



ihi™ WebFOCUS®

WebFOCUS アプリケーション作成ガイド

バージョン 9.1.0 以降 | June 2023

目次

1. TIBCO WebFOCUS のアプリケーションロジック	23
3 階層のアプリケーションロジック	23
複数プラットフォームでの処理分散.....	28
アプリケーションの公開	28
TIBCO WebFOCUS コンポーネント	28
セクション 508 アクセシビリティ	33
2. アプリケーションの管理	35
アプリケーションの概要	35
サンプルおよびチュートリアル生成.....	38
アプリケーションツリーのプロシジャおよびメタデータ	40
アプリケーションおよびパスの管理	51
アプリケーションの作成とマッピング.....	51
SQL データベースへのアプリケーションコンテンツの格納.....	55
TIBCO WebFOCUS Client リポジトリへのマッピング.....	59
ネストしたアプリケーションディレクトリ.....	62
ユーザのホームアプリケーションディレクトリ.....	65
アプリケーションパスの構成.....	70
アプリケーションツリーのフィルタ設定.....	74
ファイルの検索.....	78
アプリケーションツリーのソート.....	78
アプリケーションファイルの選択.....	78
アプリケーションコマンドの概要	79
検索パス管理コマンド	83
APP PATH.....	84
APP PREPENDPATH.....	85
APP APPENDPATH.....	85
APP MAP.....	86
APP SET METALLOCATION_SAME.....	87
APP ? METALLOCATION_SAME.....	88
APP SHOWPATH.....	88

アプリケーションおよびファイルの管理コマンド	89
APP CREATE.....	89
APP COPY.....	90
APP COPYF[ILE].....	90
APP MOVE.....	92
APP MOVEF[ILE].....	92
APP DELETE.....	94
APP DELETEDF[ILE].....	94
APP PROPERTY CODEPAGE.....	95
APP RENAME.....	96
APP RENAMEDF[ILE].....	96
APP コマンドによるファイルタイプの指定.....	97
出力リダイレクトコマンド	101
APP HOLD.....	103
APP HOLDDATA.....	104
APP HOLDMETA.....	104
APP FI[LEDEF].....	105
アプリケーションメタデータコマンドおよびカタログメタデータ	105
基本情報の取得.....	106
STATE.....	106
APP LIST.....	107
APP QUERY.....	109
拡張されたカタログ情報の取得.....	111
catalog/sysapps.....	111
catalog/sysfiles.....	112
APP HELP	114
APP コマンドの使用制限	114
メタデータおよびプロシジャへのアクセス	116
検索ルール.....	116
プロシジャファイルの作成ルール.....	117
マスターファイルおよびプロシジャの場所の特定.....	117
既存のデータファイルへのアクセス.....	118

データファイルの作成ルール.....	118
一時ファイルの割り当て	122
一時領域の使用と作成先	122
一時ディスク領域の使用	122
アプリケーションツール	124
EX プロシジャおよび変数.....	124
EX EDAMAIL.....	125
3. ユーザインタフェースのコーディング	133
使用可能なツール	134
検索パスロジックの選択.....	134
TIBCO WebFOCUS Client	136
Servlet の使用	137
動的複数選択ドロップダウンリストの使用	144
ad hoc レポートの有効化	149
JavaScript によるフォームの確認	152
WebFOCUS のオートテンプレート機能	156
オートプロンプトの構成.....	156
レスポンスオートプロンプト	159
レスポンスオートプロンプトページのコンポーネント.....	160
選択リスト.....	161
単純なフィルタ	167
カレンダーコントロール.....	168
レスポンスサンプルコード (Parameter_Type.fex).....	170
HTML オートプロンプト	172
HTML オートプロンプトページのコンポーネント.....	173
オートプロンプトに関する考慮事項	175
パラメータベースのフィルタの定義	178
単純なフィルタのオートプロンプトフォームへの追加.....	178
フィルタへの変数説明の追加.....	179
変数のフォーマット指定.....	180

変数のデフォルト値の設定.....	181
隠し変数の設定.....	183
値の単一選択リストの追加.....	184
複数選択リストの追加.....	188
表示フィールドおよびソートフィールドの静的リストの追加.....	197
表示フィールドおよびソートフィールドの動的リストの追加.....	203
オートプロンプト機能のカスタマイズ.....	210
プロシジャでの HTML テンプレートの指定.....	210
HTML オートプロンプト機能のカスタマイズ.....	211
プロシジャでの HTML テンプレートの指定.....	212
レポートを実行する URL でのプロンプトレベルの指定.....	213
オートプロンプトの内部 XML へのプロシジャ説明の追加.....	214
デフォルト TIBCO WebFOCUS ページへのレポートの表示.....	215
レポート表示用 HTML ページの設計.....	220
HOLD FORMAT HTMTABLE および -HTMLFORM による同一列内アコーディオンレポー トの表示.....	241
HOLD FORMAT HTMTABLE および -HTMLFORM による HTML HFREEZE レポートの表示.....	246
HOLD FORMAT AHTMLTAB および -HTMLFORM による Active テクノロジレポートの表 示.....	250
-HTMLFORM を使用したマルチドリルダウンメニュー項目の表示.....	251
HOLD FORMAT XML および -HTMLFORM の使用.....	254
4. ユーザインターフェースの機能強化.....	255
ヘルパーアプリケーションによるレポートの表示.....	256
複数レポートの制御.....	262
HTML Web ページへの CSRF トークンの追加.....	277
JavaScript によるドリルダウンレポートの作成.....	278
Web ビューア機能.....	284
TIBCO WebFOCUS ビューアによるレポートのナビゲート.....	286
TIBCO WebFOCUS ビューアを開く/閉じる.....	288
TIBCO WebFOCUS ビューアのボタン表示の制御.....	290

ビューアコントロールパネルの使用.....	291
レポートの検索.....	293
Web ビューアによる印刷.....	297
カスケードスタイルシート (CSS) による表示の標準化	297
実行済みレポートの表示	304
5. アプリケーションの制御フロー管理	307
ダイアログマネージャの使用	308
ダイアログマネージャの処理	310
ダイアログマネージャプロシジャの作成	313
プロシジャへのコメントの追加.....	313
変数によるプロシジャのカスタマイズ	314
デフォルト変数値の指定.....	317
式による変数値の指定.....	319
別プロシジャによる変数値の指定.....	322
外部ファイルを使用した変数値の読み取りと書き込み.....	324
外部ファイルを閉じる.....	330
変数値としての日付の指定.....	331
4 桁の西暦日付を含む変数の使用.....	331
変数の演算.....	331
DECODE 関数による変数値の変更.....	332
変数値の検証.....	332
ユーザ入力値の存在、値、長さ、タイプの確認.....	332
データファイルにアクセスせずに変数値を検証 - REGEX.....	338
フォーマット指定に対するユーザ入力値の確認.....	340
値範囲に対するユーザ入力値の確認.....	341
インデックス付き変数による個数集計.....	342
標準引用符付き文字列の作成	344
変数の作成と操作	348
ローカル変数およびグローバル変数の作成と操作.....	348
変数の連結.....	352

変数末尾ブランクの削除.....	352
変数値の表示.....	354
変数へのパラメータ値の割り当て.....	355
システム変数および統計変数の操作.....	355
関数での数値変数の使用.....	364
変数データタイプの指定.....	364
変数の操作.....	365
式での変数の使用.....	366
関数パラメータとしての変数の使用.....	367
数値関数パラメータとしての数値変数の使用.....	367
文字関数パラメータとしての数値変数の使用.....	368
プロシジャ実行の制御.....	369
スタックコマンドの実行とプロシジャの続行.....	369
スタックコマンドの実行とプロシジャの終了.....	370
プロシジャ実行のキャンセル.....	371
プロシジャ実行順序の制御.....	372
無条件分岐の実行.....	372
条件分岐の実行.....	374
複合 -IF テストの実行.....	377
プロシジャでのループの実行.....	378
-INCLUDE によるプロシジャまたはスタイルシートのプロシジャへの挿入.....	381
-INCLUDE コマンドでの完全修飾名の使用.....	396
-INCLUDE によるプロシジャのネスト化.....	398
EXEC による別プロシジャの呼び出し.....	399
プロシジャによる HTML Web ページの拡張.....	400
外部 Web ページの参照.....	400
プロシジャへの HTML コマンド埋め込み.....	401
HTML Web ページへのプロシジャの埋め込み.....	403
HTML Web ページへの変数挿入.....	408
オペレーティングシステムコマンドの発行.....	410

ディレクトリ変更のオペレーティングシステムコマンド.....	411
単一行の複数オペレーティングシステムコマンドの実行.....	411
コマンド出力の確認.....	412
ダイアログマネージャによるパスワードの管理.....	412
アプリケーションへのメッセージの送信.....	413
ダイアログマネージャプロシジャのテストとデバッグ.....	414
アプリケーションデバッグメッセージの表示.....	420
ダイアログマネージャ構文リファレンス.....	421
-* コマンド.....	421
-? コマンド.....	421
-CLOSE.....	422
-DEFAULT.....	422
-DEFAULTH.....	423
-DOS.....	423
-EXIT.....	423
-GOTO.....	424
-HTMLFORM.....	424
-IF.....	425
-INCLUDE.....	426
-label.....	426
-PASS.....	427
-QUIT.....	427
-QUIT FOCUS.....	428
-READ.....	429
-READFILE.....	429
-REPEAT.....	430
-RUN.....	431
-SET.....	431
-TSO.....	432
-TYPE.....	432
-UNIX.....	433
-WINNT.....	433

-WRITE.....	434
6. プロシジャのテストとデバッグ	435
クエリコマンドによるアプリケーションのデバッグ	435
結合構造の表示	437
一時項目 (DEFINE) の表示	438
有効な通貨データソースの表示	440
使用可能なフィールドの表示	440
ファイルディレクトリテーブルの表示	441
マスターファイルフィールド情報の表示	443
FILEDEF で割り当てた現行 ddname の表示	443
定義済み関数の表示	444
HOLD フィールドの表示	444
JOIN 構造の表示	445
国際言語サポートの表示	446
エラーメッセージ説明の表示	447
現行検索パスの表示	448
バージョン番号の表示	448
リモート TIBCO WebFOCUS Reporting Server の値の表示	449
パラメータ設定の表示	449
グラフパラメータの表示	451
接続サーバサイトコードの表示	452
スタイルシートパラメータ設定の表示	453
グローバル変数値の表示	455
使用中ファイルの識別	455
システムテーブルからの動的レポート	457
システムテーブルシノニムの概要.....	457
SYSAPPS - アプリケーションおよびアプリケーションファイルについてのレポート...	459
SYSCOLUM - テーブルおよびテーブル内のカラムについてのレポート.....	460
SYSDEFN - DEFINE FUNCTION についてのレポート.....	461
SYSERR - エラーメッセージファイルについてのレポート.....	462

SYSFILES - メタデータまたはプロシジャのディレクトリ情報についてのレポート.....	463
SYSIMP - インパクト分析情報についてのレポート.....	465
SYSINDEX - インデックス情報のついてのレポート.....	466
SYSKEYS - キー情報についてのレポート.....	467
SYSRPDIR - ストアドプロシジャについてのレポート.....	468
SYSSET - SET パラメータについてのレポート.....	469
SYSSQLOP - 関数情報についてのレポート.....	470
SYSTABLE - テーブル情報についてのレポート.....	471
データタイプについてのレポート.....	472
7. FOCUS データソースへのアクセス	475
USE コマンド	475
FOCUS データソースの指定	476
異なるファイル名の指定	479
新しいデータソースの指定	481
データソースの保護	482
データソースの連結	483
現在有効な USE オプションの表示	486
USE オプションのクリア	487
8. 環境のカスタマイズ	489
SET コマンドの使用	489
SET コマンドの発行方法	490
SET コマンドのコーディング	491
SET パラメータのタイプ	495
演算.....	496
データとメタデータ.....	497
Data Migrator タスク.....	499
日付操作タスク.....	500
WebFOCUS 固有のタスク.....	501
グラフタスク.....	502
メモリ設定および最適化タスク.....	504

レポートコード、コンテンツ、処理のタスク.....	505
レポートレイアウトと表示タスク.....	510
セキュリティタスク.....	516
SET パラメータ構文	517
3D.....	522
ACCBLN.....	522
ACCESSHTML.....	523
ACCESSIBLE.....	523
ACCESSPDF.....	524
ACROSSLINE.....	524
ACROSSPRT.....	525
ACROSSTITLE.....	525
ACRSVRBTITL.....	526
ALL	527
ALLOWCVTERR.....	528
ALTBACKPERLINE.....	529
ARCFGU.....	529
AREXPIRE.....	530
ARGRAPHENGIN.....	531
ARPASSWORD.....	531
ASNAMES.....	531
AUTODRILL.....	532
AUTOFIT.....	533
AUTOINDEX.....	534
AUTOPATH.....	534
AUTOSTRATEGY.....	535
AUTOTABLEF.....	535
AUTOTICK.....	536
BARNUMB.....	536
BASEURL.....	536
BINS.....	537
BLANKEMPTY.....	537

BLANKINDENT.....	538
BOTTOMMARGIN.....	539
BUSDAYS.....	539
BYDISPLAY.....	540
BYPANEL.....	540
CACHE.....	541
CARTESIAN.....	542
CDN.....	542
CENT-ZERO.....	543
CNOTATION.....	544
CNVERR.....	545
CNVERRLIMIT.....	545
COLLATION.....	546
COMPMISS.....	546
COMPOUND.....	547
COMPUTE.....	548
COUNTWIDTH.....	548
CSSURL.....	549
CURRENCY_DISPLAY.....	549
CURRENCY_ISO_CODE.....	550
CURRENCY_PRINT_ISO.....	550
CURRSYMB.....	551
CURSYM_D.....	552
CURSYM_E.....	552
CURSYM_F.....	552
CURSYM_G.....	553
CURSYM_L.....	553
CURSYM_Y.....	553
CUSTOM-PAGE-LENGTH.....	554
CUSTOM-PAGE-WIDTH.....	554
DATE_ORDER.....	554
DATE_SEPARATOR.....	555
DATEDISPLAY.....	556

DATEFNS.....	557
DATEFORMAT.....	557
DATETIME.....	557
DB_INFILE.....	558
DBACSENSITIV.....	559
DBAJJOIN.....	559
DBASOURCE.....	560
DEFCENT.....	561
DEFECHO.....	561
DEFINES.....	562
DIRECTHOLD.....	562
DISPLAYROUND.....	563
DMH_LOOPLIM.....	563
DMH_STACKLIM.....	563
DMPRECISION.....	564
DRILLFOCMISSING.....	564
DRILLMETHOD.....	565
DROPBLNKLINE.....	565
DTSTRICT.....	566
DUPLICATECOL.....	567
EMBEDDABLE.....	567
EMBEDHEADING.....	568
EMPTYCELLS.....	568
EMPTYREPORT.....	569
EQTEST.....	570
ERROROUT.....	570
ESTRECORDS.....	571
EUROFILE.....	571
EXCELSERVURL.....	572
EXL2KLANG.....	572
EXL2KTXDATE.....	573
EXPANDABLE.....	574
EXPANDBYROW.....	574

EXPANDBYROWTREE.....	575
EXTAGGR.....	575
EXTENDNUM.....	576
EXTHOLD.....	577
EXTRACT.....	577
EXTSORT.....	577
FIELDNAME.....	578
FILECASE.....	579
FILECOMPRESS.....	579
FILE[NAME].....	579
FILTER.....	580
FIXRET[RIEVE].....	581
FLOATMAPPING.....	581
FOC144.....	582
FOCEXURL.....	582
FOCFIRSTPAGE.....	582
FOCHTMLURL.....	583
FOCSTACK.....	583
FORMULTIPLE.....	584
GRAPH-PPI.....	584
GRAPHDEFAULT.....	585
GRAPHEDIT.....	585
GRAPHENGINE.....	585
GRAPHSERVURL.....	586
GRID.....	586
GRMERGE.....	587
GRMULTIGRAPH.....	587
GRWIDTH.....	588
GTREND.....	588
HAUTO.....	589
HAXIS.....	589
HCLASS.....	589
HDAY.....	590

HIDENULLACRS.....	590
HISTOGRAM.....	591
HLDCOM_TRIMANV.....	591
HMAX.....	592
HMIN.....	592
HNODATA.....	592
HOLDATTR.....	593
HOLDFORMAT.....	594
HOLDLIST.....	594
HOLDMISS.....	595
HOLDSTAT.....	596
HSTACK.....	596
HTICK.....	596
HTMLARCHIVE.....	597
HTMLCSS.....	597
HTMLEMBEDIMG.....	598
HTMLENCODE.....	598
INDEX.....	599
JOIN_LENGTH_MODE (JOINLM).....	600
JOINOPT.....	600
JPEGENCODE.....	601
JPEGQUALITY.....	602
JSURLS.....	602
KEEPDEFINES.....	603
KEEPFILTERS.....	603
LANG[UAGE].....	604
LAYOUTGRID.....	606
LAYOUTRTL.....	606
LEADZERO.....	607
LEFTMARGIN.....	607
LINES.....	607
LOOKGRAPH.....	608
MATCHCOLUMNORDER.....	608

MAXDATAEXCPT.....	609
MAXLRECL.....	610
MDICARDWARN.....	610
MDIENCODING.....	611
MDIPROGRESS.....	611
MESSAGE.....	612
MISS_ON.....	612
MISSINGTEST.....	613
MULTIPATH.....	614
NEG-ZERO.....	614
NODATA.....	615
NULL.....	615
OFFLINE-FMT.....	616
OLAPGRMERGE.....	616
OLDSTYRECLEN.....	617
ONFIELD.....	617
ONLINE-FMT.....	618
ORIENTATION.....	618
OVERFLOWCHAR.....	619
PAGE[-NUM].....	619
PAGE-SCALE.....	620
PAGESIZE.....	620
PANEL.....	622
PARTITION_ON.....	623
PASS.....	623
PCOMMA.....	624
PCTFORMAT.....	625
PDFLINETERM.....	625
PERMPASS.....	626
PHONETIC_ALGORITHM.....	627
POPUPDESC.....	627
PPTXGRAPHTYPE.....	628
PRFTITLE.....	629

PRINT.....	629
PRINTDST.....	629
PRINTPLUS.....	630
QUALCHAR.....	630
QUALTITLES.....	631
RANK.....	632
RECAP-COUNT.....	632
RECORDLIMIT.....	632
RIGHTMARGIN.....	633
RPAGESET.....	633
SAVEDMASTERS.....	634
SAVEMATRIX.....	634
SHADOW.....	635
SHIFT.....	635
SHORTPATH.....	635
SHOWBLANKS.....	637
SORTMATRIX.....	637
SORTMEMORY.....	638
SPACES.....	638
SQLTOPTTF.....	639
SQUEEZE.....	639
%STRICTMATH.....	640
STYLEMODE.....	640
STYLE[SHEET].....	641
SUBTOTALS.....	642
SUMMARYLINES.....	642
SUMPREFIX.....	643
SUPPRESSDRILLDT.....	644
TARGETFRAME.....	644
TEMP.....	645
TEMPERASE.....	645
TESTDATE.....	646
TIME_SEPARATOR.....	646

TITLELINE.....	646
TITLES.....	647
TOPMARGIN.....	648
UNITS.....	648
USER.....	648
USERFCHK.....	649
USERFNS.....	650
VAUTO.....	650
VAXIS.....	651
VCLASS.....	651
VGRID.....	652
VISBARORIENT.....	652
VMAX.....	653
VMIN.....	653
VTICK.....	653
VZERO.....	654
WARNING.....	654
WEBARCHIVE.....	654
WEBVIEWALLPG.....	655
WEBVIEWCLMSG.....	655
WEBVIEWCLOSE.....	656
WEBVIEWER.....	656
WEBVIEWHELP.....	657
WEBVIEWHOME.....	657
WEBVIEWTARG.....	658
WEBVIEWTITLE.....	658
WEEKFIRST.....	659
WPMINWIDTH.....	660
XLSXLOCALZIP.....	660
XLSXPAGEBRKIGNORE.....	661
XRETRIEVAL.....	661
YRTHRESH.....	662

9. TIBCO WebFOCUS ファイルの定義と割り当て663

TIBCO WebFOCUS ファイルの割り当て	663
Windows および UNIX 環境でのファイルの動的定義.....	666
FILEDEF コマンドによる論理名の割り当て.....	667
FILEDEF で割り当てた現行 ddname の表示.....	669
ファイル割り当てのクリア.....	670
Windows 環境でのアプリケーションファイル	670
Windows 環境でのマスターファイル.....	671
Windows 環境でのマスターファイルの場所の特定.....	671
Windows 環境でのアクセスファイル.....	671
Windows 環境でのプロシジャ.....	672
Windows 環境での TIBCO FOCUS データソース.....	672
Windows 環境での TIBCO FOCUS データソースの外部インデックス.....	673
Windows 環境でのサポート対象データソース.....	673
Windows 環境でのシーケンシャルデータソース.....	674
Windows 環境での WebFOCUS スタイルシートファイル.....	674
Windows 環境での関数ライブラリ.....	674
Windows 環境でのプロファイル.....	675
Windows 環境での WebFOCUS の Web ページ.....	675
Windows 環境での抽出ファイル	675
Windows 環境での HOLD ファイル.....	676
Windows 環境での SAVB ファイル.....	676
Windows 環境での SAVE ファイル.....	677
Windows 環境での HOLDMAST ファイル.....	677
Windows 環境での作業ファイル	678
Windows 環境でのファイルの有無の特定	678
UNIX 環境でのアプリケーションファイル	679
UNIX 環境でのマスターファイル.....	679
UNIX 環境でのアクセスファイル.....	680
UNIX 環境でのプロシジャ.....	680

UNIX 環境での TIBCO FOCUS データソース.....	681
UNIX 環境での TIBCO FOCUS データソースの外部インデックス.....	682
UNIX 環境でのシーケンシャルデータソース.....	682
UNIX 環境での WebFOCUS スタイルシート.....	682
UNIX 環境でのプロファイル.....	683
UNIX 環境での抽出ファイル	683
UNIX 環境での HOLD ファイル.....	683
UNIX 環境での SAVB ファイル.....	684
UNIX 環境での SAVE ファイル.....	684
UNIX 環境での HOLDMAST ファイル.....	685
UNIX 環境での作業ファイル	686
UNIX 環境でのファイルの有無の特定	686
10. ユーロ通貨サポート	689
ユーロ通貨の統合	689
通貨の換算	690
通貨データソースの作成	691
通貨データを格納するフィールドの識別	694
通貨データソースの有効化	696
通貨データの処理	697
有効な通貨データソースのクエリ	701
数値区切り文字	702
拡張通貨記号の選択	704
Legal and Third-Party Notices	709

1

TIBCO WebFOCUS のアプリケーションロジック

Web 用のセルフサービスアプリケーションを開発するには、開発者がプロジェクトの開発を開始する時点から、ユーザが Web ブラウザからレポートリクエストを送信して結果をブラウザに表示する時点までのフローを管理するソフトウェアコンポーネントが必要になります。ローカル WebFOCUS Reporting Server で開発する WebFOCUS アプリケーションのことを「プロジェクト」と呼びます。作成後、テストを完了したプロジェクトは、Web アプリケーションとして公開することができます。

トピックス

- ❑ [3 階層のアプリケーションロジック](#)
 - ❑ [アプリケーションの公開](#)
 - ❑ [TIBCO WebFOCUS コンポーネント](#)
 - ❑ [セクション 508 アクセシビリティ](#)
-

3 階層のアプリケーションロジック

公開したアプリケーションでは、Web の処理能力を十分に活用し、アプリケーションのパフォーマンス、拡張性、保守管理を強化するため、3 階層のアプリケーションロジックを採用しています。アプリケーションを複数の階層に分割することは、アプリケーションの機能を分類する 1 つの方法です。分割したコンポーネントは、異なるプラットフォーム間で公開することができます。詳細は、28 ページの「[複数プラットフォームでの処理分散](#)」を参照してください。

アプリケーションロジックには、次の 3 階層があります。

- ❑ **プレゼンテーション層** 一般に、ユーザインターフェースまたはアプリケーションのフロントエンドとして知られています。WebFOCUS のレポートアプリケーションでは、プレゼンテーション層は HTML ファイル、カスケードスタイルシート、JavaScript、Java クラス、イメージなどの各種 Web ページで構成されます。

プレゼンテーション層は Web サーバに常駐し、Web サーバおよびエンドユーザの Web ブラウザで処理されます。

- ❑ **ユーザインターフェースロジック** アプリケーションロジックの中で、プレゼンテーション層およびデータアクセスロジック以外の部分に該当します。ユーザインターフェースロジックにより、レポートコンポーネントの作成および操作が可能になります。ユーザインターフェースロジックを提供するプログラムとして、WebFOCUS Servlet が構成されています。たとえば、ユーザインターフェースロジックには、OLAP を動作させる WebFOCUS Servlet があります。

このロジックは TIBCO WebFOCUS から提供され、Application Server または Web サーバの Servlet エンジンプラグインに常駐します。

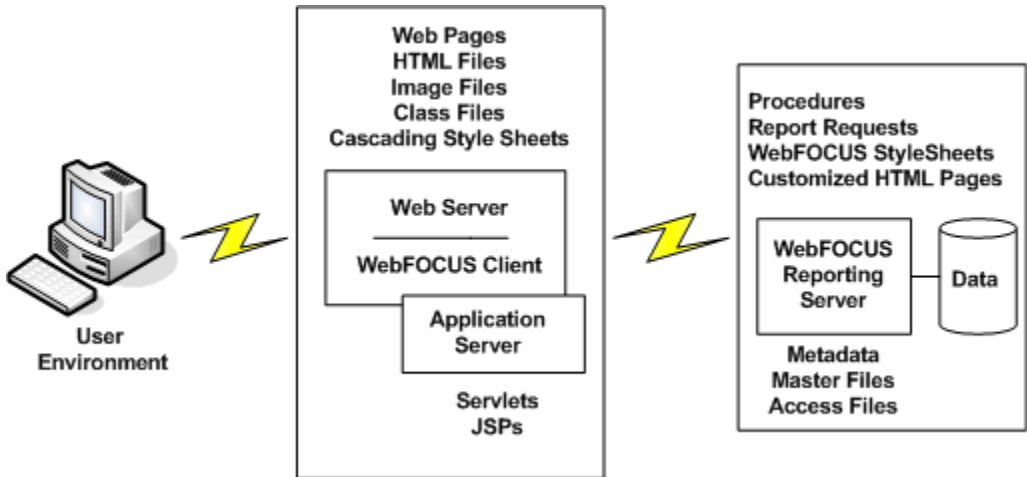
- ❑ **データアクセスロジックおよびフォーマットロジック** データソースから情報を取得します。データアクセスロジックに含まれるメタデータは、アプリケーションのアクセス先となるデータソースを定義します。このロジックは、マスターファイルとアクセスファイルで構成されます。フォーマットロジックは、WebFOCUS 開発者が作成したプロシジャ、WebFOCUS スタイルシート、カスタマイズした特定の HTML ファイル、埋め込みグラフィックスで構成されます。これらのアプリケーションオブジェクトは、レポートまたはグラフを表示したり、エンドユーザが入力する値に基づいて複雑な処理を実行したりします。

通常、データアクセスロジックは、データが格納されたサーバと同一のサーバに常駐します (TIBCO WebFOCUS Reporting Server または SUB Server)。WebFOCUS アプリケーションは、WebFOCUS のミドルウェアテクノロジーのデータアクセスロジックを活用します。

詳細は、25 ページの「[アプリケーションファイルとサーバの分割](#)」を参照してください。

参照 アプリケーションファイルとサーバの分割

下図は、セキュリティを最大限に確保するため、複数のファイアウォールの背後にすべてのユーザインターフェースロジックおよびデータアクセスロジックを構成した様子を示しています。



Web サーバは、Web ページを処理します。この Web ページが、WebFOCUS アプリケーションのプレゼンテーションロジックを提供します。Web サーバは、HTML ファイル、グラフィックイメージファイル、Java クラスファイル、カスケードスタイルシートにアクセスします。Web サーバでのファイルパスの指定方法についての詳細は、対応するプラットフォームの『TIBCO WebFOCUS インストールガイド』を参照してください。

Application Server は、WebFOCUS ツールにユーザインターフェースロジックを提供します。この Application Server に、Servlet、JSP、Java Beans が常駐します。Application Server がユーザインターフェースの背後でアプリケーションロジックを処理することにより、パフォーマンスおよびセキュリティの向上が図られます。

ユーザは、WebLogic、Websphere、Tomcat、Apache、Netscape などの他社製のサーバを選択して使用することができます。

Application Server は、Web サーバと同一の場所に常駐させることも、別の場所に常駐させることもできます。

WebFOCUS Reporting Server は、レポートに使用するユーザインターフェースロジックを内蔵したプロシジャ、およびデータソースのメタデータを格納したファイル进行处理します。

WebFOCUS Reporting Server には、レポートリクエスト、WebFOCUS スタイルシート、WebFOCUS プロシジャから呼び出すカスタマイズされた HTML ページ、マスターファイルおよびアクセスファイルが格納されます。

さらに、WebFOCUS Reporting Server は、アプリケーションが使用するデータソースにもアクセスします。

参照

TIBCO WebFOCUS のファイルタイプ

WebFOCUS アプリケーションは、次のすべてのファイルタイプを公開および使用することができます。下表は、Windows および UNIX オペレーティングシステムで使用するファイル拡張子を示しています。

ファイルタイプ	説明	公開後の場所	ファイル拡張子 (Windows/UNIX)
Web ページ	HTML ファイル、グラフィカルイメージ、Java 実行オブジェクト (Class ファイル)、カスケードスタイルシートなど、エンドユーザが Web ブラウザで表示するファイル。	Web サーバのホームディレクトリまたは Web サーバのエイリアス。	.htm .html .jpg .gif .css .js .class .jar
ユーザインターフェースロジック	Servlet および JSP。	Application Server	.jsp .class

ファイルタイプ	説明	公開後の場所	ファイル拡張子 (Windows/UNIX)
プロシジャ	レポートリクエスト、WebFOCUS スタイルシート、およびダイアログマネージャコマンドの -HTMLFORM を使用して WebFOCUS プロシジャから呼び出されるカスタム HTML など、アプリケーションの実行可能機能が格納されたファイル。ダイアログマネージャコマンドの -HTMLFORM についての詳細は、400 ページの「 プロシジャによる HTML Web ページの拡張 」を参照してください。	WebFOCUS Reporting Server のパス。	.fex .sty .htm
マスターファイル アクセスファイル	マスターファイルおよびアクセスファイル。	WebFOCUS Reporting Server のパス。	.mas .acx
データソース	サポートされているすべてのデータソースタイプ。	WebFOCUS Reporting Server または SUB Server が常駐するプラットフォーム。	なし
一時ファイル	データ抽出ファイル、アプリケーションの処理中に作成される一時ファイル、WebFOCUS が内部的に使用する一時作業ファイル。	EDA 一時ディレクトリ (デフォルト) またはユーザ定義の EDA パス。	.ftm または指定した他の拡張子。

複数プラットフォームでの処理分散

WebFOCUS アプリケーションは、一連の処理を複数のプラットフォームに分散させることができます。処理を分散すると、次の利点があります。

- ❑ **複数プラットフォーム上のデータへのアクセス** 種類の異なるデータソース間に関係を形成することができます。
- ❑ **アプリケーションの実行速度の向上** データにアクセスするプロシジャは、そのデータが存在するプラットフォーム上で実行することができます。これにより、集計または選別がすばやく実行されます。つまり、アプリケーションの実行時に、大量のデータがネットワークを経由することなく、ある特定の場所で集計または選別が実行されます。ネットワークのトラフィックが減少すると、アプリケーションの実行速度が向上します。

詳細は、対応するプラットフォームの『TIBCO WebFOCUS インストールガイド』を参照してください。

アプリケーションの公開

論理的に複数のプロジェクトコンポーネントに分割した WebFOCUS のアプリケーションを作成した場合、これらのコンポーネントを Web サーバおよび WebFOCUS Reporting Server に公開する必要があります。公開とは、所定のサーバにファイルをコピーまたは転送するプロセスのことです。各サーバがファイルを特定するためには、すべてのファイルを正しい場所に配置しておく必要があります。

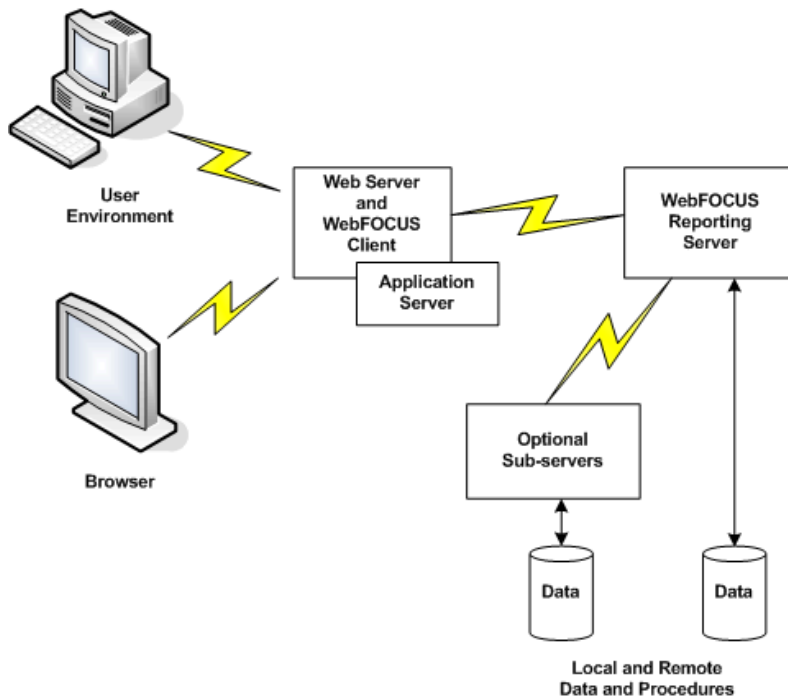
FTP などの外部ツールを使用して、所定のサーバおよびプラットフォームにファイルを転送することができます。その場合は、これらのファイルのパス情報を提供し、処理に必要なファイルの場所を正しく特定できるようにしておく必要があります。

TIBCO WebFOCUS コンポーネント

アプリケーションを作成し、Web に公開するには、そのフローを管理するデータソースコンポーネントが必要になります。WebFOCUS システムは、次のコンポーネントで構成されます。

- ❑ **標準の Web コンポーネント** Web ページの表示を管理する Web サーバおよび Web ブラウザで構成されます。
- ❑ **WebFOCUS 製品コンポーネント** データおよびリクエストの処理を管理する Designer、WebFOCUS Client、サーバコンポーネントで構成されます。
- ❑ **WebFOCUS アプリケーションコンポーネント** ユーザが作成したプロシジャおよびデータで構成されます。

下図は、各コンポーネントの具体的な関係を示しています。



App Studio - この開発環境では、Windows ベースのグラフィカルツールを使用してアプリケーションを作成、テスト、公開することも、必要に応じて手動コーディングにより最大限のカスタマイズを行うこともできます。App Studio に付属の各種ツールを使用して、レポート、グラフ、フォームのデザインなど幅広いタスクを処理します。

Web ブラウザ - Web ページを表示します。この Web ページには、プロジェクトを公開した後にクエリが返すレポートやグラフが出力されます。WebFOCUS は、Microsoft Internet Explorer で正常に動作します。

Web サーバ - リクエストを処理します。リクエストの指示により、ブラウザから HTML ファイルを取得し、ブラウザへファイルを返します。Web ページに WebFOCUS Client への呼び出しが含まれている場合、Web サーバが WebFOCUS Client を起動します。WebFOCUS Client は、変数を収集して WebFOCUS Reporting Server にリクエストを送信します。

Application Server - ユーザインターフェースロジックの処理を担当し、Servlet の実行および JSP のコンパイルを行います。

WebFOCUS Client - Web サーバに常駐します。WebFOCUS Client は、Java Servlet として実装されます。WebFOCUS は、選択したオプションをサポートする標準の Web サーバで正常に動作します。たとえば、Microsoft、IBM、NCSA のそれぞれに対応した Web サーバが存在します。

WebFOCUS Reporting Server - データアクセス、ビジネスロジックの処理、スタイルを設定した出力の生成を担当します。このサーバには、レポートプロシジャ、WebFOCUS スタイルシート、メタデータ (データソースの記述) が格納されます。プロシジャを実行し、データソースにアクセスするには、1 つまたは複数の WebFOCUS Reporting Server が必要です。

オプションの WebFOCUS SUB Server - リモートのプラットフォーム上に存在するデータソースにアクセスします。

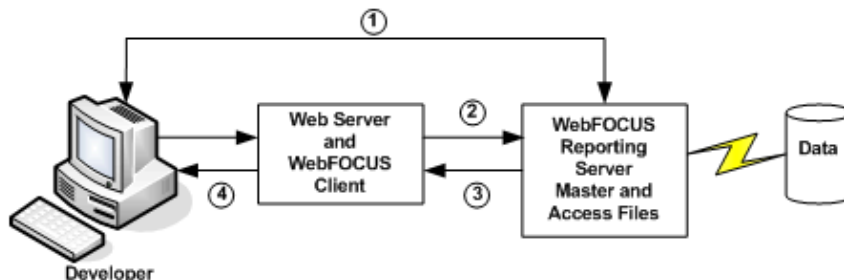
データ 1 つのサーバから、複数のデータタイプにアクセスすることができます。DBMS の各タイプに対して、データアダプタをインストールする必要があります。

プロシジャ WebFOCUS Reporting Server または SUB Server に常駐させることができます。実行時には、WebFOCUS Client がプロシジャ実行のリクエストを送信し、WebFOCUS Reporting Server がそのプロシジャにアクセスして実行します。他のプロシジャを呼び出すプロシジャもあります。呼び出されたプロシジャも、WebFOCUS Reporting Server または SUB Server に常駐します。

次の 3 つの図は、統合された 3 つのフェーズで、上記の各コンポーネントがどのように相互作用し、開発から公開に至る一連のフローを形成しているかを示しています。

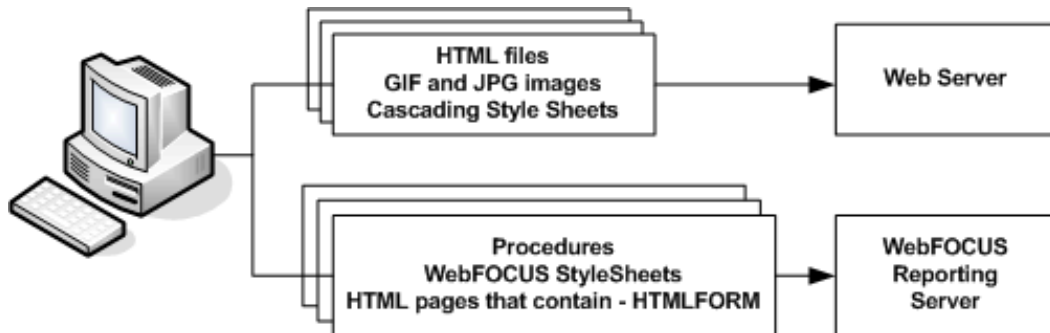
- 31 ページの「[フェーズ 1 - プロジェクトの開発とテスト](#)」
- 31 ページの「[フェーズ 2 - プロジェクトコンポーネントの分割と公開](#)」
- 32 ページの「[フェーズ 3 - 公開したアプリケーションの Web での実行](#)」

参照 フェーズ1-プロジェクトの開発とテスト



1. アプリケーション開発者は、サーバに存在するデータソースのメタデータ (シノニム) を作成します。
2. アプリケーション開発者は、App Studio のグラフィカルツールを使用してアプリケーションを作成します (必要に応じて、コーディングで補足することも可能)。アプリケーションは、基本的にレポートプロシジャおよびそのプロシジャを起動する HTML フォームで構成されます。プロシジャおよび HTML ファイルは、サーバに格納されます。
3. アプリケーション開発者は、プロシジャと HTML フォームを実行およびテストします。各テストでは、リクエストが Web サーバに渡され、WebFOCUS Client を経由して WebFOCUS Reporting Server に送信されます。ここで、Reporting Server がデータを取得してリクエストを処理します。テストプロセスでは、WebFOCUS はサーバ上のマスターファイルおよびアクセスファイルを使用してデータを解析し、そのデータにアクセスします。
4. WebFOCUS Reporting Server が WebFOCUS Client に結果を送信し、その結果が Web サーバに返されて表示されます。

参照 フェーズ2-プロジェクトコンポーネントの分割と公開



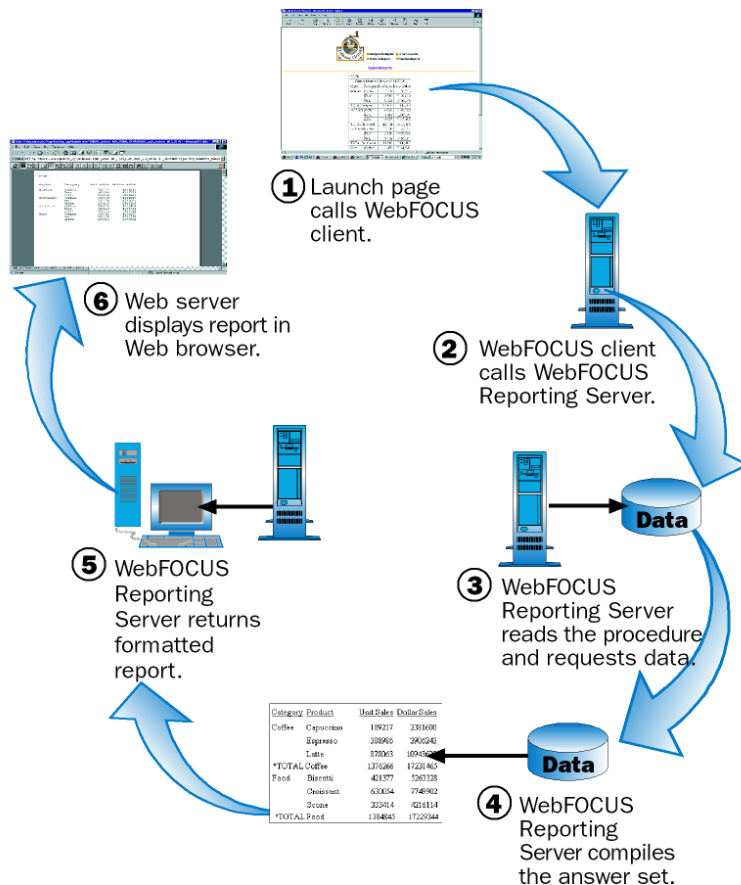
テストの完了後 (フェーズ 1)、開発者はアプリケーションを公開します。

プレゼンテーションロジックは Web サーバに常駐し、ユーザインターフェースロジックは WebFOCUS Reporting Server に常駐します。

開発者は、イメージファイル、Class ファイル、カスケードスタイルシートなどの他のプロジェクトコンポーネントを手動で所定のサーバに追加することができます。

参照

フェーズ3-公開したアプリケーションの Web での実行



1. ユーザは、Web ブラウザで起動ページを開き、レポートを選択します。起動ページは、Web サーバを経由して WebFOCUS Client を呼び出します。リクエストが、実行するレポートプロシジャおよびそのプロシジャに必要なパラメータと値を指定します。必要に応じて、ユーザが Web サーバに接続するために設定した HTTP 環境パラメータをプロシジャに渡すこともできます。プロシジャ名およびすべてのパラメータが、Web サーバ上の WebFOCUS Client に渡されます。

2. WebFOCUS Client は、このリクエストに使用するエージェント処理を開始するよう WebFOCUS Reporting Server に指示します。エージェントが、プロシジャの場所を特定して実行します。
3. レポートプロシジャは、データソース (例、Oracle) からデータを取得するようリクエストします。この場合、WebFOCUS Reporting Server のローカルデータソースまたはリモートデータソースのいずれかを使用することができます。リモートの場合、この図には表示されていませんが、中間に位置する SUB Server 経由でアクセスされます。
4. WebFOCUS Reporting Server は、データソースからのアンサーセットをコンパイルします。
5. WebFOCUS Reporting Server は、リクエストの指示に従ってアンサーセットのフォーマットを設定し (例、HTML、PDF、Excel 2000)、レポート出力を Web サーバに返します。
6. Web サーバは、レポート出力をユーザのブラウザに渡します。ブラウザは、レポートを HTML ページに表示するか、指定したファイルフォーマットに基づいて所定のデスクトップ製品を呼び出して出力を表示します。

セクション 508 アクセシビリティ

WebFOCUS は、米国リハビリテーション法 (U. S. Rehabilitation Act) 第 508 条 (Section 508) で定められた基準を満たすアクセシビリティを提供します。WebFOCUS は、アクセシビリティの基準に準拠するよう次の機能を提供します。

- ❑ ACCESSIBLE および ACCESSHTML SET パラメータは、セクション 508 に準拠する HTML レポートを出力します。ACCESSHTML パラメータは、ACCESSIBLE パラメータ (現在でもサポート対象) より優先されます。これは、ACCESSHTML パラメータが、今後追加されるアクセシビリティ標準をサポートするための拡張性を備えているためです。この HTML コードにより、自動的にページ番号およびフィールドタイトルが非表示になるとともに、ページ見出しがレポートの先頭にのみ、ページ脚注がレポートの末尾にのみ表示されます。
- ❑ ACCESSPDF SET パラメータは、セクション 508 に準拠する PDF レポートを出力します。
- ❑ WebFOCUS スタイルシート属性である SUMMARY は、レポートおよびグラフを HTML <TABLE SUMMARY> タグにマッピングし、レポートまたはグラフの説明を HTML テーブルの概要オブジェクトに配置します。
- ❑ WebFOCUS スタイルシート属性である ALT は、WebFOCUS スタイルシートで指定するドリルダウンレポートに使用します。この属性は、レポートのハイパーリンクに HTML ALT タグをマッピングします。また、HTML レポートに埋め込まれたイメージに説明テキストを追加します。

2

アプリケーションの管理

アプリケーションは、プロシジャ、マスターファイル、アクセスファイル、データファイル、HTML ファイル、PDF ファイル、GIF などの関連するコンポーネントグループの、プラットフォームに依存しない資産です。

さまざまなアプリケーション (APP) コマンドを使用し、アプリケーション自体、そのアプリケーションを構成するコンポーネントファイル、検索パスなどのアプリケーション環境を制御することができます。

トピックス

- ☐ [アプリケーションの概要](#)
- ☐ [アプリケーションツリーのプロシジャおよびメタデータ](#)
- ☐ [アプリケーションおよびパスの管理](#)
- ☐ [アプリケーションコマンドの概要](#)
- ☐ [検索パス管理コマンド](#)
- ☐ [アプリケーションおよびファイルの管理コマンド](#)
- ☐ [出力リダイレクトコマンド](#)
- ☐ [アプリケーションメタデータコマンドおよびカタログメタデータ](#)
- ☐ [APP HELP](#)
- ☐ [APP コマンドの使用制限](#)
- ☐ [メタデータおよびプロシジャへのアクセス](#)
- ☐ [一時ファイルの割り当て](#)
- ☐ [一時領域の使用と作成先](#)
- ☐ [一時ディスク領域の使用](#)
- ☐ [アプリケーションツール](#)

アプリケーションの概要

アプリケーションは、プロシジャ、マスターファイル、アクセスファイル、データファイル、HTML ファイル、PDF ファイル、GIF などの関連するコンポーネントグループの、プラットフォームに依存しない資産です。これらのコンポーネントをリポジトリに格納することにより、サーバ上に特定の領域が確保され、一定の規則に基づいてアプリケーションのコンポーネントを識別し、複数のアプリケーションでこれらのコンポーネントを共有することが可能になります。また、この構造により、ユーザのアプリケーションをプラットフォーム間で移動したり、ローカルで開発したアプリケーションを展開したりする処理が簡素化されます。

これらのコンポーネントは、実行に備えてアプリケーションごとに物理的にグループ化されます。この物理的なグループ化は、共通のルート下のアプリケーションで行うことも、ファイルシステム内の任意の場所にあるアプリケーションにマッピングで行うこともできます。このマニュアルでは、物理アプリケーションの名前またはマッピングした名前を「アプリケーション名」と呼びます。さまざまなアプリケーション (APP) コマンドを使用し、アプリケーションコンポーネントを制御および操作したり、任意のプラットフォームへの書き込みおよび展開を実行したりすることができます。

アプリケーションおよびそのコンポーネントの物理パスは、「**approot**」と呼ばれる構成パラメータで特定します。このパラメータは、インストール時に設定され、サーバ構成ファイルの `edaserve.cfg` に格納されます。このデフォルト値は、プラットフォームにより異なり、それぞれインストール ID のホームディレクトリに対応します。

アプリケーションディレクトリは、ネストすることができます。ネストしたアプリケーションディレクトリは、上位のアプリケーション内に作成されたアプリケーションとなります。詳細は、62 ページの「[ネストしたアプリケーションディレクトリ](#)」を参照してください。

各ユーザにホームアプリケーションディレクトリを作成することもできます。ユーザのホームアプリケーションを提供すると、各ユーザはフルコントロールが与えられた各自のディレクトリを所有して、ユーザ各自のアプリケーションを作成、変更、実行することができます。詳細は、65 ページの「[ユーザのホームアプリケーションディレクトリ](#)」を参照してください。

さまざまなオペレーティングシステムで製品が実行される場合、オペレーティングシステムごとに物理ファイルに対する動作、およびディレクトリやコンポーネントの参照方法が異なります。また、大文字と小文字が区別されるオペレーティングシステムもあれば、区別されないものもあります。たとえば、Windows では、ファイル名の「abc」と「ABC」は同一のファイルを表します。ディスクに保存する際に使用されたファイル名の文字種は関係しません (例、aBc)。一方、UNIX では、これらはすべて異なるファイルです。また、ファイル名に空白を使用できるオペレーティングシステムもあれば、使用できないものもあります。この製品がさまざまなプラットフォームで実行される場合、これらのプラットフォームで名前規則を共存させるには、APP コマンドまたは製品のツール (例、Reporting Server ブラウザインターフェース) を常に使用して、アプリケーションファイルを格納するアプリケーションを作成します。この方法では、大文字と小文字が正しく区別されたファイル名が作成されます。また、外部ツール (例、`mkdir myapp`、`vi mytest.fex`) を使用して、小文字の名前を使用してアプリケーションを作成します。この方法では、ファイルが適切に保存され、APP フレームワーク内で動作します。また、ファイル名に空白を使用することはできません。外部ツールは内部ツールと同様に動作しない場合があるため、外部ツールは使用しないことをお勧めします。

注意

- ❑ この表にディレクトリが記載されている場合、**approot** 値の下に小文字のアプリケーションディレクトリが作成されます。実際のアプリケーション名は小文字で指定する必要があります。また、名前にブランクを使用することはできません。
- ❑ サーバの機能には、アプリケーションのネストを有効化しないと動作しないものもあります。ホームアプリケーションとファイルアップロードがこれに該当します。

プラットフォーム	approot のデフォルト値
UNIX	.../ibi/apps
Linux	.../ibi/apps
Windows	...¥ibi¥apps

インストールの実行時に 2 つのアプリケーションが作成されます。1 つは「baseapp」と呼ばれるデフォルトアプリケーションで、もう 1 つはレガシーサンプルファイルの生成先として使用可能な「ibisamp」と呼ばれるアプリケーションです。また、サーバに接続すると、foccache という一時ディレクトリが検索パスの最初のディレクトリとして追加されます。同一ブラウザセッション内でデータを再利用できるようにする場合は、データを HOLD、SAVE、または SAVB ファイルの形式で foccache ディレクトリに格納します。ブラウザセッションがアクティブの間は、foccache ディレクトリに格納したファイルをリクエストで参照することができます。

特定のアプリケーションコンポーネントへのアクセスは、明示的または暗黙的に行うことができます。明示的アクセスは、実行時の有効な検索パスに依存します。検索パスには、デフォルトアプリケーションの baseapp が必ず含まれています。このアプリケーションを明示的に宣言する必要はありません。

検索パスは、Reporting Server ブラウザインターフェース、Data Migrator デスクトップインターフェース、アプリケーションコード内から変更することができます。また、既存の検索パスの先頭または末尾にアプリケーション名を追加して、検索パスを一時的に変更することもできます。APP コマンドについては、79 ページの「[アプリケーションコマンドの概要](#)」で簡単に説明されています。また、この章の後半で詳細に説明されています。

APPROOT 下の明示的な APP 名または APP MAP コマンド (例、myapp/myproc) のほかに、内部ディレクトリの特別な参照名もあります。これらの参照名には次のものがあります。

参照名	説明
_edatemp	現在のディレクトリです。サーバで使用される場合、これは EDACONF edatemp エージェントディレクトリです (例、ts0000001、0000001 はエージェントの tscomid)。PDS 展開の場合、これは一時 HFS ディレクトリです。edastart -t、-x、-f を使用する場合、これは現在のユーザーディレクトリです。
_edahome	EDAHOME インストールディレクトリです。
_edacnf	EDACONF 構成ディレクトリです。

EDAHOME および EDACONF catalog ディレクトリは内部的にサーバ検索パス上に存在しますが、このような特別なハンドルを使用することで、_edahome/catalog/sysapps 形式の参照名を明示的に使用することができます。sysapps は、多数の内部カタログテーブルの 1 つです。

注意：UNC (Universal Naming Convention) をサポートするプラットフォームでは、approot のネットワークドライブを指定することができます。UNC の条件は次のとおりです。

- ❑ 最初の共有場所から少なくとも 1 フォルダ下のレベル。
- ❑ ブランクは使用しない。以下はその例です。
¥¥mynode¥myshare¥accnting

サンプルおよびチュートリアル生成

さまざまなタイプの TIBCO WebFOCUS および Data Migrator サンプルファイルを生成してアプリケーションフォルダに保存し、サンプルプロシジャを実行して機能や構成をテストすることができます。

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、[新規] メニューから [チュートリアル] オプションを選択すると、[チュートリアルフレームワークの作成] ページが開きます。別の方法として、アプリケーションフォルダを右クリックし、コンテキストメニューから [新規]、[チュートリアル] を順に選択することもできます。

利用可能なチュートリアルは次のとおりです。

- ☐ WebFOCUS - 入門デモ
- ☐ WebFOCUS - Retail デモ (Rserve が構成済みの場合は、Rserve デモも含まれます)
- ☐ WebFOCUS - SSAS キューブ JOIN デモ
- ☐ WebFOCUS - 州人口デモ
- ☐ WebFOCUS - カスタム SQL セキュリティプロバイダ
- ☐ Data Migrator - 全般
- ☐ Data Migrator - イテレータ
- ☐ Data Migrator - ファイルリスナ
- ☐ Data Migrator - スタースキーマ
- ☐ レガシーサンプルテーブルとファイルの作成

以前のバージョンでは、ibisamp という固定のアプリケーションディレクトリが作成され、従来のサンプルファイルおよびデモファイルが多数格納されていました。現在のバージョンでも ibisamp フォルダはベースインストールプロセスの一部として作成されますが、このフォルダに自動的にファイルが格納されることはありません。

現在のチュートリアルには幅広いデモのオプションが用意されていますが、これらは選択時のみロードされます。従来のサンプルファイルおよびデモファイルは、[チュートリアル] ドロップダウンリストから [レガシーサンプルテーブルとファイルの作成] を選択することで、現在も利用可能です。

この変更がフォルダレベルで実装されたため、ibisamp フォルダをレガシーファイルの作成先として使用することも、その他のフォルダを選択することもできます。ドロップダウンリストから [レガシーサンプルテーブルとファイルの作成] オプションを選択すると、以前のバージョンで ibisamp フォルダに格納されていたサンプルファイルの大部分が作成されます。以前のファイルの一部は Data Migrator に関連するもので、これらのファイルは Data Migrator 専用のチュートリアルに移動されています。

アプリケーションツリーのプロシジャおよびメタデータ

[アプリケーション] ページでは、プロシジャ、シノニム、HTML ファイル、データフロー、ユーザ関数、その他のファイルなど、すべてのアプリケーションファイルが単一のアプリケーションツリーに集められています。デフォルト設定では、アプリケーション内のファイルは、ファイルタイプの順に表示されます。ファイルタイプは、各タイプに固有のアイコンで識別され、このアイコンはツリー内のファイル名の左側に表示されます。ただし、リストをソート、またはリストにフィルタを設定して、各アプリケーションのカスタムビューを生成することができます。

項目を右クリックすると、ショートカットメニューが表示されます。コンテキストメニューに表示される項目は、ユーザのロールおよびファイルタイプによって異なります。一部の項目は、管理者または管理者権限を所有するユーザのみが使用することができます。一部のコンテキストメニュー項目は、[ワークスペース] または [マイコンソール] メニュー項目へのショートカットとして機能します。

参照

アプリケーションツリーアイコン

下表は、アプリケーションツリーに表示される各ファイルタイプのアイコンを示しています。

アイコン	ファイルのタイプ
	閉じたアプリケーションディレクトリ
	開いたアプリケーションディレクトリ
メタデータ	
	シノニム (.mas)
	クラスタシノニム (.mas)
	ビジネスビューシノニム (.mas)
プロシジャ	
	プロシジャ (.fex)

アイコン	ファイルのタイプ
	Data Migrator フロー (.fex)
	Change Data Capture (CDC) (.fex)
	DBMS SQL フロー (.fex)
	ダイレクトロードフロー (.fex)
	Change Data Capture (CDC) ダイレクトロードフロー(.fex)
	ユーザ関数 (.fex)
	スケジュールのみ (.fex)
ドキュメント	
	HTML (.htm、.html、.shtml、.htt、.mht、.mhtml、.cfm、.tpl、.hta、.htb)
	Microsoft Excel ドキュメント (.xls、.xlsx、.xlsb、.xht、.xltx、.xlsm、.xltn)
	Adobe Acrobat ドキュメント (.pdf、.ai)
	XML ドキュメント (.xml、.wsd、.xsd、.wsdl、.mxm、.gcl、.xul、.dtd、.xsl、.xslt、.axl)
	JavaScript Object Notation (.json)
	Microsoft PowerPoint ドキュメント (.ppt、.pptx、.pptm)

アイコン	ファイルのタイプ
	Microsoft Word ドキュメント (.doc、.docx、.docm、.dot、.dotx、.dotm)
その他	
	SQL スクリプト (.sql)
	データファイル (.foc、.ftm、.dat、.txt、.csv、.tab、.bdat、.data、.tmp)
	グラフィックファイル (.jpg、.jpeg、.jpe、.svg、.jfif、.tif、.tiff、.ico、.gif、.bmp、.png) 注意： アイコン内のテキストは、イメージファイルのタイプを示します。
	スタイルファイル (.sty、.focstyle、.css)
	アーカイブファイル (.zip、.rar、.tar、.jar、.war)
	スクリプトファイル (.js、.jcs、.omi、.cbl、.hti、.jji、.vbs、.sh、.ctl、.bat、.t3i、.jcl、.ps)
	ログ/トレースファイル (.log、.trc、.hto、.t3o、.sta、.msg)
	構成ファイル (.cfg、.ini、.prf、.err、.nls)
	言語ファイル (.lng)
	その他のドキュメント (.prn、.dif、.fmu、.wp、.ifp、.lzx、.syl、.dmc、.tdl、.bst、.adr、.eps、.swf、.as、.trf、.idx)
	MAINTAIN ファイル (.mpt、.mnt、.fcm、.wfm、.wri、.wxi)

参照

アプリケーションディレクトリのメニューオプション

次のオプションは、[アプリケーション] ページのリボンから利用できます。

新規 新しいアプリケーションファイルを作成することができます。選択可能なメニューオプションには次のものがあります。

- ❑ **クラスタビジネスビュー** ビジネスビューページを開きます。
- ❑ **シノニム** [シノニムエディタ] を開きます。
- ❑ **フロー** フローページを開きます。
- ❑ **DBMS SQL フロー** DBMS SQL フローページを開きます。
- ❑ **プロシジャ** 新規プロシジャをテキストエディタで開きます。
- ❑ **テキストファイル** テキストエディタを開き、新しいファイルをテキストとして作成することができます。
- ❑ **ローデータのアップロード** アプリケーションディレクトリフォルダにイメージファイルおよびデータファイルをアップロードすることができます。データファイルにシノニムは作成されません。
- ❑ **アプリケーションディレクトリ** 新しいアプリケーションディレクトリフォルダを作成することができます。
- ❑ **チュートリアル** [チュートリアルフレームワークの作成] ページを開き、サンプルのテーブルおよびメタデータを作成することができます。

データの取得 [データの取得] ページを開き、ファイルのアップロードまたはアダプタおよび接続の構成が行えます。

フィルタ アプリケーションツリーに表示される項目をカスタマイズすることができます。ネストされたアプリケーションディレクトリが存在し、これらのすべてで選択したファイルを単一のリストに表示する場合、ファイルパネルの [...] (参照) ボタンをクリックします。選択可能なメニューオプションには次のものがあります。

- ❑ **プロシジャ** [すべてのプロシジャ] または [スケジュールのみ] のオプションがあります。
- ❑ **シノニム** [すべてのシノニム]、[クラスタのみ]、[ビジネスビューのみ] のオプションがあります。
- ❑ **詳細** [アプリケーションツリーのフィルタ] ページを開き、ファイルのプロパティを指定したり、使用可能なファイルタイプを選択したり、サイズや更新日などのファイルの統計を選択したりできます。詳細は、74 ページの「[アプリケーションツリーの項目をフィルタするには](#)」を参照してください。

- **インパクト分析** シノニム、フィールド、プロシジャの使用に関するレポートを表示します。選択肢として、[プロシジャ別シノニム]、[シノニム別プロシジャ]、[プロシジャ別カラム]、[カラム別プロシジャ]があります。フローレポートの [概要] または [詳細] を選択することもできます。

- **最近** ユーザが最近開いたファイルのリストが表示されます (ユーザの myhome アプリケーションの history.txt)。

- **お気に入り** ユーザがお気に入りに追加したファイルのリストが表示されます。

管理 ユーザのアプリケーションを管理することができます。次のメニューオプションが表示されます。

- **設定** [アプリケーション設定] ウィンドウを開きます。

- **外部リポジトリ** 次のオプションがあります。

SQL リポジトリ SQL リポジトリを作成、管理するオプションを選択して、アプリケーションフォルダのコンテンツを SQL データベースに格納することができます。

WebFOCUS Client リポジトリ WebFOCUS Client リポジトリへのリンク、および WebFOCUS Client リポジトリの表示、削除のオプションを選択できます。

- **アプリケーションパス** [アプリケーションパスの構成] ページを開き、アプリケーションパスに含めるアプリケーションディレクトリを選択することができます。

- **スケジュール/Email** [スケジューラエージェント]、[スケジュール済みイベント]、[ログと統計]、[スケジューラスキャンの実行] のオプションがあります。アプリケーションツリーを左側ウィンドウに表示したまま、さまざまな種類のスケジューラページおよびログと統計を表示することができます。

- **マイホームの作成** ユーザのホームアプリケーションディレクトリを作成します。

- **共有リソース** 新しいウィンドウを開き、myhome ディレクトリ下のパーソナルファイルおよびフォルダを共有し、構成済みの共有リソースを表示することができます。

- **ユーザ設定** [アプリケーションの表示設定] ページを開き、アプリケーションツリーの表示をカスタマイズすることができます。次のオプションがあります。

- **すべてのユーザの edahome、edacnf、edaprft、scaroot、edatemp、edalog、focccache を表示** サーバ管理者権限を所有するユーザにアプリケーションディレクトリツリーの内部ディレクトリを表示します。

- **説明の表示** ファイルパネルにファイルの説明を表示します。

- ❑ **APPPath 以外のアプリケーションも表示** アプリケーションツリーに非アクティブディレクトリツリーが追加され、パスに存在しないすべてのアプリケーションがそのツリーに表示されます。

このオプションは、サーバ管理者またはサーバ管理者権限を所有するユーザのみが使用することができます。これらの権限を所有するユーザは、アプリケーションパスに存在しないディレクトリを管理することができます (ファイルの追加、削除、コピー、移動、変更)。

- ❑ **ファイル/ディレクトリキャプションの表示** ファイル名がデフォルトレイアウトで表示される場合、その説明を表示します。
- ❑ **ネストされたアプリケーションのファイルを表示** ネストされたアプリケーションのファイルをファイルのリストに表示します。デフォルト設定では、ネストされたアプリケーションディレクトリの名前が表示され、これらの名前をダブルクリックすると格納されたファイルが表示されます。
- ❑ **アプリケーションプロファイルの表示** 現在のアプリケーションプロファイルを表示します。
- ❑ **アプリケーションツリーにファイルを表示** 新しいウィンドウではなく、アプリケーションツリーにファイルのリストを直接開くことができます。

リセット アプリケーションページの設定をデフォルトに戻し、その他のウィンドウを表示したり非表示にしたりできます。

参照

アプリケーションディレクトリのコンテキストメニューオプション

最上位の [アプリケーションディレクトリ] フォルダと同様に、各アプリケーションディレクトリフォルダのコンテキストメニューには、[新規]、[スケジュールと Email]、[インパクト分析] オプションが表示されます。これらのオプションについての詳細は、43 ページの「[アプリケーションディレクトリのメニューオプション](#)」を参照してください。

アプリケーションディレクトリフォルダのコンテキストメニューには、次のオプションも表示されます。

- ❑ **リフレッシュ** アプリケーション内のファイルのリストを更新します。
- ❑ **新規** クラスタビジネスビュー、シノニム、カスタムコピー、データフロー、プロシジャ、テキストファイルの新規作成、およびローデータ、アプリケーションディレクトリ、チュートリアルアップロードが可能です。
- ❑ **クイックコピー** SQL アダプタが構成されている場合にデータをコピーすることができます。

- ❑ **スケジュールと Email** アプリケーションの [スケジューラエージェント] ページまたは [スケジュール済みイベント] ページを開いたり、スケジューラのスキャンを実行したりできます。
- ❑ **インデックスの作成** 選択したマッピング済みアプリケーションディレクトリでインデックスを作成できます。
- ❑ **インデックスの更新** 選択したマッピング済みアプリケーションで既存のインデックスを更新できます。
- ❑ **インデックスの削除** 選択したマッピング済みアプリケーションで既存のインデックスを削除できます。
- ❑ **ログと統計** [ログと統計] ページを開き、指定した範囲のログレポートおよび統計レポートを表示することができます。
- ❑ **インパクト分析** アプリケーションのインパクト分析およびフローレポートを実行することができます。
- ❑ **コピー** アプリケーションディレクトリをコピーします。
- ❑ **削除** アプリケーションディレクトリを削除します。このオプションは、ユーザ自身で作成したアプリケーションディレクトリに対してのみ使用することができます。
- ❑ **切り取り** アプリケーションディレクトリを切り取ります。このオプションは、ユーザ自身で作成したアプリケーションディレクトリに対してのみ使用することができます。
- ❑ **貼り付け** アプリケーションディレクトリを貼り付けます。
- ❑ **名前の変更** [アプリケーション名の変更] ダイアログボックスを開きます。
- ❑ **権限** このアプリケーションディレクトリの [権限の管理] ウィンドウを開きます。このオプションは、サーバ管理者のみ使用できます。
- ❑ **プロパティ** [プロパティ] ウィンドウが開き、アプリケーションディレクトリのパス、ディレクトリ内のファイル数、ディレクトリの最終変更日、説明が表示されます。表示されたテキストボックスを使用して、説明を編集することもできます。

参照

アプリケーションディレクトリ内のすべてのファイルのコンテキストメニューオプション

すべてのファイルのコンテキストメニューには、標準オプションの [切り取り]、[コピー]、[貼り付け]、[名前の変更]、[削除] が表示されます。

複数のファイルを選択してから、コンテキストメニューでオプションを選択することもできます。連続したファイルのグループを選択するには、1 つ目のファイルをクリックして、Shift キーを押しながら最後のファイルをクリックします。不連続のファイルを選択するには、Ctrl キーを押しながら選択するファイルをクリックします。

すべてのファイルのコンテキストメニューには、次のオプションも表示されます。

- ❑ **開く** 各ファイルタイプに関連付けられたエディタでファイルを開きます。このオプションは、クイッククエリ、HTML ファイル、シノニムのコンテキストメニューには表示されません。
- ❑ **プロパティ** [プロパティ] ウィンドウが開き、ファイルのパス、最終更新日、ジョブタイプ、説明、およびファイルに関連付けられた権限が表示されます。
- ❑ **ダウンロード** このオプションを選択すると、ユーザの [ダウンロード] フォルダにファイルが転送されます。Windows のデフォルト設定では、このフォルダは次のディレクトリです。

`C:\¥users¥username¥Downloads`

シノニムを右クリックした場合は、コンテキストメニューに [マスターファイル] および [アクセスファイル] が表示され、いずれかを選択することができます。

- ❑ **コピー** アプリケーションファイルをコピーします。複数ファイルの選択がサポートされています。Ctrl キーを押しながら複数のファイルを選択することができます。
- ❑ **削除** アプリケーションファイルを削除します。このオプションは、ユーザ自身で作成したアプリケーションディレクトリに対してのみ使用することができます。
- ❑ **切り取り** アプリケーションファイルを切り取ります。このオプションは、ユーザ自身で作成したアプリケーションディレクトリに対してのみ使用することができます。
- ❑ **名前の変更** [アプリケーション名の変更] ダイアログボックスを開きます。
- ❑ **お気に入りに追加** ファイルをお気に入りのリストに追加します。お気に入りのリストを表示するには、リボンの [フィルタ] をクリックし、[お気に入り] を選択します。
- ❑ **権限** このアプリケーションファイルの [権限の管理] ウィンドウを開きます。このオプションは、サーバ管理者のみが使用できます。

- ❑ **プロパティ** [プロパティ] ウィンドウが開き、アプリケーションファイルのパス、最終更新日、説明、および接続ユーザの権限が表示されます。表示されたテキストボックスを使用して、説明を編集することもできます。

参照

ストアドプロシジャのコンテキストメニューオプション

ストアドプロシジャのコンテキストメニューには、[開く]、[削除]、[切り取り]、[名前の変更]、[プロパティ] のオプションが表示されます。これらのオプションについての詳細は、43 ページの「[アプリケーションディレクトリのメニューオプション](#)」を参照してください。

ストアドプロシジャのコンテキストメニューには、次のオプションも表示されます。

- ❑ **詳細を指定して開く** [フローエディタで開く] を選択し、フローエディタでプロシジャを開くことができます。
- ❑ **実行** ファイルを実行します (実行ファイルの場合)。
- ❑ **詳細を指定して実行** メニューが開いて、次のオプションが表示されます。
 - ❑ **実行して Email を送信** プロシジャ開始時または完了時の Email 通知の送信を管理することができます。
 - ❑ **デフォルトを変更して実行** 変数の値を入力後、ファイルを実行します (実行ファイルの場合)。
 - ❑ **ユーザ名を指定して実行** connections_autoprompt プロファイルの設定が y に設定され、使用する接続の [オートプロンプト接続認証情報] が y に設定されている場合、プロシジャの実行前に認証情報の入力が必要とされます。
 - ❑ **負荷実行** プロシジャ実行時のスレッド数、間隔、およびプロシジャを継続しておく時間を設定するためのウィンドウが開きます。この実行の統計および比較を生成するように選択することもできます。
 - ❑ **デフォルト負荷実行** プロシジャのデフォルト実行時のスレッド数、間隔、およびプロシジャを継続しておく時間を設定するためのウィンドウが開きます。この実行の統計および比較を生成するように選択することもできます。
 - ❑ **デバッグ** プロシジャをデバッグモードで実行することができます。コード行をスクロールして、各行の結果を確認することができます。
- ❑ **スケジュールと Email** プロシジャ開始時および完了時の Email 通知の送信を管理したり、変数のデフォルト値を変更したりできます。[スケジューラエージェント] および [スケジュール済みイベント] のリストを開くこともできます。

- ❑ **ログ** 次のオプションがあります。
 - ❑ **最新のログ** 最後に実行したプロシジャのログが表示されます。
 - ❑ **最新の出力** 最後に実行したプロシジャの出力が表示されます。
 - ❑ **ログと統計** [ログと統計] ページを開き、指定した範囲のログレポートおよび統計レポートを表示することができます。
- ❑ **分析レポート** 次のオプションがあります。
 - ❑ **インパクト分析** このプロシジャが検出されたレポートおよびそのプロパティを表示します。
 - ❑ **従属関係の分析** [従属関係の分析] ページが表示されます。選択したプロシジャで使われているシノニムおよび他のプロシジャを確認することができます。
 - ❑ **データ系列** プロシジャ内のフィールドについてのレポートおよびそのプロパティを表示します。

参照

TIBCO Data Migrator およびダイレクトロードフローのコンテキストメニュー

Data Migrator フローおよびダイレクトロードフローのコンテキストメニューには、[スケジュールと Email] および [インパクト分析] オプションが表示されます。これらのオプションについての詳細は、43 ページの「[アプリケーションディレクトリのメニューオプション](#)」を参照してください。また、[実行]、[詳細を指定して実行]、[ログ]、[従属関係の分析] オプションも表示されます。これらのオプションについての詳細は、48 ページの「[ストアドプロシジャのコンテキストメニューオプション](#)」を参照してください。

Data Migrator フローおよびダイレクトロードフローのコンテキストメニューには、次のオプションも表示されます。

- ❑ **フローを開く** データフローエディタでフローを開きます。
- ❑ **送信** フローを送信して処理します。
- ❑ **フローレポート** フローに関する詳細情報が表示されます。これらの情報には、ソースおよびターゲットの情報、ロードオプション、SQL SELECT ステートメント、変換、実行プロパティ、ログ取得があります。

参照 クイッククエリのコンテキストメニューオプション

クイッククエリのコンテキストメニューには、[スケジュールと Email] および [インパクト分析] オプションが表示されます。これらのオプションについての詳細は、43 ページの「[アプリケーションディレクトリのメニューオプション](#)」を参照してください。また、[実行] および [詳細を指定して実行] オプションも表示されます。これらのオプションについての詳細は、48 ページの「[ストアドプロシジャのコンテキストメニューオプション](#)」を参照してください。

参照 HTML ファイルのコンテキストメニューオプション

HTML ファイルのコンテキストメニューには、ファイルの標準オプション (切り取り、コピー、貼り付け、削除、プロパティ) 以外に、[実行] および [ログ] オプションも表示されます。これらのオプションについての詳細は、50 ページの「[シノニムのコンテキストメニューオプション](#)」を参照してください。

HTML ファイルのコンテキストメニューには、次のオプションもあります。

- **編集** HTML コードを編集することができます。
- **表示** ファイルをブラウザで開いたときの表示を確認することができます。

参照 カスタムページのコンテキストメニューオプション

カスタムページのコンテキストメニューには、ファイルの標準オプション (開く、切り取り、コピー、貼り付け、削除、権限、プロパティ) が表示されます。また、カスタムページのコンテキストメニューには [編集] オプションも表示されます。

参照 シノニムのコンテキストメニューオプション

シノニムのコンテキストメニューには、[分析レポート] オプションが含まれます。このオプションについての詳細は、45 ページの「[アプリケーションディレクトリのコンテキストメニューオプション](#)」を参照してください。シノニムのコンテキストメニューオプションには、[ダウンロード]、[コピー]、[削除]、[名前の変更]、[プロパティ] のオプションも含まれます。これらのオプションについての詳細は、48 ページの「[ストアドプロシジャのコンテキストメニューオプション](#)」を参照してください。

シノニムのコンテキストメニューには、次のオプションも表示されます。

- **開く** シノニムをデータアシストで開きます。
- **サンプルデータ** シノニムに対してサンプルレポートを実行します。

- ❑ **データプロファイリング** シノニムに対してレポートを実行し、そのシノニムのセグメント、フィールド、フィールド情報を、フィールド値へのドリルダウンレポートとともに表示します。
- ❑ **分析レポート** このシノニムが検知された場所およびそのプロパティについてのレポートを表示する [インパクト分析]、または [従属関係の分析] ページを表示する [従属関係の分析] を選択することができます。このシノニムを含む、またはこのシノニムへの参照を含むアプリケーションおよびシノニムを表示します。
- ❑ **メタデータ管理** クラスタシノニムの作成、シノニムの更新 (ベースシノニムのみ。クラスタシノニムでは不可)、テキストエディタでのマスターファイルの編集、テキストエディタでのアクセスファイルの表示、シノニムのメタデータの翻訳ファイルのプリペアのオプションを提供します。
- ❑ **データ管理** DBMS テーブルの再作成、テーブルの削除、サンプルデータの挿入、全データの削除、およびデータの表示/変更を行えます。
- ❑ **クイックコピー** [クイックコピー] ページを開き、新規のシノニムおよびテーブルのターゲットパラメータを指定することができます。
- ❑ **カスタムコピー** データアシストを開き、シノニムの編集と強化を行えます。
- ❑ **フロー** デモフローページを開きます。

アプリケーションおよびパスの管理

パス管理タスクは、[アプリケーション] ページから使用できます。このページにアクセスするには、Web コンソールのメニューバーにある [アプリケーション] をクリックします。リソースウィンドウにアプリケーションディレクトリツリーが表示されます。一連のボタンを使用して、パス管理タスクを効率的に実行できます。

アプリケーションの作成とマッピング

アプリケーションは、関連するコンポーネントをグループ化するように設計されています。

- ❑ 新しい物理アプリケーションを作成すると、そのアプリケーションが、選択したプロファイルの検索パスに追加可能になります。
- ❑ アプリケーション名を `approot` 外部の既存の物理ディレクトリにマッピングすると、マッピングされたアプリケーションが、選択したプロファイルの検索パスに追加可能になります。

アプリケーションの作成およびマッピングは、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のいずれかで行えます。

アプリケーションディレクトリの名前は、次の規則に準拠する必要があります。

- ❑ 最大長は 64 バイトです。
- ❑ 複数の部分で構成されたアプリケーションディレクトリでは、名前の合計長さが 512 バイトに制限されます。
- ❑ 名前には有効な任意の文字を含めることができますが、次の文字は除きます。

'	ブランク	,	カンマ
*	アスタリスク	.	ピリオド
?	疑問符	:	コロソ
<	より小さい	"	二重引用符
>	より大きい	'	一重引用符
&	アンパサンド	¥t	タブ
¥	円記号	¥0	Null 終端記号
	縦棒	/	スラッシュ
=	等号 (等しい)		

手順 アプリケーションを作成するには

1. Reporting Server ブラウザインターフェースの [アプリケーション] ページにアクセスするか、Data Migrator デスクトップインターフェースで [サーバ] ノードフォルダを展開します。
2. Reporting Server ブラウザインターフェースのリボンで [新規] をクリックし、[アプリケーションディレクトリ] を選択するか、アプリケーションを右クリックして [新規]、[アプリケーションディレクトリ] を順に選択します。

[新規アプリケーションの作成] ページが表示されます。

3. [アプリケーションタイプ] では、デフォルト値の [APPROOT 下の新規アプリケーション] を使用します。
4. [アプリケーション名] テキストボックスに名前を入力します。
5. 必要に応じて、[アプリケーションが存在する場合、再作成する] のチェックをオンにします。

警告：このオプションを選択すると、既存のアプリケーションおよびそのアプリケーション内のすべてのコンテンツが上書きされます。

6. [説明] テキストボックスに説明を入力します。
7. [APPPATH にディレクトリを追加] オプションは、デフォルトで選択されています。必要に応じて、ディレクトリを追加しない選択をすることもできます。
8. [APPPATH 内の位置] ドロップダウンリストから、位置を選択します。オプションは、[最後] および [最初] です。デフォルト値は [最後] です。
9. [プロファイル] ドロップダウンリストから、プロファイルを選択します。サーバ管理者の場合、デフォルト値は [edasprof] です。その他のユーザの場合、デフォルト値はユーザプロファイルです。
10. [OK] をクリックします。

アプリケーションが [アプリケーションディレクトリ] フォルダに追加されます。

手順 アプリケーションを物理ディレクトリにマッピングするには

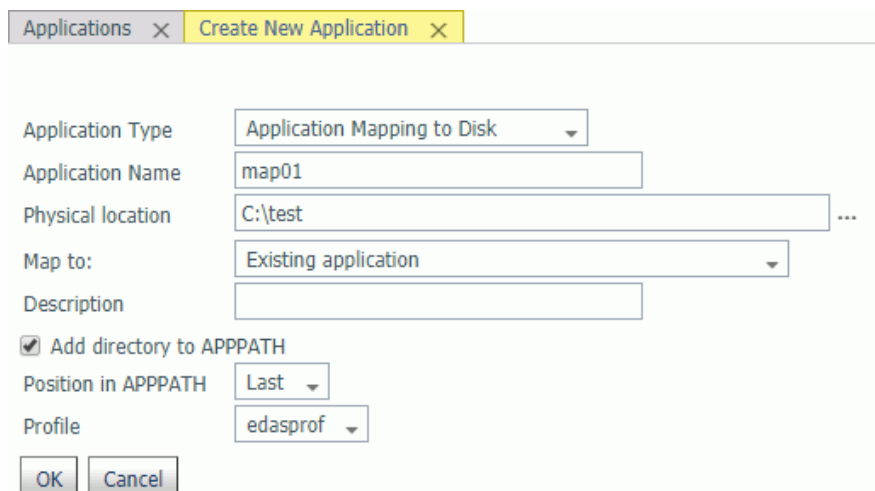
アプリケーション名をファイルシステム内の任意の場所の物理ディレクトリにマッピングすることができます。このアプリケーション名は、その後、APP コマンドで使用することができます。

アプリケーションマッピングは、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、プロファイル単位で追加および削除することができます。

1. [アプリケーション] ページで、リボンから [新規]、[アプリケーションディレクトリ] を順に選択するか、アプリケーションを右クリックして [新規]、[アプリケーションディレクトリ] を順に選択します。

[新規アプリケーションの作成] ページが表示されます。

2. [アプリケーションタイプ] ドロップダウンリストから、[ディスクへのアプリケーションマッピング] を選択します。



Applications × Create New Application ×

Application Type: Application Mapping to Disk

Application Name: map01

Physical location: C:\test ...

Map to: Existing application

Description:

☒ Add directory to APPPATH

Position in APPPATH: Last

Profile: edasprof

OK Cancel

3. [アプリケーション名] テキストボックスに名前を入力します。
4. [物理パス] のデフォルト値を受け入れるか、異なるパスを入力するか、セレクトボタンをクリックしてファイルシステム上のディレクトリに移動します。
5. [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。
6. 必要に応じて、[説明] テキストボックスに説明を入力します。
7. [APPPATH にディレクトリを追加] は、デフォルトで選択されています。必要に応じて、ディレクトリを追加しない選択をすることもできます。

8. [APPPATH 内の位置] ドロップダウンリストから、位置を選択します。オプションは、[最後] および [最初] です。デフォルト値は [最後] です。
 9. [プロファイル] ドロップダウンリストから、プロファイルを選択します。
サーバ管理者の場合、デフォルト値は [edasprof] です。その他のユーザの場合、デフォルト値はユーザプロファイルです。
 10. [OK] をクリックします。
- マッピングが [アプリケーションディレクトリ] フォルダに追加されます。

手順

アプリケーションまたはアプリケーションマッピングを削除するには

アプリケーションディレクトリおよびアプリケーションマッピングは、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のいずれかで削除することができます。

1. Reporting Server ブラウザインターフェースのメニューバーから [アプリケーション] ページにアクセスするか、Data Migrator デスクトップインターフェースで [サーバ] ノードフォルダを展開します。
2. アプリケーションまたはアプリケーションマッピングを右クリックし、[削除] または [マッピングの削除] を選択します。
確認ダイアログボックスが表示されます。
3. [OK] をクリックして、アプリケーションまたはアプリケーションマッピングを削除します。

注意： APP PATH コマンドが APP MAP と同じプロファイルにある場合、アプリケーションマッピングを削除すると、そのマッピングを参照しているすべての APP PATH コマンドから削除されます。

SQL データベースへのアプリケーションコンテンツの格納

アプリケーションフォルダのコンテンツの格納先として、物理ディレクトリを使用する代わりに、SQL データベースを使用することもできます。SQL データベースを使用するには、最初に新しい SQL リポジトリを作成するか、WebFOCUS Client リポジトリへのリンクを作成する必要があります。リポジトリの作成後、リポジトリにマッピングされたアプリケーションを作成し、ファイルを格納することができます。

警告：SQL リポジトリの作成に使用する別の SQL 接続を作成することをお勧めします。既存の SQL 接続を使用した場合、その SQL 接続からアプリケーション内のシノニムを使用できるユーザは、サーバ管理者および WSCFG 権限を持つユーザに限られます。これにより、SQL リポジトリが保護され、権限を持たないユーザ (管理者権限を所有しないユーザ) がシノニムを介して SQL リポジトリのコンテンツにアクセスできなくなります。

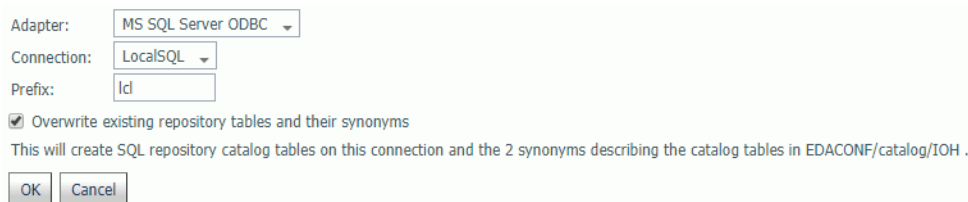
手順

アプリケーションを格納する SQL リポジトリを作成するには

SQL データベースへのアダプタ接続を事前に構成しておく必要があります。

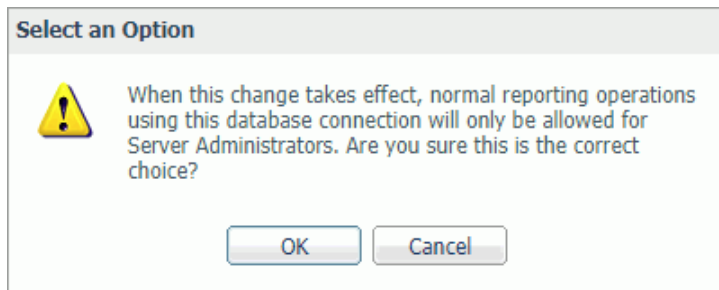
1. [アプリケーション] ページから [管理] をクリックし、[外部リポジトリ]、[SQL リポジトリ]、[新規作成] を順に選択します。

下図のように、[新規 SQL リポジトリの作成] ページが表示されます。



2. [アダプタ] ドロップダウンリストから、構成済みアダプタを選択します。
3. [接続] ドロップダウンリストから、接続を選択します。
4. [接頭語] テキストボックスに、接頭語を入力します。
5. 必要に応じて、[既存のリポジトリテーブルとシノニムを上書きする] のチェックをオンにします。
6. [OK] をクリックします。

下図のように、警告メッセージが表示されます。



7. [OK] をクリックします。

下図のように、この接続で SQL リポジトリカタログテーブルが 2 つ作成されます。

Cancel Create New Application				
Prefix	Adapter	Connection	FILETABLE	RECORDTABLE
<u>Id</u>	MS SQL Server ODBC	LocalSQL	<u>Id</u> iohfiletable	<u>Id</u> iohrecordtable

これらのテーブルは次のとおりです。

`prefixIOHFILETABLE`

`prefixIOHRECORDTABLE`

また、これらのカタログテーブルを記述した 2 つのシノニムが EDACONF/catalog/IOH に作成されます。

- 必要に応じて、[新規アプリケーションの作成] をクリックします。

[新規アプリケーションの作成] ページが表示されます。

手順

SQL コンテンツのアプリケーションを作成するには

SQL リポジトリを事前に作成しておく必要があります。

- [アプリケーション] ページで、[新規]、[アプリケーションディレクトリ] を順に選択します。

[新規アプリケーションの作成] ページが表示されます。

2. 下図のように、[アプリケーションタイプ] ドロップダウンリストから使用するリポジトリ (リポジトリ名、DBMS 名、接続名) を選択します。

The screenshot shows a 'Create New Application' dialog box with the following fields and values:

- Application Type:** Oracle (CON1)
- Application Name:** sql01
- Physical location:** SQL:/repos
- Map to:** Existing application
- Description:** (empty)
- Add directory to APPPATH:** ☒
- Position in APPPATH:** Last
- Profile:** edasprof
- Application IO Attributes:** Application IO Attributes

Buttons: Cancel, OK

注意: 選択項目は、リポジトリ、アダプタタイプ、接続名で構成されます。この例では、これらは CON01 接続で Oracle に対して実行されている repos リポジトリです。

3. 次のパラメータの値を入力または選択します。
 - ☐ **アプリケーション名** デフォルト名を受容するか、新しい名前を入力します。
 - ☐ **物理パス** 自動的に構成された値を受容することも、[...] (参照) ボタンをクリックしてパスを指定することもできます。

- ❑ **マッピング先** [既存のアプリケーション]、[新規アプリケーション (ディレクトリが作成されます)]、または [既存のアプリケーションを再作成 (すべてのファイルは削除されます)] のいずれかを選択します。
- ❑ **説明** 必要に応じて、[説明] テキストボックスに説明を入力します。
- ❑ **APPPATH にディレクトリを追加** 必要に応じて、[APPPATH にディレクトリを追加] のチェックをオフにします。デフォルト設定で、アプリケーションは APPPATH に追加されます。
- ❑ **APPPATH 内の位置** [APPPATH 内の位置] ドロップダウンリストから、アプリケーションの位置を選択します。[最後] または [最初] のいずれかを選択します。デフォルト値は [最後] です。
- ❑ **プロファイル** [プロファイル] ドロップダウンリストから、プロファイルを選択します。サーバ管理者の場合、デフォルト値は [edaprof] です。管理者以外の場合、デフォルト値は各ユーザのユーザプロファイルになります。

4. [OK] をクリックします。

アプリケーションがナビゲーションツリーに追加されます。これで、このアプリケーションを使用して、プロシジャ、シノニム、データファイルなどのコンテンツを格納することができます。

注意：リポジトリへのすべての DBMS 接続および APP MAP コマンドが実行された後、サーバプロファイル、グループプロファイル、ユーザプロファイルの実行中に、リポジトリのプロシジャを実行することも、リポジトリに格納されているマスターファイルやアクセスファイルにアクセスすることもできます。

TIBCO WebFOCUS Client リポジトリへのマッピング

アプリケーションディレクトリを TIBCO WebFOCUS Client リポジトリにマッピングすることができます。

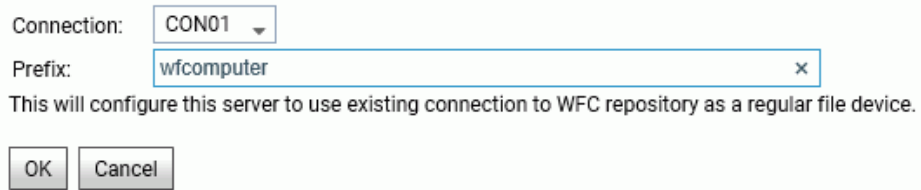
アプリケーションディレクトリを WebFOCUS Client リポジトリにマッピングするには、『TIBCO WebFOCUS データアダプタリファレンス』の説明に従って、WebFOCUS Client REST アダプタを構成しておく必要があります。

手順 既存の TIBCO WebFOCUS Client リポジトリにマッピングするには

1. [アプリケーション] ページで、リボンの [管理] をクリックし、[外部リポジトリ]、[WebFOCUS Client リポジトリ]、[既存項目へのリンク] を順に選択します。

別の方法として、[アプリケーションディレクトリ] を右クリックし、[外部リポジトリ]、[WebFOCUS Client リポジトリ]、[既存項目へのリンク] を順に選択することもできます。

下図のように、WebFOCUS Client リポジトリ接続ページが開きます。



Connection: CON01 ▼

Prefix: wfc01 x

This will configure this server to use existing connection to WFC repository as a regular file device.

