ネイティブモードでは、静的レポートがカタログに追加され、実行されます。
- ネイティブダイナミックモードでは、特定の BAPI が呼び出されて、LDB クエリに結果が表示されます。

汎用モジュールシノニムの作成パラメータ

以下は、汎用モジュールのシノニムを作成する際に、値の指定が必要となるパラメータおよびそれに関連する作業の一覧です。[作成] ボタンをクリックして、入力した値に基づいてシノニムを生成します。

なお、すべての汎用モジュールが、レポート作成に適しているとは限りません。

オブジェクトタイプ

ドロップダウンリストから [汎用モジュール] を選択します。

複数言語

シノニムは、1 つまたは複数の言語で作成することができます。複数の言語を選択した場合、選択したそれぞれの言語に対してタイトルと説明の属性を指定することができます。複数言語のタイトルおよび説明についての詳細は、[複数言語のタイトルと説明](#)を参照してください。

1. [複数言語] のチェックをオンにして、デフォルトのログイン言語 (サーバのスタートアップ時の SET LANGUAGE コマンドの値) 以外の追加言語を選択します。利用可能な言語の一覧が表示されます (デフォルトの言語は表示されません)。
2. 1 つまたは複数の追加言語を選択することができます。SAP では、WebFOCUS Reporting Server のログイン言語を使用して、すべての言語を取得します。

注意：選択した言語を実装するには、その言語をインストールしておく必要があります。また、NLS を有効にし、NLS 構成ウィザードで設定したコードページ設定と選択した言語を一致させる必要があります。WebFOCUS Reporting Server ブラウザインターフェースからこのウィザードにアクセスするには、[ワークスペース] ページのメニューバーから [設定]、[LOCALE]、[構成ウィザード] を順に選択します。詳細は、[コードページおよび複数言語のシノニム](#)を参照してください。

検索

次に表示されるウィンドウで、「検索」のオプションとして、[アプリケーションツリー]、[開発クラス]、または [名前] のいずれかをします。なお、これらのオプションは、SAP がデータをグループ化し、選別する方法と類似しています。

次に表示される選択肢は、下で説明する [検索] オプションにより異なります。

アプリケーションツリー - このオプションを選択して、[次へ] をクリックします。

次に表示されたウィンドウで、リストからアプリケーションを 1 つ選択し、[次へ] をクリックします。

開発クラス - このオプションを選択して、入力ボックスに開発クラス名を入力するか、必要に応じてワイルドカードを使用して文字列を入力します。たとえば、「abc*」（または「abc%」）と入力して、abc という文字で始まる開発クラスを選択します。また、「*abc」（または「%abc」）と入力して、abc という文字で終わる開発クラスを選択します。さらに、「*abc*」（または「%abc%」）と入力して、文字列の任意の場所に abc という文字を含む開発クラスを選択します。

[次へ] をクリックして、開発クラスを表示します。

次に表示されたウィンドウで、開発クラスの 1 つを選択し、[次へ] をクリックします。

名前 - このオプションを選択して、入力ボックスにテーブルの名前を入力するか、必要に応じてワイルドカード (* または %) を使用して文字列を入力します。

[制限値] 入力ボックスに値を入力します。デフォルト値は、2000 テーブルです。利用可能なテーブルリストが大きすぎる場合は、制限値の入力を要求するメッセージが表示されます。

必要に応じて、次のオプションを選択します。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

ウィンドウ下部のテーブル (モジュール) のリストから、シノニムを作成するテーブルを選択します。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

シノニムオブジェクトの選択

[デフォルトシノニム名] の横に表示されたチェックをオンにして、リスト内のすべての汎用モジュールを対象にシノニムを作成するか、同一のカラムで特定の行のチェックをオンにして、選択したモジュールのシノニムを作成します。

BAPI シノニムの作成パラメータ

以下は、SAP シノニムを作成する際に、値の指定が必要となるパラメータおよびそれに関連する作業の一覧です。これらのオプションは、さまざまなウィンドウで表示されます。ウィンドウに表示されたボタンをクリックして次のウィンドウに進み、最後に [シノニムの作成] ボタンをクリックします。これで、入力データに基づいてシノニムが作成されます。

オブジェクトタイプ

ドロップダウンリストから [BAPI] を選択します。

複数言語

シノニムは、1 つまたは複数の言語で作成することができます。複数の言語を選択した場合、選択したそれぞれの言語に対してタイトルと説明の属性を指定することができます。複数言語のタイトルおよび説明についての詳細は、[複数言語のタイトルと説明](#)を参照してください。

1. [複数言語] のチェックをオンにして、デフォルトのログイン言語 (サーバのスタートアップ時の SET LANGUAGE コマンドの値) 以外の追加言語を選択します。利用可能な言語の一覧が表示されます (デフォルトの言語は表示されません)。
2. 1 つまたは複数の追加言語を選択することができます。SAP では、WebFOCUS Reporting Server のログイン言語を使用して、すべての言語を取得します。

注意：選択した言語を実装するには、その言語をインストールしておく必要があります。また、NLS を有効にし、NLS 構成ウィザードで設定したコードページ設定と選択した言語を一致させる必要があります。WebFOCUS Reporting Server ブラウザインターフェースからこのウィザードにアクセスするには、[ワークスペース] ページのメニューバーから [設定]、[LOCALE]、[構成ウィザード] を順に選択します。詳細は、[コードページおよび複数言語のシノニム](#)を参照してください。

検索

次に表示されるウィンドウで、「検索」のオプションとして、[アプリケーションツリー]、[開発クラス]、または [名前] のいずれかをします。なお、これらのオプションは、SAP がデータをグループ化し、選別する方法と類似しています。

次に表示される選択肢は、下で説明する [検索] オプションにより異なります。

アプリケーションツリー - このオプションを選択し、[次へ] をクリックして、BAPI を備えたアプリケーションのリストを取得します。

次に表示されたウィンドウで、リストからアプリケーションを 1 つ選択し、[次へ] をクリックします。

開発クラス - このオプションを選択して、入力ボックスに開発クラス名を入力するか、必要に応じてワイルドカードを使用して文字列を入力します。たとえば、「abc*」（または「abc%」）と入力して、abc という文字で始まる開発クラスを選択します。また、「*abc」（または「%abc」）と入力して、abc という文字で終わる開発クラスを選択します。さらに、「*abc*」（または「%abc%」）と入力して、文字列の任意の場所に abc という文字を含む開発クラスを選択します。

次に表示されたウィンドウで、開発クラスの 1 つを選択し、[次へ] をクリックします。

次に表示されたウィンドウで、オブジェクトタイプを 1 つ選択し、[次へ] をクリックします。

名前 - このオプションを選択して、入力ボックスに BAPI の名前を入力するか、必要に応じてワイルドカード (* または %) を使用して文字列を入力します。

[制限値] 入力ボックスに値を入力します。デフォルト値は、2000 テーブルです。利用可能な BAPI のリストが大きすぎる場合は、制限値の入力を要求するメッセージが表示されます。

次に表示されたウィンドウで、BAPI を選択し、[次へ] をクリックします。

必要に応じて、次のオプションを選択します。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスはブランクにしておきます。

ウィンドウ下部の BAPI のリストから、シノニムを作成する BAPI を選択します。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

シノニムオブジェクトの選択

シノニムを作成するオブジェクトを選択します。

- すべてのオブジェクトを選択するには、[デフォルトシノニム名] カラムの横に表示

されたチェックをオンにします。

- 特定のオブジェクトを選択するには、そのオブジェクトのチェックをオンにします。

汎用モジュールおよび BAPI の制限事項

- 汎用モジュールには、少なくともインポートまたはエクスポートパラメータが 1 つ必要になります。
- 「選択オプション」タイプの内部テーブルを使用してフィルタを行う場合、複数の条件を組み合わせて適用することができます。
- 「選択オプション」タイプではない内部テーブルによるフィルタはサポートされません。

QUERY シノニムの作成パラメータ

以下は、QUERY シノニムを作成する際に、値の指定が必要となるパラメータおよびそれに関連する作業の一覧です。これらのオプションは、さまざまなウィンドウで表示されます。ウィンドウ上のボタンをクリックして、ウィンドウ間を移動します。最後に [シノニムの作成] ボタンをクリックして、入力した情報に基づいてシノニムを作成します。

オブジェクトタイプ

ドロップダウンリストから [QUERY] を選択 します。

クロスクライアントクエリ

SAP クエリは、標準領域 (クライアント固有) とグローバル領域 (クロスクライアント) という 2 つの異なるクエリ領域で作成することができます。デフォルト設定では、クエリは標準領域で作成されるため、アクションは不要です。

クエリをグローバル領域で作成するには、[クロスクライアントクエリ] チェックボックスを選択します。

重要：グローバル領域 (クロスクライアント) で作成されたクエリに対して WebFOCUS リクエストを実行可能にするためには、次の選択基準をレポートリクエストに追加する必要があります。

```
WHERE WORKSPACE EQ 'X';
```

グループおよびクエリでフィルタ

特定のグループのクエリをリストするには、[フィルタ] チェックボックスをクリックし、表示された入力ボックスにグループ名またはクエリ名あるいはその両方を入力します。必要に応じてワイルドカード文字 (*) を使用します。

[候補の選択] ボタン

このボタンをクリックして、次のウィンドウに進みます。

注意：「クエリが見つかりません」というメッセージが表示された場合は、QUERY リストが生成されていない可能性があります。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスはブランクにしておきます。

シノニムオブジェクトの選択

シノニムを作成するオブジェクトを選択します。

- すべてのオブジェクトを選択するには、[デフォルトシノニム名] カラムの横に表示されたチェックをオンにします。
- 特定のオブジェクトを選択するには、そのオブジェクトのチェックをオンにします。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

SAP クエリのシノニムのバリエーション

SAP クエリのシノニムを作成する場合、すべてのバリエント (存在する場合) が CREATE SYNONYM の実行時に検索されます。2 セグメントのマスタファイルが生成されます。先頭のセグメントには、すべての有効なバリエント用の VARIANT フィールドが保持されます。2 番目のセグメントは、実際のレポートフィールドです。レポートを実行するには、バリエントを指定する必要があります。

たとえば、以下のシノニムを使用します。

```
$          $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
$              SNAPpack generated on Thu Feb 20 11:44:07 2003  
$              SAP instance:I46      SAP Release:46C  
$              MFD generated for QUERY:BASEAPP/ALM01_02  
$          $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
FILE=BASEAPP/ALM01_02 ,SUFFIX=SQLSAP,  
    REMARKS='UG: 02 Query:ALM01',$  
SEGNAME=VARIANT,SEGTYPE=S0,$  
    FIELD=VARIANT, ALIAS=VARIANT,ACTUAL=A14,USAGE=A14,ACCEPT=  
    'CAUS' OR  
    'MY VARIANT' OR  
    ', '$
```

```
SEGNAME=REPORT,PARENT=VARIANT,SEGTYPE=S0,$
  FIELD=SKA1_SAKNR, ALIAS=SKA1_SAKNR, USAGE=A010, ACTUAL=A010,
  TITLE='G/L acct', DESC='G/L account number',$
  FIELD=SKA1_GVTYP, ALIAS=SKA1_GVTYP, USAGE=A002, ACTUAL=A002,
  TITLE='AT', DESC='P&L statement account type',$
  FIELD=SKC1C_BUKRS, ALIAS=SKC1C_BUKRS, USAGE=A004, ACTUAL=A004,
  TITLE='CoCd', DESC='Company Code',$
  FIELD=SKC1C_GJAHR, ALIAS=SKC1C_GJAHR, USAGE=A004, ACTUAL=A004,
  TITLE='Year', DESC='Fiscal year',$
  FIELD=SKC1C_GSBER, ALIAS=SKC1C_GSBER, USAGE=A004, ACTUAL=A004,
  TITLE='BA', DESC='Business Area',$
  FIELD=SKC1C_HWAER, ALIAS=SKC1C_HWAER, USAGE=A005, ACTUAL=A005,
  TITLE='Crcy', DESC='Currency Key',$
  FIELD=SKC1C_UM010, ALIAS=SKC1C_UM010, USAGE=P16.2, ACTUAL=A16,
  TITLE='SKC1C-UM01', DESC='Monthly balance',$
  FIELD=SKC1C_HWAER1, ALIAS=SKC1C_HWAER1, USAGE=A005, ACTUAL=A005,
  TITLE='Crcy', DESC='Currency Key',$
```

バリエント 'CAUS' を使用した有効なリクエストは、以下のとおりです。

```
TABLE FILE ALM01_02
PRINT
  SKA1_SAKNR
  SKA1_GVTYP
  SKC1C_BUKRS
  SKC1C_GJAHR
  SKC1C_GSBER
  SKC1C_HWAER
  SKC1C_UM010
  SKC1C_HWAER1
IF VARIANT EQ 'CAUS'
IF READLIMIT EQ 50
END
```

SAP クエリのシノニムのバリエーション用のマスターファイル内で生成される ACCEPT の値リストは、オプションであることに注意してください。レポートを実行するにはバリエーションを指定する必要があります。なお、独自のバリエーションリストを保持することができます。

IDOC 用のシノニムの作成

以前のバージョン (5.3 から 7.1 まで) のサーバでは、IDOC シノニムのサポートが WebFOCUS Reporting Server ブラウザインターフェースに統合されていました。バージョン 7.6 では、IDOC シノニムの作成機能は WebFOCUS Reporting Server ブラウザインターフェースから削除されました。削除された機能には、EDA (電子データ交換) オプションおよび ALE (Application Link Enabling) オプションが含まれます。これらのオプションは、WebFOCUS Reporting Server ブラウザインターフェースでは使用できなくなりました。

IDOC シノニムの作成のサポートは、読み取り専用に限られるようになりました。IDOC 用のシノニムは、以下に説明する方法で作成することができます。ただし、アダプタでは IDOC を表す外部ファイルの読み取りが可能です。外部ファイルの生成、およびその外部ファイルに関連付けられる DBC 属性の提供は、ユーザの責任で行います。

IDOC シノニムを生成したい場合は、以下の例に示すようにコマンドをプロシジャ内に埋め込むことができます。

```
CREATE SYNONYM i47_idocs/MATMAS03_edioff FOR MATMAS03 IDOC EDIOFF
DBMS SQLSAP
AT I47
DROP
END
```

SAP テーブルのシノニムの作成

以下のシノニムは、TABLE ソース FDD02T から生成されました。指定したシステム名は 146 です。シノニムが作成されると、この値はアクセスファイルに反映されます。

生成されたマスターファイル DD02T.mas

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$          SNAPpack generated on Tue Nov 26 17:42:06 2002
$          SAP instance:I46      SAP Release:46C
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
FILE=DD02T, SUFFIX=SQLSAP
REMARKS='R/3 DD:SAP table texts',$

SEGNAME=DD02T, SEGTYPE=S0,$
FIELD=DD02T_TABNAME                                ,TABNAME ,
A30              ,A30              ,MISSING=OFF,
TITLE='Table',   DESC='Table name',$
FIELD=DD02T_DDLANGUAGE                              ,DDLLANGUAGE ,
A1               ,A1               ,MISSING=OFF,
TITLE='Lang.',   DESC='Language key',$
FIELD=DD02T_AS4LOCAL                               ,AS4LOCAL ,
A1               ,A1               ,MISSING=OFF,
TITLE='Activation', DESC='Activation status of a Repository object',$
FIELD=DD02T_AS4VERS                                 ,AS4VERS ,
A4               ,A4               ,MISSING=OFF,
TITLE='Version of', DESC='Version of the entry (not used)', $
FIELD=DD02T_DDTEXT                                  ,DDTEXT ,
A60              ,A60              ,MISSING=OFF,
TITLE='Short text', DESC='Short text describing R/3 Repository
objects',$

```

生成されたアクセスファイル DD02T.acx

```
SYSTEM=I46,$
SEGNAME=DD02T,
TABLENAME=DD02T, TABLETYPE=TRANSP,
MANDT=NO,
KEYS=4
,$
```

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

個別のテーブルに対する SAP テーブルクラス のサポート

WebFOCUS で特定のテーブルにアクセスする機能は、そのテーブルのクラスに基づいています。クラスは、この特定のテーブル用の DD02L 内の TABCLASS フィールドにあります。以下のテーブルクラスがサポートされます。

SAP テーブルタイプ	サーバのサポート
TRANSP	個別の SQL テーブル (SAP では「透過テーブル」と呼びます) ヘマッピングします。このテーブルは、通常インターフェースでアクセスおよび読み取りを行うことができます (例外 1 を参照)。
INTTAB	汎用モジュール、論理データベースなどで使用可能な構造 (SAP では「構造」と呼びます) を記述します。基になるデータはなく、インターフェースではサポートされません。
CLUSTER	これらのテーブル (SAP では「クラスタテーブル」と呼びます) が例外 1 を満たす限り、これらはサポートされます。
POOL	これらのテーブル (SAP では「プールテーブル」と呼びます) が例外 1 を満たす限り、これらはサポートされます。
VIEW	これらのテーブルが例外 1 を満たす限り、これらは

SAP テーブルタイプ	サーバのサポート
	サポートされます。また、すべての結合テーブルでサポートされるテーブルクラスを維持します。
APPEND	これらのテーブルはサポートされません。

また、サポートされるすべての TABLE クラスで、それらがサポートされるデータタイプを持つフィールドセットから構成されている必要があります。サポートされないデータタイプは、例外 2 にリストされています。

例外

1. あるテーブル内のあるフィールドが著しい長さの RAW タイプ (たとえば、テーブル INDX 内の CLUSTD フィールド) の場合、これは、このフィールドがあるプログラムを使用したインポートまたはエクスポートに使用される可能性があることを示している場合があります。基本構造に関する情報がないため、このフィールドは英数文字列として表示できるだけです。その対応する内容は、表示時に意味のあるものではない可能性があります。
2. サポートされていないデータタイプは、VARC、PREC、LCHR、LRAW、RAWSTRING です。

SAP 汎用モジュールのサポート

汎用モジュールを使用する前に考慮すべき属性が 2 つあります (汎用モジュールの [属性] タブに表示)。

- 汎用モジュールがリリースされていない場合 ([全般データ] に表示)、その汎用モジュールは SAP に内部的なものであり、ユーザは注意して使用する必要があります。この汎用モジュールは、リリースごとに SAP の予告なしに削除されたり、動作が変更されたりする場合があります。
- [処理タイプ] タブに表示される [通常の汎用モジュール] および [リモート可能汎用モジュール] 属性は、インターフェースに関しては技術的な相違はありません。ただし、汎用モジュールで RFC が有効化されている場合、SAP がその汎用モジュールを外部化するための手順が実行されたことが示される場合があります。

これらのインターフェースは、読み取り専用です。インターフェースから呼び出された汎用モジュールが読み取り専用であることを確認する必要があります。汎用モジュールが読み取

この確認は、ユーザに委ねられます。

システムセキュリティに対しても注意が必要です。汎用モジュールで RFC が有効化されている場合でも、SAP セキュリティにバインドされているとは限りません。ほとんどの場合、SAP セキュリティが含まれている可能性があるのは、名前が「BAPI」で始まる汎用モジュールのサブセットのみです。

1. シノニム作成時

汎用モジュールには、一連の IMPORT/EXPORT/CHANGING/テーブルの各パラメータが用意されている場合があります。また、一連の例外が含まれている場合もあります。現在の制限事項については、以下を参照してください。

各パラメータは、WebFOCUS サーバデータタイプに変換可能なデータタイプにプロトタイプ化する必要があります。パラメータをマッピングできない場合は、警告が発行され (FOC44492)、そのパラメータは拒否されます。READ_TEXT 汎用モジュールの LOCAL_CAT IMPORT パラメータがその一例です。

パラメータは、LIKE または TYPE としてプロトタイプ化することができます。TYPE の場合、このタイプは外部化する必要があります。つまり、汎用モジュールが定義されている汎用グループの範囲外で定義する必要があります。たとえば、SD_PARTNER_SELECTION 汎用モジュールの SELKZ_KUPAV テーブルパラメータでは、TYPE を見つけることはできません。このパラメータは、LV09A タイプグループで定義された、LV09A_TY_PRTNR_ITAB として入力されます。この場合、エラーメッセージ (FOC44488) が発行され、パラメータは拒否されます。

インターフェースは、単一フィールドまたは単一レベル構造を処理することができますが、複合構造を処理することはできません。たとえば、ACEDS_ACCRUAL_FOR_ACRTYPE_CALC 汎用モジュールの IT_COMP_SIMUPARAM_RANGES Import パラメータがこれに該当します。この場合、エラーメッセージ (FOC44491) が発行され、パラメータは拒否されます。

テーブルパラメータの構造に、SIGN、OPTION、contains(LOW)、contains(HIGH) という 4 つの連続的なフィールドのみが含まれている場合、この特別なテーブルは選択オプションテーブルとして表示され、アクセスファイルでもそのように宣言されます。単一値インポート以外では、これが汎用モジュールに一連の制約を渡す最も効率的な方法です。

2. 制限事項

- 現在、アダプタは CHANGING パラメータをサポートしません。
- データタイプのプロトタイプは、一意の ABAP 実データタイプになるよう定義する必要があります。そのため、ANY、ANY TABLE、STANDARD TABLE、TABLE は、データタイプのプロトタイプとしてサポートされません。
- TYPE C は、デフォルトの A1 に設定されます。

- TYPE STRING は、デフォルトの A80 に設定されます。これらの 2 つのタイプ (C および STRING) の場合、シノニムエディタを活用し、個々のニーズに応じて定義済みフィールドの長さを調整することは、ユーザの判断に委ねられます。
- データタイプは、サポートされているデータタイプにする必要があります。サポートされていないデータタイプは、VARC、PREC、LCHR、LRAW、RAWSTRING です。
- IMPORT パラメータは、単一フィールドにマッピングすることができます (LIKE または TYPE の使用)。
- EXPORT パラメータは、単一フィールドまたは構造のいずれかにすることができます (LIKE または TYPE の使用)。EXPORT パラメータが、TYPE を使用して定義された構造の場合、次の 2 通りのケースがあります。
 - a. パラメータが、データディレクトリ内で「構造」として参照されている (例、BAPIRET2)
 - b. パラメータが、「データタイプ」として参照されている (例、BAPIRET2_T)
- シノニムは、すべての場合で作成することはできますが、EXPORT パラメータとしてのテーブルタイプはサポートされません。

3. エラー/警告メッセージ

- (E) (FOC44488) データタイプ (data_type) は存在しません - パラメータ (parameter_name)
- (E) (FOC44489) DDIF_FIELDINFO_GET 内部エラー - (parameter_name)
- (E) (FOC44490) DDIF_FIELDINFO_GET 文書化されていないエラー - (parameter_name)
- (E) (FOC44491) ネスト化構造はサポートされません。パラメータ - (parameter_name)
- (E) (FOC44492) パラメータ (parameter_name) にデータタイプがありません。
- (W) (FOC44493) 選択オプション (table_name (parameter_name)) は、内部テーブルにマッピングされます。

エラーの発生時、エラーを発生したパラメータがオプションの場合、そのパラメータを除外してシノニムが作成されます。そのパラメータが必須の場合、シノニムの作成は中断されます。

注意：特定のパラメータのプロパティを取得するには、DDID_FIELDINFO_GET 汎用モジュールを使用します。この汎用モジュールがパラメータのプロパティを取得できな

い場合、そのパラメータには (FOC44488) のフラグが付けられます。

SAP 汎用モジュールが生成した ABAP 例外は、TABLE 実行時に記録されるようになりました。

結果のエラーは、次のようになります。

```
(FOC1695) SAP/R3 REQUEST ERROR      :
      [optional error message sent by the function module]
```

```
(FOC1736) SAP/R3 ERROR EXECUTING ABAP4 PROGRAM : function_module_name
```

SAP データタイプサポート

以下の表は、SAP データタイプのリスト、およびマスターファイルのデータタイプにマッピングされる方法の注記です。

SAP のデータタイプ	SAP の長さ	SAP の小数	USAGE	ACTUAL
ACCP	N	0	An	An
CHAR	N	0	An	An
CLNT	3	0	A3	A3
CUKY	N	0	An	An
CURR	n	p	P(n+2).p	P(n+1)/2
DATS * 詳細は、図の後の注を参照	8	0	YYMD または DMY	DATE

SAP のデータタイプ	SAP の長さ	SAP の小数	USAGE	ACTUAL
DEC	n	p	P(n+2).p	P(n+1)/2
FLTP ** 詳細は、 図の後の注を 参照	n	p	Dmm.eeE	D8
INT1	3	0	I3 (127 に限定)	I1
INT2	5	0	I4	I2
INT3	5	0	I4	I3
INT4	10	0	I10	I4
LANG	1	0	A1	A1
LCHR	N	n	サポートされて いません	
LRAW	N	なし	サポートされて いません	
NUMC	N	0	An	An
PREC	N	なし	サポートされて いません	
QUAN	N	P	P(n+2).p	P(n+1)/2
RAW	N	なし	A2n	A2n
STRING *** 詳細は、 図の後の注を	最大長 2GB	なし	A80	A80

SAP のデータタイプ	SAP の長さ	SAP の小数	USAGE	ACTUAL
参照				
TIMS	6	0	A6	A6 (00:00:00)
VARC	N	0	An (<=255)	An (<=255)
UNIT	3	0	A3	A3

注意

- *** 日付の用法 (DATS)** は、一般的なサーバ構成で提供された国の値に基づいて動的に決定されます。このため、日付の表記は国によって異なります。
- ****FLTP** この値は、仮数部と指数部を持つ浮動小数点数として表されます。

Dmm.eeE

説明

mm (仮数部)

ee + 6 です。6 の追加は、「仮数部の符号 + 仮数の桁数 + 文字 'D' + 指数部の符号 + 2 桁の指数部」という文字列の長さです。

ee (指数部)

SAP の小数部の桁数 - 1 (SAP では小数点を桁数に含めます)

たとえば、テーブル VBAP 内のフィールド UMREF は、次のように表される 16 桁の浮動小数点数です。

ee = 16 - 1 = 15。mm = 15 + 6 = 21。

FIELDNAME=VBAP_UMREF, ALIAS=UMREF, USAGE=D21.15E, ACTUAL=D8

*** **STRING** では可変長文字の長さの順序を定義します。STRING の長さの調整が必要な場合、この調整は WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのシノニムエディタを使用して行うことができます。

SAP Open/SQL のサポート

以下の表は、FOCUS ステートメントから SAP の Open/SQL ステートメントへの変換の説明です。

FOCUS コマンド	対応する Open/SQL ステートメント
READLIMIT	UP TO <i>n</i> ROWS 複数の SELECT ステートメントを生成した場合、READLIMIT は個々の SELECT ごとに UP TO <i>n</i> ROWS に変換されます。
WHERE	通常 WHERE ステートメントに変換されます。
JOIN TO	JOIN コマンドに含まれるテーブルごとに、埋め込みの SELECT ステートメントが作成されます。
JOIN	SELECT ... SELECT SINGLE ... に変換されます。
NULL	SAP アダプタでは、WHERE 句の一部に予約語の 'NULL' を使用することはサポートされません。WHERE 句の一部に 'NULL' が含まれている場合、構文エラーが発生します。
SUM ... BY	デフォルト設定では、集計機能は無効になっています。集計機能が無効な場合、FOCUS によって個々の行が選択され、集計およびソートが実行されます。集計機能は、以下のコマンドを発行することにより有効にできます。 <pre>ENGINE SQLSAP SET OPTIMIZATION SQL</pre> 集計機能を有効にすると、SUM は SUM に変換され、BY は GROUP BY および ORDER BY に変換されます。 集計機能は、TRANSPARENT テーブルに対してのみ適用されます。

SAP の高度な機能

このトピックでは、セキュリティ、BAPI、結合、およびトレースの使用方法について説明します。

ユーザセキュリティのサポート

SAP 用の Native Interface が強化され、以下の構文でユーザ ID とパスワードを処理します。

```
ENGINE SQLSAP SET CONNECTION_ATTRIBUTES system/user,password:'client'
```

この機能を使用すれば、以下のものを含むセキュリティで保護されたリクエストを実行することができます。

- 1 つまたは複数の BAPI。BAPI は SAP によってセキュリティで保護されています。
- SAP が提供した論理データベース。サーバのコードは SAP によってセキュリティで保護されています。
- 適切な AUTHORITY-CHECK ステートメントなどの完全にプロトタイプ化された汎用モジュール (SAP またはカスタマの)。

カスタムセキュリティイグジットの使用

WebFOCUS クエリに関連して ABAP コードの悪意のあるインジェクション攻撃を回避するために、アダプタに同梱される汎用モジュール (DYNAMIC_RUN、REP_CREATE、REP_GET_B_D、REP_RUN) が強化され、デフォルト設定で、正規の許可されたコマンドから動的に生成される ABAP4 コードのホワイトリストが実行されます。許可されたコマンドのリストの SEC_CHK 汎用モジュールを参照してください。ホワイトリストに登録されていないコマンドはすべて、実行が拒否されます。

これらの汎用モジュールは、開始時 (USR_EXT_ENT*) および終了時 (USR_EXT_EXT*) にユーザ定義の汎用モジュールへの呼び出しを挿入するためにも強化されています。最初に、呼び出しは、アダプタに同梱される汎用モジュールのダミー版に挿入されます。ユーザは、これらのダミー版を変更して、サイトで要求される必要な追加項目および権限 (SAP 権限チェック) をすべて実装することができます。これらのユーザ EXIT は、呼び出し元の汎用モジュールと同一のパラメータセットを使用して呼び出されます (IMPORT/TABLES のみ)。

サンプルユーザ権限チェック

```
function /IBI/ZIBI_USR_EXT_ENT.
*"-----
*"**"Local Interface:
*" IMPORTING
*" VALUE(REPID) LIKE TRDIR-NAME OPTIONAL
*" VALUE(REPORT_NAME_PREFIX) LIKE RS38L-AREA OPTIONAL
*" VALUE(FUNCPool) LIKE RS38L-AREA OPTIONAL
*" VALUE(LDBNAME) LIKE TRDIR-LDBNAME OPTIONAL
*" VALUE(INTERFACE_MODE) LIKE DD03L-INTTYPE OPTIONAL
*" VALUE(JOB_COUNT) LIKE TRDIR-NAME OPTIONAL
*" VALUE(REPID_BD) LIKE RS38L-AREA OPTIONAL
*" TABLES
*" ITAB STRUCTURE CHAR8000 OPTIONAL
*" PROGRAM STRUCTURE ABAPXT255 OPTIONAL
*" EXCEPTIONS
*" GENERATION_ERROR
*"-----

* -----
* delete a catalogued report
* -----
*{ INSERT M6DK900965 1
*
AUTHORITY-CHECK OBJECT 'S_CARRID'
ID 'CARRID' FIELD 'AA'
ID 'ACTVT' FIELD '03'.
IF sy-subrc = 4.
MESSAGE e045(sabapdocu) WITH 'AA'.
ELSEIF sy-subrc <> 0.
MESSAGE e184(sabapdocu) WITH text-010.
ENDIF.

*} INSERT
* this is now a dummied function module
ENDFUNCTION.
```

シノニム作成時のサンプルイグジット認可 チェック

次のサンプルイグジット `META_USR_SEC` は、シノニム作成の実行時に呼び出されます。アダプタでサポートされるすべての SAP データソースタイプ (例、TABLE、BAPI、RFC FUNCTION、LDB、QUERY) について、イグジットは対応する LOCAL INTERFACE 構造にアクセスすることができます。この場合、LOCAL INTERFACE 構造は、シノニム作成候補のリストから挿入されます。次のプロトタイプコードは、現在のユーザ SY-UNAME についての AUTHORITY-CHECK の例を示しています。この認可には、`Z_table_name` というオブジェクト名および `table_name` の値が含まれます。この場合の `table_name` は 'curval' の値、つまり A_TABNAME の ID 認可フィールドを含む TABS 構造から解析されたシノニム作成リテラルです。このダミー版は、実装前にサイトの適切な認可コールでカスタマイズする必要があります。

```
FUNCTION /IBI/ZIBI_META_USR_SEC.
*"-----
*"**"Local interface:
*"  TABLES
*"      NODES STRUCTURE  SNODETEXT OPTIONAL
*"      TABS STRUCTURE   DD02VV OPTIONAL
*"      BAPIS STRUCTURE  SWOTFIND OPTIONAL
*"      LDBS STRUCTURE   LDBT OPTIONAL
*"      FUGRS STRUCTURE  TLIBT OPTIONAL
*"      FUNCTIONS STRUCTURE  TFTIT OPTIONAL
*"      DEVCLASSES STRUCTURE  TDEVCVT OPTIONAL
*"      QUERY_FIELDS STRUCTURE  DFIES OPTIONAL
*"-----
data: curval(30) type C.
data: auth_object_name(80) type C.

*"-----
* Insert needed authorization checks here
*"-----
CLEAR curval.
LOOP AT TABS INTO curval.
*-----
* Insert authorization check for the table name in curval here
* -----
CLEAR auth_object_name.
CONCATENATE 'Z_' curval INTO auth_object_name.
AUTHORITY-CHECK OBJECT auth_object_name FOR USER SY-UNAME
```



```

        ID 'A_TABNAME' FIELD curval.

        IF SY-SUBRC <> 0.
            RAISE GENERATION_ERROR.
        ENDIF.
    ENDLOOP.
ENDFUNCTION.

```

BAPI のサポート

このリリースでは、以下の例のように JOIN を含むほとんどの読み取り専用の BAPI がサポートされます。

```

CREATE SYNONYM baseapp/BUS0002_GETLIST
FOR BUS0002/GETLIST
BAPI DBMS SQLSAP AT I46
END
CREATE SYNONYM baseapp/BUS0002_GETDETAIL
FOR BUS0002/GETDETAIL BAPI
DBMS SQLSAP AT I46
END
JOIN BAPI0002_COMP_CODE IN COMPANYCODE_GETLIST TO
CCGD2_COMP_CODE IN OMPANYCODE_GETDETAIL
END
TABLE FILE COMPANYCODE_GETLIST
PRINT
CCGL0_TYPE NOPRINT
BAPI0002_COMP_CODE
BAPI0002_COMP_NAME
CCGD2_CURRENCY
CCGD2_LANGU
IF BAPI0002_COMP_CODE NE '2300' OR '6000'
END

```

結合のサポート

Native Interface では、SAP との間のすべての結合がサポートされます。パフォーマンス上の理由から、SAP 以外のデータソースから SAP のデータソースに対する結合はお勧めしません。順次ファイルにキーを保持して、以下のコードを使用する方がより効率的です。

TABLE FILE SAP PRINT FIELDS IF KEYS IS (HOLD) END

レポートの処理モードの設定

SAP では、多数の各種処理 (たとえば、バッチ、ダイアログ (オンライン)、動的、印刷など) がサポートされます。WebFOCUS for SAP の各クエリでは、SAP Application Server 内に処理が生成されます。この処理は、ダイアログ (オンライン) またはバッチ処理として実行できます。

これらの処理モードオプションにより、SAP Application Server の構成に適合させることが可能です。この構成には、各 SAP Application Server の処理タイプごとに割り当てられた特定の数の処理が存在します。

モードの設定は、WebFOCUS Reporting Server ブラウザインターフェースから変更できます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

ダイアログおよびバッチ実行モードの設定

実行処理は、ダイアログ (オンライン) モードまたはバッチモードに設定することができます。

- **ダイアログ (またはオンライン) 処理** SAP アダプタのデフォルトのモードです。SAP のダイアログ処理は、より高い優先順位で迅速に実行する必要があるタスクに適しています。SAP のダイアログ処理には、アイドルタイムアウトの制限が適用されます (デフォルト値は 15 分です)。このデフォルト値が変更されているかどうかを確認するには、SAP Basis のサポート担当者にお問い合わせください。ダイアログ処理モードは、処理時間がアイドルタイムアウト制限値を下回るレポートに設定する必要があります。レポートの処理時間がこの制限値を超えた場合、SAP Application Server ではこの処理を終了して、レポートの実行は失敗します。

このモードでは、レポートはオンラインで実行されます。セッションは、実行が完了するまでブロックされます。実行時間は、SAP システムで事前設定された制限値に従う必要があります。

次のコマンドでは、SAP アダプタをダイアログ処理モードに設定します。

```
ENGINE SQLSAP SET EXECUTIONMODE ONLINE
```

バッチ実行モードを明示的に設定していて、サーバ上の単一のセッション内でダイアログモードに切り替えたい場合は、このコマンドを使用します。

処理時間の長い WebFOCUS for SAP レポートが存在する場合は、バッチ処理の使用を検討します。

- **バッチ処理** SAP アダプタの代替モードです。SAP のバッチ処理は、優先度の低いバックグラウンドで実行可能なタスクに適しています。このモードは、処理時間が比較的長いレポート (たとえば、複数のテーブルの結合が含まれたレポートや、大量のデータを処理する必要がある月末レポートなど) に対するものです。このモードでは、レポートが実行時間の制限によってタイムアウトとなることはありません。

次のコマンドでは、SAP アダプタをバッチ処理モードに設定します。

```
ENGINE SQLSAP SET EXECUTIONMODE BATCH
```

これらの設定を WebFOCUS Reporting Server ブラウザインターフェースから行うには、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。[実行モード] ドロップダウンメニューから [オンライン] または [バッチ] を選択します。

これらのコマンドは、サーバプロファイル (たとえば、edasprof.prf) に直接挿入したり、レポートプロシジャ内に含めることができることに注意してください。

ポーリング間隔のバッチ実行への設定

リクエストをバッチ実行に対して発行する前に、エンジンでこれらのリクエストの完了を検査する頻度を定義することができます。

構文は次のとおりです。

```
ENGINE SQLSAP SET NONBLOCK n
```

説明

n

秒数で定義されるポーリング間隔です。デフォルト値は 30 秒です。

この設定を WebFOCUS Reporting Server ブラウザインターフェースから行うには、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、メ

ニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。
NONBLOCK フィールドに値を入力します。

このコマンドは、サーバプロファイル (たとえば、edasprof.prf) に直接挿入したり、レポートプロシジャ内に含めることができることに注意してください。

SAP リクエストの生成

以下のリクエストと出力は、標準の ANSI SQL コードを使用した SAP アダプタの機能の例です。これらの例では、WebFOCUS Reporting Server ブラウザインターフェースから関連するテーブルに対して CREATE SYNONYM コマンドを実行しておく必要があります。
WebFOCUS Reporting Server ブラウザインターフェースを使用して、関連するテーブルを検索するか、名前による単純な検索を実行して、SAP アプリケーションの階層を移動することができます。

すべてのレコードの取得

以下の構文では、T014 からすべてのレコードが取得されます。

```
SELECT * FROM T014;
```

この SQL ステートメントに相当する WebFOCUS の構文は、以下のとおりです。

```
TABLE FILE T014
PRINT *
END
```

出力結果は次のとおりです。

Client	信用領域	通貨	更新	FYI バリ ア ン ト	リス ク の カ テ ゴ リ	信用 制限 値	販売グ ループ	すべての コード
800	0001	EUR	000012			.00		

Client	信用領域	通貨	更新	FYI バリ ア ン ト	リス ク の カ テ ゴ リ	信用 制 限 値	販売グ ループ	すべての コード
800	1000	EUR	000012	K4		.00		
800	3000	USD	000012	K4		.00		
800	5000	JPY	000012	K4		.00		
800	6000	MXN	000012	K4		.00		
800	R100	EUR	000012	K4		.00		
800	R300	USD	000012	K4		.00		

選択したフィールドの取得

以下の構文では、MARA から選択したフィールドが取得されます。

```
SELECT MARA_MATNR, MARA_MTART, MARA_MATKL, MARA_WRKST, MARA_BISMT FROM
MARA WHERE MARA_MATKL='999';
```

この SQL ステートメントに相当する WebFOCUS の構文は、以下のとおりです。

```
TABLE FILE MARA
PRINT MARA_MATNR MARA_MTART MARA_MATKL MARA_WRKST MARA_BISMT
WHERE MARA_MATKL EQ '999';
END
```

出力結果は次のとおりです。

品目	品目タイプ	品目グループ	基本品目	製品番号
0000000000000000038	HALB	999		
AM3-GT	KMAT	999		
BG100001	HALB	999		
KR090654	HALB	999		

2 つのテーブルの結合

以下の構文では、MARA データベースと MARC データベースが結合されます。

```
SELECT MARA_MATNR, MARA_WRKST, MARC_WERKS FROM MARA, MARC WHERE
MARA_MATNR=MARC_MATNR AND MARC_WERKS='5100';
```

この SQL ステートメントに相当する WebFOCUS の構文は、以下のとおりです。

```
JOIN CLEAR *
JOIN MARA_MATNR IN MARA TO ALL MARC_MATNR IN MARC AS JO
TABLE FILE MARA
PRINT MARA_MATNR MARA_WRKST MARC_WERKS
WHERE MARC_WERKS EQ '5100';
END
```

出力結果は次のとおりです。

品目	基本品目	プラント
NC-100-212		5100
NC-100-213		5100
NC-100-311		5100

品目	基本品目	プラント
NC-100-312		5100
NC-103-101		5100
NC-103-211		5100

SAP Hana アダプタの使用

SAP Hana アダプタの構成方法について説明します。

SAP Hana アダプタを使用すると、アプリケーションから特定の SAP Hana データソースへのアクセスが可能になります。このアダプタにより、アプリケーションリクエストが SAP Hana コールに変換され、最適化されたアンサーセットがリクエストの送信元アプリケーションに返されます。

注意：バージョン 9.3.0 以降、SAP Hana アダプタは UPSERT サポートを提供します。

SAP Hana 環境の準備

SAP Hana アダプタを使用するには、アクセスするデータソースの SAP Hana ドライバをインストールする必要があります。また、サーバを起動する前に CLASSPATH 値を設定するほか、JSCOM3 サービスを実行状態にしておく必要があります。

Windows および UNIX で環境を設定するには

手順

1. サーバを起動する前に、SAP Hana ドライバファイルのパスを CLASSPATH 環境変数に追加することで、これらのファイルの格納先を設定します。

たとえば、UNIX で SAP Hana ドライバファイルのパスを /usr/driver_files/mydriver.jar に設定するには、次のコマンドを実行します。

```
CLASSPATH=/usr/driver_files/mydriver.jar;$CLASSPATH
export CLASSPATH
```

Windows では、以下のコマンドを実行します。


```
set CLASSPATH=c:¥usr¥driver_files¥mydriver.jar:%CLASSPATH%
```

サーバのワークスペースを Windows サービスとして実行する場合、CLASSPATH をシステム全体の環境変数として設定する必要があります。

同様に UNIX では、サーバを起動する前に変数を設定して、UNIX プロファイルに CLASSPATH の設定を追加します。

2. サーバを起動または再起動します。
ドライバを変更した場合にもサーバを再起動する必要があります。
3. サーバのワークスペースで以下のコマンドを実行することにより、JSCOM3 サービスが有効であることを確認します。

```
edastart -show
```

また、[スペシャルサービス] セクションに「JSCOM3 active」と表示されていることを確認します。

SAP Hana アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

アダプタは、SAP Hana データソースに接続する際に、接続情報および認証情報を必要とします。

この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。選択したサーバプロファイルにコマンドが追加されます。これらのプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用するプラットフォームでサポートされる場合) があります。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の SAP Hana データソースへの接続を宣言することができます。実際の接続は、最初のクエリ発行時に確立さ

れます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。

6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

SAP Hana の接続属性

SAP Hana アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

URL

SAP Hana データソースのロケーション URL です。

ドライバ名

SAP Hana ドライバの名前です。

```
com.sap.db.jdbc.Driver
```

詳細は、使用しているバージョンのドライバのマニュアルを参照してください。

IBI_CLASSPATH

Java サービスで使用する追加の Java クラスディレクトリ、または jar ファイル名のフルパスを定義します。値を設定するには、通信ファイルを編集するか、WebFOCUS Reporting Server ブラウザインターフェースを使用します。WebFOCUS Reporting Server ブラウザインターフェースの入力ボックスでは、1 行につき 1 つの参照を入力することができます。すべてのプラットフォームでは、ファイルを保存した際に、複数のエントリがコロン (:) で区切られた単一文字列に変換されます。ファイルを手動で編集する際は、区切り文字のコロン (:) を保持する必要があります。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (括弧は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

接続属性の手動宣言

```
ENGINE SQLHANA SET CONNECTION_ATTRIBUTES connection  
'URL'/userid,password
```

説明

SQLHANA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

接続名です。

URL

SAP Hana データソースのロケーション URL です。

userid

ターゲットデータベースに登録されたプライマリ認証 ID です。

password

プライマリ認証 ID に関連するパスワードです。

接続属性の宣言

以下の SET CONNECTION_ATTRIBUTES コマンドは、明示的なユーザ ID (MYUSER) およびパスワード (PASS) を使用して、SAP Hana ドライバを介してデータソースにアクセスします。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザインターフェースまたは Data Migrator デスクトップインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

```
ENGINE SQLHANA SET CONNECTION_ATTRIBUTES CON1  
'jdbc:sap://host:port'
```

デフォルト接続の上書き変更

複数の接続を定義している場合、最初の SET CONNECTION_ATTRIBUTES コマンドに指定した接続が、デフォルトの接続として設定されます。

デフォルト接続の変更

```
ENGINE SQLHANA SET DEFAULT_CONNECTION connection
```

説明

SQLHANA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

```
ENGINE SQLHANA SET DEFAULT_CONNECTION SAMPLE
```

SAP Hana メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および JDBC データタイプのサーバマッピングを定義するシノニムを作成します。

アダプタの識別

リクエストの解釈に必要なアダプタを識別するには、マスターファイルの SUFFIX 属性を使用します。SAP Hana アダプタを識別するには、SUFFIX 値として SQLHANA を使用します。

アダプタの識別

```
FILE[NAME]=file, SUFFIX=SQLHANA [,,$]
```

説明

file

マスターファイルのファイル名です。このファイル名は文字で開始する必要があります。また、テーブルまたはビューの内容を表す名前にすることをお勧めします。実際のファイルには拡張子 .mas を付ける必要がありますが、この属性の値には拡張子を含めないでください。

SQLHANA

アダプタの値です。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な各オブジェクトの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

SAP Hana のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

外部 SQL スクリプトのパス

[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合は、さらに次の項目が表示されます。

次の標準名前規則は UNIX に適用されます。

- [ベースパス] テキストボックスに、SQL クエリが記述されたファイルの物理ディレクトリパスを指定します。ディレクトリ名を直接入力するか、[...] (参照) ボタンをクリックしてファイルの格納先ディレクトリへ移動します。このクリックにより、[ベースパスの選択] ダイアログボックスが表示されます。
- [ドキュメント名] テキストボックスに、ファイル名を入力します。必要に応じて、ワイルドカード文字を使用します。
- [ドキュメント拡張子] テキストボックスに、スクリプトファイルの拡張子を入力し

て、その拡張子のファイルのみを表示します。

アダプタはシノニムの作成中にネイティブ API コールを発行して選択リスト内の要素のリストを取得し、各要素に対するフィールドを持つマスターファイルを構成します。生成されたアクセスファイルは、DATASET 属性の SQL スクリプトパスを参照します。これには SQL クエリが格納されたファイルの名前と拡張子を含む、フルパスが記述されています。以下はその例です。

```
DATASET=/ul/home2/apps/report3.sql
```

WebFOCUS レポートを作成する際は、SQL クエリを使用してデータにアクセスします。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

外部キーを使用したクラスタ作成 (非推奨。[シノニムの編集] の使用を推奨)

[外部キーを使用したクラスタ作成] のチェックをオンにすると、現在のテーブルに外部キーで関係付けられているテーブルすべてをこのシノニムに含めることができます。ただし、このオプションは廃止される予定で、クラスタの作成にはシノニムエディタを使用することをお勧めします。生成されるマルチテーブルのシノニムは、このテーブルの外部キーの関係をすべて定義します。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (;) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視

され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[データタイプサポートレポート](#)を参照してください。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

オーナー/スキーマ

ユーザが所有する 1 つまたは複数のオブジェクトを作成したユーザアカウントです。

テーブル名

基になるオブジェクトの名前です。

タイプ

オブジェクトタイプ (テーブル、ビューなど)。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

生成されたシノニムの例

SAP Hana のシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

マスターファイル nf29004.mas

```
FILE=DIVISION, SUFFIX=SQLHANA , $  
SEGNAME=SEG1_4, SEGTYPE=S0 , $  
FIELD=DIVISION4, DIVISION4, I9, I4, MISSING=OFF , $  
FIELD=DIVISION_NA4, DIVISION4, I9, I4, MISSING=OFF , $  
FIELD=DIVISION_HE4, DIVISION4, I9, I4, MISSING=OFF , $
```

アクセスファイル nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=EDAQA.NF29004,  
CONNECTION=CON1, KEYS=1, WRITE=YES, $
```

アクセスファイルのキーワード

下表は、アクセスファイルで使用されるキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABlename	SAP Hana テーブルを識別します。この属性に割り当てる値には、次のようにオーナー (スキーマとも呼ばれる) の名前とデータベースリンク名を含めることができます。 <pre>TABlename=[owner.]table</pre>
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre> CONNECTION=' ' は、ローカルのデータソースへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。

キーワード	説明
	<pre>KEY=fld1/fld2/.../fldn</pre>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。
KEYFLD IXFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定します。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 • KEYFLD は、親テーブルの共通カラムの FIELDNAME です。 • IXFLD は関連するテーブルの共通カラムの FIELDNAME です。 KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることをお勧めしますが、必須ではありません。 注意：KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

データタイプサポートレポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[アクセスファイル](#)を参照してください。

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアードプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

SAP Hana 環境のカスタマイズ

SAP Hana アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。

タイムアウト制限の指定

SAP Hana に対して SQL リクエストを実行した後、アダプタがその応答を待機する時間 (秒) を指定するには、TIMEOUT コマンドを使用します。

TIMEOUT コマンドの発行

```
ENGINE SQLHANA SET TIMEOUT {nn|0}
```

説明

SQLHANA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

nn

タイムアウトとなる秒数を指定します。デフォルト値は 30 です。

0

応答の待機時間が無制限であることを示します。

長時間実行中リクエストのキャンセル

WebFOCUS Reporting Server ブラウザインターフェースで、長時間実行中のリクエストをキャンセルすることができます。この操作を実行すると、ネイティブ SAP Hana ドライバの機能に応じて、リクエスト全体がキャンセルされるか、フェッチサイクルが中断されます。

長時間実行中のリクエストをキャンセルするには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [ワークスペース] ページで、[Java サービス] 下の [DEFAULT] を選択します。[DEFAULT] を右クリックし、[エージェント] を選択します。
2. [Java サービスエージェント] ウィンドウで、キャンセルする jscomid の行を選択し、右クリックして [終了] を選択します。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

```
ENGINE SQLHANA SET PASSRECS {ON|OFF}
```

説明

SQLHANA

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

SAP Hana 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)を参照してください。

Snowflake Cloud Data Warehouse アダプタの使用

Snowflake Cloud Data Warehouse アダプタの構成方法について説明します。

Snowflake Cloud Data Warehouse アダプタを使用すると、アプリケーションから Snowflake クラウドデータソースへのアクセスが可能になります。このアダプタにより、データまたはアプリケーションリクエストがネイティブの Snowflake SQL ステートメントに変換され、最適化されたアンサーセットがリクエストの送信元プログラムに返されます。

ODBC および JDBC の 2 つのタイプのアダプタが使用できます。サーバインスタンスごとに 1 つのタイプのみ構成可能です。

注意：バージョン 9.3.0 以降、Snowflake Cloud Data Warehouse アダプタは UPSERT サポートを提供します。

Snowflake JDBC 環境の準備

Snowflake は、Java 1.7 以降を必要とする JDBC タイプ 4 ドライバを使用し、64 ビット環境にインストールする必要があります。

サーバを起動する前に、Snowflake ドライバのインストール先のディレクトリパスが IBI_CLASSPATH に追加されていることを確認してください。

以下はその例です (Linux の場合)。

```
export IBI_CLASSPATH=/qas/snowflake/snowflake-jdbc-3.6.3.jar
```

Snowflake Cloud Data Warehouse

ODBC 環境の準備

Snowflake Cloud Data Warehouse ODBC 環境は、Windows および Linux Red Hat バージョン 7 および 6 で使用できます。

オペレーティングシステムに対応する Snowflake ODBC ドライバをダウンロード、インストールします。使用するオペレーティングシステムが Linux の場合、Linux ドライバマネージャのインストールが必要になります (インストール済みでない場合)。

Snowflake Cloud Data Warehouse アダプタの構成 - JDBC

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

Snowflake Cloud Data Warehouse アダプタを構成するには - JDBC

アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで構成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Snowflake Cloud Data Warehouse アダプタの構成設定 - JDBC

Snowflake Cloud Data Warehouse アダプタは、[SQL] グループフォルダ下にあります。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

URL

Snowflake データソースの URL です。

基本構文は、以下のとおりです。

```
jdbc:snowflake://Server/?warehouse=&|db=&|schema=
```

文字列の末尾に区切り文字のアンパサンド(&|)を追加することで、接続パラメータ (例、tracing=off) を追加できます。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **キーペア認証** この認証方法には、少なくとも 2048 ビットの RSA キーペアが必要です。パブリックキーはユーザに割り当てられます。ユーザ認証では、接続情報とともに送信されたプライベートキーに基づいてパブリックキーが検証されます。

Explicit

以下は、[Explicit] セキュリティの接続属性です。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

キーペア認証

以下は、[キーペア認証] セキュリティの接続属性です。

ユーザ

データソースに登録されたプライマリ認証 ID です。

Private key file password

プライベートキーを保護するために使用されるパスワードです。

Private key file location

プライベートキーの安全な場所です。プライベートキーファイルの場所はネットワーク上を指定することもできます。以下はその例です。

```
¥¥example.cloud.com¥sno¥keys¥rsa_key.p8
```

または、ローカルディレクトリを指定することもできます。以下はその例です。

```
/qa/sno/keys/rsa_key.p8
```

注意：UNIX、Linux、Windows のいずれのオペレーティングシステムの場合でも、パス名の区切り文字としてスラッシュ (/) または円記号 (¥) 文字を使用できます。

詳細パラメータ

詳細パラメータには、次の 4 つがあります。

- **ドライバ名 (オプション)** JDBC ドライバの名前です。以下はその例です。

```
net.snowflake.client.jdbc.SnowflakeDriver
```

- **IBI_CLASSPATH** ドライバの jar クラスパスです。以下はその例です。

```
/qas/snowflake/snowflake-jdbc-3.6.3.jar
```

- **データロードの許可** この接続でアップロード、クイックコピー、カスタムコピーを許可する場合は、[はい] を選択します。デフォルト値は [はい] です。
- **接続の説明 (オプション)** この接続の説明に使用するユーザ定義の記述です。このパラメータはオプションとして指定します。

環境

必須および情報として提供する環境変数を定義することができます。必須の変数は、サーバを開始する前に定義する必要があります。

環境パラメータには、次の 3 つがあります。

- **CLASSPATH** IBI-CLASSPATH で定義されていない場合に使用する、ユーザ定義のクラスおよびパッケージを検索するための JVM または Java ベースのアプリケーションディレクトリ検索パスです。
- **PATH** 現在のディレクトリに存在しない、またはフルパス名で呼び出される実行可能プログラムのディレクトリ検索パスです。この設定が必要かどうかを確認するには、ベンダーのマニュアルを確認してください。
- **LD_LIBRARY_PATH** 呼び出された実行可能ファイルまたは DLL ファイルによって直接的、または間接的に使用される DLL ファイルのディレクトリ検索パスです。この設定が必要かどうかを確認するには、ベンダーのマニュアルを確認してください。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、`CONNECTION_ATTRIBUTES` コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの `edasprof.prf` です。

新しいユーザプロファイル (`user.prf`) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (`edasprof`) に保存します。

JDBC タイプの Snowflake アダプタでは、URL のパラメータ区切り文字として、常にアンパサンド (&) をこの順序で使用します。

拡張バルクロード機能は、JDBC タイプの Snowflake アダプタでサポートされており、追加の構成手順は必要ありません。

Snowflake Cloud Data Warehouse アダプタの構成 - ODBC

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

Snowflake Cloud Data Warehouse アダプタを構成するには - ODBC

アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで構成することができます。

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開しま

す。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Snowflake Cloud Data Warehouse アダプタの構成設定 - ODBC

Snowflake Cloud Data Warehouse アダプタは、[SQL] グループフォルダ下にあります。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

DSN を使用しない

DSN を使用しない接続は、Snowflake Cloud Data Warehouse への代替接続方法です。このパラメータを選択した場合、[データソース名] (DSN) の代わりに Snowflake のサーバ、ウェアハウス、デフォルトデータベース、デフォルトスキーマの値を入力することで、アダプタが構成されます。

サーバ

Snowflake が提供するアカウントの完全なドメイン名を指定します。以下はその例です。

```
ar59294.us-east-1.snowflakecomputing.com
```

ウェアハウス

ドライバが開始するセッションで使用するデフォルトウェアハウスを指定します。以下はその例です。

```
QAWH
```

デフォルトデータベース

ドライバが開始するセッションで使用するデフォルトデータベースを指定します。以下はその例です。

```
QATST
```

デフォルトスキーマ

ドライバが開始するセッションで使用するデフォルトスキーマを指定します。以下はその例です。

```
R729999D
```

DSN

有効な Snowflake Data Source Name (DSN) です。デフォルトの DSN はありません。つまり、値を入力する必要があります。この DSN 名は、Windows では ODBC アドミニストレータで構成されたユーザ DSN、システム DSN、またはファイル DSN と一致させる必要があります。Linux では \$HOME/.odbc.ini ファイルの DSN エントリと一致させる必要があります。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **キーペア認証** この認証方法には、少なくとも 2048 ビットの RSA キーペアが必要です。パブリックキーはユーザに割り当てられます。ユーザ認証では、接続情報とともに送信されたプライベートキーに基づいてパブリックキーが検証されます。

Explicit

以下は、[Explicit] セキュリティの接続属性です。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

キーペア認証

以下は、[キーペア認証] セキュリティの接続属性です。

ユーザ

データソースに登録されたプライマリ認証 ID です。

Private key file password

プライベートキーを保護するために使用されるパスワードです。

Private key file location

プライベートキーの安全な場所です。プライベートキーファイルの場所はネットワーク上を指定することもできます。以下はその例です。

```
¥¥example.cloud.com¥sno¥keys¥rsa_key.p8
```

または、ローカルディレクトリを指定することもできます。以下はその例です。

```
/qa/sno/keys/rsa_key.p8
```

注意：UNIX、Linux、Windows のいずれのオペレーティングシステムの場合でも、パス名の区切り文字としてスラッシュ (/) または円記号 (¥) 文字を使用できます。

追加の接続文字列キーワード (オプション)

接続文字列のオプションを追加するために使用します。たとえば、Snowflake ODBC のトレースが必要ない場合は、tracing=0 パラメータが必要です。

詳細パラメータ

詳細パラメータには、次の 2 つがあります。

- **データロードの許可** この接続でアップロード、クイックコピー、カスタムコピーを許可する場合は、[はい] を選択します。デフォルト値は [はい] です。
- **接続の説明 (オプション)** この接続の説明に使用するユーザ定義の記述です。このパラメータはオプションとして指定します。

環境

必須および情報として提供する環境変数を定義することができます。必須の変数は、サーバを開始する前に定義する必要があります。

環境パラメータには、次の 3 つがあります。

- **CLASSPATH** IBI-CLASSPATH で定義されていない場合に使用する、ユーザ定義のクラスおよびパッケージを検索するための JVM または Java ベースのアプリケーションディレクトリ検索パスです。
- **PATH** 現在のディレクトリに存在しない、またはフルパス名で呼び出される実行可能プログラムのディレクトリ検索パスです。この設定が必要かどうかを確認するには、ベンダーのマニュアルを確認してください。
- **LD_LIBRARY_PATH** 呼び出された実行可能ファイルまたは DLL ファイルによって直接的、または間接的に使用される DLL ファイルのディレクトリ検索パスです。UNIX および Linux オペレーティングシステムの場合、このパラメータは UNIX ODBC ドライバマネージャのインストールパスを指定するために使用されます。この設定が必要かどうかを確認するには、ベンダーのマニュアルを確認してください。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

Snowflake Cloud Data Warehouse (JDBC) でのシノニムの作成

シノニムを作成して、サーバからアクセス可能な Snowflake Cloud Data Warehouse テーブルの一意の名前またはエイリアスを定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、動作を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが指定した Snowflake Cloud Data Warehouse テーブルに基づいて生成されます。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。

- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
 - **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
 - **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
 - **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。
3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。
 4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Snowflake Cloud Data Warehouse (JDBC) のシノニム作成パラメータ

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

データベース

テーブルまたはその他のオブジェクトを選択できるデータベースを指定します。ドロップダウンリストからデータベースを選択することができます。接続パラメータでデフォルトデータベースとして指定するデータベースを選択するには、ドロップダウンリストから [デフォルトデータベース] を選択します。

オーナー/スキーマ

ドロップダウンリストからスキーマを選択するか、選択にフィルタを設定するための文字列を入力します。必要に応じて、文字列の先頭または末尾にワイルドカード文字 (%) を挿入します。たとえば、「ABC%」と入力すると、オーナー/スキーマが文字列 ABC で始まるテーブルまたはビューが選択され、「%ABC」と入力すると、オーナー/スキーマが文字列 ABC で終わるテーブルまたはビューが選択されます。「%ABC%」と入力すると、文字列 ABC を先頭、中間、または末尾に含むオーナー/スキーマのテーブルまたはビューが選択されます。

オブジェクト名

オブジェクト名のフィルタに使用する文字列を入力します。必要に応じて、文字列の先頭や末尾にワイルドカード文字 (%) を入力します。たとえば、「ABC%」と入力して、名前が ABC で始まるオブジェクトをすべて選択します。「%ABC」と入力して、名前が ABC で終わるオブジェクトをすべて選択します。「%ABC%」と入力して、名前の先頭、中間、または末尾に ABC が含まれるオブジェクトをすべて選択します。

オブジェクト名にフィルタを設定するには、[オブジェクト名] テキストボックスに文字列を入力してから、[フィルタ] アイコンをクリックします。

データタイプマッピングのカスタマイズ

デフォルト設定のデータタイプマッピングを変更するには、[データマッピングのカスタマイズ] セクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なデータタイプマッピングは、次のとおりです。

- **整数精度** 整数精度列のマッピングを制御します。デフォルト値は 11 です。
- **倍精度浮動小数点数精度** 倍精度浮動小数点数精度列のマッピングを制御します。デフォルト値は 20 です。
- **倍精度浮動小数点数小数部** 倍精度浮動小数点数小数部列のマッピングを制御します。デフォルト値は 2 です。
- **LONGCHAR のマッピング** LONGCHAR データタイプの処理とマッピングを制御します。LONGCHAR のマッピングオプションは、[ALPHA]、[TEXT]、または [BLOB] です。デフォルト値は ALPHA です。
- **LONGCHAR 長さ** この設定は、[LONGCHAR のマッピング] の値を ALPHA に設定した場合にのみ使用できます。デフォルト値は 256 です。
- **日付時間の精度 (桁数)** 日付時間列のマッピングを制御します。この設定は、互換性のために用意されています。[ON] に設定すると、日付時間の正確な精度がシノニムフィールドのフォーマットとして保持されます。たとえば、TIMESTAMP(5) は HYYMDs ではなく HYYMD5 にマッピングされるため、小数点以下 5 桁がすべて保持されます。デフォルト値は [OFF] です。
- **日付フィールドを日付構成要素に分解** [オン] に設定すると、日付フィールド、および日付時間フィールドの日付部分が、年、四半期、月、日構成要素に分解されます。デフォルト値は [オフ] です。
- **日付の順序** マスターファイルへの日付フォーマットのマッピング (YYMD、MDYY、DMYY) を制御します。デフォルト値は、アダプタによって異なります。
- **GEOGRAPHIC_ROLE の割り当てを有効にする** [オン] に設定すると、ボキャブラリ情報および構成ファイルのロールに従ったフィールド名分析に基づいて、GEOGRAPHIC_ROLE が割り当てられます。シノニムの作成によって、GEOGRAPHIC_ROLE マスターファイル属性が、サーバ構成ファイル (edahome/catalog/geo_services.xml) で定義されている特定のジオメトリ (GEOMETRY_AREA または GEOMETRY_POINT) または地理情報 (CITY、STATE、ZIP5 など) ロールにマッピングされます。値の完全なリストを取得するには、edahome/catalog/geo_srv_roles.fex を実行します。デフォルト値は [オン] です。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は `ibisamp` です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスはブランクにしておきます。

最大行数

[シノニムの作成] ページに表示するオブジェクト数を指定します。

カラムの選択

選択したデータベースのシノニム候補として表示するカラムを選択します。

フィルタ

シノニム候補リストからの選択にフィルタを設定します。シノニム候補リストからの選択にフィルタを設定するには、フィルタとして使用する文字列を入力し、検索アイコンをクリックします。たとえば、「ABC」と入力すると、名前に ABC が含まれるすべてのオブジェクトが検索されます。

Snowflake Cloud Data Warehouse (ODBC) でのシノニムの作成

シノニムを作成して、サーバからアクセス可能な Snowflake Cloud Data Warehouse テーブルの一意の名前またはエイリアスを定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、動作を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが指定した Snowflake Cloud Data Warehouse テーブルに基づいて生成されます。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Snowflake Cloud Data Warehouse (ODBC) のシノニム作成パラメータ

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

データベース

テーブルまたはその他のオブジェクトを選択できるデータベースを指定します。ドロップダウンリストからデータベースを選択することができます。接続パラメータでデフォルトデータベースとして指定するデータベースを選択するには、ドロップダウンリストから [デフォルトデータベース] を選択します。

オーナー/スキーマ

ドロップダウンリストからスキーマを選択するか、選択にフィルタを設定するための文字列を入力します。必要に応じて、文字列の先頭または末尾にワイルドカード文字 (%) を挿入します。たとえば、「ABC%」と入力すると、オーナー/スキーマが文字列 ABC で始まるテーブルまたはビューが選択され、「%ABC」と入力すると、オーナー/スキーマが文字列

ABC で終わるテーブルまたはビューが選択されます。「%ABC%」と入力すると、文字列 ABC を先頭、中間、または末尾に含むオーナー/スキーマのテーブルまたはビューが選択されます。

オブジェクト名

オブジェクト名のフィルタに使用する文字列を入力します。必要に応じて、文字列の先頭や末尾にワイルドカード文字 (%) を入力します。たとえば、「ABC%」と入力して、名前が ABC で始まるオブジェクトをすべて選択します。「%ABC」を入力して、名前が ABC で終わるオブジェクトをすべて選択します。「%ABC%」と入力して、名前の先頭、中間、または末尾に ABC が含まれるオブジェクトをすべて選択します。

オブジェクト名にフィルタを設定するには、[オブジェクト名] テキストボックスに文字列を入力してから、[フィルタ] アイコンをクリックします。

データタイプマッピングのカスタマイズ

デフォルト設定のデータタイプマッピングを変更するには、[データマッピングのカスタマイズ] セクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なデータタイプマッピングは、次のとおりです。

- **整数精度** 整数精度列のマッピングを制御します。デフォルト値は 11 です。
- **倍精度浮動小数点数精度** 倍精度浮動小数点数精度列のマッピングを制御します。デフォルト値は 20 です。
- **倍精度浮動小数点数小数部** 倍精度浮動小数点数小数部列のマッピングを制御します。デフォルト値は 2 です。
- **LONGCHAR のマッピング** LONGCHAR データタイプの処理とマッピングを制御します。LONGCHAR のマッピングオプションは、[ALPHA]、[TEXT]、または [BLOB] です。デフォルト値は ALPHA です。
- **LONGCHAR 長さ** この設定は、[LONGCHAR のマッピング] の値を ALPHA に設定した場合にのみ使用できます。デフォルト値は 256 です。
- **日付時間の精度 (桁数)** 日付時間列のマッピングを制御します。この設定は、互換性のために用意されています。[ON] に設定すると、日付時間の正確な精度がシノニムフィールドのフォーマットとして保持されます。たとえば、TIMESTAMP(5) は HYYMDs ではなく HYYMD5 にマッピングされるため、小数点以下 5 桁がすべて保持されます。デフォルト値は [OFF] です。
- **日付フィールドを日付構成要素に分解** [オン] に設定すると、日付フィールド、および日付時間フィールドの日付部分が、年、四半期、月、日構成要素に分解されます。デフォルト値は [オフ] です。

- **日付の順序** マスターファイルへの日付フォーマットのマッピング (YYMD、MDYY、DMYY) を制御します。デフォルト値は、アダプタによって異なります。
- **GEOGRAPHIC_ROLE の割り当てを有効にする** [オン] に設定すると、ボキャブラリ情報および構成ファイルのロールに従ったフィールド名分析に基づいて、GEOGRAPHIC_ROLE が割り当てられます。シノニムの作成によって、GEOGRAPHIC_ROLE マスターファイル属性が、サーバ構成ファイル (edahome/catalog/geo_services.xml) で定義されている特定のジオメトリ (GEOMETRY_AREA または GEOMETRY_POINT) または地理情報 (CITY、STATE、ZIP5 など) ロールにマッピングされます。値の完全なリストを取得するには、edahome/catalog/geo_srv_roles.fex を実行します。デフォルト値は [オン] です。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は ibisamp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

最大行数

[シノニムの作成] ページに表示するオブジェクト数を指定します。

カラムの選択

選択したデータベースのシノニム候補として表示するカラムを選択します。

フィルタ

シノニム候補リストからの選択にフィルタを設定します。シノニム候補リストからの選択にフィルタを設定するには、フィルタとして使用する文字列を入力し、検索アイコンをクリックします。たとえば、「ABC」と入力すると、名前に ABC が含まれるすべてのオブジェクトが検索されます。

データタイプのサポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[アクセスファイル](#)を参照してください。

Snowflake Cloud Data Warehouse アダプタの設定 - JDBC

Snowflake Cloud Data Warehouse では、[設定の変更] ダイアログボックスから、さまざまな設定を行うことができます。

このダイアログボックスを開くには、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで、Snowflake Cloud Data Warehouse アダプタを右クリックし、[設定の変更] をクリックします。Snowflake Cloud Data Warehouse アダプタには、次の設定があります。

バルクロードサービス

[バルクロードサービス] には、次のオプションがあります。

BULKLOAD

データの DBMS へのロードに使用する手法です。[ON] に設定すると、拡張バルクロード機能が有効になります。[OFF] に設定すると、標準 SQL INSERT ステートメントが有効になります。デフォルト値は [OFF] です。

BULKCHECK

データベースに対してトランザクションをコミット (書き込み) する行間隔です。デフォルト値は 1000000 または DBMS 固有のいずれかです。

注意：このオプションは、[BULKLOAD] パラメータが [ON] に設定されている場合にのみ適用されます。

BLK_DFIXDELIM

データベースにロードされる中間 Flat File 内のフィールド間の区切り文字です。区切り文字は、DBMS に応じて最大 4 文字で構成できます。

次のオプションがあります。

- **TAB** タブ文字を示します。
- **a** ~などの文字列を示します。
- **0xNN** 16 進コードを示します。たとえば、カンマ (,) の 16 進コードは 0x44 です。16 進数コードは、Windows および UNIX システムでは ASCII を使用します。

デフォルト値は [TAB] です。

BLK_ESCAPE

[a] に設定すると、¥n、¥t、¥100 などのエスケープシーケンスに使用する 1 文字が指定されます。また、行または列の区切り文字として解釈されるデータ文字をエスケープするためにも使用されます。デフォルト値は [OFF] です。

BLK_DFIXENCLOSURE

データベースにロードされる中間 Flat File 内の各文字値を囲むために使用されます。囲み文字のシーケンスは、DBMS に応じて最大 4 文字で構成できます。囲み文字には、ピリオド (.)、プラス記号 (+)、マイナス記号 (-) などの数値に関する符号を使用することはできません。

次のオプションがあります。

- **a** ~などの文字列を示します。
- **0xNN** 16 進コードを示します。たとえば、カンマ (,) の 16 進コードは 0x44 です。

デフォルト値は DBMS によって異なります。

DIRECT_BULK_LOAD

データの DBMS へのロードの手法を制御します。[ON] に設定すると、DBMS ユーティリティによって、データが直接ロードされます。[OFF] に設定すると、ソースデータが中間ファイルに抽出されます。デフォルト値は、アダプタによって異なります。

BLK_SAVE_DATA_FILES

プロセスが完了するまで、中間のバルクロードデータファイルを保存することができます。たとえば、Salesforce アダプタの場合、REST API ソースデータファイルを APPDIR/TargetMFD_JOBID_nnnnnnnnnn 内のフォルダに保存できます。ここで、nnnnnnnnnn はリクエストに割り当てられたジョブ ID です。[YES] に設定すると、BLK_SAVE_REPONSE_LOG が暗黙的に YES に設定されます。デフォルト値は [NO] です。

BLK_PERM_AS_STAGE

バルクロード処理中に、永続データベーステーブルを中間テーブルとして使用する権限を制御します。デフォルト値は [ON] です。

データタイプマッピングのカスタマイズ

[データタイプマッピングのカスタマイズ] には、次のオプションがあります。

CONV_INT_PREC

整数精度列を制御します。デフォルト値は 11 です。

CONV_FLOAT_PREC

倍精度浮動小数点数精度列のマッピングを制御します。デフォルト値は 20 です。

CONV_FLOAT_SCALE

倍精度浮動小数点数小数部列のマッピングを制御します。デフォルト値は 2 です。

CONV_DEC_PREC

パック 10 進数精度列のマッピングを制御します。このデータタイプにデフォルト値はありません。

CONV_DEC_SCALE

倍精度浮動小数点数小数部列のマッピングを制御します。このデータタイプにデフォルト値はありません。

この設定は、非推奨の SET CONVERSION PRECISION p s コマンドに優先します。

CONV_LONGCHAR

LONGCHAR 列のマッピングを制御します。LONGCHAR のマッピングオプションは、[ALPHA]、[TEXT]、または [BLOB] です。デフォルト値は [ALPHA] です。

CONV_LONGCHAR_LEN

この設定は、[CONV_LONGCHAR] の値を [ALPHA] に設定した場合にのみ使用できます。デフォルト値は 256 です。

EXACT_DATETIME

日付時間列のマッピングを制御します。この設定は、互換性のために用意されています。[ON] に設定すると、日付時間の正確な精度がシノニムフィールドのフォーマットとして

保持されます。たとえば、TIMESTAMP(5) は HYMDs ではなく HYMD5 にマッピングされるため、小数点以下 5 桁がすべて保持されます。デフォルト値は [OFF] です。

メタデータ

メタデータのオプションは、次のとおりです。

ALIAS_CASE

新しいターゲットの ALIAS の大文字と小文字を設定することができます。

次のオプションがあります。

- **変換しない** 新しいターゲットの ALIAS のデフォルトの大文字と小文字は変更されません。
- **小文字** 新しいターゲットの ALIAS を小文字に変更します。
- **大文字** 新しいターゲットの ALIAS を大文字に変更します。

デフォルト値は [変換しない] です。

TABLE_OPTIONS

CREATE FILE コマンドによって生成された CREATE TABLE ステートメントにテーブルオプションパラメータを追加します。この設定にはデフォルト値はありません。

その他の設定

[その他の設定] には、次のオプションがあります。

JOINTYPE

自動パススルーが実行不可または無効になっている場合 (SQL SET APT=OFF)、FOCUS 生成 SQL 変換での JOIN タイプを制御します。

次のオプションがあります。

- **SORTMERGE** 各テーブルからデータを取得し、ソート、結合、集計します。SORTMERGE に設定すると、FOCUS 生成では常にソート/マージを使用してデータを取得します。

- **NESTEDLOOP** 外部テーブルから、選別条件に一致するすべての行を選択し、それらの行を使用して内部テーブルから修飾データを取得します。FOCUS 生成では、SQL ステートメント内の JOIN 可能なフィールドに equijoin が存在する場合にのみ、この方法が使用されます。このような equijoin が存在しない場合、FOCUS 生成はソート/マージに戻されます。

この設定にはデフォルト値はありません。

Snowflake Cloud Data Warehouse アダプタの設定 - ODBC

Snowflake Cloud Data Warehouse では、[設定の変更] ダイアログボックスから、さまざまな設定を行うことができます。

このダイアログボックスを開くには、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで、Snowflake Cloud Data Warehouse アダプタを右クリックし、[設定の変更] をクリックします。Snowflake Cloud Data Warehouse アダプタには、次の設定があります。

バルクロードサービス

[バルクロードサービス] には、次のオプションがあります。

BULKLOAD

データの DBMS へのロードに使用する手法です。[ON] に設定すると、拡張バルクロード機能が有効になります。[OFF] に設定すると、標準 SQL INSERT ステートメントが有効になります。デフォルト値は [OFF] です。

BULKCHECK

データベースに対してトランザクションをコミット (書き込み) する行間隔です。デフォルト値は 1000000 または DBMS 固有のいずれかです。

注意：このオプションは、[BULKLOAD] パラメータが [ON] に設定されている場合にのみ適用されます。

BLK_DFIXDELIM

データベースにロードされる中間 Flat File 内のフィールド間の区切り文字です。区切り文字は、DBMS に応じて最大 4 文字で構成できます。

次のオプションがあります。

- **TAB** タブ文字を示します。
- **a ~** などの文字列を示します。
- **0xNN** 16 進コードを示します。たとえば、カンマ (,) の 16 進コードは 0x44 です。16 進数コードは、Windows および UNIX システムでは ASCII を使用します。

デフォルト値は [TAB] です。

BLK_ESCAPE

[a] に設定すると、¥n、¥t、¥100 などのエスケープシーケンスに使用する 1 文字が指定されます。また、行または列の区切り文字として解釈されるデータ文字をエスケープするためにも使用されます。デフォルト値は [OFF] です。

BLK_DFIXENCLOSURE

データベースにロードされる中間 Flat File 内の各文字値を囲むために使用されます。囲み文字のシーケンスは、DBMS に応じて最大 4 文字で構成できます。囲み文字には、ピリオド (.)、プラス記号 (+)、マイナス記号 (-) などの数値に関する符号を使用することはできません。

次のオプションがあります。

- **a ~** などの文字列を示します。
- **0xNN** 16 進コードを示します。たとえば、カンマ (,) の 16 進コードは 0x44 です。

デフォルト値は DBMS によって異なります。

DIRECT_BULK_LOAD

データの DBMS へのロードの手法を制御します。[ON] に設定すると、DBMS ユーティリティによって、データが直接ロードされます。[OFF] に設定すると、ソースデータが中間ファイルに抽出されます。デフォルト値は、アダプタによって異なります。

BLK_SAVE_DATA_FILES

プロセスが完了するまで、中間のバルクロードデータファイルを保存することができます。たとえば、Salesforce アダプタの場合、REST API ソースデータファイルを APPDIR/TargetMFD_JOBID_nnnnnnnnnn 内のフォルダに保存できます。ここで、

nnnnnnnnnn はリクエストに割り当てられたジョブ ID です。[YES] に設定すると、BLK_SAVE_REPONSE_LOG が暗黙的に YES に設定されます。デフォルト値は [NO] です。

BLK_PERM_AS_STAGE

バルクロード処理中に、永続データベーステーブルを中間テーブルとして使用する権限を制御します。デフォルト値は [ON] です。

データタイプマッピングのカスタマイズ

[データタイプマッピングのカスタマイズ] には、次のオプションがあります。

CONV_INT_PREC

整数精度列を制御します。デフォルト値は 11 です。

CONV_FLOAT_PREC

倍精度浮動小数点数精度列のマッピングを制御します。デフォルト値は 20 です。

CONV_FLOAT_SCALE

倍精度浮動小数点数小数部列のマッピングを制御します。デフォルト値は 2 です。

CONV_DEC_PREC

パック 10 進数精度列のマッピングを制御します。このデータタイプにデフォルト値はありません。

CONV_DEC_SCALE

倍精度浮動小数点数小数部列のマッピングを制御します。このデータタイプにデフォルト値はありません。

この設定は、非推奨の SET CONVERSION PRECISION p s コマンドに優先します。

CONV_LONGCHAR

LONGCHAR 列のマッピングを制御します。LONGCHAR のマッピングオプションは、[ALPHA]、[TEXT]、または [BLOB] です。デフォルト値は [ALPHA] です。

CONV_LONGCHAR_LEN

この設定は、[CONV_LONGCHAR] の値を [ALPHA] に設定した場合にのみ使用できます。デフォルト値は 256 です。

EXACT_DATETIME

日付時間列のマッピングを制御します。この設定は、互換性のために用意されています。
[ON] に設定すると、日付時間の正確な精度がシノニムフィールドのフォーマットとして保持されます。たとえば、TIMESTAMP(5) は HYYMDs ではなく HYYMD5 にマッピングされるため、小数点以下 5 桁がすべて保持されます。デフォルト値は [OFF] です。

メタデータ

メタデータのオプションは、次のとおりです。

ALIAS_CASE

新しいターゲットの ALIAS の大文字と小文字を設定することができます。

次のオプションがあります。

- **変換しない** 新しいターゲットの ALIAS のデフォルトの大文字と小文字は変更されません。
- **小文字** 新しいターゲットの ALIAS を小文字に変更します。
- **大文字** 新しいターゲットの ALIAS を大文字に変更します。

デフォルト値は [変換しない] です。

TABLE_OPTIONS

CREATE FILE コマンドによって生成された CREATE TABLE ステートメントにテーブルオプションパラメータを追加します。この設定にはデフォルト値はありません。

その他の設定

[その他の設定] には、次のオプションがあります。

JOINTYPE

自動パススルーが実行不可または無効になっている場合 (SQL SET APT=OFF)、FOCUS 生成 SQL 変換での JOIN タイプを制御します。

次のオプションがあります。

- **SORTMERGE** 各テーブルからデータを取得し、ソート、結合、集計します。SORTMERGE に設定すると、FOCUS 生成では常にソート/マージを使用してデータを取得します。
- **NESTEDLOOP** 外部テーブルから、選別条件に一致するすべての行を選択し、それらの行を使用して内部テーブルから修飾データを取得します。FOCUS 生成では、SQL ステートメント内の JOIN 可能なフィールドに equijoin が存在する場合にのみ、この方法が使用されます。このような equijoin が存在しない場合、FOCUS 生成はソート/マージに戻されます。

この設定にはデフォルト値はありません。

Sybase アダプタの使用

Sybase アダプタの構成方法について説明します。

Sybase アダプタを使用すると、アプリケーションから Sybase データソースへのアクセスが可能になります。Sybase ASE サーバと Sybase IQ サーバの両方がサポートされます。

このアダプタにより、アプリケーションリクエストがネイティブの Sybase ステートメントに変換され、最適化されたアンサーセットがリクエストの送信元アプリケーションに返されます。

注意：バージョン 9.3.0 以降、Sybase アダプタは MERGE サポートを提供します。

Sybase 環境の準備 - OCS

Sybase アダプタを使用するには、Sybase Open Client (CT-Library) をインストールする必要があります。Sybase クライアントを使用すると、プラットフォーム固有の環境変数を設定した後、ローカルまたはリモートの Sybase データベースサーバに接続できるようになります。インストール後に、ローカルまたはリモートの Sybase データベースサーバに接続するようクライアントを構成する必要があります。

interfaces ファイルのパスの特定

Sybase に固有の分散アクセスを使用するには、SYBASE 環境変数を設定してインタフェースファイルが存在するディレクトリを指定する必要があります。UNIX の interfaces ファイルに相当する Windows のファイルは、sql.ini ファイルです。

interfaces ファイルは、ローカルおよびリモートアクセスの両方に必要となります。SYBASE 環境変数を設定しない場合、Sybase では Sybase というユーザのログインディレクトリの /etc/passwd が検索され、そのディレクトリの interfaces ファイルがアクセスされます。

UNIX PATH に Sybase の実行可能プログラムのパスを含める必要があります。たとえば、Sybase の PATH エントリを以下のように指定します。

```
SYBASE=/usr/sybase  
PATH=$PATH:$SYBASE/bin  
export SYBASE PATH
```

SYBASE 環境変数は、UNIX シェルからエクスポートでき、またサーバを起動したユーザのマシンの UNIX プロファイルにエクスポートすることもできます。

Sybase の実行可能プログラムおよびコンポーネントに必要な他の環境変数については、Sybase のインストールと設定の資料を参照してください。

Sybase Server 名の指定

Sybase Server 名を指定するには、DSQUERY 環境変数を使用します。サーバでは、この値を以下の場合に限り使用します。

- グローバルプロファイル (edasprof.prf) で SET CONNECTION_ATTRIBUTES コマンドを発行していない場合。
- アクセスファイルに CONNECTION= 属性を指定していない場合。

Sybase サーバ名の指定

```
DSQUERY=server  
export DSQUERY
```

説明

server

Sybase データベースサーバの名前です。

リモートの Sybase Server へのアクセス

サーバでは、Sybase クライアントの配備の標準規則を使用して、以下のサーバへの接続をサポートします。

- ローカルの Sybase サーバ。
- リモートの Sybase サーバ。リモートの Sybase サーバに接続するには、interfaces ファイル (Windows 上の sql.ini) にターゲットマシンを指示するエントリが含まれており、ターゲットマシン上でリスナプロセスが実行されている必要があります。

interfaces ファイルの新規エントリは、Sybase のツール DSEDIT を使用して作成できます。エントリは、sybinstall 機能によって自動的に作成される場合があります。詳細は、Sybase の資料を参照してください。

Unicode サポート

ASE の場合 Sybase ASE バージョン 15.0 以降で、Sybase ASE データベースが UTF-8 文字セットを使用して作成されている場合に、アダプタで Unicode データがサポートされます。

IQ の場合 Sybase IQ バージョン 12.7 以降で、Sybase IQ データベースが UTF-8 文字セットを使用して作成されている場合に、アダプタで Unicode データがサポートされます。

サーバを開始する前に、edastart ファイルまたは別のシェルフファイルで LANG 環境変数を設定しておく必要があります。たとえば、英語 (米国) の場合は、次の変数をエクスポートします。

```
export LANG=EN_US.UTF-8
```

アダプタの Unicode データサポートを利用するには、WebFOCUS Reporting Server の構成で UTF-8 のコードページ 65001 を設定する必要があります。

XA のサポート (ASE)

Sybase データソースにアクセスする読み込み/書き込みアプリケーションでは、XA 互換モードで管理されたトランザクションを実行することができます。

XA トランザクション管理機能を有効にするには、Web コンソールの構成機能を使用してサーバをトランザクション調整モードで構成する必要があります。トランザクション調整モードを使用することにより、関連するすべての DBMS に対するデータ修正の整合性が保証され、データ修正の一部がある DBMS に対してコミットされ、別の DBMS に対しては終了されることを回避することができます。

XA のサポートに関する詳しい説明は、[XA のサポート](#)を参照してください。

Sybase 環境の準備 - JDBC

Sybase 対応 JDBC アダプタを使用するには、JDBC ドライバをインストールし、サーバの起動前にその CLASSPATH 値を設定するとともに、JSCOM3 サービスを実行状態にしておく必要があります。

Windows および UNIX で JDBC 環境を設定するには

手順

1. サーバを起動する前に、JDBC ドライバファイルのパスを CLASSPATH 環境変数に追加することで、これらのファイルの格納先を識別します。

たとえば、JDBC ドライバファイルを UNIX 環境の /qas/sybase_ASE/jconn4.jar に設定するには、次のコマンドを発行します。

```
CLASSPATH=/qas/sybase_ASE/jconn4.jar;$CLASSPATH
export CLASSPATH
```

UNIX では、サーバを起動する前に変数を設定するには、UNIX プロファイルに CLASSPATH 設定を追加します。

Windows 環境では、次のコマンドを発行します。

```
set CLASSPATH=%c;%qas%sybase_ASE%jconn4.jar:
%CLASSPATH%
```

サーバのワークスペースを Windows サービスとして実行する場合、CLASSPATH をシステム全体の環境変数として設定する必要があります。

2. サーバを起動または再起動します。

Sybase アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

アダプタは、Sybase データベースサーバに接続する際に、接続情報および認証情報を必要とします。この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。選択したプロファイルにコマンドが追加されます。選択可能なプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) があります。
- グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) のいずれかに手動でコマンドを追加します。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の Sybase データベースサーバへの接続を宣言することができます。Sybase データベースサーバへの実際の接続は、その接続を参照する最初のクエリ発行時に確立されます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが

開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェイスで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェイスで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェイスのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェイスで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェイスで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェイスの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェイスのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェイスで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Sybase の接続属性 - OCS

Sybase アダプタは、[利用可能] ドロップダウンリストの [SQL] 下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

サーバ

アクセスする Sybase サーバの名前です。これは、Sybase の interfaces ファイル内のエントリと一致している必要があります。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

Sybase の接続属性 - JDBC

Sybase アダプタは、[利用可能] ドロップダウンリストの [SQL] 下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

URL

JDBC データソースのロケーション URL です。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

ドライバ名

JDBC ドライバ名です。

詳細は、使用しているバージョンのドライバのマニュアルを参照してください。

IBI_CLASSPATH

Java サービスで使用する追加の Java クラスディレクトリ、または jar ファイル名のフルパスを定義します。この値を設定するには、通信ファイルを編集するか、WebFOCUS Reporting Server ブラウザインタフェースを使用します。WebFOCUS Reporting Server ブラウザインタフェースの入力ボックスでは、1 行につき 1 つの参照を入力することができます。すべてのプラットフォームでは、ファイルを保存した際に、複数のエントリがコロン (:) で区切られた単一文字列に変換されます。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

接続属性の手動宣言

Explicit 認証 ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報が Sybase への接続時に認証情報として渡されます。

```
ENGINE SQLSYB SET CONNECTION_ATTRIBUTES connectionserver/userid,password
```

Password Passthru 認証 クライアントアプリケーションから受け取ったユーザ ID およびパスワードが、Sybase への接続時に認証情報として渡されます。

```
ENGINE SQLSYB SET CONNECTION_ATTRIBUTES connectionserver/
```

説明

connection

この接続属性のセットの識別に使用される論理名です。

SQLSYB

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

server

アクセスする Sybase サーバの名前です。

userid

Sybase に登録されたプライマリ認証 ID です。

password

プライマリ認証 ID に関連するパスワードです。

接続属性の宣言

以下の SET CONNECTION_ATTRIBUTES コマンドでは、明示的なユーザ ID (MYUSER) およびパスワード (PASS) を使用して、SAMPLESERVER という名前の Sybase データベースサーバに接続します。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

```
ENGINE SQLSYB SET CONNECTION_ATTRIBUTES CON01 SAMPLESERVER/MYUSER,PASS
```

以下の SET CONNECTION_ATTRIBUTES コマンドでは、Password Passthru 認証を使用して SAMPLESERVER という名前の Sybase データベースサーバに接続します。

```
ENGINE SQLSYB SET CONNECTION_ATTRIBUTES CON01 SAMPLESERVER/
```

デフォルト接続の上書き変更

複数の接続を定義した場合、最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続がデフォルト接続として機能します。このデフォルト値は、SET DEFAULT_CONNECTION コマンドを使用して上書きすることができます。

デフォルト接続の変更

```
ENGINE SQLSYB SET DEFAULT_CONNECTION connection
```

説明

SQLSYB

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

```
(FOC1671) コマンドの順序に誤りがあります。
```

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コ

マンドを発行することはできません。この場合は、次のメッセージが出力されます。

```
(FOC1671) コマンドの順序に誤りがあります。
```

デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

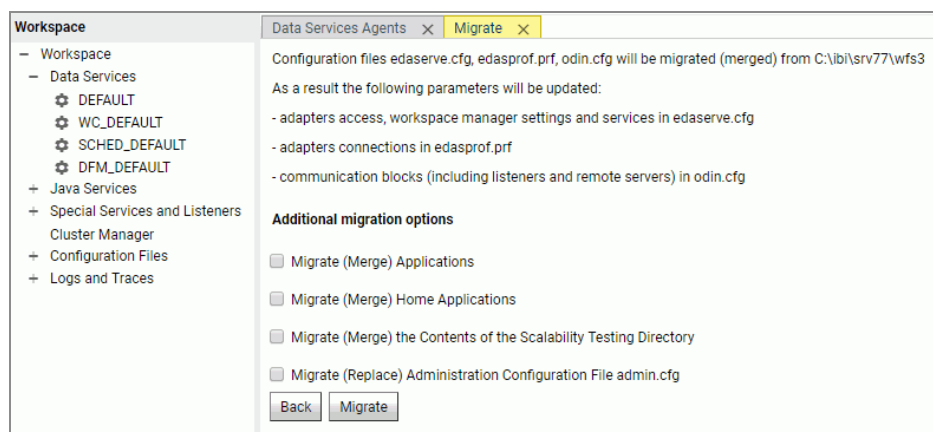
```
ENGINE SQLSYB SET DEFAULT_CONNECTION SAMPLE
```

接続情報の更新

このアダプタの CONNECTION_ATTRIBUTES コマンドの構文が拡張され、アプリケーションの開発環境から実稼働環境への移植をサポートするよう設計された論理接続名を指定できるようになりました。この拡張された構文では、既存の CONNECTION_ATTRIBUTES コマンドのマイグレートが必要になる場合があります。

WebFOCUS Reporting Server ブラウザインタフェースの [マイグレート] オプションを使用すると、現在のサーバ設定が新しいバージョンにマイグレートされます。このオプションにアクセスするには、[ツール] メニューの [ワークスペース] を選択し、[設定] から [マイグレート] を選択します。この方法を使用することをお勧めします。

[マイグレート] ウィンドウで、構成インスタンスディレクトリ (EDACONF) のフルパスを入力し、[続行] をクリックします。下図のように、その他のマイグレートオプションを選択し、[マイグレート] をクリックします。



[マイグレート] オプションを使用しない場合、以下の情報に注意してください。

- バージョン 7.6 SP01 より前のバージョンで宣言された接続名は引き続きサポートされます。
- 新規のシノニムを作成する目的で新しい接続を追加する場合、既存の接続は新しいフォーマットで再保存され、また既存のシノニムは変更を行うことなく引き続き機能します。
- 既存のシノニムを使用する目的で新しい接続を追加する場合、デフォルトの論理接続名を、既存のアクセスファイルの `CONNECTION=value` 属性で指定された値に一致するように変更する必要があります。

たとえば、バージョン 7.6 SP01 より前のバージョンで、以下のように接続が定義されている場合を想定します。

```
ENGINE SQLSYB SET CONNECTION_ATTRIBUTES DSN_A/uid,pwd
```

この DSN_A に保存されているオブジェクトに基づいてシノニムを作成すると、アクセスファイルには以下の記述が追加されます。

```
CONNECTION=DSN_A
```

この後、新しい接続を追加する際に、この既存のシノニムを再利用するには、接続名をデフォルトの CON01 から DSN_A に変更し、DSN_A として保存する必要があります。この接続は、以下のようにプロファイルに保存されます。

```
ENGINE SQLSYB SET CONNECTION_ATTRIBUTES DSN_A DSN_A/uid,pwd
```

接続範囲の制御

確立する接続ごとにアダプタを使用する場合、SET AUTODISCONNECT コマンドを使用して接続の継続を制御することができます。

接続範囲の制御

```
ENGINE SQLSYB SET AUTODISCONNECT ON {FIN|COMMAND|COMMIT}
```

説明

SQLSYB

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMAND

各リクエストの後に自動的に切断されます。SET AUTODISCONNECT コマンドは、イベントがどの程度の頻度で発生するかによって、大幅なオーバーヘッドをもたらす可能性があります。このオーバーヘッドのほとんどすべてがサーバには関連しません。このオーバーヘッドは、オペレーティングシステムとデータソースに関連します。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

Sybase メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとにシノニムを作成して、データソースの構造と Sybase のデータタイプのサーバマッピングを記述します。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な Sybase テーブルまたはビューの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。

- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを複数クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Sybase のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

データベース

テーブルやオブジェクトの選択元となるデータベースを指定するには、次のいずれかを実行します。

- [デフォルトデータベース] を選択して、デフォルトのデータベースとして設定されているデータベースを使用します。
- [データベース] ドロップダウンリストからデータベースを選択します。ここには、現在の DBMS インスタンスにあるすべてのデータベースが表示されます。
[デフォルトデータベース] が選択されている場合は、データベースを選択する前に、このチェックをオフにします。

このオプションは Sybase ASE に適用され、Sybase IQ には適用されません。

外部 SQL スクリプトのパス

[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合は、さらに次の項目が表示されます。

次の標準名前規則は UNIX に適用されます。

- [ベースパス] テキストボックスに、SQL クエリが記述されたファイルの物理ディレクトリパスを指定します。ディレクトリ名を直接入力するか、[...] (参照) ボタンをクリックしてファイルの格納先ディレクトリへ移動します。このクリックにより、[ベースパスの選択] ダイアログボックスが表示されます。
- [ドキュメント名] テキストボックスに、ファイル名を入力します。必要に応じて、ワイルドカード文字を使用します。
- [ドキュメント拡張子] テキストボックスに、スクリプトファイルの拡張子を入力して、その拡張子のファイルのみを表示します。

アダプタはシノニムの作成中にネイティブ API コールを発行して選択リスト内の要素のリストを取得し、各要素に対するフィールドを持つマスターファイルを構成します。生成されたアクセスファイルは、DATASET 属性の SQL スクリプトパスを参照します。これには SQL クエリが格納されたファイルの名前と拡張子を含む、フルパスが記述されています。以下はその例です。

```
DATASET=/u1/home2/apps/report3.sql
```

WebFOCUS レポートを作成する際は、SQL クエリを使用してデータにアクセスします。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

外部キーを使用したクラスタ作成 (非推奨。[シノニムの編集] の使用を推奨)

[外部キーを使用したクラスタ作成] のチェックをオンにすると、現在のテーブルに外部キーで関係付けられているテーブルすべてをこのシノニムに含めることができます。ただし、このオプションは廃止される予定で、クラスタの作成にはシノニムエディタを使用することをお勧めします。生成されるマルチテーブルのシノニムは、このテーブルの外部キーの関係をすべて定義します。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (:) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[Sybase データタイプのサポート](#)を参照してください。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

オーナー/スキーマ

ユーザが所有する 1 つまたは複数のオブジェクトを作成したユーザアカウントです。

テーブル名

基になるオブジェクトの名前です。

タイプ

オブジェクトタイプ (テーブル、ビューなど)。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

生成されたシノニムの例

Sybase アダプタのシノニムは、マスターファイルとアクセスファイルで構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

作成されたマスターファイル

```
FILENAME=NF29004, SUFFIX=SQLSYB  , $
SEGMENT=NF29004, SEGTYPE=S0, $
  FIELDNAME=DIVISION4, ALIAS=DIVISION4, USAGE=I11, ACTUAL=I4, $
  FIELDNAME=DIVISION_NA4, ALIAS=DIVISION_NA4, USAGE=A25, ACTUAL=A25,
    MISSING=ON, $
  FIELDNAME=DIVISION_HE4, ALIAS=DIVISION_HE4, USAGE=I11, ACTUAL=I4,
    MISSING=ON, $
```

作成されたアクセスファイル

```
SEGNAME=NF29004, TABLENAME=qatst.R729999D.NF29004,
CONNECTION=CON01, KEYS=0, $
```

Sybase IQ テーブルのコメントのシノニムへのマッピング

シノニムが作成される際に、アダプタでは Sybase IQ データベーステーブルからコメントを取得して、シノニム内に以下のように配置します。

- Sybase IQ の *Comment on Table* がマスターファイル内の REMARKS にマッピングされます。

- Sybase IQ の *Comment on Field* がマスターファイル内の DESCRIPTION にマッピングされます。

アクセスファイルのキーワード

下表は、アクセスファイルで使用されるキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABlename	Sybase のテーブルを識別します。この属性に割り当てる値には、次のようにオーナー (スキーマとも呼ばれる) の名前とデータベースリンク名を含めることができます。 <pre>TABlename=[owner.]table</pre>
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre> CONNECTION=' ' は、ローカルの Sybase データベースサーバへのアクセスを示します。 CONNECTION 属性の指定がない場合は、デフォルトのデータベースサーバへのアクセスを示します。
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の <i>n</i> フィールドに対応しています。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。

キーワード	説明
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。 <pre>KEY=fld1/fld2/.../fldn</pre>
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。
KEYFLD IXFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定します。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 - KEYFLD は、親テーブルの共通カラムの FIELDNAME です。 - IXFLD は関連するテーブルの共通カラムの FIELDNAME です。 KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることをお勧めしますが、必須ではありません。 注意：KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。
AUTO INCREMENT	自動インクリメントを有効にするには [Yes] に設定します。
START	インクリメントの順序の初期値です。

キーワード	説明
INCREMENT	インクリメントの間隔です。
INDEX_NAME INDEX_UNIQUE INDEX_ COLUMNS INDEX_ORDER	データベース内のインデックス名、およびインデックスフィールドの一意性、名前、順序を示します。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

同一 Sybase Server 上の別データベースへのアクセス

サーバが Sybase に接続する際、プライマリ認証 ID を使用します。この ID は、サーバ上のデフォルトのデータベースに関連付けられています。別のデータベース内のテーブルにアクセスするには、次のように完全修飾名を指定する必要があります。

```
database.owner.table
```

この完全修飾名は、SQL リクエスト内で使用することができます。この名前は、接続に使用されたプライマリ認証 ID に関連付けられたデフォルト以外のデータベース内のテーブルに対する CREATE SYNONYM コマンドで使用する必要があります。同一 Sybase サーバ上で複数のデータベースへのアクセスが許可されている場合、SET CONNECTION_ATTRIBUTES コマンドに指定するユーザ ID には、データベースへのアクセス権だけでなく、データベース内のテーブルへのアクセス権も必要です。このアクセス権の設定は Sybase のストアードプロシ

ジャ sp_adduser および SQL GRANT ステートメントを使用して行えます。詳細は、『Sybase SQL Reference』マニュアルを参照してください。

Sybase データタイプのサポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[アクセスファイル](#)を参照してください。

可変長データタイプのマッピングの制御

Sybase ASE の場合 SET パラメータの VARCHAR では、Sybase のデータタイプ VARCHAR、VARCHAR2、および NVARCHAR2 のマッピングを制御します。デフォルト設定では、サーバはこれらのデータタイプを可変長の文字 (AnV) としてマッピングします。

Sybase IQ の場合 SET パラメータの VARCHAR では、Sybase のデータタイプ VARCHAR および VARBINARY のマッピングを制御します。デフォルト設定では、サーバはこれらのデータタイプを可変長の文字 (AnV) としてマッピングします。

Sybase ASE - 可変長データタイプのマッピングの制御

下表は、VARCHAR 値に基づくデータタイプのマッピングを示しています。

Sybase ASE データタイプ	備考	VARCHAR ON		VARCHAR OFF	
		USAGE	ACTUAL	USAGE	ACTUAL
VARCHAR (n)	n は 1 から 255 までの整数です。	AnV	AnV	An	An
NVARCHAR	n は 1 から 255 ま	AnV	AnV	An	An

Sybase ASE データタイプ	備考	VARCHAR ON		VARCHAR OFF	
		USAGE	ACTUAL	USAGE	ACTUAL
(<i>n</i>)	での整数です。				
VARBINARY (<i>n</i>)	<i>n</i> は 1 から 255 ま での整数です (<i>m</i> = 2 * <i>n</i>)。	<i>AmV</i>	<i>AmV</i>	<i>Am</i>	<i>Am</i>

可変長データタイプのマッピングの制御 (Sybase ASE)

```
ENGINE SQLSYB SET VARCHAR {ON|OFF}
```

説明

SQLSYB

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

Sybase のデータタイプ VARCHAR、VARCHAR2、および NVARCHAR2 を可変長の文字データタイプ (*AnV*) としてマッピングします。Unicode 環境ではこれは必須です。デフォルト値は ON です。