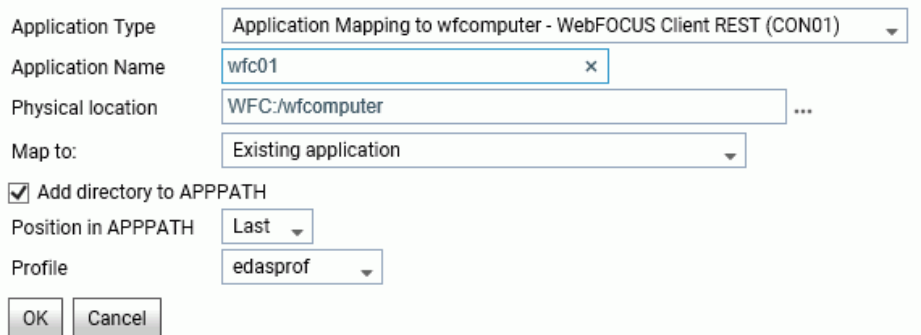
OK Cancel

2. 接頭語を入力するか、デフォルト値を受容します。

接頭語には任意の文字列を使用できますが、デフォルトの接頭語は、WebFOCUS Client を実行しているマシン名に設定されます。この名前は、WebFOCUS Client REST アダプタから取得されます。

3. [OK] をクリックします。

下図のように、[新規アプリケーション] ウィンドウが開き、[アプリケーションタイプ] が WebFOCUS Client REST 接続にマッピングされるよう設定されます。



Application Type: Application Mapping to wfc01 - WebFOCUS Client REST (CON01) ▼

Application Name: wfc01 x

Physical location: WFC:/wfc01 ...

Map to: Existing application ▼

☒ Add directory to APPPATH

Position in APPPATH: Last ▼

Profile: edasprof ▼

OK Cancel

4. 次のパラメータの値を入力または選択します。

アプリケーションタイプ

[アプリケーションマッピング hostname - WebFOCUS Client REST (CON01)] を選択します。

アプリケーション名

マッピングするリポジトリアプリケーションのアプリケーション名を入力するか、デフォルト名を受容します。

物理パス

最初の接続画面のエントリに基づいて、デフォルト WebFOCUS リポジトリパスが自動的に入力されます。必要に応じて、参照ボタンをクリックしてパスを選択することもできます。

マッピング先

次のマッピングオプションのいずれかを選択します。

- ☐ **既存のアプリケーション** 既存のアプリケーションディレクトリにリポジトリディレクトリが追加されます。
- ☐ **新規アプリケーション (ディレクトリが作成されます)** WebFOCUS Client リポジトリの新しいディレクトリが作成されます。
- ☐ **既存のアプリケーションを再作成 (すべてのファイルは削除されます)** 既存のアプリケーションディレクトリが削除され、WebFOCUS リポジトリファイルとともにアプリケーションディレクトリが再作成されます。

APPPATH にディレクトリを追加

このチェックはデフォルト設定でオンになっています。この場合、アプリケーションが [アプリケーションディレクトリ] ツリーに表示されます。

APPPATH 内の位置

デフォルト設定の位置は [最後] です。[最初] を選択することもできます。

プロファイル

ドロップダウンリストからプロファイルを選択します。デフォルト値は [edasprof] です。

5. [OK] をクリックします。

これで、アプリケーションが [アプリケーションディレクトリ] ツリーに追加されます。このアプリケーションディレクトリを使用して、WebFOCUS Client リポジトリファイルを管理することができます。

手順

TIBCO WebFOCUS Client リポジトリへのマッピングを削除するには

1. [アプリケーション] ページで、リボンの [管理] をクリックし、[外部リポジトリ]、[WebFOCUS Client リポジトリ]、[表示と削除] を順に選択します。
別の方法として、[アプリケーションディレクトリ] を右クリックし、[外部リポジトリ]、[WebFOCUS Client リポジトリ]、[表示と削除] を順に選択することもできます。

ウィンドウが開き、アプリケーションとしてマッピングされている TIBCO WebFOCUS Client リポジトリのリストが表示されます。

2. [接頭語] 列でホスト名を右クリックするか、[接頭語] 列の下向き矢印をクリックし、[リモートリポジトリへの参照を削除] を選択します。

参照を削除するかどうかを確認するダイアログボックスが開きます。

3. [OK] をクリックします。

アプリケーションディレクトリは削除されませんが、サーバで TIBCO WebFOCUS リポジトリへの参照が使用できなくなります。

ネストしたアプリケーションディレクトリ

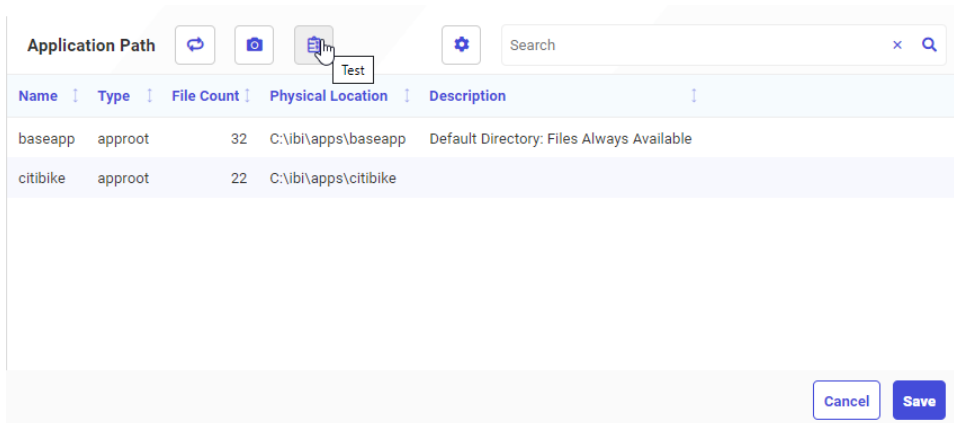
ネストしたアプリケーションディレクトリとは、上位のアプリケーション内に作成されたアプリケーションディレクトリです。サーバのデフォルト設定では、アプリケーションディレクトリは 5 段階までネストすることができます。これより下位または無制限のネストレベルを使用するには、サーバを構成する必要があります。

親ディレクトリがアプリケーションパス上に存在する場合、ネストしたアプリケーションディレクトリは、そのアプリケーションパスに暗示的に追加されます。

APP PATH コマンドは、アプリケーションパスに「ibisamp」を明示的に配置します。

APP PATH baseapp ibisamp

ただし、[管理]、[アプリケーションパス] を順に選択することで、アプリケーションパスをテストして、暗示的に追加されたすべてのディレクトリを確認することができます。[アプリケーションパスの構成] ページが開いた後、下図のように、[アプリケーションパス] ウィンドウの [テスト] をクリックします。



有効なアプリケーションパスに、[ibisamp/dimensions] などのネストされたアプリケーションが追加されています。下図はその例です。



手順

アプリケーションディレクトリのネストレベルを設定するには

1. [アプリケーション] ページのリボンで [管理] をクリックし、[設定] を選択します。
[アプリケーション設定] ウィンドウが開きます。

2. [nested_app] フィールドにアプリケーションのネストレベルを入力するか、ドロップダウンリストから [y] を選択します。デフォルト値は 5 です。[y] を選択すると、アプリケーションのネストレベルが無制限になります。

The screenshot shows a configuration window with the following fields and values:

- foccache_dir**: Empty text field with a search icon.
- foccache_maxage**: Text field containing the value "180".
- foccache_maxsize**: Text field containing the value "0".
- nested_app**: A dropdown menu currently showing the value "5".
- upload_allowed**: A text field containing a list of file extensions: "csv,ftm,dat,tab,tsv,tmp,xls,xlsx,xltm,xltx,xlsm,xlt,xml,xsd,dt,json,c".
- upload_maxsize**: Text field containing the value "0".
- homeapps**: Text field containing the path "c:\1091\ibi\homeapps".

Below the fields, there is a note: "Home Applications feature allows each connected user to keep private versions of hold files, synonyms and other WebFocus objects. Location of home application root can be on Disk (type or use File picker to set physical directory) or SQL (type SQL:/repository_prefix). Note: nested_app must not be 'n' to enable this setting."

At the bottom of the window are three buttons: "Restore Defaults", "Cancel", and "Save".

3. [保存] をクリックします。

サーバの再起動後、アプリケーションフォルダを右クリックし、コンテキストメニューから [新規]、[アプリケーションディレクトリ] を順に選択することにより、新しいアプリケーションサブディレクトリを作成することができます。

注意： ユーザのホームアプリケーションディレクトリを作成するには、ネストしたアプリケーションを有効にしておく必要があります。そのほかにも、ネストしたアプリケーションディレクトリが必要なサーバ機能があります。たとえば、データファイルをアップロードする際は、ファイルは最初に foccache 下の一時アップロードディレクトリに分類され、そこでシノニムに必要な編集や拡張を行うことができます。チュートリアルでもネストしたアプリケーションを使用します。

手順 ネストしたアプリケーションディレクトリを作成するには

1. [アプリケーション] ページでアプリケーションを右クリックし、[新規]、[アプリケーションディレクトリ] を順に選択します。

[新しいアプリケーションディレクトリを作成] ページが表示されます。

2. [アプリケーション名] テキストボックスに名前を入力します。
3. 必要に応じて、[説明] テキストボックスに説明を入力します。
4. [OK] をクリックします。

ネストしたアプリケーションがアプリケーションツリーに追加され、親アプリケーションフォルダの下に配置されます。

ユーザのホームアプリケーションディレクトリ

サーバ管理者は、各ユーザがプライベートアプリケーションディレクトリを所有できるように、サーバを構成することができます。このディレクトリは、「myhome」と呼ばれる場合がありますが、実際には homeapps ディレクトリ下の接続ユーザ ID (~userid) 名ディレクトリです。ユーザ ID ごとにアプリケーションディレクトリを提供することで、ユーザにはこのディレクトリへのフルコントロールが付与され、各自のアプリケーションの作成、変更、実行が可能になります。

サーバ管理者以外のユーザアプリケーションは、Reporting Server ブラウザインターフェースのアプリケーションツリーの 2 箇所に表示されます。これらのアプリケーションには、いずれも同一の物理パスが指定されています。これらの 2 つのアプリケーションは次のとおりです。

- ❑ (接続ユーザの) ~userid アプリケーション - APPPATH の先頭に追加されます。
- ❑ homeapps アプリケーション - サーバ管理者ロール下に登録されたユーザのみが使用可能で、APPPATH の末尾に追加されます。このユーザ ID を使用して、フォルダを展開してユーザホームアプリケーションを表示することができます。

サーバ管理者は、すべてのユーザのホームディレクトリを表示することができます。その他のユーザは、別のユーザがホームディレクトリを共有しない限り、各自のホームディレクトリ (myhome) のみを表示できます。

ホームアプリケーション内に作成されたファイルは、次の方法でプロシジャで参照することができます。

- ❑ 「myhome」をアプリケーション名として使用する場合は、次のように、「myhome」を &APPMYHOME という名前の内部変数にマッピングします。

`APP MAP myhome &APPMYHOME`

これにより、次のように、「myhome」をアプリケーション参照として使用することが可能になります。

EX myhome/fexname

&APPMYHOME 変数をアプリケーション参照として使用することもできます。

このマッピングをすべてのユーザが使用できるようにするには、edasprof サーバプロファイルに記述します。

プロシジャを実行する際、拡張子 (.fex) はオプションです。

- ❑ 特定ユーザのホームディレクトリ内のリクエストを実行するには、このユーザ ID を参照する必要があります。また、サーバユーザを登録する際、プライマリセキュリティプロバイダ名が特定ユーザ ID の先頭に追加され、このユーザ ID のアプリケーション参照にも反映されます。

構文には、アンダースコア (_) で区切りられたセキュリティプロバイダ名とユーザ ID を含めます。PTH セキュリティおよび user1 の場合、次のコマンドを発行します。

EX ~pth_user1/fexname

- ❑ ホームディレクトリは、検索パスに自動的に追加されるため、使用するプロシジャでアプリケーション参照を省略することができます。

EX fexname

同一名のプロシジャが複数のアプリケーションに存在する場合、目的のプロシジャとは異なるプロシジャが実行される場合があります。

1 つ目の参照タイプは、任意のユーザに簡単に拡張することができます。この参照タイプを使用すると、各ユーザのホームアプリケーションに格納されているデータを「myhome/data」として参照する共有アプリケーションを作成することができます。各ユーザがこの共有アプリケーションの同一プロシジャを実行すると、ユーザ自身のホームアプリケーションに格納されているデータに基づいてレポートが作成されます。

2 つ目の参照タイプを使用すると、各ユーザのプロシジャを「homeapps/pgmtst1/proc1.fex」、データを「homeapps/pgmtst1/data」として参照することで、特定のユーザに固有のアプリケーションを実行することができます。この参照タイプは、特定のアプリケーションを共有アプリケーションフォルダに移動する前に、アプリケーションをテストする目的で使用することができます。

ホームアプリケーションディレクトリは、セキュリティが設定されたサーバ上でのみ有効にする必要があります。サーバがセキュリティ OFF で実行されている場合は、すべてのアプリケーション内のファイルのフルコントロールがすべてのユーザに与えられ、ホームディレクトリは意図したとおりに機能しません。ユーザのホームアプリケーションディレクトリを作成するには、ネストしたアプリケーションを有効にしておく必要があります。

サーバ管理者は、すべてのユーザのホームアプリケーションディレクトリをモニタおよび管理することができます。そのため、接続ユーザがサーバ管理者権限を所有している場合は、Reporting Server ブラウザインターフェースのアプリケーションツリーのパスにすべてのユーザのホームディレクトリが表示されます。

サーバがホームアプリケーションに対して有効であると、ユーザのホームアプリケーションディレクトリは自動作成されません。ユーザは、Reporting Server ブラウザインターフェースの [アプリケーション] ページでこれらを作成するか、サーバ管理者に作成を依頼してください。~userid アプリケーションは、ユーザがアプリケーションにファイルをアップロードする場合など、最初の書き込みを試行すると自動的に作成されます。

ホームアプリケーションは、SQL リポジトリに格納することもできます。最初に SQL リポジトリを作成し、アプリケーション設定の [homeapps] パラメータがその SQL リポジトリを指定するよう構成しておく必要があります。詳細は、55 ページの「[SQL データベースへのアプリケーションコンテンツの格納](#)」を参照してください。

ユーザのホームアプリケーションディレクトリが作成されると、次のようになります。

- ❑ ホームアプリケーションが、サーバにより暗示的にユーザのアプリケーションパスに追加されます。ユーザは、APP コマンドを使用してそれをパスから削除することはできません。
- ❑ ホームアプリケーションディレクトリは、常にユーザのアクティブアプリケーションパスの最初にあります。
- ❑ サーバ管理者以外のユーザの場合、アプリケーションツリーにはユーザ自身のホームアプリケーションのみが表示されます。サーバ管理者またはサーバ管理者権限を持つユーザは、すべてのユーザホームアプリケーションを表示することができます。サーバ管理者は、ユーザのホームアプリケーションを管理するための権限をすべて所有します。

- 各ユーザホームアプリケーションディレクトリには、history.txt という名前のファイルが格納されており、このファイルにはユーザが最近開いたファイルのリストが含まれます。デフォルト設定で、履歴ファイルには、ユーザが最近開いたファイルが 10 個含まれます。最近使ったファイルの数を変更する、または最近使ったファイルの保存を無効にするには、[ワークスペース] ページのリボンの [設定] で [環境設定各種情報] をクリックし、[コアエンジンの設定] を選択します。[APP サービス] 下の [HISTORY_DEPTH] の設定で、最近使ったファイルのリストに保存するファイルの数を指定します。この値を 0 (ゼロ) にすると、履歴は保存されません。[アプリケーション] ページのリボンで [フィルタ] をクリックし、[最近] を選択すると、最近使ったファイルのリストを開くことができます。

手順

ホームアプリケーションディレクトリを管理するには

各ユーザのホームディレクトリを設定するには、アプリケーションディレクトリのネストを有効にしておく必要があります。デフォルト設定では、アプリケーションディレクトリは第 5 レベルまでネストすることができます。ユーザのアプリケーションディレクトリが作成されるホームディレクトリは、インストール時に設定されます。デフォルト値は [homeapps] です。homeapps ディレクトリは、[アプリケーション設定] ページで異なる物理ロケーションに変更することができます。すべてのユーザホームディレクトリは、ホームディレクトリ内でネストされます。

1. [アプリケーション] ページで [管理] をクリックし、[設定] を選択します。
[アプリケーション設定] ウィンドウが開きます。
2. homeapps ディレクトリのパスを確認します。[homeapps] パラメータの指定先は、物理ディレクトリに設定することも、既存の SQL リポジトリに設定することもできます。
3. [保存] をクリックして、これらの変更を実装します。
サーバを再起動後、[アプリケーション] メニューに移動します。サーバ管理者権限を所有するユーザの場合、リソースツリーに [ユーザホーム] フォルダが表示され、そのフォルダ下にすべてのユーザホームディレクトリが表示されます。
サーバ管理者権限を所有しないユーザの場合、ユーザ自身のホームディレクトリが [myhome] アプリケーション下に表示されます。このアプリケーションは、APPPATH の先頭に追加されます。

サーバ管理者ユーザの myhome アプリケーションは APPPATH の先頭に追加され、homeapps アプリケーションは APPPATH の末尾に追加されます。サーバ管理者ユーザは、[homeapps] フォルダを展開して、すべてのユーザのホームアプリケーションを表示することができます。

手順

SQL リポジトリにホームアプリケーションを格納するには

最初に SQL リポジトリを作成する必要があります。詳細は、56 ページの「[アプリケーションを格納する SQL リポジトリを作成するには](#)」を参照してください。

1. リボンで [管理] をクリックし、[設定] を選択します。

下図のように、[アプリケーション設定] ウィンドウが開きます。

The screenshot shows the 'Application Settings' dialog box with the following fields and values:

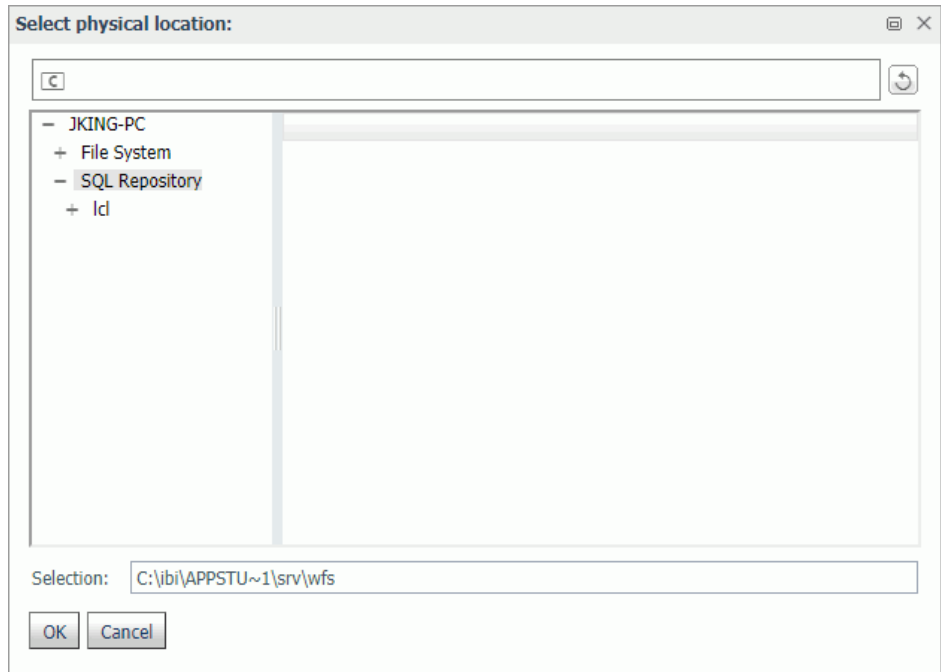
- focache_dir**: Empty text box with a search icon.
- focache_maxage**: Text box containing '180'.
- focache_maxsize**: Text box containing '0'.
- nested_app**: Dropdown menu showing '5'.
- upload_allowed**: Text box containing 'csv,ftm,dat,tab,tsv,tmp,xls,xlsx,xltm,xltx,xlsm,xlt,xml,xsd,dt,json,c'.
- upload_maxsize**: Text box containing '0'.
- homeapps**: Text box containing 'c:\1091\ib\homeapps' with a search icon.

At the bottom, there are three buttons: 'Restore Defaults', 'Cancel', and 'Save'.

2. [homeapps] テキストボックス横の参照ボタンをクリックします。

[物理パスの選択] ダイアログボックスが開きます。

3. 下図のように、[SQL リポジトリ] を選択します。



4. サブフォルダを選択し、[OK] をクリックします。
下図のように、選択したサブフォルダが [homeapps] テキストボックスに入力されます。

? homeapps *

SQL:/Id

5. [保存] をクリックします。

これで、作成するホームアプリケーションが SQL データベースに格納されます。

アプリケーションパスの構成

検索パスに追加可能なアプリケーションは、名前、タイプ、物理パスで識別されます。アプリケーションを作成したときに [APPPATH にディレクトリを追加] のチェックがオンであった場合、そのアプリケーションは検索パスに自動的に追加されています。追加されていない場合は、明示的に検索パスに追加する必要があります。

注意: [アプリケーションパス] ページで新しいプロファイルを作成することもできます。プロファイルは検索パスの保存場所です。

手順 アプリケーションパスを構成するには

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、アプリケーションパスを構成して、アプリケーションまたはマッピングを追加または削除することができます。

アプリケーションパスを構成する新しいユーザインターフェースでは、ダブルリストテーブルが使用されます。

[アプリケーションパスの構成] ページには、Reporting Server ブラウザインターフェースの [アプリケーション] ページからアクセスします。

1. リボンで [管理] をクリックし、[アプリケーションパス] を選択します。

[アプリケーションパスの構成] ページが開き、ダブルリストテーブルが表示されます。下図のように、左側ウィンドウには利用可能なすべてのアプリケーションのリスト、右側ウィンドウにはアクティブアプリケーションパス上のすべてのアプリケーションが表示されます。

Available Applications					Application Path				
Name	Type	File Count	Physical Location	Description	Name	Type	File Count	Physical Location	Description
app01	aproot	1	C:\ibi\apps\app01	ASesri	ibisamp	aproot	167	C:\ibi\apps\ibisamp	
app02	aproot	0	C:\ibi\apps\app02		baseapp	aproot	11	C:\ibi\apps\baseapp	Default Directory: Files Always Available

2. アプリケーションパスにアプリケーションを追加するには、[利用可能なアプリケーション] リストからアプリケーションをドラッグし、[アプリケーションパス] リストの任意の位置にドロップします。

[アプリケーションパス] リストでアプリケーションを上下にドラッグすることで、パス上のアプリケーションの順序を変更することができます。

3. アプリケーションパスからアプリケーションを削除するには、[アプリケーションパス] リストからアプリケーションをドラッグし、[利用可能なアプリケーション] リストにドロップします。

4. 完了後、[保存] をクリックします。

両方のリストで次の操作を行えます。

- ☐ [検索] をクリックして、アプリケーションを検索する。
- ☐ [表示] をクリックして、表示するカラムをカスタマイズしたり、デフォルト表示に戻したりする。

[アプリケーションパス] のメニューバーでは次の操作を行えます。

- ❑ [プロファイルの切り替え] をクリックして、[アプリケーションパスの構成 - プロファイルの選択] ダイアログボックスを開き、ドロップダウンリストから別のプロファイルを選択する。
- ❑ [プレビュー] をクリックして、選択したプロファイルのコマンドをプレビューする。
- ❑ [テスト] をクリックして、APP PATH コマンドおよび有効なアプリケーションパスを確認する。

ナビゲーションウィンドウが更新されます。

注意：新しいプロファイルは、[アプリケーションパスの構成 - プロファイルの選択] ダイアログボックスで作成することもできます。

手順

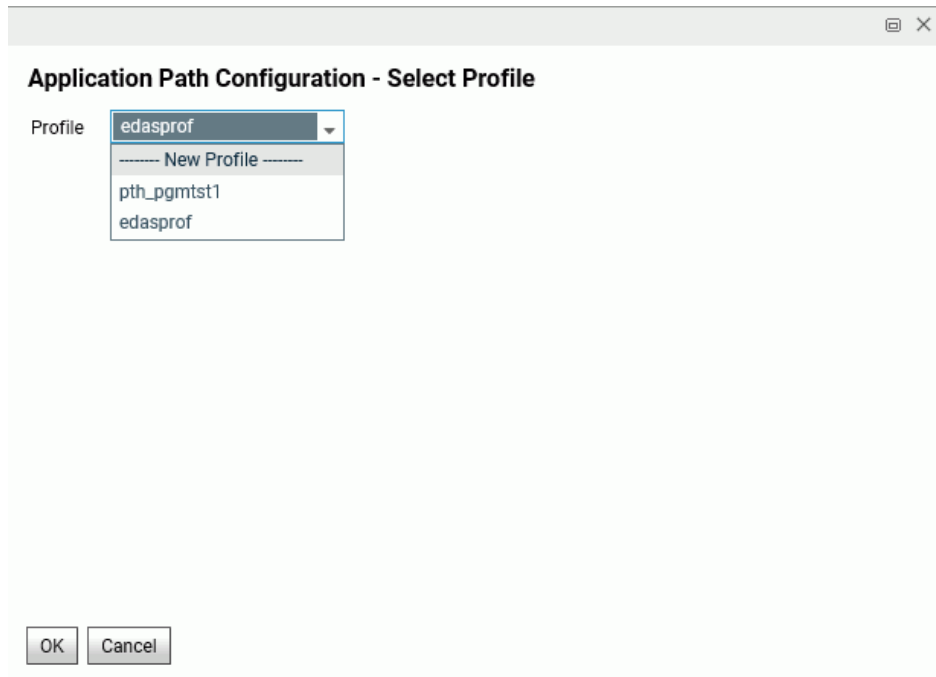
ユーザプロファイル、グループプロファイル、ロールプロファイルでアプリケーションパスを構成するには

アプリケーションパスの構成は、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のいずれかで行えます。

1. [アプリケーション] ページで [管理] をクリックし、[アプリケーションパス] を選択します。

[アプリケーションパスの構成] ページが開きます。

2. [プロファイルの切り替え] ボタンをクリックし、表示されるウィンドウの [プロファイル] ドロップダウンリストから、[新規プロファイル] を選択します。



3. [新規プロファイル名] テキストボックスに名前を入力し、[OK] をクリックします。
4. [プロファイル優先順位の変更] ドロップダウンリストからオプションを選択します。
次のオプションがあります。
 - ☐ 以前に実行したプロファイルから継承
 - ☐ 以前に実行したプロファイルを上書き
 - ☐ 以前に実行したプロファイルの最初に追加
 - ☐ 以前に実行したプロファイルの最後に追加
5. 必要に応じて、[プレビュー] をクリックしてプロファイルを確認します。
6. [保存] をクリックします。

手順 プロファイルを編集するには

1. [ワークスペース] ページに移動します。
2. ナビゲーションウィンドウで [構成ファイル]、[ユーザ/グループプロファイル] フォルダを順に展開します。
3. プロファイルを右クリックし、[編集] を選択します。

プロファイルがテキストエディタに開き、現在のパスが表示されます。

4. パス情報を編集し、[保存] アイコンをクリックします。

ヒント：構成オプションを選択および保存することにより、プロファイルの検索パスを編集することもできます。説明に従いアプリケーションパスを構成します。

アプリケーションツリーのフィルタ設定

フィルタを設定することにより、アプリケーションツリーに表示されるファイルリストをカスタマイズし、選択したファイルのみを表示することができます。フィルタは、ファイルの名前、場所、統計、タイプに基づいて設定することも、複数の項目を組み合わせた選択条件に基づいて設定することもできます。

手順 アプリケーションツリーの項目をフィルタするには

フィルタを設定することにより、アプリケーションツリーに表示される項目をカスタマイズすることができます。選択したフィルタは、ナビゲーションウィンドウに表示されるすべてのアプリケーションに適用されます。

1. [アプリケーション] ページに移動します。
2. [フィルタ] をクリックして、[プロシジャ]、[シノニム]、[詳細] のいずれかを選択します。
 - a. [プロシジャ] を選択した場合は、すべてのプロシジャを表示するか、スケジュール済みプロシジャのみを表示するかを選択することができます。
 - b. [シノニム] を選択した場合は、すべてのシノニムを表示するか、クラスタのみまたはビジネスビューのみを表示するかを選択することができます。
 - c. [詳細] を選択した場合は、[アプリケーションツリーのフィルタ] ページが開きます。

下図のように、[ファイル名] セクションでは、ファイルの名前、拡張子、説明、コンテンツに基づいてフィルタを設定することができます。名前に基づいてフィルタを設定する場合は、ワイルドカード文字としてパーセント記号 (%) を使用することができます。「e%」と指定すると、ファイル名が「e」で始まるすべてのファイルが表示されます。

Applications X Filter Applications Tree X

Set Filter Clear Filter Cancel

File Name

File Name		Sample: dm%
File Extension		Sample: txt
File Description		Sample: my desc
File Content		Sample: my text

下図のように、[ファイルの統計] セクションでは、ファイルのサイズおよび更新日に基づいてフィルタを設定することができます。

File Statistics

File Size At Least

File Size At Most

☒ Files Modified After

2018/9/25

00 ▾ 00 ▾

☒ Files Modified Before

2018/9/25

00 ▾ 00 ▾

下図のように、[ファイルタイプ] セクションでは、ファイルのタイプに基づいてフィルタを設定することができます。

File Type

☐ Metadata

 (mas)

Adapters

☐ Cluster Only

 (mas)

☐ Business View Only

 (mas)

☐ Procedures

☐ Procedures

 (fex)

☐ Flows

 (fex)

☐ Regular Flows with IUD

 (fex)

☐ DBMS SQL Flows

 (fex)

☐ Direct Load Flows

 (fex)

☐ Direct Load Flows with IUD

 (fex)

☐ Custom Copies

 (fex)

☐ Compound Reports

 (fex)

☐ Shared Metadata

 (fex)

☐ Cube Browser

 (fex)

☐ User Functions

 (fex)

☐ Scheduled Only

 (fex)

☐ Documents

☐ HTML

 (htm, html, shtml, htt, mht, mhtml, cfm, tpl, hta, htb)

☐ MS Excel Documents

 (xls, xlsx, xlsb, xht, xltx, xslm, xltm)

☐ Adobe Acrobat Documents

 (pdf, ai)

☐ XML Documents

 (xml, wsd, xsd, wsdl, mxml, gcl, xul, dtd, xsl, xsit, axl, xhtml, xht)

☐ JavaScript Object Notations

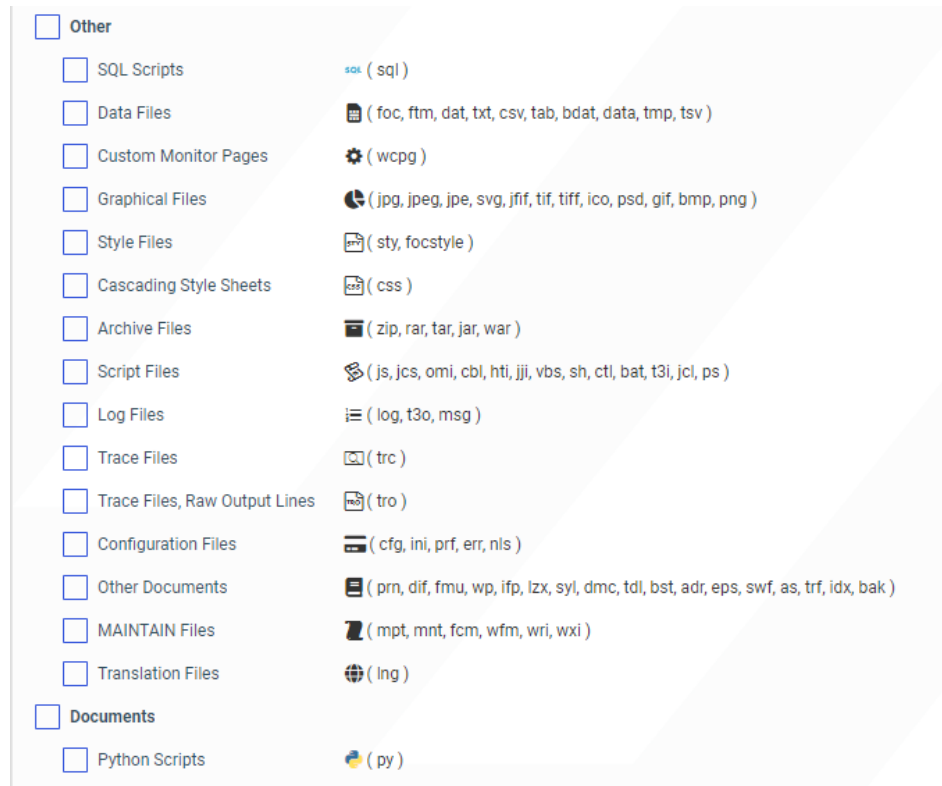
 (json)

☐ MS PowerPoint Documents

 (ppt, pptx, pptm)

☐ MS Word Documents

 (doc, docx, docm, dot, dotx, dotm)



完了後、[フィルタの設定] をクリックします。[アプリケーション] ウィンドウに、フィルタが設定されたことを示すメッセージが表示されます。

[フィルタの設定を解除] をクリックすると、フィルタをクリアすることができます。

- d. [インパクト分析] を選択すると、プロシジャ別シノニム、シノニム別プロシジャ、プロシジャ別カラム、カラム別シノニムを表示するレポートを実行することができます。
- e. [最近] を選択すると、最近開いたファイルのリスト (myhome アプリケーションの history.txt に格納) が表示されます。
- f. [お気に入り] を選択すると、お気に入りに追加したファイルのリストが表示されます。

フィルタを適用すると、[アプリケーションディレクトリ] ツリーのラベル末尾に「(フィルタ済み)」が追加されます。

注意：フィルタの設定を解除するには、[フィルタ] メニューで [クリア] を選択します。

ファイルの検索

各ファイルページの検索ツールを使用して、アプリケーションツリー上のファイルを検索することができます。

手順 ファイルを検索するには

検索ツールでは、検索条件を入力するためのテキストボックスが表示されます。

1. [アプリケーション] ページに移動します。
2. ナビゲーションウィンドウまたはファイルパネルの [検索] テキストボックスに、検索文字列を入力します。
3. [検索] ボタンをクリックします。

リストにフィルタが適用され、検索文字列に一致したオブジェクトが表示されます。検索文字列はハイライト表示されます。

アプリケーションツリーのソート

ソートを行うと、項目がアプリケーションディレクトリに表示される順序を変更することができます。

手順 アプリケーションディレクトリをソートするには

1. [アプリケーション] ページに移動します。
2. ファイルパネルでアプリケーションディレクトリを開き、フィールドタイトルをクリックします。

このフィールドの値の降順でページがソートされ、下向き矢印が表示されます。再度クリックすると、昇順でソートされます。矢印が上向きに変わります。再度クリックすると、元の順序に戻ります。矢印は非表示になります。

アプリケーションファイルの選択

アプリケーションフォルダ間でのファイルのコピーおよび貼り付け、アプリケーションフォルダからのファイルの削除、アプリケーションフォルダ内のシノニムの更新を行うことができます。また、アプリケーションフォルダ間でのサブフォルダのコピーおよび移動、アプリケーションフォルダからのサブフォルダの削除も行えます。サブフォルダをコピー、移動、削除した場合、そのサブフォルダ内のすべてのファイルやサブフォルダも同時にコピー、移動、削除されます。

手順 アプリケーションファイルをコピー、切り取り、貼り付け、削除するには

1. [アプリケーション] ページに移動します。

2. アプリケーションフォルダをクリックし、ファイルパネルに開きます。
3. ファイルパネルで、ファイルを選択します。複数のファイルを選択することもできます。
連続したファイルを選択するには、1 つ目のファイルをクリックし、Shift キーを押しながら最後のファイルをクリックします。

不連続のファイルを選択するには、Ctrl キーを押しながら選択する各ファイルをクリックします。
4. 選択したファイルを右クリックし、[コピー]、[切り取り]、[削除] のいずれかをクリックします。
5. コピーされたファイルまたは切り取られたファイルを別のアプリケーションフォルダに貼り付けるには、このアプリケーションフォルダを右クリックして [貼り付け] をクリックします。

選択したアプリケーションフォルダにファイルが貼り付けられます。

手順 アプリケーションフォルダ内のシノニムを更新するには

1. [アプリケーション] ページに移動します。
2. アプリケーションフォルダ内のシノニムを右クリックし、[メタデータ管理]、[リフレッシュ] を順に選択します。

[シノニムの更新] ページが開きます。更新する属性を選択することができます。キー、インデックス、フォーマットなど一部の属性は、デフォルト設定で選択されています。

3. 選択を完了後、[次へ] をクリックします。

ステータスページが開き、更新リクエストの結果が表示されます。

アプリケーションコマンドの概要

ここでは、アプリケーションを制御する目的で使用できる、プラットフォームに依存しないアプリケーション (APP) コマンドについて説明します。

APP コマンド COPYFILE、MOVEFILE、DELETEFILE、RENAMEFILE のファイル名およびファイルタイプの参照に、次のワイルドカード文字を使用することができます。

- ❑ アスタリスク (*) - 任意の長さの任意の組み合わせの文字 (0 文字を含む) を表します。

アスタリスク (*) は、filename または filetype パラメータ全体を表す文字として使用することもできます。

- ❑ 疑問符 (?) - 任意の 1 文字 (または 0 文字) を表します。

参照

APP コマンドクイックリファレンス

下表に示されたコマンドをクリックすると、所定の構文およびその他の詳細情報が表示されます。

検索パス管理コマンド

コマンド	説明
84 ページの「 APP PATH 」	アプリケーション検索パスを設定またはリセットします。
85 ページの「 APP PREPENDPATH 」	既存の APP PATH 検索パスの先頭にアプリケーション名を一時的に追加します。
85 ページの「 APP APPENDPATH 」	既存の APP PATH 検索パスの末尾にアプリケーション名を一時的に追加します。
86 ページの「 APP MAP 」	approot 構造の外部にある物理パスを指す仮想アプリケーションを定義するか、アプリケーションを別のアプリケーションにリダイレクトします。このコマンドは、仮想アプリケーションまたはリダイレクトされたアプリケーションを検索パスに追加できる状態にします。仮想アプリケーションまたはリダイレクトされたアプリケーションを APP PATH に自動的に追加するものではありません。
87 ページの「 APP SET METALLOCATION_SAME 」	マスターファイルおよびそれに対応するアクセスファイルが同一のアプリケーションディレクトリに存在する必要があるかどうかを指定します。
88 ページの「 APP ? METALLOCATION_SAME 」	APP SET METALLOCATION_SAME の値を取得します。
88 ページの「 APP SHOWPATH 」	検索パスで現在アクティブなすべてのアプリケーションをリスト表示します。

アプリケーション管理コマンド

コマンド	説明
90 ページの「 APP COPY 」	1 つのアプリケーションの内容を別のアプリケーションにコピーします。
89 ページの「 APP CREATE 」	approot の下にアプリケーションを作成します。
94 ページの「 APP DELETE 」	アプリケーションを削除します。
92 ページの「 APP MOVE 」	アプリケーションのコンテンツを別のアプリケーションに移動します。
95 ページの「 APP PROPERTY CODEPAGE 」	アプリケーション内のファイルのコードページを指定します。
96 ページの「 APP RENAME 」	アプリケーション名を変更します。

ファイル管理コマンド

コマンド	説明
90 ページの「 APP COPYF[ILE] 」	* アプリケーションの単一コンポーネントまたはコンポーネントタイプを別のアプリケーションにコピーします。
92 ページの「 APP MOVEF[ILE] 」	* アプリケーションの単一コンポーネントまたはコンポーネントタイプを別のアプリケーションに移動します。
96 ページの「 APP RENAMF[ILE] 」	* アプリケーションの単一コンポーネントまたはコンポーネントタイプの名前を変更します。
94 ページの「 APP DELETF[ILE] 」	* アプリケーションから単一コンポーネントまたはコンポーネントタイプを削除します。

* このマニュアルの残りの部分では、APP コマンドの短縮形を使用します。

出力ダイレクトコマンド

コマンド	説明
103 ページの「 APP HOLD 」	FILEDEF コマンドを使用してデータファイルを割り当てた場合を除き、HOLD、SAVE、SAVB、CREATE SYNONYM、APP QUERY HOLD 処理で出力ファイルをアプリケーション内に作成するかどうかを制御します。
104 ページの「 APP HOLDDATA 」	HOLD、SAVE、または SAVB コマンドで作成された一時データファイルの格納先となるアプリケーションを指定します。
104 ページの「 APP HOLDMETA 」	HOLD コマンドで作成されたマスターファイルおよびアクセスファイルの格納先となるアプリケーションを指定します。
105 ページの「 APP FI[LEDEF] 」	このコマンドは廃止され、FILEDEF という短縮名に変更されています。

ヘルプコマンド

コマンド	説明
114 ページの「 APP HELP 」	APP コマンドのリストと各コマンドの簡単な説明を表示します。

参照

アプリケーションメタデータコマンドおよびメタデータテーブル

下表に示されたコマンドをクリックすると、所定の構文およびその他の詳細情報が表示されます。

コマンド	説明
106 ページの「 STATE 」 <i>app/file.extension</i>	ファイルの有無を確認します。

コマンド	説明
107 ページの「 APP LIST 」 <code>app [HOLD]</code>	<code>approot</code> の下に存在するアプリケーションをリスト表示します。 <code>HOLD</code> オプションを使用した場合は、<code>approot</code> の下に存在するアプリケーションをリスト表示するとともに、出力を「<code>focappl.ftm</code>」という一時ファイルに書き込みます。この一時ファイルは、その後のレポートリクエストに使用することができます。
109 ページの「 APP QUERY 」 <code>app [HOLD]</code>	アプリケーション内のすべてのファイルをリスト表示します。 <code>HOLD</code> オプションを使用した場合は、アプリケーション内のすべてのファイルをリスト表示するとともに、出力を「<code>focappq.ftm</code>」という一時ファイルに書き込みます。この一時ファイルは、その後のレポートリクエストに使用することができます。
112 ページの「 catalog/sysfiles 」	このテーブルには、パス上でアクセス可能な特定タイプ (デフォルトは <code>MASTER</code>) のアプリケーション名オブジェクトのリストが格納されます。
<code>catalog/sysdirs</code>	このテーブルには、物理ディレクトリ下の物理ファイルの再帰的リストが格納されます。
111 ページの「 catalog/sysapps 」	このテーブルには、パス上の物理オブジェクトのメタデータが格納されます。
<code>catalog/systables</code>	このテーブルには、パス上のテーブル (および関連するメタデータ) のアプリケーション名が格納されます。

検索パス管理コマンド

サーバには、アプリケーションおよびシステムコンポーネントで使用するデフォルト検索パスが設定されています。次の `APP` コマンドを使用し、この検索パスを補完することができます。

❑ APP PATH

- ❑ APP PREPENDPATH
- ❑ APP APPENDPATH
- ❑ APP SET METALLOCATION_SAME
- ❑ APP ? METALLOCATION_SAME

通常、これらのコマンドは、デフォルトの検索パスの先頭にアプリケーションを追加します。例外は、現在のセッション中に作成された一時コンポーネントです。これらの一時コンポーネントは、最初に検索が実行され、ユーザが定義したパスより前に検索されます。

APP PATH コマンドを手動で発行したり、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からアプリケーション検索パスを設定したりすることができます。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からアプリケーションパスを構成した場合、APP PATH コマンドは選択可能なプロファイル (グローバル、グループ、ユーザ) に格納されます。デフォルトプロファイルは、グローバルプロファイルの edasprof です。

APP PATH

APP PATH コマンドは、approot 下のアプリケーション名を指定して、検索パスに追加するアプリケーションを設定します。複数のアプリケーション名を指定し、検索パスを拡張することもできます。

構文 検索パスへのアプリケーションの手動追加

```
APP PATH app1[/] [app2[/] ...]  
[appn[/]]
```

説明

```
app1 ... appn
```

アプリケーションフォルダ名です。アプリケーションフォルダ名の後ろにスラッシュ (/) を追加した場合、ネストされたアプリケーションフォルダ (指定したアプリケーションフォルダ下のアプリケーションサブツリー) は検索パスに含まれません。アプリケーションフォルダ名の後ろにスラッシュ (/) を追加しない場合、プロシジャで参照されているファイルの検索対象にネストされたアプリケーションを含めるかどうか、および検索対象のネストレベルは、edaserve.cfg ファイルの nested_app パラメータ値に基づいて決定されます。複数のアプリケーションフォルダ名を指定する際に、すべての名前が 1 行に収まらない場合は、1 行目の末尾フォルダに継続文字 (-) を追加した上で、次の行に残りのアプリケーションフォルダ名を記述します。

注意

- ❑ アプリケーションフォルダ名を 1 つも指定せずに APP PATH コマンドを使用すると、検索パスをリセットして初期のリストに戻すことができます。
- ❑ APP PATH コマンドでは、アプリケーションリストは検証されません。

APP PREPENDPATH

APP PREPENDPATH コマンドを使用すると、既存の APP PATH 検索パスの先頭にアプリケーションフォルダ名を一時的に追加することができます。

このコマンドを使用して検索パスを変更するには、アプリケーション内に手動でコーディングする必要があります。

構文 検索パスの先頭へのアプリケーション名の追加

```
APP PREPENDPATH app1[/] [app2[/]] ...
[appn[/]]
```

説明

```
app1 ... appn
```

アプリケーションフォルダ名です。アプリケーションフォルダ名の後ろにスラッシュ (/) を追加した場合、ネストされたアプリケーションフォルダ (指定したアプリケーション下のアプリケーションサブツリー) は検索パスに含まれません。アプリケーションフォルダ名の後ろにスラッシュ (/) を追加しない場合、プロシジャで参照されているファイルの検索対象にネストされたアプリケーションフォルダを含めるかどうか、および検索対象のネストレベルは、`nested_app` パラメータ値に基づいて決定されます。複数のアプリケーションフォルダ名を指定する際に、すべての名前が 1 行に収まらない場合は、1 行目の末尾に継続文字 (-) を追加した上で、次の行に残りのアプリケーションフォルダ名を記述します。

APP APPENDPATH

APP APPENDPATH コマンドを使用すると、既存の APP PATH 検索パスの末尾にアプリケーションフォルダ名を一時的に追加することができます。

このコマンドを使用して検索パスを変更するには、アプリケーション内に手動でコーディングする必要があります。

構文 検索パスの末尾へのアプリケーション名の追加

```
APP APPENDPATH app1[/] [app2[/]] ... [appn[/]]
```

説明

`app1 ... appn`

アプリケーションフォルダ名です。アプリケーションフォルダ名の後ろにスラッシュ (/) を追加した場合、ネストされたアプリケーションフォルダ (指定したアプリケーション下のアプリケーションサブツリー) は検索パスに含まれません。アプリケーションフォルダ名の後ろにスラッシュ (/) を追加しない場合、プロシジャで参照されているファイルの検索対象にネストされたアプリケーションフォルダを含めるかどうか、および検索対象のネストレベルは、`nested_app` パラメータ値に基づいて決定されます。複数のアプリケーションフォルダ名を指定する際に、すべての名前が 1 行に収まらない場合は、1 行目の末尾に継続文字 (-) を追加した上で、次の行に残りのアプリケーションフォルダ名を記述します。

APP MAP

APP MAP コマンドを使用すると、ファイルシステムの任意の場所にある `approot` 以外のアプリケーションフォルダにアプリケーション名を割り当てたり、アプリケーションをリダイレクトしたりできます。このアプリケーションフォルダ名は `approot` 配下の仮想アプリケーションディレクトリとなり、APP PATH コマンド、およびアプリケーションフォルダ名をパラメータとして使用する任意の APP コマンドで参照することができるようになります。

なお、マッピングは、検索パスにディレクトリを追加できる状態にしますが、検索パスにディレクトリを自動的に追加するものではありません。

構文 物理ファイルパスのマッピングまたはアプリケーションのリダイレクト

APPROOT 外部の物理ファイルパスをマッピングするには、次の構文を使用します。

```
APP MAP virtualname real_location
```

説明

`virtualname`

最大で 64 バイトのアプリケーションフォルダ名です。この名前は、後から APP PATH コマンドで 사용할 ことができます。

`real_location`

オペレーティングシステムのネイティブ形式で指定する実際のフルパス名または `ddname` です。UNIX、Linux では、パス名に大文字と小文字を混在させることができますが、MAP `virtualname` 自体では常にハンドルは区別されません (例、EX `mymap/mytest`)。

なお、実際のパス名にブランクが含まれている場合は、その名前を二重引用符で囲む必要があります。

APPROOT または APPROOT 外部のアプリケーションを別の名前にリダイレクトするには、次の構文を使用します。

```
APP MAP app1 app2/dir1/dir2/../../dirn
```

説明

app1

既存のマッピングまたはリンクされた物理アプリケーションを指定できます。新しいアプリケーションを指定することもできます。

app2

マッピング、リンクされた物理アプリケーション、または新しい物理アプリケーションを指定できます。

dir1 ... dirn

アプリケーションディレクトリ名です。

注意： 内部アプリケーション (例、nyhome、homeapps、_edaconf、_edahome) の再マッピングはサポートされません。

例

APP MAP コマンドの例

Windows での基本的な例

```
APP MAP test c:¥temptest¥
```

パス名にブランクが含まれる場合は、その名前を二重引用符で囲む必要があります。以下はその例です。

```
APP MAP test "c:¥temp test¥"
```

APPROOT 下のアプリケーションの再マッピングの例

```
APP MAP ibisamp retail
```

APP SET METALLOCATION_SAME

APP SET METALLOCATION_SAME コマンドは、マスターファイルとそれに対応するアクセスファイルの格納先を同一にする必要があるかどうかを指定します。

構文

シノニムファイルの格納先の制御

```
APP SET METALLOCATION_SAME {ON|OFF}
```

説明

ON

マスターファイルとそれに対応するアクセスファイルが同一のアプリケーションディレクトリに存在する必要があることを指定します。デフォルト値は ON です。

OFF

リクエストのマスターファイルの格納先が特定されると、サーバはアクティブ検索パスを使用して、対応するアクセスファイルを検索します。

APP ? METALLOCATION_SAME

APP ? METALLOCATION_SAME コマンドは、マスターファイルとそれに対応するアクセスファイルが同一のアプリケーションディレクトリに存在する必要があるかどうかを特定します。

構文 シノニムファイルの格納先を同一にする必要があるかどうかの特定

APP ? METALLOCATION_SAME

このクエリコマンドの結果が ON の場合、サーバは、マスターファイルとそれに対応するアクセスファイルが同一のアプリケーションディレクトリ内に存在することを前提にファイルを検索します。結果が OFF の場合、サーバは、アクティブ検索パスを使用して、特定のマスターファイルに対応するアクセスファイルを検索します。

APP SHOWPATH

APP SHOWPATH コマンドは、検索パスで現在アクティブになっているすべてのアプリケーションをリスト表示します。baseapp は、常にリストの最後に表示されます。このリストは、ナビゲーションウィンドウのアプリケーションツリーに表示されるアプリケーションリストを反映しています。

構文 アクティブアプリケーションのリスト表示

APP SHOWPATH

例 検索パスのアクティブアプリケーションのリスト表示

通常、サーバをインストールすると、「ibisamp」および「baseapp」という 2 つのデフォルトアプリケーションがインストールされます。ibisamp にはサンプルファイル、baseapp にはユーザが作成するファイルがそれぞれ格納されます。

APP SHOWPATH コマンドは、次の出力を生成します。

ibisamp
baseapp

アプリケーションおよびファイルの管理コマンド

ここでは、アプリケーションおよびそのコンポーネントファイルをさまざまな方法で管理する APP コマンドについて説明します。

APP CREATE

通常、APP CREATE コマンドは、approot ディレクトリ下にアプリケーションを作成します。例外として、z/OS 版 Server での PDS 展開があります。この場合、アプリケーションが 1 つの物理エンティティとなり、コンポーネントファイルのタイプがそれぞれ別の PDS に格納されます。

APP CREATE コマンドは、1 回のコマンド実行で必要な数のアプリケーションを任意に作成することができます。

構文 アプリケーションの手動作成

```
APP CREATE app1[/app1a...] [app2[/app2a...] ...  
[appn[/appna...] ] [DROP]
```

説明

app1 ... appn

approot 下のアプリケーションフォルダ名です。アプリケーションフォルダ名には、最大で 64 バイトの文字を指定することができます。

app1a...appna

ネストされたアプリケーションディレクトリを指定します。ネストされたアプリケーションフォルダが構成されている場合に使用できます。ネストされたアプリケーションフォルダを作成するには、親アプリケーションが存在する必要があります。

DROP

作成するアプリケーションフォルダと同一名の既存アプリケーションフォルダが存在する場合、既存アプリケーションフォルダを削除した上で、新しいアプリケーションフォルダをその名前で作成します。既存アプリケーションフォルダ内に存在するファイルもすべて削除されます。DROP オプションを使用しない場合は、メッセージが表示されます。この場合、既存アプリケーションフォルダが削除されることも、変更されることもあります。

アプリケーションフォルダ名にブランクを使用することはできません。アプリケーションフォルダ名にブランクが含まれていると、ブランクで区切られた部分はそれぞれ別のディレクトリであると認識されます。名前にブランクを含める必要がある場合は、Windows のエクスプローラなど、他の方法を使用してアプリケーションフォルダ名を作成する必要があります。その後、そのアプリケーションフォルダ名は、APP MAP コマンドを使用して APPROOT に追加にすることができます。

複数のアプリケーションフォルダ名を指定する際に、すべての名前が 1 行に収まらない場合は、1 行目の末尾に継続文字 (-) を追加した上で、次の行に残りのアプリケーションフォルダ名を記述します。

アプリケーションフォルダ名に「HOLD」という語句を使用することはできません。

APP COPY

APP COPY コマンドは、指定したアプリケーションフォルダの内容全体を別のアプリケーションフォルダにコピーします。ターゲットアプリケーションがすでに存在していることが前提となります。

構文 アプリケーションフォルダのコピー

```
APP COPY app1[/app1a...] app2[/app2a...]
```

説明

`app1[/app1a...]`

コピー元のアプリケーションフォルダです。ネストされたアプリケーションフォルダ名を指定することもできます。

`app2[/app2a...]`

最初のアプリケーションフォルダ (コピー元) の内容をコピーするアプリケーションフォルダ (コピー先) です。ネストされたアプリケーションフォルダ名を指定することもできます。

APP COPYF[ILE]

APP COPYF[ILE] コマンドは、1 つ以上のコンポーネントまたはコンポーネントタイプを特定のアプリケーションフォルダから別のアプリケーションフォルダにコピーします。

なお、コンポーネントを手動でコピーする場合は、その操作と同時に名前を変更することもできます。

マスターファイルをコピーすると、それに対応するアクセスファイルもコピーされます。ただし、アクセスファイル (ファイルタイプ FOCSQL) をコピーしても、それに対応するマスターファイルは自動的にコピーされません。

構文 アプリケーションコンポーネントの手動コピー

```
APP COPYF[ILE] app1[/app1a...]
      {filename1|*} filetype1 app2 [/app2a...]
      {filename2|*} {filetype2|*} [IFEXIST] DROP
```

説明

app1[/app1a...]

コピー元のコンポーネントが格納されたアプリケーションフォルダ名です。ネストされたアプリケーションフォルダ名を指定することもできます。

filename1

コピー元のコンポーネントです。アスタリスク (*) を使用すると、filetype で指定されたタイプのファイルがすべてコピーされます。

ファイル名およびファイルタイプの参照には、次のワイルドカード文字を使用することができます。

- ❑ アスタリスク (*) - 任意の長さの任意の組み合わせの文字 (0 文字を含む) を表します。
アスタリスク (*) は、filename または filetype パラメータ全体を表す文字として使用することもできます。
- ❑ 疑問符 (?) - 任意の 1 文字 (または 0 文字) を表します。

filetype1

コピーするコンポーネントのファイルタイプを大文字で指定します。

app2[/app2a...]

指定したコンポーネントのコピー先アプリケーションフォルダです。ネストされたアプリケーションフォルダ名を指定することもできます。

filename2

コピー後のターゲットアプリケーションフォルダ内でのコンポーネント名です。アスタリスク (*) を使用すると、ソースアプリケーションで使用されているファイル名がターゲットアプリケーションに継承されます。

filetype2

コピー後のターゲットアプリケーションフォルダ内でのコンポーネントタイプを大文字で指定します。アスタリスク (*) を使用すると、ソースアプリケーションで使用されているファイルタイプがターゲットアプリケーションに継承されます。

IFEXIST

ソースアプリケーション内のコンポーネントがターゲットコンポーネントに存在しない場合、そのコンポーネントは無視されます。

DROP

コピーするコンポーネントと同一の名前およびファイルタイプのコンポーネントがターゲットアプリケーションフォルダに存在する場合は、そのコンポーネントを上書きします。

APP コマンドでコピー可能なファイルタイプの一覧については、97 ページの「[APP コマンドによるファイルタイプの指定](#)」を参照してください。

APP MOVE

APP MOVE コマンドは、アプリケーションフォルダの内容全体を別のアプリケーションフォルダに移動します。ターゲットアプリケーションがすでに存在していることが前提となります。

構文 アプリケーションの移動

APP MOVE [app1\[/app1a...\]](#) [app2\[/app2a...\]](#)

説明

[app1\[/app1a...\]](#)

移動元のアプリケーションフォルダです。ネストされたアプリケーションフォルダ名を指定することもできます。

[app2\[/app2a...\]](#)

アプリケーションフォルダの内容の移動先となるアプリケーションフォルダです。ネストされたアプリケーションフォルダ名を指定することもできます。

APP MOVEF[ILE]

APP MOVEF[ILE] コマンドは、1 つ以上のコンポーネントまたはコンポーネントタイプを特定のアプリケーションフォルダから別のアプリケーションフォルダに移動します。

コンポーネントを手動で移動する場合は、その操作と同時に名前を変更することもできます。

マスターファイルを移動すると、それに対応するアクセスファイルも移動されます。ただし、アクセスファイル (ファイルタイプ FOCSQL) を移動しても、それに対応するマスターファイルは自動的に移動されません。

構文

アプリケーションコンポーネントの手動移動

```
APP MOVEF[ILE] app1[/app1a...]
    {filename1|*} filetype1 app2 [/app2a...]
    {filename2|*} {filetype2|*} [IFEXIST] [DROP]
```

説明

app1[/app1a...]

移動するコンポーネントが格納されているアプリケーションフォルダです。ネストされたアプリケーションフォルダ名を指定することもできます。

filename1

移動するコンポーネントの名前です。アスタリスク (*) を使用すると、filetype で指定されたタイプのファイルがすべて移動されます。

ファイル名およびファイルタイプの参照には、次のワイルドカード文字を使用することができます。

- ❑ アスタリスク (*) - 任意の長さの任意の組み合わせの文字 (0 文字を含む) を表します。

アスタリスク (*) は、filename または filetype パラメータ全体を表す文字として使用することもできます。

- ❑ 疑問符 (?) - 任意の 1 文字 (または 0 文字) を表します。

filetype1

移動するコンポーネントのファイルタイプを大文字で指定します。

app2[/app2a...]

指定したコンポーネントの移動先アプリケーションフォルダです。ネストされたアプリケーションフォルダ名を指定することもできます。

filename2

移動後のターゲットアプリケーションフォルダ内でのコンポーネント名です。アスタリスク (*) を使用すると、ソースアプリケーションで使用されているファイル名がターゲットアプリケーションに継承されます。

filetype2

移動後のターゲットアプリケーションフォルダ内でのコンポーネントタイプを大文字で指定します。アスタリスク (*) を使用すると、ソースアプリケーションで使用されているファイルタイプがターゲットアプリケーションに継承されます。

IFEXIST

ソースアプリケーション内のコンポーネントがターゲットコンポーネントに存在しない場合、そのコンポーネントは無視されます。

DROP

移動するコンポーネントと同一の名前およびファイルタイプのコンポーネントがターゲットアプリケーションに存在する場合は、そのコンポーネントを上書きします。

APP コマンドで移動可能なファイルタイプの一覧については、97 ページの「[APP コマンドによるファイルタイプの指定](#)」を参照してください。

APP DELETE

APP DELETE コマンドは、approot の下に存在するアプリケーションフォルダを削除します。

構文 アプリケーションの手動削除

```
APP DELETE app1[/app1a...] [app2[/app2a...] ...  
[appn[/appna...]]
```

説明

```
app1[/app1a...] ... [appn[/appna...]]
```

アプリケーションフォルダ名です。ネストされたアプリケーションフォルダ名を指定することもできます。複数のアプリケーションフォルダ名を指定する際に、すべての名前が 1 行に収まらない場合は、1 行目の末尾に継続文字 (-) を追加した上で、次の行に残りのアプリケーションフォルダ名を記述します。

APP DELETED[ILE]

APP DELETED[ILE] コマンドは、1 つ以上のコンポーネントまたはコンポーネントタイプをアプリケーションフォルダから削除します。

マスターファイルを削除すると、それに対応するアクセスファイルも削除されます。ただし、アクセスファイル (ファイルタイプ FOCSQL) を削除しても、それに対応するマスターファイルは自動的に削除されません。

構文 アプリケーションコンポーネントの手動削除

```
APP DELETEF[ILE] app[/appna...] {filename|*} filetype
```

説明

`appn[/appa...]`

削除するコンポーネントまたはコンポーネントタイプの削除元のアプリケーションフォルダです。ネストされたアプリケーションフォルダ名を指定することもできます。

`filename`

削除するコンポーネントの名前です。アスタリスク (*) を使用すると、`filetype` で指定されたタイプのファイルがすべて削除されます。

ファイル名およびファイルタイプの参照には、次のワイルドカード文字を使用することができます。

- ❑ アスタリスク (*) - 任意の長さの任意の組み合わせの文字 (0 文字を含む) を表します。
アスタリスク (*) は、`filename` または `filetype` パラメータ全体を表す文字として使用することもできます。
- ❑ 疑問符 (?) - 任意の 1 文字 (または 0 文字) を表します。

`filetype`

削除するコンポーネントのファイルタイプを大文字で指定します。

APP コマンドで使用可能なファイルタイプの一覧については、97 ページの「[APP コマンドによるファイルタイプの指定](#)」を参照してください。

APP PROPERTY CODEPAGE

APP PROPERTY appname CODEPAGE コマンドは、アプリケーションディレクトリ内のデータファイル以外のファイルに使用するコードページを指定します。

構文 アプリケーションのコードページ指定

```
APP PROPERTY app[/appa...] CODEPAGE number
```

説明

`app[/appa...]`

アプリケーションフォルダ名です。ネストされたアプリケーションフォルダ名を指定することもできます。

`number`

アプリケーションフォルダ内のデータファイル以外のファイルに使用するコードページ番号です。

APP RENAME

APP RENAME コマンドは、既存のアプリケーションフォルダ名を変更します。

注意: 検索パスで現在アクティブになっているアプリケーションフォルダの名前を変更することはできません。

構文 アプリケーションフォルダ名の変更

```
APP RENAME app1[/app1a...] app2[/app2a...]
```

説明

`app1[/app1a...]`

名前を変更するアプリケーションフォルダ名です。ネストされたアプリケーションフォルダ名を指定することもできます。

`app2[/app2a...]`

アプリケーションフォルダの新しい名前です。最大 64 バイトです。ネストされたアプリケーションフォルダ名を指定することもできます。

例 アプリケーションフォルダ名の変更

次の例は、「app1」を「app2」という名前に変更する方法を示しています。

```
APP RENAME app1 app2
```

APP RENAMF[ILE]

APP RENAMF[ILE] コマンドは、アプリケーションフォルダ内の 1 つ以上のコンポーネントの名前を変更します。

マスターファイルの名前を変更すると、それに対応するアクセスファイルの名前も変更されます。ただし、アクセスファイル (ファイルタイプ FOCSQL) の名前を変更しても、それに対応するマスターファイルの名前は自動的に変更されません。

構文 アプリケーションコンポーネント名の変更

```
APP RENAMEF[ILE] app[/appa... ] filename1
filename2 filetype [DROP]
```

説明

app[/appa...]

名前を変更するコンポーネントが格納されたアプリケーションフォルダ名です。ネストされたアプリケーションフォルダ名を指定することもできます。

filename1

名前を変更するコンポーネントの名前です。

ファイル名およびファイルタイプの参照には、次のワイルドカード文字を使用することができます。

- ❑ アスタリスク (*) - 任意の長さの任意の組み合わせの文字 (0 文字を含む) を表します。
アスタリスク (*) は、filename または filetype パラメータ全体を表す文字として使用することもできます。
- ❑ 疑問符 (?) - 任意の 1 文字 (または 0 文字) を表します。

filename2

コンポーネントの新しい名前です。最大で 64 バイトのコンポーネント名を指定することができます。

filetype

名前を変更するコンポーネントのファイルタイプを大文字で指定します。

DROP

既存のコンポーネントを同一のファイル名とファイルタイプで上書きします。

APP コマンドで使用可能なファイルタイプの一覧については、97 ページの「[APP コマンドによるファイルタイプの指定](#)」を参照してください。

APP コマンドによるファイルタイプの指定

APP COPYF、APP MOVEF、APP DELETEF、APP RENAMEF の各コマンドは、さまざまなファイルタイプに対して実行できるコマンドです。

下表は、APP コマンドを適用できるファイルタイプと、階層ファイルシステムの中で、オンディスクの名前 (ディレクトリ名と拡張子を含まない、ファイル名本体) に対して関連付けられるファイル拡張子を示しています。

APP コマンドがファイルタイプを要求する場合は、ファイルタイプを大文字で指定する必要があります。

注意：この一覧は、すべての製品およびバージョンでサポートされるファイルタイプを示しています。ファイルタイプによっては、特定のバージョンや特定の製品でサポートされない場合があります。

ファイルタイプ	ファイル拡張子
ACX	.acx
ADR	.adr
AFM	.afm
BMP	.bmp
BST	.bst
CSS	.css
CONTROL	.ctl
DATA	.dat
DDS	.dds
DEFAULT	APP コマンドでは、物理ファイル拡張子を特定するために APP filename 値が使用されます。これにより、ユーザ定義の不明な拡張子が APP コマンドでサポートされる場合があります (例、APP COPYFILE BASEAPP MYFILE.FOO DEFAULT BASEAPP MYFILE FOCEXEC)。
DTD	.dtd
EDANLS	.nls
EDAPRFU	.prf
EDAPROF	.prf

ファイルタイプ	ファイル拡張子
EDAPSB	.psb
EPS	.eps
ERRORS	.err
ETG	.etg
ETL	.etl
EXCEL	.xls
FMU	.fmu
FOCCOMP	.fcm
FOCDEF	.def
FOCEXEC または FEX	.fex
FOCFTMAP	.fmp
FOCPSB	.psb
FOCSQL	.acx
FOCSTYLE	.sty
FOCTEMP	.ftm
FOCUS	.foc
GIF	.gif
HLI	.hli
HTML	.htm
IBICPG	.sl
JPG	.jpg
JS	.js

ファイルタイプ	ファイル拡張子
LSN	.lsn
MAINTAIN	.mnt
MASTER または MAS	.mas MASTER には特別な動作があり、マスターファイルに対応するアクセスファイル(.acx)が存在する場合は、APP コマンドの実行時にアクセスファイルも処理されます。同様に、マスターファイルに対応するメタデータも処理されます。MAS は、アクセスファイルは処理せずに、マスターファイルのみを処理する必要がある場合に使用します。
MHT	.mht
Microsoft Access データベース	.mdb
MNTPAINT	.mpt
OMI	.omi
PDF	.pdf
PFA	.pfa
PFB	.pfb
PNG	.png
SMARTLIB	.knb
SQL	.sql
SVG	.svg
TABS	.txt
TDL	.tdl

ファイルタイプ	ファイル拡張子
TRF	.trf
TTEDIT	.tte
TXT	.txt
WINFORMS	.wfm
WSDL	.wsd
XHT	.xht
XLSM	.xlsm
XLSX	.xlsx
XLTM	.xltm
XLTX	.xltx
XML	.xml
XSD	.xsd
XSL	.xsl

出力リダイレクトコマンド

3 つの APP コマンド (APP HOLD、APP HOLDDATA、APP HOLDMETA) は、FILEDEF コマンドとともに、出力先を制御するコマンドクラスから構成されます。任意の出力先に結果を出力するには、これらのコマンドの相互作用を理解しておく必要があります。

注意：APP HOLD、APP HOLDDATA、APP HOLDMETA の各コマンドでそれぞれの動作が共通している場合は、これらのコマンドを総称して「APP HOLD*」と記載しています。FILEDEF は APP コマンドグループではありませんが、これらのファイル割り当てコマンドは APP HOLD* コマンドと相互作用 (情報交換) します。そのため、これらのコマンドについても説明します。APP FI[LEDEF] コマンドは廃止され、FILEDEF という短縮名に変更されています。

これらのコマンドの中で最も基本的なコマンドは APP HOLD です。このコマンドを使用すると、すべての出力を特定のアプリケーションに切り替えることができます。このコマンドは、CREATE SYNONYM や APP QUERY HOLD とともに使用する場合と同様に、HOLD、SAVE、SAVBなどの出力ファイルを生成する処理に使用します。

APP HOLD* コマンドは、他のアプリケーションでも使用する永久ファイルを作成する場合に特に役立ちます。ただし、アプリケーション内の不適切な位置でコマンドを発行したり、アプリケーション内で別のステップを実行する際にそのコマンドを引き続き有効にしておくと、ターゲットアプリケーションが中間ファイルや不用なファイルで一杯になる可能性があります。各コマンドの動作とコマンド間の相互作用を理解しておくことにより、このような状況を回避することができます。

参照 出力リダイレクトコマンドの相互作用

下表は、各リダイレクトコマンドに関連付けられた動作と、複数のコマンドを使用した場合の相互作用を示しています。

コマンド	スタンドアロン	注意事項
APP HOLD	HOLD、SAVE、SAVB、CREATE SYNONYM、APP QUERY HOLD コマンドで作成されたエージェント出力はすべて、指定したアプリケーションにリダイレクトされます。	APP HOLD コマンドをオフにするには、特定の appname を指定せずに APP HOLD を発行します。
APP HOLDDATA	指定したアプリケーションに HOLD、SAVE、SAVB 処理からのデータをリダイレクトします。ただし、関連するメタデータはリダイレクトしません (この表の後の注意 1 を参照)。	APP HOLD を上書き変更します。
APP HOLDMETA	指定したアプリケーションに HOLD、SAVE、SAVB 処理からのメタデータをリダイレクトします。ただし、関連するデータはリダイレクトしません (この表の後の注意 1 を参照)。	APP HOLD を上書き変更します。

コマンド	スタンドアロン	注意事項
<code>FILEDEF ddname</code>	特定の HOLD、SAVE、SAVB 処理のデータ出力を指定したターゲットに切り替えます。ただし、関連するメタデータは切り替えません (この表の後の注意 1 を参照)。 AS 句が ddname と一致している必要があります。AS 句が存在しない場合は、HOLD 出力ファイル (HOLD)、SAVE 出力ファイル (SAVE)、SAVB 出力ファイル (SAVB) のように、ddname は事前に定義されたデフォルト名と一致する必要があります。	APP HOLD および APP HOLDDATA を上書き変更します。

注意

- ❑ メタデータが関連付けられていないフォーマットもあります。たとえば、HOLD FORMAT PDF コマンドはメタデータを作成しないため、切り替えるメタデータはありません。
- ❑ CREATE SYNONYM コマンドは、構文によりアプリケーションフォルダ名を直接サポートするため、APP HOLD コマンドを使用して CREATE SYNONYM の出力を切り替える必要はなく、また使用しないことをお勧めします。

`CREATE SYNONYM appname/synonym ...`

APP HOLD

APP HOLD コマンドは、アプリケーションの HOLD、SAVE、SAVB 処理で作成された出力データファイル (あるいは、関連するマスターファイルとアクセスファイル) の格納先となるアプリケーションフォルダを定義します。

APP HOLD は、アプリケーションのすべてのユーザに共通するファイルのリフレッシュに使用するコマンドです。このコマンドは、複数のユーザが使用するアプリケーション領域を参照するため、プライベートファイルには使用しないでください。同一の HOLD 名 (HOLD または AS name) を使用した場合、結果としてユーザ間の競合が発生する場合があります。

詳細は、102 ページの「[出力リダイレクトコマンドの相互作用](#)」を参照してください。

構文 一時ファイルの格納場所の指定

`APP HOLD appname[/appnamea...]`

説明

`appname[/appnamea...]`

出力ファイルを格納するアプリケーションフォルダです。ネストされたアプリケーションフォルダ名を指定することもできます。

注意： `appname` を指定せずに `APP HOLD` コマンドを実行した場合、コマンドは無効となります。

APP HOLDDATA

`APP HOLDDATA` コマンドは、`HOLD` コマンドで作成したデータファイルの格納場所となるアプリケーションフォルダを指定します。関連情報については、102 ページの「[出力リダイレクトコマンドの相互作用](#)」を参照してください。

構文 データファイルの格納場所指定

`APP HOLDDATA appname[/appnamea...]`

説明

`appname[/appnamea...]`

アプリケーションフォルダ内の書き込み処理で作成されたデータファイルの格納場所の名前です。ネストされたアプリケーションフォルダ名を指定することもできます。

APP HOLDMETA

`APP HOLDMETA` コマンドは、アプリケーションで生成されたマスターファイルとアクセスファイルを格納する場所となるアプリケーションフォルダを指定します。関連情報については、102 ページの「[出力リダイレクトコマンドの相互作用](#)」を参照してください。

構文 マスターファイルとアクセスファイルの格納場所指定

`APP HOLDMETA appname[/appnamea...]`

説明

`appname[/appnamea...]`

アプリケーションフォルダ内に作成されたマスターファイルとアクセスファイルの格納場所の名前です。ネストされたアプリケーションフォルダ名を指定することもできます。

APP FI[LEDEF]

APP FI[LEDEF] コマンドは廃止され、FILEDEF という短縮名に変更されています。

アプリケーションメタデータコマンドおよびカタログメタデータ

開発者は、アプリケーションメタデータを確認し、その結果に応じて次の手順を決定するアプリケーションを記述したい場合があります。たとえば、記述したアプリケーションでファイルの有無やファイルの日付を確認し、ファイルの再作成などの別の手順が必要かどうかを決定します。ファイルの有無やその他の属性は複数の方法で確認することができますが、これらの方法は製品の開発過程で進化しています。ただし、これらの方法の一部には制限事項があります。その一例として、UNIX のネイティブパス名を使用する STATE コマンドがあります。このタイプのパス名は、Windows のファイルパスと一致しない場合があります。そのため、オペレーティング環境に応じて適切なコマンドを発行するために IF THEN ELSE や GOTO ロジックが必要になり、これらのロジックの使用頻度によっては作業が煩雑になる場合があります。

この問題の一部を解決するために、STATE、FILEDEF、DYNAM などのコマンドが APP 名をサポートするよう拡張されています。たとえば、APP MAP を発行し、次に STATE mymap/myproc.fex を使用します。また、利用可能なアプリケーション (APP 名) のリストや特定のアプリケーション内のファイルのリストを取得するなど、より複雑な課題に対処するために、一連の APP コマンドが開発されています (APP LIST および APP QUERY)。一方、アプリケーションのネスト (サブディレクトリ) などの機能が実装されたため、アプリケーションメタデータにアクセスする方法をさらに拡張する必要性が発生しました。

広範囲の情報を取得する必要性から、さまざまな内部テーブルが拡張または作成されています。現在、標準の TABLE または SELECT 構文を使用してアプリケーションメタデータにアクセスする主な方法は、catalog/sysapps テーブルの使用です。この方法は、ほとんどの内部アプリケーションで使用されています。ただし、従来の方法も引き続きサポートされます。目的によっては従来の方法ですばやくコーディングできますが、前述のように制限事項があります。より複雑な状況では、新しい方法を使用して情報にアクセスする必要があります。また、catalog/systables や catalog/syscolumn などのテーブルからは、テーブル固有の追加情報が提供されます。たとえば、テーブルで使用されている DBMS や特定のフィールドのデータ定義などの情報があります。この新しい方法では、情報を取得するためのアクセス方法が重複する場合があります。たとえば、ファイルの有無を確認するために使用可能な catalog/sys* テーブルは多数あります。ただし、物理パス、アプリケーションディレクトリ、属性などの詳細情報はテーブルごとに異なります。

基本情報の取得

次のコマンドは、ファイルおよびアプリケーションフォルダに関する基本情報を返します。

STATE

STATE コマンドを使用して、ファイルの有無を確認することができます。ファイル参照は、オペレーティングシステムのネイティブパスでファイル名を指定することも、先頭に APP 名を使用してファイル名を指定することもできます。ここでは、ファイル名の先頭に APP 名を使用する方法についてのみ説明します。APP 名を使用する場合、その APP 名がサーバの APPROOT 下に作成されているかどうか、および APP MAP 名として作成されているかどうかは関係しません。

ファイルが存在しない場合、STATE コマンドがその結果を示すメッセージを表示します。STATE コマンドの発行後、ファイルが存在する場合は &RETCODE システム変数に 0 (ゼロ) が格納され、ファイルが存在しない場合は 0 (ゼロ) 以外の値が格納されます。

構文

ファイルの有無の確認

```
STATE appname/filename.filetype -TYPE RETCODE &RETCODE
```

説明

appname

ファイルが格納されているアプリケーションフォルダです。

filename

ファイルの名前です。

filetype

ファイルタイプまたはファイル拡張子です。

ファイルが存在する場合、&RETCODE 値は 0 (ゼロ) になります。ファイルが存在しない場合、0 (ゼロ) 以外の値になり、この値を使用してアプリケーションロジックの移動先を制御することができます。通常は -SET または -IF コマンドで使います。STATE コマンドは、「ファイルが見つかりません」メッセージも出力します。このメッセージを省略するには、SET TRMOUT={OFF|ON} コマンドを使います。

たとえば、次の STATE コマンドは、baseapp アプリケーションフォルダ内に myproc.fex ファイルが存在するかどうかを確認します。STATE コマンドは、ファイルが存在しない場合にメッセージを表示します。-TYPE コマンドは、ファイルが存在する場合に 0 (ゼロ)、ファイルが存在しない場合に -1 を表示します。

```
STATE baseapp/myproc.fex
-TYPE RETCODE &RETCODE
```

例

STATE コマンドによるファイルの有無の確認

次の例は、コードの一部を示しています。STATE コマンドから返されるメッセージを省略し、STATE コマンドを発行して baseapp アプリケーション内に myproc.fex ファイルが存在するかどうかを確認します。次にリターンコードを確認し、ファイルが存在しない場合はファイルを作成した上で、アプリケーションの次の手順に移動します。ファイルが存在する場合は、コードがアプリケーションの次の手順 (-RESUME ラベル) に直接移動します。

```
SET TRMOUT=OFF
STATE baseapp/myproc.fex
SET TRMOUT=ON
-IF &RETCODE EQ 0 THEN GOTO RESUME;
...
* Some code to create the file goes here
...
-RESUME
```

APP LIST

APP LIST コマンドは、アプリケーションルート下 (APPROOT)、または APP MAP コマンドでマッピングされたアプリケーションフォルダ下で利用可能なアプリケーションフォルダのリストを名前順で表示します。このコマンドは、実在する利用可能アプリケーションフォルダのリストを返すため、APP 名が現在のアプリケーションマッピングに存在するかどうかは関係ありません。

構文 APPROOT 内のアプリケーションのリスト表示

APP LIST [HOLD]

HOLD オプションを使用する場合、出力は focappl.ftm という一時ファイルに書き込まれます (PDS 展開では FOCAPPL)。次にこのファイルをリクエストに使用してレポートを作成したり、catalog/focappl マスターファイルを使用してアクションを実行したりできます。

制限事項

- ❑ APP LIST では、ネストされたアプリケーションの名前は表示されません。
- ❑ ファイル名の太文字と小文字が区別されるオペレーティングシステムでは (例、UNIX)、太文字の物理ディレクトリ名は有効ではありません (これらの名前は APP LIST でも返されません)。APP 名では太文字と小文字は区別されませんが、ディスク上では APP 名が小文字で作成されます。この場合、ネイティブオペレーティングシステムによって小文字が太文字に変換される場合があります (例、PDS 太文字ファイル名)。ただし、APP LIST は、各オペレーティングシステム間で整合性を図るために、APP 名を小文字で返します。

例 APP LIST によるアプリケーションのリスト表示と使用

次のリクエストは、アプリケーションフォルダのリストを表示します。

APP LIST

APP LIST の出力は次のとおりです。

```
BEGIN-APP-LIST
15/02/2000  13.36.38  baseapp
15/02/2000  13.36.38  ggdemo
15/02/2000  13.36.38  ncp
15/02/2000  13.36.38  template
END-APP-LIST
```

次のリクエストは、HOLD オプションを使用して格納されているアプリケーションフォルダのリストを表示します。

```
APP LIST HOLD
SQL SELECT DATE, TIME, APPNAME FROM FOCAPPL;
END
```

APP LIST の出力は次のとおりです。

DATE	TIME	APPNAME
----	----	-----
15/02/2000	13.36.38	baseapp
15/02/2000	13.36.38	ggdemo
15/02/2000	13.36.38	ncp
15/02/2000	13.36.38	template

次の例では、APP LIST HOLD コマンドを使用して HOLD ファイルに対して TABLE リクエストを発行し、myapp アプリケーションフォルダ内にファイルが存在するかどうかを確認します。行が返されない場合は、アプリケーションフォルダが存在しません。その場合、アプリケーションフォルダが作成された上で、処理が続行されます。行が返された場合は、アプリケーションフォルダが作成されずに処理が続行されます。

```
APP LIST HOLD
TABLE FILE FOCAPPL
PRINT * ON TABLE HOLD WHERE APPNAME = 'myapp'
END
-IF &LINES GT 0 THEN GOTO RESUME
APP CREATE myapp
-RESUME
```

APP QUERY

APP QUERY コマンドは、指定したアプリケーションフォルダ内のファイルをリスト表示します。クエリは、指定したアプリケーションフォルダおよびネストされたアプリケーションフォルダに対して実行することができます。

構文 コンポーネントのリスト表示

```
APP QUERY app1[/app1a...] [app2[/app2a]...] ...
[appn[/appna]] [HOLD]
```

説明

app1[/app1a...appn[/appna]]

アプリケーションフォルダ名です。ネストされたアプリケーションフォルダ名を指定することもできます。複数のアプリケーションフォルダ名を指定する際に、すべての名前が1行に収まらない場合は、1行目の末尾に継続文字 (-) を追加した上で、次の行に残りのアプリケーションフォルダ名を記述します。

HOLD オプションを使用する場合、出力は focappq.ftm という一時ファイルに書き込まれます (PDS 展開では FOCAPPQ)。次にこのファイルをリクエストで使用してレポートを作成したり、catalog/focappq マスターファイルを使用してアクションを実行したりできます。

制限事項 アプリケーション内のすべてのファイルがリスト表示されます。UNIX のようなシステムでは、大文字と小文字の両方のファイル名がリストに含まれる場合がありますが (例、MYPROC.FEX と myproc.fex)、リクエストでは小文字のファイル名のみがアクセスされます。

例 アプリケーションファイルのリスト表示

次のリクエストは、アプリケーションファイルのリストを表示します。

```
APP QUERY abc
```

APP QUERY の出力は次のように表示されます。

```
BEGIN-APP-QUERY: abc
24/10/2014 21.38.28      4 F myproc1.fex
24/10/2014 21.38.35      4 F myproc1.fex
24/10/2014 21.37.49      4 F myapp1
24/10/2014 21.32.36      0 D myapp2
END-APP-QUERY
```

次のリクエストは、HOLD オプションを使用して格納されているファイルをリスト表示します。

```
APP QUERY ABC HOLD
SQL SELECT DATE, TIME, GMTTIME, SIZE, OTYPE, FILENAME, APPNAME FROM
FOCAPPQ ;
END
```

APP QUERY の出力は次のように表示されます。

DATE	TIME	GMTTIME	SIZE	OTYPE	FILENAME	APPNAME
----	----	-----	----	-----	-----	-----
24/10/2014	21.38.28	1414201108	4	F	myproc1.fex	abc
24/10/2014	21.38.35	1414201115	4	F	myproc2.fex	abc
24/10/2014	21.37.49	1414201069	4	D	myapp1	abc
24/10/2014	21.32.36	1414200756	0	D	myapp2	abc

APP QUERY ... HOLD を使用した場合、APP QUERY を使用した場合より多くの情報が出力されます。FILENAME 列は実際には 70 バイト幅ですが、上記の出力では、読みやすくするためにブランク領域が部分的に除外されています。

次の例では、APP QUERY HOLD コマンドを使用して、abc アプリケーションフォルダ内に myproc1.fex ファイルが存在するかどうかを確認します。ファイルが存在しない場合は、プロシジャが終了します。ファイルが存在する場合は、プロシジャが続行されます。

```
APP QUERY abc HOLD
TABLE FILE FOCAPPQ
PRINT * ON TABLE HOLD
WHERE APPNAME = 'abc'
WHERE FILENAME = 'myproc1.fex'
END
-IF &LINES GT 0 THEN GOTO RESUME
-TYPE Procedure Not Found ... exiting!
-EXIT
-RESUME
```

拡張されたカタログ情報の取得

ここでは、サーバカタログの検索に関する概要について説明します。

catalog/sysapps

catalog/sysapps テーブルには、パス上の物理オブジェクトのメタデータが格納されます。

ここでは、開発者が一般に必要なとする基本的な使用方法についてのみ説明します。ディスク上のマスターファイルには、ここで説明する属性より多くの属性が記述されています。マスターファイルを直接開いて、その他の使用方法を確認することができます。catalog/sys* グループの一連のファイルは、今後のバージョンで変更される場合があります (通常は上位互換性あり)。特定の従属関係 (通常はオブジェクトサイズ) を持つアプリケーションは記述しないでください。上位互換性の問題が発生する可能性があります。

例

アプリケーションフォルダ内のファイルのリスト表示

次のリクエストは、「abc」というアプリケーションフォルダ内のアプリケーションフォルダ名、アプリケーションパス、ファイル名、ファイル拡張子をリスト表示します。

```
TABLE FILE SYSAPPS
PRINT APPNAME APPLOC FNAME FEXT
WHERE APPNAME EQ 'abc' ;
END
```

出力は次のとおりです。読みやすくするためにブランク領域が部分的に除外されています。

APPNAME	APPLOC	FNAME	FEXT
----	-----	-----	-----
abc	/usr/wf/ibi/apps/abc	myproc1	fex
abc	/usr/wf/ibi/apps/abc	myproc2	fex

次の例では、SYSAPPS テーブルを使用して、abc アプリケーションフォルダ内に myproc1.fex ファイルが存在するかどうかを確認します。ファイルが存在しない場合、プロシジャが終了します。ファイルが存在する場合、プロシジャが次の手順に移動して処理が続行されます。

```
TABLE FILE SYSAPPS
PRINT * ON TABLE HOLD
WHERE APPNAME = 'abc' ;
WHERE FNAME = 'myproc1' ;
WHERE FEXT = 'fex' ;
END
-IF &LINES GT 0 THEN GOTO RESUME
-!TYPE Procedure Not Found ... exiting!
-EXIT
-RESUME
```

catalog/sysfiles

catalog/sysfiles テーブルには、選択したオブジェクトタイプの、パス上に存在するアプリケーション名オブジェクトのメタデータが格納されます。デフォルトのファイルタイプは MASTER (マスターファイル) ですが、他のタイプを設定することもできます。別の方法で表示が制限されていない限り、選択したタイプのオブジェクトがすべて表示されます。

ここでは、開発者が一般に必要なとする基本的な使用方法についてのみ説明します。ディスク上のマスターファイルには、ここで説明する属性より多くの属性が記述されています。その他の使用方法を理解するために、マスターファイルを直接開いて精査することができます。catalog/sys* グループの一連のファイルは、今後のバージョンで変更される場合があります (通常は上位互換性あり)。特定の従属関係 (通常はオブジェクトサイズ) を持つアプリケーションは記述しないでください。上位互換性の問題が発生する可能性があります。

例 APP MASTER オブジェクトのリスト表示

次のリクエストは、MASTER ファイルタイプ (デフォルト) のファイル名、アプリケーションパスを含めたファイル名、ファイル拡張子のリストを表示します。

```
TABLE FILE SYSFILES
PRINT FILENAME LGNAME PHNAME EXTENSION
END
```

出力は次のとおりです。読みやすくするためにレコードおよびブランク領域が部分的に除外されています。

FILENAME	LGNAME	PHNAME	EXTENSION
-----	-----	-----	-----
...			
mydata	MASTER	baseapp/mydata.mas	mas
mdschema	MASTER	_edahome/catalog/mdschema.mas	mas

例 APP FOCEXEC オブジェクトのリスト表示

次のリクエストは、ファイルタイプを FOCEXEC に設定し、FOCEXEC ファイルタイプのファイル名、アプリケーションパスを含めたファイル名、ファイル拡張子のリストを表示します。

```
SQL FMI SET SYSFILES FOCEXEC
TABLE FILE SYSFILES
PRINT FILENAME LGNAME PHNAME EXTENSION
END
```

出力は次のとおりです。読みやすくするためにレコードおよびブランク領域が部分的に除外されています。

FILENAME -----	LGNAME -----	PHNAME -----	EXTENSION -----
...			
myproc1	FOCEXEC	baseapp/myproc1	fex
myproc2	FOCEXEC	baseapp/myproc2	fex
...			

注意: データが限定され、1 つのオブジェクトのみが返される場合、LGNAME 列の値が DEFAULT に切り替わります。

SQL FMI SET SYSFILES コマンドの有効な値は、サーバで有効な任意のファイルタイプです。一例として、FOCUS、FOCEXEC、STY、PDF、ACCESS があります。有効なファイルタイプのリストについては、97 ページの「[APP コマンドによるファイルタイプの指定](#)」を参照してください。

例 SYSFILES テーブルによるファイルの有無の確認

次の例では、ファイルタイプを FOCEXEC に設定し、SYSFILES テーブルを使用して、ファイル名が myproc1、拡張子が fex のファイルの有無を確認します。行が返されない場合、そのファイルは存在せず、プロシジャが終了します。ファイルが存在する場合、プロシジャが処理の続行位置に移動します。

```
SQL FMI SET SYSFILES FOCEXEC
TABLE FILE SYSFILES
PRINT FILENAME ON TABLE HOLD
WHERE FILENAME = 'myproc1' ;
WHERE EXTENSION = 'fex' ;
END
-IF &LINES GT 0 THEN GOTO RESUME
- TYPE Procedure Not Found ... exiting!
-EXIT
-RESUME
```

APP HELP

APP HELP コマンドは、すべての APP コマンドに関するヘルプ情報を提供します。

構文 APP コマンドに関する情報の取得

APP HELP command parameters

説明

command

任意の有効な APP コマンドを指定します。

parameters

コマンドで使用可能なパラメータまたは必須のパラメータです。

APP コマンドの使用制限

サーバ管理者は、他のクラスのユーザが APP 環境を変更できないように APP コマンドの使用を制限することができます。この設定は、アプリケーション管理者および一般ユーザに対して構成することができます。

この制限を設定すると、このサーバ設定の影響を受けるすべてのユーザインターフェース (例、Reporting Server ブラウザインターフェース、Data Migrator デスクトップインターフェース、BI Portal) で、サーバの approot 設定で定義したすべてのアプリケーションが表示される代わりに、有効なアプリケーション検索パスのアプリケーションのみが表示されます。また、ユーザが発行する特定の APP コマンドの使用を制限することで、ユーザが本来の目的を無視してこれらのコマンドを発行することを防止します。

サーバ管理権限を持つユーザがクライアントソフトウェアからログインしたり、Reporting Server ブラウザインターフェースにログインしたりする場合は、管理者の完全な権限を確保するため、この制限は動的にオフになります。

この権限は、すべてのセキュリティ設定でサポートされます。

手順 APP コマンドの使用を制限するには

サーバ管理者は、他のユーザによる APP コマンドの使用を制限することができます。

1. [アクセスコントロール] ページに移動します。
2. [ロール] フォルダの [アプリケーション管理者] または [一般ユーザ] を展開します。
3. ユーザまたはグループを右クリックし、[全般権限] を選択します。

[全般権限] ページが表示されます。

4. デフォルト設定では、APATH 権限 ([各自のアプリケーションパスの変更 (APPLOCK を除く)]) が選択されています。下部に示されている APP コマンドの使用を現在のユーザまたはグループに対して制限するには、このチェックをオフにします。
5. [保存] をクリックし、設定を確定します。

アプリケーションパスの変更を規制すると、指定したユーザまたはグループによる次の APP コマンドの使用が制限されます。

- ☐ 次のコマンドを使用すると、エラーメッセージが表示されます。
 - ☐ APP CREATE
 - ☐ APP DELETE
 - ☐ APP RENAME
- ☐ 現在のアプリケーションパス以外のアプリケーションを参照する場合は、次のコマンドが制限されます。この場合、エラーメッセージは表示されません。
 - ☐ APP APPENDPATH
 - ☐ APP HOLD
 - ☐ APP HOLDDATA
 - ☐ APP HOLDMETA
 - ☐ APP MAP
 - ☐ APP PATH
 - ☐ APP PREPENDPATH
 - ☐ APP COPY
 - ☐ APP COPYF[ILE]
 - ☐ APP DELETEF[ILE]
 - ☐ APP MOVEF[ILE]
 - ☐ APP RENAMEF[ILE]

メタデータおよびプロシジャへのアクセス

永久ファイルには、別のアプリケーションによってセッション前に作成されたメタデータおよびプロシジャが保存されています。これらはセッション終了後も他のアプリケーションで使うことができます。

検索ルール

ファイル名がアプリケーションフォルダ名を含む完全修飾名ではない場合、検索順序は次のとおりです。

1. エージェントの現在のディレクトリ (edatemp/tsnnnnn)
2. メタデータファイルに対して APP HOLDMETA、HOLD データファイルに対して APP HOLDDATA を使用して設定されたアプリケーション
3. APP PATH 内に設定されたアプリケーション
4. baseapp アプリケーション
5. EDAHOME/catalog
6. ストアドプロシジャのみ - ファイルが見つからない場合、サーバはファイルが FILEDEF コマンドにより割り当てられたかどうかを確認します。割り当てられていれば、実行します。

例

検索パス

この例では、次のように APP HOLDMETA コマンドで検索の開始先アプリケーションフォルダが設定されている場合の検索パスについて説明します。

APP HOLDMETA APP1

たとえば、次のように 1 部構成名で記述されたプロシジャを実行します。

EX ABC

最初に、次のプロシジャが実行されます。

profile.fex in APP1 application

続いて、次のプロシジャが実行されます。

EX APP1/ABC

プロシジャ ABC が APP1 アプリケーションフォルダに存在しない場合、サーバは、標準の検索パスに基づいてプロシジャを検索して実行します。

プロシジャファイルの作成ルール

ファイル名が完全修飾されていない場合、または APP HOLD、APP HOLDMETA、APP HOLDDATA、FILEDEF コマンドを使用して別の場所に切り替えられていない場合、ファイルはエージェントの一時アプリケーション領域に作成され、エージェントが解放された後は消滅します。

関連情報については、101 ページの「[出力リダイレクトコマンド](#)」を参照してください。

マスターファイルおよびプロシジャの場所の特定

パスを設定すると、WHENCE コマンドを使用してマスターファイルおよびプロシジャの場所を特定することができます。

構文 ファイルの場所の特定

WHENCE コマンドを発行して、アプリケーションパス上で最初に一致したファイルの完全修飾パスを取得することができます。APP WHENCE コマンドを発行して、ファイルが格納されているアプリケーションパス上の最初のアプリケーションの名前を取得することができます。ネストされたアプリケーションディレクトリを指定することもできます。

アプリケーションパス上で最初に一致したマスターファイル、プロシジャ、その他の FOCUS ファイルタイプのパスを取得するには、次のコマンドを発行します。

WHENCE filename filetype

マスターファイル、プロシジャ、その他の FOCUS ファイルタイプを格納するアプリケーションパス上の最初のアプリケーションの名前を取得するには、次のコマンドを発行します。

APP WHENCE filename filetype

説明

filename

位置を特定するファイル名を指定します。

filetype

位置を特定するファイルのタイプを指定します。

例 ファイルの場所の特定

次のコマンドは、アプリケーションパス上で最初に一致した pivot_demo.xlsx という名前の Excel ファイルのパスを取得します。

WHENCE pivot_demo.xlsx

出力結果は次のとおりです。

```
C:¥ibi¥apps¥retail91¥uploads¥pivot_demo.xlsx
```

次のコマンドは、アプリケーションパス上で最初に一致した pivot_demo.xlsx という名前の Excel ファイルを格納するアプリケーションの名前を取得します。

```
APP WHENCE pivot_demo.xlsx
```

出力結果は次のとおりです。

```
APPNAME: retail/uploads
```

既存のデータファイルへのアクセス

次の方法により、既存のデータファイルを割り当てることができます。

- ❑ マスターファイルの DATASET キーワード
- ❑ 非 FOCUS データソース (FIXED、RMS、XML) に対する FILEDEF コマンド
- ❑ FOCUS データソースに対する USE コマンド

各割り当てにはそれぞれ単一の方法を使用することをお勧めします。

データファイルの作成ルール

新しく作成されたデータファイルの場所は、次のように決定されます。

1. APP HOLDDATA により設定されたアプリケーションは、すべての HOLD ファイルに対して適用されます。
2. FILEDEF コマンドの場合、各データファイルの設定が適用されます。

ファイルを作成するリクエストでは、レコード長やレコードフォーマットなどの、ファイルの DCB パラメータが決定されます。

関連情報については、101 ページの「[出力リダイレクトコマンド](#)」を参照してください。

構文

FILEDEF コマンドの発行

```
FI[LEDEF] filedes DISK app/[appa.../]physfile.ftm
```

説明

filedes

ファイルを指定します。

`app/[appa...]`

アプリケーションフォルダ名です。ネストされたアプリケーションフォルダ名を指定することもできます。

`physfile.ftm`

アプリケーション内に存在する物理ファイル名です。

構文

ファイルを連結する FILEDEF コマンドの発行

```
FI[LEDEF] concatname DISK [app1/]filename1.ext
FI[LEDEF] name2 DISK [app2/]filename2.ext
...
FI[LEDEF] namen DISK [appn/]filenamen.ext
FI[LEDEF] concatname CONCAT name2 ... namen
```

説明

`concatname`

連結するファイルのいずれかの `ddname` および連結ファイルの名前です。リクエストにはこの名前を使用します。FILEDEF CONCAT コマンドで一度使用した個別の `ddname` は、使用できなくなります。

`name2 ... namen`

連結に追加されるファイルの `ddname` です。

`app1 ... appn`

アプリケーションフォルダ名を指定します。ネストされたアプリケーションフォルダ名を指定することもできます。

`filename1.ext ... filenamen.ext`

物理ファイル名です。

例 FILEDEF を使用したファイルの連結

次のリクエストは、3 つのファイル (file1.ftm、file2.ftm、file3.ftm) を作成します。

```
APP HOLD app1
TABLE FILE WF_RETAIL_LITE
SUM COGS_US REVENUE_US
BY STATE_PROV_NAME
WHERE STATE_PROV_NAME LE 'F'
WHERE COUNTRY_NAME EQ 'United States'
ON TABLE HOLD AS file1 FORMAT ALPHA
END
-RUN
TABLE FILE WF_RETAIL_LITE
SUM COGS_US REVENUE_US
BY STATE_PROV_NAME
WHERE STATE_PROV_NAME GT 'F' AND STATE_PROV_NAME LE 'M'
WHERE COUNTRY_NAME EQ 'United States'
ON TABLE HOLD AS file2 FORMAT ALPHA
END
-RUN
TABLE FILE WF_RETAIL_LITE
SUM COGS_US REVENUE_US
BY STATE_PROV_NAME
WHERE STATE_PROV_NAME GT 'M'
WHERE COUNTRY_NAME EQ 'United States'
ON TABLE HOLD AS file3 FORMAT ALPHA
END
```

次のコマンドは、これら 3 つのファイルを連結します。

```
FILEDEF FILE1 DISK app1/file1.ftm
FILEDEF FILE2 DISK app1/file2.ftm
FILEDEF FILE3 DISK app1/file3.ftm
FILEDEF FILE1 CONCAT FILE2 FILE3
```

次のプロシジャは、連結されたファイルに対してリクエストを発行します。

```
TABLE FILE FILE1
SUM COGS_US REVENUE_US
BY STATE_PROV_NAME
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
GRID=OFF, SIZE=8,$
END
```

下図は、出力結果を示しています。

<u>STATE PROV NAME</u>	<u>COGS US</u>	<u>REVENUE US</u>
Alabama	\$12,774.00	\$18,612.25
Alaska	\$20,959.00	\$29,991.52
Arizona	\$9,915.00	\$15,100.19
Arkansas	\$10,822.00	\$15,000.70
California	\$61,009.00	\$83,133.90
Colorado	\$20,596.00	\$28,140.37
Connecticut	\$7,044.00	\$10,536.34
Delaware	\$5,043.00	\$6,254.51
District of Columbia	\$5,151.00	\$7,228.56
Florida	\$52,541.00	\$72,244.02
Georgia	\$27,356.00	\$37,988.80
Hawaii	\$2,770.00	\$3,997.04
Idaho	\$10,499.00	\$14,785.35
Illinois	\$43,464.00	\$60,364.26
Indiana	\$30,904.00	\$42,367.87
Iowa	\$8,964.00	\$12,316.32
Kansas	\$19,279.00	\$26,478.06
Kentucky	\$9,070.00	\$13,290.75
Louisiana	\$25,212.00	\$35,789.99
Maine	\$2,542.00	\$3,795.93
Maryland	\$23,169.00	\$32,319.96
Massachusetts	\$16,046.00	\$22,463.11
Michigan	\$45,308.00	\$62,878.30
Minnesota	\$35,815.00	\$50,591.70
Mississippi	\$6,661.00	\$10,078.13
Missouri	\$30,450.00	\$41,776.64
Montana	\$9,920.00	\$14,315.84
Nebraska	\$14,626.00	\$19,472.62
Nevada	\$10,381.00	\$14,380.39
New Hampshire	\$3,704.00	\$5,576.98
New Jersey	\$24,128.00	\$33,598.01
New Mexico	\$7,590.00	\$10,119.98
New York	\$49,956.00	\$69,800.34
North Carolina	\$27,789.00	\$39,716.65
North Dakota	\$9,478.00	\$13,552.55
Ohio	\$34,478.00	\$49,394.62
Oklahoma	\$19,010.00	\$26,164.76
Oregon	\$15,637.00	\$21,709.21
Pennsylvania	\$46,154.00	\$63,984.09
Puerto Rico	\$436.00	\$714.97
Rhode Island	\$583.00	\$818.97
South Carolina	\$4,309.00	\$7,033.86
South Dakota	\$6,684.00	\$9,311.33
Tennessee	\$19,889.00	\$27,732.33
Texas	\$101,081.00	\$141,345.92
Utah	\$7,074.00	\$10,599.14
Vermont	\$5,212.00	\$7,474.33
Virginia	\$32,148.00	\$44,582.23
Washington	\$18,101.00	\$24,042.12
West Virginia	\$5,001.00	\$7,000.03
Wisconsin	\$29,173.00	\$39,605.50
Wyoming	\$5,633.00	\$7,136.01

一時ファイルの割り当て

一時ファイルとは、セッションを終了した後に削除される一時的なファイルです。デフォルト設定では、すべての一時データファイル (HOLD および FOCSORT ファイル用) および一時メタデータファイル (一時マスターファイルやアクセスファイルなど) は、エージェントの一時領域に作成されます。この領域は、TSCOM ID に対応します。たとえば、TSCOM ID が TS000001 の場合、一時ファイルは `//edatemp/ts000001` にあります。

一時領域の使用と作成先

サーバは、さまざまな処理で一時領域を使用します。この一時領域は、ログとトレースや一時作業ファイル (例、HOLD ファイル) など、あらゆる用途に使用されます。この一時領域は、実際のディスク領域を消費しますが、サーバ構成に基づいて再利用されたり、クリーンアップされたりします。たとえば、セキュリティが設定されたサーバでは、切断時にエージェントディレクトリがクリーンアップされます。最終的に、サーバの再起動時にこの一時領域がクリーンアップされます。サーバが停止状態で、再起動される予定がない場合 (例、サーバがテスト用に構成されている場合)、サーバ管理者は、次のコマンドを実行して一時領域を元の状態に戻すことができます。

```
edastart -cleardir
```

一時ディスク領域の使用

デフォルト設定では、一時領域は、製品のインストールに使用されたファイルシステム内の構成ディレクトリ (ffs、wfs、または dm) 配下に作成されます。edatemp ディレクトリは 2 レベルで構成されます。

第 1 レベルのディレクトリは、トレースが有効な場合にトレースファイル用に使用されます。デフォルト設定では、このディレクトリの名前は edatemp で、一般に「EDACONF の edatemp ディレクトリ」と呼ばれます。

サブディレクトリは、さまざまな処理で作成される一時ファイルやトレースに使用されます。たとえば、TCP および HTTP リスナ、ワークスペースマネージャ、エンドユーザエージェント処理があります。各ユーザは、作業を他のユーザから分離するためにプライベートデータエージェントディレクトリに割り当てられ、このディレクトリは切断時にクリーンアップされます。エージェントサブディレクトリの名前は `tsnnnn` です。nnnn は、データエージェントに割り当てられた ID 番号を表します。

edatemp ディレクトリは、EDATEMP 変数の指定先を変更することで、別のファイルシステムを使用するよう構成することができます。構成手順と構文はプラットフォームごとに異なり、オペレーティングシステム変数として構成できる場合もあれば、サーバを起動する前にサーバ構成ファイル (edaserve.cfg) の変数として構成できる場合もあります。

サーバ構成ファイルを使用する場合は、Reporting Server ブラウザインターフェースの [ワークスペース] フォルダで edaserve.cfg を編集し、次の行を追加する方法をお勧めします。

```
edatemp = directory
```

説明

directory

プラットフォームに応じた構文を使用して、ディレクトリ名をフルパスで指定します。保存後、サーバを再起動します。

注意：サーバ構成ファイル (edaserve.cfg) を編集する方法は、Windows でのみ使用できます。

構文

EDATEMP 変数の指定

UNIX および Linux の場合 edastart.sh ファイルで edatemp 変数を次のように記述することができます。

```
export EDATEMP=/u/iway/edatemp
```

注意：この方法は、Windows にも適用されます。

構文

一時ファイルの事前割り当て

次の手法を使用して、ユーザごとのファイルを事前に割り当てることができます。

❑ UNIX および Linux の場合

```
FILEDEF XXX DISK /u/another/area/xxx.dat
```

説明

```
/u/another/area
```

ファイルを保持するのに十分な空き領域があります。

❑ Windows の場合

```
FILEDEF XXX DISK C:¥tmp¥another¥area¥xxx.dat
```

説明

```
tmp¥another¥area
```

ファイルを保持するのに十分な空き領域があります。

❑ FOCUS ファイルの場合

FILEDEF および USE コマンドを使用して、edatemp 領域の外部に FOCUS ファイルを作成することができます。

```
FILEDEF NAME DISK /{pathname}{filename}.foc .....  
USE NAME NEW  
END
```

アプリケーションツール

アプリケーションプログラムは、TABLE FILE ... END リクエストのような単純なものにするこ
とも、動的入力値、GOTO ステートメント、ループロジック (ダイアログマネージャと総称さ
れる) により制御される置換可能な変数を記述した複雑なものにすることもできます。

EX プロシジャおよび変数

EX プロシジャは、アプリケーションパスに存在する別のプロシジャを実行します。PDS 以外
のプラットフォームでは、このプロシジャの拡張子は必要なく、自動的に .fex と見なされま
す。完全な構文は次のとおりです。

```
EX[EC] [appname/]procedure[.fex] [parameter=value[,parameter=value][,...]
```

パラメータは文字 (例、LASTNAME) または数値 (例、1) にすることができ、リスト内に両者を
混在させることもできます。このプロシジャを実行する前に、特定のアプリケーション名を参
照することができ、そのアプリケーションがアプリケーションパス上に存在する必要はありま
せん。パラメータ名が数値の場合でも、次の例に示すように、リスト内の位置からそれがパラ
メータと見なされます。

```
EX mytest DETAIL=ALL,QUARTERLY,PERSON=JAMES MADISON,5=MANAGER
```

たとえば、プロシジャに &DETAIL、&2、&PERSON 変数 (パラメータ) を含める場合は、次のよ
うなコンテキストを使用します。

```
TABLE FILE ...  
WHERE PERSON IS '&PERSON' AND PERIOD EQ '&2'  
...  
END
```

変数には、ローカル変数 (&) とグローバル変数 (&&) の 2 種類があります。ローカル変数 (&) は、その変数が作成されたプロシジャ内でのみ有効です。グローバル変数 (&&) は、セッションが終了するまで有効な変数です。また、定義済みのローカル変数 (&) も多数あります。たとえば、&LINES (リクエストから返されたレコード)、&FOCUSER (現在のユーザ ID)、&FOCFOCEXEC (現在実行中のプロシジャ名)、&FOCCODEPAGE (サーバの NLS コードページ) などがこれに該当します。

EX EDAMAIL

内部プロシジャの EDAMAIL を使用することで、プロシジャから Email を送信し、その Email に特定ファイルの内容を Email 本文または添付ファイルとして含めることができます。この方法は、特別なエラー条件やレポート実行の完了などのイベント発生時に Email 通知を送信する際に役立ちます。EDAMAIL を使用するには、サーバのワークスペースで SMTP メールサーバノードが構成され、そのメールサーバが稼動している必要があります。

EDAMAIL の構文には、次の 2 種類の形式があります。

- ❑ 固定形式では、構文で定義された順序でパラメータ値を指定する必要があります。
- ❑ 拡張形式では、パラメータの位置は固定されず、名前と値を組み合わせたパラメータを使用します。拡張形式では、追加の SMTP Email クライアント機能 (例、返信先、重要度) もサポートされます。

最初のパラメータに名前と値の組み合わせであることを示す等号 (=) が含まれている場合、拡張形式であると見なされます。

Email アドレスの基本形式 (foo@foo.com または <foo@foo.com>) は、EDAMAIL の固定形式でも拡張形式でも使用することができます。Email アドレスの拡張形式 (My Support <support@foo.com>、"My Support" <support@foo.com>、"My Support <support@foo.com>"、"Support, Me <support@foo.com>") は、EDAMAIL の拡張形式でのみサポートされます。

複数の Email アドレスをサポートするパラメータ (例、to) では、適切な形式および区切り文字を使用して複数の Email アドレスを使用することができます。

EDAMAIL の固定形式で複数の Email アドレスを使用する場合は、Email アドレスの区切り文字としてセミコロン (;) を使用します。Email アドレス文字列全体は必要に応じて二重引用符 (") で囲むことができます。ただし、EDAMAIL の固定形式は基本形式の Email アドレスのみサポートします。

EDAMAIL の拡張形式で複数の Email アドレスを使用する場合は、Email アドレスの区切り文字として通常はセミコロン (;) を使用します (例外的にカンマ (,) を使用)。Email アドレス文字列全体は、いずれかのアドレスにカンマ (,) が含まれる場合 (例、Support, Me <support@foo.com>) は、一重引用符 (') で囲む必要があります。Email アドレスがすべて基本形式 (例、foo@foo.com または <foo@foo.com>) の場合は、区切り文字としてカンマ (,) を使用することができ、Email アドレス文字列全体は、一重引用符 (') または二重引用符 (") で囲む必要があります。

EDAMAIL から作成された Email ヘッダを実際に使用、表示する際は、使用中の SMTP サーバ (およびその他の中間 SMTP ホップ) およびユーザの Email クライアント (例、Outlook、Gmail、Hotmail) の機能が使用されます。これらは、送信者の Email ヘッダでは直接制御されません。そのため、Email クライアント環境によっては、Email の使用時に期待される動作が得られない場合があります。たとえば、古い Email クライアントでは、新しいヘッダタイプが認識されない場合や、返信先の値が無視される場合があります。機能上の問題が発生した場合は、最初に経験豊富な SMTP/Email 管理者が調査した上で、それがクライアント側の問題かどうかを特定する必要があります。

EDAPRINT ログには、すべての EDAMAIL 実行が記録されます。ログには、EDAMAIL コマンドを実行したユーザ ID、送信者、受信者に関する情報が表示されます。以下はその例です。

```
02/01/2019 16:58:41.395 I EDAMAIL u=PTH
$srvadmin,from=user2@ibi.com(),to=(user1@ibi.com)
```

構文 パラメータ固定形式での EX EDAMAIL の使用

構文は次のとおりです。

```
EX EDAMAIL to, cc, bcc,from, subject, [flag], filetype, data
```

説明

to

Email の受信者です。Email アドレスの区切り文字としてセミコロン (;) を使用することで、複数の Email アドレスを指定することができます。

cc

CC 受信者です。Email アドレスの区切り文字としてセミコロン (;) を使用することで、複数の Email アドレスを指定することができます。

bcc

BCC 受信者です。Email アドレスの区切り文字としてセミコロン (;) を使用することで、複数の Email アドレスを指定することができます。

from

Email の送信者です。

subject

Email の件名です。件名にカンマ (,) を含める場合は、件名を一重引用符 (') または二重引用符 (") で囲む必要があります。

flag

a または A に設定した場合、ファイルが添付されて送信されます。値を設定しない場合、ファイルが Email 本文に埋め込まれます。その他の値はすべて無視されます。この値は、filetype パラメータをブランクにした場合にも無視されます。

filetype

Email メッセージにファイルを使用する場合のデータファイルタイプを定義します。MASTER、FOCEXEC、HTML、TEXT など、あらゆるアプリケーションファイルタイプが有効です。Email メッセージ本文への埋め込み機能を使用する場合は、このパラメータをブランクにします。

data

Email メッセージ本文の埋め込みデータストリーム、または Email メッセージ本文が格納されている [app/]filename ファイルです。埋め込みデータストリーム機能を使用する場合は、filetype パラメータをブランクにする必要があります。EDAMAIL 機能を使用する際は、-LINES {n|*} 機能を使用することで、埋め込み Email メッセージ本文を複数行に分割することができます。

プロシジャ内で Email メッセージ本文の埋め込みデータストリームを複数行に分割した場合、結果の Email は単一行の出力になります。メッセージ本文が複数行で出力されるのは、ファイルオプションのメッセージ本文を使用した場合です。

例

HTML ファイルのメッセージ本文としての Email 送信

```
TABLE FILE file1
PRINT A B C
ON TABLE HOLD AS MYFILE FORMAT HTML
END
EX EDAMAIL user1@corp1.com, user2@corp1.com, File1 Report,,HTML, MYFILE
```

例

HTML ファイルのメッセージ添付ファイルとしての Email 送信

```
TABLE FILE file1
PRINT A B C
ON TABLE HOLD AS MYFILE FORMAT HTML
END
EX EDAMAIL user1@corp1.com, user2@corp1.com, File1 Report,a,HTML, MYFILE
```

例 複数行の埋め込みメッセージの Email 送信

```
...  
EX -LINES * EDAMAIL user1@corp1.com, user2@corp1.com, &SUBJECT,,  
    Run result for &TESTNAME is:  
&RESULT  
EDAMAIL*
```

構文 拡張形式での EX EDAMAIL の使用

最初のパラメータで等号 (=) が検知されると、名前と値を組み合わせた EDAMAIL 拡張形式が有効になります。パラメータ名では大文字と小文字は区別されません。また、パラメータ名は任意の順序で指定できますが、message パラメータ (使用する場合は最後に配置する必要があります)。

構文は次のとおりです。

```
EX EDAMAIL to=addresslist,  
[toaddr= {addresslist|[%[app/]addresslist.fex}, ]  
[cc=cadrlist1,]  
[ccaddr=cadrlist2,]  
[bccaddr=bcadrlist,] [from=address,]  
[fromaddr=address,] [replyto=address,] [importance=low|normal|high,]  
[subject=string,] [flags=value,] [filetype=extension,]  
[filename=file,] [message=body message]
```

説明

to=addresslist

ユーザの Email クライアントで表示される Email 受信者です。また、toaddr ヘッダが指定されていない場合は、この Email 受信者が Email クライアントの [全員へ返信] にも使用されます。Email アドレスの区切り文字としてセミコロン (;) を使用することで、複数の Email アドレスを指定することができます。Email アドレス文字列全体を一重引用符 (') で囲んだ場合は、区切り文字としてカンマ (,) を使用することもできます。

to パラメータを toaddr パラメータとともに使用する場合は、to の値に「Group Managers」などの任意の文字列を指定することができます。この場合、ほとんどの Email クライアントでは、[宛先] テキストボックスに擬似タイトルが表示され、toaddr パラメータで使った実際の Email アドレスは表示されません。[宛先] テキストボックスを強制的にブランクにするには、to="" を使用します。

toaddr=addresslist

Email の受信者です。この値を指定しない場合、SMTP サーバは to ヘッダを使用します。Email クライアントでは、この toaddr 値が [全員へ返信] にも使用されます。Email アドレスの区切り文字としてセミコロン (;) を使用することで、複数の Email アドレスを指定することができます。Email アドレス文字列全体を一重引用符 (') で囲んだ場合は、区切り文字としてカンマ (,) を使用することもできます。

toaddr=%[app/]addresslist.fex

実行時に Email の受信者リストを生成するプロシジャ (FOCEXE) です。Email クライアントでは、この toaddr 値が [全員へ返信] にも使用されます。実際のプロシジャ (FOCEXEC) は、Email アドレスを含むすべてのデータソースに対して実行できます。ただし、1 行につき 1 組の二重引用符 (") で囲まれた 1 つの Email アドレスとしてデータのフォーマットを設定するには、単一フィールドの PRINT および ON TABLE PCHOLD FORMAT COMT FORMATTED 構文を使用する必要があります。

実際のプロシジャ (FOCEXEC) の拡張子には、.fex を指定する必要があります。また、この機能は実行中のサーバコンテキストでのみ使用できます。その他のモードではサポートされません。

cc=cadrlist1

ユーザの Email クライアントで表示される CC 受信者です。また、ccaddr ヘッダが指定されていない場合は、この CC 受信者が Email クライアントの [全員へ返信] にも使用されます。Email アドレスの区切り文字としてセミコロン (;) を使用することで、複数の Email アドレスを指定することができます。Email アドレス文字列全体を一重引用符 (') で囲んだ場合は、区切り文字としてカンマ (,) を使用することもできます。

cc パラメータを ccaddr パラメータとともに使用する場合、cc の値に「Group Managers」などの任意の文字列を指定することができます。この場合、ほとんどの Email クライアントでは、[CC] テキストボックスに擬似タイトルが表示され、ccaddr パラメータで使用した実際の Email アドレスは表示されません。[CC] テキストボックスを強制的にブランクにするには、cc="" を使用します。

ccaddr=cadrlist2

CC 受信者です。この値を指定しない場合、SMTP サーバは cc ヘッダを使用します。Email クライアントでは、ccaddr 値が [全員へ返信] にも使用されます。Email アドレスの区切り文字としてセミコロン (;) を使用することで、複数の Email アドレスを指定することができます。Email アドレス文字列全体を一重引用符 (') で囲んだ場合は、区切り文字としてカンマ (,) を使用することもできます。

bccaddr=bcadrlist

BCC 受信者です。Email クライアントでは、bccaddr 値が [全員へ返信] にも使用されます。Email アドレスの区切り文字としてセミコロン (;) を使用することで、複数の Email アドレスを指定することができます。Email アドレス文字列全体を一重引用符 (') で囲んだ場合は、区切り文字としてカンマ (,) を使用することもできます。

from=address

Email クライアントで表示される Email 送信者です。また、この Email 送信者は、Email クライアントの [返信] (fromaddr ヘッダで上書きされていない場合) または [全員へ返信] にも使用されます。from パラメータを fromaddr パラメータとともに使用する場合、from の値に「The Boss」などの任意の文字列を指定することができます。この場合、ほとんどの Email クライアントで、[送信者] テキストボックスに擬似タイトルが表示され、fromaddr パラメータで使用した実際の Email アドレスは表示されません。

fromaddr=address

Email の送信者です。この値を指定しない場合、ほとんどの Email クライアントは、返信時に from ヘッダを使用します。

replyto=address

Email の送信者です。この値を指定しない場合、ほとんどの Email クライアントは、fromaddr または from パラメータ値を使用します。

importance=low|normal|high

重要度フラグがサポートされる Email クライアントでの Email 重要度です。有効な値は、high、normal、low です。

subject=string

Email の件名です。件名にカンマ (,) を含める場合は、件名を一重引用符 (') または二重引用符 (") で囲む必要があります。

flags=value

a または A に設定した場合、ファイルが添付されて送信されます。値を設定しない場合、ファイルが Email 本文に埋め込まれます。

filetype=extension

Email メッセージ本文にファイルを使用する場合のデータファイルタイプを定義します。MASTER、FOCEXEC、HTML、TEXT など、あらゆるアプリケーションファイルタイプが有効です。Email メッセージ本文への埋め込み機能を使用する場合は、このパラメータをブランクにします。

filename=file

Email メッセージ本文にファイルを使用する場合のデータファイルを定義します。Email メッセージ本文への埋め込み機能を使用する場合は、このパラメータをブランクにします。

message=body message

Email メッセージ本文を含む埋め込みデータストリームです。このパラメータを使用する場合は、EDAMAIL コマンドの最後に配置する必要があります。埋め込みデータストリーム機能を使用するには、filetype および filename パラメータに値を指定することはできません。EDAMAIL コマンドで EX -LINES {n}* 機能を使用することで、データストリームを複数行に分割することもできます。

プロシジャ内で Email メッセージ本文の埋め込みデータストリームを複数行に分割した場合、結果の Email は単一行の出力になります。メッセージ本文が複数行で出力されるのは、ファイルオプションのメッセージ本文を使用した場合です。

例 HTML ファイルのメッセージ本文としての Email 送信 - 複数アドレスと拡張形式の使用

```
TABLE FILE file1
PRINT A B C
ON TABLE HOLD AS MYFILE FORMAT HTML
END
EX EDAMAIL to=Managers,toaddr=user1@corp1.com;user2@corp1.com,
from=support1@corp1.com,subject=File1 Report, filetype=HTML,
filename=MYFILE
```


3

ユーザインタフェースのコーディング

この章では、セルフサービスアプリケーションのユーザインタフェースをコーディングする方法について説明します。ここでは、起動ページのハイパーリンクまたはフォームから WebFOCUS を呼び出してレポートを実行する方法を紹介します。フォームには、1 つまたは複数のコントロールを追加して、プロシジャの実行時に値の入力をユーザに要求することができます。

JavaScript を使用することにより、ユーザインタフェースに動的な機能を付加することができます。ここでは、JavaScript を使用して、フォームの入力必須項目を確認する方法について説明します。

さらに、ダイアログマネージャコマンドの -HTMLFORM を使用して、設計した表示ページにレポート出力を埋め込む方法についても説明します。

WebFOCUS ソフトウェアには、ユーザインタフェースのテンプレートが付属しています。これらのテンプレートの中には、タイプの異なるレポートを呼び出すためのさまざまな起動ページが含まれています。これらのテンプレートは、作成するアプリケーションに合わせて編集することができます。

コーディング機能についての詳細は、255 ページの「[ユーザインタフェースの機能強化](#)」を参照してください。

トピックス

- | | |
|--|--|
| ☐ 使用可能なツール | ☐ レスポンシブオートプロンプト |
| ☐ TIBCO WebFOCUS Client | ☐ HTML オートプロンプト |
| ☐ Servlet の使用 | ☐ オートプロンプトに関する考慮事項 |
| ☐ 動的複数選択ドロップダウンリストの使用 | ☐ パラメータベースのフィルタの定義 |
| ☐ adhoc レポートの有効化 | ☐ オートプロンプト機能のカスタマイズ |
| ☐ JavaScript によるフォームの確認 | ☐ デフォルト TIBCO WebFOCUS ページへのレポートの表示 |
| ☐ WebFOCUS のオートテンプレート機能 | ☐ レポート表示用 HTML ページの設計 |

使用可能なツール

次のツールを使用して、HTML ベースのユーザインターフェースページを手動でコーディングすることができます。

- ❑ **HTML キャンバス** WebFOCUS フォーム、レポート、グラフ、Web オブジェクトを統合した HTML ページをグラフィカルに作成することができます。
- ❑ **WebFOCUS テキストエディタ** このオプションは、WebFOCUS Designer または WebFOCUS App Studio で使用できます。
- ❑ **他の標準テキストエディタ** このオプションを使用すると、ユーザインターフェースページを Designer のワークスペースまたは WebFOCUS App Studio のプロジェクトディレクトリにコピーまたは移動できます。

検索パスロジックの選択

WebFOCUS Reporting Server のグローバルプロファイル (EDASPROF.PRF)、ユーザプロファイル、プロシジャでは、EDAPATH または APP PATH のいずれかのロジックを設定することができます。いずれのロジックを使用する場合も、WebFOCUS Reporting Server から次のファイルにアクセスできることを確認してください。

- ❑ プロシジャ
- ❑ WebFOCUS スタイルシート
- ❑ -HTMLFORM コマンドで呼び出すカスタム HTML 表示ページ
- ❑ レポート出力に埋め込むイメージ

使用するロジックに基づいて、上記のファイルが存在するディレクトリを EDAPATH または APP PATH に追加しておく必要があります。

手順

APP PATH ロジックを使用するには

ユーザインターフェースページをテキストエディタで作成する場合、次の手順を実行します。

インターフェースページで、次のように IBIAPP_app 変数を設定します。

```
<INPUT TYPE="HIDDEN" NAME="IBIAPP_app" VALUE="app_directory">
```

説明

[app_directory](#)

プロシジャの展開先となる WebFOCUS Reporting Server 上のディレクトリです。

例

APP PATH ロジックのコーディング

注意：この例で作成するファイルの格納場所についての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

1. 「SALESDIR」というプロジェクトに、「SALESSE」という名前のプロシジャを作成します。このプロシジャは、ユーザがリクエストしたカテゴリの Southeast 地域における売上数と売上高を出力します。

プロシジャ - SALESSE.FEX

```
TABLE FILE GGSALES
HEADING
"&CATEGORY Sales for Southeast"
SUM UNITS AND DOLLARS
WHERE (CATEGORY EQ '&CATEGORY') AND (REGION EQ 'Southeast');
END
```

2. 他社製テキストエディタを使用して、「LAUNCHSE」という名前の起動ページを手動でコーディングします。この起動ページには、Servlet を使用して WebFOCUS を呼び出し、SALESSE を実行するフォームが含まれています。ここでは、1 行テキストボックスを表示して、ユーザに CATEGORY に対する値を入力するよう要求します。

この起動ページに含まれるコードにより、IBIAPP_app 変数の値が SALESDIR ディレクトリに設定されます。公開後は、このディレクトリにプロシジャ (SALESSE) が常駐します。

起動ページ - LAUNCHSE.HTML

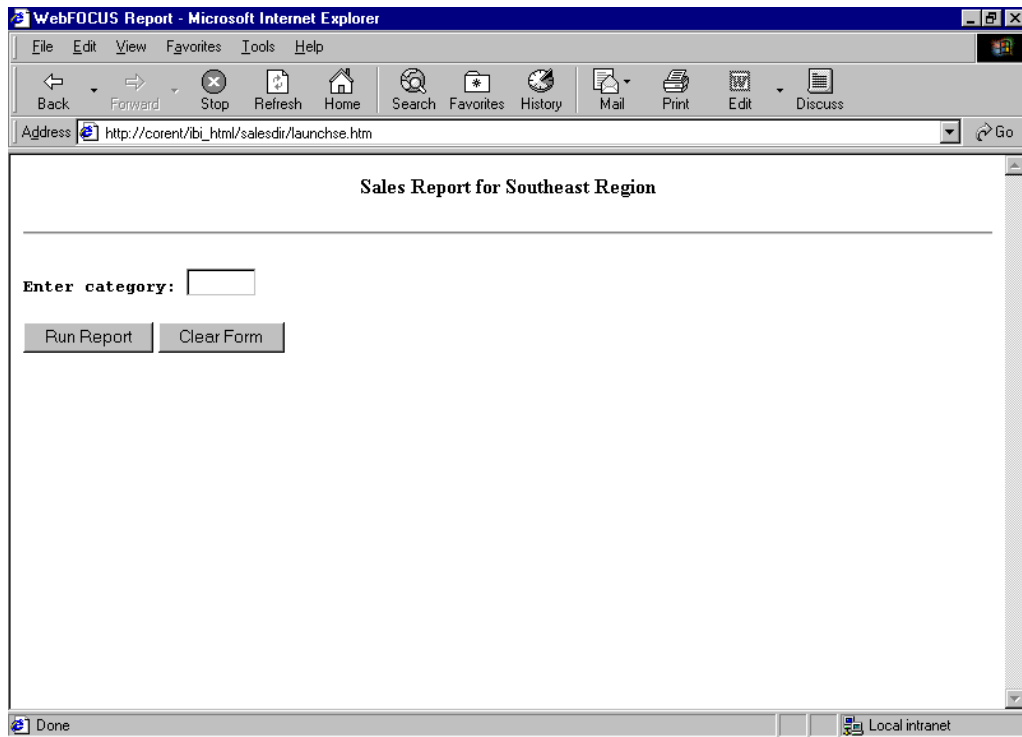
```
<HTML>
<HEAD>
<TITLE> WebFOCUS Report </TITLE>
</HEAD>
<BODY>
<H4 ALIGN=CENTER>Sales Report for Southeast Region</H4>
<HR>
<FORM ACTION="/ibi_apps/WFServlet" METHOD="get">

<INPUT TYPE="HIDDEN" NAME="IBIAPP_app" VALUE="SALESDIR">

<INPUT NAME="IBIF_ex" VALUE="salesse" TYPE="hidden">
<P ALIGN=LEFT NOWRAP><PRE>
<B>Enter category: </B><INPUT NAME="CATEGORY" TYPE="text" SIZE="6">
</PRE></P>
<INPUT NAME="submit" TYPE=SUBMIT VALUE="Run Report">
<INPUT NAME="reset" TYPE=RESET VALUE="Clear Form">
</FORM>
</BODY>
</HTML>
```

3. WebFOCUS App Studio または WebFOCUS Designer を使用して、プロシジャと起動ページを公開します。

4. ブラウザで起動ページを実行します。次のような画面が表示されます。



5. [Enter category] テキストボックスに、次のように入力します。

Food

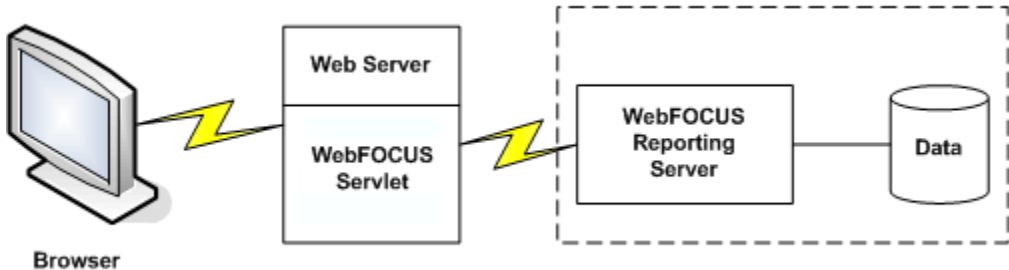
6. [Run Report] をクリックし、レポートを実行します。

PAGE 1	
Food Sales for Southeast	
Unit Sales	Dollar Sales
349829	4308731

TIBCO WebFOCUS Client

WebFOCUS をインストールすると、WebFOCUS Client が WebFOCUS Servlet を使用して起動するよう構成されます。

下図のように、WebFOCUS Servlet は Web サーバと同一のマシンに常駐し、Web サーバからのリクエストを WebFOCUS Reporting Server に転送します。



この構成では、WebFOCUS 機能がすべてサポートされ、デフォルト設定で WebFOCUS Client にインストールされた標準の WebFOCUS スクリプト (WFS) ファイルの変数が受容されます。

Servlet の使用

Servlet を使用して動的レポートを実行し、実行中に変数値を渡す場合は、次の方法で Servlet を呼び出すことができます。

- ❑ 固定値を使用するプロシジャまたは値を渡さないプロシジャを実行する場合は、HTML ハイパーリンクから呼び出します。
- ❑ ユーザが入力した値を使用してプロシジャを実行する場合は、HTML フォームから呼び出します。フォームには、値の入力を要求するラジオボタンやチェックボックスなどのコントロールを追加します。

参照

WebFOCUS Servlet 構成の要件

Web サーバで Servlet の使用を有効にするとともに、そのサーバが Servlet サポートを備えている必要があります。要件を満たす Servlet サポートには次のタイプがあります。

- ❑ ネイティブ Servlet サポートを備えた Web サーバ (例、iPlanet または WebSphere)
- ❑ Servlet エンジンプラグインを備えた Web サーバ (例、ServletExec または JRun)
- ❑ Web Application Server (例、WebSphere、WebLogic、SilverStream)

構文

ハイパーリンクからの Servlet の呼び出し

```
<A HREF="/alias/WFServlet?
IBIF_ex=procedure[&var=value[&var=value]...] ">text</A>
```

説明

alias

WebFOCUS Servlet が存在するディレクトリを指定します。Web サーバは、エイリアスを使用して物理ディレクトリに論理名を割り当てます。デフォルトのエイリアスは `ibi_apps` です。エイリアスは WebFOCUS のインストールと構成時に設定されます。

異なる Web サーバ上の WebFOCUS Servlet を呼び出すには、次のように完全修飾 URL を指定します。

http://web_server/ibi_apps/WFServlet...

procedure

実行するプロシジャ名です。

var=value

変数およびその値です。呼び出しには、プロシジャに渡す HTTP 環境変数とアプリケーション変数の両方を含めることができます。

変数と値の組み合わせを 2 つ以上渡すことができますが、それぞれの組み合わせの間にブランクを使用することはできません。変数と値の組み合わせを区切るには、区切り文字としてアンパサンド (&) を使用します。

値にブランクが埋め込まれている場合は、そのブランクをプラス記号 (+) または文字列の「%20」で置換します。

text

起動ページ上のテキストです。このテキストが、プロシジャを実行するハイパーリンクになります。

注意：URL の最大長は、4 キロバイト (4,096 バイト) です。パラメータ数に制限はありませんが、長さの合計が 4 キロバイトを超えることはできません。

例

ハイパーリンクからの Servlet の呼び出し

この例では、4 つのハイパーリンクを使用した起動ページを作成します。各ハイパーリンクは Servlet を呼び出して、次の値を渡します。

- ☐ 実行するプロシジャの名前
- ☐ プロシジャに必要な 2 つの値

注意：この例で作成するファイルの格納場所についての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

1. 「CSALES」という名前のプロシジャを作成します。このプロシジャは、特定のカテゴリおよび地域においての売上数と売上高を示す動的レポートを生成します。

プロシジャ - CSALES.FEX

```
TABLE FILE GGSALES
HEADING
"&CATEGORY Sales for &RGN Region"
SUM UNITS AND DOLLARS
WHERE (CATEGORY EQ '&CATEGORY') AND (REGION EQ '&RGN');
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

2. 「CSALES」という名前の起動ページを作成します。A HREF コードの各インスタンスが WebFOCUS Servlet を呼び出して、それを「CSALES」という名前のプロシジャに渡します。さらに、各インスタンスは CATEGORY に対して「Coffee」という値を渡し、RGN に対して「Midwest」、「Northeast」、「Southeast」、「West」のいずれかの値を渡します。起動ページのハイパーリンクは、ユーザに対して地域 (RGN) を選択するよう要求します。

起動ページ - CSALES.HTM

```
<HTML>
<HEAD>
<TITLE> WebFOCUS Report </TITLE>
</HEAD>
<BODY>
<H4 ALIGN=CENTER>Coffee Sales by Region</H4>
<HR>
<P><FONT SIZE=+2></FONT><BR>
<FONT SIZE=+1>Select a region:</FONT>
</P>
<UL TYPE=SQUARE>
<LI>

<A HREF="/ibi_apps/WFServlet?IBIF_ex=csales&CATEGORY=
Coffee&RGN=Midwest">Midwest</A>

<LI>

<A HREF="/ibi_apps/WFServlet?IBIF_ex=csales&CATEGORY=
Coffee&RGN=Northeast">Northeast</A>

<LI>
```

```
<A HREF="/ibi_apps/WFServlet?IBIF_ex=csales&CATEGORY=
Coffee&RGN=Southeast">Southeast</A>
```

```
<LI>
```

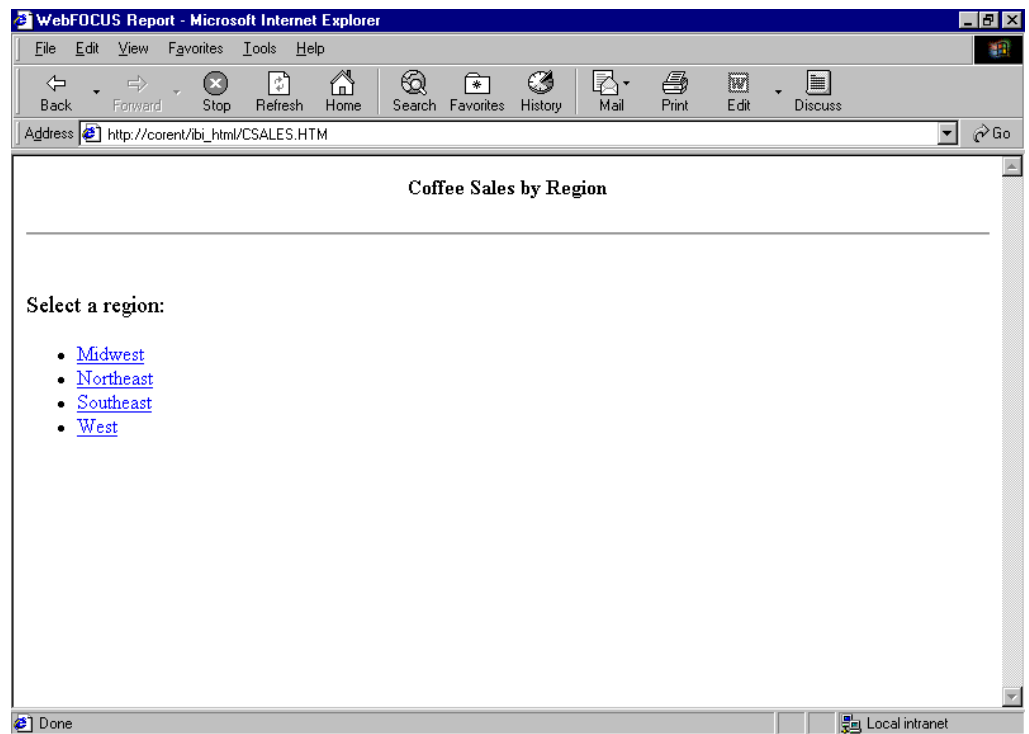
```
<A HREF="/ibi_apps/WFServlet?IBIF_ex=csales&CATEGORY=
Coffee&RGN=West">West</A>
```

```
</UL>
```

```
</BODY>
```

```
</HTML>
```

3. ブラウザで起動ページを実行します。次のような画面が表示されます。



4. ハイパーリンクの [Midwest] をクリックし、レポートを実行します。

Coffee Sales for Midwest Region	
<u>Unit Sales</u>	<u>Dollar Sales</u>
332777	4178513

構文 フォームからの Servlet の呼び出し

```
<FORM ACTION="/ibi_apps/WFServlet" METHOD="get">
<INPUT NAME="IBIF_ex" VALUE="procedure" TYPE="hidden">
.
.
.
</FORM>
```

説明

`procedure`

実行するプロシジャ名です。呼び出しには、プロシジャに渡す HTTP 環境変数とアプリケーション変数の両方を含めることができます。

参照 フォームを作成する際の規則

- ❑ フォームを開始および終了するには、HTML <FORM タグを使用します。
- ❑ 実行するプロシジャ名を指定するには、HTML <INPUT タグを使用します。
- ❑ テキストボックス、チェックボックス、ドロップダウンリストなどの必要なコントロールをフォームに追加します。
- ❑ Servlet を実行し、フォームから収集した値をプロシジャに渡すための [実行] ボタンを追加します。

例 フォームからの Servlet の呼び出し

この例では、フォームを使用した起動ページを作成し、そのフォームから Servlet を呼び出して次の値を渡します。

- ❑ 実行するプロシジャの名前。
- ❑ プロシジャが要求する 2 つの値。このフォームは、最初の値を 1 行テキストボックスに入力し、2 つ目の値をラジオボタンから選択するようユーザに要求します。

注意：この例で作成するファイルの格納場所についての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

1. 前回の例で「CSALES」というプロシ ज्याを作成しなかった場合は、そのプロシ ज्याをここで作成します。このプロシ ज्याは、ユーザがリクエストした特定のカテゴリおよび地域においての売上数と売上高を示す動的レポートを生成します。

プロシ ज्या - CSALES.FEX

```
TABLE FILE GGSales
HEADING
"&CATEGORY Sales for &RGN Region"
SUM UNITS AND DOLLARS
WHERE (CATEGORY EQ '&CATEGORY') AND (REGION EQ '&RGN');
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

2. 「SALES.HTM」という名前の起動ページを作成します。このページ内のフォームが Servlet を呼び出して、「CSALES」というプロシ ज्या名を渡します。CATEGORY に対する値を 1 行テキストボックスに入力し、RGN に対する値をラジオボタンから選択するようユーザに要求します。

また、このフォームには [Run Report] という実行ボタンおよび [Clear Form] というリセットボタンも含まれています。ユーザ入力を要求する場合は、[実行] ボタンをフォームに追加する必要があります。

起動ページ - SALES.HTM

```
<HTML>
<HEAD>
<TITLE> WebFOCUS Report </TITLE>
</HEAD>
<BODY>
<H4 ALIGN=Center>Sales Report by Category and Region</H4>
<HR>
```

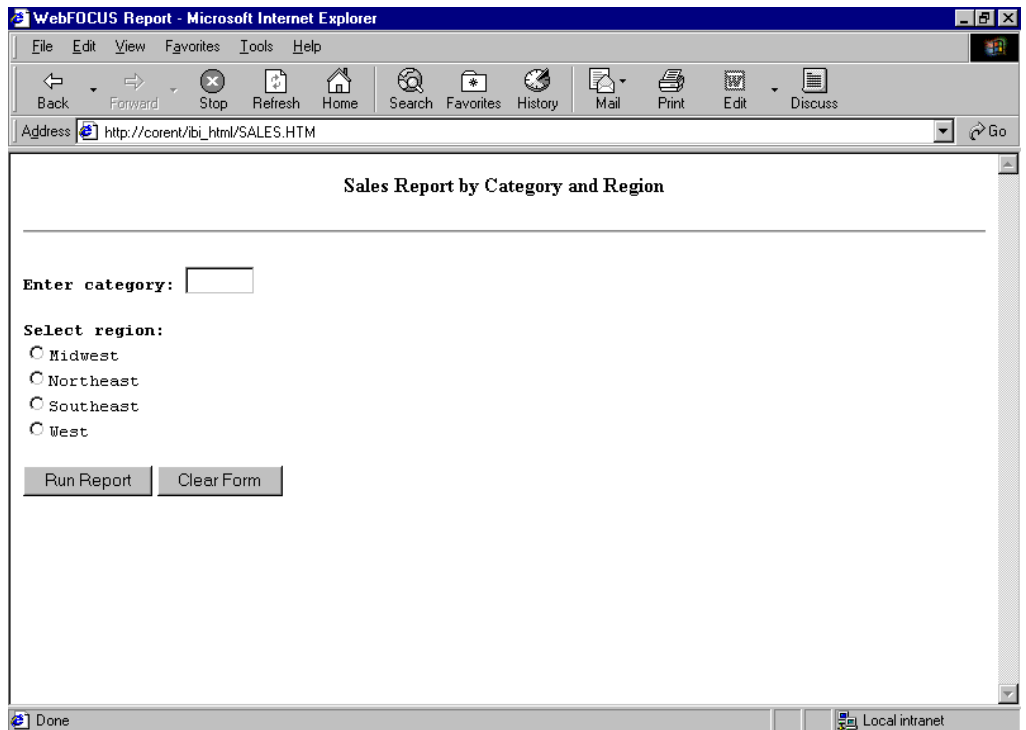
```

<FORM ACTION="/ibi_apps/WFServlet" METHOD="get">
<INPUT NAME="IBIF_ex" VALUE="csales" TYPE="hidden">
<P ALIGN=LEFT NOWRAP><PRE>
<B>Enter category: </B><INPUT NAME="CATEGORY" TYPE="text" SIZE="6">
</PRE></P>
<P><PRE><B>Select region: </B>
<INPUT NAME="RGN" TYPE=RADIO VALUE=Midwest>Midwest
<INPUT NAME="RGN" TYPE=RADIO VALUE=Northeast>Northeast
<INPUT NAME="RGN" TYPE=RADIO VALUE=Southeast>Southeast
<INPUT NAME="RGN" TYPE=RADIO VALUE=West>West
</PRE></P>
<P>
<INPUT NAME="submit" TYPE=SUBMIT VALUE="Run Report">
<INPUT NAME="reset" TYPE=RESET VALUE="Clear Form">
</P>
</FORM>

</BODY>
</HTML>

```

3. ブラウザで起動ページを実行します。次のような画面が表示されます。



4. [Enter category] テキストボックスに、次のように入力します。

Coffee

5. [Southeast] ラジオボタンを選択します。
6. [Run Report] をクリックし、レポートを実行します。

Coffee Sales for Southeast Region

<u>Unit Sales</u>	<u>Dollar Sales</u>
350948	4415408

動的複数選択ドロップダウンリストの使用

動的な複数選択ドロップダウンリストを使用すると、データソースから直接取得したフィールド値をそのドロップダウンリストの値として挿入することができます。データソースは、レコードの最大長が固定されていない状態で動的に割り当てられます。

手順 動的複数選択ドロップダウンリストを追加するには

この手順では、ダイアログマネージャコマンドの -HTMLFORM を使用して、ドロップダウンリストにデータソースの値を挿入します。この方法は、選択する値が頻繁に変化し、現在有効な値のリストを常時取得する必要がある場合に使用します。

-HTMLFORM についての詳細は、220 ページの「[レポート表示用 HTML ページの設計](#)」を参照してください。

1. ドロップダウンリストに値を挿入するプロシジャを作成します。

- a. 動的な値が格納されたテキストファイルを割り当てます。
たとえば、Windows でのコマンドは次のようになります。

```
FILEDEF textfile DISK APP/textfile.TXT
```

説明

`textfile`

値が格納されたファイルの名前です。

`APP`

APPROOT 下で、物理ファイルを格納するアプリケーションの名前です。

プラットフォームに固有のコマンドについての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

- b. DEFINE コマンドを追加して、値を識別するための一時項目 (DEFINE) を作成します。
- c. 次のコマンドを追加して、割り当てたファイルに値を保存します。

```
ON TABLE HOLD FORMAT ALPHA as textfile
```

説明

`textfile`

ファイルの名前です。ファイル名として許容される長さは、1 から 8 バイトです。

- d. 次のコマンドを追加して、レポート出力を起動ページに送信して値を表示します。

```
-HTMLFORM valuespg
```

説明

`valuespg`

起動ページの名前です。

2. 起動ページを作成します。コメントを使用して、値の表示場所を指示します。

```
!IBI.FIL.textfile;
```

または

```
<!--WEBFOCUS TABLE textfile-->
```

説明

textfile

値が格納されたファイルの名前です。

注意：HTML コメント行は「->」で閉じる必要があります。また、単一の「>」で閉じることができますが、その場合はコメント行に他の HTML タグを含めることはできません。

テキストファイルはメモリ内で動的に割り当てられ、最大長は固定されていません。

WebFOCUS Reporting Server が起動ページの場所を特定できるように、APP PATH または EDAPATH を設定しておく必要があります。検索パスについての詳細は、134 ページの「[検索パスロジックの選択](#)」を参照してください。

3. 複数の値を受容して動的レポートを生成するプロシジャを作成します。

例

動的複数選択ドロップダウンリストの追加

この例は、WebFOCUS を Windows で実行する場合の手順です。他のプラットフォームを使用する場合は、FILEDEF コマンドをプラットフォームに対応したコマンドに読み替えてください。FILEDEF コマンド、およびこの例で作成するファイルの格納場所についての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

1. 「DYNAMMUL」という名前のプロシジャを作成します。このプロシジャは、SHORT データソースの COUNTRY フィールドから取得した値をドロップダウンリストに挿入します。

プロシジャ - DYNAMMUL.FEX

```
FILEDEF DYNAMLST DISK BASEAPP/DYNAMLST.TXT
DEFINE FILE SHORT
OPTCOUNTRY/A40 = '<option>'|COUNTRY;
END
TABLE FILE SHORT
SUM OPTCOUNTRY
BY COUNTRY NOPRINT
ON TABLE HOLD FORMAT ALPHA AS DYNAMLST
END
-RUN
-HTMLFORM DYNAMIC2
-RUN
```

2. 「DYNAMIC2.HTM」という名前の起動ページを作成します。WebFOCUS Reporting Server が起動ページの場所を特定できるように、APP PATH または EDAPATH を設定しておく必要があります。検索パスについての詳細は、23 ページの「[TIBCO WebFOCUS のアプリケーションロジック](#)」を参照してください。

このサンプル起動ページは Servlet を使用します。

起動ページ - DYNAMIC2.HTM

```
<HTML>
<TITLE> DYNAMIC DROP-DOWN LIST REPORT </TITLE>
<H4> PROJECTED RETURN BY COUNTRY </H4>
<FORM ACTION="/ibi_apps/WFServlet" METHOD="GET">
<INPUT TYPE="HIDDEN" NAME="IBIF_ex" VALUE="MULRPT">
<SELECT NAME="COUNTRY" SIZE="3" MULTIPLE>
<!--WEBFOCUS TABLE DYNAMLST-->
</SELECT>
<BR><BR><INPUT TYPE="SUBMIT" VALUE="RUN REPORT!"></FORM>
</BODY>
</HTML>
```

注意：HTML コメント行は「->」で閉じる必要があります。また、単一の「>」で閉じることもできますが、その場合はコメント行に他の HTML タグを含めることはできません。

3. 「MULRPT」という名前のプロシジャを作成します。このプロシジャは、選択した 1 つまたは複数の国の Projected Return を表示します。国名を 1 つも選択しなかった場合は、データソースに存在するすべての国の Projected Return が表示されます。

プロシジャ - MULRPT.FEX

```
-SET &COUNTER=1;
TABLE FILE SHORT
SUM PROJECTED_RETURN
BY COUNTRY
-IF &COUNTRY.EXISTS THEN GOTO LOOP1 ELSE GOTO DONE;
-LOOP1
WHERE COUNTRY EQ '&COUNTRY'
-IF &COUNTRY0.EXISTS NE 1 THEN GOTO OUTLOOP;
-REPEAT OUTLOOP FOR &COUNTER FROM 2 TO &COUNTRY0;
OR '&COUNTRY.&COUNTER'
-OUTLOOP
-DONE
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

注意：実行時には、複数選択ドロップダウンリストで選択した国名数が &COUNTRY0 に格納されます。たとえば、国名を 3 つ選択した場合、&COUNTRY0 は 3 に設定され、&COUNTRY および &COUNTRY1 は最初に選択した国、&COUNTRY2 は 2 番目に選択した国、&COUNTRY3 は 3 番目に選択した国にそれぞれ設定されます。

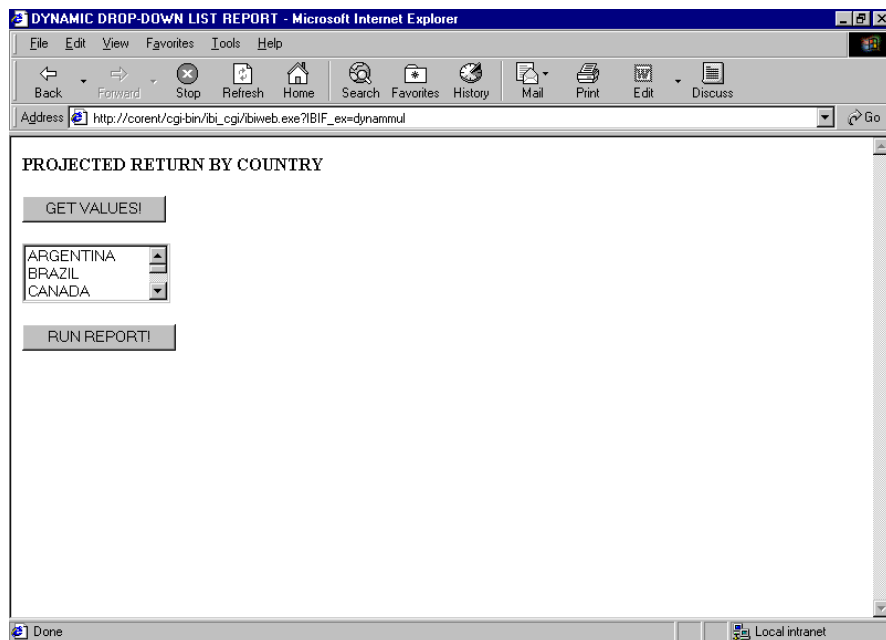
この例では、WebFOCUS 環境でオートプロンプト機能が有効化されていない場合を示しています (WebFOCUS Client 構成変数 IBIF_wfdescribe=OFF)。セルフサービス構成パラメータ (IBI_WFDESCRIBE_DEFAULT) についての詳細は、『TIBCO WebFOCUS セキュリティ管理ガイド』を参照してください。

オートプロンプト機能を有効にした環境では、次のパラメータをフォームに追加して、このフォームから実行されるリクエストのオートプロンプト機能をオフにすることができます。

```
<INPUT TYPE="HIDDEN" NAME="IBIF_wfdescribe" VALUE="OFF">  
add the above line to example file DYNAMIC2.HTM after line  
<FORM ACTION="/ibi_apps/WFServlet" METHOD="GET">
```

4. プロシジャの DYNAMMUL を実行します。起動ページが開いたときに、JavaScript を使用してプロシジャを実行することもできます。

下図のようなページが表示されます。



5. [CANADA]、[HONG KONG]、[MEXICO] を選択します。[RUN REPORT!] をクリックし、レポートを実行します。

<u>Country</u>	<u>Projected Annualized Return</u>
CANADA	3.990
HONG KONG	4.550
MEXICO	5.040

adhoc レポートの有効化

ユーザは、テキスト領域に adhoc レポートのリクエストを入力することができます。このリクエストには、WebFOCUS レポートコマンド (例、TABLE FILE filename) およびダイアログマネージャのコマンドと変数を含めることができます。ユーザがこのリクエストを送信すると、テキスト領域に使用した変数にリクエスト全体が割り当てられて、WebFOCUS に渡されます。

警告： adhoc レポートを有効にする前に、予測されるすべての結果を検討しておく必要があります。たとえば、ユーザの不注意で大量のリソースを消費するリクエストを入力したため、すべてのユーザのレスポンス時間が低下するという問題が発生する場合があります。

デフォルト設定で次の場所に格納されている IBIDIR.WFS ファイルで、adhoc レポートをサイトごとに無効にすることができます。

Windows `install_drive:\ibi\WebFOCUS82\client\wfc\etc`

UNIX `/ibi/WebFOCUS82/client/wfc/etc`

次の行がコメントとして設定されている場合は (デフォルト)、adhoc レポートを実行することができます。

```
# <SET> IBIF_adhocfex(protect)
```

この行のコメントが解除されている場合は、adhoc レポートを実行することはできません。コメントが解除されている場合、IBIF_adhocfex 変数をブラウザから WebFOCUS Client に渡すことが禁止されます。

この変数は内部的に生成されます。BI Portal では、この変数がすべてのプロシジャに使用されます。

構文 adhoc レポートの有効化

次のように、オプションの属性を構文に追加することができます。

```
<TEXTAREA NAME="IBIF_adhocfex" VALUE="value" ROWS=rows COLS=cols>
</TEXTAREA>
```

説明

value

テキスト領域に表示するデフォルトのリクエストです。

VALUE=""

rows

テキスト領域の行数です。

cols

テキスト領域の列数です。

例 adhoc レポートの有効化

注意：この例で作成するファイルの格納場所についての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

1. 「ADHOC」という名前の起動ページを作成します。このページには、レポートリクエストの入力を要求するテキスト領域が含まれています。このページが Web サーバにアクセスできる状態にしておく必要があります。

このサンプル起動ページは Servlet を使用します。

起動ページ - ADHOC.HTM

```
<HTML>
<HEAD>
<TITLE> WebFOCUS Report </TITLE>
</HEAD>
<BODY>
<H4>Enter report request:</H4>
```

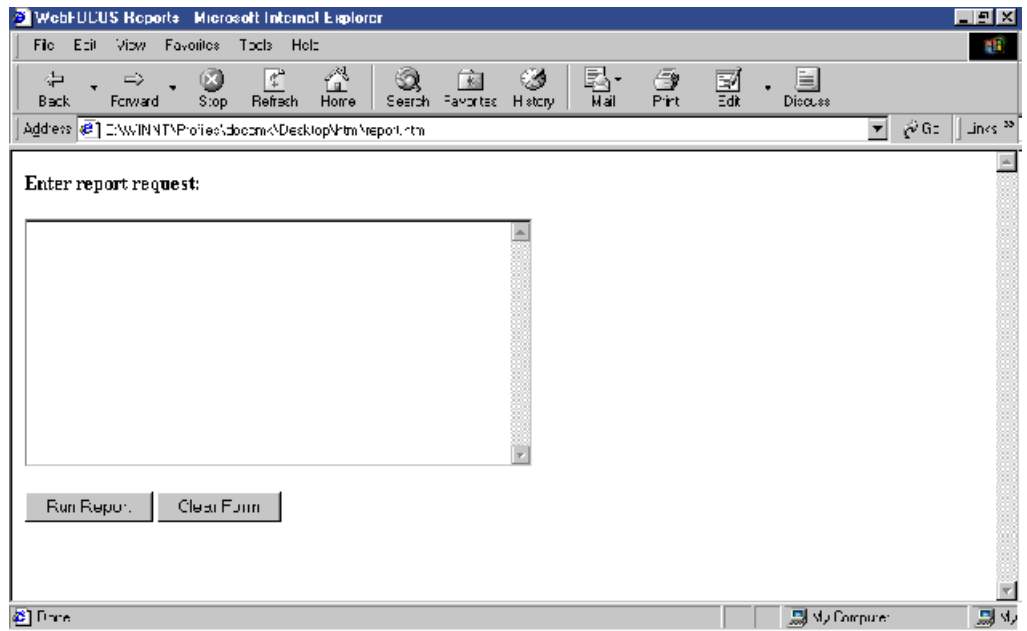
```

<FORM METHOD="get" ACTION="/ibi_apps/WFServlet">
<P ALIGN=LEFT NOWRAP><PRE>
<TEXTAREA NAME="IBIF_adhocfex" VALUE="" ROWS=12 COLS=48 ALIGN=LEFT>
</TEXTAREA>
</PRE></P>
<P>
<INPUT NAME="submit" TYPE=SUBMIT VALUE="Run Report">
<INPUT NAME="reset" TYPE=RESET VALUE="Clear Form">
</P>
</FORM>

</BODY>
</HTML>

```

2. 起動ページを実行します。実行後、次のように表示されます。



3. テキスト領域に次のリクエストを入力します。

```

TABLE FILE CENTORD
SUM QUANTITY BY HIGHEST 1 ORDER_DATE
BY PRODNAME
END

```

4. [Run Report] をクリックし、レポートを実行します。

PAGE 1		
Date Of Order:	Product Name:	Quantity:
2000/12/30	110 VHS-C Camcorder 20 X	1
	120 VHS-C Camcorder 40 X	1
	150 8MM Camcorder 20 X	2
	2 Hd VCR LCD Menu	1
	250 8MM Camcorder 40 X	1
	650DL Digital Camcorder 150 X	1
	AR2 35MM Camera 8 X	1
	Combo Player - 4 Hd VCR + DVD	2
	DVD Upgrade Unit for Cent. VCR	2
	QX Portable CD Player	1
	R5 Micro Digital Tape Recorder	2
	ZC Digital PDA - Standard	1
	ZT Digital PDA - Commercial	2

JavaScript によるフォームの確認

JavaScript は、オブジェクト指向のインタープリタ言語で、HTML と併用して静的な Web ページに動的な機能を追加します。

HTML ページに JavaScript 関数を使用すると、その関数は Web ブラウザで処理され、実行されます。WebFOCUS を呼び出す HTML ページまたはレポートの表示ページに JavaScript 関数を組み込むと、次のことが実行可能になります。

- ❑ フォームに入力した値を、WebFOCUS に送信する前に確認できる。入力値を確認することにより、制限されたデータや使用不可のデータがリクエストされることを防止します。ここでは、ユーザがフォームに入力した値を確認するための JavaScript 関数のコーディング方法について説明します。

- ❑ WebFOCUS レポートで、返されたデータの演算を実行できる。JavaScript は、WebFOCUS Reporting Server から Web ブラウザに至るまでのデータ処理要求の負荷を軽減します。この機能を説明するドリルダウンレポートの例については、255 ページの「[ユーザインタフェースの機能強化](#)」を参照してください。

JavaScript を使用すると、設計機能をさらに拡張することができます。JavaScript の使用方法および構文についての詳細は、JavaScript のマニュアルを参照してください。

例

フォームのユーザ入力値の確認

注意：この例で作成するファイルの格納場所についての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

1. 「VALIDATE」という名前の起動ページを作成します。このページは、「checkInput」という JavaScript 関数を呼び出します。このプロシジャが Web サーバにアクセスできる状態にしておく必要があります。

このサンプル起動ページは Servlet を使用します。

起動ページ - VALIDATE.HTM

```
<HTML>
<HEAD>
<TITLE> WebFOCUS Report </TITLE>

<SCRIPT LANGUAGE=JAVASCRIPT TYPE="TEXT/JAVASCRIPT">
function checkInput() {
    if(document.form2.STATE.value == "") {
        alert("You must enter a value for the state!");
        return false;
    }
    else {
        return true;
    }
    document.form2.submit();
}
</SCRIPT>
```

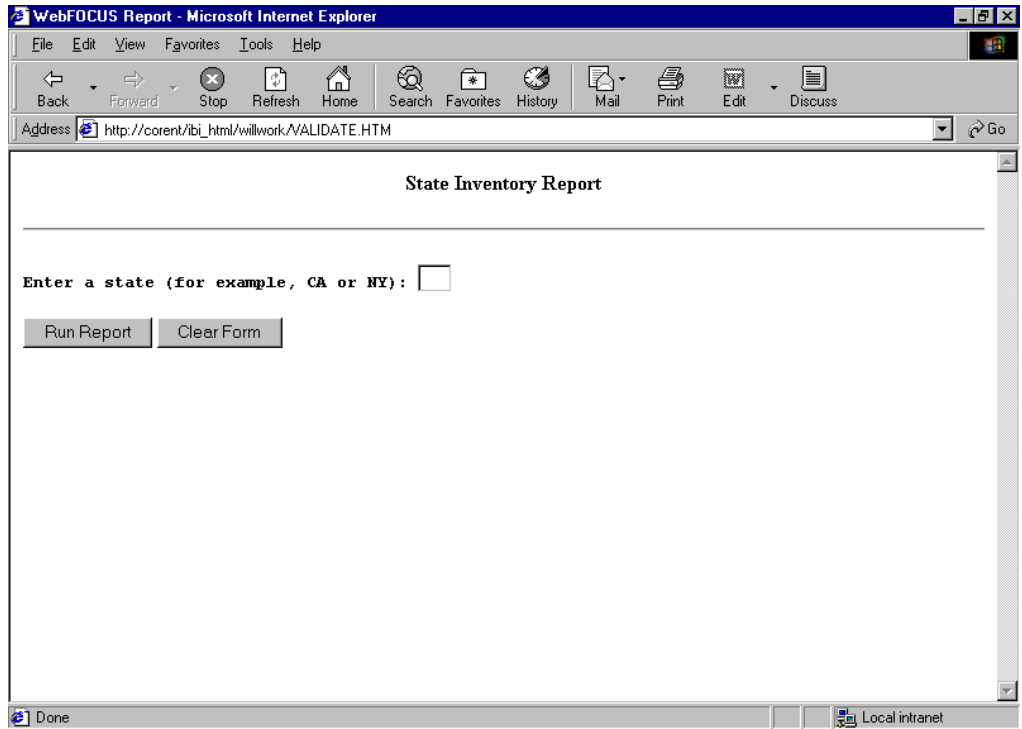
```
</HEAD>
<BODY>
<H4 ALIGN=CENTER>State Inventory Report</H4>
<HR>
<FORM ACTION="/ibi_apps/WFServlet" METHOD="get"
TARGET="_blank" NAME="form2" onSubmit="return checkInput()">
<INPUT NAME="IBIF_ex" VALUE="stsales" TYPE="hidden">
<P ALIGN=LEFT NOWRAP><PRE>
<B>Enter a state (for example, CA or NY): </B><INPUT NAME="STATE"
TYPE="text" SIZE="2">
</PRE></P>
<P><PRE></PRE></P>
<P>
<INPUT NAME="submit" TYPE=SUBMIT VALUE="Run Report">
<INPUT NAME="reset" TYPE=RESET VALUE="Clear Form">
</P>
</FORM>
</BODY>
</HTML>
```

2. 「STSALES」という名前のプロシジャを作成します。このプロシジャは、ユーザがリクエストした州 (State) においての eMart の在庫レポートを作成します。

プロシジャ - STSALES.FEX