OFF

Sybase のデータタイプ VARCHAR、VARCHAR2、および NVARCHAR2 を文字データタイプ (*A*) としてマッピングします。

Sybase IQ - 可変長データタイプのマッピングの制御

下表は、VARCHAR 値に基づくデータタイプのマッピングを示しています。

Sybase IQ データタイプ	備考	VARCHAR ON		VARCHAR OFF	
		USAGE	ACTUAL	USAGE	ACTUAL
VARCHAR (<i>n</i>)	<i>n</i> は 1 から 255 までの整数です。	<i>AnV</i>	<i>AnV</i>	<i>An</i>	<i>An</i>
VARBINARY (<i>n</i>)	<i>n</i> は 1 から 255 までの整数です。 $m = 2 * n$	<i>AmV</i>	<i>AmV</i>	<i>Am</i>	<i>Am</i>

可変長データタイプのマッピングの制御 (Sybase IQ)

```
ENGINE SQLSYB SET VARCHAR {ON|OFF}
```

説明

SQLSYB

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

Sybase のデータタイプ VARCHAR および VARBINARY を可変長の文字データタイプ (AnV) としてマッピングします。デフォルト値は ON です。

OFF

Sybase のデータタイプ VARCHAR および VARBINARY を文字データタイプ (A) としてマッピングします。

SQL 式内での末尾ブランク

TABLE アダプタ内の新しい SQL ジェネレータでは、デフォルトで文字列リテラル内の末尾ブランク、および数値リテラル内の小数部と指数の表記を含むリテラルコンテンツを保持します。これにより、生成される SQL に対する制御が向上します。

末尾ブランクが不要なケースでは、次の構文が下位互換性を確保するために使用できます。

```
ENGINE SQLSYB SET TRIM_LITERALS ON
```

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアドプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェイスで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

Sybase ストアドプロシジャに対するレポートの実行

SELECT ステートメントや TABLE コマンドなどのレポートツールを使用して、Sybase ストアドプロシジャを実行し、プロシジャの出力パラメータとアンサーセットに対するレポートを実行することができます。ストアドプロシジャの実行という方法には、次の利点があります。

- アンサーセット以外に、出力パラメータである OUT パラメータおよび OUT モードの INOUT パラメータを検索することができる。他の呼び出し方法では、アンサーセットのみが検索されます。
- TABLE および他のレポートツールを使用して、出力パラメータおよびアンサーセットの処理、フォーマット設定、表示を簡単に行うことができる。

ストアドプロシジャに対するレポートを実行するには、次の手順を実行します。

1. **シノニムの作成** ストアドプロシジャのアンサーセットのシノニムを作成します。詳細は、[ストアドプロシジャのシノニムの作成](#)を参照してください。
2. **レポートプロシジャの作成** レポートプロシジャを作成します。詳細は、[ストアドプロシジャに対するレポートの実行](#)を参照してください。
3. **レポートの実行** ストアドプロシジャを実行し、出力パラメータ (OUT および OUT モードの INOUT) およびレポートに指定されたアンサーセットフィールドに対するレポートを実行します。

注意： Sybase IQ データについては、ストアドプロシジャのサポートが SYBIQ 12.7 ESD #4 リリースで始まりました。

ストアドプロシジャのシノニムの作成

シノニムは、ストアドプロシジャのパラメータとアンサーセットを定義します。

アンサーセットの構造は、プロシジャの実行時に指定される入力パラメータ値により異なります。このため、プロシジャの実行時に指定される入力パラメータ値のセットごとに、個別のシノニムを作成する必要があります。たとえば、3つの異なる入力パラメータ値セットでストアドプロシジャを実行する場合、シノニムを3つ (値セットごとに1つ) 作成する必要があります。なお、特に明記していない限り、「入力パラメータ」は、IN パラメータおよび IN モードの INOUT パラメータを意味します。

例外 ユーザがプロシジャの内部ロジックを理解しており、入力パラメータの値の範囲とプロシジャによって返された各アンサーセットの構造の関係を把握している場合、各アンサーセットの構造に対応する単一のシノニムを作成して、各シノニムにアンサーセットの構造を返すために必要な入力パラメータの値のセットを指定することが可能です。

シノニムには、次のセグメントを含めます。

- **INPUT** - このセグメントは、IN パラメータおよび IN モードの INOUT パラメータを記述します。
IN パラメータおよび IN モードの INOUT パラメータがない場合、このセグメントはダミーフィールドを 1 つ記述します。
- **OUTPUT** - このセグメントは、OUT パラメータおよび OUT モードの INOUT パラメータを記述します。
OUT パラメータおよび OUT モードの INOUT パラメータがない場合、このセグメントは省略されます。
- **ANSWERSET n** - 各アンサーセットに 1 つ含まれます。
アンサーセットがない場合、このセグメントは省略されます。

ストアドプロシジャのシノニム作成パラメータ

パラメータ/作業	説明
オブジェクトタイプ	[ストアドプロシジャ] を選択します。
データベース	テーブルやオブジェクトの選択元となるデータベースを指定するには、次のいずれかを実行します。 • [デフォルトデータベース] を選択して、デフォルトのデータベースとして設定されているデータベースを使用します。 • [データベース] ドロップダウンリストからデータベースを選択します。ここには、現在の DBMS インスタンスにあるすべてのデータベースが表示されます。

パラメータ/作業	説明
	[デフォルトデータベース] が選択されている場合は、データベースを選択する前に、このチェックをオフにします。 このオプションは、Sybase ASE に適用されます。Sybase IQ には適用されません。
選択	プロシジャを選択します。次のシノニム作成ウィンドウの [値] テキストボックスで、プロシジャごとに固有の値を入力する必要があるため、1 回に選択できるプロシジャは、1 つだけです。
名前	シノニム名です。デフォルト値はストアードプロシジャの名前です。
アプリケーション	アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。
接頭語/接尾語	同一名のストアードプロシジャが存在する場合は、接頭語または接尾語を割り当てて、対応するシノニムを区別します。なお、シノニム名の最大文字数は 64 バイトです。 すべてのプロシジャに固有の名前が付いている場合は、[接頭語] および [接尾語] のテキストボックスは空白にしておきます。
既存のシノニムを上書きする	作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。
データタイプマッピングのカスタマイズ	データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。詳細は、Sybase データタイプのサポートを参照してください。

パラメータ/作業	説明
値	指定したプロシジャに表示されるパラメータすべてについて、このチェックボックスを選択します。 選択したプロシジャに入力パラメータ (IN パラメータ、IN モードの INOUT パラメータ) がある場合、それらのパラメータの入力が要求されます。ただし、[値] テキストボックスに明示的に入力する必要があるかどうかは、プロシジャのロジックおよび生成されるデータ構造に依存します。このため、パラメータボックスのチェックはオンにする必要がありますが、値の入力は不要な場合もあります。次のガイドラインに従います。 • 入力パラメータ値によりアンサーセットのデータ構造が変化する場合は、明示的な入力値 (によりシノニムを区別すること) が必要です。 • プロシジャの内部ロジックが分かっており、常に同一のデータ構造が生成されることが確定している場合は、明示的な入力値は必要ありません。この場合、生成する必要のあるシノニムは 1 つだけで、シノニム作成の目的では、[値] テキストボックスはブランクにすることができます。 [値] が必要な場合は、引用符を含めずに値を入力します。日付、日付時間、タイムスタンプのパラメータ値は、ISO の形式に従って入力します。明示的な値が必要なプロシジャの場合、入力パラメータにはプロシジャが実行される際に入力されるパラメータと同一の値を指定します。

ストアドプロシジャに対するレポートの実行

ストアドプロシジャのアンサーセットに対してレポートを実行する場合は、データベーステーブルに対してレポートを実行する際と同一の機能を使用することができます。

- **SQL SELECT ステートメント** 構文についての詳細は、[SELECT を使用したストアドプロシジャに対するレポートの実行](#)を参照してください。
- **TABLE および GRAPH コマンド** 構文についての詳細は、[TABLE コマンドを使用したストアドプロシジャに対するレポートの実行](#)を参照してください。

ストアドプロシジャのアンサーセットとの JOIN には、次の 2 つのファイルを使用します。

- **ホストファイル** OUTPUT および ANSWERSET セグメントのみです。
- **クロスリファレンス** ファイル内の INPUT セグメントのみです。

TABLE コマンドを使用したストアドプロシジャに対するレポートの実行

TABLE コマンドを使用してストアドプロシジャを実行するには、次の構文を使用します。

```
TABLE FILE synonym
PRINT [parameter [parameter] ... | *]
[IF in-parameter EQ value]
.
.
.
END
```

説明

synonym

実行するストアドプロシジャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアドプロシジャにパラメータを必要としない場合は、アスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

★

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

IF

IF または WHERE キーワードです。IN パラメータまたは IN モードの INOUT パラメータに値を渡す際に使用します。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

注意：アダプタが Unicode をサポートするよう構成されている場合は、IN パラメータの長さは 1000 バイト以内にする必要があります。

value

パラメータに渡す値です。

SELECT を使用したストアドプロシジャに対するレポートの実行

```
SQL
SELECT [parameter [,parameter] ... | *] FROM synonym
[WHERE in-parameter = value]
.
.
.
END
```

説明

synonym

実行するストアドプロシジャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアドプロシジャにパラメータを必要としない場合は、構文にアスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

★

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

WHERE

IN パラメータまたは IN モードの INOUT パラメータに値を指定する場合に使用します。

各パラメータの値は、それぞれ異なる行に指定する必要があります。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

value

パラメータに渡す値です。

Sybase 環境のカスタマイズ

Sybase アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。ここでは、カスタマイズオプションの概要について説明します。

NONBLOCK モードの有効化

Sybase アダプタには、NONBLOCK モードで呼び出しを実行する機能があります。デフォルト設定は、BLOCK モードです。

この機能により、アダプタはクライアントリクエストに応答して、エンジン処理の待ち状態からクエリを取り消すことができます。待機状態は、通常、SQL 解析中、アンサーセットの 1 行目がアダプタへ配信できる状態になる前、または別のアプリケーションによりロックされたオブジェクトへのアクセスの待機中に起こります。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

NONBLOCK モードの有効化

```
ENGINE SQLSYB SET NONBLOCK {0|n}
```

説明

SQLSYB

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

正の数値です。デフォルト値はゼロ (0) です。これは、アダプタが BLOCK モードで動作することを意味します。数値が 1 以上の場合、NONBLOCK の呼び出しが有効になります。この数値は、アダプタが次の項目を確認する際の待機時間間隔 (秒) を指定します。

- 検索が実行されたか。
- クライアントアプリケーションが検索のキャンセルを要求したか。
- WebFOCUS Reporting Server ブラウザインターフェースの [セッションの終了] ボタンが押されたか。

注意：通常操作には、値に 1 または 2 を指定します。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

```
ENGINE SQLSYB SET PASSRECS {ON|OFF}
```

説明

SQLSYB

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

Sybase ASE および IQ でのパスワード暗号化の有効化

アダプタが Sybase ASE または Sybase IQ データソースに接続するときのパスワードの暗号化を制御することができます。デフォルト設定では、パスワードの暗号化は無効になっています。PASSWORD_ENCRYPTION 設定を有効にすると、Sybase アダプタサーバによって確立された、その後のすべての接続が影響を受けます。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定

の変更] を選択します。[設定の変更] ウィンドウが表示されたら、PASSWORD_ENCRYPTION オプションの隣にあるプルダウンリストから [OFF]、[ON]、または [EXTENDED] を選択します。

パスワード暗号化の有効化 (ASE および IQ)

```
SQL SQLSYB SET PASSWORD_ENCRYPTION {ON|EXTENDED|OFF}
```

説明

ON

接続プロパティの CS_SEC_ENCRYPTION を設定します。この設定は、Sybase ASE データソースへの接続に対する構成だけを行う場合に使用します。

EXTENDED

接続プロパティの CS_SEC_EXTENDED_ENCRYPTION を設定します。この設定は、Sybase IQ データソースへの接続のみを構成する場合、または Sybase ASE と Sybase IQ の両方のデータソースへの接続を構成する合に使用します。

- プロパティの CS_SEC_EXTENDED_ENCRYPTION が Sybase Open Client バージョン 15 ESD #7 以降から使用できるようになりました。この値が特定のクライアントのバージョン (つまり、Sybase Open Client バージョン 15 ESD #7 よりも前のクライアントのバージョン) で使用できない場合、設定時および接続の試行時にエラーメッセージの FOC1811 が生成されます。ユーザの Open Client バージョンが 15 ESD #7 より古い場合に発生する可能性があるその他のエラーには、「SQLCODE IS 5; ct_con_props(): User API layer: external error: An illegal value of 9213 given for parameter property」などがあります。
- Sybase IQ では、CS_SEC_EXTENDED_ENCRYPTION のみがサポートされます。IQ サーバへの接続が値 ON で試みられると、結果的に Sybase のエラーメッセージ SQLCODE IS 4002; ASA Error -103: Invalid user ID or password が生成されます。

OFF

パスワードの暗号化は無効になります。OFF がデフォルト値です。

Sybase 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)および[定数として定義された一時項目 \(DEFINE\) を転送するリクエストの最適化](#)を参照してください。

検索処理でのブロックサイズの指定

Sybase アダプタでは、SELECT クエリまたはストアドプロシジャの実行結果からの配列検索がサポートされます。この手法により、ネットワークおよび CPU 使用の負荷を大幅に軽減することができます。

大きい値を指定すると仮想記憶域のコストは増えますが、大量の行を処理するリクエストの効率は改善されます。ただし、100 を超える値を指定しても僅かな効率改善しか期待できないため、100 以内の値を設定することをお勧めします。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

配列検索のブロックサイズの指定

SELECT リクエストのブロックサイズは、TABLE FILE リクエスト、MODIFY リクエスト、MATCH リクエスト、および DIRECT SQL SELECT ステートメントに適用されます。

```
ENGINE SQLSYB SET FETCHSIZE n
```

説明

SQLSYB

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

配列検索手法を使用して一度に取得する行数です。受容値は 1 から 5000 です。アダプタに応じてデフォルト値は異なります。CLOB または BLOB として処理する必要のあるフィールドが結果セットに含まれている場合は、その結果セットに使用する FETCHSIZE 値は 1 です。

挿入処理のブロックサイズの指定

INSERT のブロックサイズは、LOADONLY と組み合わせて MODIFY INCLUDE リクエストに適用されます。INSERTSIZE は、パラメータ化 DIRECT SQL INSERT ステートメントでもサポートされます。

```
ENGINE SQLSYB SET INSERTSIZE n
```

説明

SQLSYB

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

配列検索手法を使用して挿入する行数です。受容値は 1 から 5000 です。BLOB として処理する必要のあるフィールドが結果セットに含まれている場合は、その結果セットに使用する INSERTSIZE 値は 1 です。

バルクインサート API の省略

```
ENGINE SQLSYB SET FASTLOAD [ON|OFF]
```

説明

SQLSYB

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

バルクインサート API を使用します。デフォルト値は ON です。

OFF

バルクインサート API の使用を抑制します。

バルクインサート API の動作

ibi Data Migrator で Sybase のバルクインサート API を使用することができます。

Sybase アダプタでは、INSERTSIZE の値が 1 より大きい場合、LOADONLY モードでバルクインサート API が使用されます。INSERTSIZE に基づいて、中間的なデータフラッシュの実行頻度が決定されます。測定値によると、Sybase サーバの最適なパフォーマンスのためには中間的なフラッシュが必要であることが示されています。経験上、INSERTSIZE 値には、数千から数万の値を設定することをお勧めします。中間的なフラッシュをロールバックすることはできません。

バルクロード API を抑制した場合、Sybase アダプタでは指定された INSERTSIZE に従って配列挿入を実行します。

ロード中に発生するエラー (例、重複エラー) により、行のバッチ全体が拒否される場合があります。

SQL パススルーを使用した Sybase のストアドプロシージャの呼び出し

Sybase のストアドプロシージャでは、SQL パススルーの使用がサポートされます。これらのプロシージャは、Sybase 内で CREATE PROCEDURE コマンドを使用して開発する必要があります。Sybase のストアドプロシージャは、連鎖または非連鎖のトランザクションモードで実行されます。デフォルトの CHAINED プロパティを変更する場合は、[非連鎖の Sybase ストアドプロシージャの実行](#)を参照してください。

Sybase ストアドプロシジャの呼び出し

ストアドプロシジャを呼び出す構文は、以下のとおりです。これまでサポートされてきた CALLSYB 構文ではなく、以下の構文を使用することをお勧めします。

```
ENGINE SQLSYB
EX SAMPLE PARM1,PARM2,PARM3...;
TABLE ON TABLE PCHOLD
END
```

説明

SQLSYB

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

Sybase のストアドプロシジャ

```
CREATE PROCEDURE SAMPLE
AS
SELECT SSN5, LAST_NAME5, FIRST_NAME5, BIRTHDATE5, SEX5 FROM EDAQA.NF29005
go
exec sp_procxmode 'PROC1','anymode'
go
```

非連鎖の Sybase ストアドプロシジャの実行

Sybase ASE のストアドプロシジャは、連鎖または非連鎖のトランザクションモードで実行されます。

デフォルト設定では、Sybase ストアドプロシジャは連鎖トランザクションモードで実行されます。非連鎖の Sybase のストアドプロシジャを実行する場合は、まず以下のコマンドを使用してアダプタの CHAINED プロパティを OFF に設定する必要があります。

```
ENGINE SQLSYB SET CHAINED {OFF|ON};  
END
```

説明

SQLSYB

アダプタの ASE を指定します。この値は、SET SQLENGINE コマンドを発行済みの場合省略できます。

OFF

アダプタを非連鎖トランザクションモードに設定します。

ON

アダプタを連鎖トランザクションモードに設定します。

アダプタは、デフォルト設定では連鎖トランザクションモードです。

このコマンドは、サーバプロシジャ (focexec) またはサーバプロファイル (edasprof) で発行することができます。

Teradata アダプタの使用

Teradata アダプタの構成方法について説明します。

Teradata アダプタを使用すると、アプリケーションから Teradata データソースへのアクセスが可能になります。このアダプタにより、アプリケーションリクエストがネイティブの Teradata ステートメントに変換され、最適化されたアンサーセットがリクエストの送信元アプリケーションに返されます。

Teradata 環境の準備

ODBC または CLI の Teradata アダプタを使用するには、Teradata Client コンポーネントをインストールして構成する必要があります。詳細は、『Teradata Tools and Utilities Installation Guide』を参照してください。TeraGSS および ICU のほか、ODBC または CLI が必要です。サーバを開始する前に、ODBC または CLI ライブラリディレクトリへのパスを SYSTEM LIBRARY PATH に追加しておく必要があります。

WebFOCUS Reporting Server ブラウザインターフェースを使用して、Teradata アダプタを構成します。

環境を ODBC または CLI を使用して Microsoft Windows 上に設定するには

Microsoft Windows では、Teradata 環境は製品のインストール時に設定されます。

UNIX で環境を設定するには

オプションの UNIX 環境変数 ODBCINI を使用して、Teradata にアクセスすることができます。ODBCINI 変数は、.odbc.ini ファイルの絶対パスを指定しています。

以下はその例です。

```
ODBCINI=/usr/odbc/.odbc.ini  
export ODBCINI
```

注意：\$HOME ディレクトリに .odbc.ini ファイルが格納されている場合、ODBCINI を使用することはできません。

Teradata CLI および ODBC アダプタの Unicode 用の構成

Teradata CLI および ODBC アダプタは、次の場合に UTF8 フォーマットをサポートします。

使用する Teradata システムで多言語サポートが有効になっていることを確認するには、データベース管理者 (DBA) に問い合わせるか、多言語文字セットのサポートに関するマニュアルを参照してください。

なお、現在のバージョンでは、Unicode を有効にしている場合、Teradata カラム名およびタイトルの最大長は 21 文字です。

Teradata アダプタの構成

アダプタの構成では、確立する接続ごとに接続と認証の情報を指定します。

接続属性の宣言

アダプタは、Teradata データベースサーバに接続する際に、接続情報および認証情報を必要とします。この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。選択したプロファイルにコマンドが追加されます。選択可能なプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) があります。

- グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) のいずれかに手動でコマンドを追加します。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の Teradata データベースサーバへの接続を宣言することができます。Teradata データベースサーバへの実際の接続は、その接続を参照する最初のクエリの発行時に確立されます。複数の SET CONNECTION_ATTRIBUTES コマンドを実行する場合、次のように動作します。

- 最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続が、デフォルト接続として機能します。
- 複数の SET CONNECTION_ATTRIBUTES コマンドに同一の接続名が含まれている場合、最後の SET CONNECTION_ATTRIBUTES コマンドで指定された属性が使用されます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。

4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Teradata の接続属性

Teradata アダプタは、[SQL] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

WebFOCUS Reporting Server ブラウザインターフェース上の [DBMS Add] と [DBMS Change] のウィンドウのリリース番号は、Teradata DBMS のリリースではなく、Teradata ODBC ドライバを指していることに注意してください。

接続名

この接続属性のセットの識別に使用される論理名です。

データソースまたはサーバ

有効な Teradata データソース名 (DSN) です。デフォルトの DSN はありません。つまり、値を入力する必要があります。

ODBC インターフェース この DSN 名は、Windows では ODBC アドミニストレータで構成されたユーザ DSN、システム DSN、またはファイル DSN と一致させる必要があります。UNIX では \$HOME/.odbc.ini ファイルの DSN エントリと一致させる必要があります。

CLI インタフェース このフィールドのラベルは SERVER で、有効な Teradata TDP 値を指定する必要があります。Windows および UNIX では、clispb.dat ファイルの i_dbcpath 値です。デフォルト値は [DBC] または [<default>] です。この値は、TD サーバ

のホストファイルエントリから取得することもできます。たとえば、ホストファイルに td13cop1 が追加されている場合、サーバ値は td13 になります。詳細は、『Teradata Tools and Utilities』を参照してください。

セキュリティ

データベースサーバへの接続時にユーザを認証する方法には、次の 2 つがあります。

- **Explicit** ユーザ ID およびパスワードを接続ごとに明示的に指定します。この情報がデータベースへの接続時に認証情報として渡されます。
- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報としてデータベースに渡されます。

ユーザ

データソースに登録されたプライマリ認証 ID です。

パスワード

プライマリ認証 ID に関連付けられたパスワードです。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

接続属性の手動宣言

```
ENGINE SQLDBC SET CONNECTION_ATTRIBUTES {connection DSN_name|Server}  
/userid,password
```

説明

SQLDBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

この接続属性のセットの識別に使用される論理名です。

connection と *DSN_name* の間にスペースが 1 つ必要です。

DSN_name

Teradata ODBC アダプタ アクセスする Teradata のデータソース名 (DSN) です。この名前は、odbc.ini ファイルのエントリに一致させる必要があります。

サーバ

Teradata CLI アダプタ Teradata Director Program 番号 (TDPn) です。ここで、n はその番号です。これは、clispb.dat ファイル内の i_dbcpath の値です。

userid

Teradata に登録されたプライマリ認証 ID です。

password

プライマリ認証 ID に関連するパスワードです。

接続属性の宣言

以下の SET CONNECTION_ATTRIBUTES コマンドでは、明示的なユーザ ID (MYUSER) およびパスワード (PASS) を使用して、SAMPLESERVER という名前の Teradata データベースサーバに接続します。パスワードの安全性を確保するために、WebFOCUS Reporting Server ブラウザインターフェースで接続属性を指定すると、パスワードが暗号化された上でサーバプロファイルに追加されます。

```
ENGINE SQLDBC SET CONNECTION_ATTRIBUTES CON1 SAMPLESERVER/MYUSER,PASS
```

Teradata 用の接続情報の更新

このアダプタの CONNECTION_ATTRIBUTES コマンドの構文が拡張され、アプリケーションの開発環境から実稼働環境への移植をサポートするよう設計された論理接続名を指定できるようになりました。この拡張された構文では、既存の CONNECTION_ATTRIBUTES コマンドのマイグレートが必要になる場合があります。

WebFOCUS Reporting Server ブラウザインターフェースの [マイグレート] オプションを使用すると、現在のサーバ設定が新しいバージョンにマイグレートされます。このオプションにアクセスするには、[ツール] メニューの [ワークスペース] を選択し、[設定] から [マイグレート] を選択します。[マイグレート] ウィンドウで、構成インスタンスディレクトリ (EDACONF) の完全なパスを入力して [続行] ボタンをクリックします。この方法を使用することをお勧めします。

[マイグレート] オプションを使用しない場合、以下の情報に注意してください。

- 宣言されている接続名
- 新規のシノニムを作成する目的で新しい接続を追加する場合、既存の接続は新しいフォーマットで再保存され、また既存のシノニムは変更を行うことなく引き続き機能します。
- 既存のシノニムを使用する目的で新しい接続を追加する場合、デフォルトの論理接続名を、既存のアクセスファイルの CONNECTION=value 属性で指定された値に一致するように変更する必要があります。

たとえば、バージョン 7.6 SP01 より前のバージョンで、以下のように接続が定義されている場合を想定します。

```
ENGINE SQLDBC SET CONNECTION_ATTRIBUTES DSN_A/uid,pwd
```

この DSN_A に保存されているオブジェクトに基づいてシノニムを作成すると、アクセスファイルには以下の記述が追加されます。

```
CONNECTION=DSN_A
```

この後、新しい接続を追加する際に、この既存のシノニムを再利用するには、接続名をデフォルトの CON01 から DSN_A に変更し、DSN_A として保存する必要があります。この接続は、以下のようにプロファイルに保存されます。

```
ENGINE SQLDBC SET CONNECTION_ATTRIBUTES DSN_A DSN_A/uid,pwd
```

デフォルト接続の上書き変更

複数の接続を定義した場合、最初の SET CONNECTION_ATTRIBUTES コマンドで指定された接続がデフォルト接続として機能します。このデフォルト値は、SET DEFAULT_CONNECTION コマンドを使用して上書きすることができます。

デフォルト接続の変更

```
ENGINE SQLDBC SET DEFAULT_CONNECTION connection
```

説明

SQLDBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

connection

以前に SET CONNECTION_ATTRIBUTES コマンドを発行して定義された接続です。この名前が前に宣言されていなかった場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

注意

- SET DEFAULT_CONNECTION コマンドを繰り返し使用する場合は、最後のコマンドで指定した接続名がデフォルト接続になります。
- 非コミットランザクション (LUW) が保留の間は、SET DEFAULT_CONNECTION コマンドを発行することはできません。この場合は、次のメッセージが出力されます。

(FOC1671) コマンドの順序に誤りがあります。

デフォルト接続の選択

次の SET DEFAULT_CONNECTION コマンドにより、SAMPLE という名前のデータベースサーバがデフォルトのデータベースサーバとして選択されます。

```
ENGINE SQLDBC SET DEFAULT_CONNECTION SAMPLE
```

接続範囲の制御

確立する接続ごとにアダプタを使用する場合、SET AUTODISCONNECT コマンドを使用して接続の継続を制御することができます。

接続範囲の制御

```
ENGINE SQLDBC SET AUTODISCONNECT ON {FIN|COMMAND|COMMIT}
```

説明

SQLDBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

FIN

セッションが終了した後に限り、自動的に切断します。デフォルト値は、FIN です。

COMMAND

各リクエストの後に自動的に切断されます。SET AUTODISCONNECT コマンドは、イベントがどの程度の頻度で発生するかによって、大幅なオーバーヘッドをもたらす可能性があります。このオーバーヘッドのほとんどすべてがサーバには関連しません。このオーバーヘッドは、オペレーティングシステムとデータソースに関連します。

COMMIT

COMMIT または ROLLBACK を固有の SQL コマンドとして実行した後に限り、自動的に切断します。

Teradata メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとにシノニムを作成して、データソースの構造と Teradata のデータタイプのサーバマッピングを記述します。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な Teradata テーブルまたはビューの一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは、一時項目や追加のセキュリティメカニズムなど、サーバの拡張メタデータ属性のサポートを提供します。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、結果としてマスターファイルとアクセスファイルが生成されます。これらはサーバのメタデータを表します。

ストアドプロシジャのシノニムの作成、およびストアドプロシジャに対するレポートの実行についての詳細は、[ストアドプロシジャのシノニムの作成](#)を参照してください。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ペー

ジが表示されます。

- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Teradata のシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

選択したオブジェクトタイプ (テーブル、ビュー、その他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。選択したオブジェクトタイプ (テーブル、ビュー、外部 SQL スクリプト、他のサポートされるオブジェクト) に基づいて、作成するシノニムの候補を制限します。

ドロップダウンリストから [外部 SQL スクリプト] を選択すると、SQL クエリを読み取り専用レポートのシノニムとして指定することができます。シノニム候補としては、有効な SQL クエリが 1 つだけ格納された任意のファイルを使用することができます。このファイルにステートメントの終了を示す区切り文字 (セミコロン (;) またはスラッシュ (/)) やコメントを含めることはできません。

アダプタによっては、表示されるオブジェクトのチェックボックスをオンにすることで、検索をさらに絞り込むことができます。

外部 SQL スクリプトのパス

[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合は、さらに次の項目が表示されます。

次の標準名前規則は UNIX に適用されます。

- [ベースパス] テキストボックスに、SQL クエリが記述されたファイルの物理ディレクトリパスを指定します。ディレクトリ名を直接入力するか、[...] (参照) ボタンをクリックしてファイルの格納先ディレクトリへ移動します。このクリックにより、[ベースパスの選択] ダイアログボックスが表示されます。
- [ドキュメント名] テキストボックスに、ファイル名を入力します。必要に応じて、ワイルドカード文字を使用します。
- [ドキュメント拡張子] テキストボックスに、スクリプトファイルの拡張子を入力して、その拡張子のファイルのみを表示します。

アダプタはシノニムの作成中にネイティブ API コールを発行して選択リスト内の要素のリストを取得し、各要素に対するフィールドを持つマスターファイルを構成します。生成されたアクセスファイルは、DATASET 属性の SQL スクリプトパスを参照します。これには SQL クエリが格納されたファイルの名前と拡張子を含む、フルパスが記述されています。以下はその例です。

```
DATASET=/u1/home2/apps/report3.sql
```

WebFOCUS レポートを作成する際は、SQL クエリを使用してデータにアクセスします。

カーディナリティ

[カーディナリティ] のチェックをオンにすると、メタデータ作成時に、テーブルの現在のカーディナリティ (行またはタプルの数) を考慮します。カーディナリティは、等結合 (EQUIJOIN) で使用されます。検索順序はテーブルのサイズ (カーディナリティ) により決定されます。サイズの小さいテーブルから読み込まれます。

アプリケーションで使用するテーブルのカーディナリティが動的の場合、この設定を選択しても意味がありません。

外部キーを使用したクラスタ作成 (非推奨。[シノニムの編集] の使用を推奨)

[外部キーを使用したクラスタ作成] のチェックをオンにすると、現在のテーブルに外部キーで関係付けられているテーブルすべてをこのシノニムに含めることができます。ただし、このオプションは廃止される予定で、クラスタの作成にはシノニムエディタを使用することをお勧めします。生成されるマルチテーブルのシノニムは、このテーブルの外部キーの関係をすべて定義します。

サブクエリ

このオプションは、[オブジェクトタイプ] ドロップダウンリストから [外部 SQL スクリプト] を選択した場合にのみ選択可能になります。このオプションを選択すると、生成されたシノニムのアクセスファイルに SUBQUERY キーワードが追加されます。対応する SQL 文字列に、生成される SQL クエリの FROM ステートメント (抽出テーブルと呼ばれる) で使用可能な有効な構文が含まれている場合、SQL SCRIPT が FROM 句に埋め込まれたサブクエリとして処理されます。この方法では柔軟性が向上します。たとえば、このシノニムを JOIN のターゲットとして使用することができます。

SQL SCRIPT に疑問符 (?) やコロン (:) などのパラメータマーカが含まれている場合、または抽出テーブルでは無効なコンストラクトが構文に含まれている場合 (例、ORDER BY)、このキーワードは選択しないでください。実行時に、SUBQUERY=Y が存在し、SQL SCRIPT を FROM ステートメントで使用できないことが特定された場合、この設定は無視され、FOC1782 警告メッセージが発行されます。デフォルト設定では、このオプションは選択されています (SUBQUERY=Y)。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[Teradata データタイプのサポート](#)を参照してください。

メタデータの更新または作成

[作成] を選択すると、既存シノニムが同一の完全修飾名で存在する場合に、その既存シノニムが上書きされます。[更新] を選択すると、メタデータが既存シノニムに同期されます。[更新] を選択すると、次の画面が開き、DBMS カタログから取得された属性のリストが表示されます。このリストで各属性のチェックをオンにすることで、既存シノニムの属性を DBMS カタログの属性で上書きすることができます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

オーナー/スキーマ

ユーザが所有する 1 つまたは複数のオブジェクトを作成したユーザアカウントです。

テーブル名

基になるオブジェクトの名前です。

タイプ

オブジェクトタイプ (テーブル、ビューなど)。

テーブルの選択

シノニムを作成するテーブルを選択します。

- リスト内のテーブルをすべて選択するには、[すべて選択] のチェックをオンにします。
- 特定のテーブルまたはビューを選択するには、目的のテーブルまたはビューのチェックをオンにします。

生成されたシノニムの例

Teradata アダプタのシノニムは、マスターファイルとアクセスファイルから構成されます。次の例は、nf29004 テーブルのシノニムを示しています。

生成されたマスターファイル nf29004.mas

```
FILE=DIVISION, SUFFIX=SQLDBC , $  
SEGNAME=SEG1_4, SEGTYPE=S0 , $  
FIELD=DIVISION4, DIVISION4, I9, I4, MISSING=OFF , $  
FIELD=DIVISION_NA4, DIVISION_NA4, A25, A25, MISSING=ON , $  
FIELD=DIVISION_HE4, DIVISION_HE4, I9, I4, MISSING=ON , $
```

生成されたアクセスファイル nf29004.acx

```
SEGNAME=SEG1_4, TABLENAME=EDAQA.NF29004,  
CONNECTION=DSN_A, KEYS=1, WRITE=YES, $
```

Teradata コメントのシノニムへのマッピング

Teradata テーブルのシノニムを生成する際、アダプタは次のマッピングを行います。

- Teradata のテーブルのコメントを Teradata データディクショナリからシノニムのマスターファイル内の Remark 属性へ。
- Teradata のカラムのコメントを Teradata データディクショナリからシノニムのマスターファイル内の Description 属性へ。

アクセスファイルのキーワード

下表は、アクセスファイルで使用されるキーワードを示しています。

キーワード	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
TABlename	テーブルまたはビューの名前です。この値には、以下のようにロケーションまたはオーナー名を含めることができます。 TABlename=[location.][owner.]tablename
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 CONNECTION=connection
KEYS	テーブルの主キーを構成するカラムの数を指定します。マスターファイルセグメントの最初の n フィールドに対応しています。 マスターファイルの先頭にフィールドを記述せずにキーフィールドを指定する方法についての詳細は、次の KEY 属性を参照してください。
KEY	マスターファイルの先頭にキーフィールドを記述せずに、主キーとして定義するフィールドを指定します。構文は次のとおりです。 KEY=fld1/fld2/.../fldn
WRITE	テーブルに対して書き込み操作が可能かどうかを指定します。

キーワード	説明
KEYFLD IXFLD	複数テーブルのマスターファイルによって確立される関係を実装する主キーと外部キーフィールドの名前を指定します。KEYFLD と IXFLD は共に、1 組の関連するテーブルが共有するフィールドを識別します。 - KEYFLD は、親テーブルの共通カラムの FIELDNAME です。 - IXFLD は関連するテーブルの共通カラムの FIELDNAME です。 KEYFLD と IXFLD のデータタイプが一致している必要があります。また、これらのカラムの長さも一致させることをお勧めしますが、必須ではありません。 注意：KEYFLD と IXFLD カラム両方に RDBMS インデックスがあると、RDBMS が生成する JOIN の効率が大幅に向上する可能性があります。これらのカラムのデータタイプが一致している必要があります。これらのカラムの長さが一致している場合、RDBMS では JOIN がより効率的に処理されます。
AUTO INCREMENT	自動インクリメントを有効にするには [Yes] に設定します。
START	インクリメントの順序の初期値です。
INCREMENT	インクリメントの間隔です。
INDEX_NAME INDEX_UNIQUE INDEX_ COLUMNS INDEX_ORDER	データベース内のインデックス名、およびインデックスフィールドの一意性、名前、順序を示します。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

Teradata データタイプのサポート

WebFOCUS Reporting Server ブラウザインターフェースでは、SQL データタイプのマッピングオプションをレポートに表示することができます。

詳細は、[アクセスファイル](#)を参照してください。

可変長データタイプのマッピングの制御

SET パラメータの VARCHAR では、Teradata のデータタイプ VARCHAR のマッピングが制御されます。デフォルト設定では、サーバはこのデータタイプを可変長の文字 (AnV) としてマッピングします。

下表は、VARCHAR 値に基づくデータタイプのマッピングを示しています。

Teradata データタイプ	備考	VARCHAR ON		VARCHAR OFF	
		USAGE	ACTUAL	USAGE	ACTUAL
VARCHAR (n)	n は 1 から 4000 までの整数です。	AnV	AnV	An	An

可変長データタイプのマッピングの制御

```
ENGINE SQLDBC SET VARCHAR {ON|OFF}
```


説明

SQLDBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

Teradata のデータタイプ VARCHAR を可変長の文字データタイプ (AnV) としてマッピングします。デフォルト値は ON です。

OFF

Teradata のデータタイプ VARCHAR を文字データタイプ (A) としてマッピングします。

SQL 式内での末尾ブランク

TABLE アダプタ内の新しい SQL ジェネレータでは、デフォルトで文字列リテラル内の末尾ブランク、および数値リテラル内の小数部と指数の表記を含むリテラルコンテンツを保持します。これにより、生成される SQL に対する制御が向上します。

末尾ブランクが不要なケースでは、次の構文が下位互換性を確保するために使用できます。

```
ENGINE SQLDBC SET TRIM_LITERALS ON
```

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアードプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェイスで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

Teradata ストアドプロシジャに対するレポートの実行

SELECT ステートメントや TABLE コマンドなどのレポートツールを使用して、Teradata ストアドプロシジャを実行し、プロシジャの出力パラメータとアンサーセットに対するレポートを実行することができます。ストアドプロシジャの実行という方法には、次の利点があります。

- アンサーセット以外に、出力パラメータである OUT パラメータおよび OUT モードの INOUT パラメータを検索することができる。他の呼び出し方法では、アンサーセットのみが検索されます。
- TABLE および他のレポートツールを使用して、出力パラメータおよびアンサーセットの処理、フォーマット設定、表示を簡単に行うことができる。

ストアドプロシジャに対するレポートを実行するには、次の手順を実行します。

1. **シノニムの作成** ストアドプロシジャのアンサーセットのシノニムを作成します。詳細は、[ストアドプロシジャのシノニムの作成](#)を参照してください。
2. **レポートプロシジャの作成** レポートプロシジャを作成します。詳細は、[ストアドプロシジャに対するレポートの実行](#)を参照してください。
3. **レポートの実行** ストアドプロシジャを実行し、出力パラメータ (OUT および OUT モードの INOUT) およびレポートに指定されたアンサーセットフィールドに対するレポートを実行します。

ストアドプロシジャのシノニムの作成

シノニムは、ストアドプロシジャのパラメータとアンサーセットを記述します。

アンサーセットの構造は、プロシジャの実行時に指定される入力パラメータ値により異なります。このため、プロシジャの実行時に指定される入力パラメータ値のセットごとに、個別のシノニムを作成する必要があります。たとえば、3つの異なる入力パラメータ値セットでストアドプロシジャを実行する場合、シノニムを3つ (値セットごとに1つ) 作成する必要があります。なお、特に明記していない限り、「入力パラメータ」は、IN パラメータおよび IN モードの INOUT パラメータを意味します。

例外 プロシジャの内部ロジックが分かっており、プロシジャから返される各アンサーセット構造を生成する入力パラメータ値の範囲に確信がある場合、アンサーセット構造ごとにシ

ノニムを 1 つ作成し、個々のシノニムにアンサーセット構造を返すために必要な入力パラメータ値の見本のセットを指定します。

シノニムには、次のセグメントを含めます。

- **INPUT** - このセグメントは、IN パラメータおよび IN モードの INOUT パラメータを記述します。
IN パラメータおよび IN モードの INOUT パラメータがない場合、このセグメントはダミーフィールドを 1 つ記述します。
- **OUTPUT** - このセグメントは、OUT パラメータおよび OUT モードの INOUT パラメータを記述します。
OUT パラメータおよび OUT モードの INOUT パラメータがない場合、このセグメントは省略されます。
- **ANSWERSET n** - 各アンサーセットに 1 つ含まれます。
アンサーセットがない場合、このセグメントは省略されます。

ストアドプロシジャのシノニム作成パラメータ

以下のリストは、値の指定が可能なシノニム作成パラメータを示しています。

オブジェクトタイプ

[ストアドプロシジャ] を選択します。

選択

プロシジャを選択します。次のシノニム作成ウィンドウの [値] テキストボックスで、プロシジャごとに固有の値を入力する必要があるため、1 回に選択できるプロシジャは、1 つだけです。

名前

シノニム名です。デフォルト設定は、ストアドプロシジャ名です。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のストアドプロシジャが存在する場合は、接頭語または接尾語を割り当てて、対応するシノニムを区別します。なお、シノニム名の最大文字数は 64 バイトです。

すべてのプロシジャに固有の名前が付いている場合は、[接頭語] および [接尾語] のテキストボックスは空白にしておきます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

データタイプマッピングのカスタマイズ

データタイプマッピングのデフォルト設定を変更するには、このセクションを展開します。カスタマイズ可能なマッピングが表示されます。

カスタマイズ可能なマッピングについての詳細は、[Teradata データタイプのサポート](#)を参照してください。

値

指定したプロシジャに表示されるパラメータすべてについて、このチェックボックスを選択します。

選択したプロシジャに入力パラメータ (IN パラメータ、IN モードの INOUT パラメータ) がある場合、それらのパラメータの入力が要求されます。ただし、[値] テキストボックスに明示的に入力する必要があるかどうかは、プロシジャのロジックおよび生成されるデータ構造に依存します。このため、パラメータボックスのチェックはオンにする必要はありますが、値の入力は不要な場合もあります。次のガイドラインに従います。

- 入力パラメータ値によりアンサーセットのデータ構造が変化する場合は、明示的な入力値 (によりシノニムを区別すること) が必要です。
- プロシジャの内部ロジックが分かっており、常に同一のデータ構造が生成されることが確定している場合は、明示的な入力値は必要ありません。この場合、生成する必要のあるシノニムは 1 つだけで、シノニム作成の目的では、[値] テキストボックスは空白にすることができます。

[値] が必要な場合は、引用符を含めずに値を入力します。日付、日付時間、タイムスタンプのパラメータ値は、ISO の形式に従って入力します。明示的な値が必要なプロシジャの

場合、入力パラメータにはプロシジャが実行される際に入力されるパラメータと同一の値を指定します。

ストアドプロシジャに対するレポートの実行

ストアドプロシジャのアンサーセットに対してレポートを実行する場合は、データベーステーブルに対してレポートを実行する際と同一の機能を使用することができます。

- **SQL SELECT ステートメント** 構文についての詳細は、[SELECT を使用したストアドプロシジャに対するレポートの実行](#)を参照してください。
- **TABLE および GRAPH コマンド** 構文についての詳細は、[TABLE コマンドを使用したストアドプロシジャに対するレポートの実行](#)を参照してください。

ストアドプロシジャのアンサーセットとの JOIN には、次の 2 つのファイルを使用します。

- **ホストファイル** OUTPUT および ANSWERSET セグメントのみです。
- **クロスリファレンス** ファイル内の INPUT セグメントのみです。

TABLE コマンドを使用したストアドプロシジャに対するレポートの実行

TABLE コマンドを使用してストアドプロシジャを実行するには、次の構文を使用します。

```
TABLE FILE synonym
PRINT [parameter [parameter] ... | *]
[IF in-parameter EQ value]
.
.
.
END
```

説明

synonym

実行するストアドプロシジャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアドプロシジャにパラメータを必要としない場合は、アスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

*

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

IF

IF または WHERE キーワードです。IN パラメータまたは IN モードの INOUT パラメータに値を渡す際に使用します。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

注意：アダプタが Unicode をサポートするよう構成されている場合は、IN パラメータの長さは 1000 バイト以内にする必要があります。

value

パラメータに渡す値です。

SELECT を使用したストアドプロシジャに対するレポートの実行

```
SQL
SELECT [parameter [,parameter] ... | *] FROM synonym
[WHERE in-parameter = value]
.
```

·
END

説明

synonym

実行するストアドプロシジャのシノニムです。

parameter

パラメータの名前です。このパラメータの値をレポートに表示します。入力パラメータ、出力パラメータ、入出力パラメータのいずれかを指定することができます。

ストアドプロシジャにパラメータを必要としない場合は、構文にアスタリスク (*) を入力します。この場合、SELECT ステートメントの構造に合わせて、シノニムの生成時に作成されたダミーセグメントが表示されます。

★

指定したすべてのパラメータを表示すること、または必須パラメータがないことを表します。

WHERE

IN パラメータまたは IN モードの INOUT パラメータに値を指定する場合に使用します。
各パラメータの値は、それぞれ異なる行に指定する必要があります。

in-parameter

IN パラメータの名前、または IN モードの INOUT パラメータの名前です。このパラメータに値を渡します。

value

パラメータに渡す値です。

Teradata 環境のカスタマイズ

Teradata アダプタには、環境のカスタマイズおよびパフォーマンスを最適化するためのパラメータが用意されています。ここでは、カスタマイズオプションの概要について説明します。

リクエストのカラム見出しの制御

レポートのカラム見出しは、WebFOCUS Reporting Server ブラウザインターフェースを使用して指定することができます。また、SET COLNAME コマンドを発行することもできます。

コマンドによるカラム見出しの制御

```
ENGINE SQLDBC SET COLNAME {NAME|TITLE}
```

説明

SQLDBC

アダプタを指定します。このパラメータは、SET SQLENGINE コマンドを発行済みの場合省略できます。

Name

NAME オプションを指定した場合、Teradata のカラム名がカラム見出しとして使用されます。

TITLE

TITLE オプションを指定した場合、指定した名前がカラム見出しとして使用されます。TITLE がデフォルト値です。

NONBLOCK モードの有効化

Teradata アダプタには、NONBLOCK モードで呼び出しを実行する機能があります。デフォルト設定は、BLOCK モードです。

この機能により、アダプタはクライアントリクエストに応答して、エンジン処理の待ち状態からクエリを取り消すことができます。待機状態は、通常、SQL 解析中、アンサーセットの 1 行目がアダプタへ配信できる状態になる前、または別のアプリケーションによりロックされたオブジェクトへのアクセスの待機中に起こります。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取

得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

NONBLOCK モードの有効化

```
ENGINE SQLDBC SET NONBLOCK {0|n}
```

説明

SQLDBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

正の数値です。デフォルト値はゼロ (0) です。これは、アダプタが BLOCK モードで動作することを意味します。数値が 1 以上の場合、NONBLOCK の呼び出しが有効になります。この数値は、アダプタが次の項目を確認する際の待機時間間隔 (秒) を指定します。

- 検索が実行されたか。
- クライアントアプリケーションが検索のキャンセルを要求したか。
- WebFOCUS Reporting Server ブラウザインターフェースの [セッションの終了] ボタンが押されたか。

注意：通常の操作には、値に 1 または 2 を指定します。

更新または削除された行数の取得

PASSRECS は正常に実行された SQL パススルーの INSERT、UPDATE、DELETE コマンドにより影響を受ける行の数を返します。

ヒント：この設定は手動で変更することができますが、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。その場合は、メニューバーの [データの取得] をクリックした後、構成済みアダプタを右クリックし、コンテキストメニューから [設定の変更] を選択します。[設定の変更] ウィンドウが表示されます。

更新または削除された行数の取得

```
ENGINE SQLDBC SET PASSRECS {ON|OFF}
```

説明

SQLDBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した後、アプリケーションプログラム SCB のカウントメンバーで影響を受ける行の数を取得します。デフォルト値は ON です。

OFF

SQL パススルーの INSERT、UPDATE、DELETE コマンドが正常終了した場合でも、情報を取得しません。

Teradata 接続内でのトランザクションモードの設定

Teradata の接続のトランザクションモードは、WebFOCUS Reporting Server ブラウザインターフェースを使用するか、コマンドを発行して制御することができます。

Teradata 接続のトランザクションモードの設定

Teradata 接続のトランザクションモードを設定するには、プロファイルまたはプロシジャで以下のコマンドを発行します。

```
SQL SQLDBC SET TRANSACTION [ANSI|BTET|DEFAULT]
```

説明

ANSI

Teradata 接続を ANSI トランザクションモードに設定します。CLI 接続では、これがデフォルト値です。

BTET

Teradata 接続を Teradata トランザクションモード (BTET トランザクションモードとも呼ばれる) に設定します。

DEFAULT

ODBC 接続の場合のみ Teradata 接続を Teradata システムのデフォルトのトランザクションモードに設定します。ODBC 接続では、これがデフォルト値です。

Teradata システムのデフォルトの決定方法の詳細は、Teradata のマニュアルを参照してください。

注意：ODBC および CLI 接続でトランザクションモードを設定する際のそれぞれの接続範囲は異なります。

- **ODBC 接続** トランザクションモードは、接続中に設定することができます。接続を切断する必要はありません。
- **CLI 接続** トランザクションモードは、接続する確立する前にのみ設定できます。CLI 接続でトランザクションモードを変更する場合は、最初に SQL SQLDBC END SESSION コマンドを使用して、接続を終了する必要があります。この操作を実行しない場合は、次の警告メッセージが返されます。

(FOC1722) 警告 : %1%2 のトランザクションモードが設定できません。

Teradata 最適化の設定

アダプタの最適化機能を使用すると、RDBMS での処理の実行が最適化され、RDBMS とサーバ間の通信量が減少し、応答時間が改善されます。また、RDBMS 独自の内部最適化機能を有効にすることも可能です。

詳細は、[リクエストの最適化](#)および[定数として定義された一時項目 \(DEFINE\) を転送するリクエストの最適化](#)を参照してください。

挿入処理のブロックサイズの指定

Teradata アダプタでは、Teradata Client が以下のバージョンの場合に配列挿入がサポートされます。

この手法により、ネットワークおよび CPU 使用の負荷を大幅に軽減することができます。

大きい値を指定すると仮想記憶域のコストは増えますが、大量の行を処理するリクエストの効率は改善されます。

ヒント：この設定は手動で変更することができます。

挿入処理のブロックサイズの指定

INSERT のブロックサイズは、LOADONLY と組み合わせて MODIFY INCLUDE リクエストに適用されます。INSERTSIZE は、パラメータ化 DIRECT SQL INSERT ステートメントでもサポートされます。

```
ENGINE SQLDBC SET INSERTSIZE n
```

説明

SQLDBC

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

n

配列検索手法を使用して挿入する行数です。受容値は 1 から 5,000 です。デフォルト値は 1 です。LOB として処理する必要があるフィールドが結果セットに含まれている場合は、その結果セットに使用する INSERTSIZE 値は 1 です。

集計の最適化による効率の改善

集計の最適化により、クエリの効率が大幅に改善されます。

注意：Teradata では、集計の最適化は内部的に処理されます。アダプタの設定は必要ありません。ただし、Teradata 内で COLLECT STATISTICS コマンドを発行する必要があります。

非等価条件 WHERE ベース LEFT OUTER JOIN の最適化

LEFT OUTER JOIN は、ホストテーブルからすべてのレコードを選択し、それらをクロスリファレンステーブルのレコードと照合します。一致するレコードが存在しない場合、ホストレコードは保持されますが、クロスリファレンスフィールドにはデフォルト値 (ブランクまたはゼロ) が割り当てられます。アダプタは、RDBMS で条件式がサポートされるすべての WHERE ベースの LEFT OUTER JOIN コマンドを最適化することができます。