```
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY STORENAME BY PRODNAME
WHERE STATE EQ '&STATE'
WHERE STORENAME EQ 'eMart'
END
```

3. 起動ページを実行します。



4. 州 (State) の値を入力せずに、[Run Report] をクリックします。次のメッセージが表示されます。



WebFOCUS のオートテンプレート機能

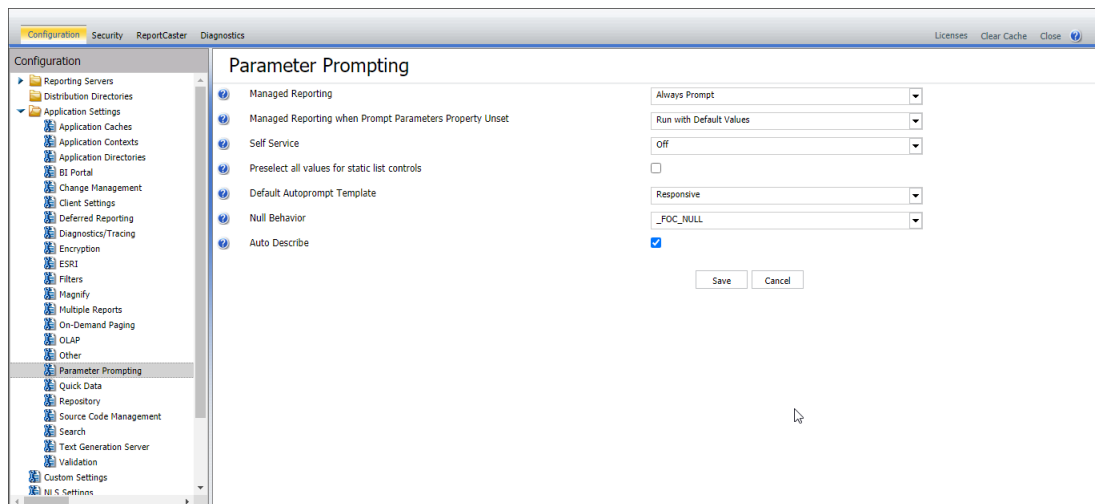
WebFOCUS のオートプロンプト機能を使用することで、プロシジャの実行に必要なパラメータ (変数) 値の入力をユーザに要求するフォームが動的に作成されます。ユーザがオートプロンプトフォームで入力または選択するパラメータ値は、レポートリクエストのフィールド値として使用することも、表示コマンドおよびソートコマンドのオブジェクトとして使用することもできます。

オートプロンプトのユーザインターフェーステンプレートには、次の 2 つがあります。

- ❑ レスポンシブオートプロンプト 詳細は、159 ページの「[レスポンシブオートプロンプト](#)」を参照してください。
- ❑ HTML オートテンプレート 詳細は、172 ページの「[HTML オートプロンプト](#)」を参照してください。

オートプロンプトの構成

オートプロンプト機能を構成するには、WebFOCUS 管理コンソールの [構成] タブの [パラメータのプロンプト] 設定を使用します。下図のように、[デフォルトオートプロンプトテンプレート] のデフォルト設定は [デザイナ] です。バージョン 8.1 の HTML オートプロンプトテンプレートも使用できます。その場合は、[HTML_TOP] または [HTML_TOP_CHECKED] 設定を選択して構成します。



[パラメータのプロンプト] 設定では、WebFOCUS Client でのパラメータプロンプトの動作を指定します。

Managed Reporting (IBIMR_PROMPTING)

WebFOCUS の BI Portal 内のリクエストを対象に、パラメータのプロンプトを有効にするかどうかを指定します。利用可能な値には、次のものがあります。

- ❑ **オフ** サイトレベルでのパラメータのプロンプトをオフにします。
- ❑ **デフォルト値で実行 (XMLRUN)** - DEFAULT コマンドで作成された変数、および値を持たない他の変数の入力进行要求します。
- ❑ **常にプロンプトを表示 (XMLPROMPT)** 値が割り当てられていない変数がほかにある場合、- DEFAULT コマンドで作成された変数の入力进行要求します。これがデフォルト値です。

[パラメータのプロンプト] オフ時の動作 (IBIMR_PROMPTINGUNSET)

[Managed Reporting] (IBIMR_PROMPTING) が [常にプロンプトを表示] (XMLPROMPT) または [デフォルト値で実行] (XMLRUN) に設定されている場合、およびプロシジャの [プロパティ] ダイアログボックスで [パラメータのプロンプト] のチェックがオフの場合に、BI Portal 内のプロシジャ (FEX) のパラメータプロンプトを有効または無効にします。利用可能な値には、次のものがあります。

- ❑ **オフ** パラメータのプロンプトを無効にします。
- ❑ **デフォルト値で実行 (XMLRUN)** - DEFAULT コマンドで作成された変数、および値を持たない他の変数の入力进行要求します。これがデフォルト値です。
- ❑ **常にプロンプトを表示 (XMLPROMPT)** 値が割り当てられていない変数がほかにある場合、- DEFAULT コマンドで作成された変数の入力进行要求します。

セルフサービス (IBI_WFDESCRIBE_DEFAULT)

セルフサービスレポートのオートプロンプトを有効または無効にします。利用可能な値には、次のものがあります。

- ❑ **オフ** オートプロンプトを無効にします。これがデフォルト値です。
- ❑ **デフォルト値で実行 (XMLRUN)** - DEFAULT コマンドで作成された変数、および値を持たない他の変数の入力进行要求します。
- ❑ **常にプロンプトを表示 (XMLPROMPT)** 値が割り当てられていないために要求される変数がほかにある場合、- DEFAULT コマンドで作成された変数の入力のみ进行要求します。
- ❑ **XML の表示 (構文エラー確認付きデバッグ) (XML)** 変数が記述された XML ドキュメントをブラウザに表示します。この設定は内部的に使用され、デバッグおよび構文エラー確認用としてのみ使用することをお勧めします。

- ❑ **XML の表示 (デバッグ) (XMLCHECK)** 変数が記述された XML ドキュメントをブラウザに表示します。この設定は内部的に使用され、デバッグ用としてのみ使用することをお勧めします。

注意：BI Portal では、IBIMR_prompting という別の変数設定が使用されます。

静的リストコントロールですべての値を事前選択 (IBI_FOCALL_DEFAULT)

実行時のレスポンスプロンプトの複数選択静的リストパラメータで、すべての値の自動選択をデフォルト設定で有効にします。

このチェックがオン (TRUE) の場合、複数選択静的リストを含むレスポンスプロンプトの選択リストパラメータには、デフォルト設定で、初期選択値として FOC_ALL が自動的に割り当てられます。これにより、これらのリストには、実行時の選択値としてすべての値が自動的に表示されます。ただし、複数選択静的リストに表示するデフォルト値が指定されている場合は、初期選択値としての FOC_ALL の自動割り当てがこの値で上書きされ、デフォルト値が初期選択値として表示されます。

このチェックがオフ (FALSE) の場合、複数選択静的リストを含むパラメータには、デフォルト設定で、初期選択値として FOC_ALL が自動的に割り当てられません。この場合、リスト内のすべての項目は、実行時に自動的に選択されません。複数選択静的リストに指定されたデフォルト値が含まれる場合、この値がデフォルト設定で表示されます。静的選択リストのすべての値をクエリに含めるには、ユーザは手動で [すべての値] オプションを選択するか、リスト内のすべての値を個別に選択する必要があります。

デフォルト設定では、このチェックはオフ (False) になっています。

デフォルトオートプロンプトテンプレート (IBI_DESCRIBE_TEMPLATES)

オートプロンプトインターフェースのレイアウトを定義するテンプレートを指定します。利用可能な値には、次のものがあります。

- ❑ **デザイナー** デザイナオートプロンプト実装およびデザイナーオートプロンプトテンプレートの使用を指定します。このテンプレートでは、オートテンプレートのインターフェースに Designer のページフォーマットが使用されます。
- ❑ **レスポンス** レスポンス jQuery ベースの実装および autoprompt.jsp テンプレートを使用します。これがデフォルト値です。
- ❑ **HTML_Top** HTML ベースの実装および autoprompt_top.html テンプレートを使用します。このテンプレートでは、パラメータがページ上部に横方向に表示されます。
- ❑ **HTML_Top_Checked** HTML ベースの実装および autoprompt_top_checked.html テンプレートを使用します。このテンプレートでは、新規ウィンドウのチェックボックスで [Run] が事前選択され、デフォルト設定ですべてのレポートが新規ウィンドウで開きます。

[選択なし] の動作 (IBIF_DESCRIBE_NULL)

動的複数選択リストで [すべての値] のオプションが選択された場合に、WebFOCUS Client が SET コマンドで変数に割り当てる値 (_FOC_NULL または FOC_NONE) を指定します。デフォルト値は _FOC_NULL です。_FOC_NULL についての詳細は、193 ページの「[_FOC_NULL の内部処理](#)」を参照してください。FOC_NONE についての詳細は、194 ページの「[FOC_NONE の内部処理](#)」を参照してください。

自動記述 (IBI_AUTODESCRIBE)

レポートおよびグラフのパラメータの自動インデックス機能を有効にします。このチェックをオン (TRUE) にすると、ユーザがレポートまたはグラフを保存する際に、基になるパラメータに自動的にインデックスが追加されます。これにより、これらのパラメータに関する情報が即座に検索可能になるため、よりタイムリーで包括的な検索結果が提供されます。

このチェックは、デフォルトでオン (TRUE) に設定されています。

レスポンスオートプロンプト

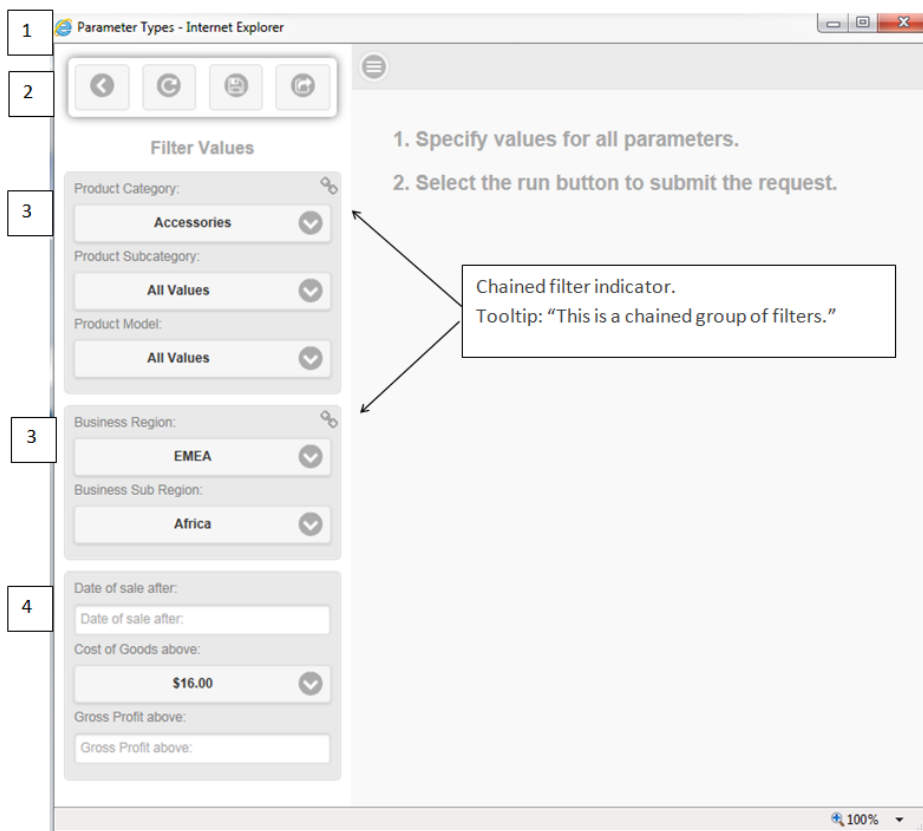
レスポンスオートプロンプト機能により、レスポンスモバイルがサポートされる斬新なユーザインターフェースデザイン、ディメンション階層フィールドの動的リストの連鎖、および YYMD フォーマット日付フィールドの日付選択フィルタでのカレンダーコントロールの使用が可能です。この日付フォーマットは、すべての構成要素 (年、月、日) が含まれたフォーマットにする必要があり、日付時間フィールドフォーマットには適用されません。

注意

- ❑ レスポンスオートプロンプトは、オートプロンプト機能のデフォルト設定です。管理コンソールを使用した、オートプロンプト機能の構成方法についての詳細は、156 ページの「[オートプロンプトの構成](#)」を参照してください。
- ❑ レスポンスオートプロンプト機能は、セクション 508 アクセシビリティ要件をサポートしていません。

レスポンスオートプロンプトページのコンポーネント

下図のように、レスポンスオートプロンプトページでは、左側にフィルタが縦方向に表示され、右側にレポート出力が表示されます。



1. ウィンドウタイトルは、ツリーまたはリポジトリからプロシジャコードを取得するリクエストからレポートおよびグラフを実行した場合に、ツリーに表示されるタイトルです。リクエストを InfoAssist 内で実行した場合、InfoAssist のレポート出力タブの末尾に数字が追加されます。リクエストが保存済みでなく、ツールまたはテキストエディタから実行された場合、このタイトルは [AdHocFex] になります。
2. オプションバーを使用して、ユーザは次のことを行えます。

- ☐  フィルタパネルを閉じる (非表示にする)。
- ☐  フィルタ値を、ページの初期画面に表示された値にリセットする。

- ❑  フィルタ値を保存する。このオプションは、リクエストがリポジトリから直接プロシジャを実行中で、ユーザに保存済みパラメータレポートの作成が許可されている場合に、使用することができます。このオプションは、リクエストが InfoAssist、テキストエディタ、または App Studio レポートキャンバスから実行される場合は表示されません。これは、リクエストコードがそれらのツールから送られ、リポジトリに保存されていない可能性があるためです。

注意

- ❑ ファイルのタイトル値の最大長は 256 バイトです。
- ❑ ファイルのタイトルがフォルダ内の既存ファイルと同一名の場合、既存ファイルを置き換えるかどうかのメッセージが表示されます。
- ❑ フィルタ値の説明の最大長は 255 バイトです。この制限を超過するテキストは、レポートの保存時に切り捨てられます。
- ❑  フィルタ値付き実行 このオプションは、選択または入力された値でリクエストを実行します。レポート出力が完全表示されるように、フィルタパネルを開じます。これは小さいデバイスの使用時に重要です。

3. 非連鎖グループでは、グループの右上にイメージが表示されません。

レスポンスオートプロンプトページで使用するリクエストコードについては、170 ページの「[レスポンスサンプルコード \(Parameter_Type.fex\)](#)」を参照してください。

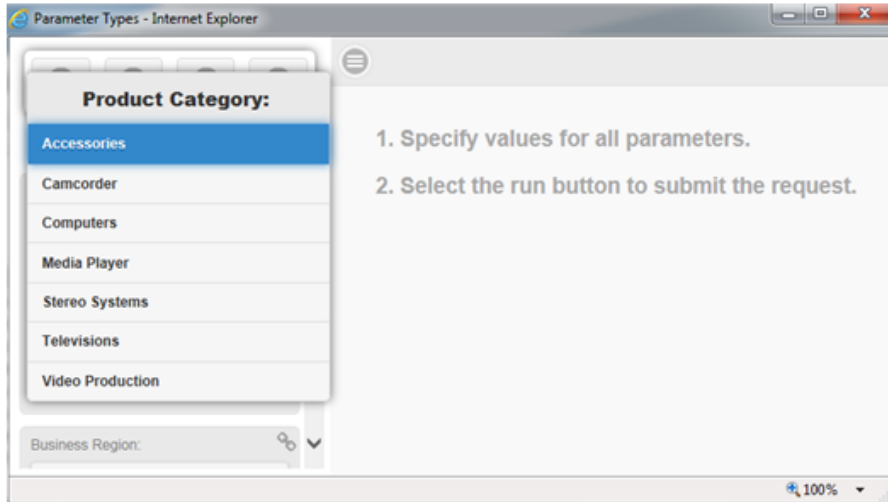
フィルタパネルは、レポート出力パネルの左上にあるフィルタパネルの [表示] イメージ  を選択することで、非表示 (閉じる) と表示を切り替えることができます。

選択リスト

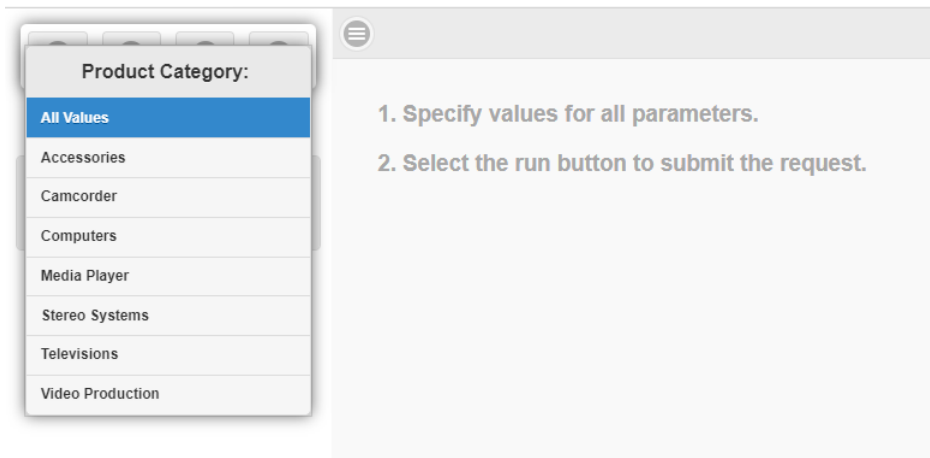
パラメータフィールドを選択すると、選択リストが表示されます。選択リストの表示は、選択リストの値の数および使用するデバイスのタイプによって異なります。選択リスト左上のオプションまたは選択リスト外側の任意の場所をクリックすると、選択リストが閉じます。

デバイスのサイズにもよりますが、値の数が通常 7 個以下の場合、選択リストはフィルタパネル上に重ねて表示されます。

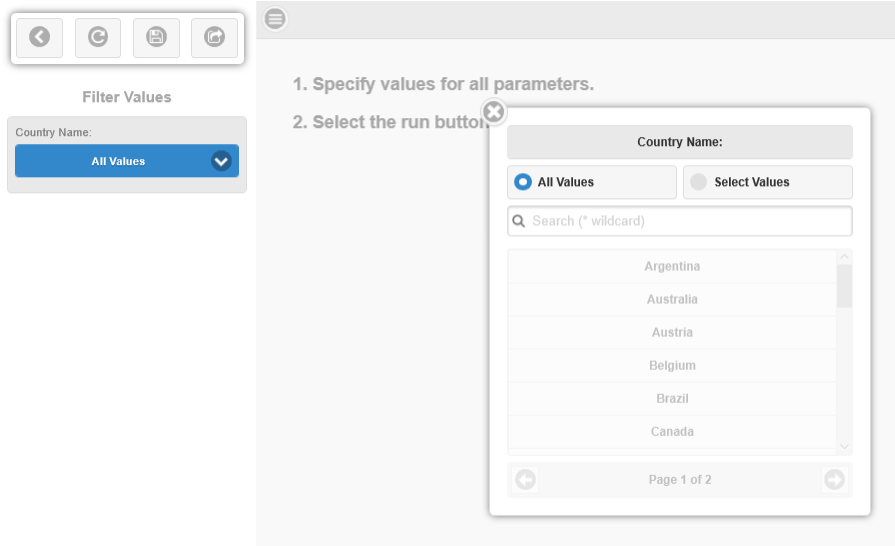
下図のように、単一選択リストでは、パラメータにデフォルト値が設定されていない場合、リストの最初の値がデフォルト値になります。



`_FOC_NULL` をデフォルト値として割り当てることで、フィルタをオプションとして識別した場合、下図のように、リストの最初の値は [すべての値] になり、デフォルト設定で、この値が選択済みになります。

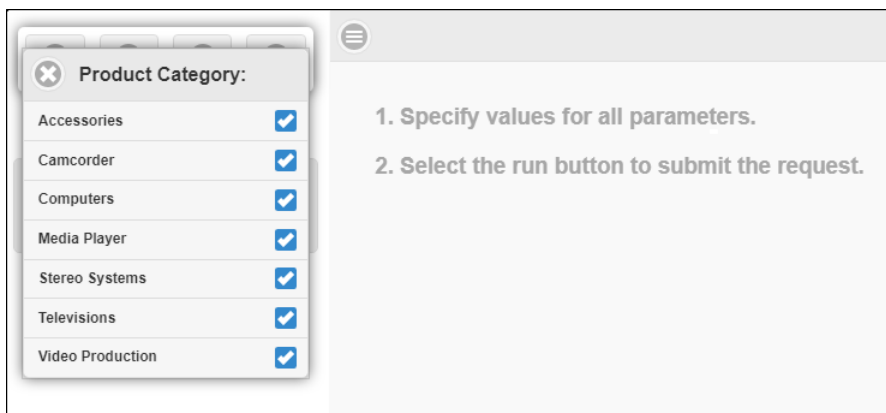


下図のように、大規模な選択リストはレポート出力パネルに重ねて表示されます。



_FOC_NULL が選択リストのデフォルト値として定義される場合、リストのサイズに関わらず、[すべての値] オプションは、デフォルトの選択として表示されます。

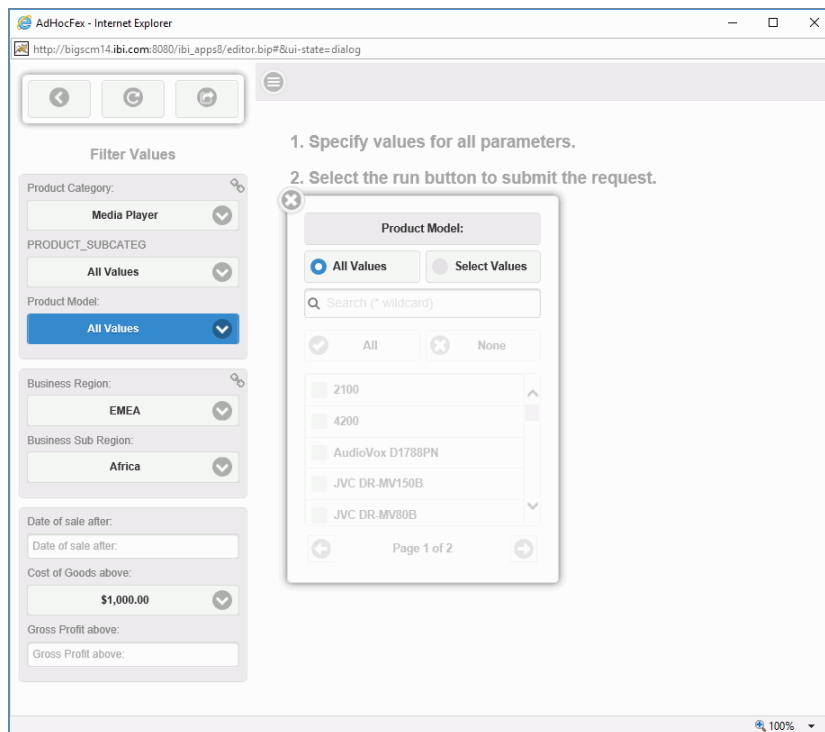
使用中の WebFOCUS のインストールで [静的リストコントロールですべての値を事前選択] (IBI_FOCALL_DEFAULT) 設定が有効の場合 (管理コンソールの [構成] タブの [パラメータのプロンプト] ページで設定)、別の値がデフォルトの表示値として特定されない限り、静的複数選択リストでは、すべての値がデフォルト設定で選択されます。



値の選択ダイアログボックスのその他のオプションの外観は、パラメータが参照するフィルタタイプによって異なります。

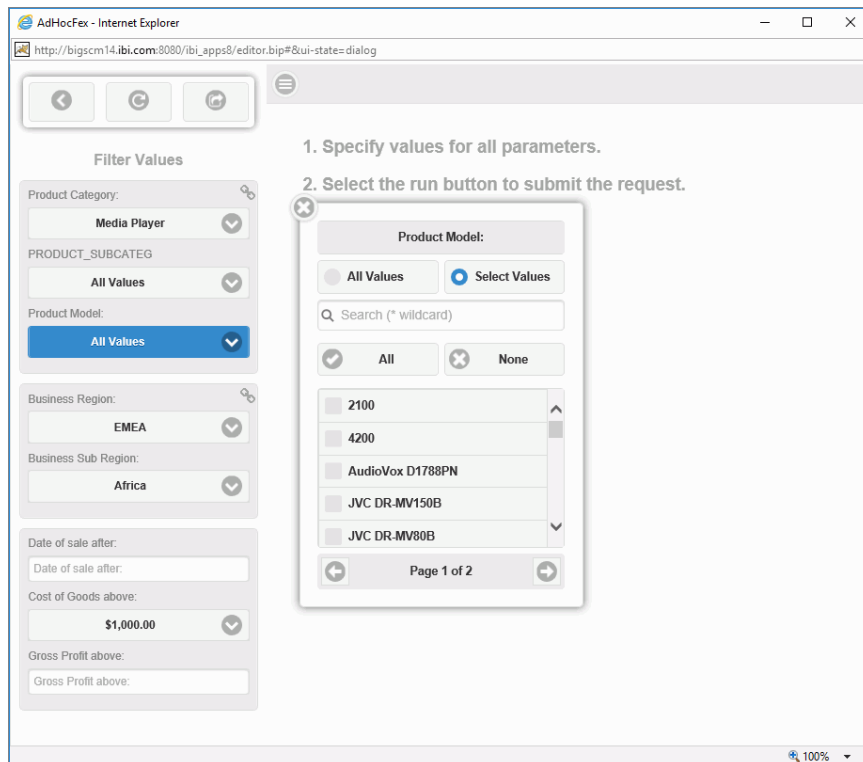
- ❑ フィルタが静的または動的複数選択リストの場合、[値の選択] オプションは、[すべての値] オプションとともに使用可能です。このオプションは、静的単一選択リストで、リスト項目の表示が単一画面に収まらない場合にも使用できます。
- ❑ また、[すべて] および [なし] オプションは、動的または静的複数選択リストのフィルタで使用できます。

下図は、パラメータのフィルタタイプが動的複数選択リストで、パラメータにデフォルト値が設定されていない場合の値の選択ダイアログボックスオプションを示しています。動的複数選択フィルタで、パラメータにデフォルト値が設定されていない場合、デフォルト値は [すべての値] になります。[すべての値] を選択した場合、フィルタは適用されず、フィールドのすべての値がレポートまたはグラフに含まれます。

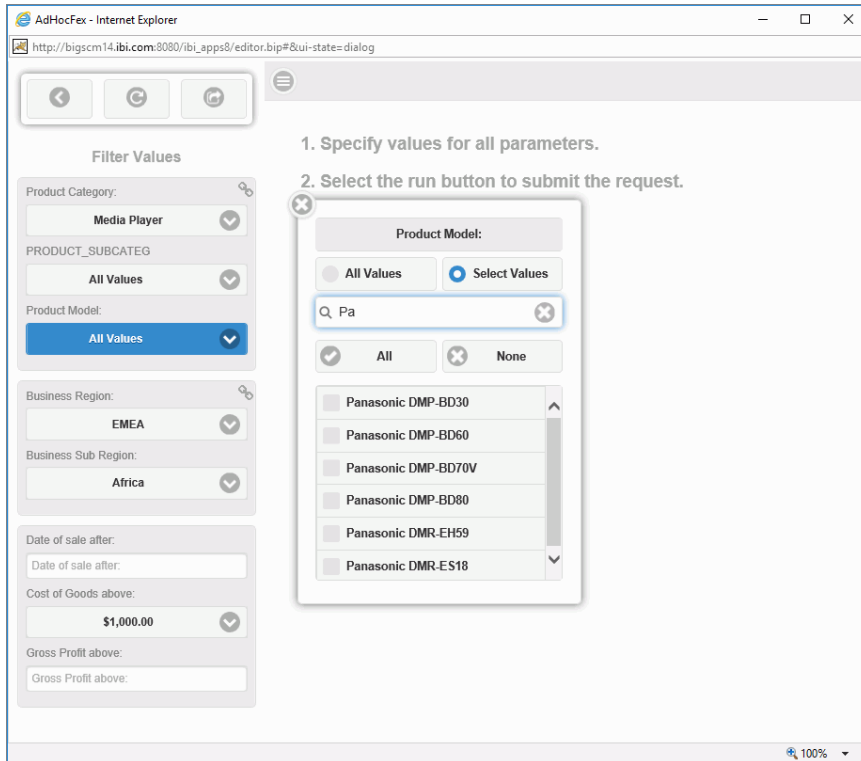


レスポンシブオートプロンプトで表示される動的リストの値は、大文字と小文字を区別したソート順で整列されます。HTML オートプロンプトで表示される動的リストの値は、WebFOCUS Reporting Server によって返されるソート順で整列されます。これは、ホストマシンのオペレーティングシステムによって決定されます。

[値の選択] オプションを選択すると、[検索]、[すべて]、[なし]、個々の値、およびページコントロールが選択可能になります。ページコントロールは、値の数が 25 個以上の場合に使用できます。



[検索] オプションは、文字を入力してリストの値にフィルタを設定します。[検索] は、選択リストに表示される値に適用されます。



[すべて] を選択すると、選択リスト内のすべての値のチェックがオンになります。検索が適用された場合は、検索結果に表示された値のみチェックがオンになります。[すべて] を選択すると、個々の値のチェックはオフになります。

[なし] を選択すると、すべての値のチェックがオフになります。

注意：値に単一選択フィルタが設定され、フィルタを選択して値の選択ダイアログボックスが表示された場合、値のリストには指定した値が選択 (強調表示) されていない状態で表示されます。オートプロンプトページの左側のフィルタパネルで、パラメータフィールド上にカーソルを移動すると、パラメータに割り当てられた値を参照することができます。

単純なフィルタ

単純なフィルタを追加して、ユーザーに変数の値の入力を求めることができます。変数がデフォルト値の `_FOC_NULL` に割り当てられ、単純なフィルタで 사용되는場合、オートプロンプトの単純なフィルタコントロールでデフォルト値が [すべての値] に設定され、デフォルト設定で [すべて] オプションのチェックがオンになります。値を入力すると、[すべて] オプションのチェックがオフになります。変数にフィルタが設定されるフィールドの値をすべて含めるように指定する場合は、[すべて] オプションを選択します。パラメータフィールドに「All Values」と入力すると、「All Values」という値でフィルタが適用され、フィールドのすべての値が返されません。

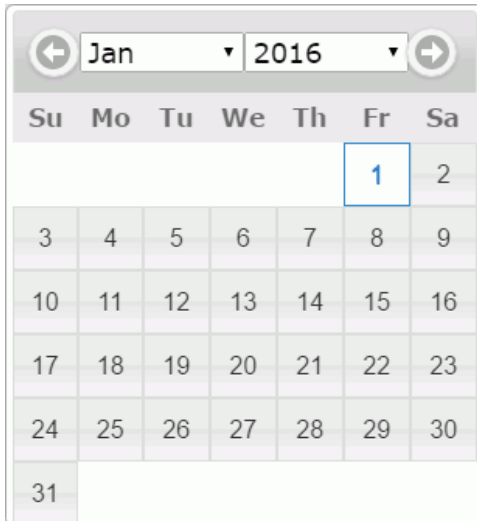
以下はその例です。

```
-DEFAULT &GROSS_PROFIT_US = _FOC_NULL;
TABLE FILE wfretail82/wf_retail
HEADING CENTER
"Product Models with Gross Profit is below &GROSS_PROFIT_US"
SUM WF_RETAIL.WF_RETAIL_SALES.GROSS_PROFIT_US
BY WF_RETAIL.WF_RETAIL_PRODUCT.PRODUCT_CATEGORY
BY WF_RETAIL.WF_RETAIL_PRODUCT.PRODUCT_SUBCATEG
BY WF_RETAIL.WF_RETAIL_PRODUCT.MODEL
WHERE WF_RETAIL.WF_RETAIL_SALES.GROSS_PROFIT_US LT &GROSS_PROFIT_US.(|
FORMAT=D20.2M).Gross Profit below:.QUOTEDSTRING;
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/javaassist/intl/EN/combine_templates/
ENWarm.sty,$
ENDSTYLE
END
```

The screenshot shows a user interface for a WebFOCUS application. On the left, there is a 'Filter Values' section with a 'Gross Profit below:' label. Below this label is a text input field containing 'All Values' and a checked checkbox labeled 'All'. Above the input field are three buttons: a back arrow, a refresh/circular arrow, and a plus icon. On the right, there is a panel with a hamburger menu icon at the top and two numbered instructions: '1. Specify values for all parameters.' and '2. Select the run button to submit the request.'

カレンダーコントロール

カレンダーコントロールは、任意の修飾子を使用した YYMD 日付フォーマットの組み合わせで表されるフィールドに対する単純なフィルタでパラメータを表示します。日付フォーマットにはすべての構成要素 (年、月、日) を含める必要があります。日付時間フィールドフォーマットはサポートされません。



← Jan ▾ 2016 ▾ →						
Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

注意

- ❑ デフォルト値は、現在の日付です。
- ❑ デフォルト値は、英語の月名、2 桁の日付、4 桁の年で指定することができます。以下はその例です。

```
-DEFAULT &STARTDATE='January 01 2016';
```

下図は、デフォルト値が 2016 年 1 月 1 日に設定されたカレンダーコントロールを示しています。

Filter Values

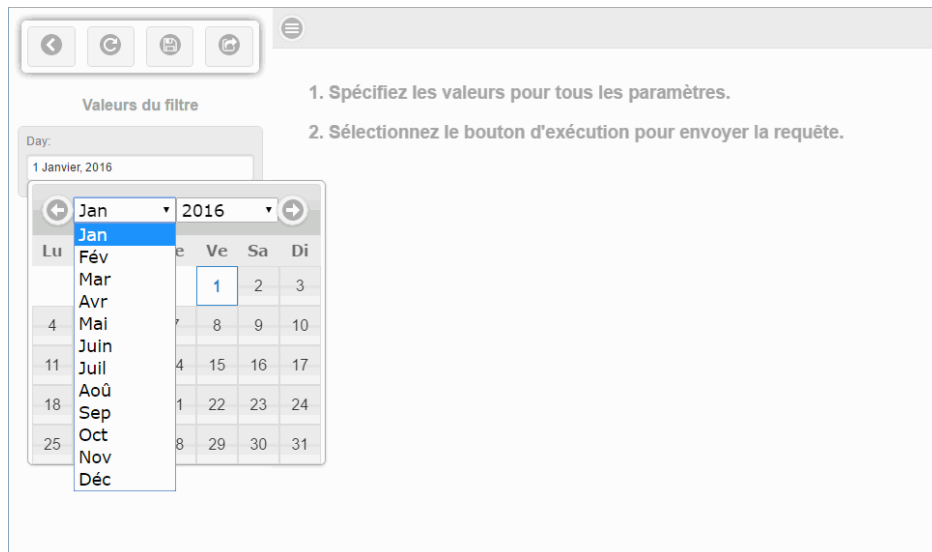
Day:
January 1, 2016

← Jan 2016 →

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

1. Specify values for all parameters.
2. Select the run button to submit the request.

カレンダーコントロールには、ユーザがログイン時に使用した言語でデフォルト日付が表示されます。下図のフィルタウィンドウでは、デフォルト値がフランス語で表示され、カレンダーコントロールのドロップダウンリストにも月名がフランス語で表示されています。



レスポンスサンプルコード (Parameter_Type.fex)

次のリクエストを使用して、160 ページの「[レスポンスオートプロンプトページのコンポーネント](#)」に示したようなレスポンスオートプロンプトページを生成することができます。

1 つ目の連鎖グループ

Product Category: Media Player (単一値選択小リスト)

Product Subcategory: All Values (複数値選択小リスト)

Product Models: All Values (値の数が 25 個以上で、オプションおよびページコントロール付きの複数値選択リスト)

2 つ目の連鎖グループ (値の連鎖の簡易表示)

Business Region: North America (単一値選択小リスト)

Business Subregion: Northeast (単一値選択小リスト)

非連鎖グループ (確認メッセージを表示するために、値の入力または選択なしで [実行] を選択)

Date of sale after: January/1/2016

Cost of Goods above: ¥16.00 (最小値)

Gross Profit above: 1000 (確認メッセージを表示するために、アルファ値を入力し、[実行] を選択)

```

SET EMPTYREPORT=ON
TABLE FILE retail_samples/wf_retail
HEADING CENTER
"Product Models sold after &TIME_DATE in &BUSINESS_REGION
&BUSINESS_SUB_REGION with:"
"Cost of Goods is above &COGS_US"
"Gross Profit is above &GROSS_PROFIT_US"

SUM WF_RETAIL.WF_RETAIL_SALES.COGS_US
WF_RETAIL.WF_RETAIL_SALES.GROSS_PROFIT_US
BY WF_RETAIL.WF_RETAIL_PRODUCT.PRODUCT_CATEGORY
BY WF_RETAIL.WF_RETAIL_PRODUCT.PRODUCT_SUBCATEG
BY WF_RETAIL.WF_RETAIL_PRODUCT.MODEL

- ** 1ST CHAINED GROUPING **
-*Single Select Dynamic List (7 values)
WHERE WF_RETAIL.WF_RETAIL_PRODUCT.PRODUCT_CATEGORY EQ
'&PRODUCT_CATEGORY.(FIND WF_RETAIL.WF_RETAIL_PRODUCT.PRODUCT_CATEGORY
IN WF_RETAIL |FORMAT=A40V).Product Category:.';

-*Multi-select Select Dynamic List (fewer than 10 values for
-*each PRODUCT_CATEGORY value) WHERE
WF_RETAIL.WF_RETAIL_PRODUCT.PRODUCT_SUBCATEG EQ
'&PRODUCT_SUBCATEG.(OR(FIND WF_RETAIL.WF_RETAIL_PRODUCT.PRODUCT_SUBCATEG
IN WF_RETAIL |FORMAT=A50V,WITHIN=PRODUCT_CATEGORY)).Product
Subcategory:.';

-* Chained Multi-select Dynamic List (All Values and # of values
-* dependent on chained selection. Selecting;
-* Product Category (Media Player) and Product Subcategory (Blu Ray)
-* more than 25 values.
-* Product Category (Media Player) and Product Subcategory
-* DVD Players) more than 10 values and fewer than 25 values.
WHERE WF_RETAIL.WF_RETAIL_PRODUCT.MODEL EQ &MODEL.(OR(FIND
WF_RETAIL.WF_RETAIL_PRODUCT.MODEL IN WF_RETAIL
|FORMAT=A50V,WITHIN=PRODUCT_SUBCATEG)).Product Model:.;

- ** 2ND CHAINED GROUPING **
-* Single Multiselect Static List (4 values)
WHERE WF_RETAIL.WF_RETAIL_GEOGRAPHY_STORE.BUSINESS_REGION EQ
'&BUSINESS_REGION.(FIND
WF_RETAIL.WF_RETAIL_GEOGRAPHY_STORE.BUSINESS_REGION IN WF_RETAIL
|FORMAT=A15V).Business Region:.';

```

```
-* Chained Single Dynamic List (10 or fewer values for each
-* Business Region selection)
WHERE WF_RETAIL.WF_RETAIL_GEOGRAPHY_STORE.BUSINESS_SUB_REGION EQ
'&BUSINESS_SUB_REGION.(FIND
WF_RETAIL.WF_RETAIL_GEOGRAPHY_STORE.BUSINESS_SUB_REGION IN WF_RETAIL
|FORMAT=A25V,WITHIN=BUSINESS_REGION).Business Sub Region:.';

-**-** NOT CHAINED GROUPING **
-*Dynamic Single Select List (Over 25 values)
WHERE WF_RETAIL.WF_RETAIL_SALES.COGS_US GT '&COGS_US.(FIND
WF_RETAIL.WF_RETAIL_SALES.COGS_US IN WF_RETAIL |FORMAT=D20.2M).Cost of
Goods above:.';

-*.Simple Prompt (Enter numeric value)
WHERE WF_RETAIL.WF_RETAIL_SALES.GROSS_PROFIT_US GT
&GROSS_PROFIT_US.(|FORMAT=D20.2M).Gross Profit above:.QUOTEDSTRING;

-*.Simple Prompt - Calendar (Select date value)
WHERE WF_RETAIL.WF_RETAIL_TIME_SALES.TIME_DATE GT
&TIME_DATE.(|FORMAT=YYMD).Date of sale after:.QUOTEDSTRING;
ON TABLE SET STYLE *
INCLUDE=IBFS:/FILE/IBI_HTML_DIR/javaassist/intl/EN/combine_templates/
ENWarm.sty,$
ENDSTYLE
END
```

HTML オートプロンプト

HTML オートプロンプト機能を使用すると、プロシジャの実行に必要な変数値の入力をユーザに要求するフォームを作成することができます。ユーザがオートプロンプトフォームで入力または選択する変数値は、レポートリクエストのフィールド値として使用することも、表示コマンドおよびソートコマンドのオブジェクトとして使用することもできます。

注意

- ❑ HTML オートプロンプトは、オートプロンプト機能のデフォルト設定ではありません。レスポンスオートプロンプトがデフォルト設定です。管理コンソールを使用した、オートプロンプト機能の構成方法についての詳細は、156 ページの「[オートプロンプトの構成](#)」を参照してください。
- ❑ HTML オートプロンプト機能は、セクション 508 アクセシビリティ要件をサポートします。
- ❑ HTML オートプロンプトは、連鎖をサポートしません。
- ❑ HTML オートプロンプトには、単純なフィルタで使用する日付フィールドのカレンダーコントロールは含まれません。カレンダーコントロールについての詳細は、168 ページの「[カレンダーコントロール](#)」を参照してください。

- HTML オートプロンプトで表示される動的リストの値は、WebFOCUS Reporting Server によって返されるソート順で整列されます。これは、マシンのオペレーティングシステムによって決定されます。レスポンスオートプロンプトで表示される動的リストの値は、大文字と小文字を区別したソート順で整列されます。

HTML オートプロンプトページのコンポーネント

オートプロンプトフォームで値の選択を要求する変数を含めたレポートリクエスト (FEX) を作成する場合、タイトルや見出しに表示する説明、および値の選択を要求する変数の説明を指定することができます。また、ユーザが値を直接入力するか、静的リストまたは動的リストから値を選択するかを指定したり、入力された値をフォーマット要件や値範囲に対して検証したりすることも可能です。

オートプロンプトの [新規ウィンドウで実行] オプションを使用すると、ユーザが選択したパラメータ値を表示、変更できる状態でレポートを実行したり、レポートを新しいウィンドウに表示したりできます。ユーザはパラメータ領域を非表示にすることで、レポートの表示領域を拡張することができます。

下図は、ページ上部に横方向に表示されるパラメータ値の選択を要求するオートプロンプトフォームを示しています。

The screenshot displays a web form titled "Parameters" with a blue header bar. Below the header, there are two main sections for parameter selection:

- Select a Product Category:** A dropdown menu with "Select All" at the top, followed by "CD Players" (highlighted in blue), "Camcorders", and "Cameras".
- Select a Product Type:** A dropdown menu with "Select All" at the top, followed by "Analog" and "Digital".

Below these sections, there are three buttons: "Run", "Reset", and "Clear Output". To the right of these buttons is a checkbox labeled "Run in a new window".

At the bottom of the form, there is a large white box with a blue border containing the following instructions:

1. Specify values for all parameters.
2. Select the run button to submit the request.

注意: オートプロンプトフォームの各リストコントロールは、それぞれの変数がプロシ ज्या内でコーディングされた順序で表示されます。

オートプロンプトフォームの [パラメータ] ウィンドウには次のオプションがあります。

オプション	説明
実行	このボタンをクリックし、レポートを実行します。
リセット	このボタンをクリックし、パラメータの選択をリセットします。
保存	リポジトリに保存されたレポート (FEX) が実行され、ユーザーに保存済みパラメータレポートの作成権限が与えられている場合、[保存] ボタンが表示されます。InfoAssist、テキストエディタ、HTML レポートキャンバスからリクエストが実行されている場合は、このオプションは使用できません。 注意 ☐ ファイルのタイトル値の最大長は 256 バイトです。 ☐ ファイルのタイトルがフォルダ内の既存ファイルと同一名の場合、既存ファイルを置き換えるかどうかのメッセージが表示されます。 ☐ フィルタ値として使用するフィールドの説明の最大長は 255 バイトです。この制限を超過するテキストは、レポート (FEX) の保存時に自動的に切り捨てられます。
出力結果のクリア	このボタンをクリックし、レポートの出力領域をクリアします。
新規ウィンドウで実行	このチェックをオンにし、レポートを新しいブラウザウィンドウで開きます。
 パラメータの表示と非表示	分割バーをダブルクリックし、レポートを全画面表示するためにパラメータを非表示にします。分割バーを再度ダブルクリックし、元のパラメータおよびレポート表示に戻ります。

レポートの実行後、出力は下図のように表示されます。

Parameters

Select a Product Category

Select All
CD Players
Camcorders
Cameras

Select a Product Type

Select All
Analog
Digital

Run Reset Clear Output
☐ Run in a new window

SALES ACROSS PRODUCT

Region : EAST State: NY

Product Category:

CD Players

Camcorders

Cameras

DVD

Digital Tape Recorders

PDA Devices

Store Name:

AV VideoTown	\$452,430.00	\$4,283,898.00	\$287,212.00	\$786,864.00	\$584,844.00	\$3,487,726.00
Audio Expert	\$359,271.00	\$6,592,944.00	\$337,962.00	\$1,666,342.00	\$1,379,586.00	\$8,250,047.00
Consumer Merchandise	\$89,298.00	\$1,267,066.00		\$493,107.00	\$244,260.00	\$1,235,460.00
eMart	\$2,892,285.00	\$21,045,462.00	\$168,191.00	\$3,196,951.00	\$1,706,370.00	\$9,184,335.00

オートプロンプトに関する考慮事項

HTML オートプロンプトには、次のような重要な考慮事項があります。

- ❑ BI Portal プロシジャが、他の BI Portal プロシジャへの `-INCLUDE` ステートメントを発行する場合、`-INCLUDE` ステートメントで参照されるプロシジャは、元のプロシジャの [パラメータのプロンプト] のプロパティ設定を継承します。
- ❑ 説明の値に左小括弧 (開き括弧) が含まれている場合、その直前にブランクを追加する必要があります。変数の説明では、アンパサンド (&) がサポートされます。その場合、説明内のアンパサンド (&) の直後にパイプ文字 (|) を追加します。次のリクエストの `Select` 句は、その使用例を示しています。

```
TABLE FILE MOVIES
HEADING
PRINT
MOVIES.MOVINFO.TITLE
MOVIES.MOVINFO.COPIES
BY MOVIES.MOVINFO.RATING
BY MOVIES.MOVINFO.CATEGORY
WHERE MOVIES.MOVINFO.RATING EQ &RATING.(OR(<General Audience,G>,
<Not Rated,NR>,<Parental Guidance,PG>,<PG Over 13,PG13>,<R Over
8,R>)).Select Favorite &| Desired Rating.;
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
UNITS=IN,
SQUEEZE=ON,
ORIENTATION=PORTRAIT,$
TYPE=REPORT,
GRID=OFF,
FONT='ARIAL',
SIZE=9,$
TYPE=TITLE,
STYLE=BOLD,$
TYPE=HEADING,
SIZE=12,
STYLE=BOLD,$
ENDSTYLE
END
```

- ❑ -HTMLFORM BEGIN と -HTMLFORM END コマンドの間に参照されているダイアログマネージャ変数は、コメントタグ内に存在する場合でも、すべて評価されます。変数を評価の対象から除外するには、NOEVAL オプション (-HTMLFORM BEGIN NOEVAL) を指定します。WebFOCUS 変数 (&COUNTRY などの値の選択を要求する変数) と、変数として解釈される HTML の記号が混在する場合、NOEVAL オプションは使用しないでください。その場合は、各変数にデフォルト値を割り当てるか、アンパサンド (&) が変数名の開始文字として解釈されないようアンパサンド (&) をパイプ文字 (|) でエスケープします。たとえば、次の例では、アンパサンド (&) がパイプ文字 (|) によってエスケープされているため、コメント内の変数定義は評価されません。

```
-HTMLFORM BEGIN
<HTML>
<BODY>
HELLO &|nbsp &COUNTRY
</BODY>
</HTML>
-HTMLFORM END
```

- ❑ 動的リストでは、WebFOCUS Client が、指定したデータソースから値を取得するためのリクエストを作成します。指定したデータソースから値を取得するためのすべての環境コマンド (例、SET コマンド) は、WebFOCUS Reporting Server プロファイル、ユーザプロファイル、WebFOCUS Client プロファイルのいずれかで発行する必要があります。
- ❑ レスポンシブオートプロンプトで表示される動的リストの値は、大文字と小文字を区別したソート順で整列されます。HTML オートプロンプトで表示される動的リストの値は、WebFOCUS Reporting Server によって返されるソート順で整列されます。これは、マシンのオペレーティングシステムによって決定されます。
- ❑ 動的リストの値は、プロシジャの実行前にオートプロンプト機能によって取得されます。そのため、TABLE リクエストを使用して同一プロシジャの後続の TABLE リクエストで使用する HOLD ファイルを作成し、動的リストに値を追加することはできません。
- ❑ オートプロンプトでは、次の項目はサポートされません。
 - ❑ INCLUDE が -INCLUDE &FILENAME として記述されている場合の INCLUDE ファイル内のパラメータ。
 - ❑ EX または EXEC コマンドが参照するプロシジャ内に記述されたパラメータ。このパラメータは、オートプロンプトでは処理されません。プロシジャでパラメータを参照する必要がある場合は、-INCLUDE ステートメントを使用します。
 - ❑ アンパサンド (&) 文字が含まれたオートプロンプト文字列。
 - ❑ 変数が -HTMLFORM BEGIN ステートメントと -HTMLFORM END ステートメントの間に記述された場合、12 バイトを超える変数名。変数は、解決されません。
 - ❑ 文字の MISSING データ値。数値の MISSING データ値はサポートされます。
- ❑ オートプロンプトでは、次の項目は無視されます。
 - ❑ -SET コマンドで作成した変数。値は明示的に指定されています。
 - ❑ -DEFAULTH コマンドで作成した変数。-DEFAULTH コマンドについての詳細は、183 ページの「[隠し変数の設定](#)」を参照してください。
 - ❑ グローバル変数 (&&NAME)。これは、マスターファイルで定義されたグローバル変数でサポートされます。レポートを実行するたびにグローバル変数値の入力を要求するプロンプトを表示する場合は、[永続変数] (IBIF_PERSISENTAMP) 設定のチェックがオフになっていることを確認してください。このチェックボックスを表示するには、管理コンソールの [構成] タブで [Client 設定] ページを開きます。詳細は、『TIBCO WebFOCUS セキュリティ管理ガイド』を参照してください。

- ❑ 標準モードで実行しているブラウザで iframe 内に出力結果を表示する場合、XML 出力が表示されません。この問題を回避するには、PCHOLD FORMAT XML が指定された WebFOCUS リクエストを実行する際に、[新規ウィンドウで実行] オプションを選択して、出力結果を新しいブラウザウィンドウに表示します。

パラメータベースのフィルタの定義

ここでは、オートプロンプトフォームに表示されるパラメータベースのフィルタの定義方法について説明します。

単純なフィルタのオートプロンプトフォームへの追加

単純なフィルタを追加して、ユーザーに変数の値の入力を求めることができます。

構文 単純なフィルタの追加

`'&variable'`

説明

`&variable`

値の入力が求められる変数名です。アンパサンド (&) を含めます。変数値が文字フィールドと比較される場合、変数を一重引用符 (') で囲む必要があります。ほとんどのデータベースエンジンでは、処理を正しく実行するために、文字列を一重引用符 (') で囲む必要があります。

変数値に一重引用符 (') が含まれる場合は、値を一重引用符 (') で囲む代わりに、接尾語 QUOTEDSTRING を変数に追加します。

例 単純なフィルタの追加

次のリクエストは、在庫数 (Quantity in Stock) の値の入力をユーザーに要求します。

QTY_IN_STOCK フィールドが数値フォーマットであるため、変数 &QTYSTOCK は一重引用符 (') で囲みません。

```
TABLE FILE ibinccen/centord
SUM CENTORD.INVSEG.QTY_IN_STOCK
BY CENTORD.INVSEG.PRODNAME
WHERE CENTORD.INVSEG.QTY_IN_STOCK LT &QTYSTOCK
ON TABLE PCHOLD FORMAT HTML
END
```

フィルタへの変数説明の追加

値の入力をユーザに要求するフォームで、変数名の代わりに表示するテキストを変数の説明としてフォームに追加することができます。変数の説明は、レポートリクエストのフィルタ (WHERE) 式に使用する変数名に追加されます。

構文 フィルタへの変数説明の追加

```
'&variable.description.'
```

説明

&variable

値の入力が求められる変数名です。アンパサンド (&) を含めます。変数値が文字フィールドと比較される場合、変数を一重引用符 (') で囲む必要があります。ほとんどのデータベースエンジンでは、処理を正しく実行するために、文字列を一重引用符 (') で囲む必要があります。

変数値に一重引用符 (') が含まれる場合は、値を一重引用符 (') で囲む代わりに、接尾語 QUOTEDSTRING を変数に追加します。

description

値の入力を要求する際に、変数名の代わりに表示する変数の説明です。

注意：説明の値には、次の制限が適用されます。

- ❑ 変数の説明では、アンパサンド (&) がサポートされます。その場合、説明内のアンパサンド (&) の直後にパイプ文字 (|) を追加します。
- ❑ デフォルトの修飾文字はまた、説明の開始および終了を指定する区切り文字にはピリオド (.) が使用されるため、変数の説明にピリオドを使用することはできません。
- ❑ 左小括弧 (開き括弧) の前には空白を追加する必要があります。

例 変数説明の追加

次のリクエストは、オートプロンプトフォーム内の QTYSTOCK フィールドに説明的な名前を追加します。変数説明は、オートプロンプトフォームのフィールドコントロールより前に記述します。QTY_IN_STOCK フィールドが数値フォーマットであるため、変数 &QTYSTOCK は一重引用符 (') で囲みません。

```
TABLE FILE ibinccen/centord
SUM CENTORD.INVSEG.QTY_IN_STOCK
BY CENTORD.INVSEG.PRODNAME
WHERE CENTORD.INVSEG.QTY_IN_STOCK LT &QTYSTOCK.Quantity In Stock.;
ON TABLE PCHOLD FORMAT HTML
END
```

構文 値範囲リストの追加

```
&variable.(FROM Range1 TO Range2).
```

説明

&variable

値のリストを提供する数値変数です。アンパサンド (&) を含めます。

Range1

値の範囲リストに使用する先頭の数値です。

Range2

値の範囲リストに使用する最終の数値です。

例 値範囲リストの追加

次のリクエストは、QTY_IN_STOCK フィールドの有効な数値のリストを追加します。このリストには、CENTORD データソースの値の中で、指定した範囲内に収まる値が挿入されます。

```
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY STORENAME BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE QTY_IN_STOCK GT &STOCK.(FROM 5000 TO 10000).
END
```

変数のフォーマット指定

変数にフォーマットを指定し、検証時およびソート時の変数の評価方法を特定することができます。

構文 フォーマットの指定

```
&variable.(|FORMAT=format)
```

説明

&variable

値の入力が求められる変数名です。アンパサンド (&) を含めます。変数値が文字フィールドと比較される場合、変数を一重引用符 (') で囲む必要があります。ほとんどのデータベースエンジンでは、処理を正しく実行するために、文字列を一重引用符 (') で囲む必要があります。

変数値に一重引用符 (') が含まれる場合は、値を一重引用符 (') で囲む代わりに、接尾語 QUOTEDSTRING を変数に追加します。

format

フィールドのフォーマットです。デフォルト値は D12.2 です。

例

フォーマットの指定

次のリクエストは、QTYSTOCK フィールドに 7 桁の整数フォーマット (I7) を指定します。

```
TABLE FILE ibinccen/centord
SUM CENTORD.INVSEG.QTY_IN_STOCK
BY CENTORD.INVSEG.PRODNAME
WHERE CENTORD.INVSEG.QTY_IN_STOCK LT &QTYSTOCK.(|FORMAT=I7).Quantity In
Stock;
ON TABLE PCHOLD FORMAT HTML
END
```

注意：数値が入力されない場合、確認メッセージが表示されます。

変数のデフォルト値の設定

変数のデフォルト値を設定することができます。デフォルト値は、ユーザが変数の値を指定しなかった場合に使用されます。変数のデフォルト値を指定するコードは、レポートリクエストより前に記述する必要があります。

構文

変数デフォルト値の設定

```
-DEFAULT &variable=value
```

説明

&variable

値の入力が求められる変数名です。アンパサンド (&) を含めます。変数値が文字フィールドと比較される場合、変数を一重引用符 (') で囲む必要があります。ほとんどのデータベースエンジンでは、処理を正しく実行するために、文字列を一重引用符 (') で囲む必要があります。

変数値に一重引用符 (') が含まれる場合は、値を一重引用符 (') で囲む代わりに、接尾語 QUOTEDSTRING を変数に追加します。

value

指定した変数のデフォルト値です。文字列に一重引用符 (') が含まれている場合は、一重引用符を 2 つ連続させることにより一重引用符を表します。ブランクやカンマ (,) などの区切り文字を含む変数は、引用符で囲む必要があります。

複数選択フィルタで使用される 1 つの変数に複数のデフォルト値を指定するには、値をそれぞれ二重引用符 (") で囲み、変数が値間で参照されるフィルタに使用される演算 (例、OR) を含める必要があります。また、文字列全体は一重引用符 (') で囲む必要があります。たとえば、次の例では、文字列内の各値は二重引用符 (") で囲まれ、文字列全体は一重引用符 (') で囲まれています。このように、文字列の先頭と末尾には 3 つの引用符が含まれます。

```
-DEFAULT &parm1 = '''value1' OR 'value2' OR 'value3''';
```

デフォルト値の _FOC_NULL は、レスポンスオートプロンプトテンプレートに基づいて [すべての値] オプションを選択リストに含める、または HTML オートプロンプトテンプレートに基づいて [すべて選択] オプションを選択リストに含めることにより、ユーザによるフィルタの選択をオプションにします。ユーザがこれらのいずれかのオプションを選択する場合プロシージャがコア WebFOCUS エンジンに渡されてリクエストを実行する前に、内部処理により、WebFOCUS フィルタ句 (例、BY、WHERE) が選択的にパラメータから削除されます。

```
-DEFAULT &parm1 = _FOC_NULL;
```

_FOC_NULL についての詳細は、193 ページの「[_FOC_NULL の内部処理](#)」を参照してください。

デフォルト値の _FOC_NONE を使用して、レスポンスオートプロンプトテンプレートに基づいて [すべての値] オプションを選択リストに含める、または HTML オートプロンプトテンプレートに基づいて [すべて選択] オプションを選択リストに含めることで、ユーザによるフィルタの選択をオプションにすることもできます。この場合、ユーザがこれらのオプションのいずれかを選択した際に、内部処理によって、値 FOC_NONE を含むダイアログマネージャ行以外のすべての行 (ダッシュ (-) で開始しない行) が削除されます。ただし、ダイアログマネージャ以外の行で開始されていても、フィルタ句が複数行にわたって記述されている場合は、この処理によって、フィルタ句全体 (例、BY または WHERE) を削除することはできません。

```
-DEFAULT &parm1 = FOC_NONE;
```

FOC_NONE についての詳細は、194 ページの「[FOC_NONE の内部処理](#)」を参照してください。

例 変数のデフォルト値の設定

次の例は、STATE フィールドに「NY」というデフォルト値を設定する方法を示しています。オートプロンプトフォーム内でこの変数にデフォルト属性を持たせるため、-DEFAULT コマンドのフィールド名の先頭にアンパサンド (&) を追加する必要があります。

```
-DEFAULT &STATE=NY
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY STORENAME BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE.2 letters for US State.'
WHERE STORENAME EQ '&STORENAME.Store Name.'
WHERE PRODNAME EQ '&PRODNAME.Product Name.'
END
```

隠し変数の設定

-DEFAULT コマンドを使用して、ダイアログマネージャ変数にデフォルト値を設定することができます。これらの変数は、WebFOCUS パラメータプロンプト機能 (HTML オートプロンプト、HTML キャンバス、ReportCaster スケジュール) に使用する XML 記述情報に返されます。

-DEFAULTH コマンドで変数の値を初期化し、その値を WebFOCUS パラメータプロンプトで使用されないようにすることができます。-DEFAULTH で初期化した変数は、パラメータプロンプトに使用されません。これらの変数はパラメータプロンプト機能に表示されないため、ユーザーには非表示になります。

変数のデフォルト値を指定するコードは、レポートリクエストより前に記述する必要があります。

構文 隠し変数の定義および初期化

```
-DEFAULTH &variable=value
```

説明

&variable

値の入力が求められる隠し変数名です。アンパサンド記号 (&) を含めます。変数値が文字フィールドと比較される場合、変数を一重引用符 (') で囲む必要があります。ほとんどのデータベースエンジンでは、処理を正しく実行するために、文字列を一重引用符 (') で囲む必要があります。

変数値に一重引用符 (') が含まれる場合は、値を一重引用符 (') で囲む代わりに、接尾語 QUOTEDSTRING を変数に追加します。

value

変数の初期値です。文字列に一重引用符 (') が含まれている場合は、一重引用符を 2 つ連続させることにより一重引用符を表します。ブランクやカンマ (,) などの区切り文字を含む変数は、引用符で囲む必要があります。

複数選択フィルタで使用される 1 つの変数に複数のデフォルト値を指定するには、値をそれぞれ二重引用符 (") で囲み、変数が値間で参照されるフィルタに使用される演算 (例、OR) を含める必要があります。また、文字列全体は一重引用符 (') で囲む必要があります。たとえば、次の例では、文字列内の各値は二重引用符 (") で囲まれ、文字列全体は一重引用符 (') で囲まれています。このように、文字列の先頭と末尾には 3 つの引用符が含まれます。

```
-DEFAULTH &parm1 = '''value1'' OR 'value2' OR 'value3''';
```

値の単一選択リストの追加

値の単一選択リストを追加することができます。このリストの値は、静的または動的のいずれかにすることができます。

構文 静的単一選択リストの追加

```
'&variable.(value,value2[,value3][,value4]...  
[|FORMAT=format] [,WITHIN=within]).[description.]'
```

説明

&variable

値のリストを提供する変数です。アンパサンド (&) を含めます。

value, value2, value3, value4...

選択可能な変数値で構成される一連の値です。

format

検証時およびソート時の変数の評価方法を指定します。FORMAT 属性はすべてのフィルタタイプで使用できます。

within

WITHIN 属性は、レスポンスオートプロンプトテンプレートでサポートされる連鎖に使用します。同一ディメンション階層内のフィールドに作成される連鎖グループの 1 つ目のフィールドでは、動的または静的フィルタが必要です。連鎖グループの 2 つ目以降のフィールドでは動的フィルタが必要です。フィルタは、ディメンション階層内のマスターファイルのフィールド参照順序に従います。パラメータ名は、マスターファイルのフィールド名と同一です。

description

オプションとして追加する変数の説明です。詳細および制限については、179 ページの「[フィルタへの変数説明の追加](#)」を参照してください。

注意：静的値を WHERE ステートメント内で指定する場合、カンマ (,) が含まれた静的値を使用することはできません。これは、表示値を指定するための区切り文字としてカンマ (,) が使用されるためです。1 つまたは複数の静的値にカンマ (,) が含まれている場合は、すべての値をファイルに保存し、動的リスト (FIND) 機能を使用します。

例 静的単一選択リストの追加

次の例は、STORENAME フィールドの有効値リストを追加する方法を示しています。ユーザーは、リストから 1 つの値のみを選択することができます。

```
-DEFAULT &STATE=NY
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY STORENAME BY PRODNAM
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE.2 letters for US State.'
WHERE STORENAME EQ '&STORENAME.(eMart,TV City,Web Sales).Store Name.'
WHERE PRODNAM EQ '&PRODNAM.Product Name.'
END
```

構文 動的単一選択リストの追加

```
'&variable.(FIND return_fieldname [,display_fieldname] IN datasource
[|SORT=sortoption] [|FORMAT=format] [,WITHIN=within]).[description.]'
```

説明

&variable

値のリストを提供する変数です。アンパサンド (&) を含めます。

return_fieldname

プロシジャに返す変数値が含まれるフィールドの名前です。

display_fieldname

オートプロンプトフォームに表示する変数値が含まれるフィールドの名前です。

datasource

return_fieldname および display_fieldname で指定したフィールドが含まれるデータソースです。JOIN で使用されるデータソースのクロスリファレンスファイルにフィールドが存在する場合は、そのフィールドが含まれるデータソースの名前を使用します。

注意

- ❑ 動的リストでは、WebFOCUS Client が、指定したデータソースから値を取得するためのリクエストを作成します。指定したデータソースから値を取得するためのすべての環境コマンド (例、SET コマンド) は、WebFOCUS Reporting Server プロファイル、ユーザープロファイル、WebFOCUS Client プロファイルのいずれかで発行する必要があります。
- ❑ レスポンシブオートプロンプトで表示される動的リストの値は、大文字と小文字を区別したソート順で整列されます。HTML オートプロンプトで表示される動的リストの値は、WebFOCUS Reporting Server によって返されるソート順で整列されます。これは、マシンのオペレーティングシステムによって決定されます。

sortoption

動的リストの戻り値をソートする方法を指定します。有効な値には、次のものがあります。

- ❑ **ASCENDING** 昇順にソートします。ソート処理が指定されていない場合、これがデフォルト値です。
- ❑ **DESCENDING** 降順にソートします。

format

ソートオプションが指定されていない場合、検証時およびソート時の変数の評価方法を指定します。FORMAT 属性はすべてのフィルタタイプで使用できます。

within

WITHIN 属性は、レスポンシブオートプロンプトテンプレートでサポートされる連鎖に使用します。同一ディメンション階層内のフィールドに作成される連鎖グループの 1 つ目のフィールドでは、動的または静的フィルタが必要です。連鎖グループの 2 つ目以降のフィールドでは動的フィルタが必要です。フィルタは、ディメンション階層内のマスターファイルのフィールド参照順序に従います。パラメータ名は、マスターファイルのフィールド名と同一です。

description

オプションとして追加する変数の説明です。詳細および制限については、179 ページの「[フィルタへの変数説明の追加](#)」を参照してください。

例 動的単一選択リストの追加

次の例は、製品名 (PRODNAME) フィールドの有効な値のリストを追加する方法を示しています。このリストには、CENTORD データソースの値が挿入されます。ユーザは、リストから 1 つの値のみを選択することができます。

```
-DEFAULT &STATE=NY
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY STORENAME BY PROD_NUM
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE.2 letters for US State.'
WHERE STORENAME EQ '&STORENAME.(eMart,TV City,Web Sales).Store Name.'
WHERE PROD_NUM EQ '&PROD_NUM.(FIND PROD_NUM,PRODNAME IN CENTORD).Product
Name.'
END
```

例 降順ソートの動的単一選択リストの追加

次の例は、製品名 (PRODNAME) フィールドの有効な値のリストを追加する方法を示しています。このリストには、CENTORD データソースの値が挿入されます。ユーザは、リストから 1 つの値のみを選択することができます。

```
-DEFAULT &STATE=NY
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY STORENAME BY PROD_NUM
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE.2 letters for US State.'
WHERE STORENAME EQ '&STORENAME.(eMart,TV City,Web Sales).Store Name.'
WHERE PROD_NUM EQ '&PROD_NUM.(FIND PROD_NUM,PRODNAME IN CENTORD|
SORT=DESCENDING).Product Name.'
END
```

例 連鎖値

連鎖フィールドは、ディメンション階層のマスタファイルで参照されるフィールドに基づき、選択可能な値を制限します。WITHIN 属性は、フィルタの連鎖を指定します。連鎖グループの 1 つ目のフィールドには動的または静的リストを使用し、連鎖内の 2 つ目以降のフィルタには動的フィルタを使用する必要があります。次の例は、製品タイプ (PRODTYPE)、製品区分 (PRODCAT)、製品名 (PRODNAME) のフィールドに連鎖値リストを指定します。たとえば、製品区分 (PRODCAT) フィールドのリストの値は、製品タイプ (PRODTYPE) フィールドの選択リストの値と同じです。

```
TABLE FILE ibinccen/centord
SUM CENTORD.INVSEG.QTY_IN_STOCK
CENTORD.INVSEG.PRICE
BY CENTORD.INVSEG.PRODTYPE
BY CENTORD.INVSEG.PRODCAT
BY CENTORD.INVSEG.PRODNAME
WHERE CENTORD.INVSEG.PRODTYPE EQ &PRODTYPE.(OR(FIND CENTORD.INVSEG.PRODTYPE
IN ibinccen/CENTORD |FORMAT=A19)).PRODTYPE:.;
WHERE CENTORD.INVSEG.PRODCAT EQ &PRODCAT.(OR(FIND CENTORD.INVSEG.PRODCAT IN
ibinccen/CENTORD |FORMAT=A22 ,WITHIN=PRODTYPE)).PRODCAT:.;
WHERE CENTORD.INVSEG.PRODNAME EQ &PRODNAME.(OR(FIND CENTORD.INVSEG.PRODNAME
IN ibinccen/CENTORD |FORMAT=A30 ,WITHIN=PRODCAT)).Product Name:.;
END
```

複数選択リストの追加

値の複数選択リストを追加することができます。このリストの値は、静的または動的のいずれかにすることができます。

構文 静的単一選択リストの追加

```
&variable.(operation (<displayvalue1,value1>,
<displayvalue2,value2>,...<displayvalueN,valueN>)
[|FORMAT=format] [,WITHIN=within]).[description.]
```

説明

&variable

値のリストを提供する変数です。アンパサンド (&) を含めます。

operation

複数值の評価方法を指定します。使用可能な値には、OR、AND、またはカンマ (,) があります。この値を省略すると、デフォルト値の OR が使用されます。

displayvalue, displayvalue2, displayvalue3, displayvalue4...

選択可能な変数値のリストです。

`value, value2, value3, value4...`

WebFOCUS Reporting Server に渡される選択可能な変数値のリストです。

`format`

検証時およびソート時の変数の評価方法を指定します。FORMAT 属性はすべてのフィルタタイプで使用できます。

`within`

WITHIN 属性は、レスポンスオートプロンプトテンプレートでサポートされる連鎖に使用します。同一ディメンション階層内のフィールドに作成される連鎖グループの 1 つ目のフィールドでは、動的または静的フィルタが必要です。連鎖グループの 2 つ目以降のフィールドでは動的フィルタが必要です。フィルタは、ディメンション階層内のマスターファイルのフィールド参照順序に従います。パラメータ名は、マスターファイルのフィールド名と同一です。

`description`

オプションとして追加する変数の説明です。詳細および制限については、179 ページの「[フィルタへの変数説明の追加](#)」を参照してください。

注意：静的値を WHERE ステートメント内で指定する場合、カンマ (,) が含まれた静的値を使用することはできません。これは、表示値を指定するための区切り文字としてカンマ (,) が使用されるためです。1 つまたは複数の静的値にカンマ (,) が含まれている場合は、すべての値をファイルに保存し、動的リスト (FIND) 機能を使用します。

例

静的複数選択リストの追加

次のリクエストは、STORENAME フィールドの有効値リストを追加する方法を示しています。ユーザは、リストから複数の値を選択することができます。

```
-DEFAULT &STATE=NY
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY STORENAME BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE.2 letters for US State.'
WHERE STORENAME EQ &STORENAME. (OR(eMart,TV City,Web Sales)).Store Name.
WHERE PRODNAME EQ '&PRODNAME.Product Name.'
END
```

構文

動的複数選択リストの追加

```
&variable.(operation (FIND return_fieldname [,display_fieldname] IN
datasource[|SORT=sortoption] [|FORMAT=format] [,WITHIN=within])).
[description.]
```

説明

&variable

値のリストを提供する変数です。アンパサンド (&) を含めます。

operation

複数値の評価方法を指定します。使用可能な値には、OR、AND、またはカンマ (,) があります。この値を省略すると、デフォルト値の OR が使用されます。

return_fieldname

プロシジャに返す変数値が含まれるフィールドの名前です。

display_fieldname

オートプロンプトフォームに表示する変数値が含まれるフィールドの名前です。

datasource

return_fieldname および display_fieldname で指定したフィールドが含まれるデータソースです。JOIN で使用されるデータソースのクロスリファレンスファイルにフィールドが存在する場合は、そのフィールドが含まれるデータソースの名前を使用します。

注意

- ❑ 動的リストでは、WebFOCUS Client が、指定したデータソースから値を取得するためのリクエストを作成します。指定したデータソースから値を取得するためのすべての環境コマンドは (例、SET コマンド)、WebFOCUS Reporting Server プロファイル、ユーザプロファイル、WebFOCUS Client プロファイルのいずれかで発行する必要があります。
- ❑ レスポンシブオートプロンプトで表示される動的リストの値は、大文字と小文字を区別したソート順で整列されます。HTML オートプロンプトで表示される動的リストの値は、WebFOCUS Reporting Server によって返されるソート順で整列されます。これは、ホストマシンのオペレーティングシステムによって決定されます。

sortoption

動的リストの戻り値をソートする方法を指定します。有効な値には、次のものがあります。

- ❑ **ASCENDING** 昇順にソートします。ソート処理が指定されていない場合、これがデフォルト値です。
- ❑ **DESCENDING** 降順にソートします。

format

ソートオプションが指定されていない場合、検証時およびソート時の変数の評価方法を指定します。FORMAT 属性はすべてのフィルタタイプで使用できます。

within

WITHIN 属性は、レスポンスオートプロンプトテンプレートでサポートされる連鎖に使用します。同一ディメンション階層内のフィールドに作成される連鎖グループの1つ目のフィールドでは、動的または静的フィルタが必要です。連鎖グループの2つ目以降のフィールドでは動的フィルタが必要です。フィルタは、ディメンション階層内のマスターファイルのフィールド参照順序に従います。パラメータ名は、マスターファイルのフィールド名と同一です。

description

オプションとして追加する変数の説明です。詳細および制限については、179 ページの「[フィルタへの変数説明の追加](#)」を参照してください。

例**動的複数選択リストの追加**

次のリクエストは、STATE フィールドの有効値リストを追加する方法を示しています。このリストには、CENTORD データソースの値が挿入されます。ユーザは、リストから複数の値を選択することができます。

```
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STORE_CODE BY STATE BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STORE_CODE EQ &STORE_CODE. (OR(FIND STORE_CODE,STORENAME IN
CENTORD)).Store Name.
WHERE STATE EQ &STATE.(OR(CA,IL,MA,NY,NJ,FL,TX)).2 letters for US State.
WHERE PRODNAME EQ '&PRODNAME.(FIND PRODNAME IN CENTORD).Product Name.'
END
```

例**降順ソートの動的複数選択リストの追加**

次のリクエストは、STATE フィールドの有効値リストを追加する方法を示しています。このリストには、CENTORD データソースの値が挿入されます。ユーザは、リストから複数の値を選択することができます。

```
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STORE_CODE BY STATE BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STORE_CODE EQ &STORE_CODE. (OR(FIND STORE_CODE,STORENAME IN CENTORD|
SORT=DESCENDING)).Store Name.
WHERE STATE EQ &STATE.(OR(CA,IL,MA,NY,NJ,FL,TX)).2 letters for US State.
WHERE PRODNAME EQ '&PRODNAME.(FIND PRODNAME IN CENTORD).Product Name.'
END
```

参照 値の複数選択リストに関する規則

- ❑ 値の複数選択リストを作成する場合、それぞれの値を一重引用符 (') で囲まないでください。複数選択リストの場合、各値は自動的に一重引用符 (') で囲まれます。値自体に一重引用符 (') を追加すると、エラーが発生します。たとえば、WHERE COUNTRY EQ '&COUNTRY' と記述すると、エラーが発生します。一重引用符 (') を使用せずに &COUNTRY と記述します。
- ❑ 値の複数選択リストの結果をドリルダウンのパラメータとして使用する場合、パラメータ間の区切り記号として挿入された埋め込みブランクは、QUOTEDSTRING を使用して処理します。

正しい記述は次のようになります。

```
FOCEXEC=DRILLDOWN(COUNTRY=&COUNTRY.QUOTEDSTRING OTHER=xxx)
```

次の記述は誤りです。&COUNTRY パラメータ内の埋め込みブランクにより、ドリルダウンが終了します。

```
FOCEXEC=DRILLDOWN(COUNTRY=&COUNTRY OTHER=xxx)
```

詳細は、344 ページの「[標準引用符付き文字列の作成](#)」を参照してください。

参照 複数選択変数値の選択

- ❑ 静的な値の複数選択リストを使用したレポートを作成し、HTML オートプロンプトテンプレートを構成した場合、[すべて選択] オプションを選択して、パラメータリストの値をすべて選択することができます。内部的には、[すべて選択] オプションを選択した場合、パラメータには、WHERE ステートメントで修飾子 AND、OR、またはカンマ (,) で区切られた複数選択リストのすべての値が設定されます。詳細は、195 ページの「[「すべて選択」の内部処理](#)」を参照してください。
- ❑ 動的な値の複数選択リストを使用したレポートでは、[すべての値] オプションを選択して、パラメータリストの値をすべて未選択の状態にすることができます。[すべての値] オプションは、レスポンスオートプロンプトテンプレートおよび HTML オートプロンプトテンプレートで使用できます。動的な複数選択リストから [すべての値] オプションを選択した場合、内部的には _FOC_NULL 値または FOC_NONE 値のいずれかが渡されます。デフォルト設定では、_FOC_NULL 値が渡されます。ただし、IBIF_describe_null 変数を使用して、[すべての値] オプションでこれらの値のどちらを渡すかを制御することができます。IBIF_describe_null は、WebFOCUS Client から送信する値を構成する変数です。

- ❑ `_FOC_NULL` 値を含む句および式は、プロシジャが WebFOCUS Reporting Server で実行される前にプロシジャから削除されます。`_FOC_NULL` についての詳細は、193 ページの「[_FOC_NULL の内部処理](#)」を参照してください。
- ❑ `FOC_NONE` 値を含むダイアログマネージャ以外の行 (ハイフン (-) で始まらない行) は、プロシジャが WebFOCUS Reporting Server で実行される前にプロシジャから削除されます。`FOC_NONE` についての詳細は、194 ページの「[FOC_NONE の内部処理](#)」を参照してください。

参照

`_FOC_NULL` の内部処理

複数選択リストで [すべての値] オプションを選択するか、手動でコーディングした場合、WebFOCUS Reporting Server に送信されるパラメータ値は、デフォルト設定で `_FOC_NULL` になります。次に、WebFOCUS Reporting Server の記述層で内部処理が実行され、プロシジャを検索して `_FOC_NULL` を含む WebFOCUS 句 (例、BY、WHERE) や式が部分的に削除されます。続いて、プロシジャがの中核エンジンに渡され、リクエストが実行されます。

次の例は、動的な複数選択リストで `_FOC_NULL` 値を生成する [すべての値] オプションを選択した場合または手動でコーディングした場合の WHERE ステートメントを示しています。

```
WHERE CATEGORY EQ &CATEGORY.(OR(FIND CATEGORY IN MOVIES)).CATEGORY. AND
DIRECTOR EQ &DIRECTOR.(OR(FIND DIRECTOR IN MOVIES)).DIRECTOR.;
```

CATEGORY で [すべての値] が返された場合でも、DIRECTOR のテストが処理されます。

次の例は、静的複数選択リストおよび動的複数選択リストで `_FOC_NULL` 値を生成する [すべての値] オプションを選択した場合または手動でコーディングした場合の WHERE ステートメントを示しています。

```
WHERE COPIES EQ
&COPIES.(<All Values,_FOC_NULL>,<1,1>,<2,2>,<3,3>).COPIES.
AND DIRECTOR EQ &DIRECTOR.(OR(FIND DIRECTOR IN MOVIES)).DIRECTOR.;
```

各リストは、プロシジャの単一行にコーディングする必要があります。

HEADING、FOOTING、SUBHEAD、SUBFOOT をコーディングする場合、パラメータに `_FOC_NULL` 値が割り当てられることのない行を少なくとも 1 行含める必要があります。

参照

FOC_NONE の内部処理

複数選択リストで [すべての値] オプションを選択した場合または手動でコーディングした場合、WebFOCUS Reporting Server に送信されるパラメータ値は FOC_NONE になります。次に、WebFOCUS Reporting Server の記述層で内部処理が実行され、プロシジャを検索して FOC_NONE を含むダイアログマネージャ以外の行 (ハイフン (-) で始まらない行) がすべて削除されます。続いて、プロシジャが WebFOCUS の中核エンジンに渡され、リクエストが実行されます。

次の例は、FOC_NONE 値を生成する [すべての値] オプションを選択した場合または手動でコーディングした場合の WHERE ステートメントを示しています。

❑ 動的な複数選択

```
WHERE CATEGORY EQ &CATEGORY.(OR(FIND CATEGORY IN MOVIES)).CATEGORY.;
```

❑ FOC_NONE を使用した静的な複数選択 (プロシジャの単一行にコーディング)

```
WHERE COPIES EQ &COPIES.(<All Values,FOC_NONE>,<1,1>,<2,2>,<3,3>).COPIES.;
```

以下は、FOC_NONE のコーディングに関する考慮事項です。

- ❑ HEADING、FOOTING、SUBHEAD、SUBFOOT をコーディングする場合、パラメータに FOC_NONE 値が割り当てられることのない行を少なくとも 1 行含める必要があります。
- ❑ 動的な複数選択リストで [すべての値] オプションが使用されている場合、または手動でコーディングした WHERE ステートメントから FOC_NONE パラメータ値が渡される場合は、次のことに注意する必要があります。
 - ❑ FOC_NONE 値を渡す WHERE ステートメントは、プロシジャの単一行にコーディングします。これにより、リクエストから WHERE ステートメントが部分的に削除されることによる FOC002 エラーの発生が回避されます。
 - ❑ 特定のフィールドで FOC_NONE 値が渡される場合、単一行に複雑な WHERE ステートメントをコーディングしないようにします。これにより、リクエストから複雑な WHERE ステートメント全体が削除されなくなります。複雑な WHERE ステートメントとは、AND または OR 演算子を使用して複数のフィールドに対して選択テストを実行するものです。

参照 「すべて選択」の内部処理

静的な複数選択リストを含むリクエストを実行すると、HTML オートプロンプト機能により [すべて選択] オプションが値リストに動的に追加されます。このオプションが選択された場合、静的な複数選択リストのすべての値がパラメータに割り当てられます。[すべて選択] 以外に特定の値が選択された場合、[すべて選択] の値は無視されます。

内部処理は、オートプロンプト機能がパラメータを選択リスト値に割り当てることで完了します。選択リスト値は、パラメータを参照する WHERE ステートメントで指定され、AND または OR 修飾子で区切られた値です。

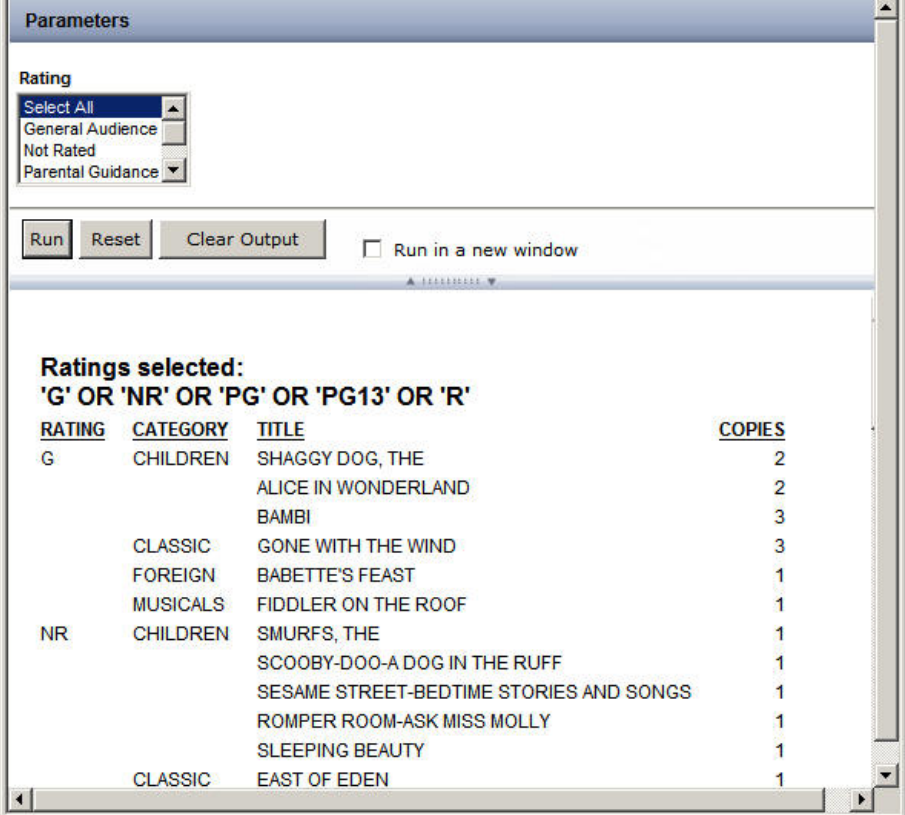
次の例では、OR 修飾子を使用した WHERE ステートメントで、&RATING パラメータの静的な複数選択リストが記述されています。&RATING パラメータは、WHERE ステートメントと HEADING で参照されています。HEADING では、パラメータ値を、OR 修飾子で区切られた複数選択リストのすべての値を設定する方法が示されています。

```

TABLE FILE MOVIES
HEADING
" Ratings selected: "
" &RATING "
PRINT
    'MOVIES.MOVINFO.TITLE'
    'MOVIES.MOVINFO.COPIES'
BY 'MOVIES.MOVINFO.RATING'
BY 'MOVIES.MOVINFO.CATEGORY'
HEADING
""
FOOTING
""
WHERE MOVIES.MOVINFO.RATING EQ &RATING.(OR(<General Audience,G>,<Not
Rated,NR>,<Parental Guidance,PG>,<PG Over 13,PG13>,<R Over 18,R>)).Rating.;
ON TABLE SET PAGE-NUM OFF
ON TABLE NOTOTAL
ON TABLE PCHOLD FORMAT HTML
ON TABLE SET HTMLCSS ON
ON TABLE SET STYLE *
    UNITS=IN,
    SQUEEZE=ON,
    ORIENTATION=PORTRAIT,
$
TYPE=REPORT,
    GRID=OFF,
    FONT='ARIAL',
    SIZE=9,
$
TYPE=TITLE,
    STYLE=BOLD,
$
TYPE=HEADING,
    SIZE=12,
    STYLE=BOLD,
$
ENDSTYLE
END

```

下図は、[すべて選択] が選択された場合の HTML オートプロンプトフォームとレポートを示しています。



Parameters

Rating
 Select All
 General Audience
 Not Rated
 Parental Guidance

Run Reset Clear Output ☐ Run in a new window

Ratings selected:
'G' OR 'NR' OR 'PG' OR 'PG13' OR 'R'

RATING	CATEGORY	TITLE	COPIES
G	CHILDREN	SHAGGY DOG, THE	2
		ALICE IN WONDERLAND	2
		BAMBI	3
	CLASSIC	GONE WITH THE WIND	3
	FOREIGN	BABETTE'S FEAST	1
NR	MUSICALS	FIDDLER ON THE ROOF	1
	CHILDREN	SMURFS, THE	1
		SCOOBY-DOO-A DOG IN THE RUFF	1
		SESAME STREET-BEDTIME STORIES AND SONGS	1
	ROMPER ROOM-ASK MISS MOLLY	1	
	SLEEPING BEAUTY	1	
	CLASSIC	EAST OF EDEN	1

WebFOCUS Client は、次の構文を作成し、WebFOCUS Reporting Server に送信されるリクエストに追加することで、&RATING パラメータに値リストを割り当てます。

```
-SET &RATING='G' OR 'NR' OR 'PG' OR 'PG13' OR 'R';
```

表示フィールドおよびソートフィールドの静的リストの追加

フィールドの静的リストでは、表示コマンドまたはソートコマンドに適用するフィールド名および表示値を選択リストから選択します。

構文 ソートフィールドまたは表示フィールドの静的単一選択リストの追加

```
cmd &var.(<dsply1[,val1>],<dsply2[,val2>],...<dsplyN[,valN>)].[desc.]
```

説明

cmd

フィールドのリストを適用するコマンドです。有効な値は、PRINT、COUNT、SUM、WRITE、ADD、BY、ACROSS です。

&var

フィールド値のリストを提供する変数です。アンパサンド (&) を含めます。

dsply1, dsply2, ...

オートプロンプトフォームに表示するエントリです。エントリを選択すると、それに対応するフィールドが自動的に選択されます。

val1, val2, ...

WebFOCUS Reporting Server に転送するフィールド名です。これらのフィールド名は、修飾フィールド名にすることができます。また、表示値とフィールド名が一致する場合は、これらのフィールド名を省略することができます。

desc

オプションとして追加する変数の説明です。

注意：説明を追加した場合は、その説明を含めたコマンド全体をレポートリクエストの 1 行に記述する必要があります。

例 SUM フィールドおよびソートフィールドの単一選択リストの追加

次のリクエストは、GGSALES データソースに対して実行され、プロシジャの実行時に、ユーザは各リストから次のいずれかを選択することができます。

- ☐ SUM フィールド - DOLLARS、BUDDOLLARS、UNITS
- ☐ BY フィールド - CATEGORY、PRODUCT
- ☐ ACROSS フィールド - REGION、ST、None (選択なし)

注意：[すべての値] オプションについての詳細は、194 ページの「[FOC_NONE の内部処理](#)」を参照してください。

```

TABLE FILE GGSales
HEADING
"Sales Report Summary for: "
"Fields: &SumFlds"
"Sort: &SortBy"
"Across: &Acrs"
" "
SUM &SumFlds.(<Sales,DOLLARS>,<Budget,BUDDOLLARS>,<Units,UNITS>).Sum.
BY &SortBy.(<Category,GGSales.SALES01.CATEGORY>,<PRODUCT>).By.
ACROSS &Acrs.(<Area,REGION>,<State,ST>,<None,FOC_NONE>).Across.
WHERE REGION NE 'West'
ON TABLE SET PAGE NOPAGE
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT=ARIAL, SIZE=10, SQUEEZE=ON, $
TYPE=HEADING, FONT=ARIAL, STYLE=BOLD, JUSTIFY=LEFT, $
TYPE=HEADING, LINE=1, FONT=ARIAL, STYLE=BOLD, JUSTIFY=LEFT, $
TYPE=TITLE, BACKCOLOR=BLACK, COLOR=WHITE, $
TYPE=ACROSSVALUE, BACKCOLOR=LIGHT BLUE, COLOR=BLUE, $
TYPE=ACROSSVALUE, BACKCOLOR=LIGHT GRAY, COLOR=GREEN, $
END

```

このリクエストを実行すると、下図のようなオートプロンプトフォームが開きます。このフォームでは、REGION ソートフィールドの選択項目は「Area」、DOLLARS フィールドの選択項目は「Sales」として表示され、CATEGORY ソートフィールド、BUDDOLLARS フィールド、UNITS フィールドの選択項目は大文字と小文字が混在した文字で表示され、PRODUCT フィールドの表示値はソートフィールド名と同一になっています。各リストからは、それぞれ 1 つの項目のみを選択することができます。

The screenshot shows a 'Parameters' dialog box. It contains three dropdown menus labeled 'Sum', 'By', and 'Across'. The 'Sum' dropdown has 'Sales' selected. The 'By' dropdown has 'Category' selected. The 'Across' dropdown has 'None' selected. Below these dropdowns are three buttons: 'Run', 'Reset', and 'Clear Output'. At the bottom right of the dialog is a checkbox labeled 'Run in a new window' which is currently unchecked.

[Sales]、[Category]、[None] を順に選択すると、次のレポートが生成されます。

Sales Report Summary for:
Fields: DOLLARS
Sort: GGSales.SALES01.CATEGORY

Category	Dollar Sales
Coffee	12757938
Food	13026996
Gifts	8718410

構文 ソートフィールドの静的複数選択リストの追加

```
cmd &var.(cmd(<dsply1[,val1>],<dsply2[,val2>],...<dsplyN[,srtN>))).  
[desc.]
```

説明

cmd

フィールドのリストを適用するコマンドです。有効な値は BY および ACROSS です。選択リストを囲んでコマンドを追加すると、複数選択リストが作成されます。

&var

フィールド値のリストを提供する変数です。アンパサンド (&) を含めます。

dsply1, dsply2, ...

オートプロンプトフォームに表示するエントリです。エントリを選択すると、それに対応するソートフィールドが自動的に選択されます。

val1, val2, ...

WebFOCUS Reporting Server に転送するフィールド名です。これらのフィールド名は、修飾フィールド名にすることができます。また、表示値とフィールド名が一致する場合は、これらのフィールド名を省略することができます。

desc

オプションとして追加する変数の説明です。

注意：説明を追加した場合は、その説明を含めたコマンド全体をレポートリクエストの 1 行に記述する必要があります。

構文

表示フィールドの静的複数選択リストの追加

```
cmd &var.(AND(<dsply1[,val1>],<dsply2[,val2>],...<dsplyN[,srtN>])).[desc.]
```

説明

cmd

フィールドのリストを適用するコマンドです。有効な値は、PRINT、COUNT、SUM、WRITE、ADD です。

&var

フィールド値のリストを提供する変数です。アンパサンド (&) を含めます。

AND

複数選択リストを作成する接続子です。

dsply1, dsply2, ...

オートプロンプトフォームに表示するエントリです。エントリを選択すると、それに対応するソートフィールドが自動的に選択されます。

val1, val2, ...

WebFOCUS Reporting Server に転送するフィールド名です。これらのフィールド名は、修飾フィールド名にすることができます。また、表示値とフィールド名が一致する場合は、これらのフィールド名を省略することができます。

desc

オプションとして追加する変数の説明です。

注意：説明を追加した場合は、その説明を含めたコマンド全体をレポートリクエストの 1 行に記述する必要があります。

例

SUM フィールドおよびソートフィールドの複数選択リストの追加

次のリクエストは、GGSALES データソースに対して実行され、プロシジャの実行時に、ユーザは次の 1 つまたは複数のフィールドを選択することができます。

- ☐ SUM フィールド - DOLLARS、BUDDOLLARS、UNITS
- ☐ BY フィールド - CATEGORY、PRODUCT
- ☐ ACROSS フィールド - REGION、ST、None (選択なし)

```
TABLE FILE GGSales
HEADING
"Sales Report Summary for: "
"Fields: &SumFlds"
"Sort: &SortBy"
"Across: &Acrs"
" "
SUM &SumFlds.(AND(<Sales,DOLLARS>,<Budget,BUDDOLLARS>,<Units,UNITS>)).Sum.
BY &SortBy.(BY(<Category,GGSales.SALES01.CATEGORY>,<PRODUCT>)).By.
ACROSS &Acrs.(ACROSS(<Area,REGION>,<State,ST>,<None,FOC_NONE>)).Across.
WHERE REGION NE 'West'
ON TABLE SET PAGE NOPAGE
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT=ARIAL, SIZE=10, SQUEEZE=ON, $
TYPE=HEADING, FONT=ARIAL, STYLE=BOLD, JUSTIFY=Center, $
TYPE=HEADING, LINE=1, FONT=ARIAL, STYLE=BOLD, JUSTIFY=Center, $
TYPE=TITLE, BACKCOLOR=BLACK, COLOR=WHITE, $
TYPE=ACROSSTITLE, BACKCOLOR=LIGHT BLUE, COLOR=BLUE, $
TYPE=ACROSSVALUE, BACKCOLOR=LIGHT GRAY, COLOR = GREEN, $
END
```

このリクエストを実行すると、下図のようなオートプロンプトフォームが開きます。このフォームでは、REGION ソートフィールドの選択項目は「Area」、DOLLARS フィールドの選択項目は「Sales」として表示され、CATEGORY ソートフィールド、BUDDOLLARS フィールド、UNITS フィールドの選択項目は大文字と小文字が混在した文字で表示され、PRODUCT フィールドの表示値はソートフィールド名と同一になっています。

Parameters			
Sum Select All Sales Budget Units	By Select All Category PRODUCT	Across Select All Area State None	
Run	Reset	Clear Output	☐ Run in a new window
1. Specify values for all parameters. 2. Select the run button to submit the request.			

[Sales]、[Units]、[Category]、[Area] を順に選択すると、次のレポートが生成されます。

Sales Report Summary for:
Fields: DOLLARS AND UNITS
Sort: GGSales.SALES01.CATEGORY
Across: REGION

Category	Region					
	Midwest		Northeast		Southeast	
	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales
Coffee	4178513	332777	4164017	335778	4415408	350948
Food	4338271	341414	4379994	353368	4308731	349829
Gifts	2883881	230854	2848289	227529	2986240	234455

表示フィールドおよびソートフィールドの動的リストの追加

フィールドの動的リストでは、選択リストに FIND コマンドを使用して、データソースからフィールド名および表示値を取得することができます。

参照 フィールド名を含むデータソースの作成

次の例では、3 つのデータソースを作成し、それらのデータソースを動的フィールドリストを作成する例で使します。

- ❑ GGDETFD データソースには、選択リストの表示フィールドとして使用する、GGSales データソースのフィールド名を格納します。
- ❑ GGBYFLD データソースには、選択リストの BY フィールドとして使用する、GGSales データソースのフィールド名を格納します。
- ❑ GGACRFLD データソースには、選択リストの ACROSS フィールドとして使用する、GGSales データソースのフィールド名を格納します。

例 GGETFLD データソースの作成

次のリクエストは、GGETFLD という名前の FOCUS データソースを作成します。

```
APP HOLD IBISAMP
DEFINE FILE SYSCOLUM
FLDDESC/A20=IF NAME EQ 'DOLLARS' THEN 'Dollars'
      ELSE IF NAME EQ 'BUDDOLLARS' THEN 'Budget'
      ELSE IF NAME EQ 'UNITS' THEN 'Units' ELSE 'N/A';
END
TABLE FILE SYSCOLUM
PRINT FLDDESC
BY NAME
WHERE TBNAME EQ 'GGSales'
WHERE NAME EQ 'DOLLARS' OR 'BUDDOLLARS' OR 'UNITS'
ON TABLE HOLD AS GGETFLD FORMAT FOCUS
END
```

例 GGBYFLD データソースの作成

次のリクエストは、GGBYFLD という名前の FOCUS データソースを作成します。

```
APP HOLD IBISAMP
DEFINE FILE SYSCOLUM
FLDDESC/A20=IF NAME EQ 'CATEGORY' THEN 'Category'
      ELSE IF NAME EQ 'PRODUCT' THEN 'Product' ELSE 'N/A';
END
TABLE FILE SYSCOLUM
PRINT FLDDESC
BY NAME
WHERE TBNAME EQ 'GGSales'
WHERE NAME EQ 'CATEGORY' OR 'PRODUCT'
ON TABLE HOLD AS GGBYFLD FORMAT FOCUS
END
```

例 GGACRFLD データソースの作成

次のリクエストは、GGACRFLD という名前の FOCUS データソースを作成します。

```
APP HOLD IBISAMP
DEFINE FILE SYSCOLUM
FLDDESC/A20=IF NAME EQ 'REGION' THEN 'Region'
      ELSE IF NAME EQ 'ST' THEN 'State' ELSE 'N/A';
END
TABLE FILE SYSCOLUM
PRINT FLDDESC
BY NAME
WHERE TBNAME EQ 'GGSales'
WHERE NAME EQ 'REGION' OR 'ST'
ON TABLE HOLD AS GGACRFLD FORMAT FOCUS
END
```

構文 表示フィールドまたはソートフィールドの動的単一選択リストの追加

```
cmd &var.(FIND return_fieldname [,display_fieldname] IN datasource).  
[description.]
```

説明

cmd

フィールドのリストを適用するコマンドです。有効な値は、PRINT、COUNT、SUM、WRITE、ADD、BY、ACROSS です。

&var

フィールド値のリストを提供する変数です。アンパサンド (&) を含めます。

return_fieldname

プロシジャに返す変数値が含まれるフィールドの名前です。

display_fieldname

オートプロンプトフォームに表示する変数値が含まれるフィールドの名前です。

datasource

return_fieldname および display_fieldname で指定したフィールドが含まれるデータソースです。JOIN で使用されるデータソースのクロスリファレンスファイルにフィールドが存在する場合は、そのフィールドが含まれるデータソースの名前を使用します。

注意

- ❑ 動的リストでは、WebFOCUS Client が、指定したデータソースから値を取得するためのリクエストを作成します。指定したデータソースから値を取得するためのすべての環境コマンド (例、SET コマンド) は、WebFOCUS Reporting Server プロファイル、ユーザープロファイル、WebFOCUS Client プロファイルのいずれかで発行する必要があります。
- ❑ レスポンシブオートプロンプトで表示される動的リストの値は、大文字と小文字を区別したソート順で整列されます。HTML オートプロンプトで表示される動的リストの値は、WebFOCUS Reporting Server によって返されるソート順で整列されます。これは、マシンのオペレーティングシステムによって決定されます。

description

オプションとして追加する変数の説明です。

例 表示フィールドおよびソートフィールドの動的単一選択リストの作成

次のリクエストは、GGSALES データソースに対して実行され、FOCUS データソースである GGDETFLD、GGBYFLD、GGACRFLD のフィールド名を取得することにより、3 つの動的単一選択リストを作成します。

```
TABLE FILE GGSALES
HEADING
"Sales Report Summary for: "
"Fields: &SumFlds"
"Sort: &SortBy"
"Across: &Acrs"
" "
SUM &SumFlds.(FIND NAME, FLDDDESC IN GGDETFLD).Sum Fields.
BY &SortBy.(FIND NAME, FLDDDESC IN GGBYFLD).By.
ACROSS &Acrs.(FIND NAME, FLDDDESC IN GGACRFLD).Across.
WHERE REGION NE 'West'
ON TABLE SET PAGE NOPAGE
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT=ARIAL, SIZE=10, SQUEEZE=ON, $
TYPE=HEADING, FONT=ARIAL, STYLE=BOLD, JUSTIFY=CENTER, $
TYPE=HEADING, LINE=1, FONT=ARIAL, STYLE=BOLD, JUSTIFY=CENTER, $
TYPE=TITLE, BACKCOLOR=BLACK, COLOR=WHITE, $
TYPE=ACROSSSTITLE, BACKCOLOR=LIGHT BLUE, COLOR=BLUE, $
TYPE=ACROSSVALUE, BACKCOLOR=LIGHT GRAY, COLOR = GREEN, $
END
```

このリクエストを実行すると、下図のようなオートプロンプトフォームが開いて、SUM、BY、ACROSS コマンドの単一選択リストが表示されます。

The screenshot shows a 'Parameters' dialog box with three dropdown menus: 'Sum Fields' (set to 'Dollars'), 'By' (set to 'Category'), and 'Across' (set to 'Region'). Below these are buttons for 'Run', 'Reset', and 'Clear Output', and a checkbox for 'Run in a new window'.

[Sum Fields] ドロップダウンリストの [Dollars]、[By] ドロップダウンリストの [Category]、[Across] ドロップダウンリストの [Region] を選択すると、次のレポートが生成されます。

Sales Report Summary for:
Fields: DOLLARS
Sort: CATEGORY
Across: REGION

	Region		
	Midwest	Northeast	Southeast
Category			
Coffee	4178513	4164017	4415408
Food	4338271	4379994	4308731
Gifts	2883881	2848289	2986240

構文 ソートフィールドの動的複数選択リストの追加

```
cmd &var.(cmd(FIND return_fieldname [,display_fieldname]
IN datasource)).[description.]
```

説明

cmd

フィールドのリストを適用するコマンドです。有効な値は BY および ACROSS です。選択リストを囲んでコマンドを追加すると、値の複数選択リストが作成されます。

&var

フィールド値のリストを提供する変数です。アンパサンド (&) を含めます。

return_fieldname

プロシジャに返す変数値が含まれるフィールドの名前です。

display_fieldname

オートプロンプトフォームに表示する変数値が含まれるフィールドの名前です。

datasource

return_fieldname および display_fieldname で指定したフィールドが含まれるデータソースです。JOIN で使用されるデータソースのクロスリファレンスファイルにフィールドが存在する場合は、そのフィールドが含まれるデータソースの名前を使用します。

注意

- ❑ 動的リストでは、WebFOCUS Client が、指定したデータソースから値を取得するためのリクエストを作成します。指定したデータソースから値を取得するためのすべての環境コマンド (例、SET コマンド) は、WebFOCUS Reporting Server プロファイル、ユーザプロファイル、WebFOCUS Client プロファイルのいずれかで発行する必要があります。
- ❑ レスポンシブオートプロンプトで表示される動的リストの値は、大文字と小文字を区別したソート順で整列されます。HTML オートプロンプトで表示される動的リストの値は、WebFOCUS Reporting Server によって返されるソート順で整列されます。これは、マシンのオペレーティングシステムによって決定されます。

description

オプションとして追加する変数の説明です。

構文

表示フィールドの動的複数選択リストの追加

```
cmd &var.(AND(FIND return_fieldname [,display_fieldname]  
IN datasource)).[description.]
```

説明

cmd

フィールドのリストを適用するコマンドです。有効な値は、PRINT、COUNT、SUM、WRITE、ADD です。

&var

フィールド値のリストを提供する変数です。アンパサンド (&) を含めます。

AND

複数選択リストを作成する接続子です。

return_fieldname

プロシジャに返す変数値が含まれるフィールドの名前です。

display_fieldname

オートプロンプトフォームに表示する変数値が含まれるフィールドの名前です。

datasource

return_fieldname および display_fieldname で指定したフィールドが含まれるデータソースです。JOIN で使用されるデータソースのクロスリファレンスファイルにフィールドが存在する場合は、そのフィールドが含まれるデータソースの名前を使用します。

注意

- 動的リストでは、WebFOCUS Client が、指定したデータソースから値を取得するためのリクエストを作成します。指定したデータソースから値を取得するためのすべての環境コマンド (例、SET コマンド) は、WebFOCUS Reporting Server プロファイル、ユーザプロファイル、WebFOCUS Client プロファイルのいずれかで発行する必要があります。
- レスポンシブオートプロンプトで表示される動的リストの値は、大文字と小文字を区別したソート順で整列されます。HTML オートプロンプトで表示される動的リストの値は、WebFOCUS Reporting Server によって返されるソート順で整列されます。これは、マシンのオペレーティングシステムによって決定されます。

description

オプションとして追加する変数の説明です。

例**表示フィールドおよびソートフィールドの動的複数選択リストの作成**

次のリクエストは、GGSALES データソースに対して実行され、FOCUS データソースである GGDETFD、GGBYFLD、GGACRFLD のフィールド名を取得することにより、3 つの動的複数選択リストを作成します。

```
TABLE FILE GGSALES
HEADING
"Sales Report Summary for: "
"Fields: &SumFlds"
"Sort: &SortBy"
"Across: &Acrs"
" "
SUM &SumFlds.(AND(FIND NAME, FLDDDESC IN GGDETFD)).Sum Fields.
BY &SortBy.(BY(FIND NAME, FLDDDESC IN GGBYFLD)).By.
ACROSS &Acrs.(ACROSS(FIND NAME, FLDDDESC IN GGACRFLD)).Across.
WHERE REGION NE 'West'
ON TABLE SET PAGE NOPAGE
ON TABLE PCHOLD FORMAT PDF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF, FONT=ARIAL, SIZE=10, SQUEEZE=ON, $
TYPE=HEADING, FONT=ARIAL, STYLE=BOLD, JUSTIFY=CENTER, $
TYPE=HEADING, LINE=1, FONT=ARIAL, STYLE=BOLD, JUSTIFY=CENTER, $
TYPE=TITLE, BACKCOLOR=BLACK, COLOR=WHITE, $
TYPE=ACROSSTITLE, BACKCOLOR=LIGHT BLUE, COLOR=BLUE, $
TYPE=ACROSSVALUE, BACKCOLOR=LIGHT GRAY, COLOR = GREEN, $
END
```

このリクエストを実行すると、下図のようなオートプロンプトフォームが開いて、SUM、BY、ACROSS コマンドの複数選択リストが表示されます。

Parameters

Sum

Select All

Sales

Budget

Units

By

Select All

Category

PRODUCT

Across

Select All

Area

State

None

Run

Reset

Clear Output

☐ Run in a new window

1. Specify values for all parameters.

2. Select the run button to submit the request.

[Dollars]、[Units]、[Category]、[Region] を順に選択すると、次のレポートが生成されます。

Sales Report Summary for:
Fields: DOLLARS AND UNITS
Sort: CATEGORY
Across: REGION

Category	Region					
	Midwest		Northeast		Southeast	
	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales	Dollar Sales	Unit Sales
Coffee	4178513	332777	4164017	335778	4415408	350948
Food	4338271	341414	4379994	353368	4308731	349829
Gifts	2883881	230854	2848289	227529	2986240	234455

オートプロンプト機能のカスタマイズ

ここでは、オートプロンプトの実行時設定およびオートプロンプトフォームの外観のカスタマイズ方法について説明します。

プロシジャでの HTML テンプレートの指定

レポートの実行時に HTML テンプレートの使用を指定する場合は、プロシジャ (FEX) の先頭に次のコードを追加します。

```
-<describe_html>template</describe_html>
```

説明

template

次のいずれかのテンプレート値に設定します。

- ☐ **autoprompt_top** - パラメータをページの上部に横方向に表示します。このテンプレート値がデフォルトです。
- ☐ **autoprompt_top_checked** - autoprompt_top と同一ですが、[新規ウィンドウで実行] のチェックがオンの状態になります。

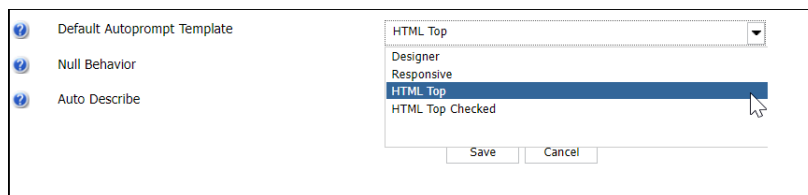
HTML オートプロンプト機能のカスタマイズ

テンプレートファイルを編集して、HTML オートプロンプト機能の外観をカスタマイズすることができます。テンプレートファイル (autoprompt_top.css) は、ibi¥WebFOCUS8¥ibi_html¥javaassist¥ibi¥html¥describe ディレクトリに格納されています。ファイルに変更を加える前に、autoprompt_top.css ファイルのバックアップコピーを作成してください。

バナーをカスタマイズする場合は、イメージを作成して describe ディレクトリに保存し、背景画像プロパティを変更します。次のカスケードスタイルシート (CSS) コードでは、背景画像プロパティが太字で表示されています。

```
#idBannerDiv {
height:41px;
background-image:url(style/logo_banner_TOP.gif);
background-position:top left;
background-repeat:no-repeat;
margin:0px;
margin-top:0px;
cursor:pointer; }
```

オートプロンプト機能を構成するには、WebFOCUS 管理コンソールの [構成] タブの [パラメータのプロンプト] 設定を使用します。[デフォルトオートプロンプトテンプレート] のデフォルト設定は、[デザイン] です。バージョン 8.1 の HTML オートプロンプトテンプレートも使用できます。その場合は、下図のように、[HTML_TOP] または [HTML_TOP_CHECKED] 設定を選択して構成します。



WebFOCUS 管理コンソールの [構成] タブの [パラメータのプロンプト] 設定を使用して、さまざまなテンプレートを選択できます。[オートプロンプト] ページの選択に使用する設定は、[デフォルトオートプロンプトテンプレート] (IBI_DESCRIBE_TEMPLATES) 設定で選択したオートプロンプトの実装タイプによって異なります。詳細は、156 ページの「[オートプロンプトの構成](#)」を参照してください。

HTML テンプレートには、次の 2 つのオプションがあります。

- ☐ **HTML_Top** HTML ベースの実装および autoprompt_top.html テンプレートを使用します。このテンプレートでは、パラメータがページ上部に横方向に表示されます。
- ☐ **HTML_Top_Checked** HTML ベースの実装および autoprompt_top_checked.html テンプレートを使用します。このテンプレートでは、新規ウィンドウのチェックボックスで [Run] が事前選択され、デフォルト設定ですべてのレポートが新規ウィンドウで開きます。

このリストには、カスタマイズしたテンプレートを追加することができます。詳細は、『TIBCO WebFOCUS セキュリティ管理ガイド』を参照してください。

プロシジャでの HTML テンプレートの指定

レポートの実行時に HTML テンプレートの使用を指定する場合は、プロシジャ (FEX) の先頭に次のコードを追加します。

```
-<describe_html>template</describe_html>
```

説明

[template](#)

次のいずれかのテンプレート値に設定します。

- ☐ **autoprompt_top** - パラメータをページの上部に横方向に表示します。このテンプレート値がデフォルトです。
- ☐ **autoprompt_top_checked** - autoprompt_top と同一ですが、[新規ウィンドウで実行] のチェックがオンの状態になります。

レポートを実行する URL でのプロンプトレベルの指定

オートプロンプトの設定は、WebFOCUS Reporting Server のアプリケーションディレクトリに格納されたレポートを実行する URL に、直接指定することができます。パラメータプロンプトに対して指定可能な WebFOCUS Client 構成設定には、次のものがあります。

セルフサービス (IBI_WFDESCRIBE_DEFAULT)

セルフサービスレポートのオートプロンプトを有効または無効にします。利用可能な値には、次のものがあります。

- ☐ **オフ** オートプロンプトを無効にします。これがデフォルト値です。
- ☐ **デフォルト値で実行 (XMLRUN)** -DEFAULT コマンドが設定されていない場合、オートプロンプト画面を表示します。変数の初期値がすべて設定されていれば、その値でプロシジャを実行します。
- ☐ **常にプロンプトを表示 (XMLPROMPT)** -DEFAULT コマンドが設定されている場合でも、オートプロンプト画面を表示します。すべての変数が -DEFAULT で指定されている場合も、オートプロンプト画面を表示します。-DEFAULT コマンドで指定された変数は、オートプロンプト画面を表示しません。
- ☐ **XML の表示 (構文エラー確認付きデバッグ) (XML)** 変数が記述された XML ドキュメントをブラウザに表示します。この設定は内部的に使用され、デバッグおよび構文エラー確認用としてのみ使用することをお勧めします。
- ☐ **XML の表示 (デバッグ) (XMLCHECK)** 変数が記述された XML ドキュメントをブラウザに表示します。この設定は内部的に使用され、デバッグ用としてのみ使用することをお勧めします。

注意：BI Portal では、IBIMR_prompting という別の変数設定が使用されます。

レポートを実行するリクエストが WebFOCUS Client に送信される際に、WebFOCUS Reporting Server に送信されるリクエストに次のパラメータが追加されます。

WFDESCRIBE=value

説明

WFDESCRIBE

パラメータプロンプトの評価に必要なレベルで WebFOCUS Reporting Server と通信します。

value

WebFOCUS Client 構成のパラメータプロンプト設定の値です。レポートリクエストを実行する際に、IBI_WFDESCRIBE_DEFAULT 設定を使用して、WebFOCUS Client に送信する URL で指定することができます。

WebFOCUS Client 変数の `IBIC_server` を使用して、レポートのサーバノードを指定することができます。`IBIC_server` についての詳細は、『TIBCO WebFOCUS セキュリティ管理ガイド』を参照してください。次の例は、WebFOCUS Reporting Server プロシジャノード `EDASERVE` で「`salessummary`」という名前の WebFOCUS Reporting Server プロシジャを実行するために使用される URL を示しています。`IBI_WFDESCRIBE_DEFAULT` は `XMLRUN` に設定されています。

```
http://hostname[:port]/ibi_apps/WFServlet?IBIF_ex=salessummary
&IBIC_server=EDASERVE&IBI_WFDESCRIBE_DEFAULT=XMLRUN
```

注意：パラメータのプロンプト構成設定に加えた変更は、開発環境またはテスト環境で評価し、その評価には既存のレポートやアプリケーションの実行を含める必要があります。

`IBI_WFDESCRIBE_DEFAULT` についての詳細は、『TIBCO WebFOCUS セキュリティ管理ガイド』を参照してください。

オートプロンプトの内部 XML へのプロシジャ説明の追加

オートプロンプト機能が使用する内部 XML に含めるプロシジャの説明を追加することができます。これは、Web サービスアプリケーションを構成するプロシジャなど、アプリケーションの統合に役立つオートプロンプトフォームについての情報を提供します。説明用のコードは、レポートリクエストより前に記述する必要があります。

構文 プロシジャ説明の追加

```
--<description>text</description>
```

説明

`text`

プロシジャの説明です。説明を複数の行に記述する場合は、1 行ごとに `<description>` と `</description>` のタグを使用する必要があります。

例 プロシジャ説明の追加

次の例は、プロシジャの説明を追加する方法を示しています。このタグは、返されたソースコードにのみ表示されます。

```

-<description>This procedure reports on Inventory </description>
-<description>by state, storename and product name. </description>
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY STATE BY STORENAME BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE'
WHERE STORENAME EQ '&STORENAME'
WHERE PRODNAME EQ '&PRODNAME'
END

```

デフォルト TIBCO WebFOCUS ページへのレポートの表示

デフォルトの WebFOCUS ページを使用すると、レポートを簡単に表示することができます。ユーザはレポートを表示するページを設計する必要はありません。WebFOCUS により、必要なタグが組み込まれた HTML ページが自動的に作成されます。ページに表示するレポート出力のフォーマットは、HTML <TABLE タグで設定されます。

下図は、デフォルトの WebFOCUS ページを使用してレポートを表示した例を示しています。

Product Name	Quantity In Stock
110 VHS-C Camcorder 20 X	352000
120 VHS-C Camcorder 40 X	202400
150 8MM Camcorder 20 X	786852
2 Hd VCR LCD Menu	1162836
250 8MM Camcorder 40 X	360438
650DL Digital Camcorder 150 X	86188
AR2 35MM Camera 8 X	57495
Combo Player - 4 Hd VCR + DVD	946890
DVD Upgrade Unit for Cent. VCR	26268
QX Portable CD Player	1936000
R5 Micro Digital Tape Recorder	141290
ZC Digital PDA - Standard	165000
ZT Digital PDA - Commercial	1491000

プロシジャ内に ON TABLE HOLD FORMAT HTML コマンドを使用して動的レポートを作成し、その出力を保存する場合は、後から呼び出して表示できる静的レポートを作成します。

ユーザがリアルタイムの情報を必要としないアプリケーションでは、静的レポートを作成することによりシステムのオーバーヘッドを大幅に削減することができます。その後、レポートはオフピークの時間帯に更新することができます。

手順 デフォルト TIBCO WebFOCUS ページに動的レポートを表示するには

1. 動的レポートを生成するプロシジャを作成します。
2. ユーザがレポートの実行に使用する起動ページを作成します。
3. 起動ページからレポートを実行します。レポートは、完全な HTML ファイルとしてブラウザに返されます。このファイルのソースを表示するには、ブラウザの各種オプションを使用します。たとえば Microsoft Internet Explorer では [表示]、[ソース] を選択します。

例 デフォルトの WebFOCUS ページでの動的レポートの表示

注意：この例で作成するファイルの格納場所についての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

1. 「STATEINV」という名前のプロシジャを作成します。このプロシジャは、動的レポートを生成します。このレポートが WebFOCUS Reporting Server にアクセスできる状態にしておく必要があります。

プロシジャ - STATEINV.FEX

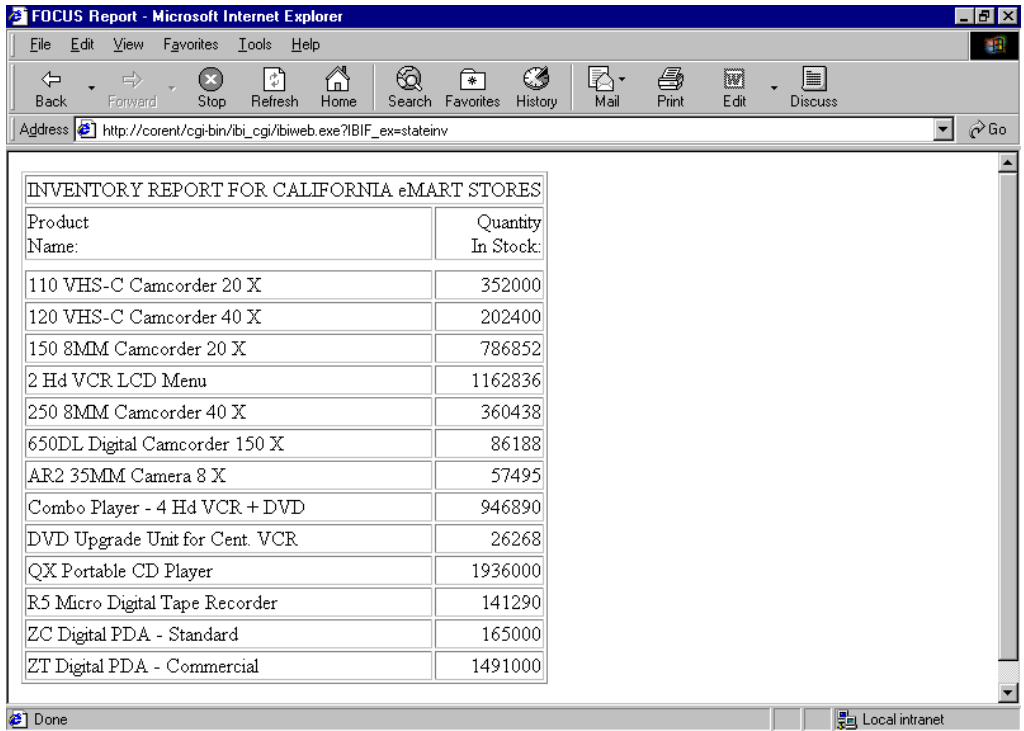
```
SET PAGE-NUM = OFF
TABLE FILE CENTORD
SUM QTY_IN_STOCK BY PRODNAME
ON TABLE SUBHEAD
"INVENTORY REPORT FOR CALIFORNIA eMART STORES"
WHERE STATE EQ 'CA'
WHERE STORENAME EQ 'eMart'
END
```

2. ユーザがレポートの実行に使用する起動ページを作成します。この例は、「RUNDYNAM」という名前の簡単な起動ページです。このページは、WebFOCUS Servlet を使用します。

起動ページ - RUNDYNAM.HTM

```
<HTML>
<BODY>
<A HREF="http://server1/servlet/WFServlet?IBIF_ex=stateinv">Click here
to run an inventory report for eMart in California.</A>
</BODY>
</HTML>
```

3. 起動ページを実行し、リンクをクリックします。デフォルトの WebFOCUS ページに動的レポートが表示されます。



Product Name	Quantity In Stock
110 VHS-C Camcorder 20 X	352000
120 VHS-C Camcorder 40 X	202400
150 8MM Camcorder 20 X	786852
2 Hd VCR LCD Menu	1162836
250 8MM Camcorder 40 X	360438
650DL Digital Camcorder 150 X	86188
AR2 35MM Camera 8 X	57495
Combo Player - 4 Hd VCR + DVD	946890
DVD Upgrade Unit for Cent. VCR	26268
QX Portable CD Player	1936000
R5 Micro Digital Tape Recorder	141290
ZC Digital PDA - Standard	165000
ZT Digital PDA - Commercial	1491000

手順

デフォルトの WebFOCUS ページに静的レポートを表示するには

- 次の手順を実行してプロシジャを作成します。
 - レポート出力を格納する HTML ファイルを識別し、そのファイルを指定された場所に保存します。たとえば、Windows では FILEDEF コマンドを使用します。プラットフォームに固有のコマンドについての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

Web サーバがファイルの場所を特定し、そのファイルをブラウザに表示できるように、Web サーバのエイリアス配下の場所を指定する必要があります。

- 次のコマンドを追加します。

ON TABLE HOLD FORMAT HTML AS report

説明

report - レポート出力を格納する仮想ファイルの名前です。ファイル名として許容される長さは、1 から 8 バイトです。拡張子は含めないでください。このファイルはディスクには保存されません。

2. プロシジャを実行し、HTML ファイルとしてレポートを作成します。この場合、次のような方法があります。

- ❑ WebFOCUS Client を呼び出す。使用例については、218 ページの「[デフォルト WebFOCUS ページでの静的レポートの表示](#)」を参照してください。
- ❑ ReportCaster を使用して、レポートの実行をスケジュールする。詳細は、ReportCaster のマニュアルを参照してください。
- ❑ WebFOCUS で、次のデフォルトディレクトリにアクセスする。

Windows `install_drive:\ibi\svr82\wfs\bin`

UNIX `/ibi/srv82/wfs/bin`

次のコマンドを発行します。

`edastart -t`

WebFOCUS プロンプト `>>` からプロシジャを実行します。

`ex procedure_name`

次のように入力して WebFOCUS ディレクトリに戻ります。

`fin`

3. 起動ページを作成し、レポートに移動するためのハイパーリンクを追加します。

例

デフォルト WebFOCUS ページでの静的レポートの表示

この例は、WebFOCUS を Windows で実行する場合の手順です。他のプラットフォームを使用する場合は、FILEDEF コマンドをプラットフォームに対応したコマンドに読み替えてください。詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

注意：この例で作成するファイルの格納場所についての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

1. HTML ファイルとしてレポートを保存するプロシジャを作成します。この例では、「STATDATA」という名前のプロシジャを使用します。左側のアルファベットは、コードのそれぞれの説明に対応しています。

プロシジャ - STATDATA.FEX

```

a. FILEDEF STATRPT DISK APP/STATRPT.HTM
   TABLE FILE GGORDER
   HEADING
   "Orders Summary"
   SUM QUANTITY AS 'Units Ordered'
   BY ORDER_DATE AS 'Order Date'
   BY PRODUCT_DESCRIPTION
   WHERE ORDER_DATE EQ '01/01/96'
b. ON TABLE HOLD FORMAT HTML AS STATRPT
   END

```

- a. このコマンドは、STATRPT ファイルを割り当て、指定したディレクトリにそのファイルを保存します。
 - b. このコマンドは、レポート出力を「STATRPT.HTM」という名前の HTML ファイルで保存します。
2. プロシジャの STATDATA を実行して、STATRPT を作成、保存します。これを実行する方法の 1 つとして、WebFOCUS Client を呼び出す方法があります。

次の簡単な HTML ページは、Servlet を使用して STATDATA を実行します。管理者は、このようなページを使用して、ユーザが後からアクセスできる静的レポートを作成することができます。

```

<HTML>
<BODY>
<A HREF="/ibi_apps/WFServlet?IBIF_ex=statdata">Click here
to run STATDATA and create a static report stored as STATRPT.HTM.</A>
</BODY>
</HTML>

```

WebFOCUS は、HTML ファイルが保存されたことを示すメッセージを表示します。

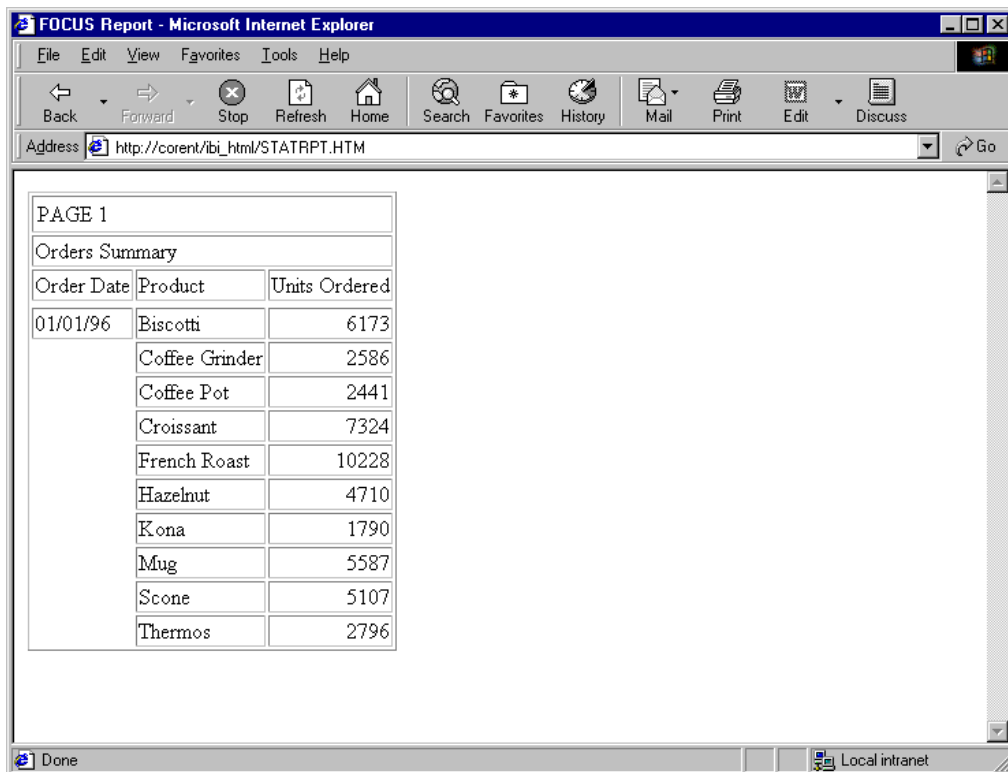
3. ユーザがレポートの実行に使用する起動ページを作成します。この例は、「STATIC」という名前の簡単な起動ページです。

```

<HTML>
<BODY>
<A HREF="STATRPT.HTM">Click here to view the Orders Summary report</A>
</BODY>
</HTML>

```

4. 起動ページを実行し、リンクをクリックします。デフォルトの WebFOCUS ページに静的レポートが表示されます。



レポート表示用 HTML ページの設計

ダイアログマネージャの -HTMLFORM コマンドを使用して、レポートを表示する HTML ページを設計することができます。ページのスタイルおよびフォーマットの設定を制御することにより、レポートの外観を自由にカスタマイズすることができます。カスタム HTML ページには、フレーム、テーブル、グラフィックイメージなどの HTML 要素を含めることができます。ただし、この HTML ページには、必須の HTML タグをすべて含める必要があります。また、WebFOCUS レポート出力の挿入場所を指定した特別な HTML コメントを含める必要があります。

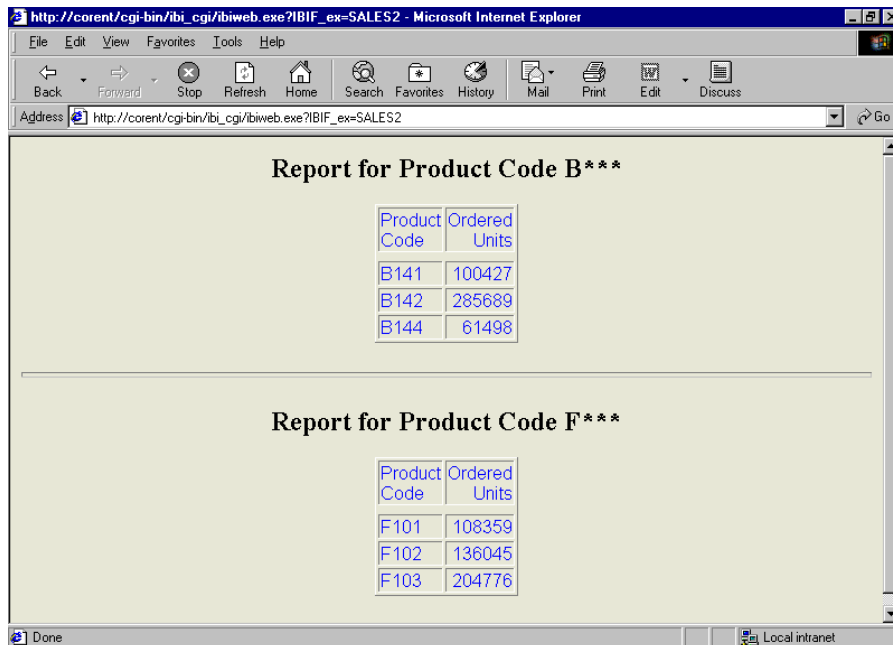
次のいずれかを実行することができます。

- ❑ 表示ページを定義した既存の外部 HTML ファイルを呼び出す。この方法は、プロシジャ内で -HTMLFORM filename コマンドを使用します。この方法では、プロシジャと HTML ページは、それぞれ別のファイルに格納されます。
- ❑ -HTMLFORM BEGIN および -HTMLFORM END コマンドを使用して、プロシジャ内にカスタム HTML ページを作成する。ここでは、このタイプのページを「インライン HTML」ページと呼びます。

1 ページに 1 つまたは複数のページを表示することができます。また、動的レポートまたは静的レポートのどちらでも表示することができます。

ドキュメントレベルの HTML タグを使用せずにグラフ出力を生成し、インタラクティブレポートおよびグラフの機能に必要な JavaScript ライブラリをロードすると、1 ページに複数の HTML5 グラフを作成することができます。詳細は、231 ページの「[HTML ページでの複数 HTML5 グラフの表示](#)」を参照してください。

下図は、カスタム HTML ページに表示した複数レポートを示しています。



-HTMLFORM コマンドおよびその機能についての詳細は、400 ページの「[プロシジャによる HTML Web ページの拡張](#)」を参照してください。

参照

-HTMLFORM 使用上の注意

- -HTMLFORM を使用して HTML レポートを表示するための HTML ページを作成し、 - HTMLFORM 内にカスタムスタイルコードを指定した場合、そのコードが、マルチドリルダウン、HFREEZE、アコーディオン、BYTOC、および Web ビューアの JavaScript ベースのレポート機能と競合する場合があります。特定のスタイル設定 (例、レポート表示領域のサイズ変更、スクロールバーの配置) が必要なフレーム内に WebFOCUS インタラクティブ JavaScript ベースの機能を表示する HTML ページを作成する場合は、App Studio HTML キャンバスを使用することをお勧めします。

手順

既存の HTML ページに動的レポートを表示するには

1. レポート出力を組み込む HTML ページを作成します。次のコメントを使用して、レポート出力を表示する位置を指定します。

```
!IBI.FIL.report;
```

または

```
<!--WEBFOCUS TABLE report-->
```

説明

report

レポート出力を格納する仮想ファイルの名前です。このファイルは、手順 2 で作成されます。

注意：HTML コメント行は「->」で閉じる必要があります。また、単一の「>」で閉じることができますが、その場合はコメント行に他の HTML タグを含めることはできません。

WebFOCUS Reporting Server がこのページの場所を特定できるように、APP PATH または EDAPATH を設定しておく必要があります。

このページが表示される際に、このコメントがレポート出力で置換されます。

2. 次の手順を実行し、プロシジャを作成します。

- a. 次のコマンドを追加します。

```
ON TABLE HOLD FORMAT HTMTABLE AS report
```

説明

report - レポート出力を格納する仮想ファイルの名前です。ファイル名として許容される長さは、1 から 8 バイトです。拡張子を含めないでください。このファイルはディスクには保存されません。

- b. この手順の最後で、ダイログマネージャのコマンドを追加します。

-HTMLFORM filename

説明

filename - レポート出力を組み込む HTML ページの名前です。このページは、手順 1 で作成します。

3. ユーザがレポートの実行に使用する起動ページを作成します。WebFOCUS は、設計済みの HTML ページにレポート出力を組み込み、HTML ファイルをブラウザに返します。

例

既存の HTML ページでの動的レポートの表示

注意：この例で作成するファイルの格納場所についての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

1. 「RPTPG.HTM」という名前の HTML ページを作成します。WebFOCUS Reporting Server がこのページの場所を特定できるように、APP PATH または EDAPATH を設定しておく必要があります。検索パスについての詳細は、23 ページの「[TIBCO WebFOCUS のアプリケーションロジック](#)」を参照してください。

左側のアルファベットは、コードのそれぞれの説明に対応しています。

HTML ページ - RPTPG.HTM

```
<HTML>
<BODY BGCOLOR="#CCCCCC">
<FONT FACE="Arial" COLOR="Black">
<H2>Orders Summary</H2>
a. !IBI.FIL.ORDERS;
</BODY>
</HTML>
```

- a. WebFOCUS は HTML コメントを読み込み、そのコメントを ORDERS (手順 2) に格納しておいたレポート出力で置換します。

2. 「ORDERS」という名前のプロシジャを作成します。左側のアルファベットは、コードのそれぞれの説明に対応しています。

プロシジャ - ORDERS.FEX

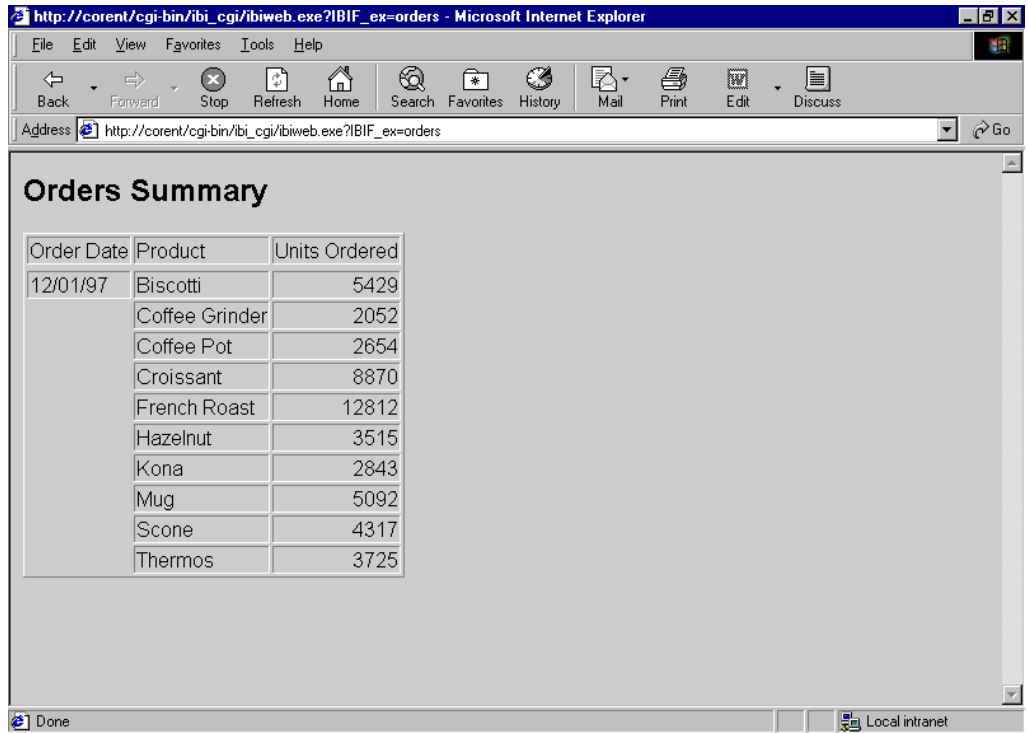
```
SET PAGE-NUM = OFF
TABLE FILE GGORDER
SUM QUANTITY AS 'Units Ordered'
BY HIGHEST 1 ORDER_DATE AS 'Order Date'
BY PRODUCT_DESCRIPTION
a. ON TABLE HOLD FORMAT HTMTABLE AS ORDERS
END
b. -HTMLFORM RPTPG
```

- a. このコマンドは、レポート出力を HTML フォーマットの仮想ファイルに格納します。そのファイルには「ORDERS」という名前が付けられます。
 - b. このコマンドは、ORDERS に格納しておいたレポート出力を「RPTPG.HTM (手順 1)」という HTML ページに送信します。
3. ユーザがレポートの実行に使用する起動ページを作成します。次の例では、「RUNRPT.HTM」という名前の簡単な起動ページが Servlet を使用します。

起動ページ - RUNRPT.HTM

```
<HTML>
<BODY>
<A HREF="/ibi_apps/WFServlet?IBIF_ex=orders">Click here for the Orders
Summary report.</A>
</BODY>
</HTML>
```

4. 起動ページを実行し、リンクをクリックします。動的レポートが、カスタム HTML ページに表示されます。



構文 インライン HTML ページでのレポートの表示

```
-HTMLFORM BEGIN
html_code
html_code
html_code
.
.
.-HTMLFORM END
```

説明

html_code

標準の HTML コードの行です。行の長さは、最大で 80 バイトです。

例 インライン HTML ページでの動的レポートの表示

注意: この例で作成するファイルの格納場所についての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

1. 「ORDERS2」という名前のプロシジャを作成します。左側のアルファベットは、コードのそれぞれの説明に対応しています。

プロシジャ - ORDERS2

- ```

SET PAGE-NUM = OFF
TABLE FILE GGORDER
SUM QUANTITY AS 'Units Ordered'
BY HIGHEST 1 ORDER_DATE AS 'Order Date'
BY PRODUCT_DESCRIPTION
a. ON TABLE HOLD FORMAT HTMTABLE AS ORDERS2
END
b. -HTMLFORM BEGIN
<HTML><HEAD><STYLE>TD {FONT-FAMILY: ARIAL; COLOR:
BLUE;}</STYLE></HEAD>
<BODY BGCOLOR="FAEBD7"><DIV align="left">
c. !IBI.FIL.ORDERS2;
</DIV></BODY></HTML>
b. -HTMLFORM END

```
- a. このコマンドは、レポート出力を HTML フォーマットの仮想ファイルに保存します。このファイルには「ORDERS2」という名前が付けられます。
  - b. これらのコマンドは、スタイルおよびフォーマットを定義するインライン HTML ページの境界を設定します。これらのコマンド間に記述された行は、標準の HTML コードです。各行の文字数を 80 バイト以上にすることはできません。
  - c. WebFOCUS は HTML コメントを読み込み、そのコメントを ORDERS に格納しておいたレポート出力で置換します。
2. ユーザがレポートの実行に使用する起動ページを作成します。次の例では、「RUNRPT.HTM」という名前の簡単な起動ページが Servlet を使用します。

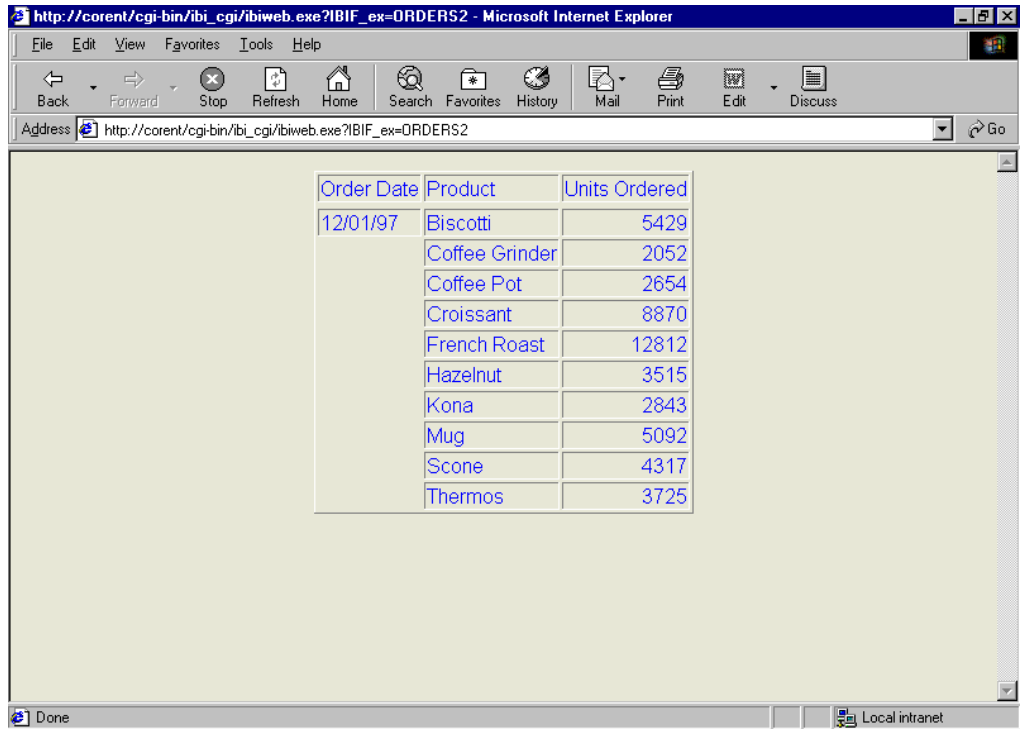
### 起動ページ - RUNRPT.HTM

```

<HTML>
<BODY>
Click here for the
Orders Summary report.
</BODY>
</HTML>

```

3. 起動ページを実行し、リンクをクリックします。動的レポートが、カスタムインライン HTML ページに表示されます。



## 例

### 既存の HTML ページでの 2 つの動的レポートの表示

**注意：**この例で作成するファイルの格納場所についての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

1. 「FIRST.HTM」という名前の HTML ページを作成します。WebFOCUS Reporting Server がこのページの場所を特定できるように、APP PATH または EDAPATH を設定しておく必要があります。

左側のアルファベットは、コードのそれぞれの説明に対応しています。

## HTML ページ - FIRST.HTM

```

 <HTML>
 <BODY BGCOLOR="#CCCCCC">
a. <H2>Report for Product Code B***</H2>
b. <!--WEBFOCUS TABLE UPPER-->
 <HR SIZE=5>
a. <H2>Report for Product Code F***</H2>
b. <!--WEBFOCUS TABLE LOWER-->
 </BODY>
 </HTML>

```

- a. このテキストが、HTML ページに表示した 2 つのレポートを識別します。
- b. WebFOCUS が、これらの HTML コメントを読み込みます。UPPER および LOWER は、手順 2 で仮想ファイルに割り当てられる名前です。HTML ページでは、WebFOCUS がこれらのコメントをレポート出力で置換します。

**注意：**HTML コメント行は「->」で閉じる必要があります。また、単一の「>」で閉じることできますが、その場合はコメント行に他の HTML タグを含めることはできません。

2. 「TWOSEALS」という名前のプロシジャを作成します。左側のアルファベットは、コードのそれぞれの説明に対応しています。

## プロシジャ - TWOSEALS.FEX

```

 SET PAGE-NUM = OFF
 TABLE FILE GGORDER
 SUM QUANTITY BY PCD
 IF PCD EQ 'B$$$'
a. ON TABLE HOLD FORMAT HTMTABLE AS UPPER
 END
 TABLE FILE GGORDER
 SUM QUANTITY BY PCD
 IF PCD EQ 'F$$$'
a. ON TABLE HOLD FORMAT HTMTABLE AS LOWER
 END
b. -RUN
c. -HTMLFORM FIRST

```

- a. これらのコマンドは、各リクエストからレポート出力を生成し、HTML フォーマットの仮想ファイルに保存します。これらのファイルには、「UPPER」および「LOWER」という名前が付けられます。
- b. このコマンドは、レポートリクエストを実行します。
- c. このコマンドは、レポート出力を「FIRST.HTM」という名前の HTML ページに送信します。このページは、手順 1 で作成します。
3. ユーザがレポートの実行に使用する起動ページを作成します。次の例の「MULTIRPT.HTM」という名前の簡単な起動ページは、Servlet を使用します。

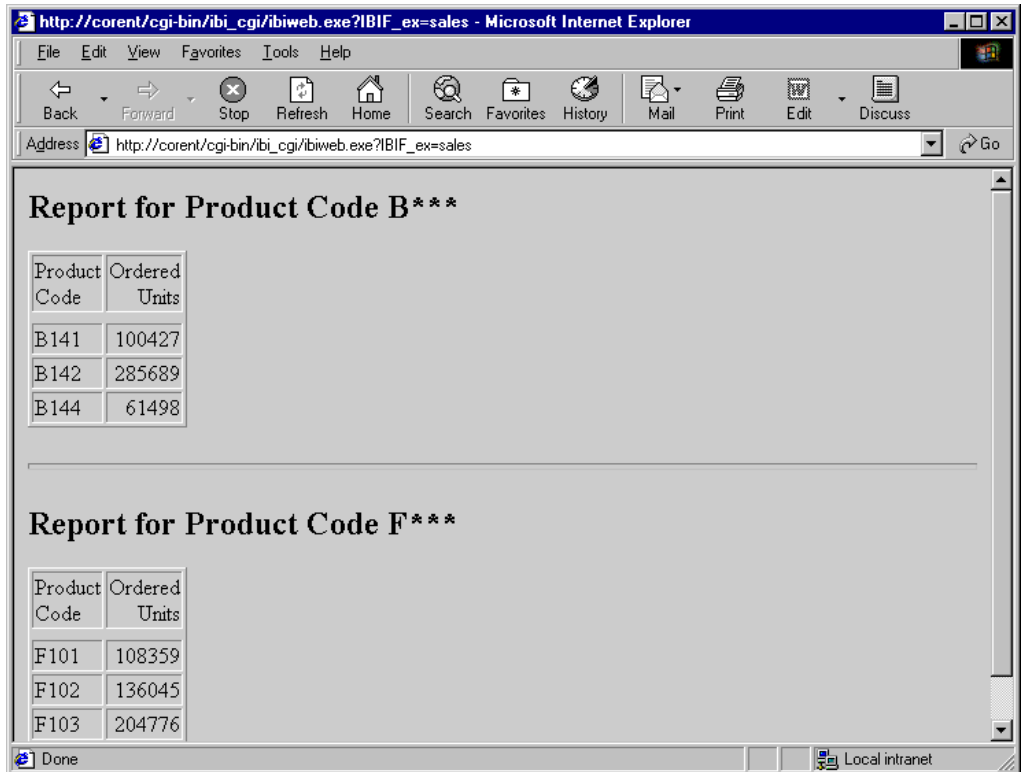
## 起動ページ - MULTIRPT.HTM

```

<HTML>
<BODY>
Click here for sales
reports for two product codes.
</BODY>
</HTML>

```

4. 起動ページを実行し、リンクをクリックします。2つの動的レポートが、カスタム HTML ページに表示されます。



## 例

## インライン HTML ページでの2つの動的レポートの表示

**注意：**この例で作成するファイルの格納場所についての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

1. 「SALES2」という名前のプロシジャを作成します。左側のアルファベットは、コードのそれぞれの説明に対応しています。

## プロシジャ - SALES2.FEX

```

SET PAGE-NUM = OFF
TABLE FILE GGORDER
SUM QUANTITY BY PCD
IF PCD EQ 'B$$$'
a. ON TABLE HOLD FORMAT HTMTABLE AS UPPER
END
TABLE FILE GGORDER
SUM QUANTITY BY PCD
IF PCD EQ 'F$$$'
a. ON TABLE HOLD FORMAT HTMTABLE AS LOWER
END
b. -RUN
c. -HTMLFORM BEGIN
<HTML><HEAD><STYLE>TD {FONT-FAMILY: ARIAL; COLOR:
BLUE;}</STYLE></HEAD>
<BODY BGCOLOR="FAEBD7"><DIV align="left">
<H2>Report for Product Code B***</H2>
d. <!--WEBFOCUS TABLE UPPER-->
<HR SIZE=5>
<H2>Report for Product Code F***</H2>
d. <!--WEBFOCUS TABLE LOWER-->
</DIV></BODY></HTML>
c. -HTMLFORM END

```

- a. これらのコマンドは、各リクエストからレポート出力を生成し、HTML フォーマットの仮想ファイルに保存します。これらのファイルには、「UPPER」および「LOWER」という名前が付けられます。
- b. このコマンドは、レポートリクエストを実行します。
- c. これらのコマンドは、スタイルおよびフォーマットを定義するインライン HTML ページの境界を設定します。これらのコマンド間に記述された行は、標準の HTML コードです。各行の文字数を 80 バイト以上にすることはできません。
- d. WebFOCUS が、これらの HTML コメントを読み込みます。UPPER および LOWER は、仮想ファイルに割り当てられる名前です。HTML ページでは、WebFOCUS がこれらのコメントを指定されたレポート出力で置換します。

**注意：**HTML コメント行は「->」で閉じる必要があります。また、単一の「>」で閉じることできますが、その場合はコメント行に他の HTML タグを含めることはできません。

2. ユーザがレポートの実行に使用する起動ページを作成します。次の例の「MULTIRPT.HTM」という名前の簡単な起動ページは、Servlet を使用します。

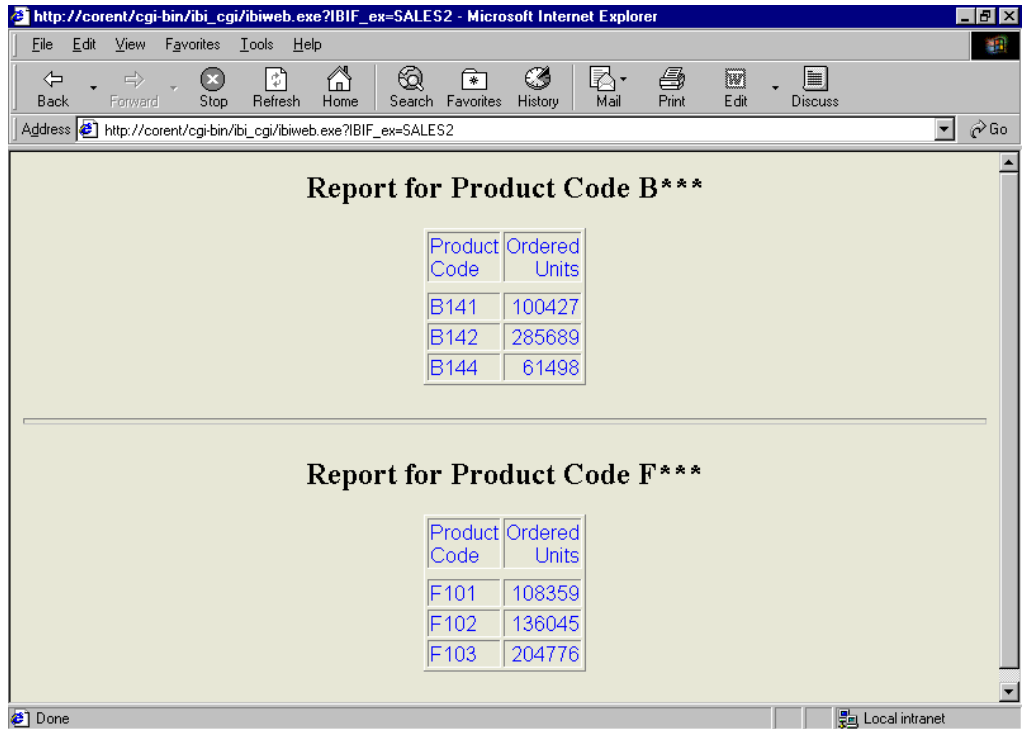
## 起動ページ - MULTIRPT.HTM

```

<HTML>
<BODY>
Click here for sales
reports for two product codes.
</BODY>
</HTML>

```

3. 起動ページを実行し、リンクをクリックします。2つの動的レポートが、カスタムインライン HTML ページに表示されます。



## 構文

### HTML ページでの複数 HTML5 グラフの表示

出力結果が有効な HTML ドキュメントになるように、-HTMLFORM に HTML5 グラフ (HOLD FORMAT JSCHART) を追加するには、ドキュメントレベルの HTML タグ (例、<html>、<head>、<body>) を使用しない HTML5 グラフ出力を生成する必要があります。このようなグラフ出力タイプは、次のコマンドを使用して生成することができます。

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

### 説明

#### OFF

ドキュメントレベルの HTML タグを含む完全な HTML レポート出力を生成します。これがデフォルト値です。

#### ON

ドキュメントレベルのタグを使用せずに、HTML フォーマットでレポート出力を生成します。-HTMLFORM で使用される HTML5 グラフを作成する場合は、この設定を使用する必要があります。

**注意：**SET EMBEDDABLE=ON は、HTML レポート出力および Java ベースのグラフフォーマットにも影響します。これらのフォーマットは、HOLD FORMAT HTMTABLE の使用に相当します。

埋め込み HTML5 グラフ出力では、グラフエンジン用に JavaScript コードをロードするために必要な `<script>` タグを使用しません (HTML ドキュメントでは、このコードのロードは 1 回のみ必要)。これは、-HTMLFORM で指定した HTML で実行する必要があります。実行するには、IBI.OBJ.IBIHEADJS トークンを -HTMLFORM に追加します。これにより、インタラクティブレポートおよびグラフの機能 (グラフエンジンを含む) で必須のすべての JavaScript ライブラリをロードするために必要なコードに自動的に展開されます。

### 例

#### HTML ページでの 2 つの HTML5 グラフの表示

次の HTML ページには、2 つの埋め込み HTML5 グラフが表示されます (SET EMBEDDABLE = ON)。ブラウザでのグラフ描画に必要な JavaScript ライブラリをロードするために、! IBI.OBJ.IBIHEADJS トークンが HTML の `<head>` タグに追加されています。

```

SET EMBEDDABLE=ON

GRAPH FILE GGSales
SUM UNITS BY CATEGORY
ON GRAPH SET LOOKGRAPH BAR
ON GRAPH HOLD FORMAT JSCHART AS GRAPH1
ON GRAPH SET STYLE *
type=data, column=n1, bucket=x-axis, $
type=data, column=n2, bucket=y-axis, $
END

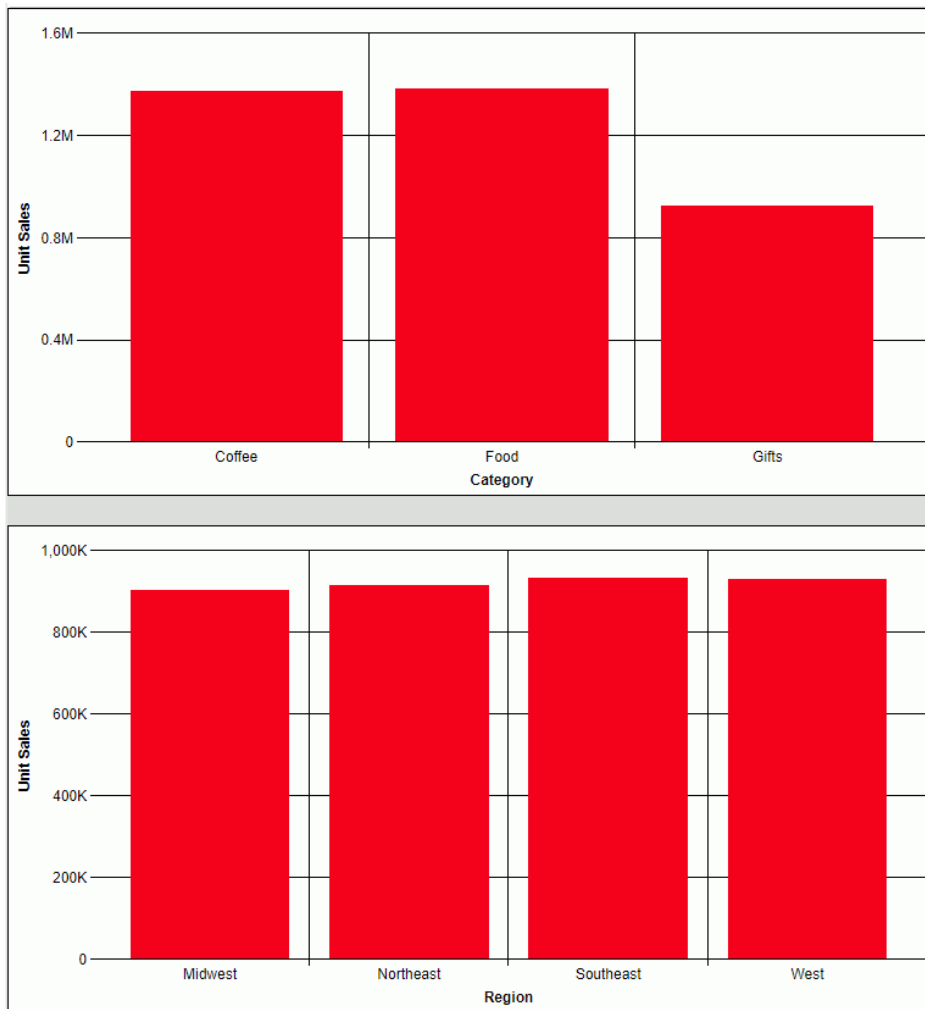
GRAPH FILE GGSales
SUM UNITS BY REGION
ON GRAPH SET LOOKGRAPH BAR
ON GRAPH HOLD FORMAT JSCHART AS GRAPH2
ON GRAPH SET STYLE *
type=data, column=n1, bucket=x-axis, $
type=data, column=n2, bucket=y-axis, $
END

-HTMLFORM BEGIN
<html>
<head>
<title>Two HTML5 graphs in an HTMLFORM</title>
!IBI.OBJ.IBIHEADJS;
</head>
<body>
!IBI.FIL.GRAPH1;

!IBI.FIL.GRAPH2;
</body>
</html>
-HTMLFORM END

```

下図は、出力結果を示しています。



## 手順

既存の HTML ページに静的レポートを表示するには

次の手順を実行して、プロシジャを作成します。

1. レポート出力とカスタム HTML ページを組み込む HTML ファイルを割り当てます。

たとえば、Windows でのコマンドは次のようになります。

```
FILEDEF htmlpage DISK app/htmlpage.HTM
```

説明

#### htmlpage

レポート出力とカスタム HTML ページを組み込むファイルの名前です。ファイル名として許容される長さは、1 から 8 バイトです。

#### app¥

ファイルを保存するアプリケーションディレクトリです。

プラットフォームに固有のコマンドについての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

2. 次のコマンドを追加します。

```
ON TABLE HOLD FORMAT HTMTABLE AS report
```

説明

#### report

レポート出力を格納する仮想ファイルの名前です。ファイル名として許容される長さは、1 から 8 バイトです。拡張子は含めないでください。このファイルはディスクには保存されません。

3. この手順の最後で、ダイログマネージャのコマンドを追加します。

```
-HTMLFORM filename SAVE AS htmlpage
```

説明

#### filename

レポート出力を組み込むカスタム HTML ページの名前です。このページは、手順 2 で作成します。WebFOCUS Reporting Server がこのページの場所を特定できるように、APP PATH または EDAPATH を設定しておく必要があります。

#### htmlpage

レポート出力とカスタム HTML ページを組み込む HTML ファイルの名前です。Web サーバがこのファイルの場所を特定できる状態にしておく必要があります。

4. カスタム HTML ページを作成します。次のコメントを追加して、レポート出力を表示する位置を指定します。

```
!IBI.FIL.report;
```

または

```
<!--WEBFOCUS TABLE report-->
```

## 説明

### report

レポート出力を格納する仮想ファイルの名前です。これは、手順 1 で作成するファイルです。

**注意：**HTML コメント行は「->」で閉じる必要があります。また、単一の「>」で閉じることができますが、その場合はコメント行に他の HTML タグを含めることはできません。

WebFOCUS Reporting Server がこのページの場所を特定できるように、APP PATH または EDAPATH を設定しておく必要があります。

HTML ページでは、WebFOCUS がこれらのコメントをレポート出力で置換します。

5. 手順 1 のプロシジャを実行し、カスタム HTML ページに静的レポートを作成します。
6. 起動ページを作成し、レポートに移動するためのハイパーリンクを追加します。

## 例

### 既存の HTML ページでの静的レポートの表示

この例は、WebFOCUS を Windows で実行する場合の手順です。他のプラットフォームを使用する場合は、FILEDEF コマンドをプラットフォームに対応したコマンドに読み替えてください。詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

**注意：**この例で作成するファイルの格納場所についての詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

### 既存の HTML Web ページでの静的レポートの表示

1. 「CONTACTS」という名前のプロシジャを作成します。左側のアルファベットは、コードのそれぞれの説明に対応しています。

#### プロシジャ - CONTACTS.FEX

- a. 

```
FILEDEF WEBPAGE DISK APPDIR/WEBPAGE.HTM
SET PAGE-NUM = OFF
TABLE FILE GGSTORES
PRINT STORE_NAME ADDRESS1
BY CITY
```
- b. 

```
ON TABLE HOLD FORMAT HTMTABLE AS AREPORT
END
```
- c. 

```
-HTMLFORM DATAOUT SAVE AS WEBPAGE
```

- a. このコマンドは、WEBPAGE.HTM ファイルを割り当て、Web サーバがアクセス可能なディレクトリにそのファイルを保存します。このコマンドは、レポート出力とカスタム HTML ページを統合します。

- b. このコマンドは、レポート出力を HTML フォーマットの仮想ファイルに保存します。このファイルには、AREPORT と名前が付けられます。このファイルはディスクに保存されません。
  - c. このコマンドは、レポート出力を「DATAOUT」という名前のカスタム HTML ページに組み込みます。DATAOUT ファイルは、手順 2 で作成します。また、このコマンドは、「WEBPAGE」という 2 つ目の HTML ファイルを指定し、レポート出力とカスタムページを統合した HTML をこのファイルに保存します。
2. 「DATAOUT」という名前のカスタム HTML ページを作成します。WebFOCUS Reporting Server がこのページの場所を特定できるように、APP PATH または EDAPATH を設定しておく必要があります。

#### HTML ページ - DATAOUT.HTM

```
<HTML>
<BODY BGCOLOR="#CCCCCC">
a. <H2>Store Contacts</H2>
b. <!--WEBFOCUS TABLE AREPORT-->
</BODY>
</HTML>
```

- a. このテキストが、レポートを識別します。
- b. WebFOCUS は HTML コメントを読み込み、そのコメントを AREPORT に格納しているレポート出力で置換します。

**注意：**HTML コメント行は「->」で閉じる必要があります。また、単一の「>」で閉じることでもできますが、その場合はコメント行に他の HTML タグを含めることはできません。

3. プロシジャの CONTACTS を実行して、カスタム HTML ページに静的レポートを作成します。これを実行する方法の 1 つとして、WebFOCUS Client を呼び出す方法があります。

```
<HTML>
<BODY>
Click here
to run CONTACTS and create a static report.
</BODY>
</HTML>
```

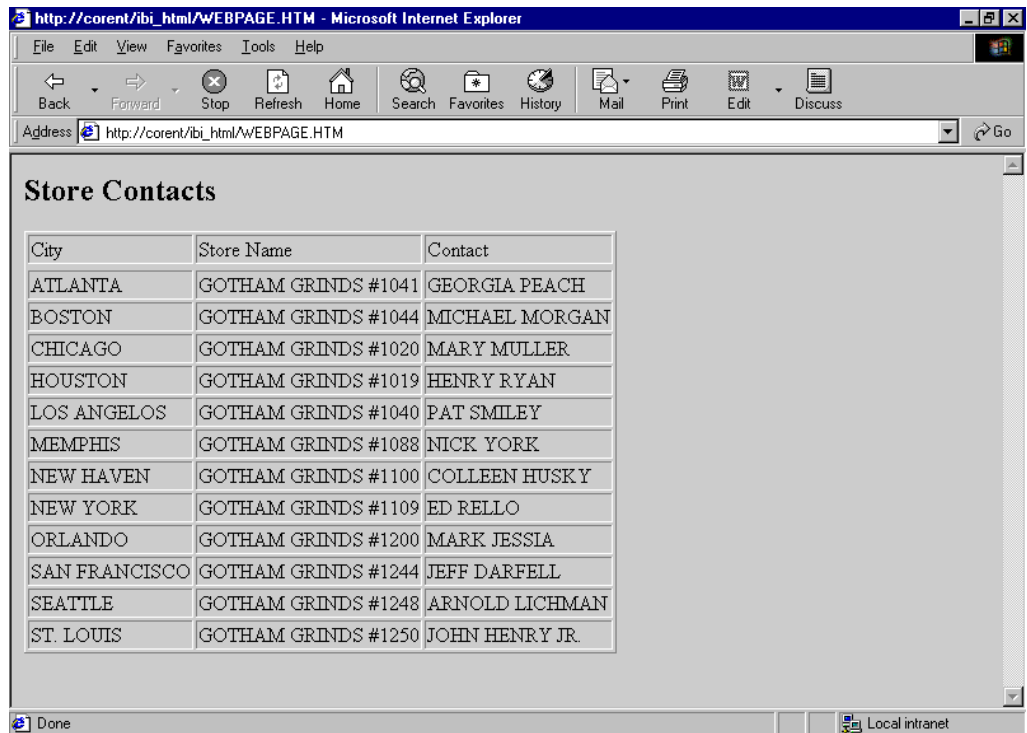
ファイルが作成されたことを示す「HTML FILE SAVED ...」というメッセージがブラウザに表示されます。

4. ユーザがレポートの実行に使用する起動ページを作成します。この例は、「ONESTAT.HTM」という名前の簡単な起動ページです。

#### 起動ページ - ONESTAT.HTM

```
<HTML>
<BODY>
Click here to view store contacts.
</BODY>
</HTML>
```

5. 起動ページを実行し、リンクをクリックします。静的レポートが、カスタム HTML ページに表示されます。



#### 既存の HTML Web ページでの 2 つの静的レポートの表示

1. 「DEMOG」という名前のプロシジャを作成します。左側のアルファベットは、コードのそれぞれの説明に対応しています。

プロシジャ - **DEMOG.FEX**

- ```

a. FILEDEF RPTPAGE DISK APPDIR/RPTPAGE.HTM
   SET PAGE-NUM = OFF
   TABLE FILE GGDEMOG
   SUM HH AS 'Number of,Households' AVGHHSZ98 AS 'Avg.,Size'
   MEDHHI98 AS 'Avg.,Income'
   BY ST
   WHERE ST EQ 'CA'
b. ON TABLE HOLD FORMAT HTMTABLE AS UPPER
   END
   TABLE FILE GGDEMOG
   SUM HH AS 'Number of,Households' AVGHHSZ98 AS 'Avg.,Size'
   MEDHHI98 AS 'Avg.,Income'
   BY ST
   WHERE ST EQ 'NY'
c. ON TABLE HOLD FORMAT HTMTABLE AS LOWER
   END
   -RUN
d. -HTMLFORM RPTOUT SAVE AS RPTPAGE

```

- このコマンドは、RPTPAGE.HTM ファイルを割り当て、Web サーバがアクセス可能なディレクトリにそのファイルを保存します。このコマンドは、レポート出力とカスタム HTML ページを統合します。
 - このコマンドは、最初のリクエストからのレポート出力を「UPPER」という仮想ファイルに保存します。
 - このコマンドは、2 つ目のリクエストからのレポート出力を「LOWER」という名前の仮想ファイルに保存します。
 - このコマンドは、レポート出力を「RPTOUT」という名前のカスタム HTML ページに組み込みます。DATAOUT ファイルは、手順 2 で作成します。また、このコマンドは、「RPTPAGE」という 2 つ目の HTML ファイルを指定し、レポート出力とカスタムページを統合した HTML をこのファイルに保存します。
2. 「RPTOUT.HTM」という名前のカスタム HTML ページを作成します。WebFOCUS Reporting Server がこのページの場所を特定できるように、APP PATH または EDAPATH を設定しておく必要があります。

HTML ページ - RPTOUT.HTM

- ```

<HTML>
<BODY BGCOLOR="#CCCCCC">
a. <H3>Demographic Report for California</H3>
b. <!--WEBFOCUS TABLE UPPER-->
 <HR SIZE=5>
a. <H3>Demographic Report for New York</H3>
b. <!--WEBFOCUS TABLE LOWER-->
 </BODY>
</HTML>

```

- このテキストは、2 つのレポートを識別します。

- b. WebFOCUS はこれらの HTML コメントを読み込み、UPPER および LOWER をレポート出力で置換します。

**注意：**HTML コメント行は「->」で閉じる必要があります。また、単一の「>」で閉じることでもできますが、その場合はコメント行に他の HTML タグを含めることはできません。

3. プロシジャの DEMOG を実行して、カスタム HTML ページに静的レポートを作成します。これを実行する方法の 1 つとして、WebFOCUS Client を呼び出す方法があります。

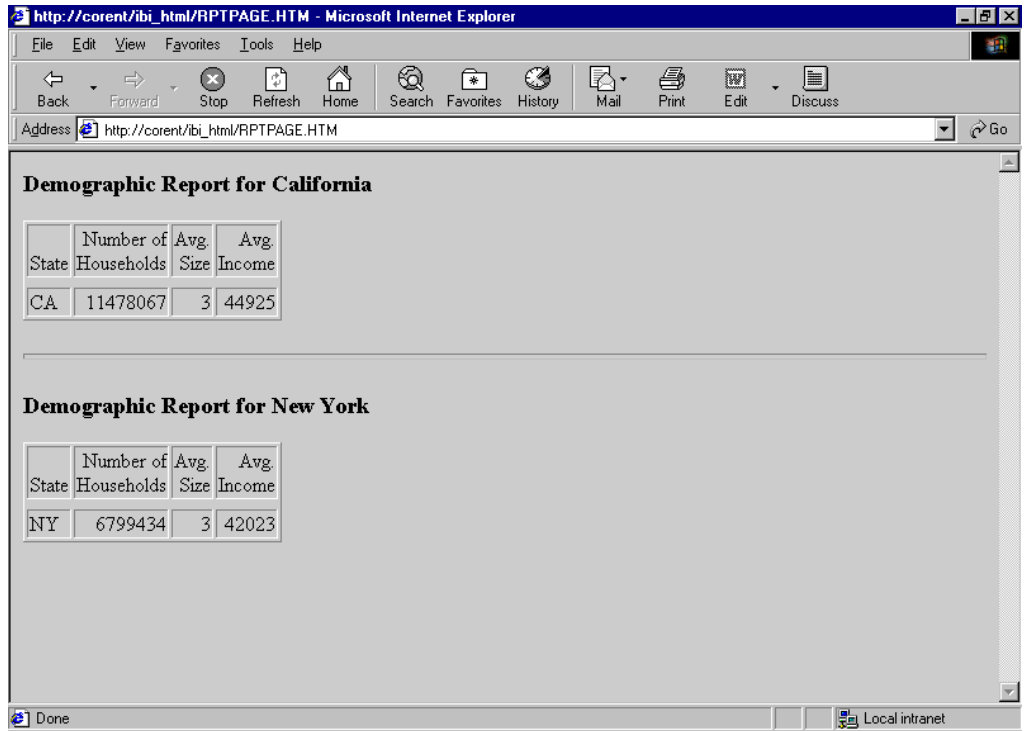
```
<HTML>
<BODY>
Click here
to run DEMOG and create two static reports.
</BODY>
</HTML>
```

4. ユーザがレポートの実行に使用する起動ページを作成します。この例は、「TWOSTATS.HTM」という名前の簡単な起動ページです。

### 起動ページ - TWOSTATS.HTM

```
<HTML>
<BODY>
Click here to view the demographic reports.
</BODY>
</HTML>
```

5. 起動ページを実行し、リンクをクリックします。静的レポートが、カスタム HTML ページに表示されます。



## HOLD FORMAT HTMTABLE および -HTMLFORM による同一列内アコーディオンレポートの表示

-HTMLFORM 機能とともに HOLD FORMAT HTMTABLE コマンドを使用することで、HTML ページに同一列内でのアコーディオンレポート (SET EXPANDBYROW、SET EXPANDBYROWTREE) を表示することができます。-HTMLFORM では、列単位のアコーディオンレポート (SET EXPANDABLE) はサポートされません。

### 構文

## HOLD FORMAT HTMTABLE コマンドおよび -HTMLFORM による同一列内アコーディオンレポートの表示

カスタム HTML ページに 1 つまたは複数の同一列内アコーディオンレポートを表示することができます。これを行うには、SET EXPANDBYROW コマンドを使用してレポートを作成し、HOLD FORMAT HTMTABLE を指定した後、ダイアログマネージャコマンドの -HTMLFORM を使用します。

プロシジャに次のコマンドを追加します。

```
ON TABLE SET {EXPANDBYROW|EXPANDBYROWTREE} ON
ON TABLE HOLD FORMAT HTMLTABLE AS report
```

説明

**report**

レポート出力を格納する仮想ファイルの名前を 1 から 8 バイトで指定します。

このレポートを実行すると、レポートデータのみが含まれた出力ファイルが作成されます。HTML の同一列内アコーディオンレポートとして出力を表示するには、次の構文および配置方法を使用することで、JavaScript コードを HTML に含める必要があります。

**IBI.OBJ.EXPBYROWJS**

HTML アコーディオンレポート機能の JavaScript です。-HTMLFORM 内の <HTML> タグの後、<BODY> タグの前にコーディングする必要があります。

```
<div id="IBI_popupHere"></div>
```

列タイトルのポップアップ説明を表示するために使用します。最初のアコーディオンレポートの表示 (!IBI.FIL.filename の使用) の前に含める必要があります。

**!IBI.OBJ.IBIGBLONLOAD**

WebFOCUS HTML レポートのグローバル環境設定のロードを要求する JavaScript です。最初の HTML レポートの表示位置を示すコメント行 (!IBI.FIL.report) の前の、<BODY> タグ内にコーディングする必要があります。

```
<BODY ONLOAD='!IBI.OBJ.IBIGBLONLOAD';>
```

**!IBI.OBJ.IBIGBLJS**

WebFOCUS HTML レポートのグローバル環境設定の JavaScript です。最後の HTML レポートの表示位置を示すコメント行 (!IBI.FIL.report) の後の、</HTML> 終了タグの前にコーディングする必要があります。

## 例 HTML Web ページでの 2 つの HTML アコーディオンレポートの表示

次のリクエストには、HTMTABLE フォーマットで保存された RPT1 および RPT2 という 2 つの同一列内アコーディオンレポートが使用されています。-HTMLFORM ブロックには、これらのレポートを HTML ページに表示するための JavaScript コードのすべてが含まれています。

```
TABLE FILE GGSales
HEADING
"Regional Budget and Sales Report"
" "
SUM BUDUNITS UNITS BUDDOLLARS DOLLARS
BY REGION
BY ST
BY CATEGORY
BY PRODUCT
ON TABLE HOLD AS RPT1 FORMAT HTMTABLE
ON TABLE SET EXPANDBYROW ON
ON TABLE SET HTMLCSS ON
ON TABLE SET DROPBLNKLINE ALL
ON TABLE SET STYLE *
TYPE=REPORT, COLOR=NAVY, FONT='ARIAL', SIZE=9, GRID=OFF, $
TYPE=HEADING, LINE=1, STYLE=BOLD, SIZE=12, JUSTIFY=CENTER, $
TYPE=TITLE, BACKCOLOR=RGB(45 111 205), COLOR=WHITE, STYLE=-UNDERLINE+BOLD, $
TYPE=DATA, BACKCOLOR=(WHITE RGB(235 235 255)), $
TYPE=SUBTOTAL, BACKCOLOR=RGB(163 200 236), STYLE=BOLD, $
END

TABLE FILE GGSales
HEADING
"Product Category Sales Report"
" "
SUM GGSales.SALES01.DOLLARS
GGSales.SALES01.BUDDOLLARS
BY GGSales.SALES01.CATEGORY
BY GGSales.SALES01.REGION
BY GGSales.SALES01.ST
ON TABLE HOLD AS RPT2 FORMAT HTMTABLE
ON TABLE SET EXPANDBYROW ON
ON TABLE SET HTMLCSS ON
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE SET SQUEEZE ON
ON TABLE SET EMPTYREPORT ON
ON TABLE SET DROPBLNKLINE ALL
ON TABLE SET STYLE *
TYPE=REPORT, COLOR=NAVY, SQUEEZE=ON, FONT='ARIAL', SIZE=9, GRID=OFF, $
TYPE=HEADING, LINE=1, STYLE=BOLD, SIZE=12, JUSTIFY=CENTER, $
TYPE=TITLE, BACKCOLOR=RGB(45 111 205), COLOR=WHITE, STYLE=-UNDERLINE+BOLD, $
TYPE=DATA, BACKCOLOR=(WHITE RGB(235 235 255)), $
TYPE=SUBTOTAL, BACKCOLOR=RGB(163 200 236), STYLE=BOLD, $
END
```

```
-HTMLFORM BEGIN
<HTML>|IBI.OBJ.EXPBYROWJS;
<BODY ONLOAD='|IBI.OBJ.IBIGBLONLOAD;'>
This HTML page is created with -HTMLFORM
and displays
 two Accordion By Row (SET EXPANDBYROW) Reports

</br>

<div id="IBI_popupHere"></div>
|IBI.FIL.RPT1;

</br>
|IBI.FIL.RPT2;
|IBI.OBJ.IBIGBLJS;
</BODY>
</HTML>
-HTMLFORM END
```

出力結果は次のとおりです。

This HTML page is created with -HTMLFORM displays  
two Accordion (SET EXPANDBYROW) Reports

HTMLTABLE data from RPT1 HOLD file

### Regional Budget and Sales Report

	Budget Units	Unit Sales	Budget Dollars	Dollar Sales
<input type="checkbox"/> Midwest	907107	905045	11194373	11400665
<input type="checkbox"/> Northeast	914359	916675	11576921	11392300
<input type="checkbox"/> CT	301367	302440	3832202	3782049
<input type="checkbox"/> MA	301605	301909	3818397	3707986
<input type="checkbox"/> NY	311387	312326	3926322	3902265
<input type="checkbox"/> Coffee	117285	116659	1468684	1459160
<input type="checkbox"/> Food	124283	125473	1568925	1555165
Biscotti	50502	51984	644415	642259
Croissant	50178	50518	640032	622095
Soone	23603	22991	284478	290811
<input type="checkbox"/> Gifts	69819	70194	888713	887940
<input type="checkbox"/> Southeast	942247	935232	11807971	11710379
<input type="checkbox"/> West	930781	932039	11641513	11652946
TOTAL	3694494	3688991	46220778	46156290

HTMLTABLE data from RPT2 HOLD file

### Product Category Sales Report

	Dollar Sales	Budget Dollars
<input type="checkbox"/> Coffee	17231455	17293886
<input type="checkbox"/> Food	17229333	17267160
<input type="checkbox"/> Midwest	4338271	4220721
<input type="checkbox"/> Northeast	4379994	4453907
CT	1424718	1436890
MA	1400111	1448092
NY	1555165	1568925
<input type="checkbox"/> Southeast	4308731	4409288
<input type="checkbox"/> West	4202337	4183244
<input type="checkbox"/> Gifts	11695502	11659732
TOTAL	46156290	46220778

## HOLD FORMAT HTMTABLE および -HTMLFORM による HTML HFREEZE レポートの表示

カスタム HTML ページに 1 つまたは複数の HTML HFREEZE レポートを表示するには、HFREEZE=ON スタイル属性を使用してレポートを作成し、HOLD FORMAT HTMTABLE コマンドを追加した上で、ダイアログマネージャコマンドの -HTMLFORM を使用します。

### 構文 **HOLD FORMAT HTMTABLE および -HTMLFORM による HTML HFREEZE レポートの表示**

レポートリクエストに次のコマンドを追加します。

```
ON TABLE HOLD FORMAT HTMTABLE AS report
```

また、リクエストのスタイルシートセクションに次のコードを追加します。

```
ON TABLE SET STYLE *
TYPE=REPORT, HFREEZE=ON, $
```

説明

**report**

レポート出力を格納する仮想ファイルの名前を 1 から 8 バイトで指定します。

このレポートを実行すると、レポートデータのみが含まれた出力ファイルが作成されます。出力を HTML HFREEZE レポートとして表示するには、次の構文および配置方法を使用することで、JavaScript コードを HTML に追加する必要があります。

```
!IBI.FIL.report
```

HTML ページでのレポートの配置を示すプレースホルダとして必要です。

```
!IBI.OBJ.IBIHEADJS
```

HTML ページの <HEAD> タグに挿入します。

## 例 HOLD FORMAT HTMTABLE および -HTMLFORM による 2 つの HTML HFREEZE レポートの表示

次のプロシジャは、-HTMLFORM を使用して作成された HTML ページに、2 つの HFREEZE レポートを作成して表示します。最初の 2 つの TABLE リクエストで HFREEZE レポートが作成され (スタイルコードで HFREEZE=ON を指定)、これらのレポートが ON TABLE HOLD FORMAT HTMTABLE により WebFOCUS Reporting Server に保存されます。

-\*Example: Displaying Two HTML HFREEZE Reports on an HTML Web Page  
-\*

```
TABLE FILE GGSales
HEADING CENTER
"Regional Sales Report"
SUM DOLLARS BUDDOLLARS
BY CATEGORY
BY PRODUCT
BY REGION
BY ST
BY CITY
ON REGION SUBTOTAL
FOOTING CENTER
"*** Scroll To See All Data ***"
ON TABLE HOLD AS TOPRPT FORMAT HTMTABLE
ON TABLE SET HTMLCSS ON
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE SET SQUEEZE ON
ON TABLE SET EMPTYREPORT ON
ON TABLE SET BYDISPLAY ON
ON TABLE SET STYLE *
TYPE=REPORT,COLOR=NAVY,SQUEEZE=ON,Font='ARIAL',SIZE=9,GRID=OFF,$
TYPE=REPORT,HFREEZE=ON,SCROLLHEIGHT=2.25,$
TYPE=HEADING,STYLE=BOLD,SIZE=12,JUSTIFY=CENTER,$
TYPE=FOOTING,STYLE=BOLD,SIZE=10,JUSTIFY=CENTER,$
TYPE=TITLE,BACKCOLOR=RGB(45 111 205),COLOR=WHITE,STYLE=-UNDERLINE+BOLD,$
TYPE=DATA,BACKCOLOR=(WHITE RGB(235 235 255)),$
TYPE=SUBTOTAL,BACKCOLOR=RGB(163 200 236),STYLE=BOLD,$
TYPE=SUBTOTAL,BORDER-TOP=LIGHT,BORDER-TOP-COLOR=RGB(111 155 246),$
END
```

```

TABLE FILE GGSales
HEADING CENTER
"Regional Sales Report"
SUM GGSales.SALES01.DOLLARS
GGSales.SALES01.BUDDOLLARS
BY GGSales.SALES01.CATEGORY
BY GGSales.SALES01.ST
BY GGSales.SALES01.REGION
ON REGION SUBTOTAL
FOOTING CENTER
"*** Scroll To See All Data ***"
ON TABLE HOLD AS BOTRPT FORMAT HTMTABLE
ON TABLE SET HTMLCSS ON
ON TABLE SET PAGE-NUM NOLEAD
ON TABLE SET SQUEEZE ON
ON TABLE SET EMPTYREPORT ON
ON TABLE SET BYDISPLAY ON
ON TABLE SET STYLE *
TYPE=REPORT,COLOR=NAVY,SQUEEZE=ON,Font='ARIAL',SIZE=9,GRID=OFF,$
TYPE=REPORT,HFREEZE=ON,SCROLLHEIGHT=2.25,$
TYPE=HEADING,STYLE=BOLD,SIZE=12,JUSTIFY=CENTER,$
TYPE=FOOTING,STYLE=BOLD,SIZE=10,JUSTIFY=CENTER,$
TYPE=TITLE,BACKCOLOR=RGB(45 111 205),COLOR=WHITE,STYLE=-UNDERLINE+BOLD,$
TYPE=DATA,BACKCOLOR=(WHITE RGB(235 235 255)),$
TYPE=SUBTOTAL,BACKCOLOR=RGB(163 200 236),STYLE=BOLD,$
TYPE=SUBTOTAL,BORDER-TOP=LIGHT,BORDER-TOP-COLOR=RGB(111 155 246),$
END

-HTMLFORM BEGIN
<HTML>
<HEAD>|IBI.OBJ.IBIHEADJS;
</HEAD>
<BODY>
This HTML page is created with -
HTMLFORM displaying two HFREEZE reports

</br>
HTMLTABLE data from TOPRPT HOLD file
!IBI.FIL.TOPRPT;
HTMLTABLE data from BOTRPT HOLD file
!IBI.FIL.BOTRPT;
</BODY>
</HTML>
-HTMLFORM END

```

下図のように、生成された HTML ページに 2 つのレポートが表示されます。

**This HTML page is created with -HTMLFORM displaying two HFREEZE reports**

HTMLTABLE data from TOPRPT HOLD file

#### Regional Sales Report

Category	Product	Region	State	City	Dollar Sales	Budget Dollars
Coffee	Capuccino	Northeast	CT	New Haven	158995	141574
Coffee	Capuccino	Northeast	MA	Boston	174344	192747
Coffee	Capuccino	Northeast	NY	New York	208756	227170
*TOTAL REGION Northeast					542095	561491
Coffee	Capuccino	Southeast	FL	Orlando	317027	285194
Coffee	Capuccino	Southeast	GA	Atlanta	352161	384281
Coffee	Capuccino	Southeast	TN	Memphis	274812	287186
*TOTAL REGION Southeast					944000	956661
Coffee	Capuccino	West	CA	Los Angeles	306468	281701
Coffee	Capuccino	West	CA	San Francisco	279830	294884
Coffee	Capuccino	West	WA	Seattle	309197	300719
*TOTAL REGION West					895495	877304
TOTAL					46156290	46220778

\*\*\* Scroll To See All Data \*\*\*

HTMLTABLE data from BOTRPT HOLD file

#### Regional Sales Report

Category	State	Region	Dollar Sales	Budget Dollars
Coffee	CA	West	2937886	2960640
*TOTAL REGION West			2937886	2960640
Coffee	CT	Northeast	1364420	1395283
*TOTAL REGION Northeast			1364420	1395283
Coffee	FL	Southeast	1463453	1408190
*TOTAL REGION Southeast			1463453	1408190
Coffee	GA	Southeast	1576915	1656902
*TOTAL REGION Southeast			1576915	1656902
Coffee	IL	Midwest	1398779	1366264
*TOTAL REGION Midwest			1398779	1366264
Coffee	MA	Northeast	1340437	1388495
*TOTAL REGION Northeast			1340437	1388495
TOTAL			46156290	46220778

\*\*\* Scroll To See All Data \*\*\*

## HOLD FORMAT AHTMLTAB および -HTMLFORM による Active テクノジレポートの表示

HOLD FORMAT AHTMLTAB およびダイアログマネージャコマンドの -HTMLFORM を使用して、カスタム HTML ページに 1 つまたは複数の HTML Active テクノジレポートを表示することができます。

プロシジャに次のコマンドを追加します。

```
ON TABLE HOLD FORMAT AHTMLTAB AS report
```

説明

**report**

レポート出力を格納する仮想ファイルの名前です。ファイル名として許容される長さは、1 から 8 バイトです。

このレポートを実行すると、Analytic Document で使用されるデータとパラメータのみが格納された出力ファイルが作成されます。完全な Analytic Document として出力を表示するには、次の構文を使用して、Analytic Document の JavaScript コードを HTML BODY に含める必要があります。

```
<BODY>
!IBI.OBJ.ACTIVEREPORTJS;
```

### 例

#### HTML Web ページでの Analytic Document レポートの表示

次の例では、Active テクノジレポートの JavaScript コードに含める HTML コードは太字で示されています。

```
TABLE FILE GGSales
SUM
 DOLLARS
 UNITS
BY REGION
BY ST
BY CITY
HEADING
"Regional Sales Summary"
ON TABLE HOLD AS REPORT1 FORMAT AHTMLTAB
END
```

```

- *
TABLE FILE GGSALES
SUM
 DOLLARS
 UNITS
BY CATEGORY
BY PRODUCT
HEADING
"Production Order Summary"
ON TABLE HOLD AS REPORT2 FORMAT AHTMLTAB
END
- *
-HTMLFORM BEGIN
<!DOCTYPE html>
<HTML>
<HEAD>
<TITLE>Displaying HTML Active Technologies Reports on an HTML Web Page</
TITLE>
</HEAD>

<BODY>
!IBI.OBJ.ACTIVEREPORTJS;