条件付き LEFT OUTER JOIN の指定

```
JOIN LEFT_OUTER FILE hostfile AT hfld1 [TAG tag1]
    [WITH hfld2]
    TO {UNIQUE|MULTIPLE}
    FILE crfld AT crfld [TAG tag2] [AS joinname]
    [WHERE expression1;
    [WHERE expression2;
    ...]

END
```

説明

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。

hostfile

ホストマスタファイルです。

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 パラメータにフィールド値を指定しても、この値が接続情報として使用されることはありません。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

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に一意の修飾子として使用するタグ名 (オプション) です。

WITH 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 に基づく条件付き JOIN では、KEEPDEFINES を ON に設定し、一時項目 (DEFINE) を JOIN コマンドを発行する前に作成しておく必要があります。

MULTIPLE

from_file と *to_file* 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。なお、ONE は UNIQUE の同義語です。

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。

ユニーク JOIN は WebFOCUS の概念です。RDBMS はユニークと非ユニークを区別しないため、クロスリファレンスファイルから条件に一致するすべての行を取得します。

たとえば、リクエストでユニーク JOIN として指定したが、実際には条件に一致する行がクロスリファレンスファイルに複数存在する場合、この JOIN の処理を RDBMS が行くと、RDBMS は一致するすべての行を返します。一方、最適化を無効にしてこの JOIN の処理を WebFOCUS が行くと、異なるレポート結果が得られます。WebFOCUS はユニーク JOIN の概念を順守するため、ホストの 1 行に対してクロスリファレンスの行を 1 行だけ返します。

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データソース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。

WHERE ベース OUTER JOIN の最適化条件

- WHERE ベースの LEFT OUTER JOIN を最適化するには、それに関係する RDBMS で式の最適化を行うことができるとともに、次の条件のいずれかを満たす必要があります。
 - JOIN WHERE コマンドに、少なくとも 1 つの「field1 EQ field2」述語が含まれている。ここで、field1 は table1 のフィールド、field2 は table2 のフィールドを表します。
または
 - 右側のテーブルに、NULL データを含まないユニークインデックスまたはキーがある。
または
 - 右側のテーブルに、長いデータタイプ (例、TEXT、IMAGE) を持たない「NOT NULL」フィールドが少なくとも 1 つ含まれている。
- アダプタの SQLJOIN OUTER 設定は、ON (デフォルト) に設定しておく必要があります。

非等価条件 LEFT OUTER JOIN の最適化

次のリクエストは、2つの Teradata のデータソース間の条件付き LEFT OUTER JOIN を作成し、結合後のデータソースを使用してレポートを作成します。このリクエストに対して生成された SQL を表示するため、STMTRACE がオンに設定されています。

```
SET TRACEUSER = ON
SET TRACEOFF = ALL
SET TRACEON = STMTRACE//CLIENT
JOIN LEFT_OUTER FILE baseapp/EQUIP AT CARS
TO ALL FILE baseapp/CARREC AT CARC
WHERE CARS NE CARC;
END
TABLE FILE baseapp/EQUIP
PRINT CARS CARC STANDARD
BY MODEL
END
```

WebFOCUS リクエストは、LEFT OUTER JOIN を取り込む単一の Teradata の SELECT ステートメントに変換され、非等価条件が ON 句で RDBMS に渡されます。

```
SELECT T1."CARS"(CHAR( 16)),T1."STANDARD"(CHAR( 40)),
T2."CARC"(CHAR( 16)),T2."MODEL"(CHAR( 24)) FROM ( EQUIP T1 LEFT
OUTER JOIN CARREC T2 ON (T1."CARS" <> T2."CARC") ) ORDER BY
T2."MODEL";
```

SQL パススルーを使用した Teradata マクロまたはストアドプロシージャの呼び出し

Teradata のマクロとストアドプロシージャでは SQL パススルーがサポートされます。

マクロは、Teradata 内で CREATE または REPLACE MACRO コマンドを使用して作成する必要があります。プロシージャは、Teradata 内で CREATE PROCEDURE コマンドを使用して開発する必要があります。

マクロの呼び出しは、マクロのコンパイルに使用したトランザクションモードで行う必要があります。トランザクションモードの設定についての詳細は、[Teradata 接続内でのトランザクションモードの設定](#)を参照してください。現在有効なトランザクションモードを確認するには、HELP セッションコマンドを発行します。


```
SQL SQLDBC HELP SESSION;
TABLE FILE SQLOUT PRINT TRANSACTION_SEMANRICS
END
```

ストアドプロシジャまたはマクロを呼び出すには、WebFOCUS Reporting Server ブラウザ インターフェースを使用するか、SET MACRO コマンドを発行して、接続を設定しておく必要があります。

```
SQL SQLDBC SET MACRO ON|OFF
```

説明

ON

マクロを呼び出すものを有効にします。これがデフォルト値です。

OFF

ストアドプロシジャを呼び出すものを有効にします。

マクロの呼び出し

以下は、マクロを呼び出すための構文の例です。

```
ENGINE SQLDBC
EX SAMPLE PARM1,PARM2,PARM3...;
TABLE FILE SQLOUT
END
```

ストアドプロシジャの呼び出し

ストアドプロシジャを呼び出す構文は、以下のとおりです。

```
ENGINE SQLDBC
EX SAMPLE PARM1,PARM2,PARM3...;
TABLE FILE SQLOUT
END
```

使用するアダプタに応じて、パラメータのサポートが異なります。

- **ODBC** スカラパラメータ (IN、OUT、INOUT) がすべてサポートされます。
- **CLI** スカラパラメータ IN、および IN モードの INOUT パラメータがサポートされます。

Web サービスアダプタの使用

Web サービスアダプタの構成方法について説明します。

Web サービスアダプタを使用することにより、クライアントアプリケーションから Web サービスプロバイダへのアクセスが可能になります。アダプタは次の変換を行います。

- 内部の DML (データ処理言語) リクエストまたは SQL リクエストを Web サービスリクエストに変換する。
- Web サービスのレスポンスをアンサーセットに変換する。

Web サービスアダプタの構成

アダプタを構成する場合、少なくとも 1 つの接続に対して接続情報および認証情報を指定します。

接続属性の宣言

ターゲット Web サービスをホストする Web サービスプロバイダにアクセスするには、アダプタに接続情報を指定する必要があります。この情報を指定するには、SET CONNECTION_ATTRIBUTES コマンドを使用します。この場合、次のいずれかの手順を実行することができます。

- WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの構成画面で、接続および認証情報を入力します。選択したプロファイルにコマンドが追加されます。選択可能なプロファイルには、グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) があります。
- グローバルプロファイル (edasprof.prf)、ユーザプロファイル (user.prf)、グループプロファイル (使用プラットフォームでサポートされている場合) のいずれかに手動でコマンドを追加します。

複数の SET CONNECTION_ATTRIBUTES コマンドを発行することで、複数の Web サービスプロバイダへの接続を宣言することができます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインターフェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

Web サービスの接続属性

Web サービスアダプタは、[プロシジャ] グループフォルダ下にあります。

以下のリストは、値の指定が可能な接続属性を示しています。属性の宣言を完了するには、[構成] ボタンをクリックします。

接続名

この接続属性のセットの識別に使用される論理名です。デフォルト値は CON01 です。

WSDL URL

Web サービスを記述する WSDL の URL です。次のいずれかをアドレスの先頭に使用することができます。

file://

http://

https://

この値は、シノニムの作成でのみ必要です。それ以外の場合、この値は無視されます。

プロキシサーバ IP アドレス

リクエストを代行受信し、実サーバに転送するプロキシサーバの IP アドレスです。

HTTPS を有効にする

次のセキュアな HTTP 接続オプションから、いずれかを選択します。

- **なし** HTTPS を有効にしません。これがデフォルト値です。
- **Windows** Windows の HTTPS 接続を有効にします。
- **SSL** SSL (セキュアソケットレイヤ) の HTTPS 接続を有効にします。このオプションを選択すると、構成が必要な追加の SSL フィールドが表示されます。

プロキシポート番号

プロキシサーバのリスナポート番号です。デフォルトポート番号は 80 です。

KERBEROS SPN

[セキュリティ] を [Trusted] に設定した場合、クライアントがサービスのインスタンスを一意に識別するためのサービスプリンシパル名を指定します。以下はその例です。

```
mydaemon/foo:4761
```

SSL 証明書

ローカルに格納されるユーザ指定の SSL 認証用のサーバの x.509 証明書の場所です。証明書ファイルは、アダプタが接続するサーバの認証に使用されます。

SSL 相互認証

このオプションを選択すると、相互認証が有効になります。

SSL 証明書タイプ

使用する SSL 証明書のタイプです。次のオプションがあります。

- **Trusted** [Trusted] では、認証済み証明書ファイル (trustedcertfile) で、次に示すように PEM フォーマットの CA 証明書のファイルを指示します。

```
-----BEGIN CERTIFICATE-----  
...(CA certificate in base64 encoding) ...  
-----END CERTIFICATE-----
```

認証済み証明書ファイルには、複数の CA 証明書を格納することができます。証明書の前後または証明書の間に、テキストを追加することができます。たとえば、証明書の説明を提供するテキストなどです。

- **Trusted 以外** キーファイル、パスフレーズ、ラベルを構成する必要があります。

SSL 証明書キーファイル

クライアントの X.509 証明書を PEM フォーマットで作成するために使用されるプライベートキーです。このオプションは、Trusted 以外の接続用の証明書とともに使用されます。

SSL 証明書パスフレーズ

キーファイルのロックを解除するために使用されるパスワードです。この値は、キーファイルが暗号化される場合にのみ必要です。

SSL 証明書ラベル

ファイルに複数の証明書が含まれている場合に、ファイル内の証明書を識別します。

セキュリティ

Web サービスプロバイダへの接続時にユーザを認証する方法には、次の 3 つがあります。デフォルト値は [なし] です。この場合、ユーザは認証されません。

- **Explicit** ユーザ ID とパスワードは接続ごとに明示的に指定します。これらの値が

接続時の認証情報として Web サービスに渡されます。

- **Password Passthru** クライアントアプリケーションから受け取ったユーザ ID とパスワードが、接続時に認証情報として Web サービスに渡されます。
- **Trusted** アダプタは、現在のオペレーティングシステムに関連している偽装プロセス用のデータベースの規則を使用してデータベースに接続します。

ユーザ

[セキュリティ] を [Explicit] に設定した場合に表示されます。Web サービスにログインするプライマリ認証 ID です。

パスワード

[セキュリティ] を [Explicit] に設定した場合に表示されます。プライマリ認証 ID に関連付けられたパスワードです。

連鎖認証

選択すると、連鎖認証が有効になります。認証作業の連鎖実行は、Web サーバが Cookie またはトークンを返し、後続の処理作業にその Cookie またはトークンを使用する場合に使用します。

シノニムのアプリケーション

このオプションは、連鎖認証にのみ使用します。この処理についての詳細は、[連鎖実行用のアダプタを構成するには](#)を参照してください。

プロファイルの選択

ドロップダウンリストからプロファイルのいずれかを選択し、CONNECTION_ATTRIBUTES コマンドを格納するプロファイルのレベルを指定します。デフォルトのプロファイルは、グローバルプロファイルの edasprof.prf です。

新しいユーザプロファイル (user.prf) またはグループプロファイル (プラットフォームで使用可能な場合) を作成するには、[プロファイルの選択] ドロップダウンリストボックスにプロファイル名を直接入力します (拡張子は自動的に追加されます)。

接続属性をサーバプロファイル (edasprof) に保存します。

エンドポイント

このオプションは、[次へ] をクリックした後の 2 つ目の構成ウィンドウに表示されます。

ドロップダウンリストから、Web サービスに使用するエンドポイント URL を選択します。

接続属性の手動宣言

ユーザ ID とパスワードは接続ごとに明示的に指定します。これらの値が接続時の認証情報として Web サービスに渡されます。構文は次のとおりです。

```
ENGINE SOAP SET CONNECTION_ATTRIBUTES connection/  
[user[,pswd]]:'endpoint_URL WSDL_URL [AUTH:applid/mf] [fieldname="value" ...]  
[PROXYS:proxy_IP_address] [PROXYP:{80|port_number}]  
[CERTIFICATE:cert_location [cert_key=key_file] [cert_phrase=pass_phrase]  
[cert_label= cert_label]]'
```

説明

SOAP

アダプタを指定します。

connection

一連の属性を識別する論理名です。

user,pswd

この値を指定すると、AUTH パラメータで定義されたユーザ ID とパスワードが接続時の認証作業に渡されます。

連鎖認証では、認証マスターファイルのヘッダまたはグループ説明のいずれかで定義されている場合、これらの項目への入力には必須になります。詳細は、[連鎖実行用のアダプタを構成するには](#)を参照してください。

endpoint_URL

Web サービスプロバイダの URL です。

WSDL_URL

Web サービスを記述する WSDL の URL です。この URL を使用してシノニムを作成します。

applid/mf

連鎖認証では、関連する認証作業を記述したマスターファイルが格納されたサーバの場所です。詳細は、[連鎖実行用のアダプタを構成するには](#)を参照してください。

fieldname=value

認証作業に追加パラメータを渡す連鎖認証では、fieldname はマスターファイルの ACCESS_PROPERTY パラメータの NEED_VALUE 値で定義したフィールドに対応します。ただし、予約語である USERNAME および PASSWORD のフィールドは除きます。

詳細は、[認証作業への追加パラメータの受け渡し](#)を参照してください。

注意：

- SET CONNECTION_ATTRIBUTES コマンドで値を指定すると、マスターファイルの XDEFAULT パラメータで設定したデフォルト値が上書きされます。
- 項目値は二重引用符 (") で囲む必要があります。項目値に二重引用符 (") および一重引用符 (') を含めることはできません。

proxys_IP_address

プロキシサーバの IP アドレスです。このサーバがリクエストを代行受信して、実サーバに転送します。

port_number

プロキシサーバのリスナポート番号です。デフォルトポート番号は 80 です。

cert_location

ローカルに格納されるユーザ指定の SSL 認証用サーバ x.509 証明書 (トラステッド CA) の場所です。証明書ファイルは、アダプタが接続するサーバの認証に使用されます。

認証済み証明書ファイル (trustedcertfile) には、次のように PEM フォーマットの CA 証明書のファイルが指定されます。

```
-----BEGIN CERTIFICATE-----
...(CA certificate in base64 encoding) ...
-----END CERTIFICATE-----
```

認証済み証明書ファイルには、複数の CA 証明書を格納することができます。証明書の前後または証明書の間に、テキストを追加することができます。たとえば、証明書の説明を提供するテキストなどです。

Trusted 以外の証明書を指定する場合は、以下の追加情報を入力します。

```
[cert_key=key_file] [cert_phrase=pass_phrase] [cert_label=cert_label]
```

key_file

クライアントの X.509 証明書を PEM フォーマットで作成するために使用されるプライベートキーです。このエントリは、Trusted 以外の接続に必須です。

pass_phrase

キーファイルのロックを解除するために使用されるパスワードです。このエントリは、キーファイルが暗号化される場合にのみ必要です。

cert_label

ファイルに複数の証明書が含まれている場合に、ファイル内の証明書を識別します。

接続属性の宣言

次の例では、SET CONNECTION_ATTRIBUTES コマンドを使用して、アプリケーションから「SAMPLEAPP」という Web アプリケーションへのアクセスを確立します。

```
ENGINE SOAP SET CONNECTION_ATTRIBUTES esriR/45678,4456789DFAFC:
'http://arcweb.esri.com/services/v2/AddressFinder
file://D:\%users%\WSDL\%ESRI\AddressFinder.wsd AUTH:ESRI/gettoken'
```

Web サービスのセキュリティ

認証作業の連鎖実行は、Web サーバが Cookie またはトークンを返し、後続の処理作業にその Cookie またはトークンを使用する場合に使用します。連鎖実行は、リクエストされた処理に関連する接続情報で定義されます。Cookie を使用する場合は、認証および実行の両方の接続情報に同一のエンドポイント URL を指定する必要があります。

Web サービスアダプタは、次のセキュリティ機能をサポートします。

- **Cookie** ユーザの認証情報を格納します。この情報は TSCOM エージェントセッションの期間 (ユーザが接続してから切断するまでの間) に限り有効です。Cookie の有効範囲は単一の TSCOM エージェントです。受信した Cookie は、後続の SOAP リクエストとともに送信元の IP へ返信されます。
- **認証トークン** 認証作業から返される値です。この値は、関連する実行作業に使用されます。

連鎖認証では、Explicit および Passthru (PING 機能の使用) セキュリティモデルがサポートされます。

連鎖認証を使用するには、認証接続を宣言する必要があります。次に、認証情報を記述したマスターファイルを作成し、そのファイルをアプリケーションに保存します。続いて、認証シノニムを使用して、関連する 1 つまたは複数の実行接続情報を作成します。詳細は、[連鎖実行用のアダプタを構成するには](#)を参照してください。

なお、接続情報にユーザ ID とパスワードの情報は含まれていても、連鎖認証が定義されていない場合は、ベーシック HTTP 認証が使用されます。この場合、ユーザ ID とパスワードは x64 アルゴリズムで暗号化された後、HTTP 接続の確立に使用されます。

連鎖実行用のアダプタを構成するには

アダプタを連鎖実行用に構成するには、次の手順を実行します。

1. 認証作業を記述する WSDL の URL を使用して、接続情報を作成します。この接続は、認証のエンドポイント URL の指定に必要です。

この手順を実行するには、ibi Data Migrator ブラウザインターフェースまたは WebFOCUS Reporting Server デスクトップインターフェースの [接続パラメータ] ウィンドウで、WSDL の URL を入力し、[次へ] をクリックした後、ドロップダウンリストからエンドポイント URL を選択します。

なお、後続の処理作業にトークンではなく Cookie を使用する場合は、この認証接続情報と手順 3 で指定する実行接続情報には同一のエンドポイント URL を指定する必要があります。

2. 関連する認証作業のシノニムを作成します。

この手順を実行するには、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースでシノニムを作成後、シノニムエディタを開き、次のガイドラインに従ってシノニムを編集します。

認証シノニムには、次の入力項目を含める必要があります。

USERNAME

ユーザ ID の値は、実行作業を定義した接続情報から取得されます。

PASSWORD

パスワードの値は、実行作業を定義した接続情報から取得されます。

認証シノニムには、次のアクセスプロパティの項目を含めることもできます。

次の項目は、SOAP リクエストを記述するセグメントに含める必要があります。

ACCESS_PROPERTY=NEED_VALUE

認証作業に使用する追加パラメータの項目を定義します。デフォルト値は、XDEFAULT パラメータで指定することができます。関連する接続情報に値を指定した場合、デフォルト値はその値で上書きされます。

複数の項目を定義することができます。

次の項目は、SOAP レスポンスを記述するセグメントに含める必要があります。

ACCESS_PROPERTY=AUTHRESP

認証作業の結果を記述する項目を定義します。正しいレスポンス値を ACCEPT 属性に入力します。複数值を許可する場合は、OR により記述します。

複数の項目を定義することができます。複数の項目を定義することができます。ただし、これらの項目のいずれかに受容不可の値が含まれている場合、この認証作業は無効と見なされます。

ACCESS_PROPERTY=AUTHTOKEN

レスポンストークンを格納する項目を定義します。このトークンは、入力値として実行作業に渡されます。指定可能な項目は 1 つです。この項目を定義しない場合、この認証作業には Cookie が返されるものと見なされます。

3. 実行リクエストの処理作業とエンドポイント、および作成済み認証作業シノニムを記述した WSDL の URL を使用して接続情報を作成します。

この手順を実行するには、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの [接続パラメータ] ウィンドウで WSDL の URL を入力し、[マスターファイルから/アプリケーションを選択] ドロップダウンリストから、関連する認証作業を記述したマスターファイルのサーバの場所を選択します。[次へ] をクリックし、ドロップダウンリストからエンドポイント URL を選択します。

4. 関連する実行作業のシノニムを作成します。

WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースで、シノニムを作成します。

5. 実行作業シノニムには、ACCESS_PROPERTY=AUTHTOKEN を使用して認証トークンを定義した入力項目を含めます (トークンを使用する場合)。

この手順を実行するには、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースでシノニムエディタを開き、シノニムを編集します。

詳細は、[Web サービスの接続属性](#)および [シノニムの作成](#)を参照してください。

連鎖認証の構成

接続情報

```
ENGINE SOAP SET CONNECTION_ATTRIBUTES esriA/, :
'https://arcweb.esri.com/services/v2/Authentication
file://D:¥users¥WSDL¥ESRI¥Authentication.wsdl'
```

```
ENGINE SOAP SET CONNECTION_ATTRIBUTES esriR/109618,8366C1649D9DFAFC:
'http://arcweb.esri.com/services/v2/AddressFinder
file://D:¥users¥WSDL¥ESRI¥AddressFinder.wsdl AUTH:ESRI/gettoken'
```

認証作業を記述したマスターファイルの抜粋

```
FILENAME=GETTOKEN, SUFFIX=SOAP , $
SEGMENT=GETTOKEN, SEGTYPE=S0, $
FIELDNAME=USERNAME, ALIAS=username, USAGE=A30, ACTUAL=A30,
ACCESS_PROPERTY=(NEED_VALUE), $
FIELDNAME=PASSWORD, ALIAS=password, USAGE=A30, ACTUAL=A30,
ACCESS_PROPERTY=(NEED_VALUE), $
FIELDNAME=__RESPONSE, USAGE=TX80L, ACTUAL=TX,
ACCESS_PROPERTY=(INTERNAL), $
SEGMENT=RESPONSE, SEGTYPE=S0, SEGSUF=XML , PARENT=GETTOKEN,
POSITION=__RESPONSE, $
FIELDNAME=RESPONSE, ALIAS=getToken00out, USAGE=A1, ACTUAL=A1,
ACCESS_PROPERTY=(INTERNAL), $
FIELDNAME=RESULT, ALIAS=Result, USAGE=A120, ACTUAL=A120,
ACCESS_PROPERTY=(AUTHTOKEN),
REFERENCE=RESPONSE, PROPERTY=ELEMENT, $
```

データ抽出作業を記述したマスターファイルの抜粋

```
FILENAME=FINDADDRESS, SUFFIX=SOAP , $
SEGMENT=FINDADDRESS, SEGTYPE=S0, $
GROUP=ADDRESS, ALIAS=address, USAGE=A210, ACTUAL=A210, $
FIELDNAME=HOUSENUMBER, ALIAS=houseNumber, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='2815', $
FIELDNAME=STREET, ALIAS=street, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='PRAIRIE AVE.', $
FIELDNAME=INTERSECTION, ALIAS=intersection, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
FIELDNAME=CITY, ALIAS=city, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='MIAMI BEACH', $
```

```

FIELDNAME=STATE_PROV, ALIAS=state_prov, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='FL', $
FIELDNAME=ZONE, ALIAS=zone, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='33140', $
FIELDNAME=COUNTRY, ALIAS=country, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='US', $

```

```

GROUP=ADDRESSFINDEROPTIONS, ALIAS=addressFinderOptions, USAGE=A30,
ACTUAL=A30, $
FIELDNAME=DATASOURCE, ALIAS=dataSource, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='GDT.Streets.US', $
FIELDNAME=TOKEN, ALIAS=token, USAGE=A120, ACTUAL=A120,
ACCESS_PROPERTY=(NEED_VALUE, AUTHTOKEN), $
FIELDNAME=__RESPONSE, USAGE=TX80L, ACTUAL=TX,
ACCESS_PROPERTY=(INTERNAL), $
SEGMENT=RESPONSE, SEGTYPE=S0, SEGSUF=XML, PARENT=FINDDADDRESS,
POSITION=__RESPONSE, $
FIELDNAME=RESPONSE, ALIAS=findAddressResponse, USAGE=A1, ACTUAL=A1,
ACCESS_PROPERTY=(INTERNAL), $

```

認証作業への追加パラメータの受け渡し

認証作業に一連の追加パラメータを渡すには、マスターファイルに XDEFAULT パラメータを追加して、デフォルト値を設定します。

ヒント：WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのシノニムエディタを使用してこのタスクを実行することができます。

```

ENGINE SOAP SET CONNECTION_ATTRIBUTES belgR: 'https://secure.securex.be/
HRAWebService/webservices/WSEmployee.asmx https://secure.securex.be/
HRAWebService/webservices/WSEmployee.asmx?WSDL AUTH:belgium/
directauthenticate DBID="HRADemo01" LANGID="1" '

```

マスターファイルの抜粋

```

FILENAME=DIRECTAUTHENTICATE, SUFFIX=SOAP, $
SEGMENT=DIRECTAUTHENTICATE, SEGTYPE=S0, $

```

```
GROUP=DIRECTAUTHENTICATE, ALIAS=DirectAuthenticate, USAGE=A120,
  ACTUAL=A120, $
FIELDNAME=USERNAME, ALIAS=usrname, USAGE=A30, ACTUAL=A30,
  ACCESS_PROPERTY=(NEED_VALUE), $
```

```
FIELDNAME=PASSWORD, ALIAS=pwd, USAGE=A30, ACTUAL=A30,
  ACCESS_PROPERTY=(NEED_VALUE), $
```

```
FIELDNAME=DBID, ALIAS=dbId, USAGE=A30, ACTUAL=A30,
  ACCESS_PROPERTY=(NEED_VALUE), $
XDEFAULT='Test', $
```

```
FIELDNAME=LANGID, ALIAS=langId, USAGE=A30, ACTUAL=A30,
  ACCESS_PROPERTY=(NEED_VALUE), $
XDEFAULT='E', $
```

```
FIELDNAME=__RESPONSE, USAGE=TX80L, ACTUAL=TX,
  ACCESS_PROPERTY=(INTERNAL), $
```

```
SEGMENT=RESPONSE, SEGTYPE=S0, SEGSUF=XML , PARENT=DIRECTAUTHENTICATE,
  POSITION=__RESPONSE, $
```

```
FIELDNAME=RESPONSE, ALIAS=DirectAuthenticateResponse, USAGE=A1,
  ACTUAL=A1, ACCESS_PROPERTY=(INTERNAL), $
```

```
FIELDNAME=DIRECTAUTHENTICATERESULT, ALIAS=DirectAuthenticateResult,
  USAGE=A5, ACTUAL=A5, ACCESS_PROPERTY=(AUTHRESP), ACCEPT='true',
  REFERENCE=RESPONSE, PROPERTY=ELEMENT, $
```

Web サービスのメタデータの管理

サーバが Web サービスプロバイダにアクセスする場合、そのサーバは Web サービス処理にパラメータを渡す方法および Web サービス処理からレスポンスを受信する方法を認識しておく必要があります。サーバがアクセスする処理ごとに、処理およびサーバマッピングを記述したシノニムを作成します。これは、Web サービスの WSDL ファイルから提供されます。

シノニムの作成

シノニムは、Web サービス処理ごとに一意の論理名 (「エイリアス」とも呼ばれる) を定義します。シノニムの使用には次の利点があります。

- シノニムでは、クライアントアプリケーションをリクエストのロケーションと ID に対する変更から分離します。リクエストは、それを使用するクライアントアプリケーションを変更せずに、移動したり名前を変更したりできます。行う必要があるのは 1 つの変更だけです。つまり、サーバ上でリクエストのシノニムを再定義するだけで済みます。
- シノニムは、一時項目 (DEFINE) やセキュリティ機構などのサーバの拡張メタデータ機能をサポートします。

シノニムを作成すると、マスターファイルおよびアクセスファイルが生成されます。これらのファイルは、サーバへの Web サービスリクエストを記述したメタデータです。

1 つのシノニムが 1 つの処理を表します。

注意：各シノニムに、それぞれ特定のプロバイダの Web サービス処理を記述します。処理パラメータを変更した場合は、シノニムを再作成する必要があります。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。
2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。

- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

Web サービスのシノニム作成パラメータ

以下のリストは、アダプタのシノニムを作成する際に、値の指定が必要なパラメータ、および実行する関連作業を示しています。これらのオプションが複数のウィンドウに表示される場合もあります。ウィンドウに表示されたボタンをクリックして次のウィンドウに進み、最後に [シノニムの作成] ボタンをクリックします。これで、入力データに基づいてシノニムが作成されます。

名前でフィルタ

シノニムの作成に使用するオペレーションにフィルタを設定しない場合は、チェックをオフにします。

オペレーションにフィルタを設定する場合は、チェックをオンにします。オペレーション名の入力が必要です。オペレーション名のフィルタに使用する文字列を入力します。必要に応じて、文字列の先頭や末尾にワイルドカード文字 (%) を入力します。

たとえば、「ABC%」と入力して、名前が ABC で始まるオペレーションを選択します。また、「%ABC」と入力して、名前が ABC で終わるオペレーションを選択します。さらに、「%ABC%」と入力して、名前の任意の位置に ABC を含むオペレーションを選択します。

オペレーションの選択ボタン

このボタンをクリックして、すべてのオペレーションを選択するか、フィルタ後のオペレーションを選択します。

フィールド名の確認

すべての特殊文字をアンダースコア (_) に変換するとともに、予約語の使用を回避するために名前の確認を行う場合は、[フィールド名の確認] のチェックをオンにします。これにより、名前に番号が付加されます。このパラメータにより、規則に一致した名前が付けられます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

[フィールド名の確認] のチェックをオフにすると、ハイフン (-)、ブランク、円記号 (¥)、フォワードスラッシュ (/)、カンマ (,)、ドル記号 (\$) の特殊文字のみがアンダースコア (_) に変換されます。名前の確認は行われません。

ユニークフィールドにする

フィールド名およびグループ名の範囲をシノニム全体に設定する場合、[ユニークフィールドにする] のチェックをオンにします。これにより、シノニムの異なるセグメントであっても重複名は使用されません。このチェックをオフにすると、範囲はセグメントに設定されます。

Strict SOAP データタイプ

デフォルト設定では、テーブルリクエストから生成された SOAP リクエストは、選別条件の値が指定されていない要素タグをすべて省略します。この設定を使用する場合は、[Strict SOAP データタイプ] のチェックをオフにします。

値が指定された要素タグとブランクの要素タグの両方を渡す場合は、[Strict SOAP データタイプ] のチェックをオンにします。

このオプションは、SOAP リクエストのボディとヘッダの両方に適用されます。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスは空白にしておきます。

既存のシノニムを上書きする

作成するシノニムが既存シノニムの完全修飾名と重複する場合に、既存シノニムの上書きを指定することができます。その場合は、[既存のシノニムを上書きする] のチェックをオンにします。

注意：シノニムを再作成するには、接続ユーザがオペレーティングシステムの書き込み権限を所有している必要があります。

デフォルトシノニム名

この列には、各シノニムに割り当てる名前が表示されます。異なる名前を指定する場合は、表示されている名前を上書きします。

シノニムの作成

リスト内のすべてのオペレーション [デフォルトシノニム名] 列の左側に表示されたチェックをオンにします。

特定のオペレーション 各オペレーションに対応するチェックをオンにします。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

マスターファイルの属性

属性	説明
PROPERTY	フィールドが XML 属性または XML 要素に対応しているかどうかを示します。
REFERENCE	Web サービスの XML スキーマドキュメントによって定義された XML 階層内のフィールドの親要素を識別します。 フォーマットは、次のとおりです。 <i>segmentname.fieldname</i> 説明 segmentname このフィールドが存在するマスターファイルのセグメント名です。 fieldname 親要素に対応するフィールド名です。

アクセスファイルの属性

属性	説明
SEGNAME	この値は、マスターファイルの SEGNAME 値に一致させる必要があります。
CONNECTION	定義済みの接続属性のうち、どのデータベースサーバに接続するかを宣言します。構文は次のとおりです。 <pre>CONNECTION=connection</pre>

属性	説明
version	使用する Web サービスのバージョンです。このアダプタは、バージョン 1.1 をサポートします。
ACTION	記述された Web サービス処理です。
OBJECT	Web サービスリクエストの完全修飾名です。
ENCODING	エンコードスキーマの URL です。
TARGETNS	処理の WSDL 記述で指定されたターゲットネームスペースです。この場合、Web サービスの WSDL のグローバルネームスペース、またはこの処理の WSDL で一意のネームスペースです。
STYLE	Web サービスドキュメント WSDL のスタイル要素値 (RPC または Document) です。
HEADER	マスターファイルのヘッダグループです (存在する場合)。
HEADERNS	ヘッダのネームスペースです。
EMPTYTAGS	ON の場合は、値が指定された要素タグとともに空白の要素タグが渡されます。OFF (デフォルト) の場合は、値が指定された要素のみが渡されます。 SOAP リクエストのボディとヘッダの両方に適用されます。

静的 JOIN の使用

マスターファイルおよびアクセスファイルを使用して、同一の物理 XML ドキュメントのさまざまなビューを記述することができます。

- マスターファイルでは、フィールド定義に REFERENCE 属性を使用して、XML ドキュメント階層のタグと PARENT 属性との間の物理関係を表します。これにより、論理的な階層関係が確立されます。
- アクセスファイルでは、KEYFLD および IXFLD 属性を使用して、XML ドキュメント階層内の主キーまたは外部キーとして動作する XML タグを識別します。これにより、論理的な JOIN 関係が確立されます。KEYFLD で定義した親フィールド (外部キー) は、クロスリファレンスの値を提供します。IXFLD で定義した従属フィールド (主キー) には、それに対応する値が格納されます。

アダプタは、実行時に値が一致した JOIN を実装します。

静的 JOIN を使用すると、データノードを任意に組み合わせて、階層的に無関係な統合スキーマ ComplexType 定義の間に JOIN 関係を作成することができます。

これらの定義に属する任意の XML タグを使用して、JOIN の組み合わせを作成することができます。また、必要に応じて、同一物理セグメントのインスタンスを増加させることで、論理的な JOIN 関係を表すこともできます。

JOIN を使用することで、Web サービスを複数実行する方法もあります。ただし、ユーザは、Web サービスの入力メッセージを記述したフィールドのプロパティを変更することはできません。入力と出力の両方で受容可能なデータタイプは、元の WSDL により制御されます。JOIN は、ルートセグメントに属するフィールドに対して使用し、ACCESS_PROPERTY パラメータを NEED_VALUE に設定する必要があります。また、ループを回避するために、JOIN は一意にする必要があります。

注意：静的 (埋め込み) JOIN は、シノニムの作成機能で直接使用することはできません。この機能を利用するには、シノニムの作成後にそのマスターファイルおよびアクセスファイルを編集する必要があります。この編集を行う場合は、ibi Data Migrator デスクトップインターフェースのシノニムエディタを使用することをお勧めします。このツールを使用すると、簡単なドラッグアンドドロップ操作でセグメント階層を分かりやすく表示したり、KEYFLD および IXFLD を編集する計算を視覚的に実行したりすることができます。ibi Data Migrator デスクトップインターフェースを使用すると、すべての重複セグメントをマスターファイルに追加した後、不必要なセグメントを削除する操作をすばやく実行することができます。

場合によっては、JOIN の記述に適切な XML の制約 (unique/key/keyref) がスキーマに含まれる場合もあります (例、Microsoft ツールで作成され、元の Microsoft SQL Server データベーステーブル間の関係を表すスキーマ)。この情報を使用して、マスターファイルを次のように編集することができます。

- 要素「unique」および「key」は、要素 (セグメント) の主キーを定義します。
- 要素「keyref」は、要素 (セグメント) の外部キーを定義します。

ibi Data Migrator デスクトップインターフェースを使用してマスターファイルを編集する方法についての詳細は、[ibi Data Migrator デスクトップインターフェースのシノニムエディタによる静的 JOIN の追加](#)を参照してください。

ibi Data Migrator デスクトップインターフェースのシノニムエディタによる静的 JOIN の追加

この例は、「GetAllEmployees」という名前のシノニムを作成する WSDL コードの一部を示しています。

このスキーマは、EID フィールドをキーとする Employee セグメントおよび ERID フィールドをキーとする EmployeeRelative セグメントの制約関係 (Constraint) を示しています。

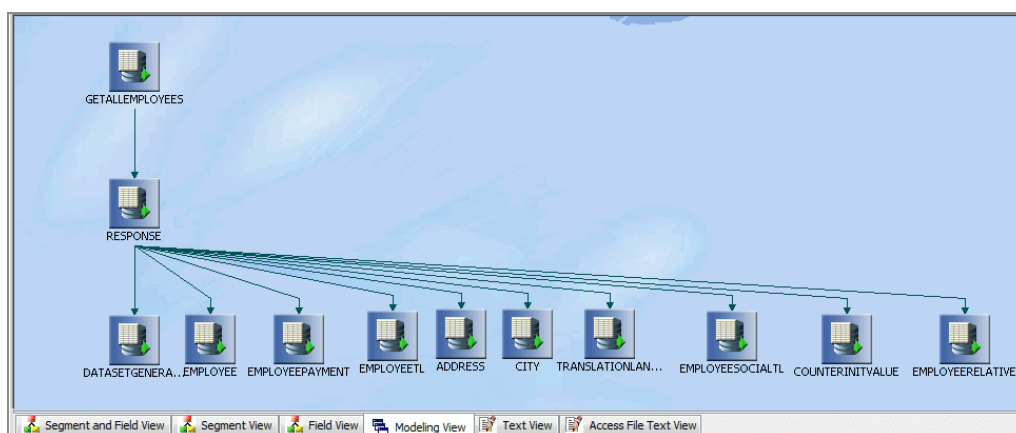
```
<xs:unique msdata:PrimaryKey="true" name="Constraint1">
  <xs:selector xpath="."//Employee" />
  <xs:field xpath="EId" />
</xs:unique>

<xs:unique msdata:ConstraintName="Constraint1" msdata:PrimaryKey="true"
  name="EmployeeRelative_Constraint1">
  <xs:selector xpath="."//EmployeeRelative" />
  <xs:field xpath="ErId" />
</xs:unique>
<xs:keyref msdata:ConstraintOnly="true"
  name="EmployeeRelative_Employee_EId_EId"
  refer="mstns:Constraint1">
  <xs:selector xpath="."//EmployeeRelative" />
  <xs:field xpath="EId" />
</xs:keyref>
```

ibi Data Migrator デスクトップインターフェースを使用すると、シノニム内の Employee と EmployeeRelative セグメントの JOIN をすばやく作成することができます。これにより、両方のセグメントを含むレポートの作成が可能になります。

1. ibi Data Migrator デスクトップインターフェースを起動し、シノニムが格納されている [アプリケーションディレクトリ] フォルダを展開します。

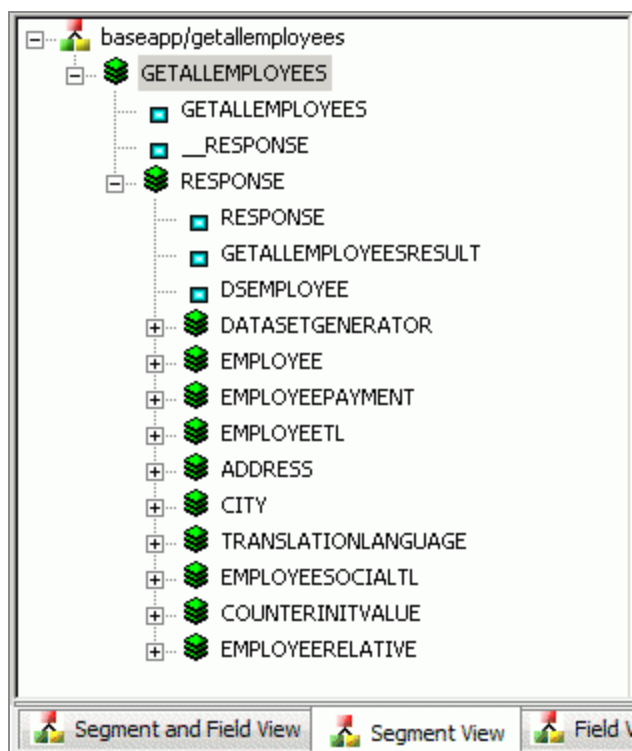
2. ナビゲーションウィンドウで [GetAllEmployees - Web Services] という名前のシノニムをダブルクリックし、[モデル表示] タブをクリックします。セグメントの最初の関係は次のように表示されます。



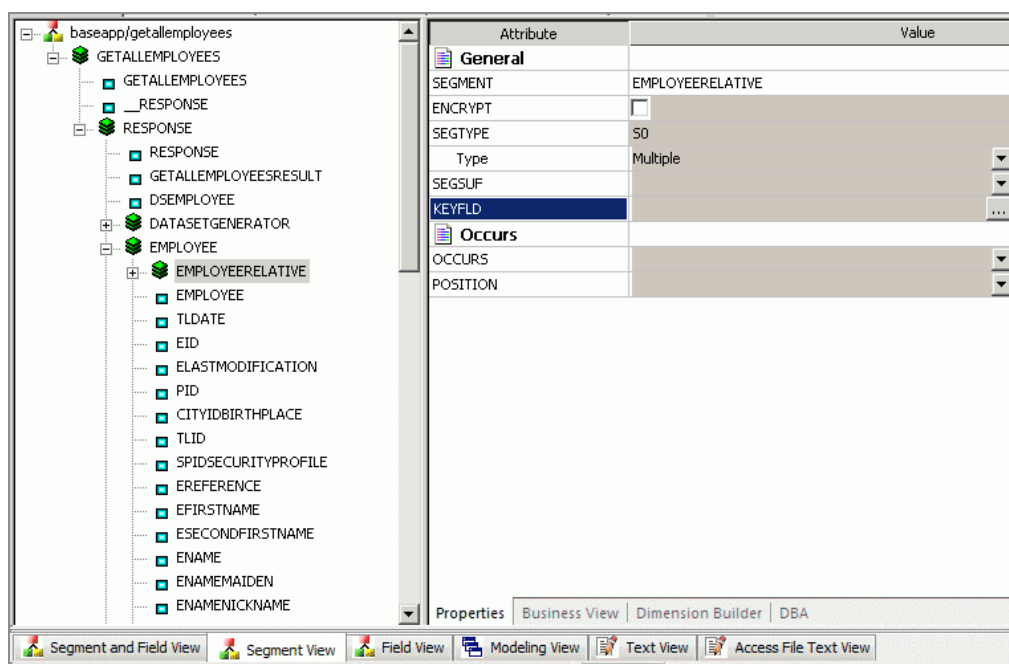
この関係から、Employee と EmployeeRelative はいずれも、作成されたシノニムの Response セグメントの子になっていることが分かります。

ここでは、両方のセグメントを含むレポートを作成することが最終目標ですが、この構造ではこのレポートを作成することはできません。このレポートを作成するには、最初に EmployeeRelative セグメントを Employee セグメントの子にする必要があります。これは、ibi Data Migrator デスクトップインターフェースのセグメント表示を使用して簡単に調整することができます。

3. [セグメント表示] タブをクリックします。下図のように、Employee および EmployeeRelative は、ともに Response セグメントの子になっています。

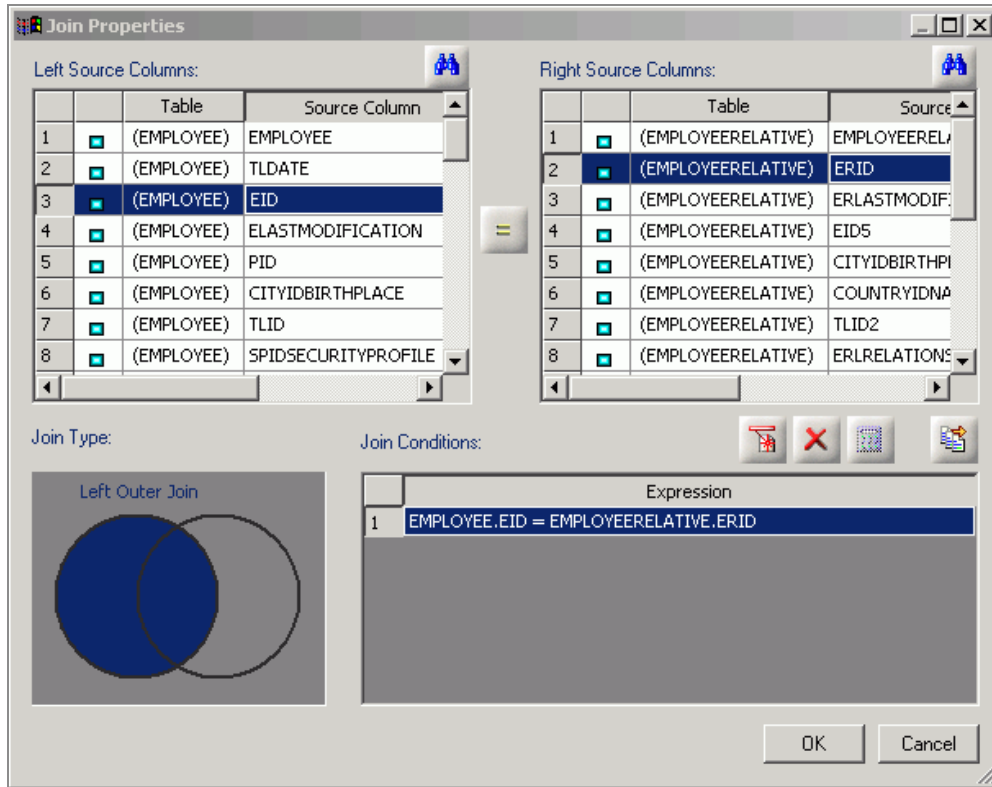


4. EmployeeRelative セグメントを Employee セグメント上にドラッグアンドドロップします。下図のように、編集後のツリーでは、EmployeeRelative が Employee の子になっています。



- 次に、2つのセグメントの結合方法を示す関係を設定する必要があります。このスキーマでは、Employee セグメントと EmployeeRelative セグメントは EID = ERID で結合されています。

この関係を反映するには、右側ウィンドウの EmployeeRelative セグメントの Keyfld に表示された [...] (参照) をクリックします。[JOIN のプロパティ] ダイアログボックスが表示されます。ここで、2つのセグメント間の結合関係を定義します。



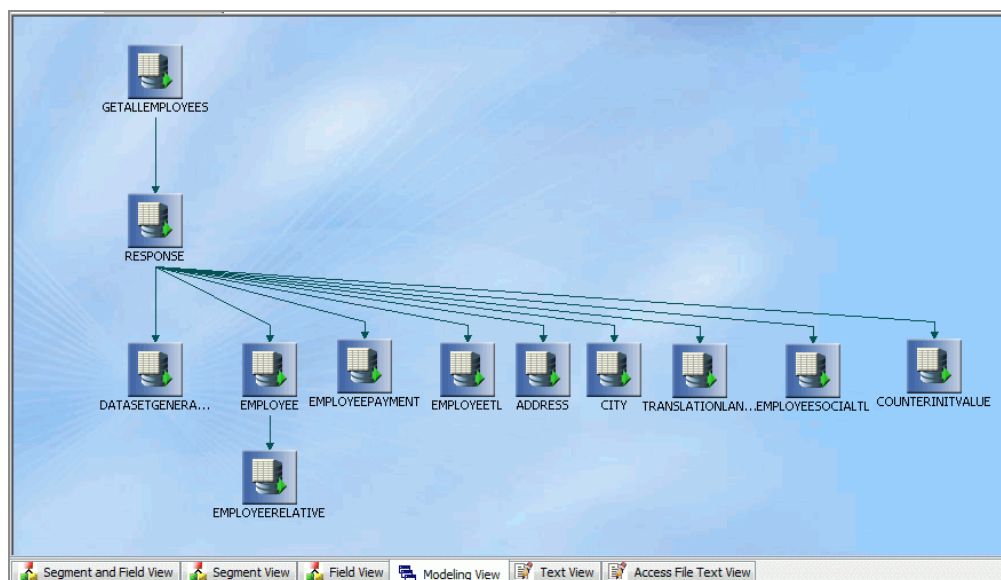
- 左下のベン図ボックスで、右側の円と重ならない左側の円の部分をクリックし、[LEFT OUTER JOIN] を指定します。このオプションにより、左側のデータソースのすべてのレコード、および右側のデータソースから条件に一致するすべてのレコードが取得されます。ここでは、従業員 (Employee) に家族 (EmployeeRelative) がいない場合でも、Employee の値をレポートに表示できる例を示しています。

ヒント： ibi Data Migrator デスクトップインターフェースで使用可能なその他の JOIN 構成の定義については、[JOIN オプション](#)を参照してください。

- ダイアログボックス上部の [左ソースカラム] には Employee セグメントのフィールド、[右ソースカラム] には EmployeeRelative セグメントのフィールドがそれぞれ表示されます。

左側から [EID]、右側から [ERID] を選択し、これらの間の [=] をクリックします。

8. [OK] をクリックして JOIN を終了し、[JOIN のプロパティ] ダイアログボックスを閉じます。
9. [モデル表示] タブをクリックして編集後の構造を表示し、EmployeeRelative が Employee の子で、レポート要件を満たす必要な関係が指定されていることを確認します。



JOIN オプション

この一覧は、ibi Data Migrator デスクトップインターフェースのベン図ボックスで利用可能な JOIN オプションを定義します。

INNER JOIN 2つのデータソースを INNER JOIN で結合し、JOIN で使用される2つのテーブルの両方に存在する行のみを抽出します。

LEFT OUTER JOIN 2つのデータソースを LEFT OUTER JOIN で結合し、左データソースのすべての行、および右データソースから条件に一致するフィールドを抽出します。

RIGHT OUTER JOIN 2つのデータソースを RIGHT OUTER JOIN で結合し、右データソースのすべての行、および左データソースから条件に一致するフィールドを抽出します。

FULL OUTER JOIN 2つのデータソースを FULL OUTER JOIN で結合し、両方のデータソースからすべての行を抽出します。

CROSS JOIN 2つのデータソースを CROSS JOIN (直積) で結合します。CROSS JOIN に、2つのテーブル間の行の組み合わせがすべて含まれます。

WSDL ファイルの概要

WSDL (Web Services Description Language) ファイルは、Web サービスを記述する XML ドキュメントです。WSDL ファイルは、次の主要要素で構成されます。

WSDL 要素	定義
definitions	すべての WSDL ドキュメントのルート要素です。この要素は、Web サービスの名前を定義し、ドキュメントのその他全体で使用する複数のネームスペースを宣言します。また、ここで説明するすべてのサービス要素がこの要素に格納されます。
types	クライアントとサーバの間で使用するすべてのデータタイプを記述します。WSDL は、特定の型システムとの関連付けはありませんが、デフォルトの選択として W3C XML スキーマを使用します。サービスが、文字列や整数などの XML スキーマ内蔵の単純タイプのみを使用する場合は、types 要素を指定する必要はありません。
message	単一メッセージリクエストおよび単一メッセージレスポンスのどちらの場合においても、一方向メッセージを記述します。この要素は、メッセージの名前を定義します。この要素には、メッセージパラメータまたはメッセージ戻り値を表すメッセージの部分要素が格納されます。
portType	複数のメッセージ要素を統合して、一方向または双方向の単一処理を生成します。たとえば、portType は 1 つのリクエストメッセージと 1 つのレスポンスメッセージを統合して、SOAP サービスでよく使用するリクエストとレスポンスの単一処理を生成することができます。なお、portType は複数処理を定義することもできます。
binding	ネットワークでサービスを実装する方法についての詳細を記述します。WSDL には SOAP サービスの定義に使用する拡張子が組み込まれているため、SOAP に特化した情報はこの要素に格納します。
service	指定したサービスを呼び出すためのアドレスを定義します。通常は、SOAP サービスを呼び出す URL を含めます。