<TABLE BORDER='1'>
<TR>
 <TD valign=top>
!IBI.FIL.REPORT1;
 </TD>
 <TD valign=top>
!IBI.FIL.REPORT2;
 </TD>
</TR>
</TABLE>
</BODY>
</HTML>
-HTMLFORM END

```

## -HTMLFORM を使用したマルチドリルダウンメニュー項目の表示

DRILLMENUITEM スタイル属性を使用してレポートを作成することで、カスタム HTML ページに複数のメニュー項目を表示することができます。

レポートノードまたはデータ要素に付属する各メニューおよびメニュー項目を定義するには、プロシジャに次の構文を追加します。

```

TYPE=type [,subtype] [,DRILLMENUITEM='description', action|'keyword']
[,NAME=name] [,PARENT=parentname],

```

### 説明

#### type

リンクを実行するために、ユーザが Web ブラウザで選択するレポートコンポーネントを特定します。TYPE 属性とその値は、構文の先頭部分に配置する必要があります。

#### subtype

追加の属性です。COLUMN、LINE、ITEM などがあり、フォーマット設定対象のレポートコンポーネントを特定するために必要です。

DRILLMENUITEM の各項目には、description または keyword の組み合わせが必要です。action を伴わない description は、デフォルト設定で自動的に無効になります。

ただし、NAME/PARENT の組み合わせでリンクされた子エントリを含む親項目については例外です。この場合、action でカスケードメニューに子が含まれます。

#### description

レポート出力で、ドリルダウンオプションのポップアップメニューに表示されるテキストです。デフォルト値は ドリルダウン  $n$  で、 $n$  はドリルダウン 1、ドリルダウン 2 などの連続する整数です。

#### 注意

- ☐ DRILLMENUITEM が、特別な値である 'SEPARATOR' に設定されている場合は、次のようになります。
- ☐ 横方向の分割線が、メニュー内の指定した位置に、メニュー境界線で定義されたスタイル属性と色属性を使用して描画されます。
- ☐ 分割線は、action と関連付けることはできません。
- ☐ DRILLMENUITEM の値はブランクにすることはできません。

#### action

リンクのタイプです。たとえば、詳細レポートや URL へのリンクです。

次の属性はオプションです。これらは、階層の定義が必要なカスケードメニューでのみ必要です。

#### name

親子項目間のリンクとして使用する現在の項目の一意の識別子です (オプション)。このノードが、リンクを特定する必要のある子メニュー項目の親として機能する場合のみ必要です。

**parentname**

現在の子項目の親メニュー項目の一意の識別子または名前です (オプション)。このノードが、階層内の別の項目の親として機能する場合のみ必要です。

**例****-HTMLFORM を使用したマルチドリルダウンメニュー項目の表示**

次の例は、DRILLMENUITEM スタイルシート属性を使用して 2 つの URL リンクにドリルダウンする方法を示しています。

```
SET POPUPDESC=ON
TABLE FILE GGSales
SUM DOLLARS UNITS
BY REGION BY ST BY CITY
HEADING
"Regional Sales Summary"
ON TABLE SET STYLE *
TYPE=DATA, COLUMN=N1,
DRILLMENUITEM='DrillDown 1', URL=http://www.ibi.com,
DRILLMENUITEM='DrillDown 2', URL=http://support.ibi.com,$
ENDSTYLE
ON TABLE HOLD FORMAT HTMLTABLE
END
-HTMLFORM BEGIN
<html>
<BODY>
<head> <title>Displaying Drilldown Menu Items Using -HTMLFORM</title>
!IBI.OBJ.IBIHEADJS;</head>
!IBI.FIL.HOLD;
</BODY>
</html>
-HTMLFORM END
```

出力結果は次のとおりです。

PAGE 1				
Regional Sales Summary				
Region	State	City	Dollar Sales	Unit Sales
<a href="#">Midwest</a>	IL	Chicago	3924401	307581
	MO	St. Louis	3761286	297727
	TX	Houston	3714978	299737
<a href="#">Northeast</a>	CT	New Haven	3782049	302440
	MA	Boston	3707986	301909
	NY	New York	3902265	312326
<a href="#">Southeast</a>	FL	Orlando	3923215	310302
	GA	Atlanta	4100107	330283
	TN	Memphis	3687057	294647
<a href="#">West</a>	CA	Los Angeles	3772003	298070
		San Francisco	3870258	312500
	WA	Seattle	4010685	321469

## HOLD FORMAT XML および -HTMLFORM の使用

- HTMLFORM で呼び出す HOLD FORMAT XML を作成する場合は、HOLD FORMAT XML にコードを追加して、ブランクの HOLD ファイルが作成されてエラーが発生することを防止する必要があります。次の例に太字で示すように、FILEDEF コードを 2 行追加します。

```
TABLE FILE GGORDER
SUM QUANTITY
ON TABLE HOLD FORMAT XML
END
-RUN
FILEDEF HOLD CLEAR
FILEDEF HOLD DISK HOLD.XML
-HTMLFORM BEGIN
!IBI.FILE.HOLD;
-HTMLFORM END
```

# 4

## ユーザインターフェースの機能強化

---

WebFOCUS スタイルシートを使用して、JavaScript 関数へのハイパーリンクを指定することができます。この機能は、ドリルダウンレポートでデータ計算を実行する際に使用します。

どのような操作でレポートを実行するかは、ユーザインターフェースの重要な部分です。ここでは、アプリケーションで WebFOCUS ビューアを使用して、長いレポートを表示する方法について説明します。また、外部カスケードスタイルシート (CSS) を使用してレポート表示にスクロールバーを追加する方法についても説明します。

起動ページに高度な機能を実装することができます。たとえば、メニューに表示するオプションをユーザごとにカスタマイズします。

この章では、インターフェースの機能性および操作性を強化するコーディング機能について説明します。

レポートは、HTML フォーマット以外でも表示することができます。次の表示方法があります。

- ❑ PDF (Portable Document Format) などのヘルパーアプリケーションを使用した HTML 以外のフォーマットで表示するレポート。
- ❑ 1 つの起動ページから種類の異なるフォーマットで表示する複数のレポート。

### トピックス

- ❑ [ヘルパーアプリケーションによるレポートの表示](#)
  - ❑ [複数レポートの制御](#)
  - ❑ [HTML Web ページへの CSRF トークンの追加](#)
  - ❑ [JavaScript によるドリルダウンレポートの作成](#)
  - ❑ [Web ビューア機能](#)
  - ❑ [カスケードスタイルシート \(CSS\) による表示の標準化](#)
  - ❑ [実行済みレポートの表示](#)
-

ヘルパーアプリケーションによるレポートの表示

ヘルパーアプリケーションまたはプラグインは、標準の HTML フォーマット以外のフォーマットでファイルをブラウザに表示する際に使用するデスクトッププログラムです。ヘルパーアプリケーションの一例として、PDF ファイルを開くための Adobe Reader があります。

Web サーバは、特殊なタイプのファイルをブラウザに送信する際に、ファイルのフォーマットを表す「MIME (Multipurpose Internet Mail Extensions)」という情報を同時に送信します。WebFOCUS は、MIME タイプに基づいて、実行するヘルパーアプリケーションの種類を特定します。そのため、Windows のエクスプローラを使用して、ファイルのコンテンツ (MIME) タイプをヘルパーアプリケーションに関連付ける必要があります。

手順 Windows のエクスプローラで MIME タイプを指定するには

- 1. Windows のエクスプローラを開きます。
- 2. [ツール] メニューから [フォルダオプション] を選択します。
- 3. [ファイルの種類] タブを選択します。
- 4. [登録されているファイルの種類] リストボックスからファイルの種類を選択し、[新規] ボタンをクリックします。

[新しい拡張子の作成] ダイアログボックスが表示されます。

- 5. 次の情報を入力します。
  - ❑ [ファイルの拡張子] テキストボックスに、ファイルタイプの拡張子を入力します。
  - ❑ [関連付けられているファイルの種類] ドロップダウンリストから、そのファイルタイプと拡張子用の MIME タイプを選択します。
- 6. [OK] をクリックし、変更を保存します。

参照 サポートされる MIME タイプ

WebFOCUS は、次の MIME タイプをサポートします。

MIME タイプ	アプリケーション	ファイル拡張子
application/x-dif	Microsoft Excel または他の表計算形式	.DIF
application/x-doc	Microsoft Word	.doc

MIME タイプ	アプリケーション	ファイル拡張子
application/x-xls application/vnd.ms-excl	Microsoft Excel	.xls、e97、.wk1、.xht
application/pdf	Adobe Reader	.pdf
text/plain	Microsoft の「メモ帳」または他のテキストエディタ	.wp または .hts
text/html	ネイティブのブラウザフォーマット	.html
text/htm	ネイティブのブラウザフォーマット	.htm
image/gif	ネイティブのブラウザフォーマット	.gif
image/jpeg	ネイティブのブラウザフォーマット	.jpg、.jpeg
XML	ネイティブのブラウザフォーマット	.xml

## 手順

### ヘルパーアプリケーションでレポートを表示するには

1. 次のコマンドを含むプロシジャを作成します。

```
ON TABLE PCHOLD FORMAT fmt
```

説明

`fmt`

データのファイルフォーマットです。

**警告：**ヘルパーアプリケーションでブラウザに表示する有効なフォーマットは、サポートされている MIME タイプでなければなりません。

2. ユーザがレポートの実行に使用する起動ページを作成します。
3. App Studio を使用してプロシジャおよび起動ページを公開します。起動ページを実行すると、WebFOCUS は指定されたフォーマットでデータをブラウザに返します。ブラウザは、該当するヘルパーアプリケーションを使用してこのデータを表示します。

## 例

### レポートの Excel 表示と表示フォーマットの選択オプション

次の例では、レポートを Excel ワークシートに表示し、表示フォーマットの選択オプションを提供します。

**Excel 2000 形式でのレポートの表示**

次の手順は、レポートを Excel 2000 形式のワークシートに表示する例を示しています。

1. 「HELPER」というプロシジャを作成して、製品の注文に関するレポートを生成します。

```
TABLE FILE GGORDER
SUM QUANTITY BY PCD
WHERE PCD IS 'B141' OR 'B142' OR 'F101' OR 'F102'
ON TABLE PCHOLD FORMAT EXL2K
END
```

2. ユーザがレポートの実行に使用する起動ページを作成します。
3. App Studio を使用してプロシジャおよび起動ページを公開します。
4. 起動ページを実行して、リンクをクリックします。WebFOCUS は、データを EXL2K フォーマットでブラウザに返します。ブラウザがヘルパーアプリケーションの Microsoft Excel 2000 を呼び出してレポートを表示します。

	A	B	C	D	E	F	G	H	I	J
1	Product Code	Ordered Units								
2	B141	100427								
3	B142	285689								
4	F101	108359								
5	F102	136045								
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										

**表示フォーマット選択オプションの提供**

次の例は、表示フォーマットの選択オプションを提供する方法を示しています。

1. 「QASTATUS」というプロシジャを作成し、特定の工場 (PLANT) の品質保証レポートを生成し、選択したフォーマット (FMT) で表示します。

```
TABLE FILE CENTQA
ON TABLE SET PAGE-NUM OFF
SUM CNT.PROBNUM AS 'Total Number,of Problems'
SUM CNT.PROBNUM AS 'Problem by,Product' BY PLANT NOPRINT BY PRODNAME
WHERE PLANT EQ '&PLANT'
HEADING CENTER
"QA Report for &PLANT"
ON TABLE PCHOLD FORMAT &FMT
END
```

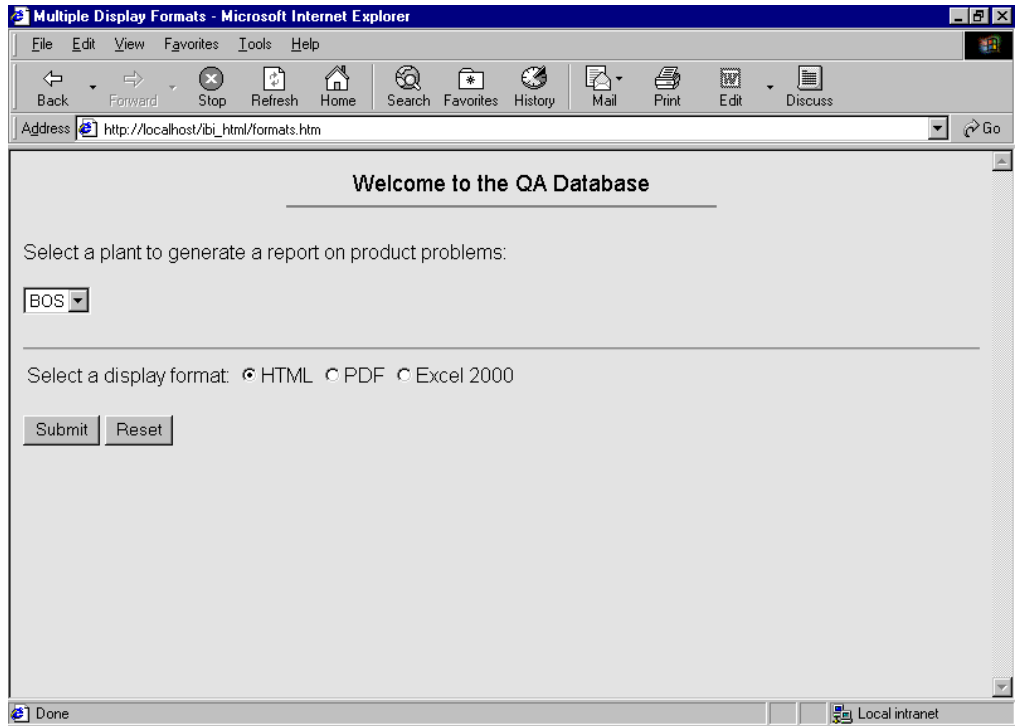
2. ユーザがレポートの実行に使用する起動ページを作成します。次のサンプル起動ページには、「FORMATS」という名前が付けられています。左側に表示されたアルファベット文字は、そのコードの意味について説明する注記に対応しています。

```
<HTML>
<HEAD>
<TITLE>Multiple Display Formats</TITLE>
</HEAD>
<BODY BGCOLOR="#E3E3E3">

<CENTER>
Welcome to the QA Database
<HR WIDTH="45%" NOSHADE>
</CENTER>
<FORM ACTION="/ibi_apps/WFServlet" METHOD="get">
<INPUT NAME="IBIF_ex" VALUE="qastatus" TYPE="hidden">
<P>
Select a plant to generate a report on product problems:
</P>
<SELECT NAME="PLANT">
<OPTION>BOS
<OPTION>DAL
<OPTION>LA
<OPTION>ORL
<OPTION>SEA
<OPTION>STL
</SELECT>
<P>
<HR>
a. <TABLE CELLSPACING="2">
<TR>
<TD CLASS="LABEL">Select a display format:</TD>
<TD><INPUT TYPE="RADIO" NAME="FMT" VALUE="HTML" CHECKED>HTML</TD>
<TD><INPUT TYPE="RADIO" NAME="FMT" VALUE="PDF">PDF</TD>
<TD><INPUT TYPE="RADIO" NAME="FMT" VALUE="EXL2K">Excel 2000</TD>
</TR>
</TABLE>
<P>
<INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="Submit">
<INPUT TYPE="RESET" NAME="RESET" VALUE="Reset">
</FORM>
</BODY>
</HTML>
```

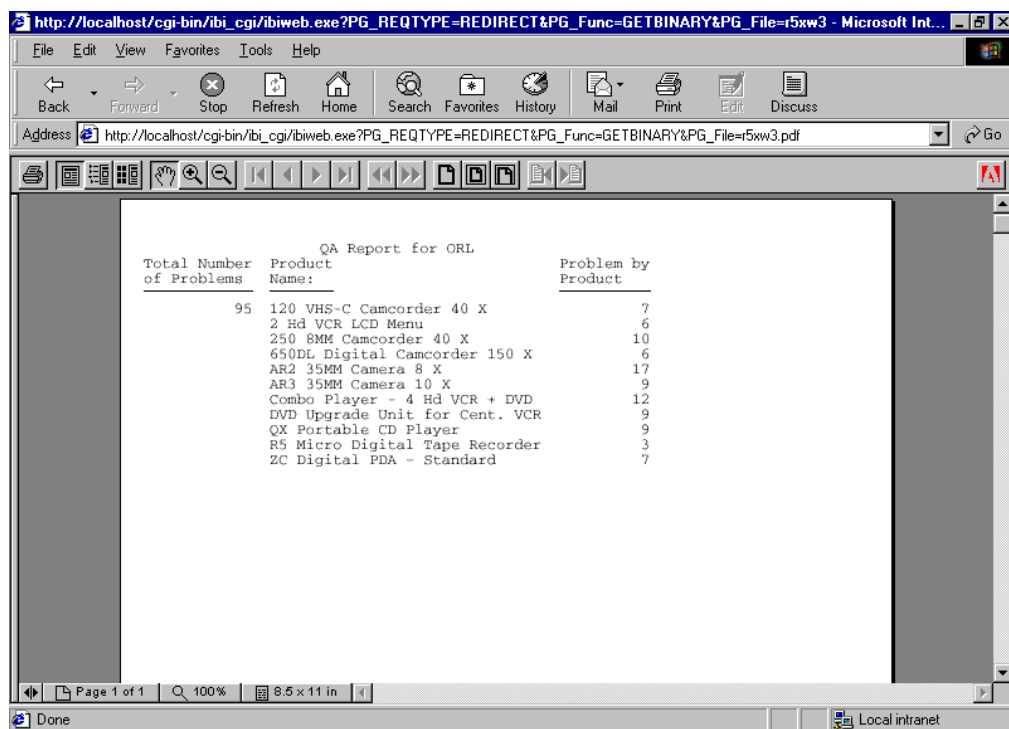
- a. テーブル内のラジオボタンは、ユーザにフォーマット (FMT) の入力并要求し、ここで入力された値がプロシジャに渡されます。
3. App Studio を使用してプロシジャおよび起動ページを公開します。

4. 起動ページを実行します。



5. ドロップダウンリストから [ORL] を選択します。フォーマットには [PDF] を選択します。  
[Submit] をクリックします。

WebFOCUS は、PDF フォーマットでブラウザにデータを返します。ブラウザがヘルパーアプリケーションの Adobe Reader を呼び出してレポートを表示します。



## 複数レポートの制御

WebFOCUS Client 変数を使用すると、HTML の FRAME 構文をコーディングせずに、複数のフレームで構成された HTML ページに複数のレポートを配置する際の表示方法を制御することができます。また、複数のレポートを目次 (TOC) にリンクさせて、各レポートを呼び出すこともできます。たとえば、この変数を使用して、1つの起動ページに HTML と PDF などの種類の異なるレポートフォーマットを組み合わせる表示します。

### 注意

- ☐ WebFOCUS Client は複数レポートの表示を制御するため、目次 (TOC) 機能は ReportCaster 配信ではサポートされていません。

次のことが可能です。

- ☐ ハイパーリンクから WebFOCUS Client を呼び出した際に変数を渡す。

- ❑ ダイアログマネージャの -TYPE コマンドを使用してプロシジャに変数を設定する。この方法を使用する場合は、作成したアプリケーションを WebFOCUS でテストします。  
WebFOCUS のテキストエディタまたは App Studio 環境でプロシジャを作成および公開後、プロシジャを実行します。

ここで説明する変数は、複数のレポートを生成するプロシジャで使用します。-RUN、-INCLUDE、EX を使用して複数のレポートリクエストをコーディングする方法についての詳細は、307 ページの「[アプリケーションの制御フロー管理](#)」を参照してください。

## 参照

### 複数レポートのフォーマット

複数のレポートでは、次の 2 つのフォーマットがサポートされます。

- ❑ 1 つのフレームセットに表示する複数のレポート。
- ❑ プロシジャのすべてのレポートを左側のメニューにリスト表示し、最初のレポートを右側に表示する目次レポート。ユーザがメニューからレポートの 1 つをクリックすると、そのレポートが実行され、デフォルト設定に基づいてウィンドウの右側に表示されます。

## 参照

### WebFOCUS Client 変数

複数のレポートを制御する変数には次のものがあります。

#### IBIWF\_mreports

複数のレポートオプションを制御して、ブラウザに目次を作成するか、フレームセットを作成して各フレームにそれぞれ 1 つのレポートを表示します。

構文は次のとおりです。

```
IBIWF_mreports = {OFF|INDEX|FRAME}
IBIWF_mrcolumns = {1|n}
IBIWF_mrrows = nIBIWF_mprefix = {Report|text}
IBIWF_morder = {FORWARD|REVERSE}
IBIWF_mframeName = {MREPORT|text}
IBIWF_index = {ON|OFF}
```

説明

#### OFF

複数のレポートオプションを無効にします。デフォルト値は OFF です。

#### INDEX

プロシジャ内のすべてのレポートをリスト表示する目次を作成します。デフォルトの連続番号は、1 から n までです。ここで、1 は最初に生成されるレポート、n は最後に生成されるレポートの番号を表します。IBIWF\_mprefix とともに使用します。

### FRAME

デフォルトのフレームセットを作成します。フレームセット内のレポート数は、`IBIWF_mrcolumns` 変数で指定します。

### **IBIWF\_mrcolumns**

`IBIWF_mreports` を `FRAME` に設定した場合に、ページの左右に並べて表示するレポートの個数です。この変数を設定しない場合、レポートは上下に並べて表示されます。

構文は次のとおりです。

```
IBIWF_mrcolumns = {1|n}
```

説明

`n`

レポートの個数です。デフォルト値は 1 です。最大値は 9 です。

### **IBIWF\_mrrows**

`IBIWF_mreports` を `FRAME` に設定した場合に、上下に並べて表示するレポートの個数です。

構文は次のとおりです。

```
IBIWF_mrrows = n
```

説明

`n`

上下に並べて表示するレポートの個数です。

**IBIWF\_mprefix**

レポートを識別するために目次の連続番号の前に付ける説明テキストです。

WebFOCUS は、IBIWF\_mreports のインデックス値で設定する番号を 1 から n の順で追加します。ここで、1 は最初に生成されるレポート、n は最後に生成されるレポートに付けられる番号を表します。

IBIWF\_mreports を FRAME に設定した場合は、この変数を使用することはできません。

構文は次のとおりです。

```
IBIWF_mprefix = {Report|text}
```

説明

[text](#)

最大で 50 バイトの文字列です。この最大長には、WebFOCUS が追加する番号は含まれません。デフォルト値は Report です。

**IBIWF\_morder**

複数のレポートをブラウザに表示する順序です。IBIWF\_mreports を FRAME に設定した場合にのみ適用されます。IBIWF\_mreports を INDEX に設定した場合は無視されます。

構文は次のとおりです。

```
IBIWF_morder = {FORWARD|REVERSE}
```

説明

[FORWARD](#)

複数のレポートをコーディングして実行した順序で表示します。これがデフォルト値です。

[REVERSE](#)

複数のレポートをコーディングして実行した順序の逆の順序で表示します。この設定は、最後のレポートが集計レポートで、Web ページの先頭に集計レポートを表示したい場合に役立ちます。

**IBIWF\_mframename**

IBIWF\_mreports を FRAME に設定した場合のフレームの名前です。この変数をコーディングしない場合は、WebFOCUS が内部的に MREPORT1 から MREPORTn までのフレーム名を付けるため、他の HTML コードと競合する可能性があります。

構文は次のとおりです。

```
IBIWF_mframename = {MREPORT|text}
```

説明

text

最大で 20 バイトの文字列です。

### IBIWF\_index

IBIWF\_mreports を INDEX に設定した場合に、目次のレポート名の末尾に付ける連続番号 (1、2、....n) の有無を制御します。

構文は次のとおりです。

```
IBIWF_index = {ON|OFF}
```

説明

ON

1 から n までの連続番号を追加します。ここで、1 は最初に生成されるレポート、n は最後に生成されるレポートに付けられます。デフォルト値は ON です。

OFF

連続番号を省略します。IBIWF\_mprefix で指定したテキストのみが適用されます。

## 構文

### ハイパーリンクからの複数レポート制御

```

text

text
```

説明

alias

WebFOCUS Client がインストールされているディレクトリを指定します。Application Server または Web サーバは、エイリアスを使用して物理ディレクトリに論理名を割り当てます。WebFOCUS のデフォルトエイリアスは ibi\_apps です。このエイリアスは、製品のインストール時にカスタマイズすることができ、インストールはこのエイリアスを構成します。

別の Web サーバで WebFOCUS を呼び出すには、Application Server または Web サーバのサーバ名とポート番号を含めた完全修飾 URL を指定します。以下はその例です。

```
<A HREF="http://servername:port/alias/WFServlet?
IBIF_ex=report1[&var=value[&var=value]...] "> text
<A HREF="http://servername:port/alias/WFServlet?
IBIF_ex=report2[&var=value[&var=value]...] "> text
```

#### servername

WebFOCUS がインストールされている Application Server または Web サーバの名前です。

#### port

サーバのリスナポート番号です。

#### report1, report2

実行するプロシジャ名です。

#### var=value

WebFOCUS Client 変数およびその値です。

変数と値の組み合わせを 2 つ以上渡すことはできますが、それぞれの組み合わせの間にブランクを含めることはできません。変数と値の組み合わせを区切るには、区切り文字としてアンパサンド (&) を使用します。値の最大長は 80 バイトです。

値にブランクが埋め込まれている場合は、そのブランクをプラス記号 (+) または文字列の「%20」に置き換えます。

変数および有効な値のリストは、263 ページの「[WebFOCUS Client 変数](#)」を参照してください。

#### text

起動ページに表示するテキストです。このテキストがプロシジャを実行するハイパーリンクになります。

## 例

### インデックス値を持つ 2 つのレポートの表示

次の例は、インデックス値を追加した 2 つのレポートを表示する方法を示しています。

1. 2 つのリクエストで構成された「TWOPTS」というプロシジャを作成します。最初のリクエストは総売上高に関するレポート、2 目目のリクエストは売上数量と売上高に関するレポートを生成します。

#### プロシジャ

```
TABLE FILE GGSALES
SUM DOLLARS BY PRODUCT
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
-RUN
```

```
TABLE FILE GGSALES
SUM UNITS DOLLARS BY PRODUCT
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
-RUN
```

2. 「TWOLNCH」という起動ページを作成します。この起動ページには、WebFOCUS Servlet を呼び出して、実行するプロシジャ名を WebFOCUS Servlet に渡すハイパーリンクが含まれています。また、変数を設定して、識別子 (Sales Analysis Report) および連続番号 (1 と 2) を各レポートに渡します。

#### 起動ページ

```
<HTML>
<BODY>
<A HREF="/ibi_apps/WFServlet?IBIF_ex=tworpts
&IBIWF_mreports=index&IBIWF_mprefix=Sales+Analysis+Report">
Run report.
</BODY>
</HTML>
```

3. App Studio を使用してプロシジャおよび起動ページを公開します。

4. ブラウザに表示された起動ページを実行し、[Run report] をクリックして総売上高のレポートを生成します。

<a href="#">Sales Analysis</a>		
<a href="#">Report 1</a>		
<a href="#">Sales Analysis</a>	<u>Product</u>	<u>Dollar Sales</u>
<a href="#">Report 2</a>	Biscotti	5263317
	Capuccino	2381590
	Coffee Grinder	2337567
	Coffee Pot	2449585
	Croissant	7749902
	Espresso	3906243
	Latte	10943622
	Mug	4522521
	Scone	4216114
	Thermos	2385829

[Sales Analysis Report 2] をクリックして、売上数量と売上高のレポートを生成します。

<a href="#">Sales Analysis</a>			
<a href="#">Report 1</a>			
<a href="#">Sales Analysis</a>	<u>Product</u>	<u>Unit Sales</u>	<u>Dollar Sales</u>
<a href="#">Report 2</a>	Biscotti	421377	5263317
	Capuccino	189217	2381590
	Coffee Grinder	186534	2337567
	Coffee Pot	190695	2449585
	Croissant	630054	7749902
	Espresso	308986	3906243
	Latte	878063	10943622
	Mug	360570	4522521
	Scone	333414	4216114
	Thermos	190081	2385829

## 構文 プロシジャから実行する複数レポートの制御

```
-TYPE WEBFOCUS CGIVAR var=value
.
.
.
-RUN
```

### 説明

**var=value**

WebFOCUS Client 変数およびその値です。1 つの -TYPE コマンドに変数と値の組み合わせを 2 つ以上含めることはできません。

変数および有効な値のリストは、263 ページの「[WebFOCUS Client 変数](#)」を参照してください。

**注意:**各リクエストの末尾に -RUN コマンドを含めて、その位置までに追加された一連の -TYPE コマンドを実行します。

## 例 複数レポートを左右に並べて表示

次の例は、複数のレポートを左右に並べて表示する方法を示しています。

1. 2 つのリクエストで構成された「SIDERPTS」というプロシジャを作成します。最初のリクエストは、店舗別の売上高に関するレポートを生成します。2 つ目のリクエストは、製品別の売上高に関するレポートを生成します。

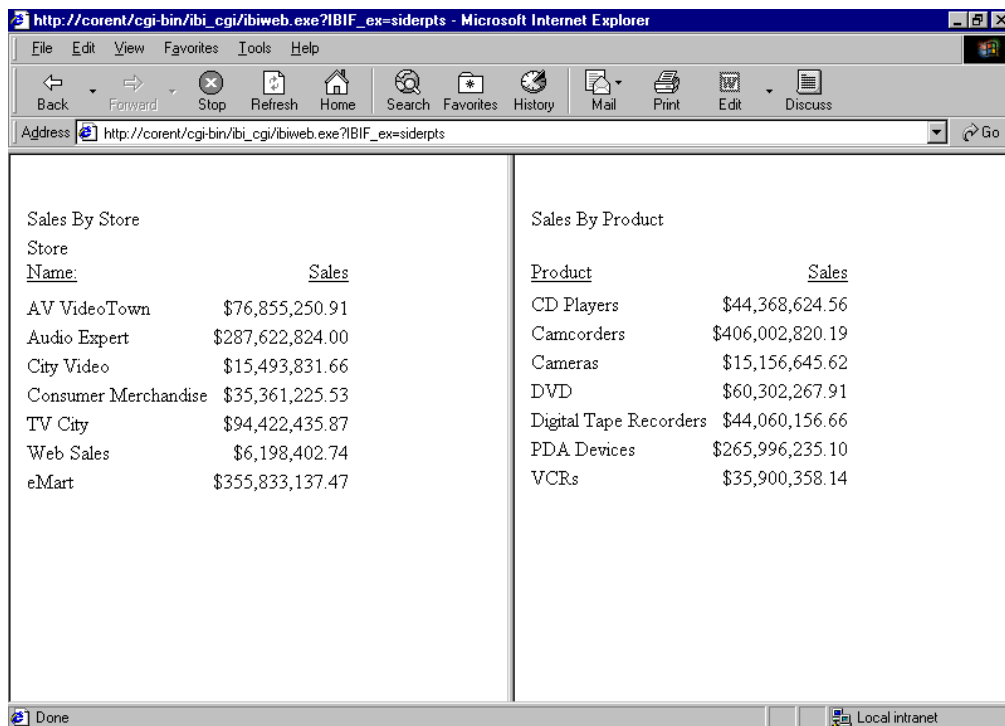
-TYPE コマンドを使用して 2 つのフレームで構成されたページを作成し、これらのレポートを左右に並べて表示します。

**プロシジャ**

```
-TYPE WEBFOCUS CGIVAR IBIWF_mreports=FRAME
-TYPE WEBFOCUS CGIVAR IBIWF_mrcolumns=2
TABLE FILE CENTORD
HEADING
"Sales By Store"
SUM LINEPRICE AS 'Sales'
BY SNAME
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
-RUN
TABLE FILE CENTORD
HEADING
"Sales By Product"
" "
SUM LINEPRICE AS 'Sales'
BY PRODCAT AS 'Product'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

2. プロシジャを実行する起動ページを作成します。
3. App Studio を使用してプロシジャおよび起動ページを公開します。

4. 起動ページをブラウザに表示してレポートを実行します。



Sales By Store		Sales By Product	
Store Name:	Sales	Product	Sales
AV VideoTown	\$76,855,250.91	CD Players	\$44,368,624.56
Audio Expert	\$287,622,824.00	Camcorders	\$406,002,820.19
City Video	\$15,493,831.66	Cameras	\$15,156,645.62
Consumer Merchandise	\$35,361,225.53	DVD	\$60,302,267.91
TV City	\$94,422,435.87	Digital Tape Recorders	\$44,060,156.66
Web Sales	\$6,198,402.74	PDA Devices	\$265,996,235.10
eMart	\$355,833,137.47	VCRs	\$35,900,358.14

## 例 目次のレポート名と連続番号の表示

次の例は、2つのレポートの名前と連続番号を目次に表示する方法を示しています。

1. 2つのリクエストで構成された「HTMPDF1」というプロシジャを作成します。最初のリクエストは HTML フォーマットで、2つ目のリクエストは PDF フォーマットで、それぞれのフォーマットで総売上高に関するレポートを生成します。

-TYPE コマンドを使用して変数を設定し、識別子 (HTML Report と PDF Report) および連続番号 (1 と 2) を各レポートに渡します。目次を作成するコマンドは、プロシジャの先頭に1回だけ記述する必要があります。

### プロシジャ

```

- TYPE WEBFOCUS CGIVAR IBIWF_mreports=INDEX
- TYPE WEBFOCUS CGIVAR IBIWF_mprefix=HTML Report
TABLE FILE GGSales
SUM DOLLARS BY PRODUCT
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
- RUN

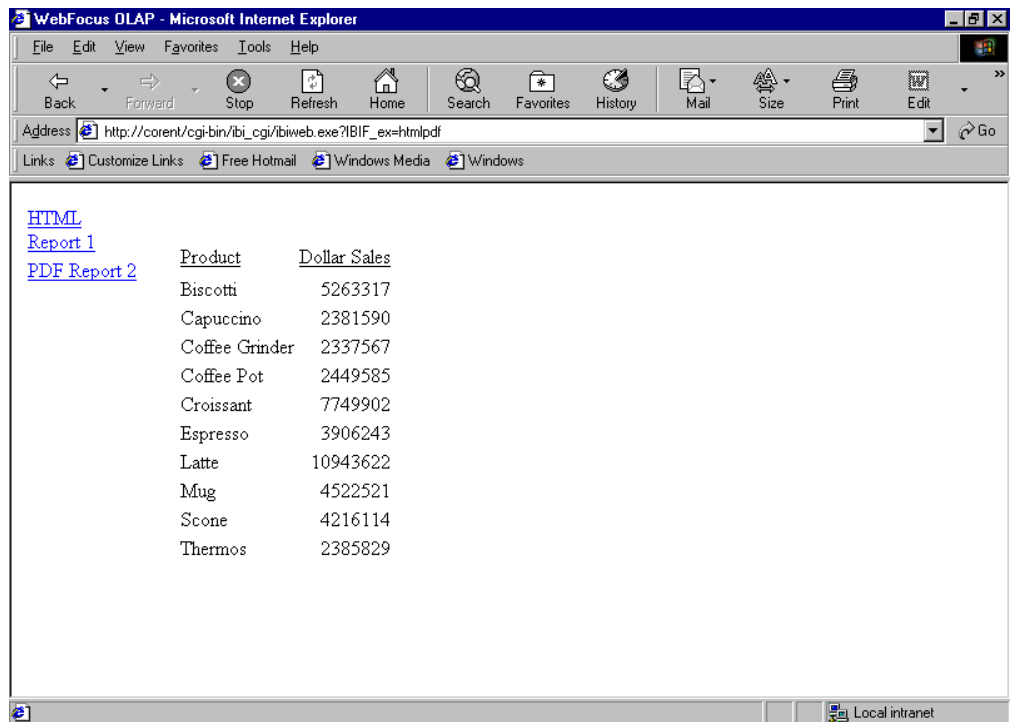
```

```

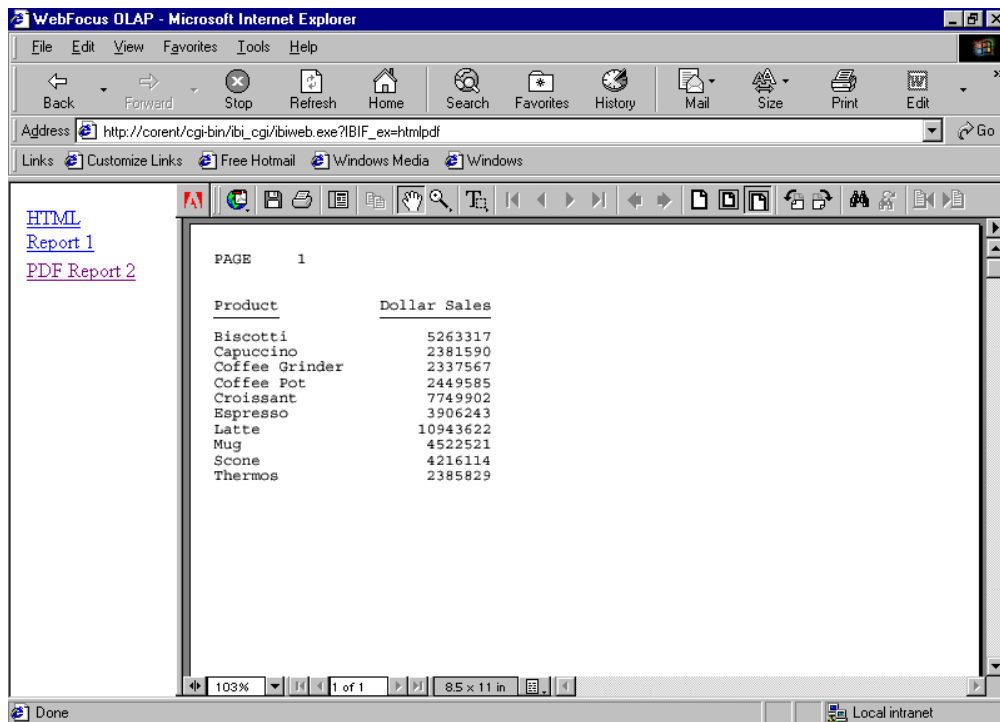
TABLE FILE GGSales
- TYPE WEBFOCUS CGIVAR IBIWF_mprefix=PDF Report
SUM DOLLARS BY PRODUCT
ON TABLE PCHOLD FORMAT PDF
END

```

2. プロシジャを実行する起動ページを作成します。
3. App Studio を使用してプロシジャおよび起動ページを公開します。
4. ブラウザで起動ページを実行します。HTML フォーマットのレポートが表示されます。



5. [PDF Report 2] をクリックして、PDF レポートを Acrobat Reader で表示します。



## 例 目次のレポート名のみ表示

次の例は、2 つのレポートの名前のみを目次に表示する方法を示しています。

1. 2 つのリクエストで構成された「HTMPDF2」というプロシジャを作成します。-TYPE WEBFOCUS CGIVAR IBIWF\_index=OFF コマンドを追加して、目次に表示する名前から連続番号を省略します。

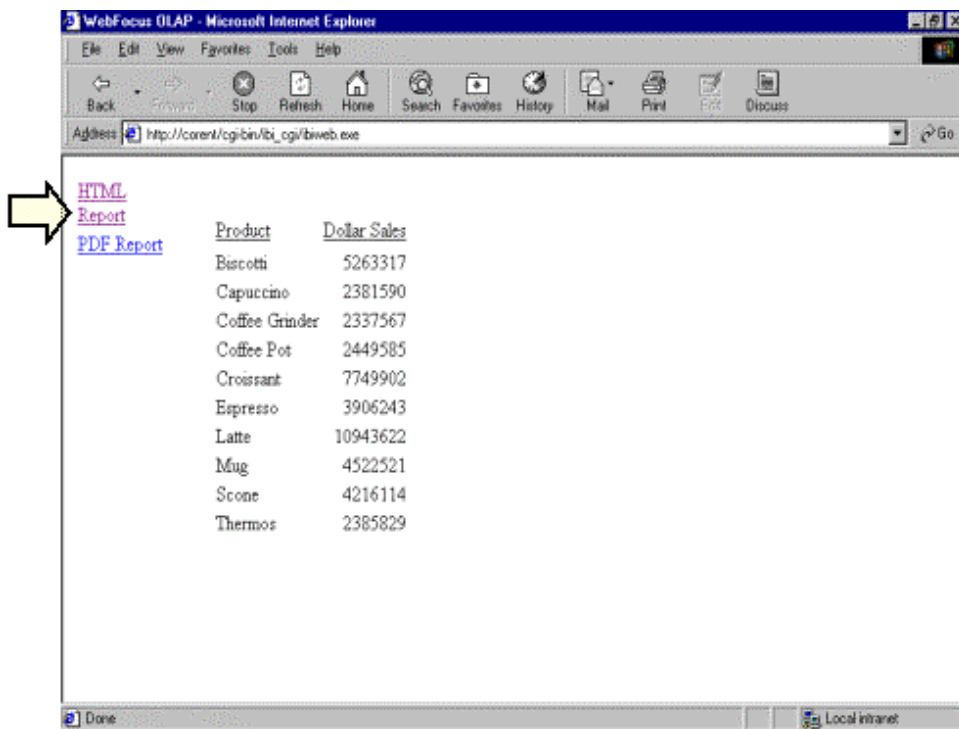
### プロシジャ

```
-TYPE WEBFOCUS CGIVAR IBIWF_mreports=INDEX
-TYPE WEBFOCUS CGIVAR IBIWF_index=OFF
-TYPE WEBFOCUS CGIVAR IBIWF_mprefix=HTML Report
TABLE FILE GGSALES
SUM DOLLARS BY PRODUCT
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
-RUN

-TYPE WEBFOCUS CGIVAR IBIWF_mprefix=PDF Report
TABLE FILE GGSALES
SUM DOLLARS BY PRODUCT
ON TABLE PCHOLD FORMAT PDF
END
```

2. プロシジャを実行する起動ページを作成します。
3. App Studio を使用してプロシジャおよび起動ページを公開します。

4. ブラウザで起動ページを実行します。連続番号が省略されて、レポートの名前のみが目次に表示されます。



## 例 目次の連続番号のみの表示

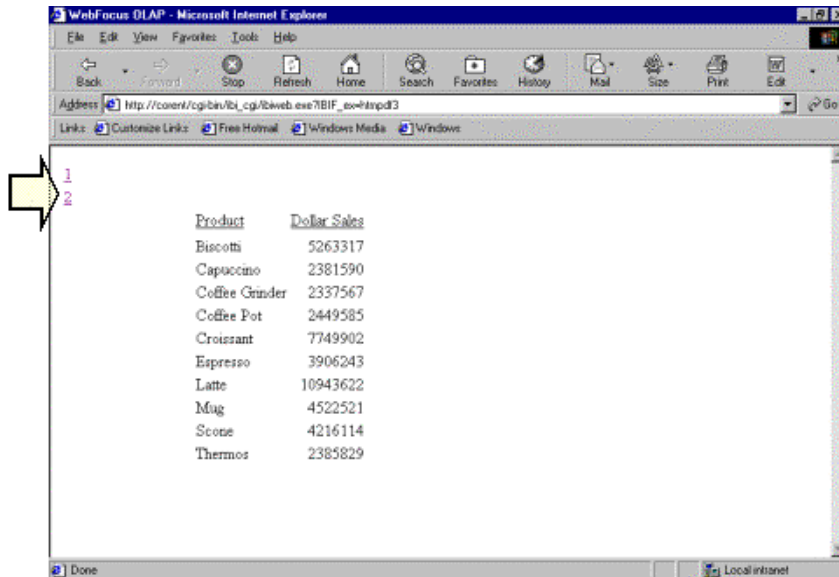
次の例は、2つのレポートの連続番号のみを目次に表示する方法を示しています。

1. 2つのリクエストで構成された「HTMPDF2」というプロシジャを作成します。-TYPE WEBFOCUS CGIVAR IBIWF\_mprefix= コマンドを追加して、レポートの名前を省略します。

```
-TYPE WEBFOCUS CGIVAR IBIWF_mreports=INDEX
-TYPE WEBFOCUS CGIVAR IBIWF_mprefix=
TABLE FILE GGSales
SUM DOLLARS BY PRODUCT
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
-RUN
```

```
TABLE FILE GGSALES
SUM DOLLARS BY PRODUCT
ON TABLE PCHOLD FORMAT PDF
END
```

2. プロシジャを実行する起動ページを作成します。
3. App Studio を使用してプロシジャおよび起動ページを公開します。
4. ブラウザで起動ページを実行します。レポートの名前が省略されて、連続番号のみが目次に表示されます。



## HTML Web ページへの CSRF トークンの追加

クロスサイトリクエストフォージェリ (CSRF) トークンを各レポートリクエストに追加すると、認証済みの Web アプリケーション上で、望ましくない操作の実行へとエンドユーザを誘導する CSRF 攻撃が防御されます。

HTML Web ページからの HTTP POST メッセージとして発行されるレポートリクエストに CSRF トークン変数を含めるには、WebFOCUS Client の site.wfs ファイルに CSRF トークン変数を追加し、その CSRF 変数への参照を -HTMLFORM ダイアログマネージャプロシジャに追加します。これにより、その CSRF 変数への参照が含まれた任意のレポートリクエストを使用して、Client 側の site.wfs スクリプトロジックから CSRF 変数をサーバに送信することが可能になります。その応答として、HTML Web ページを生成する HTTP POST メッセージで、認証済みユーザセッションの開始時に自動生成された CSRF トークン名と CSRF トークン値がサーバから返されます。

-HTMLFORM ダイアログマネージャプロシジャに CSRF トークン変数への参照を追加するには、2つの要件があります。1つ目の要件として、CSRF トークン名と CSRF トークン値の変数は、site.wfs ファイルに割り当て、Client 側のロジックに追加する必要があります。これを実行するには、管理コンソールの [カスタム設定] ページに次の値を入力し、更新されたページを保存します。

```
<SET>IBI_CSRF_Token_Name(PASS)
<SET>IBI_CSRF_Token_Value(PASS)
```

2つ目の要件として、CSRF 変数への参照は、これらを使用する各 -HTMLFORM ダイアログマネージャプロシジャの -HTMLFORM BEGIN/END セクションに追加する必要があります。以下はこの例を示しています。

```
-HTMLFORM BEGIN
<body onload="document.form.submit()">
<form name=form id=form action="/ibi_apps/run/ibfs" method="post">
<input type="hidden" name="IBFS_path" value="/WFC/Repository/sales/
salesbyregion.fex" />
<input type="hidden" name="IBFS_action" value="run" />
<input type="text" name="COUNTRY" value="ITALY" />
<input type="hidden" name="!IBI.AMP.IBI_CSRF_Token_Name;" value="!
IBI.AMP.IBI_CSRF_Token_Value;" />
</form>
-HTMLFORM END
```

site.wfs ファイルに CSRF トークン変数を追加するには、管理者が [カスタム設定] ファイルに入力し、変更を保存する必要があります。site.wfs ファイルに割り当てられた有効な CSRF トークン名および CSRF トークン値の変数は、これらの変数への参照を含むレポートリクエストごとに、WebFOCUS Client からサーバに送信されます。

## JavaScript によるドリルダウンレポートの作成

ここでは、JavaScript を使用してドリルダウンレポートを作成する方法について説明します。JavaScript 関数を呼び出してレポートの集計コンポーネントから値を渡し、詳細コンポーネントのコンテンツを動的に特定する方法について説明します。

次の例は、プロシジャのスタイルシートに JavaScript 関数へのハイパーリンクを指定する方法を示しています。ハイパーリンクを定義すると、ユーザはレポートに表示された関連オブジェクトを選択して関数を実行することができます。

JavaScript の機能と構文についての詳細は、JavaScript のマニュアルを参照してください。

## 構文

### JavaScript 関数へのハイパーリンクを指定するには

`TYPE=type, {COLUMN|ACROSSCOLUMN}=fieldname, JAVASCRIPT=func[(value)], $`

#### 説明

##### type

ハイパーリンクとして機能するレポートコンポーネントです。TYPE 属性およびその値は、スタイルシートで最初に定義しておく必要があります。

たとえば、TYPE=DATA を使用してレポートのデータフィールドからのハイパーリンクを設定するか、TYPE=REPORT を使用してレポートの任意のコンポーネントからのハイパーリンクを設定します。

##### fieldname

データソースのフィールドの名前です。このフィールドの値を選択すると、ハイパーリンクが実行されます。

##### func

JavaScript 関数の名前です。

JAVASCRIPT=func のコードの最大長は、渡された値も含めて 800 バイトです。このコードは複数行に記述することができます。コードを複数行に分割して記述する場合は、継続文字として 1 行目の最後に円記号 (¥) を追加します。ブランクが必要な箇所で行を分割する場合は、ブランクを円記号 (¥) の前に配置するか、次の行の先頭文字として配置する必要があります。

たとえば、次のように記述します。

```
JAVASCRIPT=myfunc(COUNTRY¥CAR MODEL), $
```

コードは正しく 2 行に分割されています。

##### value

JavaScript 関数に渡される値です。次のいずれかの方法で値を指定します。

- ❑ レポートのフィールド名として指定する。
- ❑ 定数として指定する。この値は一重引用符 (') で囲む必要があります。

- ❑ ダイアログマネージャの変数として指定する。プロシジャの一部 (インライン) であるスタイルシート内の変数のみを指定することができます。

通常、変数は定数値を渡すために使用しますが、その場合は '&ABC' のように一重引用符 (') で囲む必要があります。

### 例 JavaScript を使用したドリルダウンレポートの作成

次の例では、ドリルダウンレポートの集計コンポーネントを作成して、店舗別 (Store Code) の注文数を月単位で表示します。ユーザが特定の店舗コードを選択すると、ハイパーリンクが JavaScript 関数を呼び出してデータ計算を実行し、選択した店舗の詳細情報を表示します。

1. 「JDRILL」というプロシジャを作成します。左側に表示されたアルファベット文字は、そのコードの意味について説明する注記に対応しています。

```
SET MESSAGE = OFF
TABLE FILE GGORDER
SUM QUANTITY BY STORE_CODE ACROSS ORDER_DATE
IF ORDER_DATE GT 12/31/96
a. ON TABLE HOLD
END
-RUN
TABLE FILE HOLD
HEADING CENTER
"Store Sales Analysis Using JavaScript"
b. PRINT E01 AS 'Store,Code' E02 AS 'Jan' E03 AS 'Feb' E04 AS 'Mar'
 E05 AS 'Apr' E06 AS 'May' E07 AS 'Jun'
 E08 AS 'Jul' E09 AS 'Aug' E10 AS 'Sep'
 E11 AS 'Oct' E12 AS 'Nov' E13 AS 'Dec'

FOOTING CENTER
"Please click on the store code to summarize the store's data."
c. ON TABLE HOLD AS JAVATEMP FORMAT HTMTABLE
d. ON TABLE SET STYLE *
 TYPE=DATA,COLUMN=STORE_CODE,$
 JAVASCRIPT=conprint(E01 E02 E03 E04 E05 E06 E07 E08 E09 E10 E11 ¥
 E12 E13),$
 COLOR=GREEN,STYLE=ITALIC,$

 TYPE=TITLE,COLOR=RED,STYLE=BOLD,$
 TYPE=HEADING,COLOR=BLUE,STYLE=ITALIC,SIZE=11,$
 TYPE=FOOTING,COLOR=BLUE,STYLE=ITALIC,$
 END STYLE
END
e. -HTMLFORM BEGIN
<HTML>
```

```

f. <SCRIPT LANGUAGE="JavaScript">
 var spacer = "....."; var pos=0;
 var aaaa;
 function conprint(aaax,lsyr,aa1,aa2,aa3,aa4,aa5,aa6,aa7,
 aa8,aa9,aa10,aa11,aa12)
 {
 var spacer=" ";

 lsyrave=parseFloat(lsyr);
 a1=parseFloat(aa1);
 a2=parseFloat(aa2);
 a3=parseFloat(aa3);
 a4=parseFloat(aa4);
 a5=parseFloat(aa5);
 a6=parseFloat(aa6);
 a7=parseFloat(aa7);
 a8=parseFloat(aa8);
 a9=parseFloat(aa9);
 a10=parseFloat(aa10);
 a11=parseFloat(aa11);
 a12=parseFloat(aa12);
 gotota=eval(a1 + a2 + a3 + a4 + a5 + a6 + a7 + a8 + a9 + a10 +
 a11 + a12);
 goavea=gotota/12;
 gotot=Math.round(gotota);
 goave=Math.round(goavea);
 diffavea=goavea-lsyrave;
 diffave=Math.round(diffavea);
g. document.form1.text1.value=gotot;
 document.form1.text2.value=goave;
 document.form1.store.value=aaax;
 arraygo = new Array(13);
 arraygo[1]=a1;
 arraygo[2]=a2;
 arraygo[3]=a3;
 arraygo[4]=a4;
 arraygo[5]=a5;
 arraygo[6]=a6;
 arraygo[7]=a7;
 arraygo[8]=a8;
 arraygo[9]=a9;
 arraygo[10]=a10;
 arraygo[11]=a11;
 arraygo[12]=a12;
 minval=100000;
 maxval=0;
 mnmmmax=0;
 mnmmmin=0;

```

```

for(i = 1; i <= 12; i++)
{
if(arraygo[i] > maxval)
{
mnmmmax=i;
maxval = arraygo[i];
}
if(arraygo[i] < minval)
{
mnmmmin=i;
minval = arraygo[i];
}
}
rng1=maxval - minval;
rng=Math.round(rng1);
mnms=new Array(13);
mnms[1]="January";
mnms[2]="February";
mnms[3]="March";
mnms[4]="April";
mnms[5]="May";
mnms[6]="June";
mnms[7]="July";
mnms[8]="August";
mnms[9]="September";
mnms[10]="October";
mnms[11]="November";
mnms[12]="December";
document.form1.themax.value=maxval;
document.form1.themin.value=minval;
document.form1.range.value=rng;
document.form1.mnmmmax.value=mnms[mnmmmax];
document.form1.mnmmmin.value=mnms[mnmmmin]; }
</SCRIPT>
<BODY>
<FORM name="form1">
h. <!--WEBFOCUS TABLE JAVATEMP>
<INPUT TYPE=text NAME="text1" SIZE="10"> Units sold for store
<INPUT TYPE=text NAME="store" SIZE="10"> Monthly average of
<INPUT TYPE=text NAME="text2" SIZE="10">

 Top selling
month is
<INPUT TYPE=text NAME="mnmmmax" SIZE="10"> with
<INPUT TYPE=text NAME="themax" SIZE="10">units
 Slowest
month was
<INPUT TYPE=text NAME="mnmmmin" SIZE="10"> with
<INPUT TYPE=text NAME="themin" SIZE="10">units
 Range
between best and slowest months
<INPUT TYPE=text NAME="range" SIZE="10">units </FORM>
</BODY>
</HTML>
I. -HTMLFORM END
-EXIT

```

- a. このコマンドは、1997 年のデータのレポート出力を、ネイティブのマシンフォーマットを使用して「HOLD」という一時ファイルに保存します。レポートにはこのデータのみを使用するため、このサーバ抽出ファイルが作成されたことにより、後続の処理が高

速化されます。

- b. このコードは、レポートのフォーマットを設定して説明的なフィールドタイトルをレポートに含めます。最初のフィールドタイトルは「Store Code」です。その後に、各店舗の月単位の合計数量が続きます。
  - c. このコマンドは、レポート出力を HTML フォーマットで一時ファイルに保存します。このファイルには、「JAVATEMP」という名前が付けられます。このファイルは、後で作成する HTML ページに結合されます (注記 e を参照)。
  - d. このスタイルシートは、「conprint」という JavaScript 関数へのハイパーリンクを指定します。このコードは、店舗コードと月ごとの値を関数に渡します。
  - e. ダイアログマネージャの -HTMLFORM BEGIN コマンドは、JavaScript 関数が定義された HTML ページの開始を指示します。このページにレポート出力が埋め込まれます。
  - f. HTML コードが JavaScript 関数を宣言し、関数に値を渡します。
  - g. JavaScript は、HTML ページに表示する値に変数名を割り当てます。
  - h. WebFOCUS が HTML コメントを読み取り、そのコメントを JAVATEMP に保存されているレポート出力に置換します。
- 注意：**HTML コメント行は「->」で閉じる必要があります。また、単一の「>」で閉じることできますが、その場合はコメント行に他の HTML タグを含めることはできません。
- i. ダイアログマネージャの -HTMLFORM END コマンドが HTML ページの終了を指示します。
2. ユーザがレポートの実行に使用する起動ページを作成します。

3. 起動ページを実行して、リンクをクリックします。集計コンポーネントが表示されます。

[Store Sales Analysis Using JavaScript](#)

Store Code	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
R1019	3961	3119	3951	3983	4037	2888	4996	3943	4526	3962	3317	5305
R1020	4152	3944	3927	4866	3697	3454	4449	4785	4461	4588	4454	4139
R1040	4189	3766	4342	3741	4681	3953	3775	4655	3731	4127	3021	3794
R1041	4650	4818	5061	3000	4528	4751	5396	4918	3443	4995	3873	5611
R1044	3836	3599	3008	4356	4126	4414	4413	3038	4044	4109	4010	3493
R1088	4039	5224	3700	3586	3435	3947	4083	3311	4889	3710	4382	4256
R1100	4190	3652	4244	5146	4068	3848	4348	3760	4312	4151	3987	3963
R1109	3716	4031	4350	3894	4396	4225	4221	4661	4421	3447	4161	4175
R1200	3849	4604	3559	3657	4237	4688	4493	4663	3902	2995	3701	4303
R1244	4169	3903	4014	4027	4049	4028	4111	4299	4061	4913	4730	3966
R1248	3961	3283	4503	3854	5088	5324	4411	3658	3596	4586	4374	4358
R1250	3195	4102	4492	4270	3579	4486	3781	3488	4233	4309	3759	3946

[Please click on the store code to summarize the store's data.](#)

Units sold for store  Monthly average of

Top selling month is  with  units  
 Slowest month was  with  units  
 Range between best and slowest months  units

4. 店舗コードの R1019 をクリックして、詳細コンポーネントを表示します。

Units sold for store  Monthly average of

Top selling month is  with  units  
 Slowest month was  with  units  
 Range between best and slowest months  units

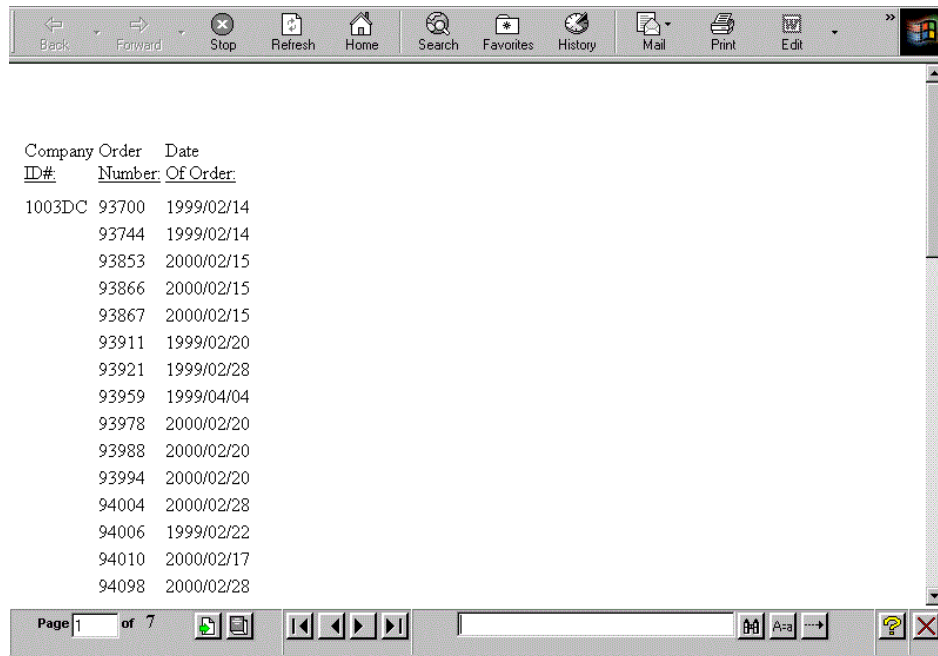
## Web ビューア機能

通常、Web サーバはレポート全体をブラウザに返します。ブラウザは、すべてのレポートを受信してからレポートを表示します。

Web ビューア機能を使用すると、レポート全体を一括ダウンロードする代わりに、最大 99999 ページのレポートの 1 ページのみをブラウザにダウンロードすることができます。残りのページは、ユーザが要求するまで Web サーバに保持されます。この機能により、レポートの先頭ページが表示されるまでの待ち時間が短縮されます。この機能は、長いレポートを扱う場合に効果があります。

Web ビューアは、WebFOCUS ビューアに実装されています。Web ビューアを使用するには、レポートを HTML フォーマットで出力する必要があります。HTML フォーマットは、WebFOCUS Client から送信されるリクエストのデフォルト設定です。

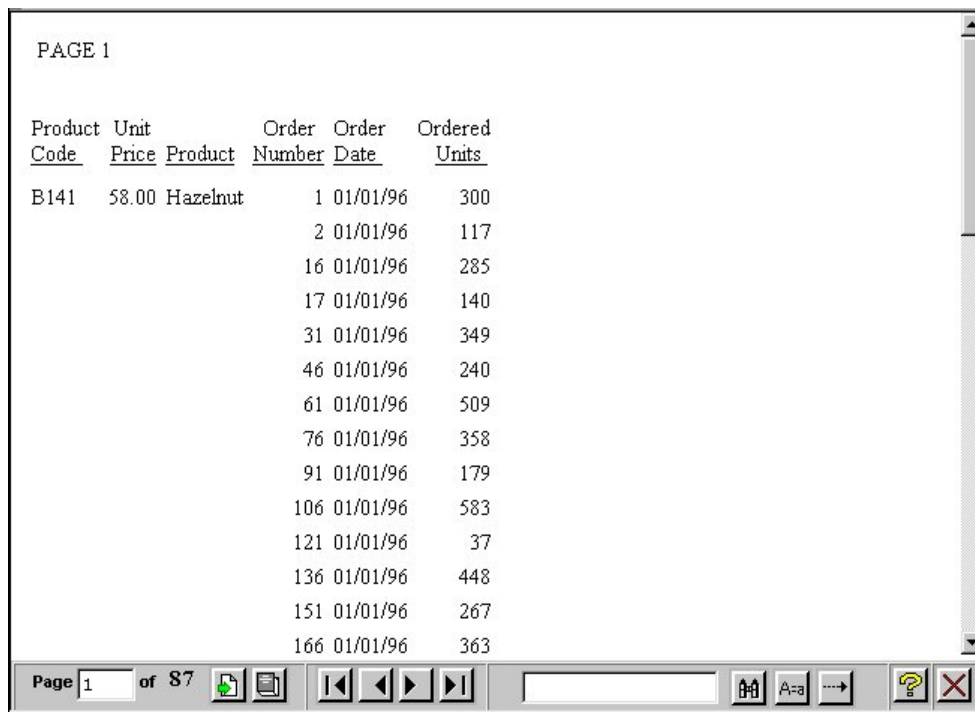
下図は、WebFOCUS ビューアに表示されたレポートを示しています。



Company	Order ID#	Date	Number	Of Order
1003DC	93700	1999/02/14		
	93744	1999/02/14		
	93853	2000/02/15		
	93866	2000/02/15		
	93867	2000/02/15		
	93911	1999/02/20		
	93921	1999/02/28		
	93959	1999/04/04		
	93978	2000/02/20		
	93988	2000/02/20		
	93994	2000/02/20		
	94004	2000/02/28		
	94006	1999/02/22		
	94010	2000/02/17		
	94098	2000/02/28		

## TIBCO WebFOCUS ビューアによるレポートのナビゲート

Web ビューアが指定されたレポートを実行すると、WebFOCUS ビューアが自動的に開き、レポートの最初のページを表示します。ビューアを開く際のターゲットフレームおよびビューアを閉じる際に表示するホームページを指定することができます。下図のように、ビューアは [レポートパネル] と [ビューアコントロールパネル] の 2 つのパネルで構成されています。



PAGE 1

Product Code	Unit Price	Product	Order Number	Order Date	Ordered Units
B141	58.00	Hazelnut	1	01/01/96	300
			2	01/01/96	117
			16	01/01/96	285
			17	01/01/96	140
			31	01/01/96	349
			46	01/01/96	240
			61	01/01/96	509
			76	01/01/96	358
			91	01/01/96	179
			106	01/01/96	583
			121	01/01/96	37
			136	01/01/96	448
			151	01/01/96	267
			166	01/01/96	363

Page 1 of 87

[レポートパネル] は大きい方のウィンドウで、レポート出力の 1 ページが表示されます。最初にレポートを実行すると、[レポートパネル] にはレポート出力の最初のページが表示されます。[ビューアコントロールパネル] には、特定ページの表示、レポート全体の Web サーバへの送信、ドキュメント内の特定文字列情報の検索を行うためのコントロールが用意されています。

## 構文

### WebFOCUS ビューアの有効化

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

または

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

## 説明

**OFF**

Web ビューアを無効にします。WebFOCUS は、標準のブラウザウィンドウにレポート全体をダウンロードします。デフォルト値は OFF です。

**ON**

Web ビューアを有効にします。WebFOCUS は、ビューアのブラウザにレポートの先頭ページをダウンロードします。1 回に表示される行数は、Windows のデスクトップの解像度の設定により異なります。

## 例

**WebFOCUS ビューアの有効化**

次の例では、プロシジャおよび起動ページは WebFOCUS で実行されます。これらのプロシジャおよび起動ページは、この環境でテスト、実行する必要があります。

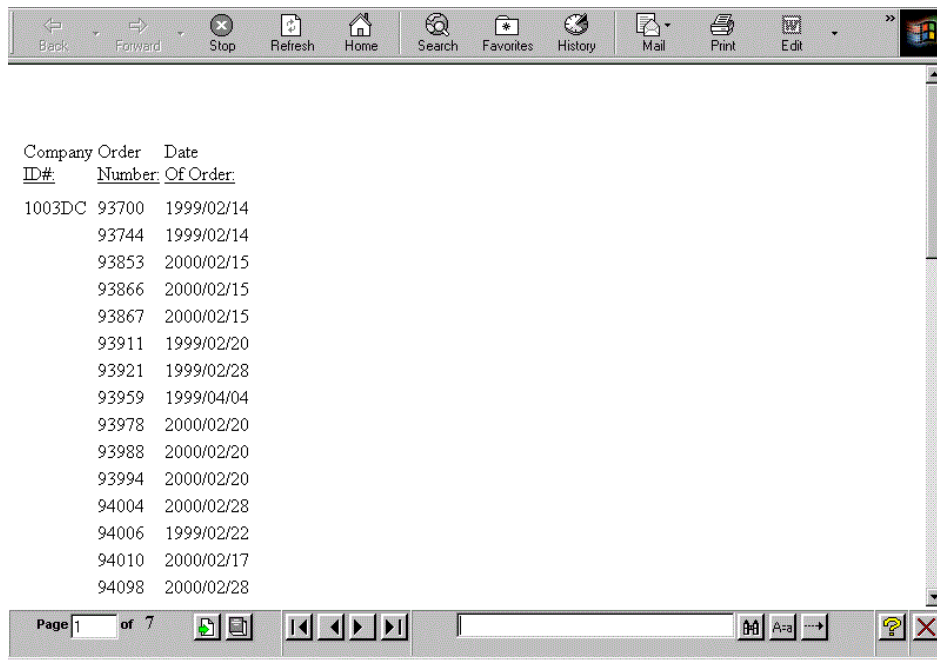
1. 「ONDEMAND」というプロシジャを作成して、ビューアに特定の店舗に関する注文レポートを表示します。

**プロシジャ**

```
SET WEBVIEWER=ON
TABLE FILE CENTORD
PRINT ORDER_NUM ORDER_DATE
BY STORE_CODE
WHERE STORE_CODE EQ '1003DC'
ON TABLE SET PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
ENDSTYLE
END
```

2. ユーザがレポートの実行に使用する起動ページを作成します。

3. 起動ページを実行して、リンクをクリックします。レポートがビューアに表示されます。



### TIBCO WebFOCUS ビューアを開く/閉じる

WebFOCUS ビューアを開く際のターゲットフレームおよびビューアを閉じる際に表示するホームページを指定することができます。

### 構文

#### WebFOCUS ビューアをターゲットフレームで開く

```
SET WEBVIEWTARG = {target_frame|OFF}
```

#### 説明

##### target\_frame

ブラウザ内の既存フレームの名前、または予約済み HTML ターゲットフレームのいずれかの名前です。

##### \_blank

ビューア を新しいブラウザウィンドウで開きます。アクセシビリティを有効にしていないレポートでは、これがデフォルト値です。

**\_self**

ビューアをアンカーと同一のフレームで開きます。

**\_parent**

ビューアをアンカーを含む上位の親フレームで開きます。

**\_top**

ビューアを現在のブラウザウィンドウで開きます。

**OFF**

レポートを実行したフレームでビューアを開きます。アクセシビリティを有効にしたレポートでは、これがデフォルト値です。

**構文****WebFOCUS ビューアを閉じた際のホームページの表示**

```
SET WEBVIEWHOME = {home_URL|OFF}
```

**説明****home\_URL**

ビューアを閉じた際に HTML ページを表示する有効な URL です。

**OFF**

ビューアを閉じた際にブランクのブラウザウィンドウを表示します。別のレポートを実行するには、URL を別途入力する必要があります。デフォルト値は OFF です。

**参照****WebFOCUS ビューアを閉じる**

コントロールフレームに表示された [閉じる] ボタンをクリックすると、ビューアが閉じて、レポートが Web サーバから削除されます。次にブラウザに表示されるページは、WEBVIEWTARG および WEBVIEWHOME の設定に基づいて次のようになります。

- ☐ WEBVIEWTARG を `_blank` に設定した場合、ビューアが表示されたウィンドウは削除されます。ブラウザのどのフレームにもページが表示されないため、WEBVIEWHOME 設定は無効になります。
- ☐ WEBVIEWTARG に `_blank` 以外の値を設定した場合、[閉じる] をクリックした時の動作は WEBVIEWHOME の設定に基づいて次のようになります。
- ☐ WEBVIEWHOME に特定の URL を設定した場合、ブラウザはビューアが表示されていたフレームにその URL に関連付けられたページを表示します。
- ☐ WEBVIEWHOME を OFF に設定した場合、ブラウザはブランクページを表示します。

## TIBCO WebFOCUS ユーアのボタン表示の制御

ビューアに [閉じる]、[ヘルプ]、[すべてのページ] ボタンを表示するかどうかを指定するコマンドを発行することができます。

### 構文 閉じるボタンの表示と非表示の制御

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

または

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

説明

ON

[閉じる] ボタンを表示します。デフォルト値は ON です。

OFF

[閉じる] ボタンを表示しません。

### 構文 ヘルプボタンの表示と非表示の制御

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

または

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

説明

ON

[ヘルプ] ボタンを表示します。デフォルト値は ON です。

OFF

[ヘルプ] ボタンを表示しません。

### 構文 「すべてのページ」 ボタンの表示と非表示の制御

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

または

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

## 説明

## ON

[すべてのページ] ボタンを表示します。デフォルト値は ON です。

## OFF

[すべてのページ] ボタンを表示しません。

## 手順

## 最終ページボタンを有効にするには

1. Internet Explorer の [ツール] メニューから、[インターネット オプション] を選択します。
2. [インターネットオプション] ダイアログボックスが開きます。
3. [インターネット一時ファイル] ボックスの [設定] をクリックします。  
[設定] ダイアログボックスが開きます。
4. [Web サイトを表示するたびに確認する] ラジオボタンを選択します。
5. [OK] をクリックし、変更を適用して [設定] ダイアログボックスを閉じます。
6. [OK] をクリックし、[インターネットオプション] ダイアログボックスを閉じます。

## ビューアコントロールパネルの使用

下図のように、[ビューアコントロールパネル] の最下部には、レポート内を移動し、文字列を検索するためのコントロールがあります。[ビューアコントロールパネル] のナビゲーションコントロールを使用して、次または前のページ、最初または最後のページ、あるいは特定のページを表示することができます。検索機能を使用すると、ビューアでレポートの全ページから、指定した文字列を検索できます。



**注意：**文字間に複数のブランクを使用している場合でも HTML では 1 つのブランクとして表示されるため、検索文字列を指定する場合は、文字間のブランクを実際の数だけ入力する必要があります。

## 手順 レポート内を移動するには

[ビューアコントロールパネル] でレポート内のページを表示するには、いくつかの方法があります。

- 特定のページを表示するには、次の手順を実行します。

1. 下図のように、[ページ] テキストボックスにページ番号を入力します。



2. 下図のような [ページへ移動] をクリックします。



- 前または次のページを連続して表示するには、下図のような [前のページ] または [次のページ] をクリックします。



- レポートの最初または最後のページを表示するには、下図のような [最初のページ] または [最後のページ] をクリックします。



- レポート全体を 1 つのドキュメントとしてブラウザにダウンロードするには、下図のような [すべてのページ] をクリックします。



- ビューアを閉じるには、下図のように、[閉じる] をクリックします。



## レポートの検索

ビューアコントロールパネルには、レポート内を検索するためのコントロールがいくつか用意されています。ビューアの検索コントロールを使用して、レポート内の語句や数字グループなどの文字列情報の選択や、一致した個々の文字列の検索が行えます。語句の大文字小文字の正確な一致 (大文字と小文字を区別した検索)、または検索の方向 (レポート上の検索開始地点から前方または後方のいずれか) を指定して、さらに検索方法をカスタマイズすることができます。これらのコントロールを使用して、レポートを検索します。

- 大文字小文字を区別した検索を実行するには、下図のような [大文字と小文字の区別] をクリックします。



- レポートを後方へ検索するには、下図のような [検索方向] をクリックします。



- 特定の文字列を検索するには、下図のように、検索する文字列を入力して [検索] をクリックします。



## 手順

### レポートを検索するには

1. [検索] テキストボックスに文字列を入力します。
2. 大文字小文字を区別した検索を実行するには、[大文字と小文字の区別] アイコンをクリックします。  
  
ビューアでは、[大文字と小文字の区別] が有効な場合、赤いラインが引かれた状態で表示されることに注意してください。
3. 検索を開始するには、次のいずれかをクリックします。
  - 検索方向 - 現在のページから最初のページに向けて、文字列を検索します。  
  
または
  - 検索 - レポートの現在のページから最後に向けて検索します。

ビューアがレポートを検索し、最初に一致した文字列を下線付きでページ上部に表示します。下線付きの一致した文字列が表示されない場合は、ビューアのウィンドウに表示されるまでページを下方向にスクロールします。

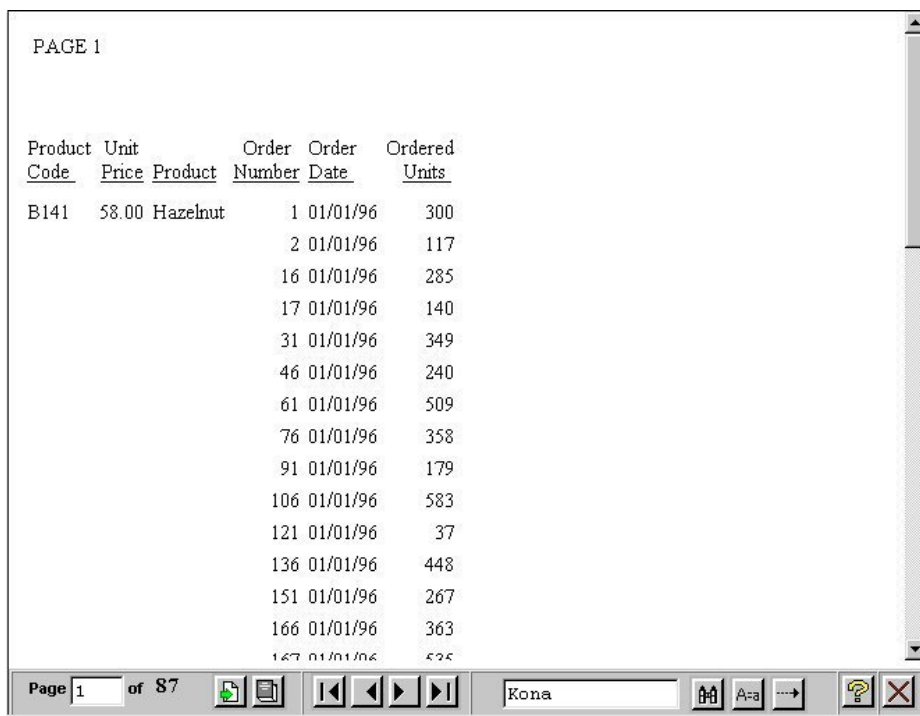
4. 一致する他の文字列を検索するには、再度 [検索] アイコンをクリックします。

## 例

### ビューアコントロールパネルを使用した検索

この例では、ビューアコントロールパネルを使用して、「Sales」という大規模なレポート内を移動し、販売しているコーヒーのタイプである文字列「Kona」を検索します。レポートを実行した後、WebFOCUS は、ビューアにレポートの最初のページを表示します。

1. Kona の売上を検索するには、下図のように、テキストボックスに「Kona」と入力し、[検索] をクリックします。



PAGE 1

Product Code	Unit Price	Product	Order Number	Order Date	Ordered Units
B141	58.00	Hazelnut	1	01/01/96	300
			2	01/01/96	117
			16	01/01/96	285
			17	01/01/96	140
			31	01/01/96	349
			46	01/01/96	240
			61	01/01/96	509
			76	01/01/96	358
			91	01/01/96	179
			106	01/01/96	583
			121	01/01/96	37
			136	01/01/96	448
			151	01/01/96	267
			166	01/01/96	363
			167	01/01/96	525

Page 1 of 87

Kona

ビューアにレポートが返され、下図のように、検索文字列と最初に一致した文字列が下線付きで表示されます。

	4260	12/01/97	230
	4261	12/01/97	175
	4274	12/01/97	254
	4275	12/01/97	194
	4276	12/01/97	380
	4289	12/01/97	307
	4290	12/01/97	427
	4291	12/01/97	306
	4304	12/01/97	221
	4305	12/01/97	396
	4306	12/01/97	149
	4307	12/01/97	376
B144	76.00	<u>Kona</u>	35
			01/01/96
			471
			50
			01/01/96
			377
			80
			01/01/96
			47
			125
			01/01/96
			510
			140
			01/01/96
			38
			155
			01/01/96
			347
			215
			02/01/96
			472

2. 再度 [検索] をクリックし、「Kona」と一致する次の文字列を検索します。

## 手順

### カスケードスタイルシートで検索結果をカスタマイズするには

WebFOCUS ビューアではレポートを検索し、最初に一致した文字列が下線付きで表示されます。色とスタイルを指定したカスケードスタイルシート (CSS) を適用し、検索結果をカスタマイズすることができます。

1. テキストエディタを使用して、新しいテキストファイルを開きます。
2. 次のカスケードスタイルシート (CSS) コードを入力します。

```
BODY {
 font : x-small Verdana, Arial, Helvetica;
}
U {
 background : Blue;
 text-decoration : none;
 color : White;
 font : bold;
}
```

上記の CSS サンプルコードにより、レポート本文の下線付きテキストが変更され、背景色が青に、テキストが太字に、文字色が白に設定されます。

3. ファイルを、カスケードスタイルシート (.css) として保存します。

**注意：**ファイル拡張子として「.css」を入力します。たとえば、「findcolor.css」と入力します。

CSS ファイルの格納場所は、WebFOCUS 環境によって異なります。

**注意：**CSS ファイルは、Web 上でアクセス可能なパスに格納します。WebFOCUS インストールでは、Web 上でアクセス可能なコンテンツは、/ibi\_apps/ibi\_html エイリアスに格納されます。

4. レポートを InfoAssist またはテキストエディタで開き、カスケードスタイルシートを適用します。

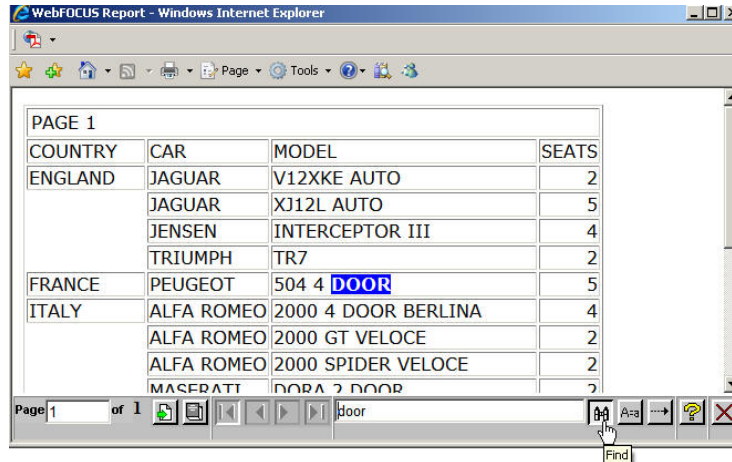
次のコードは、レポートに「findcolor.css」というカスケードスタイルシートファイルを適用します。

```
TABLE FILE CAR
PRINT CAR MODEL SEATS
BY COUNTRY
ON TABLE SET WEBVIEWER ON
ON TABLE SET STYLE *
CSSURL=/ibi_apps/ibi_html/findcolor.css, $
ENDSTYLE
END
```

**注意：**レポートを実行する WebFOCUS 環境と異なる場所に CSS ファイルが格納されている場合は、完全修飾 URL を指定します。たとえば、「CSSURL=http://hostname[:port]/ibi\_apps/ibi\_html/findcolor.css \$」と指定します。ここで、「hostname[:port]」は、CSS ファイルにアクセス可能な Web サーバまたは Application Server のポート番号を表します。

5. レポートを実行します。
6. [検索] テキストボックスに文字列を入力し、[検索] をクリックします。

WebFOCUS ビューアでは、レポートを検索し、最初に一致した文字列が下線付きで表示されます。下図の例では、findcolor.css カスケードスタイルシートファイルを使用したレポートにより、「DOOR」が検索され、青色で反転表示されています。



## Web ビューアによる印刷

ブラウザのツールバーの [印刷] ボタンを使用してレポートをローカルで印刷するには、ブラウザのキャッシュをクリアしておく必要があります。また、[印刷] ボタンを使用する前に、ウィンドウをクリックしてアクティブにしておく必要があります。

キャッシュをクリアする場合は、各ブラウザで指定された手順に従ってください。

## カスケードスタイルシート (CSS) による表示の標準化

カスケードスタイルシート (CSS) を使用すると、HTML を拡張して HTML ページのフォーマットを設定することができます。カスケードスタイルシート (CSS) は、フォーマットを設定する HTML ページに組み込んだり、複数のページで共有できるように別のファイル (拡張子は .CSS) に格納したりすることができます。カスケードスタイルシート (CSS) を別のファイルに格納した場合は、「外部カスケードスタイルシート (CSS)」と呼ばれます。

ここでは、外部カスケードスタイルシート (CSS) を使用してレポートにスクロールバーを効果的に追加する方法、および標準フォント、フォントサイズ、色、その他の表示特性を設定する方法について説明します。

## 例 レポートへのスクロールバーの追加

次の例は、レポートにスクロールバーを追加する方法を示しています。

1. 2つのレポートリクエストおよび2つのグラフィッククエストで構成された「SCROLL」というプロシジャを作成します。各リクエストは、WebFOCUS スタイルシートを使用して、スタイル設定を個別に追加します。レポートで使用する WebFOCUS スタイルシートは、外部カスケードスタイルシート (CSS) を呼び出して、標準化されたアプリケーションのスタイル設定を追加します。

左側に表示されたアルファベット文字は、そのコードの意味について説明する注記に対応しています。

### プロシジャ - SCROLL

```
SET PAGE-NUM=OFF

TABLE FILE CENTORD
HEADING
"Sales By Store"
SUM LINEPRICE AS 'Sales'
BY SNAME
a. ON TABLE SET STYLE *
b. TYPE=HEADING, CLASS=HEAD, $
```

```

b. TYPE=TITLE, CLASS=TITLE, $
TYPE=REPORT, GRID=OFF, $
TYPE=DATA, COLUMN=SNAME, COLOR=RED, STYLE=BOLD,
WHEN=LINEPRICE LT 10000000, $
TYPE=DATA, COLUMN=LINEPRICE, COLOR=RED, STYLE=BOLD,
WHEN=LINEPRICE LT 10000000, $
ENDSTYLE
ON TABLE HOLD AS CREPORT1 FORMAT HTMTABLE
END

GRAPH FILE CENTORD
SUM LINEPRICE
ACROSS SNAME
a. ON GRAPH SET STYLE *
UNITS=IN, LEFTMARGIN=0.250000,
RIGHTMARGIN=0.250000, TOPMARGIN=0.250000,
BOTTOMMARGIN=0.250000, SQUEEZE=ON, ORIENTATION=PORTRAIT, $
DEFMACRO=COND0001, MACTYPE=RULE, WHEN=N1 LE 100000.00, $
TYPE=REPORT, FONT='VERDANA', SIZE=10, BACKCOLOR=NONE,
STYLE=NORMAL, $
TYPE=DATA, ACROSSCOLUMN=N1, COLOR=RGB(144 24 24), $
TYPE=DATA, ACROSSCOLUMN=N1, COLOR=YELLOW, MACRO=COND0001, $
ENDSTYLE
ON GRAPH SET LOOKGRAPH 3D_BAR
ON GRAPH SET GRAPHEDIT OFF
ON GRAPH SET GRAPHSTYLE *
setConnectLineMarkers(true);
set01LabelDisplay(true);
set01AxisSide(0);
set01MajorGridDisplay(false);
set01MinorGridDisplay(false);
setY1LabelDisplay(true);
setY1AxisSide(0);
setY1MajorGridDisplay(false);
setY1MinorGridDisplay(false);
setPieFeelerTextDisplay(1);
setPieLabelDisplay(0);
setTextFormatPreset(getPieSliceLabel(),1);
setLegendDisplay(true);
setFootnoteString("Store Sales");
setTextJustHoriz(getFootnote(),1);
setFontStyle(getFootnote(),2);
ENDSTYLE
ON GRAPH SET BARNUMB OFF
ON GRAPH SET 3D ON
ON GRAPH SET GRID ON
ON GRAPH SET VAXIS 200
ON GRAPH SET HAXIS 300
ON GRAPH HOLD AS CGRAPH1 FORMAT HTMTABLE
END

```

```

SET LOOKGRAPH=PIE
SET GRAPHEDIT=OFF
SET GRID=ON
SET BARNUM=ON
SET 3D=ON
SET VAXIS=250
SET HAXIS=250
GRAPH FILE CENTORD
SUM LINEPRICE
BY PRODCAT
a. ON GRAPH SET GRAPHSTYLE *
 setLegendDisplay(true);
 setTitleDisplay(true);
 setTextRotation(getO1Label(),0);
 setY1LabelFormat(10);
 setO1LabelAutofit(false);
 setO1LabelStagger(false);
 setTextRotation(getO1Label(),0);
 setFontSizeVC(getO1Label(),1500);
 setY1LabelAutofit(false);
 setFontSizeVC(getY1Label(),1800);
 setTextWrap(getLegendText(0),true);
 setRect(getLegendArea(), new Rectangle(9000, -8000,8000, 15000));
 setAutofit(getLegendText(0),false);
 setFontSizeVC(getLegendText(0),850);
 setGroupLabel(0,"Sales By Product");
 setPieTilt(45);
ENDSTYLE
ON GRAPH HOLD AS CGRAPH2 FORMAT HTMTABLE
END

TABLE FILE CENTORD
HEADING
"Sales By Product"
SUM LINEPRICE AS 'Sales'
BY PRODCAT AS 'Product'
a. ON TABLE SET STYLE *
b. TYPE=HEADING, CLASS=HEAD, $
b. TYPE=TITLE, CLASS=TITLE, $
 TYPE=REPORT, GRID=OFF, $
ENDSTYLE
ON TABLE HOLD AS CREPORT2 FORMAT HTMTABLE
END
c. -HTMLFORM scrollpg

```

- a. これらのコマンドは、WebFOCUS スタイルシートの開始を指示します。
- b. WebFOCUS スタイルシートの CLASS 属性は、手順 3 で作成する「SCROLLSS.CSS」という外部カスケードスタイルシート (CSS) で定義された一連のスタイル設定の特性を指しています。
- c. このコマンドは、手順 2 で作成する「SCROLLPG」という HTML 表示ページを呼び出します。この表示ページにレポート出力が組み込まれます。

2. 「SCROLLPG.HTM」という HTML 表示ページを作成します。APP PATH または EDAPATH を使用して、WebFOCUS Reporting Server がこのページを参照できるように設定しておく必要があります。検索パスについての詳細は、23 ページの「[TIBCO WebFOCUS のアプリケーションロジック](#)」を参照してください。

Windows および UNIX では、-HTMLFORM コマンドが呼び出す HTML ファイルには、拡張子の .HTM が必要です。

表示ページは、手順 3 で作成する外部カスケードスタイルシート (CSS) にリンクします。

### HTML の表示ページ - SCROLLPG.HTM

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Cent Corp Demo KPI Template</title>
<link rel="stylesheet"
 href="/ibi_apps/ibi_html/NewCentCorp/scrollss.css"
 type="text/css">
</head>
<body>
<div>!IBI.FIL.CREPORT1;</div>
<div>!IBI.FIL.CGRAPH1;</div>
<div>!IBI.FIL.CGRAPH2;</div>
<div>!IBI.FIL.CREPORT2;</div>
<div id="toolbar">

</div>
</body>
</html>
```

3. 「SCROLLSS」というカスケードスタイルシート (CSS) を作成します。この SCROLLSS は、レポートおよびグラフを表示ページに配置し、レポート表示にスクロールバーを追加するとともに、テキストおよび背景の色を設定します。Web サーバがカスケードスタイルシート (CSS) を参照できるよう設定しておく必要があります。

左側に表示されたアルファベット文字は、そのコードの意味について説明する注記に対応しています。

### カスケードスタイルシート - **SCROLLSS.CSS**

```
a. { color: Navy;
 font-weight : bold;
}
/* the following rule controls the default styling for fonts in
the application */
body, td {
 font-family : Verdana, Geneva, Arial, Helvetica, sans-serif;
 font-size : 12px;
 font-style : normal;
 font-variant : normal;
 font-weight : normal;
/* width : 10%; */
/* the rule below allows the scroll bars to be customized */
scrollbar-base-color : Blue;
scrollbar-arrow-color : white;
}
#toolbar {
 position: relative;
 top: 500;
 left: 0;
 width: 430px;
 height: 20pt;
 padding: 6px;
 background: 0033ff;
 border : thin outset;
}
/* the following defines position and properties of top left
report */
.report1 { position: absolute;
a. top: 30;
 left: 60;
 width: 200;
 height: 200;
 border: thin outset;
b. overflow: auto;
}
/* the following defines position and properties of top right
graph */
.graph1 { position: absolute;
```

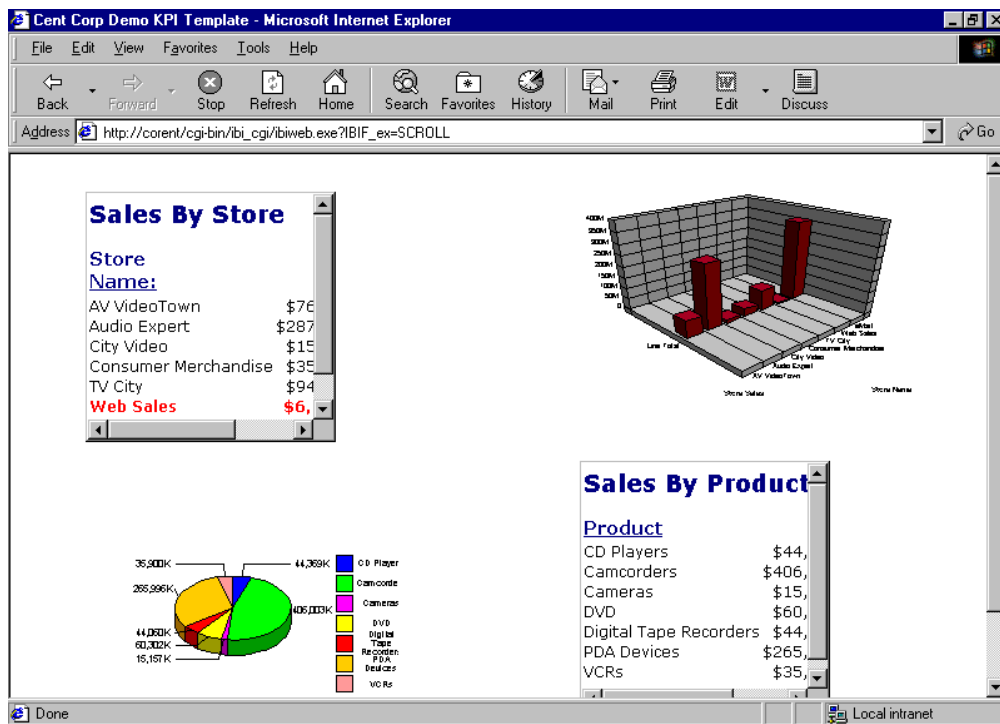
```

c. top: 0;
 left: 440;
 overflow: auto;
 }
 /* the following defines position and properties of bottom left
 graph */
 .graph2 { position: absolute;
 top: 246;
 left: 60;
 overflow: auto;
 }
 /* the following defines position and properties of bottom right
 report */
 .report2 { position: absolute;
 top: 245;
 left: 455;
 width: 200;
 height: 200;
 border: thin outset;
 overflow: auto;
 }
 /* this class sets the styling for a WebFOCUS heading */
 .HEAD {
d. font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;
 font-size: 14pt;
 font-weight: bold;
 color: #333366;
 position: relative;
 top: -13px;
 }
 /* this class sets the styling for WebFOCUS column titles */
 .TITLE {
e. font-family : Verdana, Geneva, Arial, Helvetica, sans-serif;
 font-size : 11pt;
 font-weight : bold;
 color: Navy;
 /*position: relative;
 top : 3px; */
 }
 /* this rule controls the defaults for all tables */
 TABLE {
 border-collapse: collapse;
 /* table-layout : fixed; */
 }
 /* the two rules below controls styling for various headings */
 h2 { font-size: 14pt; color: gold; }
 h3 { font-size: 14pt; color: white; }
 /* this rule sets the style for the mouse over effect on an
 anchor */
 a:hover { color: red; }

```

- a. HTML ページに最初のレポートを配置します。
- b. レポート全体がフレーム内に収まらない場合は、スクロールバーを作成します。

- c. HTML ページに最初のグラフを配置します。
- d. WebFOCUS のレポート見出しに適用するスタイル特性を設定します。
- e. WebFOCUS のフィールドタイトルに適用するスタイル特性を設定します。
4. プロシジャの SCROLL を実行する起動ページを作成します。この起動ページから、下図のようなレポートが生成されます。



## 実行済みレポートの表示

リクエストを再実行せずに、実行済みのレポートをブラウザに再表示することができます。WebFOCUS は、設定した期間の間はキャッシュ内のレポートを保持します。新しいリクエストの出力が前回のリクエストの出力と同一で、キャッシュ内にその出力が保持されている場合は、[戻る]、[更新]、[再読み込み] の各ボタンを使用して前回のレポートをブラウザに表示することができます。

キャッシュ内にレポートを保持する期間を指定するには、EXPIRE\_REPORTS パラメータを使用します。このパラメータはすべての出力に影響し、WebFOCUS のインストール時にカスタマイズすることができます。EXPIRE\_REPORTS パラメータは、次の方法で変更することができます。

- ❑ CGIVARS.WFS ファイルに EXPIRE\_REPORTS の値を設定します。このファイルのデフォルトの格納場所は次のとおりです。

**Windows** `install_drive:\ibi\WEBFOCUS82\client\conf\etc`

**UNIX** `/ibi/WEBFOCUS}82/client/conf/etc`

- ❑ SITE.WFS ファイルに EXPIRE\_REPORTS および変更した値を追加します。SITE.WFS ファイルの値は、他の .WFS ファイルの値を上書きします。

## 構文

### レポートの保持期間の設定

`EXPIRE_REPORTS = {n|300}`

#### 説明

**n**

キャッシュ内にレポートを保持する時間 (秒数) です。デフォルト値は 300 です。

レポートを常に再実行して表示するには、EXPIRE\_REPORTS を 1 に設定します。

ブラウザの [戻る]、[再読み込み]、[更新] ボタンを使用してキャッシュ内の出力レポートを表示するには、EXPIRE\_REPORTS に大きい値を設定します (例、4,000,000,000)。



# 5

## アプリケーションの制御フロー管理

---

ダイアログマネージャは、アプリケーションコンポーネントの実行を制御する WebFOCUS 言語の一部です。プロシジャの実行を動的に制御できるため、柔軟にアプリケーションを設計することができます。また、ダイアログマネージャを使用すると、プロシジャに変数を使用し、実行時にその値を指定することもできます。

ここでは、Windows、UNIX の各オペレーティングシステムでの使用方法について説明します。

### トピックス

- ダイアログマネージャの使用
  - ダイアログマネージャの処理
  - ダイアログマネージャプロシジャの作成
  - 変数によるプロシジャのカスタマイズ
  - 標準引用符付き文字列の作成
  - 変数の作成と操作
  - 関数での数値変数の使用
  - プロシジャ実行の制御
  - プロシジャ実行順序の制御
  - プロシジャによる HTML Web ページの拡張
  - オペレーティングシステムコマンドの発行
  - ダイアログマネージャによるパスワードの管理
  - アプリケーションへのメッセージの送信
  - ダイアログマネージャプロシジャのテストとデバッグ
  - ダイアログマネージャ構文リファレンス
-

## ダイアログマネージャの使用

ダイアログマネージャは、さまざまなコマンドや変数を使用してアプリケーションのフローを制御できる WebFOCUS 言語の一部です。次の説明は、ダイアログマネージャの代表的な使用方法です。

- ❑ **変数によるプロシジャのカスタマイズ** 変数を使用してプロシジャの実行を動的に変更したり、システムから情報を取得したりすることができます。詳細は、314 ページの「[変数によるプロシジャのカスタマイズ](#)」を参照してください。プロシジャに使用する変数の作成方法についての詳細は、348 ページの「[ローカル変数およびグローバル変数の作成と操作](#)」を参照してください。
- ❑ **プロシジャ実行の制御** コマンドの実行順序、外部ファイルの終了、プロシジャの終了やキャンセルを指示することで、プロシジャを制御することができます。詳細は、369 ページの「[プロシジャ実行の制御](#)」を参照してください。
- ❑ **プロシジャ実行順序の制御** リクエストの条件付き実行、プログラムループを使用した繰り返し実行、別のプロシジャの呼び出しを行えます。詳細は、372 ページの「[プロシジャ実行順序の制御](#)」を参照してください。
- ❑ **HTML Web ページの拡張** プロシジャに HTML コマンドを追加して、Web ページの機能を拡張することができます。詳細は、400 ページの「[プロシジャによる HTML Web ページの拡張](#)」を参照してください。
- ❑ **オペレーティングシステムコマンドの発行** オペレーティングシステムコマンドを発行して、リクエストを実行する環境をセットアップすることができます。オペレーティングシステムコマンドの発行についての詳細は、410 ページの「[オペレーティングシステムコマンドの発行](#)」を参照してください。
- ❑ **パスワードの管理** パスワードの割り当ておよび変更を直接実行することができます。詳細は、412 ページの「[ダイアログマネージャによるパスワードの管理](#)」を参照してください。
- ❑ **アプリケーションへのメッセージの送信** プロシジャの処理中にアプリケーションにメッセージを送信し、プロシジャの目的、結果、役立つ情報などを表示することができます。詳細は、413 ページの「[アプリケーションへのメッセージの送信](#)」を参照してください。

## 参照

## ダイアログマネージャコマンドの概要

下表は、ダイアログマネージャで使用可能なコマンドの概要を示しています。これらコマンドの詳細は、それぞれのコマンドについてのトピックを参照してください。

コマンド	説明
- *	コメントを示します。
-?	ローカル変数の値を表示します。
-DEFAULT -DEFAULTS	変数に初期値を設定します。
-DEFAULTH	隠し変数に初期値を設定します。
-DOS	DOS オペレーティングシステムのコマンドを実行します。
-EXIT	スタックコマンドを実行し、プロシジャを終了します。
-GOTO	任意のラベルに移動します。
-HTMLFORM	ブラウザにレポートを送信します。
-IF	式の評価に基づいて、プロシジャの実行フローを決定します。
-INCLUDE	別のダイアログマネージャプロシジャを呼び出します。
-label	-GOTO または -IF のターゲットです。
-PASS	パスワードを発行、管理します。
-QUIT	スタックコマンドを実行せずにプロシジャを終了します。
-QUIT FOCUS	プロシジャおよび FOCUS を終了します。
-READ	外部ファイルからデータを読み込みます。
-READFILE	最初にマスターファイルを読み込み、マスターファイルの各フィールドの ACTUAL フォーマットに基づいてダイアログマネージャ変数を作成します。

コマンド	説明
<code>-REPEAT</code>	ループを実行します。
<code>-RUN</code>	スタックコマンドを実行します。
<code>-SET</code>	変数をリテラル値または式で計算された値に設定します。
<code>-? SET parameter &amp;myvar</code>	SET パラメータの現在値を &myvar 変数に格納します。
<code>-TYPE</code>	クライアントアプリケーションにメッセージを送信します。
<code>-UNIX</code>	WebFOCUS で、UNIX オペレーティングシステムコマンドを実行します。
<code>-VMS</code>	WebFOCUS で、VMS オペレーティングシステムコマンドを実行します。
<code>-WINNT</code>	Windows オペレーティングシステムのコマンドを実行します。
<code>-WRITE</code>	外部ファイルにデータを書き込みます。

## ダイアログマネージャの処理

ダイアログマネージャプロシジャにコマンドおよび変数を含めることにより、ユーザ入力と環境条件を使用して、実行時にアプリケーションを変更することができます。ダイアログマネージャがプロシジャを使用してどのようにアプリケーションのコマンドおよび変数を処理するか理解することが重要です。

ダイアログマネージャは、次の手順でプロシジャを実行します。

1. ダイアログマネージャは、プロシジャの行を 1 行ずつ読み取り、変数が見つかったとその変数に値を代入します。

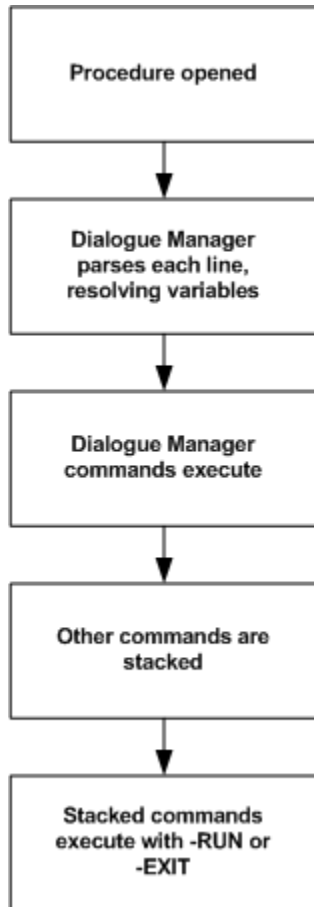
すべての変数に対して値を割り当てるか、デフォルト値を指定しておく必要があります。値が見つからない場合は、エラーメッセージが生成されます。

2. ダイアログマネージャで読み取られたダイアログマネージャコマンドはすぐに実行されます。

他の WebFOCUS コマンドは、以降の実行に備えて一時的に格納されます。これらのコマンドは、「スタックコマンド」と呼ばれます。

3. -RUN および -EXIT コマンドは、すべてのスタックコマンドを実行します。

下図は、ダイアログマネージャプロシジャの処理手順を示しています。



## 例 プロシジャとその実行プロセス

次の例は、サンプルプロシジャおよびその処理手順を示しています。

このプロシジャには、&PLANT 変数を使用されています。ここでは、プロシジャの実行時に、&PLANT 変数に対して「BOS」という値を代入する場合を想定します。左側の番号は、そのコードの説明に対応しています。

```
1. -IF &PLANT EQ 'DONE' THEN GOTO GETOUT;

2. TABLE FILE CEN THR
 HEADING
 "SALARY REPORT FOR LOCATION: &PLANT"
 PRINT SALARY POSITION
 BY LNAME
 WHERE PLANT IS '&PLANT'
 END

3. -RUN

4. -EXIT

 -GETOUT
 -TYPE NO PROCESSING DONE: EXITING
```

このプロシジャは、次のように処理されます。

1. 先頭行を実行する前に、&PLANT の値がプロシジャに渡されます。この値は、HTML インターフェースから入力するか、他のダイアログマネージャコマンドで事前に設定しておきます。ダイアログマネージャは、先頭行の &PLANT 変数に「BOS」という値を代入し、その値が DONE であるかをテストします。テストは失敗し、ダイアログマネージャは次の行へ進みます。

値が BOS ではなく、DONE であった場合は、制御は -GETOUT ラベルに渡され、「NO PROCESSING DONE: EXITING」というメッセージが生成されます。その場合、ダイアログマネージャはレポートリクエストを省略して次へ進みます。

2. この 6 行は、レポートリクエストです。ダイアログマネージャは、これらの行を読み取って変数の存在を確認し、&PLANT に BOS を代入します。各行を処理しながら、後で実行するためスタックに配置します。

ダイアログマネージャが END コマンドを処理した段階で、スタックコマンドは次のようになります。

```
TABLE FILE CEN THR
HEADING
"SALARY REPORT FOR LOCATION: BOS"
PRINT SALARY POSITION
BY LNAME
WHERE PLANT IS 'BOS'
END
```

ダイアログマネージャは次の行を処理します。

プロシジャに &ECHO 変数を含めると、この手順で示すように、変数に代入した値を表示することができます。&ECHO 変数についての詳細は、414 ページの「[ダイアログマネージャプロシジャのテストとデバッグ](#)」を参照してください。

3. -RUN は、スタックコマンドを実行します。これらのコマンドは、処理のために送信されません。
4. -EXIT は、プロシジャを終了します。

プロシジャで変数を使用する方法についての詳細は、314 ページの「[変数によるプロシジャのカスタマイズ](#)」を参照してください。

## ダイアログマネージャプロシジャの作成

プロシジャは、システムエディタで作成することができます。プロシジャを作成する場合は、次の規則に従う必要があります。

- ❑ ダイアログマネージャコマンドは、1 列目のハイフン (-) で開始します。これは、ダイアログマネージャコマンドにのみ適用される規則です。
- ❑ ダイアログマネージャコマンドと他のテキストの間には、1 つ以上のブランクを挿入する必要があります。
- ❑ ダイアログマネージャコマンドが 1 行に収まらない場合は、次の行の先頭にハイフン (-) を追加して行を継続する必要があります。継続する行は、ハイフンの直後から開始することも、ブランクを挿入してから開始することもできます。

## プロシジャへのコメントの追加

プロシジャを複数のユーザで使用する場合、そのプロシジャにコメントを追加しておく役立ちます。特に、日付、バージョン、その他の関連情報が一目で分かるように、コメントはプロシジャの見出しに追加することをお勧めします。次の 2 つのスタイルのコメントが使用できます。

- ❑ **WebFOCUS スタイルのコメント** ハイフンとアスタリスクの組み合わせ (-\*) は、コメントの開始を表します。この場合、コメントは単一行に表示されます。
- ❑ **c スタイルのコメント** このスタイルのコメントは、開始タグ記号 (/\*) と終了タグ (\*/) で囲みます。c スタイルのコメントは、任意の位置および複数行に使用することができます。

## 参照

### プロシジャへの WebFOCUS スタイルのコメントの追加

次の方法でプロシジャにコメントを追加することができます。

- ❑ ハイフンとアスタリスクの組み合わせ (-\*) の後にテキストを追加します。必要に応じて、テキストの先頭にブランクを配置することができます。

- ❑ コメントは、プロシジャの先頭または末尾、あるいはコマンド間に配置します。コメントは、コマンドと同一の行に追加することはできません。次のようにコメントを追加することはできません。

```
-RUN -*Version 3 06/10/00
```

### 例 プロシジャへの WebFOCUS スタイルのコメントの追加

次の例は、プロシジャの先頭に WebFOCUS スタイルのコメントを追加する方法を示しています。

```
-* Version 1 08/26/02 HRINFO Procedure
TABLE FILE CENTHR
.
.
.
```

### 例 プロシジャへの C スタイルのコメントの追加

次の例は、プロシジャに C スタイルのコメントを追加する方法を示しています。

```
TABLE FILE GGSales /* this is a multi-line comment
that will not interfere with processing and will be ignored
until the comment is closed with */
SUM /* Another comment */ DOLLARS
.
.
.
```

## 変数によるプロシジャのカスタマイズ

変数を使用してプロシジャをカスタマイズし、処理を制御することができます。実行時に変数の値を指定することにより、その処理を動的に制御することができます。変数の値は、次の方法で指定することができます。

- ❑ プロシジャで直接指定する。-DEFAULT、-SET、-READ コマンドを使用すると、プロシジャで値を直接指定することができます。
- ❑ プロシジャから別のプロシジャを呼び出す際に指定する。プロシジャから別のプロシジャを呼び出す EXEC コマンドに、変数名およびそれに対応する値をパラメータとして含めることができます。

- ❑ Web ページからプロシジャを呼び出す際に指定する。Web ページは、フォームまたはハイパーリンクから WebFOCUS スクリプトを呼び出します。フォームを使用する場合は、エンドユーザがフォームに値を入力します。ハイパーリンクを使用する場合は、プロシジャ名および変数の値を HTML <A HREF コードに直接追加します。各変数の値は、WebFOCUS スクリプトへの呼び出しに含まれます。
- ❑ ドリルダウンレポートで指定する。WebFOCUS スタイルシートを使用して、パラメータを渡すことができます。
- ❑ 外部ファイルから指定する。-READ コマンドを使用して、外部ファイルから値を指定することができます。

ダイアログマネージャの変数を使用して、WebFOCUS レポートスタイルシートのコマンドをカスタマイズすることはできません。

変数のタイプおよびその使用方法についての詳細は、348 ページの「[ローカル変数およびグローバル変数の作成と操作](#)」を参照してください。

## 参照

### 変数の値を指定する際の規則

変数の値には、次の規則が適用されます。

- ❑ -TYPE または -WRITE コマンドを使用する場合、変数値の最大長は 32,000 バイトです。
- ❑ 値にカンマ (,)、等号 (=)、ブランクが含まれている場合、その変数を式に使用するには、変数名を一重引用符 (') で囲む必要があります。たとえば、&CITY の値が「NY, NY」の場合、式ではこれを '&CITY' として記述する必要があります。
- ❑ ローカル変数またはグローバル変数に対して値を指定すると、-SET または -READ などのコマンドで値を変更しない限り、その値がプロシジャで常時使用されます。
- ❑ IF THEN コマンドと併せて SET コマンドで変数を指定する場合は、明示的に ELSE 句を使用する必要があります。
- ❑ EDASPROF.PRF などの WebFOCUS Reporting Server プロファイルに変数を追加しておく、と、プロシジャを実行するごとにグローバル変数に値を代入することができます。

## 例

### 変数値によるプロシジャのカスタマイズ

この例は、-DEFAULT および -SET コマンドを使用して、変数の値を指定する方法を示しています。エンドユーザは、HTML フォームのプロンプトに対して、&CODE1 に B10、&CODE2 に B20、&REGIONMGR に SMITH の値をそれぞれ指定します。

**注意：**この例では、ページ番号および罫線を表示しないフォーマット設定で出力しています。

この例の左側の番号は、そのコードの説明に対応しています。

```
1. -DEFAULT &VERB=SUM
2. -SET &CITY=IF &CODE1 GT 'B09' THEN 'STAMFORD' ELSE 'UNIONDALE';
3. -TYPE REGIONAL MANAGER FOR &CITY
 SET PAGE=OFF
4. TABLE FILE SALES
 HEADING CENTER
 "MONTHLY REPORT FOR &CITY"
 "PRODUCT CODES FROM &CODE1 TO &CODE2"
 " "
 &VERB UNIT_SOLD AND RETURNS AND COMPUTE
 RATIO/D5.1 = 100 * (RETURNS/UNIT_SOLD);
 BY PROD_CODE
 IF PROD_CODE IS-FROM &CODE1 TO &CODE2
 FOOTING CENTER
5. "REGION MANAGER: ®IONMGR"
 "CALCULATED AS OF &DATE"
 ON TABLE SET STYLE *
 TYPE=REPORT, GRID=OFF,$
 END
6. -RUN
```

プロシジャは、次の手順で実行されます。

1. -DEFAULT コマンドが、&VERB の値を SUM に設定します。
2. -SET コマンドが、エンドユーザがフォームで &CODE1 に対して入力した値に基づいて、&CITY に対する値を指定します。エンドユーザが &CODE1 の値として B10 を入力したため、&CITY は STAMFORD になります。
3. エンドユーザがこのレポートを実行すると、WebFOCUS は &CITY の値とともに次のようなメッセージを出力します。

**REGIONAL MANAGER FOR STAMFORD**

-TYPE コマンドによる出力は、HTML ページには表示されませんが、HTML ソースファイルに書き込まれます。メッセージを表示するには、ドキュメントのソースファイルを開く必要があります。

4. このスタックは、次のような行で構成されます。

```

TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR STAMFORD"
"PRODUCT CODES FROM B10 TO B20"
" "
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.1 = 100 * (RETURNS/UNIT_SOLD);
BY PROD_CODE
IF PROD_CODE IS-FROM B10 TO B20
FOOTING CENTER
"REGION MANAGER: SMITH"
"CALCULATED AS OF 07/08/93"
END

```

5. エンドユーザは、フォームから &REGIONMGR の値を指定しています。WebFOCUS が実行時に現在の日付を挿入します。
6. -RUN コマンドは、すべてのスタックコマンドを実行します。レポートリクエストからの出力は次のようになります。

```

MONTHLY REPORT FOR STAMFORD
PRODUCT CODES FROM B10 TO B20

```

PROD_CODE	UNIT_SOLD	RETURNS	RATIO
B10	103	13	12.6
B12	69	4	5.8
B17	49	4	8.2
B20	40	1	2.5

```

REGION MANAGER: SMITH
CALCULATED AS OF 07/08/93

```

## デフォルト変数値の指定

-DEFAULT コマンドは、ローカル変数またはグローバル変数のデフォルト値を設定します。ストアドプロシジャ内の変数にデフォルト値を指定することにより、そのプロシジャを常時正しく実行させることができます。

-DEFAULTS は -DEFAULT と同義です。

1 つの変数に対して複数の -DEFAULT コマンドを発行することができます。グローバル変数の場合は、これらの -DEFAULT コマンドをそれぞれ異なるプロシジャで発行することができます。この変数の値を他の方法で設定しない限り、最後に発行した -DEFAULT コマンドが引き続き有効になります。

ただし、-SET コマンドの発行、ユーザが入力した値の取得、ファイルからの値の読み取りなどの方法で変数の値を設定すると、同一変数に対してそれ以降に発行した -DEFAULT コマンドは無視されます。

-DEFAULTH コマンドを使用して変数を初期化することにより、その変数を WebFOCUS のパラメータプロンプトで使用されることを回避することができます。-DEFAULTH で初期化された変数は、パラメータプロンプトに使用する XML 記述情報には返されません。これらの変数はパラメータプロンプト機能に表示されないため、ユーザには非表示になります。

## 構文 デフォルト値の指定

```
-DEFAULT[S] [&[&]name=value , [&[&]name=value] [;]
```

説明

**&name**

変数名です。

**value**

変数に割り当てるデフォルト値です。

;

オプションの句読点記号です。

注意：-DEFAULTS は -DEFAULT と同義です。

## 構文 隠し変数デフォルト値の指定

```
-DEFAULTH [&[&]name=value , [&[&]name=value]
```

説明

**&name**

変数名です。この変数は、パラメータプロンプトに使用する XML 記述情報には返されないため、ユーザには非表示になります。

**value**

変数に割り当てるデフォルト値です。

## 例 デフォルト値の指定

次の例は、-DEFAULT を使用して &PLANT のデフォルト値を設定する方法を示しています。

```
-DEFAULT &PLANT=BOS
TABLE FILE CENTHR
.
.
.
```

## 式による変数値の指定

変数に値を割り当てる場合、`-SET` コマンドを使用して式の計算値を割り当てたり、リテラル値を割り当てたりすることができます。また、`IN FILE` 句を使用して文字値がファイルに存在するかどうかをテストし、その結果を変数に割り当てることもできます。ファイルにテスト値が存在すれば変数の値は 1 になり、存在しなければ 0 (ゼロ) になります。

変数値を数値に設定する場合、使用可能な文字は、数字、先頭のマイナス記号 (-)、および小数部を表すためのピリオド (.) のみです。EDIT オプションや CDN 値に関係なく、ダイアログマネージャで使用する数値には、これらの文字のみがサポートされます。

## 構文

### 式による値の割り当て

```
-SET &[&]name= {expression|value};
```

```
-SET &[&]var3= &var1 IN FILE filename1 [OR &var2 IN FILE filename2 ...];
```

#### 説明

##### &name

変数名です。

##### expression

有効な式です。式は複数の行に記述することができます。その場合は、コマンドの末尾にセミコロン (;) を追加する必要があります。

##### value

変数に割り当てるリテラル値、あるいは数式または論理式です。リテラル値にカンマ (,) またはブランクが含まれている場合は、値を一重引用符 (') で囲む必要があります。値自体に一重引用符が含まれている場合は、一重引用符を 2 つ使用することで一重引用符を表します。

##### &[&]var3

変数です。等号の右側に表示される計算式の結果が True の場合、値は 1、結果が False の場合、値は 0 (ゼロ) です。

##### &var1

filename1 で検索する値を含む変数です。

## &var2

filename2 で検索する値を含む変数です。

## 参照

### IN FILE 使用上の注意

- ❑ IN FILE 句の結果は文字値 (1 または 0) であり、-SET コマンド内で、AND 演算子および OR 演算子で結合した論理式で使用できます。この値は、-SET コマンド内での結合など、文字演算子内の引数としては使用できません。
- ❑ IN FILE は、ファイル内の値と検索文字列がファイルの左端からバイトを数えて完全に一致している場合のみ、値 1 を返します。
- ❑ ファイルの順序は任意で、重複する値を使用することもできます。一致が見つかるか、ファイルの終わりまで到達すると、検索は停止します。ファイルが割り当てられているが存在しない場合は、値 0 (ゼロ) が返されます。ファイルが割り当てられていない場合、ソース表示に FOC351 メッセージが表示されます。
- ❑ IN FILE 句内の変数の長さにより、比較に使用するファイルの、各レコード開始位置からのバイト数が決まります。バイト数が完全一致した場合のみ、1 が返されます。変数の末尾ブランク数が、比較ファイルの末尾ブランク数と同一でない場合、一致とは見なされません。次の例では、「ABC 」の値のみが一致と見なされます (「ABC」の後ろに 3 つのブランクが挿入されているもののみ)。

```
-SET &VAR1 = 'ABC ';
-SET &VAR2 = &VAR1 IN FILE FILE1;
```

## 構文

### ダイアログマネージャの演算精度の設定

DMPRECISION 設定を使用すると、ダイアログマネージャの -SET コマンドで正確な数値変数の値を計算することができます。

倍精度演算を使用する場合でも、この演算精度を設定していないと、数値計算の結果は整数値として返されます。この演算精度を設定をせずに小数点以下の桁数を持つ数値を返すには、FTOA 関数への入力として演算を入力する必要があります。FTOA 関数には、戻り値の小数点以下の桁数を設定することができます。

SET DMPRECISION コマンドを使用すると、演算結果にデフォルトの小数部切り捨てを適用するか、端数処理の桁数として 9 桁までを指定することができます。

```
SET DMPRECISION = {OFF|n}
```

## 説明

**OFF**

小数点以下の切り捨て (端数処理しない) を指定します。デフォルト値は OFF です。

**n**

端数処理の桁数を指定する 0 (ゼロ) から 9 の正の数です。n に 0 (ゼロ) を指定すると、端数処理された整数値が表示されます。

## 注意

- ❑ SET DMPRECISION を使用する場合は、SET DMPRECISION の後に -RUN を追加し、数値の -SET コマンドの前で必ず数値精度を設定するようにします。
- ❑ 実際の倍精度への変換はオペレーティングシステムの規則に従うため、プラットフォームにより値が異なる場合があります。

## 例

**SET DMPRECISION の使用**

下表は、20 を 3 で除算した値をさまざまな DMPRECISION (DMP) 設定で表した結果を示しています。

SET DMPRECISION =	結果
OFF	6
0	7
1	6.7
2	6.67
9	6.666666667

## 例

**式での変数値の設定**

次の例は、-SET コマンドを使用して、IF 論理式に基づいて変数の &STORECODE に対して「14Z」または「14B」という値を割り当てる方法を示しています。&CODE の値は、エンドユーザが指定します。

```
-SET &STORECODE = IF &CODE GT C2 THEN '14Z' ELSE '14B';
TABLE FILE SALES
SUM UNIT_SOLD AND RETURNS
BY PROD_CODE
IF PROD_CODE GE &CODE
BY STORE_CODE
IF STORE_CODE IS &STORECODE
END
```

## 例 ファイル変数値存在確認テスト

次のプロシジャは、CAR ファイルの国名が格納された「COUNTRY1」という文字 HOLD ファイルを作成します。次に、変数「&C」を FRANCE に設定します。FRANCE が HOLD ファイル内に存在する場合、IN FILE 句は &IN1 に対して値 1 を返し、存在しない場合は、値 0 (ゼロ) を返します。

```
TABLE FILE CAR
PRINT COUNTRY
ON TABLE HOLD AS COUNTRY1 FORMAT ALPHA
END
-RUN
-SET &C = 'FRANCE';
-SET &IN1 = &C IN FILE COUNTRY1;
-TYPE THE VALUE IS &IN1
```

出力結果から、FRANCE は COUNTRY1 ファイルに存在することが分かります。

```
THE VALUE IS 1
```

## 別プロシジャによる変数値の指定

プロシジャで別のプロシジャを呼び出す際に使用する EXEC コマンドで、変数の値を指定することができます。この方法を使用してアプリケーションフローを制御する方法についての詳細は、399 ページの「[EXEC による別プロシジャの呼び出し](#)」を参照してください。

名前と値の組み合わせを使用して値を割り当てます。複数の組み合わせを使用する場合は、それぞれをカンマ (,) で区切ります。すべての組み合わせが 1 行に収まらない場合は、行の末尾にカンマ (,) を追加して、次の行に残りの組み合わせを記述します。組み合わせの順序は、プロシジャに記述された順序と一致させる必要はありません。

EXEC を使用して、プロシジャで使用するすべての変数ではなく、一部の変数に限定して値を指定することができます。この場合、EXEC コマンドで指定しない値は、別のダイアログマネージャコマンドを使用して指定する必要があります。

順序指定の変数 (番号付けされた変数) の場合、EXEC コマンドに変数名を指定する必要はありません。WebFOCUS が、プロシジャ内での順序に従ってこれらの変数に EXEC の値を照合します。そのため、各変数の値は正しい順序で入力する必要があります。

## 参照

### EXEC で名前付き/順序指定変数を使用する際の規則

EXEC コマンドでは、名前付きの変数と順序指定の変数を混在させることができます。その場合は、次の規則に従う必要があります。

- ❑ 名前付き変数の場合、名前が値に関連付けられている必要があります。
- ❑ 順序指定の変数値は、プロシジャ内でこれらの変数に付けられた番号の順序で指定する必要があります。

次のコマンドは、名前付き変数および順序指定の変数に有効です。

```
EX HRINFO SALARY=72000, 5, BOS, STATUS=EMPLOYED
```

## 構文

### 別プロシジャによる変数値の指定

```
EX[EC] procedure name=value
```

説明

**procedure**

名前と値の組み合わせが格納されたプロシジャの名前です。

**name**

変数名です。

**value**

変数に渡す値です。

**注意:** BI Portal で EXEC を使用する場合、EX と EXEC の処理方法が異なる点に注意する必要があります。プロシジャでコーディングされた EX ステートメントは、WebFOCUS Client で処理されます。WebFOCUS Client は、BI Portal リポジトリ内でこのプロシジャを検索します。EXEC ステートメントで参照されるプロシジャは、WebFOCUS Client では処理されず、WebFOCUS Reporting Server に渡されて処理されます。これらのプロシジャは、BI Portal リポジトリ内では検索されません。

## 例

### 別プロシジャによる値の指定

ここでは、「SLRPT」というプロシジャについて考察します。

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR &CITY"
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY PROD_CODE
IF PROD_CODE IS-FROM &CODE1 TO &CODE2
BY CITY
IF CITY EQ &CITY
END
```

次のように EX コマンドを使用して、このプロシジャを別のプロシジャから呼び出し、変数の値をパラメータとして指定することができます。

```
EX SLRPT CITY=STAMFORD, CODE1=B10, CODE2=B20
```

## 例 順序指定変数の使用

ここでは、次の例について考察します。

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR &1"
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY PROD_CODE
IF PROD_CODE IS-FROM &2 TO &3
BY CITY
IF CITY EQ &1
END
```

次のように、このプロシジャを呼び出す EX コマンドを指定します。

```
EX SLRPT STAMFORD, B10, B20
```

このコマンドは、順序指定の最初の変数に STAMFORD、2 番目の変数に B10、3 番目の変数に B20 をそれぞれ代入します。

## 外部ファイルを使用した変数値の読み取りと書き込み

-READ および -WRITE コマンドを使用して、外部ファイルからの変数値の読み取りおよび外部ファイルへの変数値の書き込みを行うことができます。

- -READ コマンドを使用して、値を指定することができます。たとえば、外部ファイルにレポート期間の開始日と終了日を格納しておくことができます。ダイアログマネージャは外部ファイルからこれらの値を読み取り、WHERE コマンドでレポートリクエストの選択データ範囲を制限する変数に使用します。

- ❑ -WRITE コマンドを使用して、外部ファイルに変数値を保存することができます。たとえば、その日の総売上高を外部ファイルに保存しておき、これを次の日の総売上と比較できるようにします。

外部ファイルとして、固定フォーマットファイル (データを固定フィールドに格納) または可変長フォーマットファイル (データをカンマ区切り形式で格納) のどちらでも選択することができます。

-READ または -WRITE コマンドを使用する場合、外部ファイルをプロシジャの FILEDEF コマンドで指定する必要があります。-RUN コマンドで、FILEDEF コマンドと -READ または -WRITE コマンドを分離する必要があります。

また、-READFILE コマンドを使用してファイルを読み取ることもできます。-READFILE コマンドは、最初にマスターファイルを読み込み、マスターファイルの各フィールドの ACTUAL フォーマットに基づいてダイアログマネージャ変数を作成します。さらに、ファイルを読み込み、必要に応じて、作成された変数に戻す前にフィールドを数値から文字列に変換します。USAGE フォーマットの表示オプションは変数に継承されません。変数名は、フィールド名の前にアンパサンド (&) を追加したのになります。

## 構文

### 外部ファイルからの変数値の取得

```
-READ filename[,] [NOCLOSE] &name[.format.][,][&name][.format.]
```

#### 説明

**filename[, ]**

外部ファイル名です。これは、使用するオペレーティングシステムで定義されたものでなければなりません。filename の後のブランクは固定フォーマットファイル、カンマ (,) は可変長フォーマットファイルを表します。

- ❑ UNIX および Windows プラットフォームでは、外部ファイル用の FILEDEF が必要です。

#### NOCLOSE

-READ 処理が完了するまで、外部ファイルを開いた状態にします。NOCLOSE オプションで開いた状態にしておいたファイルは、-CLOSE filename コマンドを使用するか、-WRITE コマンドを直後に使用して閉じることができます。

**&name[, ]**

変数名です。可変長フォーマットファイルでは、複数の変数名をカンマ (,) で区切ることができます。変数のリストが 1 行に収まらない場合は、先頭行の末尾にカンマ (,) を追加し、次の行の先頭にハイフン (-)、ブランクを順に追加してリストを継続します。カンマ区切りファイルなどの可変長フォーマットファイルでは、次の行の先頭にハイフン、ブランク、カンマ (-,) またはハイフン、カンマ (-,) を入れて開始します。

### 可変長フォーマット

```
-READ EXTFILE, &CITY, &CODE1, - &CODE2
```

### 固定フォーマット

```
-READ EXTFILE &CITY.A8. &CODE1.A3., - ,&CODE2.A3.
```

#### **.format.**

変数のフォーマットです。自由フォーマットファイルでは、この値の指定はオプションです。固定ファイルフォーマットでは、format に長さを指定するか、タイプと長さを指定します。タイプは、デフォルトの A (文字) または I (数値) のいずれかです。format の値は、ピリオド (.) で区切る必要があります。カンマ区切りファイルでは format は無視されます。

**注意：**.format. を使用する代わりに、-SET を使用し、対応するブランク数を一重引用符 (') で囲むことにより変数の長さを指定することができます。以下はその例です。

```
-SET &CITY=' ' ;
-SET &CODE1=' ' ;
-SET &CODE2=' ' ;
```

## 構文 外部ファイルへの変数値の書き込み

```
-WRITE filename [NOCLOSE] text
```

### 説明

#### **filename**

書き込み先または読み取り元のファイルの名前です。UNIX および Windows プラットフォームでは、filename に呼び出し先プロシジャの絶対パスとファイル名を指定する必要があります。

#### **NOCLOSE**

-WRITE 処理が完了するまで外部ファイルを開いた状態にしておきます。NOCLOSE オプションを使用して開いた状態にしておいたファイルは、-CLOSE filename コマンドを使用して閉じることができます。

#### **text**

変数とテキストの組み合わせです。

## 例 外部ファイルからの値の読み取り

EXTFILE が、次のデータを含んだ固定フォーマットファイルであることを想定します。

```
STAMFORDB10B20
```

ファイルの末尾を検出するため、次のコードがシステム変数の &IORETURN をテストします。すべてのレコードを読み取ると、0 (ゼロ) の値は検出されなくなります。

```
-READ EXTFILE &CITY.A8. &CODE1.A3. &CODE2.A3.
-IF &IORETURN NE 0 GOTO RESUME;
 TABLE FILE SALES
 SUM UNIT_SOLD
 BY CITY
 IF CITY IS &CITY
 BY PROD_CODE
 IF PROD_CODE IS-FROM &CODE1 TO &CODE2
 END
-RESUME
.
.
.
```

## 例 外部ファイルによる読み取りと書き込み

次の例は、テキストファイルからの読み取り、テキストファイルへの書き込み、およびプロシージャで FILEDEF コマンドを使用する方法を示しています。左側の数字は、それぞれの説明に対応しています。この例では、ユーザが HTML フォームで &CITY の値を指定します。

```
1. -TOP
2. FILEDEF PASS DISK D:PASS.DAT
 -RUN
3. -WRITE PASS &CITY
 TABLE FILE SALES
 HEADING CENTER
 "LOWEST MONTHLY SALES FOR &CITY"
 " "
 PRINT DATE PROD_CODE
 BY LOWEST 1 UNIT_SOLD
 BY STORE_CODE
 BY CITY
 WHERE CITY EQ '&CITY'
 FOOTING CENTER
 "CALCULATED AS OF &DATE"
 ON TABLE SAVE AS INFO
 END
4. -RUN
```

```
5. FILEDEF LOG DISK D:LOG.DAT
 -RUN
 MODIFY FILE SALES
 COMPUTE
 TODAY/I6=&YMD;
 CITY='&CITY';
 FIXFORM X5 STORE_CODE/A3 X15 DATE/A4 PROD_CODE/A3
 MATCH STORE_CODE DATE PROD_CODE
 ON MATCH TYPE ON LOG
 "<STORE_CODE><DATE><PROD_CODE><TODAY>"
 ON MATCH DELETE
 ON NOMATCH REJECT
 DATA ON INFO
 END
6. -RUN
 EX SLRPT3
7. -RUN
8. -GOTO TOP
9. -QUIT
```

呼び出し側のプロシジャから呼び出されるプロシジャ SLRPT3 には、次のコードが含まれています。

```
10. -READ PASS &CITY.A8.
 TABLE FILE SALES
 HEADING CENTER
 "MONTHLY REPORT FOR &CITY"
 "LOWEST SALES DELETED"
 " "
 PRINT PROD_CODE UNIT_SOLD RETURNS DAMAGED
 BY STORE_CODE
 BY CITY
 WHERE CITY EQ '&CITY'
 FOOTING CENTER
 "CALCULATED AS OF &DATE"
 END
11. -RUN
```

このプロシジャは、次のように処理されます。

1. -TOP は、プロシジャの開始を表します。
2. FILEDEF コマンドは、「PASS」という名前のファイルを定義します。
3. -WRITE コマンドは、&CITY の値を「PASS」という名前のテキストファイルに書き込みます。ここでは、書き込む値を STAMFORD と想定します。
4. -RUN コマンドは、スタックされたレポートリクエストを実行します。この例では、SAVE コマンドで「INFO」というテキストファイルが作成されます。これは、レポートリクエストの結果が格納されたシーケンシャルファイルです。
5. FILEDEF コマンドは、その次の MODIFY リクエスト用のログファイルを定義します。

6. -RUN ステートメントは、スタックされた MODIFY コマンドを実行します。データは、先のレポートリクエストで作成された INFO ファイルから直接取得され、FIXFORM ステートメントを使用して入力されます。UNIT\_SOLD に最下位の値を含む製品がデータソースから削除され、ログファイルに記録されます。
7. 次の -RUN コマンドは、「SLRPT3」というプロシジャを実行します。
8. -GOTO TOP ステートメントは、プロシジャの開始点に制御を渡します。
9. -QUIT は、処理を終了します。
10. -READ コマンドは、&CITY の値を「PASS」というテキストファイルに書き込みます。ここでは、「STAMFORD」という値を渡します。
11. -RUN コマンドは、レポートリクエストを実行し、呼び出しプロシジャに制御を返します。

## 構文

### マスターファイルフィールドのダイアログマネージャ変数への読み込み

**-READFILE** [app/]mastername

説明

**app**

ファイルが格納されているアプリケーションディレクトリです。

**mastername**

読み取るマスターファイルの名前です。

## 参照

### -READFILE 使用上の注意

- ❑ -RUN コマンドではファイルは閉じません。ファイルを閉じるには、-CLOSE コマンドを発行する必要があります。閉じる前にファイルの削除、変更、格納場所の変更をしないよう注意します。
- ❑ 同一名のフィールドが複数存在する場合、マスターファイルの最後のフィールド値を含む変数が 1 つだけ作成されます。
- ❑ マスターファイルに DBA 制限が格納されている場合、-READFILE は機能しません。次のメッセージが表示されます。

(FOC339) -READ で指定されたファイルが割り当てられていません : -READFILE filename

- ❑ -READFILE は、テキストフィールドではサポートされていません。次のメッセージが表示されます。

(FOC702) このオプションでは TEXT フィールドはサポートされていません : fieldname

- ❑ -READFILE で、XFOCUS データソースを読み込むことはできません。読み込みファイルには、マスターファイルが関連付けられている必要があります。

## 例

### -READFILE を使用したフィールドのダイアログマネージャ変数への読み込み

次のリクエストにより、バイナリ HOLD ファイルが作成されます。さらに、-READFILE を使用して HOLD ファイルから最初のレコードが読み込まれ、ダイアログマネージャ変数として取り込まれる変数が表示されます。変数名は、フィールド名の前にアンパサンド (&) を追加したものに なります。

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME DEPARTMENT CURR_SAL
BY EMP_ID
ON TABLE HOLD AS READF1 FORMAT BINARY
END
-RUN
-READFILE READF1
-TYPE LAST_NAME IS &LAST_NAME
-TYPE FIRST_NAME IS &FIRST_NAME
-TYPE DEPARTMENT IS &DEPARTMENT
-TYPE CURR_SAL IS &CURR_SAL
-TYPE EMP_ID IS &EMP_ID
```

出力結果は次のとおりです。

```
> NUMBER OF RECORDS IN TABLE= 12 LINES= 12

HOLDING BINARY FILE...
LAST_NAME IS STEVENS
FIRST_NAME IS ALFRED
DEPARTMENT IS PRODUCTION
CURR_SAL IS 11000.00
EMP_ID IS 071382660
```

### 外部ファイルを閉じる

-CLOSE コマンドは、-READ または -WRITE コマンドで NOCLOSE オプションを使用して開いた外部ファイルを閉じます。NOCLOSE オプションは、WRITE 処理が完了するまでファイルを開いた状態にします。

## 構文

### 外部ファイルを閉じる

```
-CLOSE filename
```

説明

`filename`

オペレーティングシステムが認識できる物理ファイルに関連付けられた識別名です。

## 変数値としての日付の指定

変数を使用して、ダイアログマネージャに日付を渡すことができます。ダイアログマネージャに渡す日付の最大長は、ブランクを含めて 20 バイトです。ダイアログマネージャの変数は、完全な日付フォーマットのみを受容します。つまり、MDY または MDYY を任意の順序で使用可能です。

例

### 2つの日付の差の設定

以下の例では、変数 &DELAY を &LAER から &NOW までの日数に設定して、ブラウザに結果を返しています。

```
-SET &NOW = 'JUN 30 2002';
-SET &LATER = '2002 25 AUG';
-SET &DELAY = &LATER - &NOW;
-TYPE &DELAY
```

### 4桁の西暦日付を含む変数の使用

4 桁の西暦日付に 4 桁の年を使用していない場合、世紀を識別する変数を使用することができます。

## 変数の演算

-SET を使用して、論理式や算術式、またはそれらの組み合わせの結果に基づいて、代替の変数に対する値を定義することができます。

構文

### 変数の演算

```
-SET &name = expression;
```

説明

`&name`

式によって値を割り当てる、ユーザ入力変数です。

`expression`

式です。式はこの式は、『TIBCO WebFOCUS Language リファレンス』に記載されている規則に従いますが、ここで定義されている制限事項が適用されます。-SET コマンドの終了には、式の後にセミコロン (;) が必要です。

## 例 変数値の変更

次の例は、-SET を使用して、テストに基づいて変数値を変更する方法を示しています。

```
-SET &OPERATOR = IF &CHOICE EQ A THEN 'GT' ELSE 'LT';
TABLE FILE SALES
PRINT PROD_CODE UNIT_SOLD RETAIL_PRICE
BY STORE_CODE BY DATE
IF RETAIL_PRICE &OPERATOR 2.00
END
```

ユーザが「A」という文字を入力する場合、-SET はその文字の値 GT を \$OPERATOR に割り当てます。さらに、値 GT はプロシジャ内で &OPERATOR 変数に渡され、実行時に FOCUS コマンドは次のように展開されます。

```
IF RETAIL_PRICE GT 2.00
```

## DECODE 関数による変数値の変更

DECODE 関数を使用して、変数を関連付けられた値に変更することができます。

## 例 変数値の変更

この例では、変数は映画のランク付け (Rating) を表します。

```
&SELECT. ENTER A,B,C,D,E.
-SET &RATING = DECODE &SELECT (A G B PG13 C R D NR E PG);
TABLE FILE MOVIES
PRINT TITLE BY RATING
IF RATING EQ &RATING
END
```

## 変数値の検証

入力した値が有効であること、およびその値をプロシジャで正しく使用できることを確認するために、値の存在、タイプ、長さをテストすることができます。また、値が受容可能なフォーマットであること、および値が特定の値範囲内に収まることを確認することもできます。たとえば、文字データが入力された変数に対して数値演算を実行しないよう事前にテストします。

### ユーザ入力値の存在、値、長さ、タイプの確認

値を検証するには、変数値に接尾語を追加します。

- ❑ .EXIST 接尾語は、値の存在をテストします。
- ❑ .EVAL 接尾語は、変数の値を、その変数で指定された式の値やコマンドに即時に置換します。

- ❑ **.LENGTH** 接尾語は、値の長さをテストします。
- ❑ **.TYPE** 接尾語は、値のタイプをテストします。
- ❑ **.DATE\_LOCALE** 接尾語は、**DATE\_SEPARATOR**、**DATE\_ORDER**、**TIME\_SEPARATOR** のパラメータ値を日付システム変数に適用します。

## 参照

### .EVAL 使用上の注意

ダイアログマネージャ変数は、実行時に代入される値のプレースホルダです。一部の状況では、**.EVAL** 演算子を使用して変数の値を強制的に評価しない限り、変数を含むコマンドが検出された時点で変数の値が解決されていない場合があります。IF ステートメントに **.EVAL** 演算子が必要になる例として、変数がラベルに埋め込まれている場合があります (例、**GOTO AB&label.EVAL**)。 **.EVAL** 演算子は、変数が一重引用符 (') で囲まれている場合にも常に必要になります。

## 構文

### 変数値の存在、値、長さ、タイプの確認

```
-IF &name{suffix} expression GOTO label...;
```

説明

**&name**

ユーザが入力する変数です。

**suffix**

次のいずれかです。

- ❑ **.EXIST** 値の有無をテストします。値が存在しない場合、式に 0 (ゼロ) が渡されます。値が存在する場合、0 (ゼロ) 以外の値が渡されます。
- ❑ **.LENGTH** 値の長さをテストします。値が存在しない場合、式に 0 (ゼロ) が渡されます。値が存在する場合、値のバイト数が渡されます。
- ❑ **.TYPE** 値のタイプをテストします。値が  $10^9-1$  以下と認識された場合は、式に「N」が渡され、4 バイトの浮動小数フォーマットとして格納されます。ダイアログマネージャでは、数値フィールドの数値演算は、式の結果の算出後の切り捨てにより整数になります。値が数値として解釈されない場合は、文字を表す「A」が式に渡されます。

- ❑ **.EVAL** 変数の値を、その変数で指定された式の値やコマンドに即時に置換します。**.EVAL** 演算子を使用すると、変数の値を即座に評価できるため、プロシジャの動的な変更が可能になります。**.EVAL** 演算子は、実行時にコードを変更する場合に特に役立ちます。コマンドプロシジャを実行すると、他の処理を実行する前に、指定された値で式が置換されます。**.EVAL** 演算子を使用しない場合、コマンドの代わりに変数を使用することはできません。

#### expression

指定した接頭語と &name を変数として使用する有効な式です。

#### GOTO label

分岐先のラベルを指定します。

### 例

#### 変数の存在のテスト

次の例では、値を指定しなかった場合、&OPTION.EXIST が 0 (ゼロ) になり、制御は -CANTRUN ラベルに渡されます。-HTMLFORM が、プロシジャを実行できないというメッセージを表示します。値を指定した場合、制御は -HRINFO ラベルに渡されます。

```
-IF &OPTION.EXIST GOTO HRINFO ELSE GOTO CANTRUN;
.
.
.
-HRINFO
 TABLE FILE CENTHR
 .
 .
 .
 END
-EXIT
-CANTRUN
-HTMLFORM BEGIN
<HTML>
<BODY>
TOTAL REPORT CAN'T BE RUN WITHOUT AN OPTION.
</BODY>
</HTML>
-HTMLFORM END
-EXIT
```

### 例

#### HTML フォーマットメッセージによる変数値の長さテスト

&OPTION に対して指定した値の長さが 1 バイトより大きい場合、制御は -FORMAT ラベルに渡されます。次に、制御は -HTMLFORM コマンドにまで進み、1 バイト文字のみを指定できることをユーザに通知するメッセージを表示します。

```

-IF &OPTION.LENGTH GT 1 GOTO FORMAT ELSE
-GOTO PRODSALES;
.
.
.
-PRODSALES
TABLE FILE SALES
.
.
.
END
-EXIT
-FORMAT
-HTMLFORM BEGIN
<HTML>
<BODY>
ONLY A SINGLE CHARACTER IS ALLOWED.
</BODY>
</HTML>
-HTMLFORM END

```

## 例 変数値の長さテスト

&OPTION の長さが 1 バイトより長い場合、制御は FORMAT ラベルに渡されます。次に、-HTMLFORM コマンドが、1 バイト文字のみを入力できることをユーザに通知するメッセージを表示します。

```

-TOP
-IF &OPTION.LENGTH GT 1 GOTO FORMAT ELSE
-GOTO PRODSALES;
.
.
.
-PRODSALES
TABLE FILE SALES
.
.
.
END
-EXIT
-FORMAT
-HTMLFORM BEGIN
<HTML>
<BODY>
PLEASE ENTER ONLY ONE CHARACTER.
</BODY>
</HTML>
-HTMLFORM
END

```

## 例 変数値のタイプのテスト

&OPTION が文字でない場合、制御は -NOALPHA ラベルに渡されます。-HTMLFORM は HTML ページを表示し、文字のみを入力できることをユーザに通知します。

```
-IF &OPTION.TYPE NE A GOTO NOALPHA ELSE
- GOTO HRINFO;
.
.
.
-HRINFO
TABLE FILE CENTHR
.
.
.
END
-EXIT
-NOALPHA
-HTMLFORM BEGIN
<HTML>
<BODY>
ENTER A LETTER ONLY.
</BODY>
</HTML>
-HTMLFORM
```

## 例 WebFOCUS での変数解析用の .EVAL の使用

次のようなコードを想定します。

```
-SET &A='-TYPE';
&A HELLO
```

この場合、WebFOCUS が &A の値を認識できないことを示すエラーメッセージが生成されます。

```
{FOC1517} UNRECOGNIZED COMMAND -type HELLO
```

&A 変数に .EVAL 演算子を追加することで、WebFOCUS で変数を正しく解析できるようになります。次のように .EVAL 演算子を追加します。

```
-SET &A='-TYPE';
&A.EVAL HELLO
```

次の出力が生成されます。

```
HELLO
```

## 例 変数の即時評価

次の例は、レコード選択の式で .EVAL 演算子を使用する方法を示しています。左側の番号は、このプロシジャの後に記載したそれぞれの説明に対応しています。

```

1. -SET &R='IF SALARY GT 60000';
2. -IF &Y EQ 'YES' THEN GOTO START;
3. -SET &R = '-*';
 -START
4. TABLE FILE CENTHR
 PRINT SALARY
 BY PLANT BY LNAME
5. &R.EVAL
 END

```

プロシジャは、次の手順で実行されます。

1. プロシジャが &R の値を「IF SALARY GT 60000」に設定します。
2. &Y が YES の場合、プロシジャは 2 番目の -SET コマンドを省略して、START ラベルに分岐します。
3. &Y が NO の場合、プロシジャは 2 番目の -SET コマンドへ進み、&R の値をコメントを表す「-\*」に設定します。  
レポートリクエストがスタックされます。
4. プロシジャが &R の値を評価します。エンドユーザがレコード選択テストを行うよう指定した場合、&R の値は「IF SALARY GT 60000」となり、この行がスタックされます。
5. エンドユーザがレコード選択テストを行うよう指定しなかった場合、&R の値は「-\*」となり、この行は無視されます。

## 参照 日付時間ロケールパラメータのシステム日付変数への適用

DATE\_SEPARATOR パラメータは、日付区切り文字を定義し、DATE\_ORDER パラメータは、日付構成要素の順序をユーザのロケールに基づいて定義します。TIME\_SEPARATOR パラメータは、&TOD 変数の時間区切り文字を定義します。これらの設定をダイアログマネージャシステム変数 (例、&DATE、&TOD、&YMD、&DATEfmt、&DATXfmt) とともに使用するには、システム変数に .DATE\_LOCALE 接尾語を追加します。これにより、ローカライズされているシステム変数と、ローカライズされていないシステム変数の共存が可能になります。

## 例 システム変数の日付時間パラメータの設定

次のリクエストは、&DATE システム変数に DATE\_ORDER および DATE\_SEPARATOR パラメータを適用します。

```
SET DATE_SEPARATOR = DASH
SET DATE_ORDER = DMY
- TYPE NON-LOCALIZED: &DATE
- TYPE LOCALIZED: &DATE.DATE_LOCALE
```

出力結果は次のとおりです。

```
NON-LOCALIZED: 04/07/17
LOCALIZED: 07-04-17
```

## データファイルにアクセスせずに変数値を検証 - REGEX

REGEX マスクを使用することで、データにアクセスせずに変数値を検証することができます。REGEX マスクでは、検証文字列として使用する正規表現を指定します。正規表現は、検索パターンを作成するために一連の特殊文字とリテラル文字を組み合わせたものです。

Web 上には、正規表現に関する参考情報が多く提供されています。概要については、『TIBCO WebFOCUS サーバ管理者ガイド』の「セキュリティ」の「正規表現の概要」を参照してください。

エラーが発生した場合、次のメッセージが表示されます。

```
(FOC2909) 正規表現が無効です：
(FOC2910) レスポンスが正規表現と一致しません：
```

## 構文 REGEX マスクによる変数値の検証

```
&variable.(|VALIDATE=REGEX,REGEX='regexexpression').
```

説明

**&variable**

検証する変数です。

**regexexpression**

受容可能な値を指定する正規表現です。

## 例 REGEX マスクによる ID 番号の検証

次のリクエストは、xxxxxxx または xxx-xx-xxxx フォーマットの ID 番号を検証します。

```
-REPEAT NEXTFMT FOR &FMTCNT FROM 1 TO 2
-SET &EMPID1=DECODE &FMTCNT(1 '071382660' 2 '818-69-2173');
-SET &EMPID=IF
 &EMPID1.(|VALIDATE=REGEX,REGEX='^%d{3}%-?%d{2}%-?%d{4}$').Employee
ID. CONTAINS '-'
- THEN EDIT(&EMPID1,'999$99$9999') ELSE &EMPID1;
TABLE FILE EMPLOYEE
HEADING
" "
"Testing EMPID = &EMPID1</
1"
PRINT EID CSAL
WHERE EID EQ '&EMPID.EVAL'
ON TABLE SET PAGE NOPAGE
ON TABLE SET STYLE *
GRID=OFF,$
END
-RUN
-NEXTFMT
```

出力結果は次のとおりです。

```
Testing EMPID = 071382660
```

```
EMP_ID CURR_SAL
071382660 $11,000.00
```

```
Testing EMPID = 818-69-2173
```

```
EMP_ID CURR_SAL
818692173 $27,062.00
```

## 例 REGEX の使用と無効な値

次のリクエストでは、&EMPID1 の 2 つ目の値が REGEX マスクに適合しないため、この値は無効です。

```
-REPEAT NEXTFMT FOR &FMTCNT FROM 1 TO 2
-SET &EMPID1=DECODE &FMTCNT(1 '071382660' 2 '818-69-2173');
-TYPE EMPID1 = &EMPID1
-SET &EMPID=&EMPID1.(|VALIDATE=REGEX,REGEX='^%d{3}%d{2}%d{4}$').Employee
ID.;
-TYPE EMPID = &EMPID
-NEXTFMT
```

出力結果の FOC2910 メッセージには、&EMPID1 の 2 つ目の値が拒否されたことが示されています。

```
EMPID1 = 071382660
EMPID = 071382660
EMPID1 = 818-69-2173
 ERROR AT OR NEAR LINE 7 IN PROCEDURE __WCFEX FOCEXEC *
(FOC2910) RESPONSE DOES NOT MATCH THE REGULAR EXPRESSION: 818-69-2173
 ERROR AT OR NEAR LINE 7 IN PROCEDURE __WCFEX FOCEXEC *
(FOC295) A VALUE IS MISSING FOR: &EMPID1
```

## 例 REGEX の使用と無効な正規表現

次のリクエストでは、REGEX マスクは有効な正規表現ではありません。

```
-SET &EMPID1='071382660';
-SET &EMPID=&EMPID1.(|VALIDATE=REGEX,REGEX='^%d{3}%d{2}}%d{4}$').Employee
ID.;
```

出力結果の FOC2909 メッセージには、正規表現が無効であることが示されています。

```
ERROR AT OR NEAR LINE 5 IN PROCEDURE __WCFEX FOCEXEC *
(FOC2909) INVALID REGULAR EXPRESSION: ^%d{3}%d{2}}%d{4}$
 ERROR AT OR NEAR LINE 5 IN PROCEDURE __WCFEX FOCEXEC *
(FOC295) A VALUE IS MISSING FOR: &EMPID1
```

## フォーマット指定に対するユーザ入力値の確認

変数の入力値を一連のフォーマット条件と比較することで、変数値がフォーマット指定に準拠することを要件とするよう指定することができます。指定したフォーマットで値が入力されていない場合、入力文字数が多すぎることを示すメッセージがユーザに提示されます。ユーザはメッセージを確認した上で、有効な値を再入力することができます。

## 参照 変数のフォーマット指定

文字フォーマットを指定するには、「A」の後に文字数を記述します。文字数は、1 から 3968 までの値です。

整数フォーマットを指定するには、「I」の後に入力桁数を記述します。入力桁数は 1 桁から 10 桁までの範囲で指定し、値は  $2^{31}-1$  未満にする必要があります。また、この数字に使用する値には、小数点を含めることができます。

フォーマット指定は、ピリオド (.) で囲む必要があります。

入力変数値に対してフィールド名をテストする場合は、入力変数のフォーマットを指定します。このフォーマットを指定せず、入力値の長さがマスターファイルのフォーマット指定の長さを超える場合、プロシジャが終了し、エラーメッセージが表示されます。続行するには、プロシジャを再実行する必要があります。ただし、フォーマットを指定した場合に入力値がフォーマットの長さを超えると、入力値が拒否され、ユーザに値の再入力が要求されます。

**注意：**パラメータ値は、内部的に文字コードとしてメモリに格納されます。数値演算を実行する際は、最初にパラメータ値が倍精度浮動小数点数に変換されます。次に演算の結果が元の文字コードに戻され、小数点以下の桁数が切り捨てられます。そのため、数値コードで小数点以下の桁数を検索するテストは実行しないでください。

## 例

### フォーマット指定によるユーザ入力の確認

ここでは、STATE および QTY\_IN\_STOCK フィールドのフォーマット指定について考察します。

```
TABLE FILE CENTORD
BY QTY_IN_STOCK
BY STATE
BY PROD_NUM
BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE STATE EQ '&STATE.A2.US State:.'
WHERE QTY_IN_STOCK LT '&PRODNUM.I7.Quantity in Stock below:.'
END
```

上記の例で、ユーザが STATE 値として 3 文字以上を入力した場合、入力値が拒否され、「入力値の文字列が超過しています」というメッセージが表示された上で、ユーザに値の再入力が必要されます。

上記の例で、ユーザが QTY\_IN\_STOCK 値として 8 桁以上の数値を入力した場合、入力値が拒否され、「入力内の桁が多すぎます。[WebFOCUS フォーマットは 'I7']」というメッセージが表示された上で、ユーザに値の再入力が必要されます。

### 値範囲に対するユーザ入力値の確認

変数の入力値を値範囲と比較することで、変数値が範囲内に収まることを要件とするよう指定することができます。指定した範囲内に入力値が収まらない場合、値の範囲外であることを示すメッセージがユーザに提示されます。ユーザはメッセージを確認した上で、有効な値を再入力することができます。

## 参照 変数の値範囲の指定

値の範囲は (FROM value1 TO value2) の形式で指定し、範囲全体を括弧で囲みます。ここで、value 1 は value2 より小さい数値を表します。以下はその例です。

```
&QTY_IN_STOCK.(FROM 1 TO 10000)
```

**注意：**パラメータ値は、内部的に文字コードとしてメモリに格納されます。数値演算を実行する際は、最初にパラメータ値が倍精度浮動小数点数に変換されます。次に演算の結果が元の文字コードに戻され、小数点以下の桁数が切り捨てられます。そのため、数値コードで小数点以下の桁数を検索するテストは実行しないでください。

## 例 値範囲指定によるユーザ入力値の確認

次の例では、&QTY\_IN\_STOCK 変数に対して値の範囲を指定します。

```
TABLE FILE CENTINV
PRINT QTY_IN_STOCK
BY PROD_NUM
BY PRODNAME
ON TABLE SUBHEAD
"Inventory Report"
WHERE QTY_IN_STOCK LT &QTY_IN_STOCK.(FROM 1 TO 10000).Quantity in Stock
below:.
END
```

上記の例で、ユーザが &QTY\_IN\_STOCK 変数値として 1 未満の値または 10000 を超える値を入力した場合、入力値が拒否され、「value entered is not within range 1 - 10000」というメッセージが表示されます。

## インデックス付き変数による個数集計

ある変数の値を別の変数の値に付加して、インデックス付き変数を作成することができます。この機能は、ローカル変数およびグローバル変数の両方に適用されます。

インデックス付き変数の値が数値の場合、従来のコンピュータプログラミング言語における配列と同様の効果が得られます。たとえば、インデックス &K の値が 1 から 10 に変わる場合、変数 &AMOUNT.&K は、&AMOUNT1 から &AMOUNT10 まで 10 個の変数のうちのいずれかになります。

数値インデックスはカウンタとして使用することができます。また、プロシジャで設定、増加、テストすることができます。

## 構文 インデックス付き変数の作成

```
-SET &name.&index[.&index...] = expression;
```

説明

**&name**

変数名です。

**&index**

数値または文字の変数で、この値が &name に付加されます。ピリオド (.) は必須です。

複数のインデックスを使用する場合は、すべてのインデックス値が連結され、その文字列が変数名に付加されます。

たとえば &V1120 は、&V.&I.&J.&K の場合、&I=1、&J=12、&K=0 になります。

**expression**

有効な式です。

## 例 ループでのインデックス付き変数の使用

インデックス付き変数をループで使用することができます。次の例は、従来のプログラミング言語で使用される DO ループと同等のループを作成する方法を示しています。

```
-SET &N = 0;
-LOOP
-SET &N = &N+1;
-IF &N GT 12 GOTO OUT;
-SET &MONTH.&N=&N;
-TYPE &MONTH.&N
-GOTO LOOP
-OUT
```

この例では、&MONTH がインデックス付き変数で、&N がインデックスです。インデックスの値は、-SET コマンドから指定されます。最初の -SET がインデックスの値を 0 (ゼロ) に初期化し、プロシジャがループを通過するたびに 2 番目の -SET がインデックスの値を 1 ずつ増加させます。

参照する前にインデックスの値が定義されていない場合は、ブランク値であると見なされます。このため、インデックス付き変数の名前と値は変わりません。

インデックス付き変数は、システム制限の 384 個の変数に含まれます。

## 標準引用符付き文字列の作成

ほとんどのデータベースエンジンでは、処理を正しく実行するために、文字列を一重引用符 (') で囲む必要があります。また、文字列に一重引用符 (') が含まれている場合は、一重引用符を 2 つ連続させることにより一重引用符を表します。ブランクやカンマ (,) などの区切り文字を含む変数は、引用符で囲む必要があります。

ダイアログマネージャの変数に QUOTEDSTRING 接尾語を付加することにより、変数の内容は次の 2 つの規則に従って変換されます。

❑ 文字列内に含まれる一重引用符 (') は、すべて 2 つの一重引用符に変換されます。

❑ 文字列は、一重引用符 (') で囲まれます。

ダイアログマネージャのコマンドは、一重引用符 (') で囲んでいない文字列および埋め込みブランクを含む文字列を取り扱う場合、その処理方法が変わります。明示的または暗示的な -PROMPT コマンドは、このような文字列を読み取ることができます。入力した文字列全体は、.QUOTEDSTRING を実行した際に一重引用符 (') で囲まれます。

**注意：**-SET コマンドを使用して文字列を参照する場合、エラーが発生しないように文字列を一重引用符 (') で囲む必要があります。

### 構文 標準引用符付き文字列の作成

`&var.QUOTEDSTRING`

説明

`&var`

ダイアログマネージャの変数です。

### 例 標準引用符付き文字列の作成

次の例は、入力文字列に QUOTEDSTRING 接尾語を加えた場合の結果を示しています。

```

-SET &A = ABC;
-SET &B = 'ABC';
-SET &C = O'BRIEN;
-SET &D = 'O'BRIEN';
-SET &E = 'O''BRIEN';
-SET &F = O''BRIEN;
-SET &G = OBRIEN';
-TYPE ORIGINAL = &A QUOTED = &A.QUOTEDSTRING
-TYPE ORIGINAL = &B QUOTED = &B.QUOTEDSTRING
-TYPE ORIGINAL = &C QUOTED = &C.QUOTEDSTRING
-TYPE ORIGINAL = &D QUOTED = &D.QUOTEDSTRING
-TYPE ORIGINAL = &E QUOTED = &E.QUOTEDSTRING
-TYPE ORIGINAL = &F QUOTED = &F.QUOTEDSTRING
-TYPE ORIGINAL = &G QUOTED = &G.QUOTEDSTRING

```

出力結果は次のとおりです。

```

ORIGINAL = ABC QUOTED = 'ABC'
ORIGINAL = ABC QUOTED = 'ABC'
ORIGINAL = O'BRIEN QUOTED = 'O''BRIEN'
ORIGINAL = O'BRIEN QUOTED = 'O''BRIEN'
ORIGINAL = O'BRIEN QUOTED = 'O''BRIEN'
ORIGINAL = O''BRIEN QUOTED = 'O''''BRIEN'
ORIGINAL = OBRIEN' QUOTED = 'OBRIEN'''

```

**注意：**-SET コマンドは、文字列を囲んでいる一重引用符 (') を削除します。上記の例では、QUOTEDSTRING 変換の前に、-SET &B = 'ABC' の結果が (出力に示されるよう) ORIGINAL = ABC に変換されています。

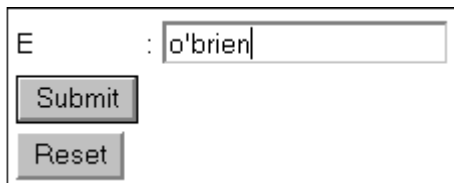
## 例

### 標準引用符付き文字列へのユーザ入力の変換

次の例は、-TYPE コマンドが引用符付きまたは引用符なしの入力を受容して、引用符付きの出力を表示する様子を示しています。

```
-TYPE THE QUOTED VALUE IS: &E.QUOTEDSTRING
```

出力結果は次のとおりです。



The screenshot shows a web form with a text input field labeled 'E' containing the text 'o'brien'. Below the input field are two buttons: 'Submit' and 'Reset'.

**THE QUOTED VALUE IS: 'o''brien'**

## 例 リレーショナルデータアダプタでの引用符付き文字列の使用

次のプロシジャは、VIDEOTRK データソースから「SQLVID」という名前の Oracle テーブル を作成します。

```
TABLE FILE VIDEOTRK
SUM CUSTID EXPDATE PHONE STREET CITY STATE ZIP
 TRANSDATE PRODCODE TRANSCODE QUANTITY TRANSTOT
BY LASTNAME BY FIRSTNAME
WHERE LASTNAME NE 'NON-MEMBER'
ON TABLE HOLD
END
-RUN
CREATE FILE SQLVID
-RUN
MODIFY FILE SQLVID
FIXFORM FROM HOLD
DATA ON HOLD
END
```

次のような SQL トランスレータのリクエストについて考察します。

```
SET TRACEUSER = ON
SET TRACEON = STMTRACE//CLIENT
SQL
SELECT *
FROM SQLVID WHERE LASTNAME = &1.QUOTEDSTRING;
END
```

このリクエストを実行する際に、姓 (ここでは、O'BRIEN) を入力する必要があります。

```
PLEASE SUPPLY VALUES REQUESTED
1=
O'BRIEN
```

生成された SQL リクエストでは、例として使用した文字列が正しく一重引用符 (') で囲まれ、文字列に含まれていた一重引用符には 2 つの引用符が使用されています。

```
SELECT SQLCOR01.CIN , SQLCOR01.LN , SQLCOR01.FN ,
SQLCOR01.EXDAT , SQLCOR01.TEL , SQLCOR01.STR , SQLCOR01.CITY ,
SQLCOR01.PROV , SQLCOR01.POSTAL_CODE , SQLCOR01.OUTDATE ,
SQLCOR01.PCOD , SQLCOR01.TCOD , SQLCOR01.NO , SQLCOR01.TTOT
FROM SQLVID SQLCOR01 WHERE SQLCOR01.LN = 'O''BRIEN';
```

次の入力例は、前例で示したように、正しい引用符付きの文字列に変換されます。

```
'O'BRIEN'
'O''BRIEN'
```

この他にもさまざまなケースがあります。

❑ 有効な文字列であるが、データベースの値と一致せず、行を返さない文字列。たとえば、`O'''BRIEN` は、WHERE 述語で `'O''''''BRIEN'` になります。

❑ 無効な文字列であるため、次のいずれかのメッセージを生成する文字列。

エラー - セミコロンまたは END を使用してください。

エラー - 引用符が使用されていない、または正しい位置にありません。

エラー - (入力値が) 有効なカラムではありません。

エラー - 行の構文エラー... 引用符が一致していません。

一重引用符が含まれていない文字列は、引用符なしで入力することも、一重引用符で囲んで入力することも可能です (SMITH または 'SMITH')。

リクエストに QUOTEDSTRING 接尾語なしで `&1` を使用した場合、O'Brien のレコードを検索する入力文字列に使用できるのは次のとおりです。

```
'''O'''BRIEN'''
'''O'''BRIEN'''
```

QUOTEDSTRING 接尾語なしで `&1` を使用した場合、一重引用符が含まれていない文字列で利用できる形式は `'''SMITH'''` です。

QUOTEDSTRING 接尾語を使用せずに、一重引用符で囲まれた文字列を使用できるようにするには、リクエストに `&'1'` を使用します。この場合、O'Brien のレコードを検索するには、次のように QUOTEDSTRING 接尾語から得られた文字列を入力する必要があります。

```
'O''''BRIEN'
```

`&'1'` を使用して一重引用符が含まれていない文字列を入力する場合は、引用符なしで入力することも、一重引用符で囲んで入力することもできます (SMITH または 'SMITH')。

**注意：**`&'1'.QUOTEDSTRING'` という形式はサポートされていません。

## 参照

### 引用符付き文字列の使用に関する注意

❑ パラメータの文字列をドリルダウンに渡す場合に、パラメータに複数選択された値が含まれている場合は、QUOTEDSTRING を使用する必要があります。詳細は、192 ページの「[値の複数選択リストに関する規則](#)」を参照してください。

❑ 文字列の開始部分で一重引用符に不整合が見つかった場合は、無効な入力と見なされて、次のメッセージが生成されます。

(FOC257) 引用符がありません

## 変数の作成と操作

変数を使用してプロシジャをカスタマイズすることができます。詳細は、314 ページの「[変数によるプロシジャのカスタマイズ](#)」を参照してください。変数は次の 2 つに分類されます。

- ❑ ローカル変数およびグローバル変数 - この変数に対するユーザ定義の値は、実行時に指定する必要があります。詳細は、348 ページの「[ローカル変数およびグローバル変数の作成と操作](#)」を参照してください。
- ❑ システム変数および統計変数 - この変数の値は事前に定義されているため、プロシジャが変数を参照する際にシステムから自動的に渡されます。詳細は、349 ページの「[ローカル変数およびグローバル変数の名前規則](#)」を参照してください。

## 参照

### 変数の特性

次の特性は、すべての変数に適用されます。

- ❑ 変数には数値またはテキスト文字列が格納され、プロシジャ内の任意の場所で変数を使用することができます。

**注意：**ダイアログマネージャ変数には、文字データのみが格納されます。関数または式がダイアログマネージャの変数に数値を返す場合は、この値は変数に格納される前に整数に切り捨てられて文字フォーマットに変換されます。

- ❑ 変数は、コマンド、データソースフィールド、動詞、句を参照することができます。
- ❑ すべての変数値の長さ合計は、最大で 32 キロバイトです。
- ❑ 変数値は、プロシジャの実行時にコマンドに直接渡すか、プロシジャ内の -DEFAULT、-SET、-READ コマンドから渡すことができます。

## ローカル変数およびグローバル変数の作成と操作

ローカル変数およびグローバル変数は、ユーザ定義の値を実行時に指定する変数です。

- ❑ ローカル変数の値は、プロシジャの実行中は保持されますが、そのプロシジャの処理が終了した時点で失われます。変数の値は、同一変数名を含む他のプロシジャには渡されません。

ローカル変数は、変数名の前に 1 つのアンパサンド (&) を付けて識別します。

- ❑ グローバル変数の値は、WebFOCUS Reporting Server に接続している間は保持され、1 つのプロシジャの実行が終了した時点で次のプロシジャに渡されます。

WebFOCUS では、リクエストを送信するたびに WebFOCUS Reporting Server に新しい WebFOCUS セッションが作成されるため、異なるレポートリクエスト間ではグローバル変数の値は保持されません。そのため、1 つのリクエストで複数のプロシジャを呼び出す場合は、同一のグローバル変数の値を使用することができます。

グローバル変数の値をすべてのプロシジャで有効にするには、EDASPROF.PRF などの WebFOCUS Reporting Server プロファイルに変数を追加します。

グローバル変数は、変数名の前に 2 つのアンパサンド (&&) を付けて識別します。

**注意：**値の中にカンマ (,) を埋め込むことはできません。

## 参照

### ローカル変数およびグローバル変数の名前規則

システム変数および統計変数は事前に名前が定義された変数ですが、ローカル変数およびグローバル変数の名前はユーザが定義します。ローカル変数およびグローバル変数には、次の名前規則が適用されます。

- ❑ ローカル変数名の前には、必ずアンパサンド (&) を 1 つ付けます。
- ❑ グローバル変数名の前には、必ずアンパサンド (&) を 2 つ付けます。
- ❑ 変数名にブランクを含めることはできません。
- ❑ 変数の値にブランク、カンマ (,)、等号 (=) が含まれている可能性がある場合は、変数を参照する際にその変数を一重引用符 (') で囲みます。
- ❑ 変数名には、A から Z、0 から 9、アンダースコア (\_) の各文字を任意に組み合わせて使用することができます。変数名の先頭には英文字を使用する必要があります。
- ❑ 変数に名前ではなく数字を割り当てることで、位置変数を作成することができます。
- ❑ 変数名にはアンダースコア (\_) を使用することはできますが、プラス記号 (+)、マイナス記号 (-)、アスタリスク (\*)、スラッシュ (/)、ピリオド (.), アンパサンド (&)、セミコロン (;) の特殊文字を使用することはできません。

## 構文

### 変数名の指定

&[&]name

### 説明

#### name

変数名です。1つのアンパサンド (&) はローカル変数を表し、2つのアンパサンド (&&) はグローバル変数を表します。1つのアンパサンド (&) の後に数字を指定すると、順序指定の変数になります。

名前を割り当てる際には特定の規則に従う必要があります。詳細は、349 ページの「[ローカル変数およびグローバル変数の名前規則](#)」を参照してください。

### 例 変数名の指定

次の例は、正しく指定された変数名を示しています。

```
&WHICHPRODUCT
&WHICH_CITY
'&CITY'
&&CITY
```

次の例は、正しく指定されていない変数名を示しています。その理由は以下のとおりです。

```
&WHICH CITY
```

ブランクが含まれています。

```
&WHICH -CITY
```

ハイフン (-) が含まれています。

```
WHICHCITY
```

先頭にアンパサンド (&) が付いていません。

### 例 ローカル変数の使用

ここでは、「SALESREPORT」というプロシジャについて考察します。このプロシジャでは、ローカル変数の &CITY、&CODE1、&CODE2 を使用しています。

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR &CITY"
"PRODUCT CODES FROM &CODE1 TO &CODE2"
" "
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY CITY
IF CITY EQ &CITY
BY PROD_CODE
IF PROD_CODE IS-FROM &CODE1 TO &CODE2
END
```

このプロシジャを呼び出した際に、次の値を入力するとします。

```
EX SALESREPORT CITY=STAMFORD, CODE1=B10, CODE2=B20
```

ダイアログマネージャは、各変数に対して次のように値を代入します。

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR STAMFORD"
"PRODUCT CODES FROM B10 TO B20"
" "
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY CITY
IF CITY EQ STAMFORD
BY PROD_CODE
IF PROD_CODE IS-FROM B10 TO B20
END
```

プロシジャの実行が終了した時点で、STAMFORD、B10、B20 のそれぞれの値は失われます。

## 例

### グローバル変数の使用

次の例は、「&&CITY」、「&&CODE1」、「&&CODE2」という 3 つのグローバル変数を使用する方法を示しています。ここでは、最初のプロシジャ PROC1 で変数に値が代入されます。それぞれの値が保持されて、2 つ目のプロシジャ PROC2 に渡されます。

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR &&CITY"
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY CITY
IF CITY EQ &&CITY
BY PROD_CODE
IF PROD_CODE IS-FROM &&CODE1 TO &&CODE2
END
EX PROC2
```

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR &&CITY AND PRODUCT &&CODE1"
PRINT UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY CITY
IF CITY EQ &&CITY
IF PROD_CODE EQ &&CODE1
END
```

## 変数の連結

変数を文字列に付加したり、複数の変数またはリテラルを組み合わせて変数を連結することができます。変数を連結する場合、各変数と連結記号との間に空白を挿入して、それぞれを分離する必要があります。

### 構文

#### 変数の連結

```
-SET &name3 = &name1 || &name2;
```

説明

**&name3**

連結後の変数名です。

**&name1 || &name2**

連結する変数名です。連結記号の前後に空白を挿入して区切ります。

## 変数末尾空白の削除

ダイアログマネージャの TRUNCATE 関数を使用して、変数末尾の空白を削除することができます。TRUNCATE は、関数をサポートする -SET および -IF コマンドなどのダイアログマネージャコマンドのみで 사용할 ことができます。-TYPE コマンドで使用したり、プロシジャに渡す引数で使用したりすることはできません。

TRUNCATE が 1 回の処理で取り扱える変数は 1 つです。ダイアログマネージャの TRUNCATE 関数で複数の引数を使用すると、次のエラーメッセージが生成されます。

(FOC03665) 外部関数 'TRUNCATE' のロードエラーが発生しました。

**注意：**ユーザ定義の関数がこの関数と同一名であっても競合しません。

### 構文

#### 末尾空白の削除

```
-SET &var2 = TRUNCATE(&var1);
```

説明

**&var2**

ダイアログマネージャの変数です。この変数に、切り捨てられた文字列が返されます。この変数の長さは、末尾の空白を除く、元の文字列または変数の長さになります。元の変数に空白のみが格納されている場合は、長さ 1 の空白が 1 つだけ返されます。

**&var1**

ダイアログマネージャ変数、または一重引用符 (') で囲まれたリテラルです。システム変数および統計変数を使用できるほか、ユーザが作成したローカル変数およびグローバル変数を使用することもできます。

**例 末尾ブランクの削除**

次の例は、TRUNCATE を使用して末尾のブランクを削除する方法を示しています。

```
-SET &LONG = 'ABC ' ;
-SET &RESULT = TRUNCATE(&LONG);
-SET &LL = &LONG.LENGTH;
-SET &RL = &RESULT.LENGTH;
-HTMLFORM BEGIN
<HTML>
<BODY>
<P>LONG = &LONG LENGTH = &LL</P>
<P>RESULT = &RESULT LENGTH = &RL</P>
</BODY>
</HTML>
-HTMLFORM END
```

出力結果は次のとおりです。

```
LONG=ABC LENGTH=06
```

```
RESULT=ABC LENGTH=03
```

**例 ブランク文字列での末尾ブランクの削除**

次の例は、TRUNCATE を使用して末尾のブランクを削除する方法を示しています。ここでは、ブランクのみで構成された文字列のため、返される文字列の長さは 1 になります。

```
-SET &LONG = ' ' ;
-SET &RESULT = TRUNCATE(&LONG);
-SET &LL = &LONG.LENGTH;
-SET &RL = &RESULT.LENGTH;
-HTMLFORM BEGIN
<HTML>
<BODY>
<P>LONG = &LONG LENGTH = &LL</P>
<P>RESULT = &RESULT LENGTH = &RL</P>
</BODY>
</HTML>
-HTMLFORM END
```

出力結果は次のとおりです。

```
LONG = LENGTH = 06
```

```
RESULT = LENGTH = 01
```

## 例 関数の引数としての TRUNCATE 関数の使用

次の例は、TRUNCATE を EDIT 関数の引数として使用する方法を示しています。

```
-SET &LONG = 'ABC ' ;
-SET &RESULT = EDIT(TRUNCATE(&LONG)|'Z','9999');
-SET &LL = &LONG.LENGTH;
-SET &RL = &RESULT.LENGTH;
-HTMLFORM BEGIN
<HTML>
<BODY>
<P>LONG = &LONG LENGTH = &LL</P>
<P>RESULT = &RESULT LENGTH = &RL</P>
</BODY>
</HTML>
-HTMLFORM END
```

出力結果は次のとおりです。

```
LONG = ABC LENGTH = 06
```

```
RESULT = ABCZ LENGTH = 04
```

## 変数値の表示

クエリを発行して、変数の値を表示することができます。

## 構文 ローカル変数値の表示

```
-? &[string]
```

説明

`source_string`

変数名です。このパラメータを指定しない場合は、ローカル変数、グローバル変数、システム変数、統計変数の現在の値がすべて表示されます。

## 構文 グローバル変数値の表示

```
? &&
```

## 変数へのパラメータ値の割り当て

SET パラメータの値をローカル変数に取り込むことができます。

WebFOCUS の場合、結果はブラウザウィンドウに返されるか、リクエストから他の HTML 出力がある場合は HTML ファイルのコメントとして返されます。HTML のコメントを表示するには、Web ブラウザでこの操作に該当する機能を使用します。たとえば、Microsoft Internet Explorer では、[ソースの表示] を使用します。

### 構文 変数へのパラメータ値の割り当て

```
-? SET parameter variable
```

説明

**parameter**

SET パラメータです。

**variable**

値の格納する変数です。

### 例 変数へのパラメータ値の割り当て

次のように入力すると、ASNames の値が &ABC の値として格納されます。

```
-? SET ASNames &ABC
```

このコマンドから &ABC を省略すると、ASNames の値を格納するための &ASNames 変数が作成されます。

## システム変数および統計変数の操作

システム変数および統計変数 - この変数の値は事前に定義されているため、プロシジャが変数を参照する際にシステムから自動的に渡されます。システム変数および統計変数は、名前の先頭にアンパサンド (&) が 1 つ付いています。これら変数の例として、出力した行数を表す &LINES や、現在の日付を表す &DATE があります。

- システム変数の値を使用して、処理を制御することができます。たとえば、選択条件に一致するレコードがないためレポートリクエストに読み取るレコードが存在しない場合に、その処理を停止したり、選択条件の変更を可能にしたりすることができます。

システム変数の一覧については、356 ページの「[WebFOCUS のシステム変数](#)」を参照してください。

- 統計変数の値をプロシジャに使用して、最後に実行したリクエストに関する情報を取得することができます。ダイアログマネージャが、統計変数の値を自動的に提供します。

統計変数の一覧については、363 ページの「[WebFOCUS の統計変数](#)」を参照してください。

参照 **WebFOCUS のシステム変数**

WebFOCUS のシステムから提供された値を上書きするには、-SET または -DEFAULT コマンドを使用して新しい値を明示的に変数に割り当てます。ただし、これらの値を上書きすることは推奨しません。

下表は、WebFOCUS で使用できるシステム変数の一覧です。

システム変数	説明	フォーマットまたは値
<a href="#">&amp;APPROOT</a>	ディレクトリおよびデータが格納されます。デフォルト設定では、WebFOCUS がプロジェクトファイルを検索するアプリケーションルートディレクトリ (APPROOT) になります。	d:¥ibi¥apps
<a href="#">&amp;AUTOINDEX</a>	TABLE リクエストにインデックスフィールドに対する等価テストまたは範囲テストが含まれている場合、インデックスフィールドを自動的に利用してデータを迅速に取得します。FOCUS データソースにのみ適用されます。  TABLE リクエストに代替ファイルビューが含まれている場合 (例、TABLE FILE filename.filename)、AUTOINDEX は実行されません。TABLE リクエストに BY HIGHEST または BY LOWEST 句が使用され、AUTOINDEX が ON に設定されている場合、インデックス検索は実行されません。	No
<a href="#">&amp;DATE</a>	現在の日付を返します。	MM/DD/YY