FindAddress Web サービスのマスターファイル を生成する WSDL セグメント

次の例は、FindAddress Web サービス関数に使用する入力と出力の要素を示しています。

```
<!-- edited with XMLSPY v5 rel.3 U (http://www.xmlspy.com)-->
<wsdl:definitions
  xmlns:tns="http://arcweb.esri.com/v2"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tme="http://www.themindexelectric.com/"
  xmlns:ns11="http://www.themindexelectric.com/package/
com.esri.is.services.common.v2.geom/"

  xmlns:ns12="http://www.themindexelectric.com/package/com.esri.is.services.commo
n.v2/"
  xmlns:ns13="http://www.themindexelectric.com/package/
com.esri.is.services.glue.v2.addressfinder/"
  xmlns:ns14="http://www.themindexelectric.com/package/java.lang/"
  targetNamespace="http://arcweb.esri.com/v2" name="AddressFinder">
  <wsdl:types>
```

```
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.themindexelectric.com/package/
com.esri.is.services.glue.v2.addressfinder/">
      <xsd:complexType name="Address">
        <xsd:sequence>
          <xsd:element name="houseNumber" nillable="true" type="xsd:string"/>
          <xsd:element name="street" nillable="true" type="xsd:string"/>
          <xsd:element name="intersection" nillable="true" type="xsd:string"/>
          <xsd:element name="city" nillable="true" type="xsd:string"/>
          <xsd:element name="state_prov" nillable="true" type="xsd:string"/>
          <xsd:element name="zone" nillable="true" type="xsd:string"/>
          <xsd:element name="country" nillable="true" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="AddressFinderOptions">
```

```

    <xsd:sequence>
      <xsd:element name="dataSource" nillable="true" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>

```

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.themindelectric.com/package/
com.esri.is.services.common.v2.geom/">
  <xsd:complexType name="Point">
    <xsd:sequence>
      <xsd:element name="x" type="xsd:double"/>
      <xsd:element name="y" type="xsd:double"/>
      <xsd:element name="coordinateSystem" nillable="true"
type="ns11:CoordinateSystem"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="CoordinateSystem">
    <xsd:sequence>
      <xsd:element name="projection" nillable="true" type="xsd:string"/>
      <xsd:element name="datumTransformation" nillable="true"
type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="Envelope">
    <xsd:sequence>
      <xsd:element name="minx" type="xsd:double"/>
      <xsd:element name="miny" type="xsd:double"/>
      <xsd:element name="maxx" type="xsd:double"/>
      <xsd:element name="maxy" type="xsd:double"/>
      <xsd:element name="coordinateSystem" nillable="true"
type="ns11:CoordinateSystem"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>

```

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.themindelectric.com/package/
com.esri.is.services.common.v2/">
  <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
  <xsd:import namespace="http://www.themindelectric.com/package/
com.esri.is.services.common.v2.geom/" />
  <xsd:complexType name="LocationInfo">
    <xsd:sequence>
      <xsd:element name="matchType" nillable="true" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>

```

```

    <xsd:element name="candidates" nillable="true"
type="ns12:ArrayOfLocation"/>
    <xsd:element name="hasMore" type="xsd:boolean"/>
    <xsd:element name="errorCode" nillable="true" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Location">
  <xsd:sequence>
    <xsd:element name="point" nillable="true" type="ns11:Point"/>
    <xsd:element name="description1" nillable="true" type="xsd:string"/>
    <xsd:element name="description2" nillable="true" type="xsd:string"/>
    <xsd:element name="score" type="xsd:double"/>
    <xsd:element name="matchType" nillable="true" type="xsd:string"/>
    <xsd:element name="type" nillable="true" type="xsd:string"/>
    <xsd:element name="locationExtent" nillable="true"
type="ns11:Envelope"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ArrayOfLocation">
  <xsd:complexContent>
    <xsd:restriction base="soapenc:Array">
      <xsd:attribute ref="soapenc:arrayType" wsdl:arrayType="ns12:Location
[]"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.themindelectric.com/package/java.lang/">
  <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/">
  <xsd:complexType name="ArrayOfstring">
    <xsd:complexContent>
      <xsd:restriction base="soapenc:Array">
        <xsd:attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string
[]"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
</wsdl:types>

```

```

<message name="getAddress0In">
  <part name="point" type="ns11:Point"/>
  <part name="addressFinderOptions" type="ns13:AddressFinderOptions"/>

```

```

    <part name="token" type="xsd:string"/>
  </message>
  <message name="getAddress0Out">
    <part name="Result" type="ns13:Address"/>
  </message>

```

```

<portType name="IAddressFinder">
  <operation name="getAddress" parameterOrder="point addressFinderOptions
token">
    <documentation>Returns an address from an x,y-
coordinate.</documentation>
    <input name="getAddress0In" message="tns:getAddress0In"/>
    <output name="getAddress0Out" message="tns:getAddress0Out"/>
  </operation>
</portType>

```

```

<binding name="IAddressFinder" type="tns:IAddressFinder">
  <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getAddress">
    <soap:operation soapAction="getAddress" style="rpc"/>
    <input>
      <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/"
      namespace="http://arcweb.esri.com/v2"/>
    </input>
    <output>
      <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/"
      namespace="http://arcweb.esri.com/v2"/>
    </output>
  </operation>
</binding>

```

```

<service name="AddressFinder">
  <port name="IAddressFinder" binding="tns:IAddressFinder">
    <soap:address
location="http://arcweb.esri.com/services/v2/AddressFinder"/>
  </port>
</service>
</wsdl:definitions>

```


次の例は、シノニムの作成プロセスで生成されたマスターファイルを示しています。

```
FILENAME=M6ILO, SUFFIX=SOAP      , $
  SEGMENT=GETADDRESS, SEGTYPE=S0, $
    GROUP=POINT, ALIAS=point, USAGE=A76, ACTUAL=A100, $
      FIELDNAME=X, ALIAS=x, USAGE=D20.2, ACTUAL=A20, ACCESS_PROPERTY=(NEED_
VALUE), $
      FIELDNAME=Y, ALIAS=y, USAGE=D20.2, ACTUAL=A20, ACCESS_PROPERTY=(NEED_
VALUE), $
    GROUP=COORDINATESYSTEM, ALIAS=coordinateSystem, USAGE=A60, ACTUAL=A60, $
      FIELDNAME=PROJECTION, ALIAS=projection, USAGE=A30, ACTUAL=A30,
      MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
      FIELDNAME=DATUMTRANSFORMATION, ALIAS=datumTransformation, USAGE=A30,
ACTUAL=A30,
      MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
    GROUP=ADDRESSFINDEROPTIONS, ALIAS=addressFinderOptions, USAGE=A30,
ACTUAL=A30, $
      FIELDNAME=DATASOURCE, ALIAS=dataSource, USAGE=A30, ACTUAL=A30,
      MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
      FIELDNAME=TOKEN, ALIAS=token, USAGE=A30, ACTUAL=A30,
      ACCESS_PROPERTY=(NEED_VALUE), $
      FIELDNAME=__RESPONSE, USAGE=TX80L, ACTUAL=TX, ACCESS_PROPERTY=(INTERNAL),
$
```

```
  SEGMENT=RESPONSE, SEGTYPE=S0, SEGSUF=XML      , PARENT=GETADDRESS,
POSITION=__RESPONSE, $
    FIELDNAME=RESPONSE, ALIAS=getAddress00out, USAGE=A1, ACTUAL=A1,
ACCESS_PROPERTY=(INTERNAL), $
    SEGMENT=RESULT, SEGTYPE=S0, PARENT=RESPONSE, $
      FIELDNAME=RESULT, ALIAS=Result, USAGE=A1, ACTUAL=A1,
ACCESS_PROPERTY=(INTERNAL),
      REFERENCE=RESPONSE, PROPERTY=ELEMENT, $
      FIELDNAME=HOUSENUMBER, ALIAS=houseNumber, USAGE=A30, ACTUAL=A30,
      MISSING=ON,
      REFERENCE=RESULT, PROPERTY=ELEMENT, $
      FIELDNAME=STREET, ALIAS=street, USAGE=A30, ACTUAL=A30,
      MISSING=ON,
      REFERENCE=RESULT, PROPERTY=ELEMENT, $
      FIELDNAME=INTERSECTION, ALIAS=intersection, USAGE=A30, ACTUAL=A30,
      MISSING=ON,
      REFERENCE=RESULT, PROPERTY=ELEMENT, $
      FIELDNAME=CITY, ALIAS=city, USAGE=A30, ACTUAL=A30,
      MISSING=ON,
      REFERENCE=RESULT, PROPERTY=ELEMENT, $
      FIELDNAME=STATE_PROV, ALIAS=state_prov, USAGE=A30, ACTUAL=A30,
      MISSING=ON,
```

```
REFERENCE=RESULT, PROPERTY=ELEMENT, $
FIELDNAME=ZONE, ALIAS=zone, USAGE=A30, ACTUAL=A30,
MISSING=ON,
REFERENCE=RESULT, PROPERTY=ELEMENT, $
FIELDNAME=COUNTRY, ALIAS=country, USAGE=A30, ACTUAL=A30,
MISSING=ON,
REFERENCE=RESULT, PROPERTY=ELEMENT, $
```

次の例は、シノニムの作成プロセスで生成されたアクセスファイルを示しています。

```
SEGNAME=GETADDRESS, CONNECTION=CON06, VERSION=1.1, OBJECT=getAddress,
ACTION=getAddress, ENCODING=http://schemas.xmlsoap.org/soap/encoding/,
TARGETNS=http://arcweb.esri.com/v2, STYLE=RPC, $
ID=ns11, NS=http://www.themindexelectric.com/package/
com.esri.is.services.common.v2.geom/ , $
FIELD=point, TYPE=Point, NS_ID=ns11, $
FIELD=coordinateSystem, TYPE=CoordinateSystem, NS_ID=ns11, $
ID=ns13, NS=http://www.themindexelectric.com/package/
com.esri.is.services.glue.v2.addressfinder/ , $
FIELD=addressFinderOptions, TYPE=AddressFinderOptions, NS_ID=ns13, $
```

次のプロシジャを使用して、FindAddress マスターファイルからデータを取得することができます。

```
SET ALL=ON
TABLE FILE FINDADDRESS
PRINT LOCATION.X LOCATION.Y
WHERE FINDADDRESS.HOUSENUMBER EQ '27'
WHERE FINDADDRESS.STREET EQ 'Stevenson Drive'
WHERE FINDADDRESS.ZONE EQ '07746'
WHERE FINDADDRESS.DATASOURCE EQ 'GDT.Streets.US'
END
```

SOAP ヘッダを使用した WSDL ドキュメント

シノニム作成プロセスは、ルートセグメントに新しいグループを生成して、WSDL ドキュメントで定義された SOAP ヘッダレイアウトを記述します。

次の WSDL には SOAP ヘッダが記述されています。

バインディングセクション

```
<input>
  <soap:body use="literal"/>
  <soap:header part="header" message="tns:jobsHeader" use="literal"/>
</input>
```

メッセージ記述

```
<message name="jobsHeader">
  <part element="tns:ibsinfo" name="header"/>
</message>
```

タイプセクション

```
<xs:element name="ibsinfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="xs:string" name="service"/>
      <xs:element type="xs:string" name="method"/>
      <xs:element type="xs:string" name="license"/>
      <xs:element type="xs:string" minOccurs="0" name="disposition"/>
      <xs:element type="xs:string" minOccurs="0" name="Username"/>
      <xs:element type="xs:string" minOccurs="0" name="Password"/>
      <xs:element type="xs:string" minOccurs="0" name="language"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

生成されたマスターファイル

```
SEGMENT=LOCATION, SEGTYPE=S0, $
GROUP=HEADER, ALIAS=Header, USAGE=A210, ACTUAL=A210, $
GROUP=IBSINFO, ALIAS=ibsinfo, USAGE=A210, ACTUAL=A210, $
  FIELDNAME=SERVICE, ALIAS=service, USAGE=A30, ACTUAL=A30,
    ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=METHOD, ALIAS=method, USAGE=A30, ACTUAL=A30,
    ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=LICENSE, ALIAS=license, USAGE=A30, ACTUAL=A30,
    ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=DISPOSITION, ALIAS=disposition, USAGE=A30, ACTUAL=A30,
    ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=USERNAME, ALIAS=Username, USAGE=A30, ACTUAL=A30,
    ACCESS_PROPERTY=(NEED_VALUE), $
  FIELDNAME=PASSWORD, ALIAS=Password, USAGE=A30, ACTUAL=A30,
    ACCESS_PROPERTY=(NEED_VALUE), $
```

```
FIELDNAME=LANGUAGE, ALIAS=language, USAGE=A30, ACTUAL=A30,
ACCESS_PROPERTY=(NEED_VALUE), $
```

結果として作成されるアクセスファイルに含まれる項目

```
SEGNAME=LOCATION, CONNECTION=PSIBS, VERSION=1.1, OBJECT=LOCATION,
ACTION=EFREM.LOCATIONRequest@test@@, HEADER=HEADER,
TARGETNS=urn:xxxsoftware:ibse:jul2003:LOCATION,
HEADERNS=urn:schemas-xxxsoftware-com:iwse, STYLE=DOCUMENT, $
```

なお、ヘッダコンポーネントのうち少なくとも 1 つの値を指定した場合 (リクエストで明示的に指定、またはマスターファイルのデフォルト値として指定)、SOAP ヘッダが SOAP リクエストに生成されます。ヘッダ値を指定しない場合、SOAP ヘッダは生成されません。

出力ヘッダを使用した WSDL ドキュメント

シノニム作成プロセスは、新しい出力セグメントを生成して、SOAP リクエストから返されたヘッダデータを記述します。SOAP レスポンスのヘッダとボディを記述したセグメントは、そのレスポンスの子セグメントとなり、互いに兄弟関係になります。

WSDL の抜粋

```
<operation name="TrackMessagesBulk">
< soap:operation soapAction="http://www.strikeiron.com/TrackMessagesBulk"
    style="document" />
    <input>
        <soap:body use="literal" />
    </input>
    <output>
        <soap:body use="literal" />
        <soap:header message="s0:TrackMessagesBulkResponseInfo"
            part="ResponseInfo" use="literal" />
    </output>
</operation>
```

作成されたマスターファイル

```
SEGMENT=RESPONSE, SEGTYPE=S0, SEGSUF=XML , PARENT=TRACKMESSAGESBULK
POSITION=__RESPONSE, $
FIELDNAME=RESPONSE, ALIAS=Envelope, USAGE=A1, ACTUAL=A1,
ACCESS_PROPERTY=(INTERNAL), $
FIELDNAME=HEADER, ALIAS=Header, USAGE=A1, ACTUAL=A1,
```

```

    ACCESS_PROPERTY=(INTERNAL), $
    FIELDNAME=BODY, ALIAS=Body, USAGE=A1, ACTUAL=A1,
    ACCESS_PROPERTY=(INTERNAL), $
    SEGMENT=HEADER, SEGTYPE=S0, PARENT=RESPONSE, $
    FIELDNAME=RESPONSE, ALIAS=ResponseInfo, USAGE=A1, ACTUAL=A1,
    ACCESS_PROPERTY=(INTERNAL), REFERENCE=HEADER, $
    FIELDNAME=RESPONSECODE, ALIAS=ResponseCode, USAGE=I11, ACTUAL=A11,
    REFERENCE=HEADER.RESPONSE, PROPERTY=ELEMENT, $
    FIELDNAME=RESPONSE1, ALIAS=Response, USAGE=A30, ACTUAL=A30,
    REFERENCE=HEADER.RESPONSE, PROPERTY=ELEMENT, $
    SEGMENT=BODY, SEGTYPE=S0, PARENT=RESPONSE, $
    FIELDNAME=RESPONSE, ALIAS=TrackMessagesBulkResponse, USAGE=A1,
    ACTUAL=A1, ACCESS_PROPERTY=(INTERNAL), REFERENCE=BODY, $
    FIELDNAME=TRACKMESSAGESBULKRESULT, ALIAS=TrackMessagesBulkResult,
    USAGE=A1, ACTUAL=A1, ACCESS_PROPERTY=(INTERNAL),
    REFERENCE=BODY.RESPONSE, PROPERTY=ELEMENT, $

```

複数のヘッダ

Web サービスアダプタは、WSDL ドキュメントの複数のヘッダとボディの定義をマスター ファイルのコンポーネントに変換します。各入力ヘッダとボディは、入力 SOAP セグメントのそれぞれ異なる GROUP 記述になります。各出力ヘッダボディに対し、特別なフィールドの記述がレスポンスセグメントに作成されます。

複合データタイプを使用した WSDL ドキュメント

SOAP レスポンスドキュメントでは、MIXED プロパティを使用した複合データタイプがサポートされます。

```

<xsd:element name="letterBody">
  <xsd:complexType mixed="true">
    <xsd:sequence>
      <xsd:element name="salutation">
        <xsd:complexType mixed="true">
          <xsd:sequence>
            <xsd:element name="name" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

```

        </xsd:complexType>
    </xsd:element>
    <xsd:element name="quantity" type="xsd:positiveInteger"/>
    <xsd:element name="productName" type="xsd:string"/>
    <xsd:element name="shipDate" type="xsd:date" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>

```

生成されたマスターファイル

```

SEGMENT=LETTERBODY, SEGTYPE=S0, $
  FIELDNAME=LETTERBODY, ALIAS=letterBody, USAGE=A50, ACTUAL=A50, $
  FIELDNAME=SALUTATION, ALIAS=salutation, USAGE=A50, ACTUAL=A50,
    REFERENCE=LETTERBODY, PROPERTY=ELEMENT, $
  FIELDNAME=NAME, ALIAS=name, USAGE=A50, ACTUAL=A50,
    REFERENCE=SALUTATION, PROPERTY=ELEMENT, $
  FIELDNAME=QUANTITY, ALIAS=quantity, USAGE=P32, ACTUAL=A32,
    REFERENCE=LETTERBODY, PROPERTY=ELEMENT, $
  FIELDNAME=PRODUCTNAME, ALIAS=productName, USAGE=A50, ACTUAL=A50,
    REFERENCE=LETTERBODY, PROPERTY=ELEMENT, $
  FIELDNAME=SHIPDATE, ALIAS=shipDate, USAGE=YYMD, ACTUAL=A10, MISSING=ON,
    REFERENCE=LETTERBODY, PROPERTY=ELEMENT, $

```

Web サービスアダプタでは、テキストコンポーネントおよび下位要素の元の順序が保持されないという制限があります。次の SOAP レスポンスドキュメントとレポートはこの制限を示しています。

```

<letterBody>
  <salutation>Dear Mr.
    <name>Robert Smith</name>.
  </salutation>
  Your order of
  <quantity>1</quantity>
  <productName>BabyMonitor</productName>
  shipped from our warehouse on
  <shipDate>1999-05-21</shipDate>.
</letterBody>

```

LETTERBODY	SALUTATION NAME	QUANTITY	PRODUCTNAME	SHIPDATE
-----	-----	-----	-----	-----

Your order of	shipped from our warehouse on	.Dear Mr.	.Robert Smith	
1	BabyMonitor	1999/05/21		

抽象入力データタイプを使用した WSDL ドキュメント

Web サービスアダプタは、ACTUAL および USAGE フォーマットに A1 を指定した入力フィールドを作成します。このフィールドの ACCESS プロパティは、フロントエンドツールが入力値を要求しないように初期値として INTERNAL が設定されています。必要に応じて、この定義を手動で変更することができます。

メッセージ記述

```
<message name="ivpIn">
  <part element="m1:ivp" name="parameters"/>
</message>
Input message definition:
<xs:element name="ivp">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
```

生成されたマスターファイル

```
FIELDNAME=IVP, ALIAS=ivp, USAGE=A1, ACTUAL=A1, ACCESS_PROPERTY=(INTERNAL), $
```

属性とデフォルト値のマッピング

要素属性は、サービス入力メッセージ定義の一部としてサポートされます。

- 制限値は、ACCEPT リストとしてマスターファイル内の対応するフィールド定義に格納されます。
- デフォルト値は、XDEFAULT パラメータに格納されます。

属性とデフォルト値をマッピングした WSDL ドキュメント

```
<xs:element name="component">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute use="optional" default="browse" name="perform">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="browse"/>
              <xs:enumeration value="insert"/>
              <xs:enumeration value="update"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

生成されたマスターファイル

```
FIELDNAME=COMPONENT, ALIAS=component, USAGE=A30, ACTUAL=A30,
ACCESS_PROPERTY=(NEED_VALUE), $

FIELDNAME=PERFORM, ALIAS=perform, USAGE=A6, ACTUAL=A6,
ACCESS_PROPERTY=(NEED_VALUE),

XDEFAULT='browse',
ACCEPT='browse' OR 'insert' OR 'update',
REFERENCE=COMPONENT, PROPERTY=ATTRIBUTE, $
```

なお、このフィールドに明示的な値を指定しない場合、SOAP リクエストの生成時に XDEFAULT の値が使用されます。

入力値としての配列の使用

Web サービスアダプタは、入力値としての配列の使用をサポートします。配列は、マスターファイルの GROUP 定義に ACCESS_PROPERTY=ARRAY_ITEM として記述されます。GROUP

は、オブジェクトの 1 つのインスタンスを記述します。次のいずれかの方法を使用して、オブジェクトの配列に入力値を指定することができます。

- GROUP 内の個々のフィールドに並列 OR による選別条件を指定します。
- GROUP 自体に並列 OR による選別条件を指定します。なお、複数の GROUP フィールドの値はスラッシュ (/) で区切ります。
- 次のように指定して、ファイルから値を読み取ります。

```
(IF group EQ (ddname))
```

ファイル内のコンストラクトでは、配列を記述した GROUP のみを使用することができます。以下はその例です。

```
FILEDEF CARRAY DISK C:¥Users¥.....¥countryarray.ftm
.
.
.
IF COUNTRY EQ (CARRAY)
```

なお、入力配列コンポーネントには、最後の選別条件の値のみが返されます。

入力値としての配列の使用

次の例では、ESRI SOAP リクエストの構造を定義したセグメントが、住所を表す値の組み合わせのリストを取得しています。GROUP ADDRESSES は、ADDRESSES 配列からのオブジェクト項目の 1 つを定義するために使用されます。また、ACCESS_PROPERTY=ARRAY_ITEM を追加し、配列を構成する各フィールドの入力値を指定することで、この GROUP 定義を拡張します。

マスターファイル

```
SEGMENT=FINDDADDRESS, SEGTYPE=S0, $
GROUP=ADDRESSES, ALIAS=addresses, USAGE=A210, ACTUAL=A210,
ACCESS_PROPERTY=ARRAY_ITEM, $
FIELDNAME=HOUSENUMBER, ALIAS=houseNumber, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='2815', $
FIELDNAME=STREET, ALIAS=street, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='PRAIRIE AVE.', $
FIELDNAME=INTERSECTION, ALIAS=intersection, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), $
FIELDNAME=CITY, ALIAS=city, USAGE=A30, ACTUAL=A30,
```

```

MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='MIAMI BEACH', $
FIELDNAME=STATE_PROV, ALIAS=state_prov, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='FL', $
FIELDNAME=ZONE, ALIAS=zone, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='33140', $
FIELDNAME=COUNTRY, ALIAS=country, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE), XDEFAULT='US', $
GROUP=ADDRESSFINDEROPTIONS, ALIAS=addressFinderOptions, USAGE=A30,
ACTUAL=A30, $
FIELDNAME=DATASOURCE, ALIAS=dataSource, USAGE=A30, ACTUAL=A30,
MISSING=ON, ACCESS_PROPERTY=(NEED_VALUE),
XDEFAULT='GDT.Streets.US', $
FIELDNAME=TOKEN, ALIAS=token, USAGE=A120, ACTUAL=A120,
ACCESS_PROPERTY=(NEED_VALUE, AUTHTOKEN), $
FIELDNAME=__RESPONSE, USAGE=TX80L, ACTUAL=TX,
ACCESS_PROPERTY=(INTERNAL), $

```

注意：この例では紹介していませんが、ACCESS_PROPERTY=ARRAY_ITEM の値を使用して、ネストされた GROUP 定義を作成することができます。ネストされた定義では、内側の GROUP オブジェクトが配列内の配列に相当します。ネストされた配列を使用すると、任意の階層オブジェクトを記述することができます。

リーフコンポーネントの並列 OR による選別

リーフコンポーネントの並列 OR による選別方法について説明します。

XML 配列項目オブジェクトはオブジェクトコンポーネントのサブセットで作成できるため、GROUP 内のすべてのフィールドに値を指定する必要はありません。ただし、次の例で示す 2 つの選別方法では、ミッシング値に対する処理は異なります。

どちらのリクエストも MATCH 配列に同一要素を作成します。

フィールドによる明示的な選別

GROUP 内の個々のフィールドに並列 OR による選別条件を指定する場合、値の数と位置は、配列 GROUP のすべてのフィールドにおいて同一である必要があります。

次の各フィールドの下には値が 3 つあります。配列がこの入力から生成されると、各行の 1 つ目の値の位置合わせが行われ、その後、2 つ目の値と 3 つ目の値に同様の位置合わせが行われます。このため、値を省略または誤った位置に記述すると、正しい配列が行われなくなります。

```

IF HOUSENUMBER EQ '2815' OR 'Two' OR '1250'
IF STREET EQ 'PRAIRIE AVE.'OR 'Penn Plaza' OR 'Broadway'
IF CITY EQ 'MIAMI BEACH' OR 'NY' OR 'NY'
IF STATE_PROV EQ 'FL' OR 'NY' OR 'NY'

```

```
IF ZONE EQ '33140' OR '10121' OR '10001'
IF COUNTRY EQ 'US' OR 'US' OR 'US'
```

この選別リクエストでは、INTERSECTION フィールドには値がないことに注意してください。個々のフィールドを明示的に指定する際、配列に含まないフィールドは省略することができます。

グループ全体への選別

GROUP に並列 OR による選別条件を指定する場合、GROUP 内の個々のフィールドの値は、スラッシュ (/) で区切ります。この時、INTERSECTION などのミッシング値も考慮する必要があります。これは、選別リクエストの各行でスラッシュを 2 つ入力 (//) することで指定します。

```
IF COUNTRY EQ '2815/PRAIRIE AVE.//MIAMI BEACH/FL/33140/US' OR
               'Two/Penn Plaza//NY/NY/10121' OR
               '1250/Broadway//NY/NY/10001'
```

この例では、INTERSECTION は「STREET」と「CITY」の間の「//」で表されています。

XML 要素とその属性への入力値としての配列の使用

標準の Web サービスの配列方法は、親 XML 配列要素値に関連付けられた XML 属性値に適用することができます。親配列要素は、ACCESS_PROPERTY パラメータ内の ARRAY_ITEM 値で記述します。関連付けられた属性は、PROPERTY パラメータが ATTRIBUTE に設定された状態で、REFERENCE パラメータにより親要素にリンクされます。通常の配列のように、要素と属性に入力値を指定することができます。

XML 要素とその属性への入力値としての配列の使用

マスターファイルの抜粋

```

FIELDNAME=KEY, ALIAS=key, USAGE=A30, ACTUAL=A30,
ACCESS_PROPERTY=(ARRAY_ITEM), $
FIELDNAME=NAME, ALIAS=name, USAGE=A30, ACTUAL=A30,
ACCESS_PROPERTY=(NEED_VALUE), REFERENCE=KEY, PROPERTY=ATTRIBUTE, $

```

選別ステートメントを含むリクエスト

```

IF KEY EQ 'SHARE' OR 'BAE01A'
IF NAME EQ 'Setid' OR 'Location'

```

生成された SOAP リクエスト

選別ステートメントにより、SOAP リクエストにキーと名前の配列が生成されます。

```

<SOAP-ENV:Body>
  <m:GetQuote xmlns:m="http://ws.cdyne.com/">
    <m:key xsi:type="xsd:string" name="Setid" >SHARE</m:key>
    <m:key xsi:type="xsd:string" name="Location" >BAE01A</m:key>
  </m:GetQuote>
</SOAP-ENV:Body>

```

WSDL スキーマでのネームスペース修飾子の編集

ネームスペース修飾子は、WSDL スキーマに基づいて SOAP リクエストドキュメントで生成されます。アクセスファイルで修飾ルールを編集し、その変更を SOAP XML リクエストの作成方法に反映させることができます。

修飾コントロール情報は、アクセスファイルのセグメント、ネームスペース (ID)、要素または属性 (マスターファイルのフィールド) の各レベルに格納されます。

- セグメントレベルの修飾は、ターゲットネームスペースに属するボディおよびヘッダのそれぞれに存在するすべての要素および属性の修飾子を制御します。セグメントレベルの修飾情報は、ターゲットネームスペースのスキーマレベルから取得されます。
- ID レベルの修飾は、記述されたネームスペースのすべての要素および属性の修飾子を制御します。ID レベルの修飾情報は、参照スキーマに対応する特定のネームスペースから取得されます。
- フィールドレベルの修飾は、特定の要素および属性の修飾子を制御し、セグメントお

よび ID の設定を上書きします。フィールドレベルの修飾情報は、特定のスキーマ要素のタイプ定義から取得されます。

新しいパラメータは、アプリケーションのセグメントおよび ID の両方のレコードに挿入されます。

- ELEMFORM および ATTRFORM (SOAP ボディ用)
- HELEMFORM および HATTRFORM (SOAP ヘッダ用)

新しい FORM パラメータは、要素または属性を記述したフィールドレコードに挿入されます。

新しいパラメータが受容可能な値は、qualified および unqualified です。

アクセスファイルに新しいパラメータが見つからない場合は、デフォルト設定の動作としてヘッダおよびボディのすべての要素には完全修飾が適用されますが、属性には修飾は適用されません。

高度なコンセプトであるネームスペース、スキーマ、修飾についての詳細は、W3C 勧告の『XML Schema Part 0: Primer』で「Advanced Concepts I: Namespaces, Schemas & Qualification」を参照してください。このドキュメントには、<http://www.w3.org/TR//2001/REC-xmlschema-0-20010502/#NS> からアクセスすることができます。

データタイプのサポート

下表は、サーバが XSD データタイプをマスターファイルでマッピングする方法を示しています。

XSD データタイプ	USAGE	ACTUAL
string	A30	A30
double	D20.2	A20
float	F15.2	A15
decimal	P20.3	A20
int	I11	A11

XSD データタイプ	USAGE	ACTUAL
short	I6	A6
long	P20	A20
boolean	A5	A5
dateTime	HYMDm 詳細は、 日付時間の処理 を参照してください。	A35
time	HHISsm	A15
date	YYMD	A10
gYearMonth	HYYM	A8
gYear	HYY	A5
gMonthDay	HMD	A6
gDay	HD	A3
gMonth	HM	A4
integer	P33	A33
nonPositiveInteger	P33	A33
negativeInteger	P33	A33
nonNegativeInteger	P32	A32
unsignedLong	P20	A20
unsignedInt	P10	A10

XSD データタイプ	USAGE	ACTUAL
unsignedShort	I5	A5
positiveInteger	P32	A32
byte	I4	A4
unsignedByte	I4	A4
normalizedString	A30	A30
token	A30	A30
Name	A30	A30
NMTOKEN	A30	A30
ID	A30	A30
hexBinary	A30	A30
language	A30	A30
anyURI	A30	A30
QName	A30	A30

日付時間の処理

日付時間フォーマットは ISO 8601:2000 標準日付時間と互換性がある表記の出力値を生成し、入力値を受容します。

ISO 標準日付時間表記の有効化

```
SET DTSTANDARD = {OFF|ON|STANDARD|STANDARDU}
```

説明

OFF

ISO 8601:2000 標準日付時間表記との互換性を提供しません。

ON|STANDARD

ISO 標準フォーマットの認識および出力を有効にします。これを有効にすると、日付と時間の区切り文字として T を使用すること、秒を桁数で表すときの区切り文字にピリオド (.) およびカンマ (,) を使用すること、世界時の末尾に Z を使用すること、タイムゾーン情報付きの入力の受容が可能になります。STANDARD は ON の同義語です。

STANDARDU

ISO 標準フォーマットを (STANDARD と同様に) 有効にし、可能な場合は入力文字列を対応する世界時の時間に変換します。世界時は、以前は「グリニッジ標準時」として知られていました。このオプションにより、アプリケーションで一貫した日付時間値を格納することが可能になりました。

プロシジャに FILEDEF SOAPTSCQ を使用した SOAP リクエストの取り込み

プロシジャ内で FILEDEF SOAPTSCQ コマンドを使用して、生成された SOAP リクエストを取り込み、ファイルシステムの所定の場所に格納することができます。実際には SOAP リクエスト自体は実行されないため、レポートに返されるレコードはありません。

FILEDEF SOAPTSCG を使用した SOAP リクエストの取り込み

この例では、プロシジャは「SYMBOL」および「USERNAME」という 2 つのパラメータを持つ Web サービス処理に対してレポートを実行します。生成された SOAP リクエストは、

「SOAPrequest.xml」というファイルに格納されます。

```
FILEDEF SOAPTSCQ DISK C:¥IBI¥APPS¥XIGNITE¥SOAPrequest.xml
TABLE FILE XQUOTES_GETQUOTE
PRINT SYMBOL1 LAST
WHERE SYMBOL EQ `IBM`
WHERE USERNAME EQ `myusername`
END
```

以下は、プロシジャ実行後の SOAPrequest.xml の内容を示しています。

```
<soap_tscq>
<soap_url>http://www.xignite.com/xQuotes.asmx</soap_url>
<soap_action>http://www.xignite.com/services/GetQuote</soap_action>
<soap_body>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.xignite.com/services/">
<SOAP-ENV:Header>
<m:Header xmlns:m="http://www.xignite.com/services/">
<m:Username xsi:type="xsd:string">myusername</m:Username>
  </m:Header>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<m:GetQuote xmlns:m="http://www.xignite.com/services/">
<m:Symbol xsi:type="xsd:string">IBM</m:Symbol>
  </m:GetQuote>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
</soap_body>
</soap_tscq>
```

WebFOCUS Client アダプタの使用

WebFOCUS Client アダプタを構成すると、WebFOCUS Client RESTful Web サービス経由で WebFOCUS との統合が可能になります。このアダプタへの接続を構成することで、WebFOCUS への [Explicit] および [Password Passthru] 認証が行われます。認証に成功すると、クロスサイトリクエストフォージェリ (CSRF) トークンが返され、このトークンが自動的に各 WebFOCUS RESTful Web サービスコールに渡されます。

構成した接続を右クリックして [DBMS オブジェクトの表示] を選択し、[シノニムとサンプルの作成] をクリックすると、マスターファイル、アクセスファイル、プロシジャで構成される一連のサンプルがロードされます。これらのサンプルのそれぞれは、特定の WebFOCUS タスクを実行します (例、セキュリティ管理、ReportCaster スケジュール、フォルダ管理)。

アダプタを構成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで [データの取得] をクリックします。
2. ibi™ WebFOCUS® Client アダプタのアイコンを右クリックし、[構成] を選択します。
[ibi™ WebFOCUS® Client の構成の追加] ページが開きます。
3. [ibi™ WebFOCUS® Client の構成の追加] ページで、次のパラメータを構成します。

接続名

接続の名前です。

WebFOCUS ベース URL

WebFOCUS へのアクセスに使用する URL を「http://computername:port/ibi_apps」の形式で入力します。

セキュリティ

[Trusted]、[Explicit]、[Password Passthru] のいずれかを選択します。[Explicit] 認証の場合、WebFOCUS ユーザ ID とパスワードを入力します。

4. [テスト] をクリックして、接続パラメータを確認します。
5. 接続パラメータの確認後、[構成] をクリックします。

アダプタでのシノニムとサンプルの作成

1. WebFOCUS Client アダプタの接続を右クリックし、[DBMS オブジェクトの表示] を選択します。
[ibi™ WebFOCUS® Client シノニムの作成] ページが開きます。
2. [アプリケーション] テキストボックスに、シノニムとサンプルを格納するアプリケーション名を入力するか、参照ボタンをクリックしてアプリケーションを指定します。
3. [シノニムとサンプルの作成] をクリックします。
FOCEXEC を使用した一連のシノニムおよびフォルダが、選択したアプリケーションに作成されます。

XML アダプタの使用

XML アダプタの構成方法について説明します。

XML アダプタを使用すると、アプリケーションから XML データソースへのアクセスが可能になります。このアダプタは、階層ツリーを読み取り、アンサーセットをリクエストの送信元アプリケーションに返します。

XML 環境の準備

XML アダプタでは、環境変数を設定する必要はありません。

XML アダプタの構成

XML アダプタは、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースから構成することができます。

アダプタの構成

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

または

ibi Data Migrator デスクトップインターフェースで、[アダプタ] フォルダを展開します。

WebFOCUS Reporting Server ブラウザインターフェースで [データの取得] ページが開き、構成済みのアダプタが表示されます。 ibi Data Migrator デスクトップインター

フェースで [アダプタ] フォルダが開きます。

2. WebFOCUS Reporting Server ブラウザインターフェースで [新規データソース] (プラス (+) ボタン) をクリックして表示されたページ、 ibi Data Migrator デスクトップインターフェースのアダプター一覧からアダプタを特定します。

WebFOCUS Reporting Server ブラウザインターフェースで、[新規データソース] ボタンをクリックし、表示される [利用可能] ドロップダウンリストからアダプタのカテゴリを選択することも、検索オプション (拡大鏡アイコン) を使用して特定の文字列を検索することもできます。

3. ibi Data Migrator デスクトップインターフェースで、適切なグループフォルダと特定のアダプタフォルダを展開します。グループフォルダについての説明は、アダプタの「接続属性」に記載されています。
4. アダプタの名前またはバージョンを右クリックして、[接続の追加] を選択します。
[アダプタの構成を追加] ウィンドウが表示されます。
5. アダプタに必要なパラメータの値を入力します。詳細は、構成する特定のアダプタについての説明を参照してください。
6. [構成] をクリックします。構成されたアダプタが、WebFOCUS Reporting Server ブラウザインターフェースの構成済みアダプタリストまたは ibi Data Migrator デスクトップインターフェースのリソースツリーのアダプタリストに追加されます。

WebFOCUS Reporting Server ブラウザインターフェースで、アダプタがアスタリスク (*) 付きで [利用可能] リストに表示されている場合は、少なくとも 1 つの接続が構成済みであることを示しています。アダプタを右クリックし、[接続の追加] を選択することで、[利用可能] リストのいずれかから追加の接続を構成することができます。

XML メタデータの管理

サーバがデータソースにアクセスする際、データソースに保存されているデータの解釈方法を認識する必要があります。サーバがアクセスするデータソースごとに、データソースの構造および XML データタイプのサーバマッピングを定義するシノニムを作成します。

シノニムの作成

シノニムを作成して、サーバからアクセス可能な XML データ構造の一意の名前 (エイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケー

ションから隠すことができるという点で便利です。また、シノニムは一時項目 (DEFINE) や他のセキュリティ機構などの、サーバの拡張メタデータ機能をサポートします。

シノニムを使用すると、クライアントアプリケーションを変更せずに、動作を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。シノニムを作成すると、指定した XML ドキュメントに基づいてマスターファイルとアクセスファイルが生成されます。

シノニムを作成するには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースの [アプリケーション] ページで、[データの取得] をクリックします。

2. このページ (簡略モード) の [利用可能なデータソース] セクションのアダプタを右クリックし、[接続の表示] を選択します。接続を右クリックします。

選択するアダプタのタイプに応じて、コンテキストメニューに、次のいずれかのオプションが表示されます。

- **DBMS オブジェクトの表示** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **メタデータオブジェクトの作成** シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **ローカルファイルの表示** ファイルの選択ダイアログボックスが表示されます。適切なタイプのファイルを選択すると、シノニムオブジェクトおよびプロパティの選択ページが表示されます。
- **トピックの表示** 環境で、トピックのシノニムオブジェクトおよびプロパティの選択ページが表示されます。

3. 使用するアダプタの説明に従って、アダプタの必須パラメータの値を入力します。

4. パラメータ値を入力後、[追加] ボタンをクリックします。

このボタンのラベルは、[次へ]、[シノニムの作成]、[ベースシノニムの作成]、[クラスタシノニムの作成]、[ベースシノニムの更新] のいずれかになる場合があります。

ほとんどのアダプタのシノニム作成の処理は統合されているため、単一ページにすべての必須パラメータを入力することができます。ただし、LDAP などの一部のアダプタでは、[シノニムの作成] ボタンが表示されるページに移動するまで、[次へ] ボタンを数回クリックする必要があります。

結果

作成したシノニムが、指定したアプリケーションディレクトリに追加されます。

注意：シノニムの作成時に [フィールド名の確認] のチェックをオンにすると (選択可能な場合)、サーバによって特殊文字が調整され、予約語の有無が確認されます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

XML のシノニム作成パラメータ

以下のリストは、アダプタのシノニムを作成する際に、値の指定が必要なパラメータ、および実行する関連作業を示しています。これらのオプションが複数のウィンドウに表示される場合もあります。ウィンドウに表示されたボタンをクリックして次のウィンドウに進み、最後に [シノニムの作成] ボタンをクリックします。これで、入力データに基づいてシノニムが作成されます。

シノニムは、XML ドキュメントまたは XML スキーマのいずれかに基づいて作成することができます。これらは、下図のように、ローカルファイルシステムまたは URL で指定できます。

Create Synonym for XML

^ Create Synonym options

XML Document

? ☐ HTTP Location

? Local Document

Schema Definition

? ☐ HTTP Location

? Local Schema

▼ Advanced

☐ Customize data type mappings

? Application ... ? Prefix ? Suffix

? Synonym Name

- XML ドキュメントに基づいてシノニムを作成する場合は、[XML ドキュメント] セク

ションの必須パラメータを入力します。

- XML スキーマに基づいてシノニムを作成する場合は、[スキーマ定義] セクションの必須パラメータを入力します。

HTTP パス

URL を使用してドキュメントインスタンスを選択します。この選択には、[ドキュメントパス]、[ドキュメント名]、および [ドキュメント拡張子] が必要になります。これらのテキストボックスは、[HTTP パス] のチェックをオンにした場合のみ使用可能になります。シノニムを作成するために使用する XML ドキュメントが格納されたディレクトリの http アドレスを入力します。この機能は、XML ドキュメントがローカルファイルの場合は使用できません。URL は、http:// または https:// で始める必要があります。

ローカルドキュメント

ドキュメントインスタンスのパスを定義します。[HTTP パス] を選択していない場合、物理パスまたはアプリケーションディレクトリ、および XML ドキュメント名を入力するか、参照ボタンをクリックしてドキュメントを選択します。

ドキュメント名

XML ドキュメントの名前を入力します。

ドキュメント拡張子

ドキュメントの拡張子を入力します。デフォルト値は xml です。

詳細設定セクション

シノニムフィールド名の処理オプション

フィールド名の確認

すべての特殊文字をアンダースコア (_) に変換するとともに、予約語の使用を回避するために名前の確認を行う場合は、[フィールド名の確認] のチェックをオンにします。これにより、名前に番号が付加されます。このパラメータにより、規則に一致した名前が付けられます。詳細は、[特殊文字および予約語の確認](#)を参照してください。

[フィールド名の確認] のチェックをオフにすると、ハイフン (-)、ブランク、円記号 (¥)、フォワードスラッシュ (/)、カンマ (,)、ドル記号 (\$) の特殊文字のみがアンダースコア (_) に変換されます。名前の確認は行われません。

ユニークフィールドにする

フィールド名およびグループ名の範囲をシノニム全体に設定する場合、[ユニークフィールドにする] のチェックをオンにします。これにより、シノニムの異なるセグメントで

あっても重複名は使用されません。このチェックをオフにすると、範囲はセグメントに設定されます。

シノニム名

シノニムに割り当てる名前を指定します。異なる名前を指定する場合は、表示されている名前を上書きします。

書き込み

スキーマファイル (.xsd) を使用する場合、このオプションを選択すると、シノニムのアクセスファイルに詳細情報が追加されます。この追加情報には、XML の複合データタイプの CHOICE と SEQUENCE の各定義が反映され、XML MODIFY で使用されます。

位置

シノニムに基づく要素のサブツリーへの XPATH を定義します。

アプリケーション

アプリケーションディレクトリの 1 つを選択します。デフォルト値は baseapp です。

接頭語/接尾語

同一名のテーブルが存在する場合、接頭語または接尾語を追加して両者を区別します。たとえば、同名のテーブルが人事名簿と給与支払名簿に存在する場合、人事名簿には「HR」という接頭語を付けます。なお、シノニム名の最大文字数は 64 バイトです。

テーブルおよびビューの名前がすべて固有の場合は、[接頭語]、[接尾語] テキストボックスはブランクにしておきます。

これらのパラメータを入力後、リボンの [シノニムの作成] をクリックします。

empxml xsd のシノニムの作成

empxml.xsd

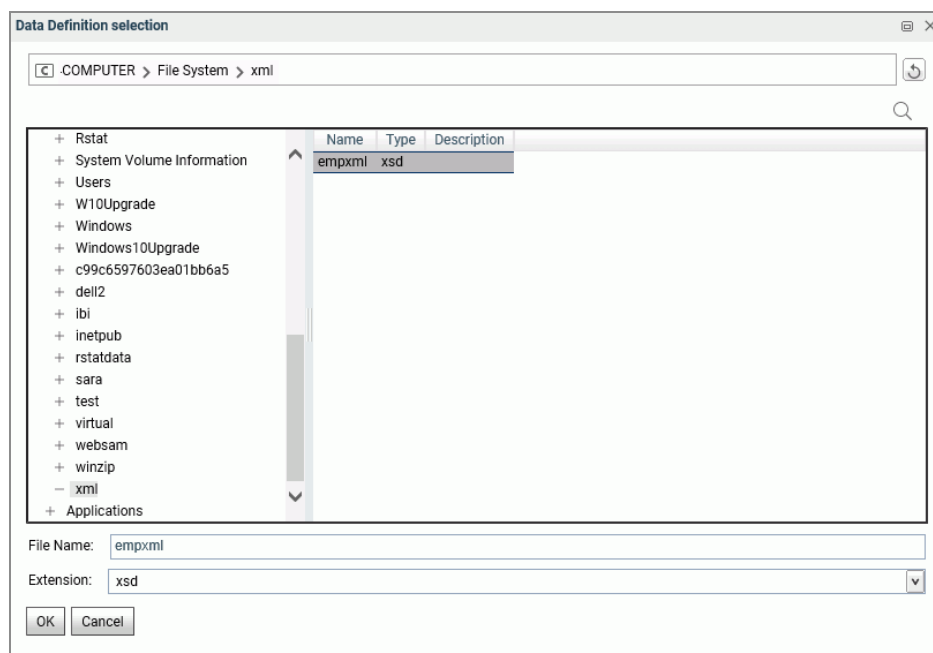
```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://example.org/employee/"
  targetNamespace="http://example.org/employee/">
  <xsd:simpleType name="genderIdentifiers">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="male"/>
      <xsd:enumeration value="female"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

```

        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="socialSecurityNumber">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="¥d{3}¥-¥d{2}¥-¥d{4}" />
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="annotatedAge">
        <xsd:simpleContent>
            <xsd:extension base="xsd:unsignedShort" />
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:complexType name="Person">
        <xsd:sequence>
            <xsd:element name="name" type="xsd:string" />
            <xsd:element name="sex" type="tns:genderIdentifiers" />
            <xsd:element name="age" type="tns:annotatedAge" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="Employee">
        <xsd:complexContent>
            <xsd:extension base="tns:Person">
                <xsd:sequence>
                    <xsd:element name="ssnum"
type="tns:socialSecurityNumber" />
                    <xsd:element name="salary" type="xsd:double" />
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="employee" type="tns:Employee" />
</xsd:schema>

```

empxml xsd からシノニムを生成するには、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースの [シノニムの作成] ウィンドウで xsd ファイルのパスを入力するか、下図のように、[...] (参照) ボタンをクリックしてローカルファイルシステム上の xsd ファイルを指定します。



上記の場合、xsd ファイルは次のファイルシステム上にあります。

```
c:\¥xml¥empxml.xsd
```

1. [次へ] をクリックします。
2. [名前] テキストボックスで [empxml] を選択します。
3. リボンの [シノニムの作成] をクリックします。シノニムが作成され、指定したアプリケーションディレクトリに追加されます。デフォルトのディレクトリは ibisamp です。
4. ナビゲーションウィンドウで [ibisamp] アプリケーションフォルダを展開した後、[empxml] シノニムを右クリックし、[テキストとして編集] を選択して、生成されたマスターファイルを表示します。
5. [empxml] シノニムを右クリックし、[テキストとしてアクセスファイルを編集] を選択して、対応するアクセスファイルを表示します。

作成されたマスターファイル empxml.mas

```
FILENAME=EMPXML, SUFFIX=XML      , $
  SEGMENT=EMPLOYEE, SEGTYPE=S0, $
    FIELDNAME=EMPLOYEE, ALIAS=employee, USAGE=A1, ACTUAL=A1, ACCESS_PROPERTY=
  (INTERNAL), $
    FIELDNAME=NAME, ALIAS=name, USAGE=A10, ACTUAL=A10,
```

```

REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=SEX, ALIAS=sex, USAGE=A10, ACTUAL=A10,
ACCEPT='male' OR 'female',
REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=AGE, ALIAS=age, USAGE=I5, ACTUAL=A5,
REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=SSNUM, ALIAS=ssnum, USAGE=A10, ACTUAL=A10,
REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=SALARY, ALIAS=salary, USAGE=E24.16, ACTUAL=A24,
REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $

```

作成されたアクセスファイル empxml.acx

```

SEGNAME=EMPLOYEE,
TARGETNS=http://example.org/employee/,
LOCATION=C:\xml\empxml.xsd, $
ID=tns,
NS=http://example.org/employee/,
ELEMFORM=unqualified,
ATTRFORM=unqualified, $
FIELD=EMPLOYEE,
TYPE=Employee,
NS_ID=tns, $
FIELD=SEX,
TYPE=genderIdentifiers,
NS_ID=tns, $
FIELD=AGE,
TYPE=annotatedAge,
NS_ID=tns, $
FIELD=SSNUM,
TYPE=socialSecurityNumber,
NS_ID=tns, $

```

スキーマサブセットからのシノニム作成

大規模な XML スキーマからシノニムを作成する場合、階層内の特定の位置に基づくサブセットを使用することでパフォーマンスを向上させることができます。これにより、XML ルートのサブツリー (子要素) から小規模なシノニムが作成されます。

構文は次のとおりです。

```
CREATE SYNONYM child_element FOR schema.xsd
DBMS XML
AT app_dir
PARMS POSITION=/parent_element/child_element
DROP
```

```
END
```

説明

child_element

シノニムの作成元となるサブツリーです。

schema.xsd

シノニムの基礎となるスキーマです。

app_dir

シノニムを作成するアプリケーションディレクトリです。

PARMS POSITION

スキーマ階層内の位置を指定します。

parent_element

スキーマ階層の最上位の親要素です。

次の例では、chapter サブツリーのシノニムが作成されます。chapter は book の子要素で、book は xmlbook スキーマ階層の最上位の要素です。

```
CREATE SYNONYM chapter FOR xmlbook.xsd
DBMS XML
AT baseapp
PARMS POSITION=/book/chapter
DROP
END
```

アクセスファイルに XPATH=/book が追加されて、スキーマ階層内の chapter の位置 (シノニムでのルートセグメント) が指定されます。

シノニムに基づいて TABLE リクエストを発行する場合、または XML を記述する場合は、ノードの実際のパスを指定する必要があります。chapter は元のスキーマでのルートノードではないため、スキーマの最上位から chapter までのパスは /book/chapter になります。

シノニムに基づいて XML ファイルを記述する場合、chapter 要素は `<book><chapter>...</chapter></book>` のように book 要素で囲みます。

シノニムに基づいて XML ファイルのレポートを作成する場合、chapter は book の直下のため、book は無視する必要があります。

シノニムの管理

シノニムを作成後、WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのナビゲーションウィンドウでシノニム名を右クリックし、次のオプションにアクセスすることができます。

オプションのリストについては、[シノニムの管理オプション](#)を参照してください。

リレーショナル DBMS の XML データタイプからの XML ドキュメントへのアクセス

XML ドキュメントは、任意のデータソースのフィールドまたはカラムに保存される場合があります。これらのドキュメントからのレポート作成は、元のデータを記述する親セグメントに関連付けられたサブツリーとしてドキュメント構造を定義することでサポートされます。

シノニムの作成プロセスは、DBMS 内のデータに対して実行するか、XML ドキュメントに対して実行する必要があります。次に、2つのマスターファイルを組み合わせて、XML マスターファイルを DBMS に対して作成されたマスターファイルの子としなければなりません。この場合、FILEDEF は必要ありません。

WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのツールを使用して RDBMS から XML データにアクセスするには

手順

1. WebFOCUS Reporting Server ブラウザインターフェースまたは ibi Data Migrator デスクトップインターフェースのシノニム作成機能を使用して、XML データのフィールドを含む RDBMS データソース用シノニムを生成します。XML データは、ネイティブデータソースに格納するために使用するデータタイプとは関係なく、マスターファイルのシノニムに TX フィールドとしてマッピングされます。たとえば、Db2 V9 では XML データタイプは TX に、多くの RDBMS では CLOB データタイプは TX に、それぞれマッピングされます。
2. 生成されたマスターファイルをシノニムエディタで開きます。マスターファイルは [テキスト表示] の右側ウィンドウに表示されます。たとえば、Progress データソースのマスターファイルは次のように表示されます。

```
FILE=XMLPRO1 ,SUFFIX=SQLPRO ,  
SEGNAME=XMLPRO1 ,SEGTYPE=S0 ,  
FIELD=FLD1 ,FLD1 ,A2 ,A2 ,MISSING=ON ,  
FIELD=FLD2 ,FLD2 ,TX50 ,TX ,MISSING=ON ,  
FIELD=FLD3 ,FLD3 ,TX50 ,TX ,MISSING=ON ,
```

この例では、2 つの TX フィールドが存在します。フィールドにはそれぞれ異なるデータが格納され、マスターファイルでは別個のセグメント宣言が必要です。

3. ibi Data Migrator デスクトップインターフェースで、「TX」として記述されているフィールドをクリックします。ポップアップメニューに [外部 XML のマッピング] オプションが表示されます。

[外部 XML のマッピング] オプションを選択すると、XML データが直接読み取られ、その構造が作成されます。結果のマスターファイルには、「SEGSUF=XML」という新しいセグメントとして表現される XML データ定義が格納されます。これは、[テキスト表

示] モードで、元の RDBM セグメントの後に表示されます。マスターファイルは次のように表示されます。

```
FILE=XMLPRO1      ,SUFFIX=SQLPRO  ,$
SEGNAME=XMLPRO1  ,SEGTYPE=S0      ,$
FIELD=FLD1 ,FLD1 ,A2      ,A2      ,MISSING=ON  ,$
FIELD=FLD2 ,FLD2 ,TX50 ,TX      ,MISSING=ON  ,$
FIELD=FLD3 ,FLD33,TX50 ,TX      ,MISSING=ON  ,$
SEGMENT=ORDER, SEGTYPE=S0,POSITION=FLD2,PARENT=XMLPRO1,SEGSUF=XML, $
  FIELDNAME=ORDER, ALIAS='Order', USAGE=A1, ACTUAL=A1, $
  FIELDNAME=KEY, ALIAS='key', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=CUSTOMER, ALIAS='Customer', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=STATUS, ALIAS='Status', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=TOTALPRICE, ALIAS='TotalPrice', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=DATE, ALIAS='Date', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=PRIORITY, ALIAS='Priority', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=CLERK, ALIAS='Clerk', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=SHIPPRIORITY,ALIAS='ShipPriority',USAGE=A10,ACTUAL=A10,$
  FIELDNAME=COMMENT, ALIAS='Comment', USAGE=A10, ACTUAL=A10, $
SEGMENT=POORDER, SEGTYPE=S0,POSITION=FLD3,PARENT=XMLPRO1,SEGSUF=XML $
  FIELDNAME=POORDER, ALIAS='Order', USAGE=A1, ACTUAL=A1, $
  FIELDNAME=KEY, ALIAS='key', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=CUSTOMER, ALIAS='Customer', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=ADDRESS, ALIAS='Address', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=STATE, ALIAS='State', USAGE=A10, ACTUAL=A10, $
```

ヒント：名前フィールドや住所フィールドなど、TX フィールドに複数の XML フォーマットが存在する場合、[外部 XML のマッピング] を複数回選択し、それぞれのフォーマットに別の SEGSUF=XML セグメントを作成することができます。

4. シノニムエディタの [ファイル] メニューから [保存] を選択し、更新したマスターファイルを保存します。

RDBMS から XML データに手動でアクセスするには

XML データを格納する、1 つまたは複数の列を持つテーブルが RDBMS 内に存在するとします。この XML データからレポートを作成するには、次の手順を実行します。

手順

1. 対象の DBMS のフォーマットを使用して、リレーショナルデータソースのマスター

ファイルを作成します。

```
FILE=XMLPRO1 ,SUFFIX=SQLPRO , $
SEGNAME=XMLPRO1 ,SEGTYPE=S0 , $
FIELD=FLD1 ,FLD1 ,A2 ,A2 ,MISSING=ON , $
FIELD=FLD2 ,FLD2 ,TX50 ,TX ,MISSING=ON , $
FIELD=FLD3 ,FLD3 ,TX50 ,TX ,MISSING=ON , $
```

2. RDBMS テーブルの列内の XML ドキュメント用のマスターファイルを作成します。異なるフォーマットの 2 つの XML ドキュメントがある場合、それぞれにマスターファイルを作成する必要があります。
3. 各マスターファイルを手動で結合します。XML マスターファイルの各ルートセグメントに、POSITION、PARENT、SEGSUF という 3 つのフィールドを追加します。POSITION キーワードでは、XML ドキュメントを含むフィールドを特定します。PARENT フィールドでは、元のデータソースを定義します。フィールド SEGSUF では、サブツリーを表す XML ドキュメントのルートセグメントを定義します。マスターファイル内のすべてのフィールドの長さの合計は、FOCUS の制限である 32 キロバイトを超えてはなりません。超えた場合、クエリはエラーになります。

```
FILENAME=BASEAPP/ORDER, SUFFIX=XML , $
SEGMENT=ORDER, SEGTYPE=S0, $
  FIELDNAME=ORDER, ALIAS='Order', USAGE=A1, ACTUAL=A1, $
  FIELDNAME=KEY, ALIAS='key', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=CUSTOMER, ALIAS='Customer', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=STATUS, ALIAS='Status', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=TOTALPRICE, ALIAS='TotalPrice', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=DATE, ALIAS='Date', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=PRIORITY, ALIAS='Priority', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=CLERK, ALIAS='Clerk', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=SHIPPRIORITY, ALIAS='ShipPriority', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=COMMENT, ALIAS='Comment', USAGE=A10, ACTUAL=A10, $
FILENAME=BASEAPP/PORDER, SUFFIX=XML , $
SEGMENT=PORDER, SEGTYPE=S0, $
  FIELDNAME=PORDER, ALIAS='POrder', USAGE=A1, ACTUAL=A1, $
  FIELDNAME=KEY, ALIAS='key', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=CUSTOMER, ALIAS='Customer', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=ADDRESS, ALIAS='Address', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=STATE, ALIAS='State', USAGE=A10, ACTUAL=A10, $
```

結合されたマスターファイル

```
FILE=XMLPRO1 ,SUFFIX=SQLPRO , $
```

```

SEGNAME=XMLPRO1 ,SEGTYPE=S0      , $
FIELD=FLD1 ,FLD1 ,A2      ,A2      ,MISSING=ON , $
FIELD=FLD2 ,FLD2 ,TX50 ,TX      ,MISSING=ON , $
FIELD=FLD3 ,FLD33,TX50 ,TX      ,MISSING=ON , $
SEGMENT=ORDER, SEGTYPE=S0,POSITION=FLD2,PARENT=XMLPRO1,SEGSUF=XML, $
  FIELDNAME=ORDER, ALIAS='Order', USAGE=A1, ACTUAL=A1, $
  FIELDNAME=KEY, ALIAS='key', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=CUSTOMER, ALIAS='Customer', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=STATUS, ALIAS='Status', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=TOTALPRICE, ALIAS='TotalPrice', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=DATE, ALIAS='Date', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=PRIORITY, ALIAS='Priority', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=CLERK, ALIAS='Clerk', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=SHIPPRIORITY,ALIAS='ShipPriority',USAGE=A10,ACTUAL=A10,$
  FIELDNAME=COMMENT, ALIAS='Comment', USAGE=A10, ACTUAL=A10, $
SEGMENT=POORDER, SEGTYPE=S0,POSITION=FLD3,PARENT=XMLPRO1,SEGSUF=XML $
  FIELDNAME=POORDER, ALIAS='Order', USAGE=A1, ACTUAL=A1, $
  FIELDNAME=KEY, ALIAS='key', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=CUSTOMER, ALIAS='Customer', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=ADDRESS, ALIAS='Address', USAGE=A10, ACTUAL=A10, $
  FIELDNAME=STATE, ALIAS='State', USAGE=A10, ACTUAL=A10, $

```

静的 JOIN の使用

マスターファイルおよびアクセスファイルを使用して、同一の物理 XML ドキュメントのさまざまなビューを記述することができます。

- マスターファイルでは、フィールド定義に REFERENCE 属性を使用して、XML ドキュメント階層のタグと PARENT 属性との間の物理関係を表します。これにより、論理的な階層関係が確立されます。
- アクセスファイルでは、KEYFLD および IXFLD 属性を使用して、XML ドキュメント階層内の主キーまたは外部キーとして動作する XML タグを識別します。これにより、論理的な JOIN 関係が確立されます。KEYFLD で定義した親フィールド (外部キー) では、クロスリファレンスの値を提供します。IXFLD で定義した下位フィールド (主キー) には、それに対応する値が格納されます。

アダプタは、実行時に値が一致した JOIN を実装します。

静的 JOIN を使用すると、データノードを任意に組み合わせ、階層的に無関係な統合スキーマ ComplexType 定義の間に JOIN 関係を作成することができます。

これらの定義に属する任意の XML タグを使用して、JOIN の組み合わせを作成することができます。また、必要に応じて、同一物理セグメントのインスタンスを増加させることで、論理的な JOIN 関係を表すこともできます。

注意：静的 (埋め込み) JOIN は、シノニムの作成機能で直接使用することはできません。この機能を利用するには、シノニムの作成後にそのマスターファイルおよびアクセスファイルを編集する必要があります。この編集を行う場合は、ibi Data Migrator デスクトップインターフェースのシノニムエディタを使用することをお勧めします。このツールを使用すると、簡単なドラッグアンドドロップ操作でセグメント階層を分かりやすく表示したり、KEYFLD および IXFLD を編集する計算を視覚的に実行したりすることができます。ibi Data Migrator デスクトップインターフェースを使用すると、すべての重複セグメントをマスターファイルに追加した後、不必要なセグメントを削除する操作をすばやく実行することができます。

場合によっては、JOIN の記述に適切な XML の制約 (unique/key/keyref) がスキーマに含まれる場合もあります (例、Microsoft ツールで作成され、元の Microsoft SQL Server データベーステーブル間の関係を表すスキーマ)。この情報を使用して、マスターファイルを次のように編集することができます。

- 要素「unique」および「key」は、要素 (セグメント) の主キーを定義します。
- 要素「keyref」は、要素 (セグメント) の外部キーを定義します。

そのようなスキーマの例については、[SQL Server によってエクスポートされたスキーマから生成されたマスターファイルとアクセスファイルの変更](#)を参照してください。

SQL Server によってエクスポートされたスキーマから生成されたマスターファイルとアクセスファイルの変更

2つのリンクが、Employee と EmployeeTL という要素 (マスターファイルの EMPLOYEE と EMPLOYEEETL というセグメント) 間の GetAllEmployeesFullDS.xml ファイル内に定義されたスキーマに存在するとします。以下の2つのバリエーションがあります。

EId (ペア) による EmployeeTL と Employee の JOIN

この例では、セグメント EMPLOYEE はマスターファイルのセグメント EMPLOYEEETL の子となり、アクセスファイルは、外部キーと主キーを次のように記述します。

```
<xs:unique name="Constraint1" msdata:PrimaryKey="true">
  <xs:selector xpath=" ../Employee" />
```

```

    <xs:field xpath="EId" />
</xs:unique>

<xs:keyref name="EmployeeTl_Employee_EId_EId" refer="Constraint1"
msdata:ConstraintOnly="true">
    <xs:selector xpath="..//EmployeeTl" />
    <xs:field xpath="EId" />
</xs:keyref>

```

アクセスファイル

```
SEGNAME=EMPLOYEE, KEYFLD=EID, IXFLD=EID, $
```

TLInformation と EtId による Employee と EmployeeTL の JOIN

この例では、セグメント EMPLOYEETL はマスターファイルのセグメント EMPLOYEE の子となり、アクセスファイルには、外部キーと主キーを次のように記述します。

```

<xs:unique name="EmployeeTl_Constraint1"
msdata:ConstraintName="Constraint1" msdata:PrimaryKey="true">
    <xs:selector xpath="..//EmployeeTl" />
    <xs:field xpath="EtId" />
</xs:unique>

<xs:keyref name="Employee_EmployeeTl_EtId_TLInformation"
refer="EmployeeTl_Constraint1" msdata:ConstraintOnly="true">
    <xs:selector xpath="..//Employee" />
    <xs:field xpath="TLInformation" />
</xs:keyref>

```

アクセスファイル

```
SEGNAME=EMPLOYEEETL, KEYFLD=TLINFORMATION, IXFLD=ETID, $
```

JOIN 構造の例 - 元の構造と変更後の構造

この例では、静的 JOIN を使用していない元の構造、および静的 JOIN を使用する変更後の構造を示します。

静的 JOIN を使用しない元の構造

```

GETALLEMPLOYEEERESULT
01      S0
*****
*GETALLEMPLO>+
*GETALLEMPLO>+
*DIFFGRAM  +
*DSEMPLOYEE  +
+
*****
*****
I
+-----+-----+-----+-----+-----+-----+
I      I      I      I      I      I      I
I DATASETGENERATOR I EMPLOYEE      I EMPLOYEEEL      I ADDRESS      I CITY      I EMPLOYEEERELATIVE
02      I S0      03      I S0      04      I S0      05      I S0      06      I S0      07      I S0
*****
*DATASETGENE>+ *EMPLOYEE  + *EMPLOYEEEL  + *ADDRESS  + *CITY  + *EMPLOYEEEREL>+
*OBJECTTYPE  + *TINFORMATION + *ETID  + *AID  + *CITYID + *RID  +
*ISCOLLECTION+ *EID  + *AIDOFFICIAL + *  + *  + *  +
*OBJECTID  + *CITYIDBIRTH + *AIDRESIDENTI + *CITYID  + *  + *  +
+ + + + +
*****
*****

```

この元の構造では、すべての子セグメントに共通の親 (ルートセグメントの GETALLEMPLOYEEERESULTS) があります。

シノニムの一部として作成された対応するマスターファイル内では、各セグメント宣言の PARENT 属性によってセグメント間の論理関係が記述され、各フィールドの REFERENCE 属性によってファイル内の要素間の物理的な親子関係の情報が記述されます。

作成されたマスターファイル

```
FILENAME=GETALLEMPLOYEEFULLDS, SUFFIX=XML,
  DATASET=C:\Users\yn05149\apps\xml\GetAllEmployeesFullDS.xml, $
$
```

```
SEGMENT=GETALLEMPLOYEEERESULT, SEGTYPE=S0, $
  FIELDNAME=GETALLEMPLOYEEERESULT, ALIAS=GetAllEmployeesResponse, USAGE=A1,
    ACTUAL=A1, ACCESS_PROPERTY=(INTERNAL), $
  FIELDNAME=GETALLEMPLOYEEERESULT1, ALIAS=GetAllEmployeesResult, USAGE=A1,
    ACTUAL=A1, ACCESS_PROPERTY=(INTERNAL),
    REFERENCE=GETALLEMPLOYEEERESULT, PROPERTY=ELEMENT, $
  FIELDNAME=DIFFGRAM, ALIAS=diffgram, USAGE=A1, ACTUAL=A1,
    ACCESS_PROPERTY=(INTERNAL),
    REFERENCE=GETALLEMPLOYEEERESULT1, PROPERTY=ELEMENT, $
  FIELDNAME=DSEMPLOYEE, ALIAS=DSEmployee, USAGE=A1, ACTUAL=A1,
    ACCESS_PROPERTY=(INTERNAL),
    REFERENCE=DIFFGRAM, PROPERTY=ELEMENT, $
$
```

```
SEGMENT=DATASETGENERATOR, SEGTYPE=S0, PARENT=GETALLEMPLOYEEERESULT, $
  FIELDNAME=DATASETGENERATOR, ALIAS=DataSetGenerator, USAGE=A1, ACTUAL=A1,
```

```

    ACCESS_PROPERTY=(INTERNAL),
    REFERENCE=DSEMPLOYEE, PROPERTY=ELEMENT, $
    FIELDNAME=OBJECTTYPE, ALIAS=ObjectType, USAGE=A30, ACTUAL=A30,
    REFERENCE=DATASETGENERATOR, PROPERTY=ELEMENT, $
    FIELDNAME=ISCOLLECTION, ALIAS=IsCollection, USAGE=A5, ACTUAL=A5,
    REFERENCE=DATASETGENERATOR, PROPERTY=ELEMENT, $
    FIELDNAME=OBJECTID, ALIAS=ObjectId, USAGE=I11, ACTUAL=A11,
    REFERENCE=DATASETGENERATOR, PROPERTY=ELEMENT, $
$

```

```

SEGMENT=EMPLOYEE, SEGTYPE=S0, PARENT=GETALLEMPLOYEESRESULT, $
    FIELDNAME=EMPLOYEE, ALIAS=Employee, USAGE=A1, ACTUAL=A1,
    ACCESS_PROPERTY=(INTERNAL),
    REFERENCE=DSEMPLOYEE, PROPERTY=ELEMENT, $
    FIELDNAME=EID, ALIAS=EIId, USAGE=I11, ACTUAL=A11,
    REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
    FIELDNAME=CITYIDBIRTHPLACE, ALIAS=CityIdBirthplace, USAGE=I11,
    ACTUAL=A11,
    REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
    FIELDNAME=TLID, ALIAS=TLId, USAGE=I11, ACTUAL=A11,
    REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
    FIELDNAME=EFIRSTNAME, ALIAS=EFirstname, USAGE=A30, ACTUAL=A30,
    REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
    FIELDNAME=ENAME, ALIAS=EName, USAGE=A30, ACTUAL=A30,
    REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
    FIELDNAME=EBIRTHDATE, ALIAS=EBirthdate, USAGE=HYMDm, ACTUAL=A35,
    REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
    FIELDNAME=IEID, ALIAS=IEId, USAGE=I11, ACTUAL=A11,
    REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
    FIELDNAME=TLINFORMATION, ALIAS=TLInformation, USAGE=I11, ACTUAL=A11,
    REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
$

```

```

SEGMENT=EMPLOYEEETL, SEGTYPE=S0, PARENT=GETALLEMPLOYEESRESULT, $
    FIELDNAME=EMPLOYEEETL, ALIAS=EmployeeTl, USAGE=A1, ACTUAL=A1,
    ACCESS_PROPERTY=(INTERNAL),
    REFERENCE=GETALLEMPLOYEESRESULT.DSEMPLOYEE, PROPERTY=ELEMENT, $
    FIELDNAME=ETID, ALIAS=EtId, USAGE=I11, ACTUAL=A11,
    REFERENCE=EMPLOYEEETL, PROPERTY=ELEMENT, $
    FIELDNAME=EID, ALIAS=EIId, USAGE=I11, ACTUAL=A11,
    REFERENCE=EMPLOYEEETL, PROPERTY=ELEMENT, $
    FIELDNAME=AIDOFFICIAL, ALIAS=AIdOfficial, USAGE=I11, ACTUAL=A11,
    REFERENCE=EMPLOYEEETL, PROPERTY=ELEMENT, $
    FIELDNAME=AIDRESIDENTIAL, ALIAS=AIdResidential, USAGE=I11, ACTUAL=A11,
    REFERENCE=EMPLOYEEETL, PROPERTY=ELEMENT, $

```

```

FIELDNAME=DEPID, ALIAS=DepId, USAGE=I11, ACTUAL=A11,
REFERENCE=EMPLOYEEETL, PROPERTY=ELEMENT, $
$

```

```

SEGNAME=ADDRESS, SEGTYPE=S0, PARENT=GETALLEMPLOYEESRESULT, $
FIELDNAME=ADDRESS, ALIAS=Address, USAGE=A1, ACTUAL=A1,
PROPERTY=ELEMENT, REFERENCE=GETALLEMPLOYEESRESULT.DSEMPLOYEE,
ACCESS_PROPERTY=(INTERNAL), $
FIELDNAME=AID, ALIAS=AId, USAGE=I11, ACTUAL=A11,
PROPERTY=ELEMENT, REFERENCE=ADDRESS, $
FIELDNAME=CITYID, ALIAS=CityId, USAGE=I11, ACTUAL=A11,
PROPERTY=ELEMENT, REFERENCE=ADDRESS, $
FIELDNAME=AADDRESS, ALIAS=AAddress, USAGE=A30, ACTUAL=A30,
PROPERTY=ELEMENT, REFERENCE=ADDRESS, $
FIELDNAME=AADDRESSNMBR, ALIAS=AAddressNmbr, USAGE=A30, ACTUAL=A30,
PROPERTY=ELEMENT, REFERENCE=ADDRESS, $
FIELDNAME=ATEL1, ALIAS=ATel1, USAGE=A30, ACTUAL=A30,
PROPERTY=ELEMENT, REFERENCE=ADDRESS, $
FIELDNAME=ATEL2, ALIAS=ATel2, USAGE=A30, ACTUAL=A30,
PROPERTY=ELEMENT, REFERENCE=ADDRESS, $
$

```

```

SEGNAME=CITY, SEGTYPE=S0, PARENT=GETALLEMPLOYEESRESULT, $
FIELDNAME=CITY, ALIAS=City, USAGE=A1, ACTUAL=A1, PROPERTY=ELEMENT,
REFERENCE=GETALLEMPLOYEESRESULT.DSEMPLOYEE, ACCESS_PROPERTY=(INTERNAL),
$
FIELDNAME=CITYID, ALIAS=CityId, USAGE=I11, ACTUAL=A11,
PROPERTY=ELEMENT, REFERENCE=CITY, $
FIELDNAME=COUNTRYID, ALIAS=CountryId, USAGE=I11, ACTUAL=A11,
PROPERTY=ELEMENT, REFERENCE=CITY, $
FIELDNAME=CITYNAME, ALIAS=CityName, USAGE=A30, ACTUAL=A30,
PROPERTY=ELEMENT, REFERENCE=CITY, $
FIELDNAME=CITYPOSTALCODE, ALIAS=CityPostalcode, USAGE=A30, ACTUAL=A30,
PROPERTY=ELEMENT, REFERENCE=CITY, $
$

```

```

SEGNAME=EMPLOYEERELATIVE, SEGTYPE=S0, PARENT=GETALLEMPLOYEESRESULT, $
FIELDNAME=EMPLOYEERELATIVE, ALIAS=EmployeeRelative, USAGE=A1, ACTUAL=A1,
PROPERTY=ELEMENT,
REFERENCE=GETALLEMPLOYEESRESULT.DSEMPLOYEE, ACCESS_PROPERTY=(INTERNAL),
$
FIELDNAME=ERID, ALIAS=ErId, USAGE=I11, ACTUAL=A11,
PROPERTY=ELEMENT, REFERENCE=EMPLOYEERELATIVE, $
FIELDNAME=EID, ALIAS=EId, USAGE=I11, ACTUAL=A11,

```

```

        PROPERTY=ELEMENT, REFERENCE=EMPLOYEEERELATIVE, $
        FIELDNAME=CITYIDBIRTHPLACE, ALIAS=CityIdBirthplace, USAGE=I11,
    ACTUAL=A11,
        PROPERTY=ELEMENT, REFERENCE=EMPLOYEEERELATIVE, $
        FIELDNAME=TLID, ALIAS=TLId, USAGE=I11, ACTUAL=A11,
        PROPERTY=ELEMENT, REFERENCE=EMPLOYEEERELATIVE, $
        FIELDNAME=ERLRELATIONSHIP, ALIAS=ErLRelationship, USAGE=A30, ACTUAL=A30,
        PROPERTY=ELEMENT, REFERENCE=EMPLOYEEERELATIVE, $
        FIELDNAME=ERNAME, ALIAS=ErName, USAGE=A30, ACTUAL=A30,
        PROPERTY=ELEMENT, REFERENCE=EMPLOYEEERELATIVE, $
        FIELDNAME=ERFIRSTNAME, ALIAS=ErFirstname, USAGE=A30, ACTUAL=A30,
        PROPERTY=ELEMENT, REFERENCE=EMPLOYEEERELATIVE, $
    $

```

作成されたアクセスファイル シノニムを作成すると、作成されたアクセスファイルにはエンティティの省略形のみが格納されます。ただし、このファイルを編集し、必要な変更を行うことができます。詳細は、静的 JOIN を定義する構造の説明を参照してください。

静的 JOIN を使用する変更後の構造

変更後の構造では、元の 2 レベルの構造は 5 レベルの構造に置き換わっています。この新しい構造では、同じ物理セグメント (CITY と ADDRESS) のインスタンスが、新しい論理結合の関係を反映して増加しています。


```

      GETALLEMPLOYEERESULT
01      SO
*****
*GETALLEMPLO>**
*GETALLEMPLO>**
*DIFFGRAM      **
*DSEMPLOYEE     **
*               **
*****
      I
      +-----+
      I               I
      I DATASETGENERATOR I EMPLOYEE
02      I SO          03      I SO
*****
*DATASETGENE>**      *EMPLOYEE      **
*OBJECTTYPE      **      *TLINFORMATIO**
*ISCOLLECTION**      *EID          **
*OBJECTID        **      *CITYIDBIRTHP**
*               **      *               **
*****
      I
      +-----+
      I               I               I
      I EMPLOYEE TL      I BCITY          I EMPLOYEE RELATIVE
04      I SO          09      I SO          10      I SO
*****
*EMPLOYEE TL      **      *CITY          **      *EMPLOYEE REL>**
*ETID            **      *CITYID        **      *EID          **
*AIDOFFICIAL     **      *               **      *               **
*AIDRESIDENTI    **      *               **      *               **
*               **      *               **      *               **
*****
      I
      +-----+
      I               I
      I OADDRESS      I RADDRESS
05      I SO          07      I SO
*****
*ADDRESS          **      *ADDRESS      **
*AID              **      *AID          **
*CITYID           **      *CITYID       **
*               **      *               **
*****
      I               I
      I               I
      I OCITY          I RCITY
06      I SO          08      I SO
*****
*CITY              **      *CITY          **
*CITYID            **      *CITYID       **
*               **      *               **
*               **      *               **
*****
*****

```

変更後のマスターファイル

対応するマスターファイルでは、親子関係は手動で拡張および再編成され、新しい階層全体で正確なポインタを提供するように参照が変更されています。

このマスターファイルの例では、次の点に注意してください。

- フィールド定義内の REFERENCE 属性は、XML ドキュメント階層内の要素とセグメントレベルの PARENT 属性 (論理的な階層関係を確立する) 間の物理関係を反映します。追加されたセグメント、または階層内のその位置が変更されたセグメントでは、マスターファイルの REFERENCE 属性が、通常、次の 2 つで構成されるようになることに

注意してください。

```
segment.fieldname
```

通常、アダプタはカレントセグメント内の各 REFERENCE の解決を試みます。これが不可能な場合、親のチェーン内の REFERENCE の最初のオカレンスを検索します (SEGMENT EMPLOYEE 内の REFERENCE 属性の部分参照)。修飾された (2 つの部分で構成される) 参照により、指定されたセグメントが強制的に直接解決されます (SEGMENT EMPLOYEE 内の REFERENCE 属性の部分参照)。この例では、いずれの表記も正しく、同じ結果が生成されます。

- マスターファイル内の ALIAS 属性は、ネイティブの XML スキーマの要素名にマッピングされるため、変更することはできません。ただし、FIELDNAME は常に ALIAS に関連付けられているため、フィールド名は参照を壊すことなく変更可能です。
- REFERENCE 属性では、フィールド名を指示します。このため、フィールド名が変更されると、このフィールド名が指定されている参照をそれに応じてすべて変更する必要があります。

```
FILENAME=GETALLEMPLOYEEFULLDS, SUFFIX=XML,
DATASET=C:\Users\yn05149\apps\xml\GetAllEmployeesFullDS.xml, $
$
```

```
SEGMENT=GETALLEMPLOYEESRESULT, SEGTYPE=S0, $
FIELDNAME=GETALLEMPLOYEESRESULT, ALIAS=GetAllEmployeesResponse, USAGE=A1,
ACTUAL=A1, ACCESS_PROPERTY=(INTERNAL), $
FIELDNAME=GETALLEMPLOYEESRESULT1, ALIAS=GetAllEmployeesResult, USAGE=A1,
ACTUAL=A1, ACCESS_PROPERTY=(INTERNAL),
REFERENCE=GETALLEMPLOYEESRESULT, PROPERTY=ELEMENT, $
FIELDNAME=DIFFGRAM, ALIAS=diffgram, USAGE=A1, ACTUAL=A1,
ACCESS_PROPERTY=(INTERNAL),
REFERENCE=GETALLEMPLOYEESRESULT1, PROPERTY=ELEMENT, $
FIELDNAME=DSEMPLOYEE, ALIAS=DSEmployee, USAGE=A1, ACTUAL=A1,
ACCESS_PROPERTY=(INTERNAL),
REFERENCE=DIFFGRAM, PROPERTY=ELEMENT, $
$
```

```
SEGMENT=DATASETGENERATOR, SEGTYPE=S0, PARENT=GETALLEMPLOYEESRESULT, $
FIELDNAME=DATASETGENERATOR, ALIAS=DataSetGenerator, USAGE=A1, ACTUAL=A1,
ACCESS_PROPERTY=(INTERNAL),
REFERENCE=DSEMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=OBJECTTYPE, ALIAS=ObjectType, USAGE=A30, ACTUAL=A30,
REFERENCE=DATASETGENERATOR, PROPERTY=ELEMENT, $
```

```

FIELDNAME=ISCOLLECTION, ALIAS=IsCollection, USAGE=A5, ACTUAL=A5,
REFERENCE=DATASETGENERATOR, PROPERTY=ELEMENT, $
FIELDNAME=OBJECTID, ALIAS=ObjectId, USAGE=I11, ACTUAL=A11,
REFERENCE=DATASETGENERATOR, PROPERTY=ELEMENT, $
$

```

```

SEGMENT=EMPLOYEE, SEGTYPE=S0, PARENT=GETALLEMPLOYEESRESULT, $
FIELDNAME=EMPLOYEE, ALIAS=Employee, USAGE=A1, ACTUAL=A1,
ACCESS_PROPERTY=(INTERNAL),
REFERENCE=DSEMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=EID, ALIAS=EIId, USAGE=I11, ACTUAL=A11,
REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=CITYIDBIRTHPLACE, ALIAS=CityIdBirthplace, USAGE=I11,
ACTUAL=A11,
REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=TLID, ALIAS=TLId, USAGE=I11, ACTUAL=A11,
REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=EFIRSTNAME, ALIAS=EFIRSTNAME, USAGE=A30, ACTUAL=A30,
REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=ENAME, ALIAS=ENAME, USAGE=A30, ACTUAL=A30,
REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=EBIRTHDATE, ALIAS=EBirthdate, USAGE=HYMDm, ACTUAL=A35,
REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=IEID, ALIAS=IEId, USAGE=I11, ACTUAL=A11,
REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=TLINFORMATION, ALIAS=TLInformation, USAGE=I11, ACTUAL=A11,
REFERENCE=EMPLOYEE, PROPERTY=ELEMENT, $
$

```

```

SEGMENT=EMPLOYEEETL, SEGTYPE=S0, PARENT=EMPLOYEE, $
FIELDNAME=EMPLOYEEETL, ALIAS=EmployeeTl, USAGE=A1, ACTUAL=A1,
ACCESS_PROPERTY=(INTERNAL),
REFERENCE=GETALLEMPLOYEESRESULT.DSEMPLOYEE, PROPERTY=ELEMENT, $
FIELDNAME=ETID, ALIAS=EtId, USAGE=I11, ACTUAL=A11,
REFERENCE=EMPLOYEEETL, PROPERTY=ELEMENT, $
FIELDNAME=EID, ALIAS=EIId, USAGE=I11, ACTUAL=A11,
REFERENCE=EMPLOYEEETL, PROPERTY=ELEMENT, $
FIELDNAME=AIDOFFICIAL, ALIAS=AIdOfficial, USAGE=I11, ACTUAL=A11,
REFERENCE=EMPLOYEEETL, PROPERTY=ELEMENT, $
FIELDNAME=AIDRESIDENTIAL, ALIAS=AIdResidential, USAGE=I11, ACTUAL=A11,
REFERENCE=EMPLOYEEETL, PROPERTY=ELEMENT, $
FIELDNAME=DEPID, ALIAS=DepId, USAGE=I11, ACTUAL=A11,
REFERENCE=EMPLOYEEETL, PROPERTY=ELEMENT, $
$

```

```

SEGNAME=OADDRESS, SEGTYPE=S0, PARENT=EMPLOYEE, $
  FIELDNAME=ADDRESS, ALIAS=Address, USAGE=A1, ACTUAL=A1,
    PROPERTY=ELEMENT, REFERENCE=GETALLEMPLOYEESRESULT.DSEMPLOYEE,
    ACCESS_PROPERTY=(INTERNAL), $
  FIELDNAME=AID, ALIAS=Aid, USAGE=I11, ACTUAL=A11,
    PROPERTY=ELEMENT, REFERENCE=OADDRESS.ADDRESS, $
  FIELDNAME=CITYID, ALIAS=CityId, USAGE=I11, ACTUAL=A11,
    PROPERTY=ELEMENT, REFERENCE=OADDRESS.ADDRESS, $
  FIELDNAME=AADDRESS, ALIAS=AAddress, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=OADDRESS.ADDRESS, $
  FIELDNAME=AADDRESSNMBR, ALIAS=AAddressNmbr, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=OADDRESS.ADDRESS, $
  FIELDNAME=ATEL1, ALIAS=ATel1, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=OADDRESS.ADDRESS, $
  FIELDNAME=ATEL2, ALIAS=ATel2, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=OADDRESS.ADDRESS, $
$

```

```

SEGNAME=OCITY, SEGTYPE=S0, PARENT=OADDRESS, $
  FIELDNAME=CITY, ALIAS=City, USAGE=A1, ACTUAL=A1, PROPERTY=ELEMENT,
    REFERENCE=GETALLEMPLOYEESRESULT.DSEMPLOYEE, ACCESS_PROPERTY=(INTERNAL),
$
  FIELDNAME=CITYID, ALIAS=CityId, USAGE=I11, ACTUAL=A11,
    PROPERTY=ELEMENT, REFERENCE=OCITY.CITY, $
  FIELDNAME=COUNTRYID, ALIAS=CountryId, USAGE=I11, ACTUAL=A11,
    PROPERTY=ELEMENT, REFERENCE=OCITY.CITY, $
  FIELDNAME=CITYNAME, ALIAS=CityName, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=OCITY.CITY, $
  FIELDNAME=CITYPOSTALCODE, ALIAS=CityPostalcode, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=OCITY.CITY, $
$

```

```

SEGNAME=RADDRESS, SEGTYPE=S0, PARENT=EMPLOYEE, $
  FIELDNAME=ADDRESS, ALIAS=Address, USAGE=A1, ACTUAL=A1,
    PROPERTY=ELEMENT, REFERENCE=GETALLEMPLOYEESRESULT.DSEMPLOYEE,
    ACCESS_PROPERTY=(INTERNAL), $
  FIELDNAME=AID, ALIAS=Aid, USAGE=I11, ACTUAL=A11,
    PROPERTY=ELEMENT, REFERENCE=RADDRESS.ADDRESS, $
  FIELDNAME=CITYID, ALIAS=CityId, USAGE=I11, ACTUAL=A11,
    PROPERTY=ELEMENT, REFERENCE=RADDRESS.ADDRESS, $
  FIELDNAME=AADDRESS, ALIAS=AAddress, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=RADDRESS.ADDRESS, $
  FIELDNAME=AADDRESSNMBR, ALIAS=AAddressNmbr, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=RADDRESS.ADDRESS, $
  FIELDNAME=ATEL1, ALIAS=ATel1, USAGE=A30, ACTUAL=A30,

```

```

    PROPERTY=ELEMENT, REFERENCE=RADDRESS.ADDRESS, $
    FIELDNAME=ATEL2, ALIAS=ATel2, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=OADDRESS.ADDRESS, $
$

```

```

SEGNAME=RCITY, SEGTYPE=S0, PARENT=RADDRESS, $
    FIELDNAME=CITY, ALIAS=City, USAGE=A1, ACTUAL=A1, PROPERTY=ELEMENT,
    REFERENCE=GETALLEMPLOYEESRESULT.DSEMPLOYEE, ACCESS_PROPERTY=(INTERNAL),
$
    FIELDNAME=CITYID, ALIAS=CityId, USAGE=I11, ACTUAL=A11,
    PROPERTY=ELEMENT, REFERENCE=RCITY.CITY, $
    FIELDNAME=COUNTRYID, ALIAS=CountryId, USAGE=I11, ACTUAL=A11,
    PROPERTY=ELEMENT, REFERENCE=RCITY.CITY, $
    FIELDNAME=CITYNAME, ALIAS=CityName, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=RCITY.CITY, $
    FIELDNAME=CITYPOSTALCODE, ALIAS=CityPostalcode, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=RCITY.CITY, $
$

```

```

SEGNAME=BCITY, SEGTYPE=S0, PARENT=EMPLOYEE, $
    FIELDNAME=CITY, ALIAS=City, USAGE=A1, ACTUAL=A1, PROPERTY=ELEMENT,
    REFERENCE=GETALLEMPLOYEESRESULT.DSEMPLOYEE, ACCESS_PROPERTY=(INTERNAL),
$
    FIELDNAME=CITYID, ALIAS=CityId, USAGE=I11, ACTUAL=A11,
    PROPERTY=ELEMENT, REFERENCE=BCITY.CITY, $
    FIELDNAME=COUNTRYID, ALIAS=CountryId, USAGE=I11, ACTUAL=A11,
    PROPERTY=ELEMENT, REFERENCE=BCITY.CITY, $
    FIELDNAME=CITYNAME, ALIAS=CityName, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=BCITY.CITY, $
    FIELDNAME=CITYPOSTALCODE, ALIAS=CityPostalcode, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=BCITY.CITY, $
$

```

```

SEGNAME=EMPLOYEERELATIVE, SEGTYPE=S0, PARENT=EMPLOYEE, $
    FIELDNAME=EMPLOYEERELATIVE, ALIAS=EmployeeRelative, USAGE=A1, ACTUAL=A1,
    PROPERTY=ELEMENT,
    REFERENCE=GETALLEMPLOYEESRESULT.DSEMPLOYEE, ACCESS_PROPERTY=(INTERNAL),
$
    FIELDNAME=ERID, ALIAS=ErId, USAGE=I11, ACTUAL=A11,
    PROPERTY=ELEMENT, REFERENCE=EMPLOYEERELATIVE.EMPLOYEERELATIVE, $
    FIELDNAME=EID, ALIAS=EId, USAGE=I11, ACTUAL=A11,
    PROPERTY=ELEMENT, REFERENCE=EMPLOYEERELATIVE.EMPLOYEERELATIVE, $
    FIELDNAME=CITYIDBIRTHPLACE, ALIAS=CityIdBirthplace, USAGE=I11,
    ACTUAL=A11,

```

```

    PROPERTY=ELEMENT, REFERENCE=EMPLOYEEERELATIVE.EMPLOYEEERELATIVE, $
    FIELDNAME=TLID, ALIAS=TlId, USAGE=I11, ACTUAL=A11,
    PROPERTY=ELEMENT, REFERENCE=EMPLOYEEERELATIVE.EMPLOYEEERELATIVE, $
    FIELDNAME=ERLRELATIONSHIP, ALIAS=ErLRelationship, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=EMPLOYEEERELATIVE.EMPLOYEEERELATIVE, $
    FIELDNAME=ERNAME, ALIAS=ErName, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=EMPLOYEEERELATIVE.EMPLOYEEERELATIVE, $
    FIELDNAME=ERFIRSTNAME, ALIAS=ErFirstname, USAGE=A30, ACTUAL=A30,
    PROPERTY=ELEMENT, REFERENCE=EMPLOYEEERELATIVE.EMPLOYEEERELATIVE, $
$

```

変更後のアクセスファイル

変更後のアクセスファイルには、JOIN の論理構造に使用される次の各キーが示されます。

- KEYFLD で定義した親フィールド (外部キー) では、クロスリファレンスの値を提供します。
- IXFLD で定義した下位フィールド (主キー) には、それに対応する値が格納されます。

SEGNAME=EMPLOYEEETL,	KEYFLD=TLINFORMATION,	IXFLD=ETID,	\$
SEGNAME=OADDRESS,	KEYFLD=AIDOFFICIAL,	IXFLD=AID,	\$
SEGNAME=OCITY,	KEYFLD=CITYID,	IXFLD=CITYID,	\$
SEGNAME=RADDRESS,	KEYFLD=AIDRESIDENTIAL,	IXFLD=AID,	\$
SEGNAME=RCITY,	KEYFLD=CITYID,	IXFLD=CITYID,	\$
SEGNAME=BCITY,	KEYFLD=CITYIDBIRTHPLACE,	IXFLD=CITYID,	\$
SEGNAME=EMPLOYEEERELATIVE,	KEYFLD=EID,	IXFLD=EID,	\$

データタイプのサポート

シノニムの作成処理では、シノニム作成用のソースとして XSD または XML ドキュメントを使用します。

マスターファイルに ALPHA としてマッピングされる、XML ドキュメントの数値または日付フィールドに対して算術演算 (-、+、>、<) を実行しようとした場合、算術演算が文字列リテラルに対して実行されるため、期待した結果を得られない場合があります。この問題を解決するには、フィールドの USAGE 属性を以降の各セクションで説明するように変更することが必要になる場合があります。

注意：フィールドの ACTUAL 属性で定義されたデータタイプは、ALPHA にしておく必要があります。

デフォルト設定を変更するには、SET VARCHAR コマンドを発行する必要があります。

下表は、サーバが XSD データタイプをマスターファイルでマッピングする方法を示しています。これらのマッピングの一部のデフォルト設定を変更することができます。詳細は、[文字列の長さの変更](#)および [数値カラムの精度と小数点以下の桁数の変更](#)を参照してください。

XSD データタイプ	USAGE 属性	ACTUAL 属性
decimal	P20.3	A20
integer	P33	A33
nonPositiveInteger	33	A33
negativeInteger	P33	A33
boolean	A5	A5
long	P20	A20
int	I11	A11
short	I6	A6
byte	I4	A4
nonNegativeInteger	P32	A32
unsignedLong	P20	A20
unsignedInt	P10	A10
unsignedShort	I5	A5
unsignedByte	I4	A4
positiveInteger	P32	A32
double	D20.2	A20
float	F15.2	A15

XSD データタイプ	USAGE 属性	ACTUAL 属性
dateTime	HHYMDm	A27
time	HHISsm	A15
date	YYMD	A10
gYearMonth	HHYM	A8
gYear	HHY	A5
gMonthDay	HMD	A6
gDay	HD	A3
gMonth	HM	A4
string	A30	A30
normalizedString	A30	A30
token	A30	A30
Name	A30	A30
NMTOKEN	A30	A30
ID	A30	A30
hexBinary	A30	A30
language	A30	A30
anyURI	A30	A30
QName	A30	A30

ACTUAL および USAGE 属性の設定

```
ENGINE XML SET VARCHAR {ON|OFF}
```

マスターファイルの各フィールドの ACTUAL および USAGE 属性は、A10 に設定されています。この設定は、SET FIELDLENGTH コマンドを使用して変更することができます。

```
ENGINE XML SET FIELDLENGTH nnn
```

説明

nnn

シノニム作成処理時に、マスターファイル内のすべてのフィールドの ACTUAL 属性および USAGE 属性に割り当てる長さを指定します。最大の長さは 3000 です。

文字列の長さの変更

デフォルト設定では、XML アダプタでシノニムを作成すると、XML スキーマ内で文字列として定義されているフィールドは、マスターファイル内にすべて 10 バイトの固定長の文字データタイプとしてマッピングされます。この長さ (可変長とバイト数) は、SET VARCHAR および SET LENGTH コマンドを使用して変更することができます。

可変長と固定長の文字列の切り替え

デフォルト設定では、XML アダプタでシノニムを作成すると、XML スキーマ内で文字列として定義されているフィールドは、マスターファイルでは固定長の文字データタイプとしてマッピングされます。このデフォルトの固定長は、SET VARCHAR コマンドを使用して可変長に変更することができます。

```
ENGINE XML SET VARCHAR {ON|OFF}
```

説明

XML

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

ON

XML スキーマ内で文字列として定義されているフィールドは、マスターファイルの USAGE 属性および ACTUAL 属性で可変長の文字データタイプ (AnV) としてマッピングされます。

OFF

XML スキーマ内で文字列として定義されているフィールドは、マスターファイルの USAGE 属性および ACTUAL 属性で固定長の文字データタイプ (A) としてマッピングされます。これがデフォルト値です。

文字列長さの変更

デフォルト設定では、XML アダプタでシノニムを作成すると、XML スキーマ内で文字列として定義されているフィールドは、マスターファイルで 10 バイトの文字データタイプとしてマッピングされます。このデフォルトの長さは、SET FIELDLENGTH コマンドを使用して変更することができます。

```
ENGINE XML XML SET FIELDLENGTH length
```

説明

XML

アダプタを指定します。SET SQLENGINE コマンドをすでに発行している場合は、この値を省略することができます。

length

XML スキーマ内で文字列として定義されているすべてのフィールドの ACTUAL 属性および USAGE 属性に割り当てる長さ (バイト単位) を指定します。最大の長さは 3000 です。

数値カラムの精度と小数点以下の桁数の変更

SELECT リクエストによってサーバに返される数値カラムの長さや小数点以下の桁数は、ログインプロファイルまたはストアードプロシジャで別の長さや小数点以下の桁数を指定することで変更することができます。変換の設定は、CREATE SYNONYM によって生成されたフィールドの USAGE と ACTUAL のフォーマットでマスターファイル内に反映されます。これは、サーバによるフィールドの処理とフォーマットの方法に影響を与えます。

ヒント：この設定は、手動で変更することも、WebFOCUS Reporting Server ブラウザインターフェースで変更することもできます。

詳細は、[デフォルトの精度と小数点以下の桁数の上書き変更](#)を参照してください。

変換

XML ドキュメント内のデータには、日付または数値が反映される場合がありますが、マスターファイルのシノニム内のフィールドは、すべて ALPHA データタイプに設定されます。

数値

数値フィールドに対する算術演算を有効にするには、数値フィールドの USAGE 属性に指定されたデータタイプを変更する必要があります。この変更では、XML ドキュメント内のデータに応じて、整数 (Integer) (I)、倍精度浮動小数点数 (Double Float) (D)、またはパック 10 進数 (Decimal) (P) のいずれかを指定します。Integer (I), Double Float (D) or Decimal (P). データタイプを倍精度浮動小数点数 (D) またはパック 10 進数 (P) に変更する場合、必要に応じて小数点以下の桁数および精度を使用して XML ドキュメント内のデータを記述します。

また、数値フィールドの ACTUAL 属性に指定された ALPHA データタイプの長さを XML ドキュメント内のデータの最大長に合わせて変更することをお勧めします。

XML の日付

日付に対する算術演算を有効にするには、日付フィールドの USAGE 属性に指定されたデータタイプを変更する必要があります。この変更では、XML ドキュメント内で使用されている

日付フォーマットに応じて、YYMD、MDYY、または DMYY のいずれかを指定します。

また、日付フィールドの ACTUAL 属性に指定された ALPHA データタイプの長さを 10 に設定する必要があります。

注意：

- マスターファイルでデータタイプを日付データタイプのいずれかに設定した場合、レポートの作成元となる XML ドキュメントのそれぞれで、日付フォーマットがマスターファイルで定義した日付フォーマットに一致していることを確認する必要があります。データタイプが異なる場合、エラーが発生します。
- アンサーセット内の日付フォーマットでは、XML ドキュメント内で使用されている日付フォーマットが使用されず、常に YYYY/MM/DD に設定されます。

XML での日付の使用

XML ドキュメント内の日付フォーマットが以下の場合	使用する USAGE=
1996-01-30	YYMD
01-30-1996	MDYY
30-01-1996	DMYY

XML ドキュメントとマスターファイルの関連付け

XML ドキュメントとマスターファイルの関連付けには 2 つの方法があります。マスターファイルで、XML ドキュメントを指定する DATASET 属性を特定することも、FILEDEF コマンドを明示的に発行して XML ドキュメントとマスターファイルを関連付けることもできます。

FILEDEF コマンドを適切に記述するためには、XML ドキュメントの編成方法を決定する必要があります。ドキュメントが単一のファイルに格納され、同じ DTD で記述されている場合、その編成を「カラム (COLUMN)」と呼びます。各種ドキュメントが単一のディレクトリに格

納され、すべて同じ DTD で記述されている場合は、その編成を「コレクション (COLLECTION)」と呼びます。

ソース	コレクション編成	カラム編成
disk	サポートされています	サポートされています
http	サポートされていません	サポートされています
https	サポートされていません	サポートされています

FILEDEF を使用した XML ドキュメントとマスターファイルの関連付け

```
FILEDEF ddname source filename
```

説明

ddname

必要なマスターファイルと一致する論理参照名です。

source

XML ドキュメントのロケーションです。有効な値は http、https、disk です。

filename

XML カラムでは、ソースファイルへのフルパスです。

XML コレクションでは、ソースディレクトリへのフルパスです。ディレクトリ名の後に、ワイルドカード文字として「*.*」を付加する必要があります。

HTTP または HTTPS セッションを使用して XML ドキュメントを読み込む場合、ファイル名の先頭を 'http://' または 'https://' で始める必要があります。

XML カラムに対する FILEDEF コマンドの発行

注意：WebFOCUS Reporting Server ブラウザインタフェースまたは ibi Data Migrator デスクトップインタフェースの [シノニムの作成] ウィンドウで作業を行う場合、FILEDEF の作成方法についての詳細は、[XML のシノニム作成パラメータ](#)を参照してください。

Windows C:¥ に ordercol.xml ドキュメントが格納されている場合、次の FILEDEF をプロファイルで使用します。

```
FILEDEF ordercol disk C:¥ordercol.xml
```

UNIX /usera/xml/ に ordercol.xml ドキュメントが格納されている場合、次の FILEDEF をプロファイルで使用します。

```
FILEDEF ordercol DISK /usera/xml/ordercol.xml
```

HTTP および HTTPS http://www.test.com/xmltest/ (または https://www.test.com/xmltest/) に ordercol.xml ドキュメントが格納されている場合、次の FILEDEF をプロファイルで使用します。

```
FILEDEF order http http://www.test.com/xmltest/ordercol.xml
```

または

```
FILEDEF order http https://www.test.com/xmltest/ordercol.xml
```

XML コレクションに対する FILEDEF コマンドの発行

注意：WebFOCUS Reporting Server ブラウザインタフェースまたは ibi Data Migrator デスクトップインタフェースの [シノニムの作成] ウィンドウで作業を行う場合、FILEDEF の作成方法についての詳細は、[XML のシノニム作成パラメータ](#)を参照してください。

Windows C:¥ORDERS に order1.xml および order2.xml の各ドキュメントが格納されている場合、次の FILEDEF をプロファイルで使用します。

```
FILEDEF order disk C:¥orders¥*.*
```

UNIX /usera/orders に ordercol.xml ドキュメントが格納されている場合、次の FILEDEF をプロファイルで使します。

```
FILEDEF ordercol DISK //'TEST.XML.DOC(dtd)'
```

XA のサポート

XA トランザクション管理機能および実装環境について説明します。

ここでは、XA トランザクション管理機能および実装環境について説明します。

XA トランザクション管理

アプリケーションでは通常、任意のリソースに対して実行される手順がトランザクションとして定義されます。

トランザクションとは、次のような完全作業単位のことです。

- 結果がすべてコミットまたはロールバックされる。
- 共有リソースが、有効な状態から別の有効な状態に変換される。
- 共有リソースへの変更が、トランザクションがコミットされるまでトランザクション外部で可視化されない。
- コミットの変更が、システム障害またはメディア障害によって妨害されない。

サーバは、複数のデータソースにアクセスでき、異なるデータソース間の異種 JOIN を実行できます。複数のリソース管理で実行される作業を表すトランザクションは、グローバルトランザクションまたは分散トランザクションと呼ばれます。

リレーショナルおよび非リレーショナルリソースのほとんどが、リソース管理で管理されます。X/Open 分散トランザクション処理 (DTP) モデルは、複数のアプリケーションプログラムによる、複数のリソース管理が提供するリソースへのアクセスを可能にし、これらの作業のグローバルトランザクションへの統合を可能にするソフトウェアアーキテクチャです。

CICS および IMS トランザクション、リレーショナルデータソース、CICS 制御の VSAM データセットにアクセスする読み取り/書き込みアプリケーションは、XA 互換モードで管理されるトランザクションを実行することができます。トランザクションコーディネータコンポーネントは、2 段階のコミットプロトコルを使用し、異なるデータベース管理システム間のカスタムトランザクションを完了します。

XA トランザクション管理機能の使用では、既存の ODBC、JDBC、WebFOCUS MAINTAIN アプリケーションへの変更は必要ありません。XA トランザクション管理機能を有効にするには、Web コンソールの構成機能を使用してサーバをトランザクション調整モードで構成する

必要があります。トランザクション調整モードでは、XA 互換性のある上記 DBMS に対して実行されるリクエストはすべて、XA 分散トランザクションとして処理され、2 段階コミットモードで完了されます。これにより、関連するすべての DBMS に対するデータ修正の整合性が保証されます。つまり、データ修正の一部が 1 つの DBMS に対してはコミットされ、別の DBMS に対してはコミットされないということを回避することができます。

サポート対象のインターフェース

XA 互換アダプタ

- UNIX/Windows/Linux 環境の Db2 (CLI インターフェース使用)
- Oracle
- Microsoft SQL Server (OLE DB インターフェース使用)
- Informix
- Sybase ASE

XA 非互換アダプタ

- CICS Transactions
- Informix CISAM
- Micro Focus CISAM
- IMS Transactions
- IMS
- VSAM CICS

実装

DTP モデルに準拠するためには、サーバ実装トランザクションコーディネータによるグローバルトランザクション管理機能を使用します。トランザクションコーディネータはプライベート実装ですが、この実装の範囲内で XA プロトコルに完全準拠しています。

- トランザクションコーディネータはプライベート実装のため、パブリックインターフェースは備えていません。外部アプリケーションによるトランザクションコーディ

ネータ使用のメカニズムはありません。

- 明示的な指示がない場合、XA 互換であっても、すべてのリソース管理がサポートされるとは限りません。

トランザクションコーディネータの実装環境は次のとおりです。

- edaserve.cfg のキーワード (transaction_coordination_mode=on) で呼び出されます。
- 任意のエージェント内部で開始される作業単位はすべて、このエージェント内のグローバルトランザクションとして開始されます。
- 調整は、現在サポートされているすべての XA 互換インターフェースに適用されます。
- 無効にすることはできません。
- すべての XA 互換リレーショナルリソースが含まれ、暗示的に有効化されます。
- すべての非リレーショナルリソースは、明示的に無効にし、除外することができます。
- すべてのトランザクションは、グローバルトランザクションになります。各コミットおよびロールバックの完了後にトランザクションが開始されます。
- XA モードでは、CREATE/DROP TABLE コマンドは使用できません。

ベンダー環境

UNIX、Windows での Db2 (CLI 使用) の実装環境は次のとおりです。

- DB2CLI.INI ファイルは、共通セクションにエントリを追加する必要があります。

```
[COMMON]
DISABLEMULTITHREAD=1
```

- グローバルファイルの DB2CLI.INI のみが使用できます。この場合、以下の格納先を指定する必要があります。
 - UNIX/NT の場合は、インストールホーム

MS SQL Server の実装環境は次のとおりです。

- DTS が必要です。

注意：クライアントとサーバが異なるマシン上にある場合、両方のマシンに DTS をインストールする必要があります。

Sybase の実装環境は次のとおりです。

- Sybase XA 構成ファイルが必要です。
- XA では、バルクロードおよび高速ロード機能はなく、メッセージは生成されません。挿入は標準モードで実行されます。
- ユーザは、独自の XA 構成ファイルを作成し、これをプロファイルで指定することができます。

XA 構成ファイルの作成

```
ENGINE SQLSYB SET XACONFIGFILE path
```

説明

path

ファイル名を含める必要があります。

集計の最適化のサポート

クエリの効率を大幅に向上させる、集計の最適化について説明します。

集計の最適化により、クエリの効率が大幅に改善されます。この機能を正しく使用するには、リレーショナルアダプタ処理の原理を理解し、RDBMS オプティマイザについての知識が必要になります。

リレーショナルアダプタおよび集計 SQL クエリ

リレーショナルアダプタは、WebFOCUS リクエストおよびファイルの階層構造に基づいて、最適化された SQL クエリを作成します。リレーショナルアダプタは常に、処理の大部分 (例、レコード選別、JOIN、集計) をリレーショナルエンジンに委任する SQL クエリを作成します。集計 SQL クエリを作成するには、JOIN およびすべてのテストを RDBMS に渡すインターフェースが必要です。

集計を RDBMS に渡すことができない場合は、トレースファイルに説明が書き込まれます。通常は、集計を RDBMS に渡せるように、リクエストまたはファイル階層を修正することが可能です。

RDBMS での集計の最適化

集計が RDBMS に委任されると、その RDBMS が一般に「集計の最適化」と呼ばれるメカニズムを使用して、クエリを最適化します。集計の最適化は、Db2、Teradata、Oracle などの RDBMS に実装されています。

その実装は、ほとんどの RDBMS で類似しています。データベース管理者は、データベース上で実行する一般的なレポートタイプに基づいて、事前集計されたデータを挿入する 1 つまたは複数のオブジェクトを作成します。これらのオブジェクトは、Db2 では SUMMARY TABLES、Teradata では JOIN INDEXES、Oracle では MATERIALIZED VIEWS または SNAPSHOTS と呼ばれます。通常、事前集計されたオブジェクト (SUMMARY TABLES、JOIN INDEXES、SNAPSHOTS) のサイズは、関連するテーブルを統合したサイズより相当小さくな

ります。RDBMS オプティマイザは、入力クエリを評価し、可能な場合はその入力クエリを再作成します。これにより、事前集計されたデータが、アンサーセットの作成に使用されます。その結果、CPU 時間と I/O 処理の両方が大幅に減少します。

リレーショナルアダプタで生成される集計 SQL は、事前集計されたデータを使用する RDBMS オプティマイザに完全に適合しています。ただし、Db2、Teradata、Oracle のオプティマイザには、事前集計されたデータの使用に関する特別な規則があります。

集計の最適化の設定

```
SET AGGREGATE_AWARENESS {FRESH_ONLY|OLD_OK|OFF}
```

説明

FRESH_ONLY

各 RDBMS に関連するパラメータに異なる値を設定します。

OLD_OK

各 RDBMS に関連するパラメータに異なる値を設定します。

OFF

オプションを選択しない場合、ターゲット RDBMS の動作はデータベース構成オプションに基づいて決定されます。この設定のデフォルト値はありません。

集計最適化の使用時の注意

FRESH_ONLY および OLD_OK により値が設定されるアダプタ固有のパラメータは次のとおりです。

- Db2 では、CURRENT REFRESH AGE および CURRENT QUERY OPTIMIZATION です。
- Oracle では、QUERY_REWRITE_INTEGRITY および QUERY_REWRITE_ENABLED です。

これらの値は、サーバプロファイルで設定することも、ダイレクトパススルーを使用してプロシジャの一部として設定することもできます。

- Teradata では、集計の最適化は内部的に処理されます。

推奨項目については RDBMS のマニュアルを参照する必要がありますが、一般的なチェックリスト項目は次のとおりです。

- 事前集計されたオブジェクトおよびマスターテーブルに関する統計を更新する (Db2 では RUNSTATS、Teradata では COLLECT STATISTICS、Oracle では ANALYZE TABLE)。事前集計されたオブジェクトおよびマスターテーブルに関する統計を更新する (Db2 では RUNSTATS、Teradata では COLLECT STATISTICS、Oracle では ANALYZE TABLE)。
- 生成された SQL クエリと、事前集計されたオブジェクトの作成に使用されたクエリの互換性を確認する。一般的な項目として、次の互換性を確認します。
 - 集計演算子 (カラム関数) および集計オブジェクト (カラム関数の引数)
 - GROUP BY 句
 - JOIN

クラスタ JOIN

マスターファイルの SEGSUF 属性を使用して、異種クラスタ JOIN を作成する方法について説明します。

ここでは、マスターファイルの SEGSUF 属性について説明します。この属性を使用すると、異種クラスタ JOIN (例、タイプの異なるデータソース間の JOIN) の作成が可能になります。

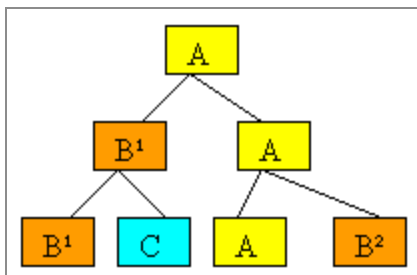
埋め込み JOIN

埋め込み JOIN は、マスターファイルで定義する等価 JOIN (EQUIJOIN) です。埋め込み JOIN という名前は、JOIN 定義がマスターファイルに埋め込まれることに由来します。この JOIN は「クラスタ JOIN」とも呼ばれます。

複数のデータソース間で埋め込み JOIN を定義することができます。

- **同一タイプのデータソース間** たとえば、2 つの Db2 テーブル間で埋め込み JOIN を定義します。これは「同種の埋め込み JOIN」と呼ばれます。
- **異なるタイプのデータソース間** たとえば、3 つの Db2 テーブル、3 つの CA-IDMS/DB、1 つの IMS セグメント間で埋め込み JOIN を定義します。これは「異種の埋め込み JOIN」と呼ばれます。

下図は、異種の埋め込み JOIN を示しています。A が Db2 テーブル、B が CA-IDMS/DB レコード、C が IMS セグメントです。



マスターファイルでは、セグメント宣言ごとに、1 つの RDBMS テーブルまたはビュー、あるいは 1 つのレコードを記述します。埋め込み JOIN には、最大で 64 個のセグメントを含めることができます。

マスターファイルで記述された JOIN 構造全体は「ツリー」と呼ばれます。

特定のセグメントと、そのセグメントに直接関連する同一データソースタイプの下位セグメントは「サブツリー」と呼ばれます。ツリー最上位のサブツリーは「ルートサブツリー」と呼ばれます。

上図の例では、4 つのサブツリーがあります。

- A-A-A
- B¹-B¹
- B²
- C

B¹-B¹ および B² は同一データソースタイプですが、直接関連していないため、それぞれ異なるサブツリーになります。

A-A-A はルートサブツリーです。

埋め込み JOIN のマスターファイル

埋め込み JOIN のセグメント宣言では、1 つの RDBMS テーブルまたはビュー、あるいは 1 つの非 RDBMS レコードを記述します。最大で 64 個のセグメントを含めることができます。

複数のマスターファイル (TABLE リクエストでのみ使用) に同一のテーブルを含める場合、同一の定義を繰り返して記述する必要はありません。マスターファイルのいずれかにテーブルを記述し、その他のマスターファイルでは CRFILE 属性を使用して既存の定義を参照します。

マスターファイルでの等価 JOIN の定義

埋め込み等価 JOIN (EQUIJOIN) では、PARENT セグメント属性を使用して、テーブル間およびレコード間の関係を識別します。

```
FILENAME=mtname, SUFFIX=rootsuffix    [, $]
SEGNAME=table1, SEGTYPE= {S0}    [, CRFILE=crfile1]    [, $]
FIELD=name, ..., $
.
.
.
```



```

SEGNAME=table2, [SEGSUF=branchsuffix,] SEGTYPE=relationship,
PARENT=table1 [,CRFILE=crfile2][,$]
FIELD=name,...,$
.
.
.

```

説明

mtname

埋め込み JOIN マスターファイルの名前です。この名前は、1 から 8 バイトまでの長さです。

rootsuffix

マスターファイルで記述する内容に応じて、次の値を指定します。

- 同種の埋め込み JOIN の場合、rootsuffix でデータソースへのアクセスに使用するアダプタを指定します。
- 異種の埋め込み JOIN の場合、rootsuffix で JOIN の最上位ツリーへのアクセスに使用するアダプタを指定します。ルートセグメントと、そのセグメントに直接関連する同一 DBMS タイプの下位セグメントで表されるテーブルまたはレコードへのアクセスに使用するアダプタを指定します。

table1

埋め込み JOIN のルートテーブルまたはルートレコードの SEGNAME 値です。このセグメントがリモートセグメント定義を参照している場合、table1 の値を、リモートデータソースの定義全体が記述されているマスターファイルの SEGNAME 値に一致させる必要があります。

name

任意のフィールド名です。

table2

関連するテーブルまたはレコードの SEGNAME 値です。このセグメントがリモートセグメント定義を参照している場合、table2 の値を、リモートデータソースの定義全体が記述されているマスターファイルの SEGNAME 値に一致させる必要があります。

SEGSUF

異種の埋め込み JOIN に使用して、別のサブツリー (ルートサブツリー以外) の先頭位置を識別するとともに、そのサブツリーへのアクセスに使用するアダプタを指定します。サブ

ツリーは、テーブルまたはレコードを表す特定のセグメントと、そのセグメントに直接関連する同一データソースタイプの下位セグメントすべてで構成されます。SEGSUF は、サブツリーの最上位セグメントで指定します。

branchsuffix

SEGSUF の値を指定します。

relationship

テーブルまたはレコードと、その親テーブルまたは親レコードとの関係を表すタイプを指定します。利用可能な値は、次のとおりです。

so - 関連するテーブルまたはレコードが、その親テーブルまたは親レコードと 1 対 n の関係、あるいは n 対 n (非ユニーク) の関係にあることを示します。

u - 関連するテーブルまたはレコードが、その親テーブルまたは親レコードと 1 対 1 の関係、あるいは n 対 1 (ユニーク) の関係にあることを示します。

KL - リモートセグメント定義を参照します。関連するテーブルまたはレコードが、その親テーブルまたは親レコードと 1 対 n の関係、あるいは n 対 n (非ユニーク) の関係にあることを示します。

KLU - リモートセグメント定義を参照します。関連するテーブルまたはレコードが、その親テーブルまたは親レコードと 1 対 1 の関係、あるいは n 対 1 (ユニーク) の関係にあることを示します。

crfile1

リモートセグメント定義を参照します。table1 の定義全体が記述されたマスターファイルの名前を指定します。

crfile2

リモートセグメント定義を参照します。table2 の定義全体が記述されたマスターファイルの名前を指定します。

COBOL ファイル記述の変換

COBOL ファイル記述によるシノニムの作成方法について説明します。

ここでは、次のアダプタを使用した COBOL ファイル記述によるシノニムの作成方法について説明します。

- C-ISAM
- Flat File および Delimited Flat File
- CICS Transactions
- IMS
- IMS Transactions
- RMS
- VSAM および VSAM CICS

ここで説明する内容は、各アダプタの章に記載したシノニム作成についての情報を補完するものです。

COBOL ファイル記述によるシノニムの作成

シノニムを作成して、サーバからアクセス可能な各ファイルの一意の名前 (またはエイリアス) を定義します。シノニムは、実際のデータソースの格納先や ID をクライアントアプリケーションから隠すことができるという点で便利です。また、シノニムは一時項目 (DEFINE) や他のセキュリティ機構などの、サーバの拡張メタデータ機能をサポートします。

シノニムを使用すると、クライアントアプリケーションを変更せずに、機能を継続させたまま、オブジェクトの移動や名前の変更が行えます。必要になるのは、サーバ上のシノニムの再定義のみです。

CREATE SYNONYM コマンドは、COBOL FD (ファイル記述) の変換機能を使用して、COBOL ファイル定義をマスターファイルの作成元となる内部メタデータフォーマットに変換します。

COBOL ファイル記述の変換の制御

さまざまなオプションを利用して、COBOL ファイル記述のマスターファイルへの変換を制御することができます。これらのオプションは、COBOL FD を使用するアダプタの [シノニムの作成] ウィンドウから利用できます。カスタマイズオプションが選択されていない場合、デフォルト変換設定が適用されます。

ここでは、いくつかのカスタマイズオプションについて補足情報を提供します。

COBOL ファイル記述のカスタマイズオプション

COBOL の FD の変換方法は、[シノニムの作成] ウィンドウの [オプションをカスタマイズ] を選択してカスタマイズすることができます。以下の各オプションが右側のウィンドウに追加されます。

パラメータ	定義
エラー時	次のいずれかを選択します。 • 続行 - エラーの発生時に、マスターファイルの生成を続行します。デフォルト値は [続行] です。 • 中断 - エラーの発生時に、マスターファイルの生成を中止します。 • コメント - エラーの発生時に、コメント付きのマスターファイルを生成します。
ハイフン処理	次のいずれかを選択します。 • No - マスターファイルのフィールド名から COBOL の名前にあるすべてのハイフンを削除します。 • Yes - COBOL の名前にあるすべてのハイフンを下線文字で置換します。デフォルト値は [Yes] です。

パラメータ	定義
再定義	COBOL の REDEFINE フィールドは、以下の 3 つの方法のいずれかで処理することができます。次のいずれかを選択します。 • セグメント - REDEFINE フィールドをマスターファイル内のセグメントとして記述します。デフォルト値は [セグメント] です。 • コメント - REDEFINE フィールドをマスターファイル内のコメントとして記述します。 • なし - REDEFINE フィールドを完全に除外します。
OCCURS	OCCURS 構造をセグメントとして記述するには、[セグメント] を選択します。セグメントとして記述しない場合は、[フィールド] を選択します。デフォルト値は [セグメント] です。
配置	次のいずれかを選択します。 • Yes - 数値フィールドの桁揃えを行えるようにレコードに遊びバイトを挿入します。 • No - 遊びバイトによる桁揃えは行わずにマスターファイルを生成します。デフォルト値は [No] です。
スキップするハイフン数	FD Translator では、左から N 番目の (メニューで選択した N の値に応じて) ハイフンまでの文字を削除します。 • 0 - COBOL の名前全体を保持します。デフォルト値は 0 です。 • All - 接頭語をすべて削除します。
ORDER フィールド	次のいずれかを選択します。 • Yes - マスターファイルに ORDER フィールドを生成します。

パラメータ	定義
	• No - マスターファイルに ORDER フィールドを生成しません。デフォルト値は [No] です。
レベル 88	次のいずれかを選択します。 • コメント - COBOL レベル 88 フィールドをコメントとしてマスターファイルに含めます。 • スキップ - レベル 88 フィールドは含めません。デフォルト値は [スキップ] です。
ゾーン形式数値フィールド	ゾーン数値の格納方法を設定します。
数値フィールド編集オプション	
ゼロ	次のいずれかを選択します。 • 非表示 - フィールドの値が 0 の場合、数字の 0 を表示しません。 • 表示 - リーディングゼロを表示します (例、00124)。 • なし - フォーマットは行いません。
ネガティブ値	次のいずれかを選択します。 • ブラケットネガティブ - 負の値を括弧で囲みます (例、(1234))。 • クレジットネガティブ - 負の値に CR を付加します (例、1234 CR)。 • なし - フォーマットは行いません。
ドル記号	次のいずれかを選択します。 • 浮動 - 浮動ドル記号とカンマを挿入します (例、\$1,123)。

パラメータ	定義
	• 固定 - 固定ドル記号とカンマを挿入します (例、\$ 1,123)。 • なし - フォーマットは行いません。
1000 の位を分離	次のいずれかを選択します。 • カンマ挿入 - 適切な位置にカンマを挿入します。 • なし - フォーマットは行いません。

REDEFINE フィールド

COBOL の REDEFINE フィールドは、以下の 3 つの方法のいずれかで処理することができます。

- マスターファイルでセグメントとして REDEFINE を記述 (セグメントの場合 S を追加)
- マスターファイルでコメントとして REDEFINE を記述 (コメントの場合 C を追加)
- REDEFINE をすべて除外 (なしの場合 N を追加)

REDEFINE をセグメントとして記述することで、すべての最上位の REDEFINE に即時アクセスできますが、ユーザのカスタマイズが必要な場合があります。ネストされた REDEFIENE は、このオプションを使用した場合もコメントとして記述されます。セグメントオプションは、これをサポートするソフトウェアのバージョンでのみ使用できます。REDEFINE をコメントとして追加することで、マスターファイルの出力をカスタマイズし、任意の再定義を選択することができます。

フィールドはコメントとして記述され、カラム 1 にドル記号 (\$) が追加されます。元の定義ではなくフィールドの再定義を使用する場合は、最初の定義を削除するか、カラム 1 にドル記号 (\$) を挿入して再定義からドル記号 (\$) を削除します。

REDEFINE フィールドの生成

次の COBOL 入力記述について考察します。

```

01 CLIENT-REC.
  02 NAME-ADDR-AREA PIC X(57).
  02 NAME-AND-ADDR
    REDEFINES NAME-ADDR-AREA.
    03 NAME PIC X(20).
    03 STREET PIC X(15).
    03 CITY PIC X(15).
    03 STATE PIC X(2).
    03 ZIP PIC X(5).
    03 ZIP_NUMERIC REDEFINES ZIP PIC 9(5).

```

この入力から、Generate REDEFINE Fields に入力した値に基づいて、次の 3 つのマスタファイルが生成されます。

値 S の場合

FILE=REDEFS,	SUFFIX=FIX,	\$
SEGNAME=CLIENSEG,	SEGTYPE=S0,	\$
GROUP=CLIENTREC,	ALIAS=E01, USAGE=A57,	ACTUAL=A57, \$
FIELD=NAMEADDRAREA,	ALIAS=E02, USAGE=A57,	ACTUAL=A57, \$
SEGNAME=NAMEASEG,	SEGTYPE=U, PARENT=CLIENSEG,	
OCCURS=1,	POSITION=NAMEADDRAREA,	\$
GROUP=NAMEANDADDR,	ALIAS=E03, USAGE=A57,	ACTUAL=A57, \$
FIELD=NAME,	ALIAS=E04, USAGE=A20,	ACTUAL=A20, \$
FIELD=STREET,	ALIAS=E05, USAGE=A15,	ACTUAL=A15, \$
FIELD=CITY,	ALIAS=E06, USAGE=A15,	ACTUAL=A15, \$
FIELD=STATE,	ALIAS=E07, USAGE=A2,	ACTUAL=A2, \$
FIELD=ZIP,	ALIAS=E08, USAGE=A5,	ACTUAL=A5, \$
\$ FIELD=ZIP_NUMERIC,	ALIAS=E09, USAGE=P5,	ACTUAL=Z5, \$

値 C の場合

FILE=REDEFC,	SUFFIX=FIX,	\$
SEGNAME=CLIENSEG,	SEGTYPE=S0,	\$
GROUP=CLIENTREC,	ALIAS=E01, USAGE=A57,	ACTUAL=A57, \$
FIELD=NAMEADDRAREA,	ALIAS=E02, USAGE=A57,	ACTUAL=A57, \$
\$ GROUP=NAMEANDADDR,	ALIAS=E03, USAGE=A57,	ACTUAL=A57, \$
\$ FIELD=NAME,	ALIAS=E04, USAGE=A20,	ACTUAL=A20, \$
\$ FIELD=STREET,	ALIAS=E05, USAGE=A15,	ACTUAL=A15, \$
\$ FIELD=CITY,	ALIAS=E06, USAGE=A15,	ACTUAL=A15, \$
\$ FIELD=STATE,	ALIAS=E07, USAGE=A2,	ACTUAL=A2, \$
\$ FIELD=ZIP,	ALIAS=E08, USAGE=A5,	ACTUAL=A5, \$
\$ FIELD=ZIP_NUMERIC,	ALIAS=E09, USAGE=P5,	ACTUAL=Z5, \$

値 N の場合

FILE=REDEFN,	SUFFIX=FIX,	\$		
SEGNAME=CLIENSEG,	SEGTYPE=S0,	\$		
GROUP=CLIENTREC,	ALIAS=E01,	USAGE=A57,	ACTUAL=A57,	\$
FIELD=NAMEADDRAREA,	ALIAS=E02,	USAGE=A57,	ACTUAL=A57,	\$

コメントとしてのレベル 88

COBOL FD のレベル 88 フィールドは、特定の値とフィールド名を関連付けます。レベル 88 のフィールドと値は、マスターファイルでは必要ありません。ただし、これらをコメントとして追加することができます。これらを含める場合は Y (yes)、含めない場合は N (no) を入力します。レベル 88 フィールドを追加することで、レベル 88 フィールドを必要とするファイルの RECTYPE の値が簡単に識別できます (RECTYPE および値は手動で追加する必要があります。レベル 88 のフィールドの多くが COBOL FD で発生するため、適切な RECTYPE フィールドと値を正しく特定することは不可能です)。レベル 88 のフィールド名には通常、これらのフィールドが表す値が記述されるため、コメントとしての追加がマスターファイルの記述方法にもなります。

コメントとしてのレベル 88 の生成

次の COBOL 入力記述について考察します。

```

01  EMPL-REC.
    02  EMPL-SOC-SEC-NUM          PIC S9(8) COMP-3.
    02  EMPL-STATUS              PIC X(1).
        88  EMPL-ACTIVE          VALUE 'A'.
        88  EMPL-INACTIVE        VALUE 'I'.
        88  EMPL-RETIRED         VALUE 'R'.

```

この入力から、Generate LEVEL 88 as Comments に入力した値に基づいて、次の 2 つのマスターファイルが生成されます。

値 Y の場合

FILE=LEV88Y,	SUFFIX=FIX,	\$		
SEGNAME=RECSEG,	SEGTYPE=S0,	\$		
FIELD=SOCSECNUM,	ALIAS=E01,	USAGE=P9,	ACTUAL=P5,	\$
FIELD=STATUS,	ALIAS=E02,	USAGE=A1,	ACTUAL=A1,	\$
\$	ACTIVE, VALUE 'A'.	\$		

\$	INACTIVE, VALUE 'I'.	\$
\$	RETIRED, VALUE 'R'.	\$

レベル 88 のフィールドは、これらが記述するフィールド (STATUS) の下にインデントされます。値は左揃えで配置され、COBOL のレコードに多数の値が存在する場合に、マスターファイルに最大の情報を含めることができます。COBOL のレベル 88 の値が FD の 1 行に収まらない場合は、1 行目のみマスターファイルに含まれます。

値 N の場合

FILE=LEV88N,	SUFFIX=FIX,	\$		
SEGNAME=RECSEG,	SEGTYPE=S0,	\$		
FIELD=SOCSECNUM,	ALIAS=E01,	USAGE=P9,	ACTUAL=P5,	\$
FIELD=STATUS,	ALIAS=E02,	USAGE=A1,	ACTUAL=A1,	\$

セグメントとしての OCCURS

COBOL FD の変換は、すべての形式の COBOL OCCURS 構造 (固定、可変、ネスト) をサポートします。OCCURS 構造は、マスターファイルのセグメントとして、または個別の番号付きフィールドとして記述できます。OCCURS 構造をセグメントとして記述する場合は Y (yes)、そうでない場合は N (no) を追加します。

セグメントとして記述される固定およびネストされた反復グループでは、OCCURS 句を使用してグループの出現回数を記述します。固定フォーマットの反復グループでは、OCCURS の値は出現回数になります。可変フォーマットの反復グループでは、この値が COBOL DEPENDING ON フィールドの名前になります。POSITION 属性を使用してレコード内の位置を記述し、これを内部生成された POSITION フィールドでマスターファイルに予約します。この変換で生成されたすべての内部 POSITION フィールドは、末尾に POSN が追加されます。

番号付きフィールドで記述される固定フォーマットの反復グループには、出現ごとに COBOL フィールド名の末尾に一意の番号が追加されます。可変フォーマットの反復グループの出現回数は不明のため、個別に記述することはできません。実際に、反復グループのいずれかが可変フォーマットである場合、これらはすべてセグメントとして記述されます (メニュー選択にかかわらず、固定フォーマットの反復も含まれます)。

単純な固定フォーマットの OCCURS

次の COBOL 入力記述について考察します。

```

01  TBL-REC.
    02  TBL-ENTRY OCCURS 2 TIMES.
        03  TBL-AMT-A                PIC S9(5).
        03  TBL-AMT-B                PIC S9(5).

```

この変換により、Describe OCCURS as Segments に入力した値に基づき、この入力から次の 2 つのマスタファイルが生成されます。

値 Y の場合

FILE=OCCURSFY,	SUFFIX=FIX,	\$		
SEGNAME=TBLRESEG,	SEGTYPE=S0,	\$		
GROUP=TBLREC,	ALIAS=E01,	USAGE=A20,	ACTUAL=A20,	\$
FIELD=TBLSEGPOSN,	ALIAS=E02,	USAGE=A20,	ACTUAL=A20,	\$
SEGNAME=TBLSEG,	SEGTYPE=S0,	PARENT=TBLRESEG,		
OCCURS=2,	POSITION=TBLSEGPOSN,			\$
GROUP=TBLENTRY,	ALIAS=E03,	USAGE=A16,	ACTUAL=A10,	\$
FIELD=TBLAMTA,	ALIAS=E04,	USAGE=P6,	ACTUAL=Z5,	\$
FIELD=TBLAMTB,	ALIAS=E05,	USAGE=P6,	ACTUAL=Z5,	\$

値 N の場合

FILE=OCCURSFN,	SUFFIX=FIX,	\$		
SEGNAME=TBLRESEG,	SEGTYPE=S0,	\$		
GROUP=TBLREC,	ALIAS=E01,	USAGE=A32,	ACTUAL=A20,	\$
GROUP=TBLENTRY1,	ALIAS=E02,	USAGE=A16,	ACTUAL=A10,	\$
FIELD=TBLAMTA1,	ALIAS=E03,	USAGE=P6,	ACTUAL=Z5,	\$
FIELD=TBLAMTB1,	ALIAS=E04,	USAGE=P6,	ACTUAL=Z5,	\$
GROUP=TBLENTRY2,	ALIAS=E05,	USAGE=A16,	ACTUAL=A10,	\$
FIELD=TBLAMTA2,	ALIAS=E06,	USAGE=P6,	ACTUAL=Z5,	\$
FIELD=TBLAMTB2,	ALIAS=E07,	USAGE=P6,	ACTUAL=Z5,	\$

可変フォーマットの OCCURS

次の COBOL 入力記述について考察します。

```

01  TABLE-REC.
    02  TABLE-COUNT                PIC S9(2) COMP.
    02  TABLE-ENTRY
        OCCURS 1 TO 10 TIMES
        DEPENDING ON TABLE-COUNT.

```

```
03  TABLE-AMT-A          PIC S9(5).
03  TABLE-AMT-B          PIC S9(5).
```

この変換により、Describe OCCURS as Segments に入力した値 (Y または N) に基づき、この入力から次のマスターファイルが生成されます。

```
FILE=OCCURSVY,          SUFFIX=FIX,          $
SEGNAME=TABLESEG,      SEGTYPE=S0,          $
  FIELD=TABLECOUNT,  ALIAS=E01,    USAGE=I3,    ACTUAL=I2,    $
SEGNAME=TABLESE2,      SEGTYPE=S0,    PARENT=TABLESEG,
OCCURS=TABLECOUNT,
  GROUP=TABLEENTRY,    ALIAS=E02,    USAGE=A16,    ACTUAL=A10,    $
  FIELD=TABLEAMTA,     ALIAS=E03,    USAGE=P6,     ACTUAL=Z5,     $
  FIELD=TABLEAMTB,     ALIAS=E04,    USAGE=P6,     ACTUAL=Z5,     $
```

レポートの作成元 (TABLEAMTA および TABLEAMTB) のフィールドは、OCCURS セグメント内に存在します。テーブルの個数は、TABLECOUNT フィールドによって自動的に定義されます。

ネストされた OCCURS

次の COBOL 入力記述について考察します。

```
01  TABLE-REC.
    02  TABLE-LEVEL-A OCCURS 2.
        03  TABLE-LEVEL-B OCCURS 3.
            04  TABLE-AMT          PIC S9(5) COMP-3.
```

この変換により、Describe OCCURS as Segments に入力した値に基づき、この入力から次の 2 つのマスターファイルが生成されます。

値 Y の場合

```
FILE=OCCURSNY,          SUFFIX=FIX,          $
SEGNAME=TABLESEG,      SEGTYPE=S0,          $
  GROUP=TABLEREC,      ALIAS=E01,    USAGE=A18,    ACTUAL=A18,    $
  FIELD=TABLESE2POSN,  ALIAS=E02,    USAGE=A18,    ACTUAL=A18,    $
SEGNAME=TABLESE2,      SEGTYPE=S0,    PARENT=TABLESEG,
OCCURS=2,              POSITION=TABLESE2POSN,
  GROUP=TABLELEVELA,    ALIAS=E03,    USAGE=A9,     ACTUAL=A9,     $
  FIELD=TABLESE3POSN,   ALIAS=E04,    USAGE=A9,     ACTUAL=A9,     $
SEGNAME=TABLESE3,      SEGTYPE=S0,    PARENT=TABLESE2,
```

OCCURS=3,	POSITION=TABLESE3POSN,			\$
GROUP=TABLELEVELB,	ALIAS=E05,	USAGE=A8,	ACTUAL=A3,	\$
FIELD=TABLEAMT,	ALIAS=E06,	USAGE=P6,	ACTUAL=P3,	\$

この変換により、ネストされたセグメントを含むネストされたテーブル構造が定義されます。最後のレポートフィールド TABLEAMT のみ参照する必要があります。

値 N の場合

FILE=OCCURSNN,		SUFFIX=FIX,		\$
SEGNAME=TABLESEG,	SEGTYPE=S0,			\$
GROUP=TABLELREC,	ALIAS=E01,	USAGE=A48,	ACTUAL=A18,	\$
GROUP=TABLELEVELA1,	ALIAS=E02,	USAGE=A24,	ACTUAL=A9,	\$
GROUP=TABLELEVELB1,	ALIAS=E03,	USAGE=A8,	ACTUAL=A3,	\$
FIELD=TABLEAMT11,	ALIAS=E04,	USAGE=P6,	ACTUAL=P3,	\$
GROUP=TABLELEVELB2,	ALIAS=E05,	USAGE=A8,	ACTUAL=A3,	\$
FIELD=TABLEAMT12,	ALIAS=E06,	USAGE=P6,	ACTUAL=P3,	\$
GROUP=TABLELEVELB3,	ALIAS=E07,	USAGE=A8,	ACTUAL=A3,	\$
FIELD=TABLEAMT13,	ALIAS=E08,	USAGE=P6,	ACTUAL=P3,	\$
GROUP=TABLELEVELA2,	ALIAS=E09,	USAGE=A24,	ACTUAL=A9,	\$
GROUP=TABLELEVELB4,	ALIAS=E10,	USAGE=A8,	ACTUAL=A3,	\$
FIELD=TABLEAMT21,	ALIAS=E11,	USAGE=P6,	ACTUAL=P3,	\$
GROUP=TABLELEVELB5,	ALIAS=E12,	USAGE=A8,	ACTUAL=A3,	\$
FIELD=TABLEAMT22,	ALIAS=E13,	USAGE=P6,	ACTUAL=P3,	\$
GROUP=TABLELEVELB6,	ALIAS=E14,	USAGE=A8,	ACTUAL=A3,	\$
FIELD=TABLEAMT23,	ALIAS=E15,	USAGE=P6,	ACTUAL=P3,	\$

ORDER フィールド

レポート作成で特定のオカレンスを選択する場合は、OCCURS セグメントの最後のエントリとして、マスターファイルに ORDER フィールドを追加します。

FIELD=fieldname,	ALIAS=ORDER,	USAGE=I4,	ACTUAL=I4,	\$

ALIAS の値は ORDER、ACTUAL の値は I4 にする必要があります。フィールド名は任意です。USAGE の値は整数 (I) にする必要がありますが、長さは 1 から 9 までのいずれかにすることができます。ORDER フィールドには自動的に値が入力されます。

ネストされた OCCURS での ORDER フィールドの順序

次の例は、ORDER フィールドの SEQUENCE をネストされた固定フォーマットの OCCURS セグメント 2 および 3 に追加します。

```

FILE=OCCURSNY,                                SUFFIX=FIX,                                $
SEGNAME=TABLESEG,    SEGTYPE=S0,                                $
  GROUP=TABLEREC,    ALIAS=E01,    USAGE=A18,    ACTUAL=A18,    $
  FIELD=TABLESE2POSN, ALIAS=E02,    USAGE=A18,    ACTUAL=A18,    $
SEGNAME=TABLESE2,    SEGTYPE=S0,    PARENT=TABLESEG,
OCCURS=2,            POSITION=TABLESE2POSN,                                $
  GROUP=TABLELEVELA, ALIAS=E03,    USAGE=A9,    ACTUAL=A9,    $
  FIELD=TABLESE3POSN, ALIAS=E04,    USAGE=A9,    ACTUAL=A9,    $
  FIELD=SEQUENCEA,    ALIAS=ORDER, USAGE=I4,    ACTUAL=I4,    $
SEGNAME=TABLESE3,    SEGTYPE=S0,    PARENT=TABLESE2,
OCCURS=3,            POSITION=TABLESE3POSN,                                $
  GROUP=TABLELEVELB, ALIAS=E05,    USAGE=A8,    ACTUAL=A3,    $
  FIELD=TABLEAMT,    ALIAS=E06,    USAGE=P6,    ACTUAL=P3,    $
  FIELD=SEQUENCEB,    ALIAS=ORDER, USAGE=I4,    ACTUAL=I4,    $

```

ゾーン数値フィールドの使用

COBOL FD では、PICTURE 9(n) 句を使用して、英数文字をゾーン 10 進数の数値データとして記述することができます。この句は、算術演算が実行されない数値文字列データ (例、電話番号、ID 番号、日付) でよく使用されます。これらのフィールドは、パック 10 進数 (値 P を入力) または英数文字 (値 A を入力) のUSAGE フォーマットで記述することができます。ACTUAL フォーマットはゾーン (Z) で記述されます。

集計または算術演算で COBOL 数値表示フィールドを使用する場合は、パック 10 進数のオプションを使用します。これらの要素を表示目的のみで使用する場合は、英数文字オプションを使用します。PICTURE 9(n) フォーマットの単純な数値表示要素のみが、この機能の対象となります。符号付きの要素 (PICTURE S9(n)) および小数点付きの要素 (PICTURE 9(n)V(m)) は、常にパック 10 進数で記述されます。

ゾーン数値フィールドの使用

次の COBOL 入力記述について考察します。

```

01 EMPL-REC.
    02 EMPL-SOC-SEC-NUM          PIC 9(8).
    02 EMPL-HIRE-DATE.
        03 EMPL-HIRE-YEAR        PIC 9(2).
        03 EMPL-HIRE-MONTH       PIC 9(2).
        03 EMPL-HIRE-DAY         PIC 9(2).
    02 EMPL-SICK-DAYS-ALLOWED    PIC S9(2).
    02 EMPL-SICK-DAYS-TAKEN      PIC 9(2).
    02 EMPL-YTD-HOURS-WORKED     PIC 9(4)V9(2).

```

この変換により、Zoned Numeric Field Usage に入力した値に基づき、この入力から次の 2 つのマスタファイルが生成されます。

値 P の場合

FILE=ZONEDP,		SUFFIX=FIX,		\$
SEGNAME=RECSEG,	SEGTYPE=S0,			\$
GROUP=REC,	ALIAS=E01,	USAGE=A56,	ACTUAL=A24,	\$
FIELD=SOCSECNUM,	ALIAS=E02,	USAGE=P8,	ACTUAL=Z8,	\$
GROUP=HIREDATE,	ALIAS=E03,	USAGE=A24,	ACTUAL=A6,	\$
FIELD=HIREYEAR,	ALIAS=E04,	USAGE=P2,	ACTUAL=Z2,	\$
FIELD=HIREMONTH,	ALIAS=E05,	USAGE=P2,	ACTUAL=Z2,	\$
FIELD=HIREDAY,	ALIAS=E06,	USAGE=P2,	ACTUAL=Z2,	\$
FIELD=SICKDAYSALLO,	ALIAS=E07,	USAGE=P3,	ACTUAL=Z2,	\$
FIELD=SICKDAYSTAKE,	ALIAS=E08,	USAGE=P2,	ACTUAL=Z2,	\$
FIELD=YTDHOURSWORK,	ALIAS=E09,	USAGE=P7.2,	ACTUAL=Z6,	\$

値 A の場合

FILE=ZONEDA,		SUFFIX=FIX,		\$
SEGNAME=RECSEG,	SEGTYPE=S0,			\$
GROUP=REC,	ALIAS=E01,	USAGE=A32,	ACTUAL=A24,	\$
FIELD=SOCSECNUM,	ALIAS=E02,	USAGE=A8,	ACTUAL=Z8,	\$
GROUP=HIREDATE,	ALIAS=E03,	USAGE=A6,	ACTUAL=A6,	\$
FIELD=HIREYEAR,	ALIAS=E04,	USAGE=A2,	ACTUAL=Z2,	\$
FIELD=HIREMONTH,	ALIAS=E05,	USAGE=A2,	ACTUAL=Z2,	\$
FIELD=HIREDAY,	ALIAS=E06,	USAGE=A2,	ACTUAL=Z2,	\$
FIELD=SICKDAYSALLO,	ALIAS=E07,	USAGE=P3,	ACTUAL=Z2,	\$
FIELD=SICKDAYSTAKE,	ALIAS=E08,	USAGE=A2,	ACTUAL=Z2,	\$
FIELD=YTDHOURSWORK,	ALIAS=E09,	USAGE=P7.2,	ACTUAL=Z6,	\$

生成された 2 つのマスタファイルには、実際に数値および文字フォーマットで使用するゾーン 10 進数フィールドが含まれますが、変換では 2 つのケースを識別することができません。そのため、状況に最も適したオプションを選択する必要があります。必要に応じて、生成されたマスタファイルを編集して、パック 10 進数または英数文字にフォーマットを

変更することができます。これらのゾーンフィールドを含む GROUP フィールドの USAGE の長さも、[フォーマット変換に関する一般的注意事項](#)のフォーマット変換規則に従って変更する必要があります。

数値フィールド編集オプション

COBOL FD 変換では、生成されたマスターファイルのすべての数値フィールドに、指定した編集オプションが自動的に追加されます。編集オプションは、数値データの表示方法に影響します。次のオプションがあります。

S - フィールドの値が 0 の場合、数字の 0 (ゼロ) を表示しません

C - 適切な場合、カンマを挿入します (例、1,234)

B - 負の値を括弧で囲みます (例、(1234))

R - 負の値をクレジットとして表示します (例、1234 CR)

M - ドル記号 (浮動) およびカンマを表示します (例、\$1,234)

N - ドル記号 (固定) およびカンマを表示します (例、\$ 1,234)

L - 先頭の 0 (ゼロ) を表示します (例、001234)

最大 5 つの編集オプションを入力することができます。数値フィールドの編集が必要ない場合は、このオプションをブランクにしておくこともできます。編集オプションは、任意の順序と組み合わせで入力できますが、相互排他的な SL、MN、BR の組み合わせは除きます。

数値フィールド編集オプション

次の COBOL 入力記述について考察します。

```
01  PAYROLL-REC.
    05  PAYROLL-NAME.
        10  PAYROLL-LAST-NAME    PIC X(20).
        10  PAYROLL-FIRST-NAME   PIC X(10).
    05  PAYROLL-AMT-FIELDS.
        10  PAYROLL-AMT-CURR-YR   PIC S9(7)V99 COMP-3.
        10  PAYROLL-AMT-PREV-YR   PIC S9(7)V99 COMP-3.
```

この変換では、Numeric Field Options に MB を指定した際に、この入力から次のマスターファイルが生成されます。

FILE=EDITOPT,		SUFFIX=FIX,		\$
SEGNAME=RECSEG,	SEGTYPE=S0,			\$
GROUP=REC,	ALIAS=E01,	USAGE=A46,	ACTUAL=A40,	\$
GROUP=NAME,	ALIAS=E02,	USAGE=A30,	ACTUAL=A30,	\$
FIELD=LASTNAME,	ALIAS=E03,	USAGE=A20,	ACTUAL=A20,	\$
FIELD=FIRSTNAME,	ALIAS=E04,	USAGE=A10,	ACTUAL=A10,	\$
GROUP=AMTFIELDS,	ALIAS=E05,	USAGE=A16,	ACTUAL=A10,	\$
FIELD=AMTCURRYR,	ALIAS=E06,	USAGE=P11.2MB,	ACTUAL=P5,	\$
FIELD=AMTPREVVYR,	ALIAS=E07,	USAGE=P11.2MB,	ACTUAL=P5,	\$

フィールド名情報

この変換では、COBOL フィールド名からマスターファイルのフィールド名が生成されます。この場合、以下が適用されます。

1. Skip 'n' Hyphens オプションは、メニューで入力した n の値に基づき、左側から n 番目のハイフンまでの文字を削除します。次に、Remove Hyphens または Use Underbars メニューオプションの設定に基づき、残りのハイフンをすべて削除するか、これらをアンダースコア () で置換します。最後に、Field Name Length オプションが 12 に設定されている場合、左端から 12 文字のみが使用されます。
2. 生成済みのフィールド名を複製する場合は、これを固有の名前にするために修飾子が追加されます。たとえば、次の COBOL 構造について考察します。

```
02  EMPL-PORTION.
    03  SOC-SEC-NUM  PIC 9(9).
02  MGR-PORTION.
    03  SOC-SEC-NUM  PIC 9(9).
```

SOCSECNUM は、1 つ目のエレメンタリフィールド名から生成され、SOCSECNUM2 は、2 つ目のエレメンタリフィールド名から生成されます。

下表は、フィールド名の生成を制御するメニューオプションの可能な組み合わせ結果を示しています。

COBOL フィールド名 - PAYROLL-REC-IN-FICA			
フィールド名長さ	Remove Hyphens または Use Underbars	Skip 'n' Hyphens	生成されたフィールド名
12	Remove Hyphens	0	PAYROLLRECIN
		1	RECINFICA
		2	INFICA
		3	FICA
	Use Underbars	0	PAYROLL_REC
		1	REC_IN_FICA
		2	IN_FICA
		3	FICA
30	Remove Hyphens	0	PAYROLLRECINFICA
		1	RECINFICA
		2	INFICA
		3	FICA

COBOL フィールド名 - PAYROLL-REC-IN-FICA			
フィールド名長さ	Remove Hyphens または Use Underbars	Skip 'n' Hyphens	生成されたフィールド名
	Use Underbars	0	PAYROLL_REC_IN_FICA
		1	REC_IN_FICA
		2	IN_FICA
		3	FICA

フォーマット変換に関する一般的注意事項

フォーマット変換は、COBOL フォーマットに基づいて ACTUAL および USAGE フォーマットを定義するプロセスです。下表のような変換が実行されます。

COBOL フォーマット	ACTUAL	USAGE
PICTURE X(n)	An	An
PICTURE 9(n)	Zn (packed option)	Pn (packed option)
PICTURE 9(n)	An (alpha option)	An (alpha option)

COBOL フォーマット	ACTUAL	USAGE
PICTURE S9(n)	Zn	Pn+1
PICTURE 9(n)V9(m)	Zn+m	Pn+m+1.m
PICTURE S9(n)V9(m)	Zn+m	Pn+m+2.m
PICTURE 9(n) COMP (1 # n # 4)	I2	I9
PICTURE 9(n) COMP (5 # n # 9)	I4	I9
PICTURE 9(n) COMP (n > 9)	A8	A8
COMP-1	F4	F8
COMP-2	D8	D15
PICTURE 9(n) COMP- 3	P(n+2/2)	Pn
PICTURE S9(n) COMP- 3	P(n+2/2)	Pn+1

COBOL フォーマット	ACTUAL	USAGE
PICTURE 9(n)V9(m) COMP-3	P(n+m+2/2)	Pn+m+1.m
PICTURE S9(n)V9(m) COMP-3	P(n+m+2/2)	Pn+m+2.m

フォーマット変換には、次の制限事項が適用されます。

- 英数文字フィールドの最大長は 256 バイトです。256 バイトを超えるエレメンタリフィールドは、256 バイトに切り捨てられ、合計長さを示すために未使用フィールドが挿入されます。256 バイトを超えるグループフィールドにはコメントが追加されます。
- 小数点を含まない符号なしのゾーンフィールドは、Zoned Numeric Field Usage オプションに入力した値に基づいて、パック 10 進数 (P) または英数文字 (A) の USAGE で記述されます。
- COMP-4 バイナリフィールドは、COMP フィールドと同様に記述されます。
- PICTURE 句内の 9 桁を超えるバイナリフィールドは、ACTUAL フォーマット A8 または USAGE フォーマット A8 で記述されます。
- パック 10 進数 USAGE フォーマットの最大長は、生成されたマスターファイルを使用するソフトウェアのバージョンによって異なります。長いパック 10 進数フィールド (16 桁以上) をサポートするバージョンでは、ACTUAL の最大長は 16 バイト、USAGE の最大長は 31 桁 (または、小数点および符号を含めて 32 バイト) です。以前のバージョンでは、ACTUAL の最大長は 8 バイト、USAGE の最大長は合計 15 桁 (小数点および符号を含む) です。小数点以下の桁数は、USAGE の合計長さより 1 桁少ない桁数に制限されます。
- 一般的に、パック 10 進数フィールドの USAGE の長さは、小数点の左側の桁数 (n)、小数点以下の桁数 (m)、および先頭のマイナス符号の 1 桁 (s) (存在する場合)、小数点の 1 桁 (v) (存在する場合) の合計として計算されます。
- GROUP フィールドの ACTUAL および USAGE フォーマットは、常に英数文字 (A) です。ACTUAL の長さは、各コンポーネントの ACTUAL 長さの合計です。USAGE の長さは、以下の合計です。
- すべての英数文字 (A) コンポーネントの USAGE 長さ
- 長いパック 10 進数 (P) の場合、各コンポーネントに対して 16 桁

- 短いパック 10 進数 (P) および倍精度浮動小数点数 (D) の場合、各コンポーネントに対して 8 桁
- 整数 (I) および単精度浮動小数点数 (F) の場合、各コンポーネントに対して 4 桁

INPUT としての複数レコード

この変換により、INPUT として受信する各 COBOL レコード記述 (01 レベルフィールド) に対して、新しいセグメントが生成されます。複数のレコードが存在する場合は、追加の DUMMY セグメントが作成され、レコード記述の共通の親として使用されます。複数の 01 レベルのレコードを INPUT として送信する場合、手動で RECTYPE フィールドを追加し、異なるレコードタイプを識別する必要があります。

INPUT としての複数レコード

次の COBOL 入力記述について考察します。

```
01  HDR-REC.
    02  HDR-KEY          PIC X(1).
        88  HDR-VALUE    VALUE 'H'.
    02  HDR-DATA         PIC X(10).
01  DET-REC.
    02  DET-KEY          PIC X(1).
        88  DTL-VALUE    VALUE 'D'.
    02  DTL-DATA         PIC X(10).
```

この入力から、複数セグメントを含む次のマスターファイルが生成されます。

```
FILE=MULT01,          SUFFIX=FIX,          $
SEGNAME=DUMMY,        SEGTYPE=S0,           $
FIELD=,               ALIAS=,               USAGE=A1,      ACTUAL=A1,      $
SEGNAME=HDRRESEG,     SEGTYPE=S0,   PARENT=DUMMY,  $
GROUP=HDRREC,         ALIAS=E01,   USAGE=A11,      ACTUAL=A11,      $
FIELD=HDRKEY,         ALIAS=E02,   USAGE=A1,      ACTUAL=A1,      $
$   HDRVALUE, VALUE 'H'.          $
FIELD=HDRDATA,        ALIAS=E03,   USAGE=A10,      ACTUAL=A10,      $
SEGNAME=DETRESEG,     SEGTYPE=S0,   PARENT=DUMMY,  $
GROUP=DETREC,         ALIAS=E04,   USAGE=A11,      ACTUAL=A11,      $
FIELD=DETKEY,         ALIAS=E05,   USAGE=A1,      ACTUAL=A1,      $
```

```

$      DTLVALUE, VALUE 'D'.
FIELD=DTLDATA,      ALIAS=E06,      USAGE=A10,      ACTUAL=A10,      $

```

これを編集し、RECTYPE 情報を追加すると、次のようなマスターファイルが完成されます。

```

FILE=MULT01,          SUFFIX=FIX,          $
SEGNAME=DUMMY,        SEGTYPE=S0,          $
FIELD=,              ALIAS=,              USAGE=A1,      ACTUAL=A1,      $
SEGNAME=HDRRESEG,     SEGTYPE=S0,          PARENT=DUMMY,  $
GROUP=HDRREC,         ALIAS=E01,          USAGE=A11,     ACTUAL=A11,     $
FIELD=RECTYPE,        ALIAS=H,            USAGE=A1,      ACTUAL=A1,      $
$      HDRVALUE, VALUE 'H'.          $
FIELD=HDRDATA,        ALIAS=E03,          USAGE=A10,     ACTUAL=A10,     $
SEGNAME=DETRESEG,     SEGTYPE=S0,          PARENT=DUMMY,  $
GROUP=DETREC,         ALIAS=E04,          USAGE=A11,     ACTUAL=A11,     $
FIELD=RECTYPE,        ALIAS=D,            USAGE=A1,      ACTUAL=A1,      $
$      DTLVALUE, VALUE 'D'.          $
FIELD=DTLDATA,        ALIAS=E06,          USAGE=A10,     ACTUAL=A10,     $

```

フィールド最大数

01 レベルの各レコード入力には、最大 4,000 のフィールドを含めることができます。入力として任意の数の 01 レベルエントリを変換で使用することができますが、各レコードはこのフィールド最大数を超えることはできません。OCCURS 構造の各オカレンスは、この制限の対象になります。最大数を超えた場合、エラーメッセージが出力され、プログラムの実行が停止されます。この制限は、変換に対して適用されるもので、生成されたマスターファイルには適用されません。

フィールド最大数

次の COBOL 入力記述について考察します。

```

01  REC1.
    02  FLD1  OCCURS  5000  TIMES  PIC  X(1).

```

この入力により、次のメッセージが生成されます。

```
FR011E - INTERNAL TABLE OVERFLOW
```

この制限を回避するには、COBOL FD を編集し、OCCURS の総数を一時的に減らします。実行後に、マスターファイルを編集し、オカレンス数を元に戻します。

次の COBOL 入力記述について考察します。

```
01  REC1.
    02  FLD1  OCCURS  2000  TIMES  PIC  X(1).
01  REC2.
    02  FLD2  OCCURS  2000  TIMES  PIC  X(1).
```

複数のセグメントを含む次のマスターファイルが、この入力により生成されます。

FILE=MAXRECS,	SUFFIX=FIX,	\$
SEGNAME=DUMMY,	SEGTYPE=S0,	\$
FIELD=,	ALIAS=,	USAGE=A1, ACTUAL=A1, \$
SEGNAME=REC1SEG,	SEGTYPE=S0, PARENT=DUMMY,	\$
\$ GROUP=REC1,	ALIAS=E01, USAGE=A2000,	ACTUAL=A2000, \$
FIELD=FLD1SEGPOSN,	ALIAS=E02, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E03, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E04, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E05, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E06, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E07, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E08, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E09, USAGE=A208,	ACTUAL=A208, \$
SEGNAME=FLD1SEG,	SEGTYPE=S0, PARENT=REC1SEG,	
OCCURS=2000,	POSITION=FLD1SEGPOSN,	\$
FIELD=FLD1,	ALIAS=E10, USAGE=A1,	ACTUAL=A1, \$
SEGNAME=REC2SEG,	SEGTYPE=S0, PARENT=DUMMY,	\$
\$ GROUP=REC2,	ALIAS=E11, USAGE=A2000,	ACTUAL=A2000, \$
FIELD=FLD2SEGPOSN,	ALIAS=E12, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E13, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E14, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E15, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E16, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E17, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E18, USAGE=A256,	ACTUAL=A256, \$
FIELD=FILLER,	ALIAS=E19, USAGE=A208,	ACTUAL=A208, \$
SEGNAME=FLD2SEG,	SEGTYPE=S0, PARENT=REC2SEG,	
OCCURS=2000,	POSITION=FLD2SEGPOSN,	\$
FIELD=FLD2,	ALIAS=E20, USAGE=A1,	ACTUAL=A1, \$

各 01 レベルのフィールド数が 4,000 より少ないため、この記述は最大 4,000 のフィールド数制限を超過していません。

2000 年

COBOL FD では、日付としてフィールドを記述できないため、変換によって、日付フィールドを確かに特定し、これらに日付フォーマットを適用することができません。ただし、シノニム作成時に生成されたマスターファイルを編集し、日付フォーマットおよび日付のデフォルト値を追加することができます。これらを利用するために設計されたアダプタソフトウェアで適用されます。

COBOL FD の構文要件

COBOL FD の変換では、入力として構文的に正しい COBOL ファイル記述が必要です。シノニム作成時の COBOL ファイル記述のスキャンでは、さまざまな構文チェックが実行されます。COBOL ステートメントのいずれかが無効な場合、デバッグメッセージが表示され、プログラムの実行が停止されます。この場合、マスターファイルは生成されません。構文チェックではさまざまなエラーを特定できますが、正しくコンパイルされる有効な COBOL ファイル記述のみを指定する必要があります。

COBOL FD 構文のガイドライン

最低要件として、有効な COBOL ファイル記述は次のガイドラインに従う必要があります。

- レベル番号は、1 桁または 2 桁の整数です。値は、1 から 49、66、77、88 が使用できます。
- レベル番号 01 および 77 は、カラム 8 から 11 で開始されます。その他のレベル番号はすべて、カラム 8 から 72 で開始されます。
- コメント行は、カラム 7 のアスタリスク (*) で識別されます。
- 継続行は、カラム 7 のハイフン (-) で識別されます。
- COBOL ファイル記述は、レベル番号 01 のカラムで開始し、記述された最終カラムで終了します。
- COBOL フィールド名は、大文字で記述されます。
- これ以外の COBOL 命令は、入力として許可されません。
- 埋め込みタブおよびブランク行は、入力として許可されません。

- 複数レベルの VALUE 句では、最大 9 行がサポートされます。
マスターファイルでは、1 つのレベルのみコメントとして設定されます。

データセット圧縮イグジット - ZCOMP

データセットの圧縮に使用する ZCOMP イグジットの使用方法について説明します。

ZCOMP イグジットは、VSAM、ISAM、CISAM、RMS、および他の SAM タイプのデータ構造に適用されます。また、特定の FIX 構造にも適用されます。

圧縮イグジットは 3 つの関数と main エントリポイントで構成され、集合的に ZCOMP イグジットと呼ばれます。main エントリポイントは、以下をエクスポートする ZCOMP です。

- 初期ハウスキepingのための ZCOMP0
- 実際の処理のための ZCOMP1 - ユーザがコーディングされたフィールドにアクセスしてこれを復号化し、圧縮されたレコードを展開して、他の指定されたデータ操作を実行することができるよう、設定されています。
- ZCOMP2 - セッション終了時のクリーンアップを実行するために呼び出されます。

アンカーストレージは、永続性と再入可能性を維持するため、ZCOMP0、ZCOMP1、ZCOMP2 に渡されます。

ZCOMP の参照コードは提供されますが、実際の DLL はユーザ定義され、特定のニーズに合わせて保持されることが想定されます。イグジットは、標準のユーザ定義のイグジットプロシジャに従ってコンパイルおよびリンクされる必要があります。

ZCOMP を使用するために、特別なマスターファイルのコーディング要件はありません。SUFFIX の値は、VSAM (KSDS または ESDS ファイル)、CISAM、RMS、FIX (シーケンシャルファイル)、または PRIVATE (GETPRV ユーザイグジットによるファイルアクセス) にすることができます。

ZCOMP イグジットの呼び出し

ZCOMP イグジット DLL は、C、BAL、または参照例で使用されるパラメータ送信規則に従い、適切な DLL としてコンパイルおよびリンクされる任意の 3GL 言語で記述することができます。生成される DLL には、DLL の物理名と一致するエントリポイントが必要です。デフォルト設定の DLL 名は ZCOMP です。

ZCOMP ユーザイグジットの記述後、プラットフォームに基づいて、次のようにこれをコンパイルおよびリンクします。

プラットフォーム	コンパイル/リンクメカニズム
USS、UNIX、Windows、IBM i	GENCPGM
VM/ESA	GENFSAM EXEC を使用し、VWSET で DLL またはリンクを作成

注意

- メインフレームの GENFSAM JCL および EXEC は、ZCOMP および GETPRV ユーザイグジットの両方を同時にリンクするために設定されます。これらのイグジットのいずれか 1 つのみを実装する場合、VM EXEC は必要なルーチンへの正確なリンクを生成しますが、z/OS の JCL は正確なリンクを生成しません。後者の場合、GENFSAM メンバーの INCLUDE OBJECT ステートメントをコメントアウトする必要があります。

プロセスのどの時点で ZCOMP イグジットの DLL がロードされるかは、プラットフォームおよびリンクが VWSET で実行されるかどうかによって異なります。

- VWSET にリンクされるイグジットを使用するプラットフォームは、ユーザのセッション開始時にデフォルト ZCOMP イグジットのロードを受信し、終了時にアンロードされます。
- DLL としてイグジットを使用するプラットフォームは、明示的なリクエスト (例、SELECT または TABLE) に応答してのみロードします。これらのリクエストは、データリクエストの前に発行する必要があります。次のように、コマンドでデータソースを指定します。

```
ENGINE VSAMX SET ZCOMP dllname
ENGINE CISAM SET ZCOMP dllname
ENGINE RMS SET ZCOMP dllname
```

ユーザは、必要に応じて、セッション中に DLL を動的に変更することができます。現在の DLL は、使用中の DLL への変更時およびユーザのセッション終了時にアンロードされます。

明示的な DLL のアンロードを強制するには、新しい DLL 名で SET コマンドを使用します。DLL のパスは、IBICPG 環境変数で定義されます。main DLL エントリポイント関数 (DLL 名と同一にする必要がある) は、アダプタに ZCOMP0、ZCOMP1、ZCOMP2 の関数のアドレスを提供します。

VSAM エンジン、SUFFIX=VSAM ファイルに加えて SUFFIX=PRIVATE および複数セグメントの SUFFIX=FIX ファイルの処理にも使用されます。FIX および PRIVATE では、次のコマンドが使用されます。

ENGINE VSAMX SET ZCOMP

ZCOMP のロードの結果

アダプタ (マスターファイルの SUFFIX= value で定義) は、レコードを読み取ります。読み取りが成功するたびに、ZCOMP イグジットの DLL がロードされると、アダプタによりデータの展開などの操作を実行する ZCOMP1 が呼び出されます。

ZCOMP1 ロジックは、受信するパラメータ情報に基づいて何を実行するかを判断します。通常、DDNAME の値を使用して、関連するデータソースを展開したり、操作したりする必要があるかどうかが決まります。

- 展開も操作も必要ない場合は、ユーザイグジットは通常、A(IRECLLEN) を A(ORECLLEN) に、A(A(IREC)) を A(A(OREC)) に移動し、A(STATCODE) を 0 (ゼロ) にしてアダプタに戻ります。これは、パススルーと同様の動作です。関連する情報については、[ZCOMP1 BAL パラメータリスト](#)を参照してください。
- 展開またはその他のタイプの操作が必要な場合は、ユーザイグジットがこれらのタスクを行います。

レコードの転送

次の例は、単純にレコードを転送します。これは、展開または別のタイプの操作がアプリケーション固有の操作であるためです。

処理の完了後、ユーザイグジットは A(ORECLLEN) または A(A(OREC)) に戻り、ゼロステータスコードに戻ります。ステータスコードがゼロ以外の場合は、次のメッセージが生成されます。

```
(FOC1150) ZCOMP DECOMPRESS ERROR: status
```

このエラーにより、TABLE リクエストが終了します。

ZCOMP C ソースファイル (zcomp.c)

ibisamp アプリケーションには、ZCOMP FOCEXEC が格納され、このイグジットの例を「実行する」ために使用することができます。.c および .h のソースは、使用するプラットフォーム

ムの EDAHOME/etc/src3gl と同等のディレクトリで特定することができます。

```
/*-----
 * Copyright (c) 2003-2021, TIBCO Software Inc. All rights reserved.
 * @MFSM_NOPROLOG@ source control tag, please ignore and do not delete.
 * Description:Example of the dynamic ZCOMP exit functions
 *-----*/
```

```
/*
 This zcomp sample is an arbitrary example in which we simply move
 an uncompressed input buffer and size to the output buffer and size.
 Effectively, this is a pass through of operation just passing the
 buffer along; no real decompression occurs in the sample.If this was
 coded as a real application, actual decompression would be done using
 a third party compression/decompression routine linked in the sample
 for the purpose of actual use in the ZCOMP1 exit.
```

Typically this procedure would be built via gencpgm and IBICPG set.
Once the sample exit has been built, it may be exercised to
observe behavior using the zcomp.fex procedure.

```
*/
```

```
/* stdio needed for printf() lines, your code may not need stdio */
#include <stdio.h>
#include "zcomp.h"
void zcomp0 (
    long    *status      /* out: status, 0=OK                */
  ,char    *filename     /* in : ddname                */
  ,char    *userid       /* in : userid or jobname     */
  ,long    *reserv1      /* reserved:NOT TO BE USED BY EXIT */
  ,char    **reserv2     /* reserved:NOT TO BE USED BY EXIT */
  ,long    *reserv3      /* reserved:NOT TO BE USED BY EXIT */
  ,char    **reserv4     /* reserved:NOT TO BE USED BY EXIT */
  ,long    *user_wk      /* i/o : work area for the user exit */
)
{
    /* Arbitrary to show behavior; would not be coded in a real
application */
    printf("ZCOMP0 called to do initial housekeeping\n");
    *status = 0;
}
```

```

void zcomp1 (
    long    *status      /* out: status, 0=OK                */
  ,char    *filename     /* in : ddname                  */
  ,char    *userid       /* in : userid or jobname      */
  ,long    *inlen        /* in : length of original record */
  ,char    **inabuf      /* in : input record buffer    */
  ,long    *outlen       /* out : length of decoded record */
  ,char    **outabuf     /* out : decoded output record buffer */
  ,long    *user_wk      /* i/o : work area for the user exit */
)
{
    /* Arbitrary to show behavior; would not be coded in a real
application */
    printf("ZCOMP1 called to do the actual processing\n");
    /* In a real application decompression steps would normally be done
here */
    *outlen = *inlen;
    *outabuf = *inabuf;
    *status = 0;
}

```

```

void zcomp2 (
    long    *user_wk      /* I/O work area for the user exit */
)
{
    /* Arbitrary to show behavior; would not be coded in a real
application */
    printf("ZCOMP2 called to perform clean-up, session is over\n");
}

```

```

void zcomp(p_zcompep pzc)
{
    /* Arbitrary to show behavior; would not be coded in a real application
*/
    printf("ZCOMP called to initialize entry points for ZCOMP0, ZCOMP1 and
ZCOMP2\n");
    pzc->zcomp0 = zcomp0;
    pzc->zcomp1 = zcomp1;
    pzc->zcomp2 = zcomp2;
}

```

ZCOMP ヘッダファイル (zcomp.h)

```

/*-----
 * Copyright (c) 2003-2021, TIBCO Software Inc. All rights reserved.
 * @MFSM_NOPROLOG@ source control tag, please ignore and do not delete.
 * Description:Example of the dynamic ZCOMP exit functions
 *-----*/

```

```

#ifndef ZCOMP_H
#define ZCOMP_H 1
/*
-----
    -perform the set-up in the user exit
-----
*/
typedef void t_zcomp0 (
    long      *status      /* out: status = 0 if OK*/
, char      *filename     /* in : ddname          */
, char      *userid       /* in : userid or jobname */
, long      *reserv1      /* reserved:NOT TO BE USED BY THE EXIT */
, char      **reserv2     /* reserved:NOT TO BE USED BY THE EXIT */
, long      *reserv3      /* reserved:NOT TO BE USED BY THE EXIT */
, char      **reserv4     /* reserved:NOT TO BE USED BY THE EXIT */
, long      *user_wk      /* i/o : work area for the user exit */
);

```

```

/*
-----
    -decompress a record    and return the address of the new
      record and its length.The routine is responsible for allocating
      a buffer for the decompressed record.
-----
*/
typedef void t_zcomp1 (
    long      *status      /* out: status = 0 if OK*/
, char      *filename     /* in : ddname          */
, char      *userid       /* in : userid or jobname */
, long      *inlen        /* in : length of original rec */
, char      **inabuf      /* in : a' input record    */
, long      *outlen       /* out : length of decoded  rec */
, char      **outabuf     /* out : a' decoded record  */
);

```



```
,long      *user_wk      /* i/o : work area for the user exit */
);
```

```
/*
-----
    -deallocate buffer used by zcomp1
    called at the end of focus session.
-----
*/
typedef void  t_zcomp2 (
    long      *user_wk      /* i/o : work area for the user exit */
);
/*
-----
    -definition of EPs exported by the dynamic zcomp exit
-----
*/
typedef struct s_zcompep
{
    t_zcomp0 *zcomp0; /* p' to zcomp0          */
    t_zcomp1 *zcomp1; /* p' to zcomp1          */
    t_zcomp2 *zcomp2; /* p' to zcomp2          */
} t_zcompep, *p_zcompep;
```

```
/*
    Main and only one entry point in the ZCOMP module (dll)
*/
#ifdef __cplusplus
extern "C" {
#endif
typedef void  t_zcomp(p_zcompep pzc);
t_zcomp zcomp;
#ifdef __cplusplus
} /* extern "C" */
#endif /* __cplusplus */
#endif
```

ZCOMP1 BAL パラメータリスト

パラメータ	説明	長さとフォーマット
A(STATCODE)*	フルワードのバイナリステータスコードのポインタ	4 バイトの整数
A(DDNAME)	使用されている 8 バイトのファイル名のポインタ	8 バイトの文字
A(USERID)	予約済み (将来使用)	8 バイトの文字
A(IRECLLEN)	元のレコード長のポインタ	4 バイトの整数
A(A(IREC))	元のレコードのポインタのポインタ	4 バイトの整数
A(ORECLLEN)*	変更後のレコード長のポインタ	4 バイトの整数
A(A(OREC))*	変更後のレコードのポインタのポインタ	4 バイトの整数
A(USERWD)**	フルワードのポインタ	4 バイトの整数

* これらのパラメータはユーザが指定します。

** このパラメータを使用して、リエントラント処理でユーザストレージをアンカーとして指定することができます。

注意

- 元のポインタは変更せず、これらのポイント先のポインタまたは値のみを変更します。
- ZCOMP1 への入力時に、ORECLLEN および OREC パラメータは NULL になります。ユーザは、これらを正しく入力する必要があります。
- セッション終了の処理中に、ZCOMP2 エントリポイントへの呼び出しが行われます。これにより、ユーザは、必要な他のグローバルクリーンアップを実行することができます。
- ZCOMP1 によって返されたパラメータは検証されません。ユーザのルーチンにより、

有効なアドレスおよび長さが ZCOMP1 から呼び出し元に返される必要があります。

- 正しくないパラメータがルーチンから返されると、予期しない結果が発生します。

ダイナミックプライベートユーザイグジット

ダイナミックプライベートユーザイグジット API の使用方法について説明します。

ダイナミックプライベートユーザイグジットは、標準のアダプタが使用できないデータソースにアクセスするカスタム 3GL ダイナミックリンクライブラリ (DLL) の作成および統合を可能にする API です。この技法は、自社 (または不明な) データソースを使用する企業が WebFOCUS のデータアクセスを利用することができるメカニズムを提供します。

このイグジットの参照には、GETPRV および SUFFIX=PRIVATE という用語がよく使用されます。

このイグジットの参照コードおよび作成方法は、GETPRV イグジットの一部として提供されますが、ユーザは、特定のアプリケーションのニーズに合わせて実際の DLL の記述およびメンテナンスを行う必要があります。また、ユーザ定義のイグジットの設定および作成に関する標準手順に従ってこのイグジットのコンパイルおよびリンクを行う必要があります。

トラブルシューティングを円滑に行うため、ユーザ (または VAR) は、各アプリケーションのコーディングの適正基準に従う必要があります。問題が発生した場合は、若干の変更をサンプルに加えて問題を再現可能にする必要があります。特定のアプリケーションコードには、不要な複雑さで解決処理を遅くする独自のデータまたはメソッドが含まれている可能性があるため、このメソッドを使用してすべての問題を再現する必要があります。

FOCSAM および GETPRV ユーザイグジット

FOCSAM は、さまざまな下位物理検索モジュールを使用する製品内で、キーを使用したシーケンシャルデータソースにアクセスするための基本ロジックです。FOCSAM には、FOCSAM の一部である標準的な下位検索ルーチンの代わりに呼び出し可能なユーザイグジットが含まれます。このイグジットにより、ibi™ 製品内で使用される内部構造に依存しないユーザ定義のコード (DLL) と FOCSAM の論理検索機能 (例、レコード選択ロジック、複数レコードファイル内のミッシングレコードの処理、各種ファイル間の JOIN) を統合することができます。

MODNAME= *value* を含むアクセスファイルは、任意のテーブルで使用される特定の物理 DLL を指定します。DLL 自体には、ユーザ定義サブルーチンに対して ibi™ 標準で規定された同一名のエントリポイントが必要です。DLL は EDACONF ユーザディレクトリに物理的に格納するか、IBICPG 環境変数で別のディレクトリパスを示すよう設定する必要があります。同一モジュール内の追加エントリポイントにもアクセスできますが、追加のエントリポイントを使用する他のモジュールが必要な場合は、プライベートユーザイグジットのモジュールに、「ローダ」ロジックを追加してこれらのエントリポイントを使用可能にする必要があります。これらの規則は、すべてのユーザ定義のロード可能な DLL サブルーチンに適用される規則と同一です。

ダイナミックプライベートユーザイグジットを使用するには

ここでは、ダイナミックプライベートユーザイグジットの使用手順の概要を説明します。

手順

1. FOCSAM、DLL、必須キーの関連性を示すレコードレイアウトを指定します。FOCSAM の観点では、これは特定のマスターファイル内のメタデータレイアウトになります。DLL の観点では、これは API ポインタのいずれかの取得レコードデータの構造になります。
2. 選択したレコードレイアウトと一致するメタデータでマスターファイルを作成します。
3. レコードデータへのアクセスおよび取得に使用する DLL を指定するアクセスファイルを作成します。
4. データアクセスとレコードデータの取得を実行する DLL を設定および作成します。
5. サーバの起動前に DLL のディレクトリパスを示す IBICPG 環境変数を設定するか、EDACONF ユーザディレクトリに DLL を格納します。
6. 別のテーブルへのアクセスと同様にデータにアクセスします (メタデータおよび DLL の実際の仕様はこれより複雑です)。

GETPRV イグジットの全般機能

このイグジットの全般機能は次のとおりです。

- CONTEXT パラメータ - 再入可能性をサポートし、ストレージ要件を削減し、呼び出しのパフォーマンスを強化します。
- 複数の同時実行イグジットプロセッサのサポート - アクセスファイルで提供されます。この場合、特定のユーザイグジットモジュールはマスターファイルに基づいて名前が付けられます。
- 実行時のユーザイグジットの動的呼び出し - この場合、メジャーリリース間での再コンパイルまたは再リンクは必要ありません。

注意：API の開示部分は変更されないため、メジャーリリース間でアプリケーションのコード変更、再コンパイル、再リンクの必要はほぼありません。ただし、ibi™ はそのような変更を行う権限を確保し、このイグジットを使用するユーザは、ソースコードの変更が必要となった場合、またはアプリケーションを新しいプラットフォームに移動する必要がある場合に使用できるよう、ソースコードの保護措置を講じる必要があります。

- 初期化とシャットダウンの呼び出し - イグジットによる初期および最終のハウスキーピングの実行を許可します。
- 制御および読み取りのオプション
 - 制御オプション - (O) OPEN file、(R) OPEN request (position)、(C) CLOSE、(F) FIN (FOCUS FIN または WebFOCUS Agent Exit)
 - 読み取りオプション - (S) Sequential Read、(G) Generic (GE) Read、(E) Direct (EQ) Read
- 再帰的 JOIN での同一ファイル上の複数配置の保持

GETPRV イグジット使用時の機能上の要件

機能的に、このイグジットは、キー順の VSAM データソースまたはそのようなファイルとして表すことができる他のデータソースに対して通常使用される取得コールの代わりに使用されます。このイグジットは、OCCURS 句で表される内部レコード構造や、FOCUS IF/WHERE 条件の下位基準への変換を処理する必要はありません。これらの機能は、FOCSAM 内のドライブ論理によって実行されます。

ユーザ定義のコードは、次の規則に準拠する必要があります。

- コードは、キーの完全な値からレコードの取得を可能にする必要があります。キーは、一意である (E Direct (EQ) Read) と見なされ、アクセスの試行は 1 回のみです (連続読み取りではない)。

- コードは、最初の配置および取得目的では、完全キーまたは部分キー (G Generic (GE) Read) の値以上のレコードの取得を可能にする必要があります。最初の取得後、アクセスはシーケンシャルリードに切り替えられます。ここで、キーの変更が検出され、配置/取得 (ジェネリックリード) ロジックが連続キーに対して繰り返されます (例、IF COUNTRY EQ ENGLAND OR ITALY)。それ以外の場合は、取得を終了します (CLOSE)。アクセスおよび取得対象レコードは、適切な配置およびキー変更の検出を可能にするため、ソート順に並べられていることを前提とします。イグジットのアプリケーションコードで API を使用して固有情報の取得を実行する場合、ソート順の情報はマスクされるか不要になる場合があります。ただし、参照サンプルのようなプレーンテキストファイルのコンテキストで使われる場合は、ソート順情報の必要性は明らかです。
- コードは、ファイルの現在位置から開始して次の論理レコードを取得可能にする必要があります (S Sequential Read)。連続したシーケンシャルリードは、昇順 (ビットごと) でレコードを返す必要があります。セカンダリーキーでの Generic (GE) Read 以降の後続のシーケンシャルリードを除き、キー情報または非キー情報は渡されません。簡略化するため、渡されたキー情報を無視し、FOCSAM として取得されたレコードに選別条件を適用することもできます。ただし、これは効率的ではありません。
- レコードを取得する Direct Read および Generic Read では、後続のシーケンシャルリードのためにファイル内の開始位置を指定する必要があります。要求されたレコードの取得に失敗した Direct Read は、これらの位置を指定する必要がありません。これは、最初のリクエストのみが試行されるためです。
- オープンファイルリクエスト (0) では、コードは論理的または物理的にファイルを開く必要があります。
- シーケンシャルリードの結果、論理的なファイルの末尾に達した場合、レコード取得長さを 0 (ゼロ) に設定することで EOF シグナルが返されます。後続のシーケンシャルリードでは、エラー表示ではなく EOF シグナルを返す必要があります (例、JOIN の処理時)。エラー条件処理変数は、EOF を示すために使用することはできません。
- 後続のオープンリクエストが正常に実行されるためには、結果的にすべてのリソースを解放し、ファイル内の位置をクリアする close 関数の指定が必要です。
- 連続した close コールは問題なく実行される必要があります。たとえば、すでに閉じられているファイルのクローズなどの条件を、エラー条件を生成せずに処理できる必要があります。
- DDNAME に基づき、CONTEXT パラメータを使用して、一意の領域を取得し、保持することができます。
- コードは、連続的に再利用可能で、リエントラントである必要があります。これは、最新のコンパイラおよびリンカの標準ですが、MVS ではこれは AMODE 31 としてリンクします。

GETPRV イグジットの物理実装

GETPRV イグジットは、サブルーチン DLL の構築およびアクセスに関する標準手順に従って、別の動的にロード可能なモジュールとしてリンクされ、呼び出されます。このプロセスの概要は次のとおりです。

- set EDHOME プログラムを記述します。
- GENCPGM で 3-GL アプリケーションを作成し、コンパイルします。
- IBICPG の名前を、アプリケーションが作成されたディレクトリの名前に設定します。
- サーバを起動し、SQL または TABLE リクエストを発行します。

プログラムの正常な作動後は、最後の 2 つの手順 (IBICPG の設定とサーバの起動) のみが必要です。

FOCSAM の READ 機能のみがサポートされます。

GETPRV でのデータアクセスに使用するマスターファイル

GETPRV イグジットを使用してアクセスするデータのマスターファイルおよび属性は、他の FOCSAM データソースの属性と同一です。ただし、SUFFIX の値は PRIVATE にする必要があります (その他の値はイグジットを呼び出しません)。また、主キーは ALIAS=KEY の値を使用して GROUP として指定する必要があります。以下はその例です。

```
FILE=filename, SUFFIX=PRIVATE, $
SEGNAME=ROOT, SEGTYPE=S0, $
GROUP=keyname, ALIAS=KEY, USAGE=xx, ACTUAL=xx, $
  FIELD=fieldname1, ALIAS=aliasname1, USAGE=xx, ACTUAL=xx, $
  FIELD=fieldname2, ALIAS=aliasname2, USAGE=xx, ACTUAL=xx, $
```

セカンダリキーは、通常次のように設定されます。

```
FILE=filename, SUFFIX=PRIVATE, $
SEGNAME=ROOT, SEGTYPE=S0, $
GROUP=keyname, ALIAS=KEY, USAGE=xx, ACTUAL=xx, $
  FIELD=fieldname1, ALIAS=aliasname1, USAGE=xx, ACTUAL=xx, $
  FIELD=fieldname2, ALIAS=aliasname2, USAGE=xx, ACTUAL=xx, $
```



```
FIELD=fieldname2, ALIAS=KEY1      , USAGE=xx, ACTUAL=xx, INDEX=I, $
FIELD=fieldname2, ALIAS=KEY2      , USAGE=xx, ACTUAL=xx, INDEX=I, $
```

また、次のように設定することもできます。

```
FILE=filename, SUFFIX=PRIVATE, $
SEGNAME=ROOT, SEGTYPE=S0, $
GROUP=keyname , ALIAS=KEY , USAGE=xx, ACTUAL=xx, $
FIELD=fieldname1, ALIAS=aliasname1, USAGE=xx, ACTUAL=xx, $
FIELD=fieldname2, ALIAS=aliasname2, USAGE=xx, ACTUAL=xx, $
GROUP=keyname , ALIAS=KEY1 , USAGE=xx, ACTUAL=xx, INDEX=I, $
FIELD=fieldname2, ALIAS=aliasname3, USAGE=xx, ACTUAL=xx, $
GROUP=keyname , ALIAS=KEY2 , USAGE=xx, ACTUAL=xx, INDEX=I, $
FIELD=fieldname2, ALIAS=aliasname4, USAGE=xx, ACTUAL=xx, $
```

キーシーケンシャルソースの記述に関する標準マニュアルに記載されるように、すべてのキーおよび不連続キーの標準規則が、SUFFIX=PRIVATE の使用に適用され、必要に応じて参照する必要があります。

テーブルを参照するには、マスターファイルの物理名 (例、mytable.mas) を指定し、サーバのアプリケーションパス上のアプリケーションディレクトリにマスターファイルを格納する必要があります。アプリケーションファイルの格納に PDS を使用する MVS 構成では、マスターファイルは、DDNAME MASTER に割り当てられた PDS または特定のアプリケーションのマスターファイルの PDS に格納する必要があります。

GETPRV でのデータアクセスに使用するアクセスファイル

アクセスファイルは必要です。アクセスファイルは、MODNAME 値を指定することで、プライベートイグジットの実際の物理名を指定します。DLL 名前規則 (例、libxxxx.so) を使用する UNIX のようなプラットフォームでは、xxxx の部分が MODNAME として指定する値になります。以下はその例です。

```
MODNAME=pgmname, $
```

テーブルを参照するには、アクセスファイルの物理名 (例、mytable.acx) を指定し、サーバのアプリケーションパス上のアプリケーションディレクトリにアクセスファイルを格納する必要があります。アプリケーションファイルの格納に PDS を使用する MVS 構成では (ファイ

ル拡張子を使用しない)、アクセスファイルは、DDNAME ACCESS に割り当てられた PDS または特定のアプリケーションのアクセスファイルの PDS に格納する必要があります。

呼び出しパラメータと作業領域

ユーザ定義の検索ルーチンは、スタンドアロンの DLL プログラムとして記述する必要があります。標準の C 言語規則とプログラム名の最大長 (8 バイト) 以外に、このプログラムに対する制限はほとんどありません。動的なメモリ割り当てを完全にサポートする言語 (アセンブリ言語または COBOL のような C 言語も使用可能) でプログラムを記述することをお勧めします。ただし、拡張パラメータへのアクセスおよび、一部のプラットフォームでの永続アドレスメモリの設定機能に一部制限があります。

文字以外のパラメータのサイズは、長さとポインタで指定されます。32 ビットアプリケーションでは 4 バイト、64 ビットアプリケーションでは 8 バイトです。

ユーザ定義ルーチンの基本機能は、返されるレコードを作業領域に挿入するために、開く、配置、読み取り、閉じるに対する呼び出しパラメータを指定することです。詳細は、[呼び出しパラメータ](#)および[作業領域制御ブロックパラメータ](#)を参照してください。同梱されたサンプルを使用することもできます。

呼び出しパラメータ

下表はパラメータのリストです。

パラメータ	定義
NCHAR	イグジットで指定される長さです。正の数は、取得され、返されるレコードの長さをバイト数で示します。0 (ゼロ) は、取得したレコードがないか、ファイルの末尾 (EOF) に達したことを示します。これらに機能上の違いはありません。 NCHAR は、データを取得するすべての正しい読み取りでは、0 (ゼロ) 以外に設定されます。読み取りオプションの (S)、(E)、(G) で使用されます。

パラメータ	定義
DDN	FOCSAM で指定される 8 バイトの引数です。値は、リクエストのテーブル名に相当します。たとえば、TABLE FILE MYTABLE は、MYTABLE の DDN (DDNAME と呼ばれる) になり、左揃えに配置され、空白でパディングされます。(F) 以外のすべてのオプションで使用されます。このパラメータを変更しないでください。値は通常、開こうとする任意のファイルを示すために使用されます。ただし、ファイル名の詳細情報パラメータが使用可能で空白でない場合は除きます。
ABUF	イグジットで指定されるポインタです。このポインタは、取得され、FOCSAM に返されるレコードの絶対アドレスです。すべての読み取りオプション (S)、(E)、(G) で使用されます。このポインタを変更しないでください。その代わりに、ポインタで指定されたパスを変更できます。
RC	イグジットで指定されるリターンコードの長さです。0 (ゼロ) は、エラーがないことを示します。0 (ゼロ) 以外は、何らかのエラーを示します。0 (ゼロ) 以外の特定の値が特定のエラーハンドラを有効にすることはありません。ハンドラは 1 つのみで、FOC1143 メッセージを返します。特定のテキストが、詳細情報の RETTEXT パラメータで返されると、汎用 FOC1143 メッセージと特定のテキストが表示されます。すべてのオプションで使用されます。
KEY	FOCSAM で指定される 255 バイトの文字列です。これには、ジェネリックリードの直接キーまたは部分キーの完全なキー値が含まれます。シーケンシャルリードでは有意義ではありません。キー値は左揃えで配置され、最大長は 255 バイトです。 イグジットは、正確なキーの長さとフォーマットを把握する必要があります。単一キーの単純なケースでは、値のハードコードで十分といえます。それ以外の場合、情報は詳細情報パラメータから取得することも

パラメータ	定義
	できます。KEY は、オプション (E) および (G) で使用されます。このパラメータを変更しないでください。
OPT	FOCSAM で指定される 4 バイトの引数です。値は、リクエストのタイプを示します。このパラメータを変更しないでください。次のオプションがあります。 READ OPTIONS 'S' = Sequential Read 'E' = Direct Read (EQ) 'G' = Generic Read (GE) CONTROL OPTIONS 'O' = OPEN File 'R' = OPEN Request (position) (再帰的 JOIN で使用) 'C' = CLOSE File 'F' = FIN (FOCUS FIN) - 最終ハウスキーピング手順の実行が必要 これらの制御および読み取り引数には、末尾に 3 つのブランクを含める必要があります。このイグジットの標準 C ヘッダファイル (getprv.h) には、IBI 使用のみの追加参照が含まれる場合があります、顧客使用目的ではサポートされません。
* (getprv_info)	FOCSAM で指定されるポインタです。詳細情報の作業領域構造を指定します (getprv_info)。詳細は、作業領域制御ブロックパラメータを参照してください。 このパラメータを変更しないでください。

注意：イグジットに渡されるパラメータは、最後のパラメータで最上位ビットをオンにする
と区切られません。プログラムでこの最上位ビットのスキャンが実行されないようにしてく
ださい。

作業領域制御ブロックパラメータ

次の作業領域制御ブロックのパラメータリストは、基本の呼び出しパラメータの拡張機能です。リストには次のものが含まれます。

パラメータ	定義
EYECATCH	FOCSAM で指定される 8 バイトの文字列です。値には、常に文字列 'PRIVATE ' が含まれます。
PFMCB	イグジットで指定されるポインタです。値は通常、イグジットプログラムによるオプション (O) 処理で設定され、後続の FOCSAM コールで変更なしで返されます。このパラメータは一般的に、再入可能性の保持に使用される動的作業領域の指定に使用されます。
PFACB	イグジットで指定されるポインタです。値は通常、イグジットプログラムによるオプション (O) 処理で設定され、オプション (C)、(R)、(E)、(G)、(S) では後続の FOCSAM コールで変更なしで返されます。DDN により固有値になります。これは一般的に、物理ファイルのコンテキストとして使用されます。このパラメータは、オプション (F) では有効ではありません。
PFRPL	イグジットで指定されるポインタです。値は通常、イグジットプログラムによるオプション (R) 処理で設定され、オプション (E)、(G)、(S) では後続の FOCSAM コールで変更なしで返されます。上記の PFACB パラメータ内の各表示で固有の値になります。これは一般的に、論理ファイルのコンテキストとして使用されます。このパラメータは、オプション (F) では有効ではありません。
KEYLEN_FIL	FOCSAM で指定される長さです。値は、イグジットによるオプション (O) 処理で設定され、ファイルのキーの全体長さが含まれます。
KEYLEN_REQ	FOCSAM で指定される長さです。値は、イグジットによるオプション (G) および (E) 処理で設定され、複数

パラメータ	定義
	フィールドのグループキーでは KEYLENF より小さい場合があるため、ダイレクトリードでは実際のキーの長さが含まれます。
ERRTEXT	イグジットで指定されるポインタです。値は、返されるエラーメッセージの絶対開始アドレスです。 FOC1143 およびこのメッセージは、イグジットで返される RC パラメータが 0 (ゼロ) 以外の場合に表示されます。
LERRTEXT	イグジットで指定される長さです。値は、上記の ERRTEXT メッセージの長さを含む必要があります。
INDEX	マスターファイルのメタデータに基づき、FOCSAM で指定される、4 バイトのバイナリ整数です。このオプションには、ファイルにアクセスするインデックス番号が含まれます。 0 マスターファイルの主キーです (ALIAS=KEY または DKEY の場合) 1, 2, ... マスターファイルのセカンダリインデックスです。 ALIAS= KEY1, KEY2 through KEYn and INDEX=I または ALIAS= DKEY1, DKEY2 through DKEYn and INDEX=I
RESERVED1	IBI で使用するために予約されたポインタです。
USERID	FOCSAM で指定される 8 バイトの文字列です。値は、

パラメータ	定義
	イグジットにアクセスするユーザ ID です。
TRACENUM	イグジットで指定される長さです。値は、TRACEARR ポインタの行数を含む必要があります。
TRACEARR	イグジットで指定されるポインタです。値は、返され、標準のトレース機能に挿入されるトレース行の絶対開始アドレスです。
RESERVED2、 RESERVED3、 RESERVE4D	IBI の使用に予約された 3 つのポインタのセットです。
FILENAME	FOCSAM で指定されるポインタです。このポインタのパスには、テーブル名に対して FILEDEF または DDNAME が発行された場合、ファイル名が含まれます。以下はその例です。 <pre>FILEDEF MYTABLE DISK /home/ralph/myotherdata.ftm TABLE FILE MYTABLE ... END</pre> ポインタには、<code>/home/ralph/myotherdata.ftm</code> が含まれ、開こうとする実際のファイルとして、またはファイルや機能を示すためのフラグとして使用することができます。
INDEXNAM	FOCSAM で指定される 8 バイトの文字列です。値は、セカンダリキーによるアクセスの場合、セカンダリキーの ALIAS です。特定のリクエストにアクセス可能なセカンダリキーは 1 つのみです。ただし、セカンダリキーが、複数フィールドで構成される GROUP である場合があります。
RESERVED5、 RESERVED6	IBI で使用するために予約された 2 つの long のセットです。

パラメータ	定義
RESERVED7	IBI で使用するために予約されたポインタです。

呼び出しパラメータおよび作業領域制御ブロックパラメータのサンプル C ヘッダファイル実装

参考として、次のファイル (SUFFIX=PRIVATE の getprv.h) が、C プログラムに同梱されています。これには、呼び出しパラメータと作業領域制御ブロックパラメータの両方が含まれます。

このファイルを追加しない場合は、ユーザ独自のヘッダファイルを記述し C プログラムに追加する必要があります。

```
typedef struct getprv_inf_s {
char          eye[8];          /* I:Eye Catcher "PRIVATE "          */
void          *pfmcb;          /* 0:Pointer to handle for getprv    */
/*      Set up by user at first option 0      */
/* I:Pointer to handle for getprv              */
/*      Passed to user by all other calls      */
void          *pfacb;          /* 0:Pointer to handle for file      */
/*      Set up by user at option 0              */
/* I:Pointer to handle for file                */
/*      Passed to user at option C,R,E,G,S      */
void          *pfrpl;          /* 0:Pointer to handle for request   */
/*      Set up by user at option R              */
/* I:Pointer to handle for request              */
/*      Passed to user at option E,G,S          */

long          keylen_fil;      /* I:Key length (whole) for the file */
/*      Used at option 0                      */
long          keylen_req;      /* I:Key length for the direct read request */
/*      Used at direct read options G,E        */
char          *retttext;       /* 0:Native db error msg text        */
long          lretttext;       /* 0:Length of native db error msg text */
long          index;           /* I:Index # by which to access file: */
/*      0 = Primary Key, (usually as a GROUP=) */
/*      coded in MFD as ALIAS= KEY or DKEY      */
/*      1, 2, ... = Secondary Indexes,          */
```



```

/*      coded in MFD as ALIAS = KEYn and INDEX=I  */
/*      or ALIAS = DKEYn and INDEX=I  */

```

```

void*      reserved1; /* IBI Use Only */
char       userid[8]; /* User ID                               */
long       tracenum;  /* Number of 0 terminated trace lines */
char       *tracearr; /* Array of trace lines
*/
void       *reserved2; /* IBI Use Only */
void*      reserved3; /* IBI Use Only */
char       *reserved4; /* IBI Use Only */
char const *filename; /* Full path name for UNIX, Windows, OS400 and
USS. */

```

```

char       indexnam[8]; /* For secondary indexes, ALIAS name from MFD
otherwise */
/* blank.Populated prior to 0 (Open) option in FOCUS
7.2*/
/* and up.Field in MFD must have INDEX=I
*/
long       reserved5; /* IBI Use Only */
long       reserved6; /* IBI Use Only */
void*      reserved7; /* IBI Use Only */
} getprv_inf_t ;

```

```

typedef void getprv_t (
long       *nchar      /* 0:Length of data record read      */
/*      0 = EOF or no record found  */
/*      Used in all read options S,E,G */
,char      *ddn        /* I:DDName to read                  */
/*      Used in all options except F  */
,char      **abuf      /* 0:Address buffer to return records */
/*      used in all read options S,E,G */
,long      *rc         /* 0:Return code.0 if ok else non 0  */
/*      Used in all options           */
,char      *key        /* I:Key value for read              */
/*      Used in read options E and G  */

```

```

,char      *opt        /* I:Read option :                   */
/*      S   Sequential read          */
/*      G   GE read                   */
/*      E   EQ read                   */
/*      Control options               */

```

```

/*      O      Open file                      */
/*      R      Open request (position)         */
/*      C      Close file                      */
/*      F      FIN of focus                    */
,getprv_inf_t * /* in/out for extended info above */
);

```

SUFFIX=PRIVATE のサンプル (getprv.c)

完全サンプル (製品に同梱) は、長すぎるためこのマニュアルにすべて記載することはできません。記載されたセクションは、作成とテストの要件および手順に関する記述です。実際のサンプルには、モデルとして使用可能な広範な注釈が含まれます。

- 次の必須サンプルファイルがあることを確認します。

ファイル	説明
getprv.c	サンプルファイルです。
getprv.h	サンプルで使用される h ファイルです。
getprv.mas	サンプルを呼び出すテストリクエストで使用されるマスターファイルです。
getprv.acx	サンプルを呼び出すテストリクエストで使用されるアクセスファイルです。
getprv.ftm	サンプルを呼び出すテストリクエストで使用されるデータファイルです。
getprv.fex	サンプルで使用されるテーブル (.mas および .acx) を呼び出すテストリクエストのプロシジャです。

マスターファイル、アクセスファイル、データまたはプロシジャ (focexec) がない場合は、getprv.c ファイルでこれらの作成方法が説明されています。getprv.c および getprv.h ファイルがない場合は、技術サポートに問い合わせてください。

- その他の要件には、インストール済みおよび構成済みサーバへのアクセス、gencpgm

(サーバソフトウェアに含まれる)、実行ツール (例、サーバ構成 edastart、rdaapp または WebFOCUS Reporting Server ブラウザインターフェース) があります。gencpgm は、バージョン 7.1.1 以降である必要がありますが、生成するプログラムは、5x 以降のサーバで使用できます。

EDAHOME 環境変数を設定し、次のプラットフォームで gencpgm が正しく動作できるようにします。

プラットフォーム	コマンド
UNIX、USS、OS400	<pre>export EDAHOME=/home/iadmin/ibi/srv71/home</pre>
Windows	<pre>set EDAHOME=C:\ibi\srv71\home</pre>

適切なプラットフォーム固有の gencpgm パスを使用し、DLL を作成します。以下は、UNIX の例です。

```
$EDAHOME/bin/gencpgm.sh -m cpgm getprv.c
```

- DLL の格納先ディレクトリの名前に IBICPG 環境変数 (DLL の検索先のサーバを指示) を設定します。この場合、次のプラットフォームの表記を使用します。

プラットフォーム	コマンド
UNIX、USS、OS400	<pre>export IBICPG='pwd'</pre>
Windows	<pre>set IBICPG=C:\ibi</pre>

別の方法として、作業ディレクトリとして EDACONF ユーザディレクトリを使用することができます。この場合、DLL は自動的に選択されます。

- "edastart -t"、"edastart -x"、または Web コンソールや rdaapp などのフロントエンドツールを使用し、リクエストを実行します。フロントエンドツールを使用して実行

する場合、DLL が EDACONF ユーザディレクトリに存在するか、IBICPG 変数がサーバの起動前に設定されている必要があります。

edastart x で実行するには、適切なプラットフォーム固有のパスを使用します。以下はその例です。

```
/home/iadmin/ibi/srv71/ffs/bin/edastart -x ex getprv
```

- 初期テストによるデータの取得後 (シーケンシャルリード)、テストを変更し、さまざまな IF または WHERE ステートメントを追加して、Generic Read (G) および Direct Read (E) の動作を観察、実行します。サンプルにはさまざまな printf ステートメントが含まれているため、サンプルの動作を追跡することができます。実際のアプリケーションでは、これらのステートメントは削除、コメントアウト、または ifdef 処理されます。

特殊文字および予約語の確認

WebFOCUS Reporting Server ブラウザインターフェースによる特殊文字および予約語の確認方法について説明します。

シノニムの作成時に [フィールド名の確認] のチェックをオンにすると、サーバが特殊文字を調整するとともに、予約語の有無を確認します。

特殊文字の確認

シノニムの作成時に [フィールド名の確認] のチェックをオンにすると、次の特殊文字がアンダースコア (_) に変換されます。

文字	説明	文字	説明
-	ダッシュ	%	パーセント
	ブランク		縦棒
¥	円記号	&	アンパサンド
/	フォワードスラッシュ	(	開き括弧
,	カンマ	)	閉じ括弧
\$	ドル		縦棒
.	ピリオド	<	より小さい
;	セミコロン	>	より大きい
+	プラス符号	'	アポストロフィ

文字	説明	文字	説明
*	アスタリスク	=	等号
.	ピリオド	"	二重引用符

予約語の確認

シノニムの作成時に [フィールド名の確認] のチェックをオンにすると、次の予約語の有無が確認されます。作成したシノニムで、次の予約語を名前として使用することはできません。

予約語
ALIAS、ALL、ALTER、AND、ANY、AS、ASC、AUTHORIZATION、AVG
BEGIN、BETWEEN、BINARY、BIT、BOTH、BY
CALL、CASE、CAST、CHAR、CHARACTER、CHECK、CLOSE、COALESCE、COLFETCH、COMMIT、CONCAT、CONNECT、CORRESPONDING、COUNT、CREATE、CROSS、CURRENT、CURSOR
DATABASE、DATE、DATETIME、DAY、DAYS、DEALLOCATE、DEC、DECIMAL、DECLARE、DEFAULT、DELETE、DESC、DISTINCT、DO、DOUBLE、DROP
ELSE、ELSEIF、END、ESCAPE、EXCEPT、EXECUTE、EXISTS、EXPLAIN
FETCH、FLOAT、FOR、FOREIGN、FROM、FULL、FUNCTION
GET、GRANT、GRAPHIC、GROUP
HAVING、HOUR、HOURS
IF、IMAGE、INCLUDE、IN、INDEX、INDICATOR、INNER、INOUT、INSERT、INT、INTEGER、INTERSECT、INTO、IS、ITERAT

予約語
JOIN
KEY
LEADING、LEAVE、LEFT、LIKE、LOCK、LOGICAL、LONG、LOOP
MAX、MICROSECOND、MICROSECONDS、MILLISECOND、MILLISECONDS、MIN、MINUTE、MINUTES、MONEY、MONTH、MONTHS
NATURAL、NOT、NULL、NULLIF、NUMBER、NUMERIC
OF、ON、ONLY、OPEN、OPTIMIZE、OPTION、OR、ORDER、OUT、OUTER
PACKAGE、PERCENT、PLAN、PRECISION、PREPARE、PRIMARY、PROCEDURE、PROGRAM、PURGE
QUERYNO
RAW、REAL、REFERENCES、REPEAT、RESET、RETURN、RETURNS、REVOKE、RIGHT、ROLLBACK、ROW、ROWS
SCHEMA、SECOND、SECONDS、SELECT、SERIAL、SET、SMALLFLOAT、SMALLINT、SOME、STOGROUP、SUM、SYNONYM、SYSNAME
TABLE、TABLESPACE、TEXT、THEN、TIME、TIMESTAMP、TO、TRAILING、TRUNCATE
UNION、UNIQUE、UNTIL、UPDATE、USER、USER_TYPE_NAME、USING
VALUES、VARBINARY、VARCHAR、VARGRAPHIC、VARYING、VIEW
WHEN、WHERE、WHILE、WITH、WORK
YEAR、YEARS

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, iWay, Omni-Gen, 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.