システム変数	説明	フォーマットまたは値
&DATEfmt &DATXfmt	現在の日付値または日付時間値を返します。ここで、fmt は有効な任意の日付または日付時間フォーマットを表します。&amp;DATEfmt を使用すると、返された値の末尾の空白が保持されます。&amp;DATXfmt を使用すると、返された値の末尾の空白が省略されます。   <b>注意:</b> 連結記号 ( ) を使用して構成要素間のピリオド (.) を削除する方法はサポートされません。構成要素間のピリオド (.) を削除して値を返すには、&amp;YYMD または &amp;DATEHYYMDN を使用します。	現在の日付値または日付時間値を返します。ここで、fmt は有効な任意の日付または日付時間フォーマットを表します。DATE 接頭語の後には多くの日付フォーマットオプションを追加してさまざまな変数名を作成できるため、ユーザ独自の変数名を作成する際は、DATE を接頭語にした変数名は作成しないことをお勧めします。
&DMY	現在の日付を返します。	DDMMYY
&DMYY	現在の日付を返します (4 桁の年)。	DDMMCCYY
&ECHO	プロシジャのテストおよびデバッグを目的として、実行したコマンド行を表示します。	ON, OFF, ALL, or NONE
&EXITRC	オペレーティングシステムコマンドの実行から返されるリターンコード値です。&EXITRC を参照すると、-RUN コマンドと同様に、すべてのスタックコマンドが強制実行されます。	コマンドから返される値はすべて有効ですが、0 (ゼロ) は正常な実行 (成功) と見なされます。
&FOCCODEPAGE	サーバで使用されているコードページを返します。	整数値

システム変数	説明	フォーマットまたは値
<code>&amp;FOCFEXURL</code>	ドリルダウンをリモートで起動および実行します。ドリルダウンプログラムは、ローカルマシンまたはリモートマシンのどちらにでも常駐させることができます。	<code>/ibi_apps/ WFServlet? IBIF_webapp= / ibi_apps&amp;IBIC_serv er</code>
<code>&amp;FOCFEXNAME</code>	別プロシジャの EX コマンドまたは - INCLUDE コマンドで実行された場合も含め、実行中のプロシジャ名を返します。この変数は、&FOCFOCEXEC 変数とは異なります。&FOCFOCEXEC は呼び出し側のプロシジャ名のみを返します。	
<code>&amp;FOCFOCEXEC</code>	EX コマンドを使用して名前が類似した多数のリクエストを実行する場合に、そのレポート処理を管理します。&FOCFOCEXEC を使用すると、プロシジャの完全修飾パスが返されるため、実行中のプロシジャを簡単に特定することができます。リクエストまたはダイアログマネージャコマンドに &FOCFOCEXEC を指定して、現在のプロシジャの名前を表示することができます。	

システム変数	説明	フォーマットまたは値
&FOCHTMLURL	/ibi_apps/ibi_html 以外のエイリアスを使用したリソースへのアクセスを可能にします。   WebFOCUS Reporting Server で /ibi_html 以外のエイリアスを生成するには、SET FOCHTMLURL コマンドを使用して、/ibi_apps/ibi_html の代わりに生成するエイリアスを設定します。   多くの場合、このコマンドはサーバプロファイル (EDASPROF.PRF) で使用するか、インストールのデフォルト値を設定するために WFS ファイル (site.wfs) で使用します。	/ibi_apps/ibi_html
&FOCINCLUDE	-INCLUDE コマンドを使用して名前が類似した多数のリクエストを追加する場合、そのレポート処理を管理します。リクエストまたはダイアログマネージャコマンドに &amp;FOCINCLUDE を指定して、現在のプロシジャの名前を表示することができます。	
&FOCLANGCODE	3 文字の言語コードを格納します (例、ドイツ語は GER、フランス語は FRE)。言語が英語の場合 (AME または ENG)、この変数はブランクになります。	
&FOCMODE	オペレーティングシステムを特定します。	CMS、CRJE、MSO、OS、TSO のいずれか。
&FOCNEXTPAGE	最後のレポートで使用する最終ページ番号で値が決定する変数です。この値は、最後のレポートで使用する最終ページ番号より 1 つだけ大きい数字になります。	0

システム変数	説明	フォーマットまたは値
&FOCQUALCHAR	修飾フィールド名の区切りに使用されている文字を返します。	: ! %   ¥
&FOCREL	FOCUS バージョン番号を識別します。	バージョン番号。
&FOCSECUSER	セキュリティオンで稼働中の Reporting Server で実行中エージェントに接続しているユーザのユーザ ID を返します。	
&FOCSECGROUP	&FOCSECUSER に格納されているユーザ ID のプライマリグループ ID を返します。	
&FOCSECGROUPS	&FOCSECUSER に格納されているユーザ ID のグループ ID のリストを返します。  返されるリストの長さは、4000 バイトに制限されています。	
&MDY	現在の日付を返します。数値比較に役立ちます。	MMDDYY
&MDYY	現在の日付を返します (4 桁の年)。	MMDDCCYY

システム変数	説明	フォーマットまたは値
&RETCODE	サーバコマンドが実行された後に返される値です。   &amp;RETCODE は、-RUN コマンドなどのスタックコマンドをすべて実行します。	サーバコマンドで定義された値。   コマンドから返される値はすべて有効ですが (例、CALLPGM フラグ値)、0 (ゼロ) は正常な実行 (成功) と見なされます。   例外として、ハイフン (-) で始まるオペレーティングシステムコマンド (例、-DOS、-UNIX、-WINNT) の &amp;RETCODE 値があります。この &amp;RETCODE 値は、これらの実行中コマンドの成功を表すのではなく、サーバがオペレーティングシステムへのサブプロセスを作成し、コマンドの実行に成功したことを表します。この場合、一般に &amp;RETCODE 値は 0 (ゼロ) です。これは、特定のコマンドの結果に関係なく、作成されたサブプロセスが正常に実行されるためです。この場合は &amp;EXITRC 変数を使用してコマンド結果を確認するか、ハイフン (-) で始まるコマンド以外のコマンドを使用する必要があります。

システム変数	説明	フォーマットまたは値
&SETFILE	SET FILE コマンドからの値が格納されます。	
&TOD	現在の時刻を返します。処理中に正確な時刻を取得するには、HHMMSS 関数を使用します。	HH.MM.SS
&YMD	現在の日付を返します。	YYMMDD
&YYMD	現在の日付を返します (4 桁の年)。	CCYYMMDD

**例**                    **&DATE による日付の取得**

次の例は、システム変数の &DATE をリクエストに組み込む方法を示しています。システム変数の &DATE を脚注に使用して、現在のシステム日付をレポートの下部に挿入します。

```
TABLE FILE SALES
SUM UNIT_SOLD
BY PROD_CODE
FOOTING
"CALCULATED AS OF &DATE"
END
```

**例**                    **&FOCFOCEXEC によるプロシジャ名の取得**

次の例では、システム変数 &FOCFOCEXEC をレポートリクエストに組み込み、現在のプロシジャの名前を表示しています。

```
SET PAGE=OFF
TABLE FILE EMPLOYEE
"REPORT: &FOCFOCEXEC -- EMPLOYEE SALARIES"
PRINT CURR_SAL BY EMP_ID
ON TABLE SET STYLE *
TYPE=REPORT, SIZE=10, GRID=OFF,$
END
```

このリクエストを「SALPRINT」というプロシジャに格納した場合、出力結果は次のようになります。

```
REPORT: SALPRINT -- EMPLOYEE SALARIES

EMP_ID CURR_SAL
```

071382660	\$11,000.00
112847612	\$13,200.00
115360218	\$ .00
117593129	\$18,480.00
119265415	\$9,500.00
119329144	\$29,700.00
121495681	\$ .00
123764317	\$26,862.00
126724188	\$21,120.00
219984371	\$18,480.00
326179357	\$21,780.00
451123478	\$16,100.00
543729165	\$9,000.00
818692173	\$27,062.00

## 参照

## WebFOCUS の統計変数

WebFOCUS では、次の統計変数を使用することができます。

統計変数	説明
<a href="#">&amp;BASEIO</a>	入出力操作の数です。
<a href="#">&amp;FOCDISORG</a>	FOCUS ファイルの非整合率です。
<a href="#">&amp;FOCERRNUM</a>	プロシジャの実行後、最終エラー番号を FOCnnnnn フォーマットで表示します。複数のエラーが発生した場合は、この変数は、最新のエラー番号を保持します。エラーが発生しなかった場合は、この変数値は 0 (ゼロ) になります。
<a href="#">&amp;INDEXIO</a>	インデックス入出力操作の数です。
<a href="#">&amp;LINES</a>	最後のアンサーセットで出力された行の数です。
<a href="#">&amp;READS</a>	外部ファイルからの物理読み取りの数です。

統計変数	説明
&RECORDS	最後のアンサーセットにより抽出されたレコード数です。
&SORTIO	ソートされた入出力操作の数です。

**例**                    **&LINES による出力行数のカウント**

次の例は、統計変数の &LINES の値を計算する方法を示しています。&LINES の値が 0 (ゼロ) の場合、制御は -RPT2 ラベルで識別される TABLE FILE EMPLOYEE リクエストに渡されます。&LINES の値が 0 (ゼロ) 以外の場合、制御は -REPTDONE ラベルに渡されて処理が終了します。

```
TABLE FILE SALES
HEADING CENTER
"MONTHLY REPORT FOR &CITY"
SUM UNIT_SOLD AND RETURNS AND COMPUTE
RATIO/D5.2 = 100 * (RETURNS/UNIT_SOLD);
BY CITY
IF CITY EQ &CITY
BY PROD_CODE
IF PROD_CODE IS-FROM &CODE1 TO &CODE2
END
-RUN
-IF &LINES EQ 0 GOTO RPT2 ELSE GOTO REPTDONE;
-RPT2
TABLE FILE EMPLOYEE
.
.
.
END
-RUN
-QUIT
-REPTDONE
-EXIT
```

関数での数値変数の使用

変数に格納されるデータが、文字、数値、文字と数値の混在のいずれの場合であっても、WebFOCUS はすべての変数を文字フォーマットの文字列として格納します。変数に使用可能なデータタイプは、文字と数値の 2 種類のみです。

変数データタイプの指定

変数のデータタイプは、データの内容のみに基づいて決定されます。そのため、-SET コマンドで数値を引用符で囲んだ場合でも、変数のデータタイプには影響しません。

たとえば、次のリクエストは、&A、&B、&C 変数に数値データを格納します。

```
-SET &A=12345;
-SET &B='12345';
-SET &C=123.45
-TYPE &A &B &C
-TYPE &A.TYPE &B.TYPE &C.TYPE
```

出力結果では、&A、&B、&C 変数のすべてが数値データタイプになっています。

```
12345 12345 123.45
N N N
```

## 変数の操作

変数を表示、代入、連結、追加する場合、変数に格納されている値は変換されません。

### 代入

```
-SET &C=123.45
IF RETAIL_COST EQ &C
```

次のように代入されます。

```
IF RETAIL_COST EQ 123.45
```

次の例についても考察します。

```
-SET &D= &C;
-TYPE &D &D.TYPE
```

出力結果では、&D には &C と同一の値が格納され、データタイプも数値になっています。

```
123.45 N
```

### 連結

&F 変数は、&A と &C の連結により作成されます。

```
-SET &F = &A | &C;
-TYPE &F &F.TYPE
```

出力結果では、&F の値は、&A の値の後に &C の値を連結した値になり、データタイプは数値になっています。

```
12345123.45 N
```

次の例では、文字列にアンパサンド (&) を埋め込むことで &E 変数を作成します。このアンパサンド (&) は変数名の開始記号とは見なされず、文字列内の文字記号として扱われます。

```
-SET &E = 1234&C;
-TYPE &E &E.TYPE
```

出力結果では、変数のデータタイプは数値ではなく、文字になっています。この結果は、「1234」という文字列と &C 変数を連結したものではありません。

```
1234&C A
```

連結を使用する方法でも、これと同一の結果を得ることができます。

```
-SET &G = AT&|T;
-TYPE &G &G.TYPE
```

出力結果は次のとおりです。

```
AT&T A
```

## 式での変数の使用

式で変数を使用する場合、その式を処理するために変換が必要になる場合があります。式で使用する変数は、一般にリテラルと見なされ、式を処理する前にその値が代入されます。この場合、式の処理に必要なデータ変換が実行されます。変数に格納されている数値は、整数と見なされます。式を整数として評価できる場合は、そのまま整数で評価されます。

次の例では、&C に 123.55 が設定されます。次に、式によって &C に 100 が加算されて &D が作成されます。

```
-SET &C=123.55;
-SET &D=&C + 100;
-TYPE &D &D.TYPE
```

出力結果では、&D のデータタイプは数値ですが、整数演算が使用されるため、その値の 123.55+100 の末尾が切り捨てられて整数の 223 になります。

```
223 N
```

次の式では、式の数値リテラル (100.49) が整数でないため、倍精度への変換が必要になります。

```
-SET &C=123.55;
-SET &D=&C + 100.49;
-TYPE &D &D.TYPE
```

出力結果では、&C の値を倍精度に変換することで演算が実行されましたが、その値が &D に返される前に末尾が切り捨てられています。

224 N

出力結果で小数部を保持するには、DMPRECISION パラメータの値を、出力変数に返す小数部の桁数に設定します。

以下はその例です。

```
SET DMPRECISION=2
-RUN
-SET &C=123.55;
-SET &D=&C + 100.49;
```

出力結果では、小数部が保持されています。

224.04 N

## 関数パラメータとしての変数の使用

関数に数値変数を渡す場合の変数の取り扱い方法は、関数パラメータのデータタイプによって異なります。

### 数値関数パラメータとしての数値変数の使用

関数呼び出しで数値パラメータとして数値変数を使用する場合、その変数はフィールドとして扱われます。変数はフィールドとして扱われますが、マスターファイルやプロシジャでデータタイプが指定されていないため、変数にはデフォルトのデータタイプである倍精度が指定されます。

出力変数のデータタイプは、その変数に結果が返されたときの値の内容によって決定されます。変数の値が数字と小数点のみの場合、データタイプは数値になります。変数の値にその他の記号が含まれている場合、データタイプは文字になります。

たとえば、FTOA 関数は、倍精度 (D) または単精度 (F) の数値を、2 つ目のパラメータの括弧内で指定されたフォーマットの文字列に変換します。

```
FTOA (number_to_convert, '(format)', 'alpha_output_format')
```

次の例では、&C が 123.55 に設定され、その値が FTOA 関数に渡されて円記号 (¥) を含む文字列に変換されます。

```
-SET &C=123.55;
-SET &G=FTOA(&C, '(D7.2M)', 'A11');
-TYPE &G &G.TYPE
```

出力結果では、文字列の ¥123.55 が &G に返されています。この変数値には数字と小数点以外の記号が含まれているため、データタイプは文字になります。

```
$123.55 A
```

次の例では、返されるフォーマットには円記号 (¥) は指定されていません。

```
-SET &A=12345;
-SET &G=FTOA(&A/100, '(D7.2)', 'A11');
-TYPE &G &G.TYPE
```

返された文字列は数字と小数点のみのため、データタイプは数値になります。

```
123.45 N
```

ただし、この数値の桁数が 1 つ増えると、カンマ (,) 付きの数値として返されるため、データタイプは文字になります。

```
-SET &A=123456;
-SET &G=FTOA(&A/100, '(D7.2)', 'A11');
-TYPE &G &G.TYPE
```

出力結果は次のとおりです。

```
1,234.56 A
```

## 文字関数パラメータとしての数値変数の使用

関数呼び出しで文字パラメータとして数値変数を使用する場合、その変数を使用する前に数値を文字列に変換する必要があります。これにより、フォーマットエラーによる失敗が回避されます。これを行うには、数値を文字に変換する関数のいずれかを使用するか、変数に文字データタイプを割り当てるために文字列と数値を連結します。

たとえば、次の例では、&C が文字列に変換され、その文字列が &G に返されます。次に、&G が RJUST 関数に渡され、値が右揃えに設定されて &H 変数に返されます。

```
-SET &C=123.55;
-SET &G=FTOA(&C, '(D7.2M)', 'A11');
-SET &H = RJUST(11,&G, 'A11');
-TYPE &G &G.TYPE
-TYPE &H &H.TYPE
```

出力結果は次のとおりです。

```
$123.55 A
$123.55 A
```

## プロシジャ実行の制御

ダイアログマネージャコマンドを使用して、プロシジャの実行を制御することができます。この目的に使用するコマンドには、-RUN、-EXIT、-QUIT、-QUIT FOCUS があります。

- ❑ -RUN は、すべてのスタックコマンドを即座に実行し、外部ファイルを閉じた上で、プロシジャを続行します。詳細は、369 ページの「[スタックコマンドの実行とプロシジャの続行](#)」を参照してください。
- ❑ -EXIT は、強制的にスタックコマンドを実行し、プロシジャを終了します。詳細は、369 ページの「[スタックコマンドの実行とプロシジャの続行](#)」を参照してください。
- ❑ -QUIT は、すべてのスタックコマンドの実行をキャンセルし、プロシジャを即座に終了します。詳細は、371 ページの「[プロシジャ実行のキャンセル](#)」を参照してください。
- ❑ -QUIT FOCUS は、プロシジャおよび FOCUS を終了します。詳細は、371 ページの「[プロシジャ実行のキャンセル](#)」を参照してください。

## スタックコマンドの実行とプロシジャの続行

-RUN コマンドを使用して、スタックコマンドを実行した上で、プロシジャを続行することができます。

-RUN コマンドは、すべてのスタックコマンドを実行し、-READ または -WRITE コマンドを使用して開いたすべての外部ファイルを閉じます。スタックコマンドの実行後、-RUN の次の行からプロシジャの処理が続行します。

### 例      スタックコマンドの実行とプロシジャの続行

次の例は、-RUN を使用してスタックコードを実行した後、プロシジャに戻る方法を示しています。左側の番号は、コードの説明に対応しています。

```

1. TABLE FILE SALES
 PRINT PROD_CODE UNIT_SOLD
 BY CITY
 END
2. -RUN
 TABLE FILE EMPLOYEE
 PRINT LAST_NAME FIRST_NAME
 BY DEPARTMENT
 END

```

このプロシジャは、次のように処理されます。

1. 最初の 4 行は、レポートリクエストです。それぞれの行は、後から実行するためにスタックに配置されます。

2. -RUN は、スタックコマンドを実行します。これらは、処理のために WebFOCUS Reporting Server に送信されます。-RUN の次の行から処理が続行します。

## スタックコマンドの実行とプロシジャの終了

-EXIT コマンドを使用すると、スタックコマンドを実行した後、プロシジャを終了することができます。-EXIT は、スタックコマンドを見つけると、即座にそのコマンドを強制実行します。

- WebFOCUS では、-EXIT はすべての外部ファイルを閉じて、プロシジャを終了します。別のプロシジャが呼び出したプロシジャの場合は、呼び出し側のプロシジャに制御が返されます。

### 例

#### スタックコマンドの実行とプロシジャの終了

次の例では、最初のレポートリクエストまたは 2 番目のレポートリクエストのいずれかが実行されますが、両方は実行されません。

```
1. -SET &PROC = 'SALES'
2. -IF &PROC EQ 'EMPLOYEE' GOTO EMPLOYEE;
 -SALES
3. TABLE FILE SALES
 SUM UNIT_SOLD
 BY PROD_CODE
 END
4. -EXIT
 -EMPLOYEE
 TABLE FILE EMPLOYEE
 PRINT LAST_NAME
 BY DEPARTMENT
 END
```

このプロシジャは、次のように処理されます。

1. ダイアログマネージャは、&PROC に SALES を渡します。
2. -IF テストが完了し、&PROC の値が EMPLOYEE でないためにテストが失敗すると、制御は次の行の -SALES に渡されます。

この場合、&PROC の値が EMPLOYEE であれば、制御は -EMPLOYEE に渡されることになります。

3. FOCUS コードが処理され、後から実行するためスタックに配置されます。
4. -EXIT は、スタックコマンドを実行します。出力が WebFOCUS Client アプリケーションに送信され、プロシジャが終了します。

-EMPLOYEE ラベルより下のリクエストは実行されません。

この例は、暗黙的に終了する方法も示しています。&PROC の値が EMPLOYEE の場合、-IF テストの後に制御が -EMPLOYEE ラベルに渡されるため、このプロシジャが -EXIT を検出することはありません。TABLE FILE EMPLOYEE リクエストが実行され、プロシジャは自動的に終了します。

## プロシジャ実行のキャンセル

-QUIT コマンドを使用して、プロシジャの実行をキャンセルすることができます。-QUIT は、すべてのスタックコマンドの実行をキャンセルし、プロシジャを即座に終了します。別のプロシジャが呼び出したプロシジャの場合でも、制御は WebFOCUS アプリケーションに直接返されます。このコマンドは、テストまたは演算の結果が別の処理を必要としない場合に役立ちます。

### 構文 プロシジャ実行のキャンセル

`-QUIT`

### 構文 プロシジャ実行のキャンセルと FOCUS の終了

`-QUIT FOCUS [n]`

説明

`n`

オペレーティングシステムのリターンコード番号です。定数または変数のいずれかを指定することができます。変数には整数を指定する必要があります。値を指定しない場合または整数以外の値を指定した場合、異常終了したことを示すデフォルトのリターンコードが、オペレーティングシステムに送信されます。

ユーザが制御するリターンコードの主な役割は、処理中の問題を検知することです。リターンコードの値により、処理の続行または終了を決定します。これは、特にバッチ処理を行う場合に有効な方法です。

### 例 プロシジャ実行のキャンセル

次の例は、-QUIT を使用して、プロシジャ実行のキャンセルを -IF テストの結果に基づいて決定する方法を示しています。

```
1. -DEFAULT &CODE='B11';
2. -IF &CODE EQ '0' OR &CODE EQ 'DONE' GOTO QUIT;
3. TABLE FILE SALES
 SUM UNIT_SOLD
 WHERE PROD_CODE EQ &CODE
 END
4. -QUIT
```

このプロシジャは、次のように処理されます。

1. -DEFAULT コマンドは、&CODE のデフォルト値を B11 に設定します。
2. 「B11」という値が &CODE に渡されます。
3. FOCUS コードが処理され、後で実行するためにスタックされます。
4. -QUIT は、スタックコマンドの実行をキャンセルし、プロシジャを終了します。

## プロシジャ実行順序の制御

次の方法を使用して、プロシジャの実行順序を制御することができます。

- ❑ **無条件分岐** コントロールをラベルに転送します。
- ❑ **条件分岐** テスト結果に基づいてコントロールをラベルに転送します。
- ❑ **ループ** 1つの関数をプロシジャ内で繰り返し実行します。
- ❑ **別プロシジャの呼び出し** 別のプロシジャの一部または全体を現在のプロシジャに組み込みます。

## 無条件分岐の実行

-GOTO コマンドを使用して、指定したラベルに制御を転送する無条件分岐を実行することができます。

ダイアログマネージャは、プロシジャの検索をはじめて実行する際に、検知したすべてのラベルのアドレスを記録します。これにより、これらのラベルを次に使用する際に即座に見つけることができます。ただし、ダイアログマネージャが特定のラベルをバイパスする -GOTO コマンドを検知した場合、これらのラベルはリストに記録されません。ダイアログマネージャが -GOTO コマンドを処理する際は、最初にラベルのリストを確認します。ターゲットラベルがリストに存在する場合、制御はそのラベルに渡されます。ターゲットラベルがリストに存在しない場合、ダイアログマネージャはプロシジャ内を前方検索してターゲットラベルを見つけます。終点に到達してもラベルが見つからない場合は、プロシジャの先頭から検索を続行します。-GOTO と関連付けられていないラベルを検出しても、ダイアログマネージャは何も実行しません。

-GOTO を検出しても、それに対応するラベルが見つからない場合、プロシジャの実行を中断して、エラーメッセージを表示します。

**注意：**ダイアログマネージャコマンドをラベルとして使用しないことをお勧めします。

## 構文

### 条件分岐の実行

```
-GOTO label
.
.
-label [TYPE text]
```

#### 説明

##### -label

ユーザが定義する名前で、最大長は 64 バイトです。この名前にブランクを含めたり、-QUIT および -EXIT 以外のダイアログマネージャコマンドを使用したりすることはできません。数値演算や論理演算、関数と混同しやすい語、予約語は使用しないでください。

**注意：**CONTINUE という語句は -IF コマンドの一部でない -GOTO コマンド内のラベルとして使用できますが、CONTINUE は -IF コマンド内ではラベルとして認識されません。この場合、CONTINUE は常に -IF 直後のコマンドに移動します。

label のテキストは、プロシジャ内の -GOTO コマンドの前後に記述することができます。

**注意：**label を -GOTO コマンドの後に記述する場合、ハイフン (-) は前置しません。

##### TYPE text

クライアントアプリケーションにメッセージを送信します。

## 例

### 無条件分岐の実行

次の例は、行ごとに「\*」を前置するのではなく、無条件分岐を使用して FOCUS コードすべてをコメントとして扱う方法を示しています。

```
-GOTO DONE
TABLE FILE SALES
PRINT UNIT_SOLD RETURNS
BY PROD_CODE, CITY
END
-RUN
-DONE
```

## 条件分岐の実行

条件分岐は、-IF... GOTO コマンドを使用して変数値をテストし、テスト結果に基づいて指定したラベルにコントロールを転送します。これにより、リクエストの実行を制御できるため、プロシジャで実行する部分と実行しない部分を選択することで動的なプロシジャの作成が可能になります。

たとえば、稼働中のデータソースから抽出ファイルが作成されたかどうかを確認することができます。抽出ファイルが存在する場合は、プログラムはその抽出ファイルに対して一連のレポートを実行します。抽出ファイルが存在しない場合は、プログラムはレポートを回避して分岐し、ログファイルにメッセージを書き込みます。

**注意：**-IF テストでは、テストごとにターゲットラベルが存在する必要はありません。

## 構文

### 条件分岐の実行

```
-IF expression [THEN] {GOTO label1|CONTINUE} [ELSE IF...;] [ELSE {GOTO label2|CONTINUE}];
```

説明

**expression**

有効な式です。リテラルにブランクまたはカンマ (,) が含まれていない場合は、一重引用符 (') で囲む必要はありません。

**THEN**

コマンドを読みやすくするための任意指定のコマンドです。

**label1**

ユーザが定義する名前で、-IF テストが TRUE の場合にこのラベルに制御が渡されます。この名前の最大長は 64 バイトです。この名前にブランクを含めたり、-QUIT および -EXIT 以外のダイアログマネージャコマンドを使用したりすることはできません。数値演算や論理演算、関数と混同しやすい語、予約語は使用しないでください。CONTINUE という語句は -IF コマンドの一部でない -GOTO コマンド内のラベルとして使用できますが、CONTINUE は -IF コマンド内ではラベルとして認識されません。この場合、CONTINUE は常に -IF 直後のコマンドに移動します。

label のテキストは、プロシジャ内の -IF 条件の前後に記述することができます。

**CONTINUE**

-IF コマンドのセミコロン (;) に続くコマンドに移動します。

**注意：**-IF ステートメントのラベルとして CONTINUE を使用することはできません。

**ELSE IF**

複合 -IF テストを指定します。-IF コマンドを終了する場合は、すべてのロジックが指定されたことを表すセミコロン (;) を末尾に追加する必要があります。詳細は、377 ページの「[複合 -IF テストの実行](#)」を参照してください。

**ELSE GOTO**

-IF テストに失敗した場合にコントロールを label2 に渡します。

コマンドが複数行にわたる場合は、継続行の先頭にハイフン (-) と 1 つ以上のブランクを追加する必要があります。

**例 条件分岐の実行**

次の例では、&OPTION が S に等しい場合、制御が -PRODSALES ラベルに渡されます。等しくない場合は、プロシジャの次の行にある -PRODRETURNS ラベルに渡されます。

```
-IF &OPTION EQ 'S' GOTO PRODSALES;
-PRODRETURNS
TABLE FILE SALES
PRINT PROD_CODE UNIT_SOLD
BY STORE_CODE
END
-EXIT
-PRODSALES
TABLE FILE SALES
SUM UNIT_SOLD
BY PROD_CODE
END
-EXIT
```

次のコマンドは、両方の転送について明示的に指定しています。

```
-IF &OPTION EQ 'S' GOTO PRODSALES ELSE
 - GOTO PRODRETURNS;
```

継続行の先頭にハイフン (-) とブランクを追加していることに着目してください。

**例 システム変数および統計変数のテスト**

次の例では、リクエストによりデータ (&LINES) が取得されると、プロシジャが -PRODSALES ラベルに分岐します。データが取得されない場合は、プロシジャは終了します。

```
TABLE FILE SALES
SUM UNIT_SOLD
BY PROD_CODE BY CITY
WHERE TOTAL UNIT_SOLD GE 50
ON TABLE HOLD
END
-IF &LINES NE 0 GOTO PRODSALES;
-EXIT
-PRODSALES
TABLE FILE SALES
SUM UNIT_SOLD
BY PROD_CODE ACROSS CITY
END
```

## 例

### スタックコマンドの実行とプロシジャの終了

次の例では、最初のレポートリクエストまたは 2 番目のレポートリクエストのいずれかが実行されますが、両方は実行されません。ここでは、このレポートを HTML フォームから起動する場合を想定します。このフォームは、「&PROC」というテキスト入力変数を使用して、ユーザにプロシジャの実行を要求します。ユーザは、「SALES」または「EMPLOYEE」のいずれかの値を入力することができます。

```
1. -IF &PROC EQ 'EMPLOYEE' GOTO EMPLOYEE;
2. -SALES
TABLE FILE SALES
SUM UNIT_SOLD
BY PROD_CODE
END
3. -EXIT
-EMPLOYEE
TABLE FILE EMPLOYEE
PRINT LAST_NAME
BY DEPARTMENT
END
```

このプロシジャは、次のように処理されます。

1. ダイアログマネージャは、&PROC に SALES を渡します。-IF テストが完了し、&PROC の値が EMPLOYEE でないためにテストが失敗すると、制御は次の行の -SALES に渡されます。  
この場合、&PROC の値が EMPLOYEE であれば、制御は -EMPLOYEE に渡されることになります。
2. FOCUS コードが処理され、後から実行するためスタックに配置されます。
3. -EXIT は、スタックコマンドを実行します。出力が WebFOCUS Client アプリケーションに送信され、プロシジャが終了します。  
-EMPLOYEE ラベルより下のリクエストは実行されません。

この例は、暗黙的に終了する方法も示しています。&PROC の値が EMPLOYEE の場合、-IF テストの後に制御が -EMPLOYEE ラベルに渡されるため、このプロシジャが -EXIT を検出することはありません。TABLE FILE EMPLOYEE リクエストが実行され、プロシジャは自動的に終了します。

## 例 条件分岐によるプロシジャ実行のキャンセル

次の例は、-QUIT を使用して、-IF テストの結果に基づいてプロシジャの実行をキャンセルする方法を示しています。ここでは、このレポートを HTML フォームから起動する場合を想定します。このフォームは、「&SELECT」というテキスト入力変数を使用して、ユーザにプロシジャの実行を要求します。

```
-IF &SELECT EQ 'DONE' GOTO QUIT;
-REPORT
 TABLE FILE SALES
 SUM UNIT_SOLD
 BY PROD_CODE
 WHERE AREA EQ '&SELECT';
 END
-RUN
-QUIT
```

ユーザは、ブラウザの [戻る] ボタンをクリックし、SELECT に新しい値を入力してフォームを送信することにより、レポートを連続的に実行することができます。製品コードを選択すると、プロシジャは次の行へ進み、-QUIT を検出する前にリクエストが実行されます。ユーザが「DONE」と入力すると、制御が -QUIT に渡されます。プロシジャは終了します。

## 複合 -IF テストの実行

複合 -IF テストは、互いにネストされた複数の -IF テストで構成される集合体です。各テストがターゲットラベルを指定する場合、複合 -IF テストを使用することができます。

## 例 複合 -IF テストの使用

次の例では、&OPTION の値が R または S と異なる場合に、プロシジャが終了します (-GOTO QUIT)。-QUIT は、GOTO のターゲットレベルおよび実行コマンドの両方として機能します。ここでは、このレポートを HTML フォームから起動する場合を想定します。このフォームは、「&OPTION」というテキスト入力変数を使用して、ユーザにプロシジャの実行を要求します。

```
-IF &OPTION EQ 'R' THEN GOTO PRODRETURNS ELSE IF
- &OPTION EQ 'S' THEN GOTO PRODSALES ELSE
- GOTO QUIT;
-PRODRETURNS
TABLE FILE SALES
PRINT PROD_CODE UNIT_CODE
BY STORE_CODE
END
-EXIT
-PRODSALES
TABLE FILE SALES
SUM UNIT_SOLD
BY PROD_CODE
END
-QUIT
```

### プロシジャでのループの実行

-REPEAT コマンドでプロシジャ内でループを使用することにより、1つの関数を繰り返し実行することができます。ループは、さまざまな処理に使用することができます。たとえば、ダイアログマネージャのループにオペレーティングシステムの呼び出しを埋め込んで、ファイルに名前を付けたり、名前を変更したりすることができます。ループを使用してインデックス付き変数を埋め込んだり、リクエストの出力を2番目のリクエストに使用したりすることも可能です。

処理ループは、指定した回数だけ実行したり、条件が満たされるまで実行したりすることができます。ループは、次のいずれかが発生した場合に終了します。

- ❑ ループが完了した場合。
- ❑ -QUIT または -EXIT コマンドが発行された場合。
- ❑ -GOTO が発行されて、ループ外のラベルに移動する場合。

**注意：**プロシジャ内で別の -GOTO を発行して再度ループに戻った場合、そのループは先に終了した箇所から続行します。

また、-SET コマンドを使用して変数の値を増加させることにより、ループの繰り返し回数を制限することもできます。

### 構文

#### ループの指定

```
-REPEAT label n TIMES
```

または

```
-REPEAT label WHILE condition;
```

または

```
-REPEAT label FOR &variable [FROM fromval] [TO toval] [STEP s]
```

説明

**label**

繰り返し実行するコード (ループ) を指定します。2 つのループのラベル名が異なる場合、一方のラベルに他方のラベルのループを含めることができます。

**n TIMES**

ループを実行する回数を指定します。n の値には、ローカル変数、グローバル変数、定数のいずれかを指定することができます。変数を指定した場合、評価は 1 回に限定されるため、ループの実行回数を変更することはできません。ループを途中で終了するには、-QUIT または -EXIT を使用します。

**WHILE condition**

ループを実行する条件を指定します。条件は、True または False を返す論理式です。条件が True の場合にループが実行されます。

**注意：**条件の後にセミコロン (;) を続ける必要があります。

**&variable**

ループの実行を開始する際にテストする変数です。fromval および toval の値が指定されている場合は、それらの値と比較されます。s が正の値のときは、&variable が toval 以下の場合にのみループが実行されます。s が負の値のときは、&variable が toval 以上の場合にのみループが実行されます。

**fromval**

ループを実行する &variable の最小値です。デフォルト値は 1 です。

**toval**

ループを実行する &variable の最大値です。デフォルト値は 1,000,000 です。

**STEP s**

&variable の値を定数の s ずつ増加させます。正または負の値を指定することができます。デフォルトの増分値は 1 です。

**注意：**パラメータの FROM、TO、STEP は、任意の順序で使用することができます。

**例**

**ループの繰り返し**

次の例は、-REPEAT の構文要素を示しています。

```
-REPEAT label n TIMES
```

以下はその例です。

```
-REPEAT LAB1 2 TIMES
-TYPE INSIDE
-LAB1 TYPE OUTSIDE
```

出力結果は次のとおりです。

```
INSIDE
INSIDE
OUTSIDE
```

```
-REPEAT label WHILE condition;
```

以下はその例です。

```
-SET &A = 1;
-REPEAT LABEL WHILE &A LE 2;
-TYPE &A
-SET &A = &A + 1;
-LABEL TYPE END: &A
```

出力結果は次のとおりです。

```
1
2
END: 3
```

```
-REPEAT label FOR &variable FROM fromval TO toval STEP s
```

以下はその例です。

```
-REPEAT LABEL FOR &A STEP 2 TO 4
-TYPE INSIDE &A
-LABEL TYPE OUTSIDE &A
```

出力結果は次のとおりです。

```
INSIDE 1
INSIDE 3
OUTSIDE 5
```

### 例 ループの制御

次の例は、-SET を使用してループを制御する方法を示しています。

```

1. -DEFAULT &N=0
2. -START
3. -SET &N=&N+1;
4. EX SLRPT
 -RUN
5. -IF &N GT 5 GOTO NOMORE;
6. -GOTO START
5. -NOMORE TYPE EXCEEDING REPETITION LIMIT
 -EXIT

```

プロシジャは、次の手順で実行されます。

1. -DEFAULT コマンドは、&N を初期値の 0 (ゼロ) に設定します。
2. -START は、ループの実行を開始します。これは、無条件 -GOTO のターゲットでもあります。
3. -SET コマンドは、ループが実行されるたびに &N の値を 1 ずつ増加させます。
4. FOCUS コマンドの EX SLRPT がスタックされます。次に -RUN が、スタックコマンドを実行します。
5. -IF コマンドは、&N 変数の現在値をテストします。この値が 5 より大きい場合、コントロールが -NOMORE ラベルに渡され、エンドユーザにメッセージを表示して強制終了します。&N の値が 5 以下の場合、コントロールは次のダイアログマネージャコマンドに移動します。
6. -GOTO は、コントロールを -START ラベルに渡し、ループを続行します。

## -INCLUDE によるプロシジャまたはスタイルシートのプロシジャへの挿入

-INCLUDE コマンドを使用して、レポートプロシジャ (.fex) またはスタイルシートファイル (.sty) を別のレポートプロシジャ (.fex) に挿入することができます。部分的なコードを挿入することもできます。部分的なプロシジャには、見出しテキストを含めたり、呼び出し側のプロシジャのテストに基づいて実行時に挿入するコードを含めたりすることができます。

挿入されるプロシジャおよびスタイルシートファイルは、WebFOCUS リポジトリまたは Reporting Server のアプリケーションディレクトリに格納することができます。ただし、挿入されるファイルおよび呼び出し元プロシジャに応じて、構文が異なります。

**注意：**レポートリクエストで「INCLUDE」という名前のスタイルシート属性を使用して、レポートプロシジャとは別の場所に格納されたスタイルシートを適用することもできます。INCLUDE 属性は、スタイルシートファイルのみで使用可能な属性であり、ここで説明する -INCLUDE コマンドと同一ではありません。

## 参照

### -INCLUDE の使用方法

-INCLUDE コマンドは、次の目的で 사용할 수 있습니다。

- ❑ 環境を制御する。たとえば、挿入したプロシジャでサーバ名やユーザ ID などの変数を設定してから、呼び出し側のプロシジャの処理を続行할 수 있습니다。
- ❑ セキュリティメカニズムとして 사용할 수 있습니다。挿入したプロ시ジャ를暗号화したり、パスワード를直接設定したり할 수 있습니다。
- ❑ 挿入されるプロ시ジャ가複数存在する場合にコード를短縮する。たとえば、-INCLUDE &NEWLINES コマンド를使用して挿入される프로시ジャ를特定することにより、GOTO コマンドの使用回수를低減할 수 있습니다。
- ❑ 標準の見出しや脚注など、アプリケーション全体で繰り返し使用されるコードの一部를再使用する。これにより、1 つ의모듈에加えた変更가アプリケーション全体で有効になります。
- ❑ 스타일시트 파일을挿入する。

## 参照

### 리포지트리프로시ジャ의 IBFS 풀パス参照의取得

리포지트리에 레포트프로시ジャまたは스타일시트 파일을作成すると、레포트의作成先ワーク스페이스、指定した파일名、파일타입에基づいて、一意의パスと파일名の参照가作成されます。파일의名前規則により、文字以外の特殊文字가置換され、파일名가変更される場合があります。

-INCLUDE 코만드를使用して리포지트리에存在する레포트프로시ジャまたは스타일시트 파일을挿入する場合、通常は、その IBFS 풀パス를指定します。ただし、挿入する파일および呼び出し元프로시ジャ가同一리포지트리폴더에格納されている場合は例外です。この場合、-INCLUDE 코만드에相對パス를指定할 수 있습니다。詳細は、387 페이지の「[相對パスによる리포지트리프로시ジャまたは스타일시트 파일의挿入](#)」를参照してください。

파일名を含めた IBFS 풀パス情報は、ホームページで 파일을右クリックし、[프로파티]를クリックすること で取得할 수 있습니다。

下図は、レポートプロシジャ (.fex) の [プロパティ] ダイアログボックスに表示されたフルパスを示しています。

001Report1

**General** Advanced Query Detail Server

Title 001Report1

Name 001Report1.fex

Summary

Path IBFS:/WFC/Repository/Public/001Report1.fex

Created Wednesday, July 29, 2020 12:43:18 AM EDT by admin

Modified Wednesday, July 29, 2020 12:43:18 AM EDT by admin

Accessed Wednesday, July 29, 2020 12:43:35 AM EDT by admin

Tool report

Owner admin

Size 1 KB

Publish ☐ Yes ☒ No

Show ☒ Yes ☐ No

Save Cancel

## 構文

**TIBCO WebFOCUS Reporting Server** プロシジャまたはスタイルシートファイルの別のプロシジャへの挿入

次の構文は、Reporting Server のアプリケーションディレクトリに格納されているプロシジャまたはスタイルシートファイルを、Reporting Server のアプリケーションディレクト内の別のレポートプロシジャに挿入する方法を示しています。

**-INCLUDE** [appname[/subapp...]/filename[.ext]]

## 説明

### appname[/subapp...]

サーバプロシジャまたはスタイルシートファイルが格納されている Reporting Server 上のアプリケーションディレクトリパスです。このアプリケーションディレクトリパスを指定しない場合、ファイルは Reporting Server のアプリケーションパス上に存在する必要があります。

### filename[.ext]

Reporting Server 上に格納されているプロシジャまたはスタイルシートファイルの名前です。レポートプロシジャ (.fex) を参照する場合、ファイル拡張子の指定はオプションです。スタイルシートファイルを組み込む場合は、.sty 拡張子を指定します。

appname/filename パスタイプを使用しない場合、ファイルシステムパスを使用することができます。

- ❑ UNIX および Windows プラットフォームでは、filename に挿入するプロシジャの絶対パスとファイル名を指定する必要があります。たとえば、/ibi/apps/sales/headings.fex のようにします。

## 例

### TIBCO WebFOCUS Reporting Server プロシジャおよびスタイルシートファイルの挿入

salesrptsetup.fex プロシジャは Reporting Server の ibisamp アプリケーションディレクトリに格納されています。このファイルには、製品売上レポートで共通して使用される 2 つのフィールドを作成する次のコマンドが含まれています。

```
DEFINE FILE GGSales
PROFIT/D12.3= DOLLARS - BUDDOLLARS;
INVENTORY/D12 = UNITS - BUDUNITS;
END
```

salesrptstyle.sty スタイルシートファイルは Reporting Server の ibisamp アプリケーションディレクトリに格納されています。このファイルには、次のスタイルコマンドが含まれています。

```
TYPE=REPORT,COLOR=NAVY,Font='ARIAL',SIZE=9,GRID=OFF,$
TYPE=HEADING,LINE=1,STYLE=BOLD,SIZE=12,JUSTIFY=Center,$
TYPE=TITLE,BACKCOLOR=RGB(45 111 205),COLOR=WHITE,STYLE=UNDERLINE+BOLD,$
TYPE=DATA,BACKCOLOR=(WHITE RGB(235 235 255)),$
TYPE=SUBTOTAL,BACKCOLOR=RGB(163 200 236),STYLE=BOLD,$
```

次のプロシジャは Reporting Server 上に存在し、salesrptsetup.fex プロシジャファイルに salesrptstyle.sty スタイルシートファイルを組み込みます。

```
-INCLUDE ibisamp/salesrptsetup
TABLE FILE GGSales
HEADING
"Regional Inventory and Profit Report"
SUM BUDUNITS UNITS INVENTORY AS 'Inventory'
BUDDOLLARS DOLLARS PROFIT AS 'Profit'
BY REGION
BY CATEGORY
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
-INCLUDE ibisamp/salesrptstyle.sty
ENDSTYLE
END
```

出力結果は次のとおりです。

Regional Inventory and Profit Report							
Region	Category	Budget Units	Unit Sales	Inventory	Budget Dollars	Dollar Sales	Profit
Midwest	Coffee	335526	332777	-2,749	4086032	4178513	92,481.000
	Food	339263	341414	2,151	4220721	4338271	117,550.000
	Gifts	232318	230854	-1,464	2887620	2883881	-3,739.000
Northeast	Coffee	335920	335778	-142	4252462	4164017	-88,445.000
	Food	351431	353368	1,937	4453907	4379994	-73,913.000
	Gifts	227008	227529	521	2870552	2848289	-22,263.000
Southeast	Coffee	355693	350948	-4,745	4431429	4415408	-16,021.000
	Food	351509	349829	-1,680	4409288	4308731	-100,557.000
	Gifts	235045	234455	-590	2967254	2986240	18,986.000
West	Coffee	358784	356763	-2,021	4523963	4473517	-50,446.000
	Food	335361	340234	4,873	4183244	4202337	19,093.000
	Gifts	236636	235042	-1,594	2934306	2977092	42,786.000

## 例 見出しを含むプロシジャファイルの挿入

次のリクエストは、salehead.fex 内に格納されている見出しを組み込みます。

```
TABLE FILE SALES
-INCLUDE SALEHEAD
SUM UNIT_SOLD AND RETURNS AND COMPUTE
DIFF/I5 = UNIT_SOLD-RETURNS;
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
TYPE=HEADING, COLOR=NAVY, FONT=ARIAL, STYLE=BOLD+ITALIC, SIZE=10,
JUSTIFY=CENTER,$
TYPE=TITLE, COLOR=TEAL, STYLE=BOLD, SIZE=10,$
ENDSTYLE
END
```

salehead.fex は Reporting Server のアプリケーションディレクトリに格納されており、次の構文が含まれています。

```
HEADING
"THE ABC CORPORATION"
"RETAIL SALES DIVISION"
"MONTHLY SALES REPORT"
```

下図のように、この見出しはレポートリクエストに追加され、レポート出力に表示されます。

<b>THE ABC CORPORATION</b>		
<b>RETAIL SALES DIVISION</b>		
<b>MONTHLY SALES REPORT</b>		
<u>UNIT SOLD</u>	<u>RETURNS</u>	<u>DIFF</u>
645	58	587

## 構文

**IBFS フルパスによるリポジトリプロシジャまたはスタイルシートファイルの挿入**

次の構文は、IBFS パスを使用して、リポジトリに格納されているレポートプロシジャファイルまたはスタイルシートファイルを別のレポートプロシジャに挿入する方法を示しています。

```
-INCLUDE IBFS:/path/filename
```

説明

```
IBFS:/path/filename
```

ファイル名を含めたレポートプロシジャのフルパス、またはスタイルシートのファイル名と拡張子 (.fex) です。レポートプロシジャ (.fex) を参照する場合、ファイル拡張子の指定はオプションです。

**注意：**この構文は、リポジトリおよび Reporting Server のアプリケーションディレクトリに格納されたレポートプロシジャで使用することができます。

## 例

**IBFS フルパスによるリポジトリプロシジャまたはスタイルシートファイルの挿入**

salesrptsetup.fex プロシジャは、リポジトリの [Public] ワークスペースに存在し、製品売上レポートで共通して使用される 2 つのフィールドを作成する次のコマンドが含まれています。

```
DEFINE FILE GGSales
PROFIT/D12.3= DOLLARS - BUDDOLLARS;
INVENTORY/D12 = UNITS - BUDUNITS;
END
```

salesrptstyle.sty スタイルシートファイルは、[Public] ワークスペースに存在し、次のスタイルコマンドが含まれています。

```
TYPE=REPORT,COLOR=NAVY, FONT='ARIAL', SIZE=9, GRID=OFF, $
TYPE=HEADING, LINE=1, STYLE=BOLD, SIZE=12, JUSTIFY=CENTER, $
TYPE=TITLE, BACKCOLOR=RGB(45 111 205), COLOR=WHITE, STYLE=UNDERLINE+BOLD, $
TYPE=DATA, BACKCOLOR=(WHITE RGB(235 235 255)), $
TYPE=SUBTOTAL, BACKCOLOR=RGB(163 200 236), STYLE=BOLD, $
```

次のプロシジャは、リポジットリプロシジャファイルの salesrptsetup.fex を組み込みます。

```
-INCLUDE IBFS:/WFC/Repository/Public/salesrptsetup.fex
TABLE FILE GGSales
HEADING
"Regional Inventory and Profit Report"
SUM BUDUNITS UNITS INVENTORY AS 'Inventory'
BUDDOLLARS DOLLARS PROFIT AS 'Profit'
BY REGION
BY CATEGORY
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
-INCLUDE IBFS:/WFC/Repository/Public/salesrptstyle.sty
ENDSTYLE
END
```

出力結果は次のとおりです。

Regional Inventory and Profit Report							
Region	Category	Budget Units	Unit Sales	Inventory	Budget Dollars	Dollar Sales	Profit
Midwest	Coffee	335526	332777	-2,749	4086032	4178513	92,481.000
	Food	339263	341414	2,151	4220721	4338271	117,550.000
	Gifts	232318	230854	-1,464	2887620	2883881	-3,739.000
Northeast	Coffee	335920	335778	-142	4252462	4164017	-88,445.000
	Food	351431	353368	1,937	4453907	4379994	-73,913.000
	Gifts	227008	227529	521	2870552	2848289	-22,263.000
Southeast	Coffee	355693	350948	-4,745	4431429	4415408	-16,021.000
	Food	351509	349829	-1,680	4409288	4308731	-100,557.000
	Gifts	235045	234455	-590	2967254	2986240	18,986.000
West	Coffee	358784	356763	-2,021	4523963	4473517	-50,446.000
	Food	335361	340234	4,873	4183244	4202337	19,093.000
	Gifts	236636	235042	-1,594	2934306	2977092	42,786.000

## 構文

### 相対パスによるリポジットリプロシジャまたはスタイルシートファイルの挿入

次の構文は、相対パスを使用して、リポジットリに格納されているレポートプロシジャファイルまたはスタイルシートファイルを別のレポートプロシジャに挿入する方法を示しています。この構文は、挿入するファイルと呼び出し元レポートプロシジャが、同一リポジットリフォルダに存在する場合にのみサポートされます。

```
-INCLUDE ./filename.ext
```

## 説明

`/filename.ext`

挿入するプロシジャ (.fex) またはスタイルシート (.sty) の相対パスを指定します。この構文は、挿入するファイルと呼び出し元プロシジャが同一リポジトリフォルダに存在する場合にのみ使用できます。

**注意：**この構文は、プロシジャを Reporting Server から実行する場合はサポートされません。

## 例

### 相対パスによる見出し付きファイルの挿入

次のプロシジャファイル heading.fex には、以下の見出し構文が含まれています。

```
HEADING
"This is the included heading file"
" "
```

次のプロシジャ include\_heading.fex は、heading.fex と同一のリポジトリフォルダに格納され、-INCLUDE コマンドと相対パスが含まれています。

```
TABLE FILE EMPLOYEE
-INCLUDE ./heading.fex
PRINT CURR_SAL BY LAST_NAME BY FIRST_NAME
WHERE DEPARTMENT EQ 'MIS'
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
TYPE=HEADING, SIZE = 16, STYLE=BOLD,$
END
```

出力結果は次のとおりです。

### **This is the included heading file**

<u>LAST NAME</u>	<u>FIRST NAME</u>	<u>CURR_SAL</u>
BLACKWOOD	ROSEMARIE	\$21,780.00
CROSS	BARBARA	\$27,062.00
GREENSPAN	MARY	\$9,000.00
JONES	DIANE	\$18,480.00
MCCOY	JOHN	\$18,480.00
SMITH	MARY	\$13,200.00

## 構文

**IBFS パスによる TIBCO WebFOCUS Reporting Server プロシジャまたはスタイルシートの挿入**

WebFOCUS IBFS パスを使用して、Reporting Server のアプリケーションディレクトリに存在するプロシジャまたはスタイルシートファイルを別のプロシジャに挿入するには、次のように入力します。

```
-INCLUDE IBFS://EDA/nodename/appname/filename.ext
```

## 説明

**nodename**

WebFOCUS Client 構成で定義された Reporting Server ノードです。IBFS://EDA/ パスは、プロシジャまたはスタイルシートファイルが Reporting Server 上に存在することを示します。

**appname**

サーバプロシジャまたはスタイルシートファイルが格納されている Reporting Server 上のアプリケーションディレクトリパスです。このパスを指定しない場合、ファイルは Reporting Server のアプリケーションパス上に存在する必要があります。

**filename.ext**

Reporting Server 上に格納されているプロシジャまたはスタイルシートファイルの名前です。レポートプロシジャを参照する場合、ファイル拡張子の指定はオプションです。スタイルシートファイルを組み込む場合は、.sty 拡張子を指定します。

**注意：**この構文は、リポジトリおよび Reporting Server のアプリケーションディレクトリに格納されたレポートプロシジャで 사용할 수 있습니다。

## 例

**IBFS パスによる TIBCO WebFOCUS Reporting Server プロシジャおよびスタイルシートファイルの挿入**

salesrptsetup.fex プロシジャは Reporting Server の ibisamp アプリケーションディレクトリに格納されています。このファイルには、製品売上レポートで共通して使用される 2 つのフィールドを作成する次のコマンドが含まれています。

```
DEFINE FILE GGSales
PROFIT/D12.3= DOLLARS - BUDDOLLARS;
INVENTORY/D12 = UNITS - BUDUNITS;
END
```

salesrptstyle.sty スタイルシートファイルは Reporting Server の ibisamp アプリケーションディレクトリに格納されています。このファイルには、次のスタイルコマンドが含まれています。

```
TYPE=REPORT,COLOR=NAVY, FONT='ARIAL', SIZE=9, GRID=OFF, $
TYPE=HEADING, LINE=1, STYLE=BOLD, SIZE=12, JUSTIFY=CENTER, $
TYPE=TITLE, BACKCOLOR=RGB(45 111 205), COLOR=WHITE, STYLE=UNDERLINE+BOLD, $
TYPE=DATA, BACKCOLOR=(WHITE RGB(235 235 255)), $
TYPE=SUBTOTAL, BACKCOLOR=RGB(163 200 236), STYLE=BOLD, $
```

次のプロシジャは、Reporting Server プロシジャ salesrptsetup.fex に Reporting Server スタイルシートファイル salesrptstyle.sty を組み込みます。

```
-INCLUDE IBFS://EDA/EDASERVE/ibisamp/salesrptsetup.fex
TABLE FILE GGSales
HEADING
"Regional Inventory and Profit Report"
SUM BUDUNITS UNITS INVENTORY AS 'Inventory'
BUDDOLLARS DOLLARS PROFIT AS 'Profit'
BY REGION
BY CATEGORY
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
-INCLUDE IBFS://EDA/EDASERVE/ibisamp/salesrptstyle.sty
ENDSTYLE
END
```

出力結果は次のとおりです。

Regional Inventory and Profit Report							
Region	Category	Budget Units	Unit Sales	Inventory	Budget Dollars	Dollar Sales	Profit
Midwest	Coffee	335526	332777	-2,749	4086032	4178513	92,481.000
	Food	339263	341414	2,151	4220721	4338271	117,550.000
	Gifts	232318	230854	-1,464	2887620	2883881	-3,739.000
Northeast	Coffee	335920	335778	-142	4252462	4164017	-88,445.000
	Food	351431	353368	1,937	4453907	4379994	-73,913.000
	Gifts	227008	227529	521	2870552	2848289	-22,263.000
Southeast	Coffee	355693	350948	-4,745	4431429	4415408	-16,021.000
	Food	351509	349829	-1,680	4409288	4308731	-100,557.000
	Gifts	235045	234455	-590	2967254	2986240	18,986.000
West	Coffee	358784	356763	-2,021	4523963	4473517	-50,446.000
	Food	335361	340234	4,873	4183244	4202337	19,093.000
	Gifts	236636	235042	-1,594	2934306	2977092	42,786.000

構文

レガシー BI Portal レポート構文による TIBCO WebFOCUS Reporting Server プロシジャまたはスタイルシートファイルの挿入

WebFOCUS アプリケーション作成では、レポートのスタイル設定と環境の設定 (例、JOIN、DEFINE) を共通のワークスペースに格納し、セキュリティを定義することで、これらを一元的に管理することをお勧めします。セキュリティの定義では、ユーザの読み取り権限および実行権限は許可するが、変更権限は許可しないように設定します。ただし、リポジットのアプリケーションでは、Reporting Server 上のアプリケーションディレクトリに格納されているレポートプロシジャとスタイルシートファイルの参照が必要になる場合があります。

**注意：**このセクションの情報は、マイグレートされたレガシー BI Portal アプリケーションを対象として記述されています。

レガシー BI Portal アプリケーションは、Reporting Server プロシジャまたはスタイルシートファイルを、Reporting Server の -MRNOEDIT コマンドを前置した -INCLUDE 構文によって挿入します。

```
-MRNOEDIT -INCLUDE appname[/subapp...]/filename[.ext]
```

説明

**appname[/subapp...]**

サーバプロシジャまたはスタイルシートファイルが格納されている Reporting Server 上のアプリケーションディレクトリパスです。このアプリケーションディレクトリパスを指定しない場合、ファイルは Reporting Server のアプリケーションパス上に存在する必要があります。

**filename[.ext]**

Reporting Server 上に格納されているプロシジャまたはスタイルシートファイルの名前です。レポートプロシジャ (.fex) を参照する場合、ファイル拡張子の指定はオプションです。スタイルシートファイルを組み合わせる場合は、.sty 拡張子を指定します。

**注意：**このレガシー構文が含まれたプロシジャを Reporting Server のアプリケーションディレクトリに移動する場合、-MRNOEDIT 接頭語を削除する必要があります。

## 例 レガシー BI Portal レポート構文による TIBCO WebFOCUS Reporting Server プロシジャおよびスタイルシートファイルの挿入

以下は、-MRNOEDIT コマンドおよび -INCLUDE コマンドを使用して、Reporting Server プロシジャおよびスタイルシートファイルをリポジトリプロシジャに組み込む例を示しています。

salesrptsetup.fex プロシジャファイルには、製品売上レポートで共通して使用される 2 つのフィールドを作成する、次のコマンドが含まれています。このファイルは、Reporting Server の / ibi/apps/ibisamp ディレクトリに格納されています。

```
DEFINE FILE GGSALLES
PROFIT/D12.3= DOLLARS - BUDDOLLARS;
INVENTORY/D12 = UNITS - BUDUNITS;
END
```

次の salesrptstyle.sty スタイルシートファイルには、スタイルコマンドが記述され、このファイルは Reporting Server 上の /ibi/apps/ibisamp ディレクトリに格納されています。

```
TYPE=REPORT,COLOR=NAVY, FONT='ARIAL', SIZE=9, GRID=OFF, $
TYPE=HEADING, LINE=1, STYLE=BOLD, SIZE=12, JUSTIFY=CENTER, $
TYPE=TITLE, BACKCOLOR=RGB(45 111 205), COLOR=WHITE, STYLE=UNDERLINE+BOLD, $
TYPE=DATA, BACKCOLOR=(WHITE RGB(235 235 255)), $
TYPE=SUBTOTAL, BACKCOLOR=RGB(163 200 236), STYLE=BOLD, $
```

次のリポジトリプロシジャは、salesrptsetup プロシジャファイルと salesrptstyle スタイルシートファイルを組み込みます。これらのファイルは、両方とも Reporting Server 上に存在します。

```
-MRNOEDIT -INCLUDE ibisamp/salesrptsetup
TABLE FILE GGSales
HEADING
"Regional Inventory and Profit Report"
SUM BUDUNITS UNITS INVENTORY AS 'Inventory'
BUDDOLLARS DOLLARS PROFIT AS 'Profit'
BY REGION
BY CATEGORY
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
-MRNOEDIT -INCLUDE ibisamp/salesrptstyle.sty
ENDSTYLE
END
```

出力結果は次のとおりです。

Regional Inventory and Profit Report							
Region	Category	Budget Units	Unit Sales	Inventory	Budget Dollars	Dollar Sales	Profit
Midwest	Coffee	335526	332777	-2,749	4086032	4178513	92,481.000
	Food	339263	341414	2,151	4220721	4338271	117,550.000
	Gifts	232318	230854	-1,464	2887620	2883881	-3,739.000
Northeast	Coffee	335920	335778	-142	4252462	4164017	-88,445.000
	Food	351431	353368	1,937	4453907	4379994	-73,913.000
	Gifts	227008	227529	521	2870552	2848289	-22,263.000
Southeast	Coffee	355693	350948	-4,745	4431429	4415408	-16,021.000
	Food	351509	349829	-1,680	4409288	4308731	-100,557.000
	Gifts	235045	234455	-590	2967254	2986240	18,986.000
West	Coffee	358784	356763	-2,021	4523963	4473517	-50,446.000
	Food	335361	340234	4,873	4183244	4202337	19,093.000
	Gifts	236636	235042	-1,594	2934306	2977092	42,786.000

**注意：**リポジトリプロシジャを Reporting Server のアプリケーションディレクトリに移動する場合、-MRNOEDIT コマンドを削除する必要があります。以下はその例です。

```
-INCLUDE ibisamp/salesrptsetup
```

**構文****-INCLUDE による TIBCO WebFOCUS Reporting Server アプリケーションディレクトリの WebFOCUS リポジトリとのリンク**

WebFOCUS Client REST アダプタを Reporting Server 上で構成した場合、WebFOCUS リポジトリとのリンクを設定したアプリケーションディレクトリを作成し、このアプリケーションディレクトリを -INCLUDE コマンドでサーバパスに含めて使用することができます。

WebFOCUS リポジトリを Reporting Server のアプリケーションディレクトリにリンクする方法についての詳細は、『TIBCO WebFOCUS サーバ管理者ガイド』を参照してください。

次の構文は、WebFOCUS リポジトリにリンクされた Reporting Server 上のアプリケーションディレクトリを使用して、WebFOCUS リポジトリに格納されているレポートプロシジャまたはスタイルシートファイルを挿入する方法を示しています。

**-INCLUDE [wfcapp[/subapp...]/filename[.ext]]**

**説明****wfcapp[/subapp...]**

挿入するファイルが格納されているリポジトリワークスペースです。リンクが設定された Reporting Server のアプリケーションディレクトリ (wfcapp) を使用して参照されます。

**filename[.ext]**

リポジトリパス上に格納されているプロシジャまたはスタイルシートファイルの名前です。レポートプロシジャ (.fex) を参照する場合、ファイル拡張子の指定はオプションです。スタイルシートファイルを組み込む場合は、.sty 拡張子を指定します。

## 例 -INCLUDE で WebFOCUS リポジトリにリンクされた TIBCO WebFOCUS Reporting Server のアプリケーションディレクトリ

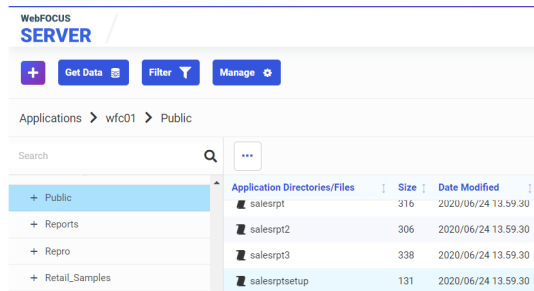
下図のように、Reporting Server には、WebFOCUS リポジトリにリンクされたアプリケーションディレクトリがあります。

The screenshot shows a configuration window for an application. It contains several fields and buttons:

- Application Type:** A text box containing "Application Mapping".
- Physical Location:** A text box containing "WFC:/wfcomputer" with a three-dot menu icon to its right.
- Map to:** A dropdown menu showing "New application (directory will be created)".
- Mapping defined in profile:** A text box containing "EDASPROF".
- APPPATH defined in profile:** A text box containing "EDASPROF".
- Number of files:** A text box containing "112".
- Mixed case support:** A text box containing "DEF".
- Codepage:** A text box containing "Server default".
- Buttons:** "Cancel" and "Save" buttons at the bottom right.

このアプリケーションディレクトリに割り当て済みの名前は wfc01 です。

下図のように、リンクが設定された wfc01 アプリケーションディレクトリ内の「Public」という名前のリポジトリワークスペースには、salesrptsetup.fex ファイルが格納されています。



salesrptsetup.fex ファイルには、製品売上レポートで共通して使用される 2 つのフィールドを作成する、次のコマンドが含まれています。

```
DEFINE FILE GGSales
PROFIT/D12.3= DOLLARS - BUDDOLLARS;
INVENTORY/D12 = UNITS - BUDUNITS;
END
```

Reporting Server 上の次のレポートプロシジャは、リンクが設定されたアプリケーションディレクトリを使用して、salesrptsetup.fex ファイルを組み込みます。

```
-INCLUDE wfc01/Public/salesrptsetup.fex
TABLE FILE GGSales
"Regional Inventory and Profit Report"
SUM BUDUNITS UNITS INVENTORY AS 'Inventory'
BUDDOLLARS DOLLARS PROFIT AS 'Profit'
BY REGION
BY CATEGORY
ON TABLE SET STYLE *
-INCLUDE ibisamp/salesrptsty.sty
ENDSTYLE
END
```

出力結果は次のとおりです。

Regional Inventory and Profit Report							
Region	Category	Budget Units	Unit Sales	Inventory	Budget Dollars	Dollar Sales	Profit
Midwest	Coffee	335526	332777	-2,749	4086032	4178513	92,481.000
	Food	339263	341414	2,151	4220721	4338271	117,550.000
	Gifts	232318	230854	-1,464	2887620	2883881	-3,739.000
Northeast	Coffee	335920	335778	-142	4252462	4164017	-88,445.000
	Food	351431	353368	1,937	4453907	4379994	-73,913.000
	Gifts	227008	227529	521	2870552	2848289	-22,263.000
Southeast	Coffee	355693	350948	-4,745	4431429	4415408	-16,021.000
	Food	351509	349829	-1,680	4409288	4308731	-100,557.000
	Gifts	235045	234455	-590	2967254	2986240	18,986.000
West	Coffee	358784	356763	-2,021	4523963	4473517	-50,446.000
	Food	335361	340234	4,873	4183244	4202337	19,093.000
	Gifts	236636	235042	-1,594	2934306	2977092	42,786.000

## 参照

### TIBCO WebFOCUS ReportCaster で -INCLUDE を使用する際の考慮事項

ReportCaster のスケジュール内の各タスクが Reporting Server で処理される場合、レポート、グラフ、HTML ファイルなど、単一のアンサーセットの取得がサポートされます。そのため、-INCLUDE を使用して特定のプロシジャと ReportCaster でスケジュールするプロシジャを統合する場合、取得するアンサーセットは 1 つにする必要があります。単一ドキュメント内の複数のレポートを配信する必要がある場合は、複合レポート機能を使用します。

### -INCLUDE コマンドでの完全修飾名の使用

ダイアログマネージャの -INCLUDE コマンドは、完全修飾ファイル名を使用して、標準の検索パスに存在しないファイルをプロシジャに挿入することができます。この方法により、特定のプロシジャを検索する時間が短縮され、パフォーマンスが向上します。

Reporting Server 上のプロシジャの -INCLUDE コマンドでは、相対パスはサポートされないため、プラットフォームに固有の完全修飾名を使用する必要があります。プラットフォームに固有の完全修飾ファイル名を使用する場合、その最大長はオペレーティングシステムで定義されます。ただし、プロシジャ行の最大長 32 キロバイト以内であることが条件です。

WebFOCUS リポジトリに格納されたプロシジャでは、相対パス構文 (/filename.ext)を使用して、同一リポジトリフォルダ内のファイルをインクルード (-INCLUDE) することができます。

完全修飾名は、その名前が指定された -INCLUDE コマンドにのみ適用され、他の -INCLUDE コマンドには継承されません。

## 構文

### UNIX での完全修飾ファイル名の使用

```
-INCLUDE /path/filename.ext
```

説明

**path**

プロシジャが格納されたファイルへの完全修飾パスです。

**filename**

プロシジャが格納されたファイルの名前です。

**ext**

プロシジャが格納されたファイル拡張子です。

## 構文

### Windows での完全修飾ファイル名の使用

```
-INCLUDE drive:¥path¥filename.ext
```

説明

**drive**

プロシジャへのパスを含むドライブ名です。

**path**

プロシジャが格納されたファイルへの完全修飾パスです。

**filename**

プロシジャが格納されたファイルの名前です。

**ext**

プロシジャが格納されたファイル拡張子です。

## 例

### -INCLUDE コマンドでの完全修飾名の使用

ここでは、「HEADINGS」というプロシジャに、リクエストで使用する次の見出しテキストが含まれている場合を想定します。

```
HEADING
"THIS IS THE INCLUDED HEADING FILE"
" "
```

Windows の例

```
TABLE FILE EMPLOYEE
-INCLUDE c:¥ibi¥srv82¥mydataarea¥headings.fex
PRINT CURR_SAL BY LAST_NAME BY FIRST_NAME
WHERE DEPARTMENT EQ 'MIS'
END
```

UNIX の例

```
TABLE FILE EMPLOYEE
-INCLUDE /u2/prog/headings.data
PRINT CURR_SAL BY LAST_NAME BY FIRST_NAME
WHERE DEPARTMENT EQ 'MIS'
END
```

出力結果は次のとおりです。

```
THIS IS THE INCLUDED HEADING FILE
LAST_NAME FIRST_NAME CURR_SAL
BLACKWOOD ROSEMARIE $21,780.00
CROSS BARBARA $27,062.00
GREENSPAN MARY $9,000.00
JONES DIANE $18,480.00
MCCOY JOHN $18,480.00
SMITH MARY $13,200.00
```

### -INCLUDE によるプロシジャのネスト化

1 つの呼び出し側プロシジャから、別のプロシジャをいくつでも呼び出すことができます。また、ここで示すように、-INCLUDE コマンドをそれぞれのプロシジャ内でネストすることもできます。ネストできる -INCLUDE の数に制限はありません。

```
-PRODSALES
-INCLUDE FILE1
-RUN
```

```
FILE1
-INCLUDE FILE2
-RUN
```

```
FILE2
-INCLUDE FILE3
-RUN
```

```
FILE3
-INCLUDE FILE4
-RUN
```

```
FILE4
-RUN
```

ファイルの 1 から 4 が、元のプロシジャに組み込まれます。挿入されたすべてのファイルが、元のプロシジャの一部と見なされます。

## EXEC による別プロシジャの呼び出し

EXEC コマンドを使用して、プロシジャから別のプロシジャを呼び出すことができます。呼び出し先のプロシジャは、完全に実行可能である必要があります。呼び出し先のプロシジャは、その内容も含めて完全に別のプロシジャとして機能します。呼び出し先のプロシジャは、呼び出し側のプロシジャで定義されたローカル変数 (&variable) を使用することはできません。なお、コマンドラインで明示的に呼び出し先のプロシジャへ渡される場合は除きます。ただし、実行される (呼び出し先の) プロシジャは、呼び出し側のプロシジャで定義されたグローバル変数 (&&variables) を使用することができます。

EXEC コマンドは検出された段階でスタックされ、スタックコマンドを実行するダイアログマネージャコマンドが見つかった時点で実行されます。呼び出し先のプロシジャは、完全に実行可能である必要があります。

-INCLUDE コマンドで呼び出すプロシジャは、WebFOCUS Reporting Server の検索パスに存在する必要があります。

## 構文 EXEC コマンドによるプロシジャの呼び出し

EX[EC] procedure

説明

procedure

プロシジャの名前です。

**注意:** BI Portal で EXEC を使用する場合、EX と EXEC の処理方法が異なる点に注意する必要があります。プロシジャでコーディングされた EX ステートメントは、WebFOCUS Client で処理されます。WebFOCUS Client は、BI Portal リポジトリ内でこのプロシジャを検索します。

EXEC ステートメントで参照されるプロシジャは、WebFOCUS Client では処理されず、WebFOCUS Reporting Server に渡されて処理されます。これらのプロシジャは、BI Portal リポジトリ内では検索されません。

## 例 EXEC によるプロシジャの呼び出し

次の例は、「DATERPT」というプロシジャを呼び出す方法を示しています。

```
-IF &OPTION EQ 'S' GOTO PRODSALES ELSE GOTO PRODRETURNS;
.
.
.
-PRODRETURNS
 EX DATERPT
.
.
.
-RUN
```

## プロシジャによる HTML Web ページの拡張

ダイアログマネージャの -HTMLFORM コマンドを使用してプロシジャに HTML コマンドを挿入することにより、Web ページの機能を拡張することができます。-HTMLFORM は、標準の HTML 要素をすべてサポートします。これらの要素には、文字のスタイル設定、ハイパーリンク、グラフィックイメージ、テーブル、フォーム、フレームなどがあります。次の方法で、プロシジャと HTML コマンドを組み合わせることができます。

- ❑ レポート出力と結合する Web ページファイルを、プロシジャから参照します。
- ❑ プロシジャに HTML コマンドを埋め込みます。この場合、エディタを使用してプロシジャに構文を入力します。
- ❑ HTML コマンドにプロシジャを埋め込みます。
- ❑ WebFOCUS エスケープシーケンスで指定したローカル変数およびグローバル変数を Web ページに挿入します。

## 外部 Web ページの参照

プロシジャで HTML Web ページを指定することにより、HTML コマンドをプロシジャに挿入することができます。

### 構文      プロシジャからの外部 Web ページの参照

```
-HTMLFORM filename
```

説明

**filename**

ターゲット Web ページが格納された HTML ファイルです。HTML ファイルの名前は、大文字にする必要があります。

- ❑ UNIX および Windows では、Web ページが格納された HTML ファイルの 8 バイトのファイル名です。ファイル拡張子は .HTML ではなく、.HTM にする必要があります。

- ❑ -HTMLFORM で呼び出す HTML ファイルは、WebFOCUS Reporting Server で定義したパスに存在していなければなりません。

外部の HTML ファイルを使用する場合は、そのファイル内でダイアログマネージャコマンドおよび変数を使用することはできません。ただし、WebFOCUS エスケープシーケンスは使用することができます。詳細は、408 ページの「[HTML Web ページへの変数挿入](#)」を参照してください。

## プロシジャへの HTML コマンド埋め込み

プロシジャに HTML コマンドを埋め込むことにより、これらのコマンドをプロシジャに挿入することができます。

**注意：**-RUN コマンドは、-HTMLFORM BEGIN コマンドと -HTMLFORM END コマンドの間の行に挿入することはできません。

## 構文 WebFOCUS プロシジャへの HTML コマンド埋め込み

```
-HTMLFORM BEGIN
 .
 .
 .
-HTMLFORM END
```

説明

**-HTMLFORM BEGIN**  
HTML コマンドの開始を表します。

**-HTMLFORM END**  
HTML コマンドの終了を表します。

**注意：**ダイアログマネージャコマンド以外のすべての行は、HTML と見なされます。

-HTMLFORM BEGIN- と -HTMLFORM END- コマンドの間で参照されているダイアログマネージャ変数は、コメントタグ内に存在する場合でも、すべて評価されます。そのため、各変数にデフォルト値を割り当てるか、アンパサンド (&) が変数名の開始文字として解釈されないようアンパサンド (&) をパイプ文字 (|) でエスケープする必要があります。たとえば、次の例では、アンパサンド (&) がパイプ文字 (|) によってエスケープされているため、コメント内の変数定義は評価されません。

```
-HTMLFORM BEGIN
<HTML>
<HEAD>
<SCRIPT>
/* escaping the ampersand in a comment: &|x=y */
</SCRIPT>
</HEAD>
<BODY>
Test variable in Javascript comments
</BODY>
</HTML>
-HTMLFORM END
```

### 例 プロシジャへの Web ページの挿入

次のプロシジャは、レポートリクエストおよび -HTMLFORM コマンドで構成され、レポートを表示する Web ページをカスタマイズしています。レポートは、Web ブラウザのウィンドウに中央揃えで表示され、青の Arial フォントが使用されます。

```
TABLE FILE SHORT
SUM PROJECTED_RETURN
BY COUNTRY
ON TABLE HOLD AS SHORT FORMAT HTMTABLE
END
```

```
-HTMLFORM BEGIN
<HTML><HEAD><STYLE>TD {FONT-FAMILY: ARIAL; COLOR: BLUE; }</STYLE></HEAD>
<BODY><DIV ALIGN="CENTER">
!IBI.FIL.SHORT;
</DIV></BODY></HTML>
-HTMLFORM END
```

次のような Web ページが作成されます。

PAGE 1	
Country	Projected Annualized Return
ARGENTINA	4.200
BRAZIL	4.200
CANADA	3.990
CZECH REP	12.110
ENGLAND	4.060
FRANCE	3.920
GUATEMALA	4.760
HONDURAS	4.760
HONG KONG	4.550
ISRAEL	3.780
JAPAN	4.550
MEXICO	5.040
SAUDI ARABIA	4.200
SWEDEN	4.040
UNITED STATES	3.430

## HTML Web ページへのプロシジャの埋め込み

-HTMLFORM コマンドを使用して、Web ページに 1 つまたは複数のレポート (HTML ファイル) を埋め込むことができます。

### 手順 HTML Web ページにプロシジャを埋め込むには

-HTMLFORM コマンドを使用して作成した Web ページにレポートを埋め込むには、次の手順を実行します。

1. レポートデータを取得してフォーマットを設定した後、各レポートの出力をそれぞれ異なる一時ファイルに保存するレポートプロシジャを作成します。レポートプロシジャでは、次の構文を使用します。

ON TABLE HOLD FORMAT HTMTABLE AS reportname

**注意：**出力用に作成された一時ファイルは、ユーザが使用することはできません。

2. レポートを埋め込んだ Web ページの表示に外部 HTML ファイルを使用する場合は、レポートプロシジャに次のコマンドを使用してその外部ファイルを指定する必要があります。

-HTMLFORM filename

**注意：**外部 HTML ファイルを使用しない場合、-HTMLFORM BEGIN および -HTMLFORM END コマンドを使用して HTML コマンドをプロシジャの他のコマンドと分離することにより、Web ページ用の HTML コマンドを同一ファイル内に挿入することができます。

3. Web ページ用の HTML コマンドでは、HTML エスケープコードを使用して、それぞれのレポートまたは HTML ファイルを表示する場所を個別に特定します。エスケープコードの構文は次のとおりです。

!IBI.FIL.reportname

説明

reportname

Web ページに埋め込むレポートファイルの名前です。

**注意：**レポートファイル名に使用できる文字数は最長 512 バイトです。

!IBI.FIL エスケープシーケンスを使用して埋め込んだファイルには、他のエスケープシーケンスを含めることができます。存在しないファイルを参照した場合は、エラーメッセージが返されます。

## 例

### 単一 HTML Web ページへの複数レポートの埋め込み

この例は、1 つの HTML Web ページに複数の HTML レポートを埋め込む方法を示しています。次のコードは、自動車に関する同一のデータを、国別、車体別、メーカー別という 3 種類の方法でソートして、3 つのレポートを生成します。

```
TABLE FILE CAR
"REPORT 1 - BY COUNTRY"
PRINT CAR MODEL BODYTYPE
BY COUNTRY
ON TABLE HOLD FORMAT HTMTABLE AS REPORT1
ON TABLE SET STYLE *
TYPE=HEADING,STYLE=BOLD,COLOR=RED,$
ENDSTYLE
END
TABLE FILE CAR
"REPORT 2 - BY BODY TYPE"
PRINT CAR MODEL COUNTRY
BY BODYTYPE
ON TABLE HOLD FORMAT HTMTABLE AS REPORT2
ON TABLE SET STYLE *
TYPE=HEADING,STYLE=BOLD,COLOR=BLUE,$
ENDSTYLE
END
TABLE FILE CAR
"REPORT 3 - BY MANUFACTURER"
PRINT MODEL BODYTYPE COUNTRY
BY CAR
ON TABLE HOLD FORMAT HTMTABLE AS REPORT3
ON TABLE SET STYLE *
TYPE=HEADING,STYLE=BOLD,COLOR=GREEN,$
ENDSTYLE
END
-HTMLFORM BEGIN
!IBI.FIL.REPORT1;
!IBI.FIL.REPORT2;
!IBI.FIL.REPORT3;
-HTMLFORM END
```

下図は、1 つの HTML ドキュメントに埋め込まれた 3 つのレポートを示しています。

PAGE 1			
REPORT 1 - BY COUNTRY			
COUNTRY	CAR	MODEL	BODYTYPE
ENGLAND	JAGUAR	V12XKE AUTO	CONVERTIBLE
	JAGUAR	XJ12L AUTO	SEDAN
	JENSEN	INTERCEPTOR III	SEDAN
	TRIUMPH	TR7	HARDTOP
FRANCE	PEUGEOT	504 4 DOOR	SEDAN
ITALY	ALFA ROMEO	2000 4 DOOR BERLINA	SEDAN
	ALFA ROMEO	2000 GT VELOCE	COUPE
	ALFA ROMEO	2000 SPIDER VELOCE	ROADSTER
	MASERATI	DORA 2 DOOR	COUPE
JAPAN	DATSUN	B210 2 DOOR AUTO	SEDAN
	TOYOTA	COROLLA 4 DOOR DIX AUTO	SEDAN
W GERMANY	AUDI	100 LS 2 DOOR AUTO	SEDAN
	BMW	2002 2 DOOR	SEDAN
	BMW	2002 2 DOOR AUTO	SEDAN
	BMW	3.0 SI 4 DOOR	SEDAN
	BMW	3.0 SI 4 DOOR AUTO	SEDAN
	BMW	530I 4 DOOR	SEDAN
	BMW	530I 4 DOOR AUTO	SEDAN

PAGE 1			
REPORT 2 - BY BODY TYPE			
BODYTYPE	CAR	MODEL	COUNTRY
CONVERTIBLE	JAGUAR	V12XKE AUTO	ENGLAND
COUPE	ALFA ROMEO	2000 GT VELOCE	ITALY
	MASERATI	DORA 2 DOOR	ITALY
HARDTOP	TRIUMPH	TR7	ENGLAND
ROADSTER	ALFA ROMEO	2000 SPIDER VELOCE	ITALY
SEDAN	JAGUAR	XJ12L AUTO	ENGLAND
	JENSEN	INTERCEPTOR III	ENGLAND
	DATSUN	B210 2 DOOR AUTO	JAPAN
	TOYOTA	COROLLA 4 DOOR DIX AUTO	JAPAN
	ALFA ROMEO	2000 4 DOOR BERLINA	ITALY
	AUDI	100 LS 2 DOOR AUTO	W GERMANY
	BMW	2002 2 DOOR	W GERMANY
	BMW	2002 2 DOOR AUTO	W GERMANY
	BMW	3.0 SI 4 DOOR	W GERMANY
	BMW	3.0 SI 4 DOOR AUTO	W GERMANY
	BMW	530I 4 DOOR	W GERMANY
	BMW	530I 4 DOOR AUTO	W GERMANY
	PEUGEOT	504 4 DOOR	FRANCE

### 単一 HTML Web ページへの変数の埋め込み

この例では、製品コードと数量を含むレポートを生成し、ダイアログマネージャのシステム変数にある現在の時刻と日付を表示します。次の例は、このレポートの表示に必要な要素を組み合わせる方法を示しています。この例の左側の番号は、そのコードの説明に対応しています。

#### 手順 1 - First.htm

レポートを挿入する Web ページを作成します。この Web ページは、WebFOCUS が格納先を特定できるように、EDAPATH または APPPATH で指定したディレクトリに保存する必要があります。このファイルを first.htm とします。

```
<HTML>
<BODY>
2. Time:
3. !IBI.FIL.&TOD

2. Date:
3. !IBI.FIL.&DATE

<HR SIZE=5>
4. <!--WEBFOCUS TABLE UPPER-->
</BODY>
</HTML>
```

**注意：**HTML コメント行は「->」で閉じる必要があります。また、単一の「>」で閉じることができませんが、その場合はコメント行に他の HTML タグを含めることはできません。

#### 手順 2 - Amper.fex

次のようなレポートリクエストを作成します。

```
TABLE FILE GGORDER
SUM QUANTITY BY PCD
IF PCD EQ 'B$$$$'
1. ON TABLE HOLD FORMAT HTMTABLE AS UPPER
END
5. -RUN
6. -HTMLFORM first
```

#### 手順 3 - Launch.htm

次のように、レポートプロシジャを起動する Web ページを作成します。

```
<HTML>
<BODY>
<P>
<P>
Click here.
</BODY>
</HTML>
```

プロシジャおよび 2 つの Web ページを作成した後、プロシジャを起動する Web ページをブラウザから呼び出すことができます。プロシジャを起動すると、レポートが実行されます。

1. HOLD コマンドは、レポート出力を一時ファイルに抽出します。この一時ファイルから、レポート出力を Web ページに送信することができます。
2. Time: および Date: の各コードは、Web ページのレポートの上部に表示するテキストを指定します。
3. WebFOCUS が読み取るこの HTML コメントは、Web ページに出力する変数を識別します。
4. WebFOCUS が読み取るこの HTML コメントは、表示するレポートを識別します。このテーブル名は、プロシジャの HOLD コマンドで割り当てられた名前です。Web ページでは、WebFOCUS により、それぞれのコードが対応するレポートに置換されます。
5. -RUN コマンドは、レポートを実行します。
6. ダイアログマネージャコマンドは、レポート出力を Web ページが格納された HTML ファイルである first.htm のコンテンツと結合します。

次のような Web ページが作成されます。

Time: 10.29.36

Date: 08/30/01

PAGE 1	
Product Code	Ordered Units
B141	100427
B142	285689
B144	61498

### HTML Web ページへの変数挿入

WebFOCUS では、ダイアログマネージャの変数を使用して、Web ページに変数の値を挿入することができます。この場合、WebFOCUS のエスケープコードを使用して変数名を指定します。変数の値は、HTML Web ページのテキスト、または HTML フォームの入力フィールドの値として使用することができます。ユーザが作成した変数、または &DATE (今日の日付) や &TOD (時刻) などのシステム変数を挿入することができます。1 つの Web ページに、複数の変数置換タグを含めることができます。

-HTMLFORM コマンドを使用して作成した Web ページで変数を使用する場合、-SET または -DEFAULT コマンドでその変数を最初に宣言する必要があります。変数の使用方法についての詳細は、314 ページの「[変数によるプロシジャのカスタマイズ](#)」を参照してください。

## 構文 HTML Web ページ変数値の取得

```
<!--WEBFOCUS VAR [&]&variable-->
```

または

```
!IBI.{AMP|GLB}. [&]&variable;
```

説明

**AMP**

ローカル変数の値を返します。

**GLB**

グローバル変数の値を返します。

**variable**

ローカル変数またはグローバル変数です。

**注意：**HTML コメント行は「->」で閉じる必要があります。また、単一の「>」で閉じることもできますが、その場合はコメント行に他の HTML タグを含めることはできません。

## 構文 HTML Web ページ変数長の取得

```
!IBI.{AML|GLL}. [&]&variable;
```

説明

**AML**

ローカル変数の長さを返します。

**GLL**

グローバル変数の長さを返します。

## 構文 Web ページでの変数の使用

フォームの入力フィールドで使用するローカル変数の構文は次のとおりです。

```
<INPUT TYPE=TEXT NAME=variable VALUE="!IBI.AMP.variable;">
```

説明

**variable**

変数名です。

値を代入する際には、感嘆符 (!) からセミコロン (;) までの文字列全体が置換されます。変数の値がコマンドストリームに内部エスケープシーケンスを含む場合、その変数を指定することはできません。

## オペレーティングシステムコマンドの発行

オペレーティングシステムコマンドを発行して、リクエストを実行する環境をセットアップすることができます。たとえば、リクエストを実行する前に、プログラムによるディレクトリの変更、ファイル名の変更、圧縮ファイルの解凍、ファイルのコピー、およびその他の操作を行うことができます。環境のカスタマイズが完了後、プログラムはレポートおよびグラフィックエラストを開始することができます。

## 構文 オペレーティングシステムコマンドの実行

`op_system command`

説明

`op_system`

オペレーティングシステムを指定します。

`-DOS` または `DOS` - DOS オペレーティングシステムを指定します。

`-UNIX` または `UNIX` - UNIX オペレーティングシステムを指定します。

`-WINNT` または `WINNT` - Windows オペレーティングシステムを指定します。

`-SYSTEM!` - 任意のオペレーティングシステムを指定します。

`command`

オペレーティングシステムコマンドです。

注意

- ❑ オペレーティングシステムコマンド先頭のハイフン (-) の有無の違いは、先頭にハイフン (-) を付けた場合、強制実行するための `-RUN` を使用せずに、一連のダイアログマネージャコマンド内でオペレーティングシステムコマンドを実行できる点です。また、先頭にハイフン (-) を付けたオペレーティングシステムコマンドは、そのコマンドがオペレーティングシステムで実際に実行されない場合、無視 (バイパス) されます。たとえば、アプリケーションを複数の異なるオペレーティングシステムで実行し、現在のディレクトリを表示する場合 (オペレーティングシステムごとに異なるコマンドの実行が必要)、単純に次のコードを使用することができます。

```
-UNIX pwd
-WINNT chdir
-VMS SHOW DEFAULT
-AS/400 pwd
```

- オペレーティングシステムコマンドの `-SYSTEM` および `!` は、使用中のオペレーティングシステムを確認しないという利点があるため、複数のオペレーティングシステムで特定のコマンドが共通している場合、オペレーティングシステムコマンドの `-SYSTEM` または `!` を使用することでコーディング作業が簡素化されます。

## ディレクトリ変更のオペレーティングシステムコマンド

すべてのプラットフォームで使用されるディレクトリ変更コマンドは `cd` コマンドですが、2 つ目のリクエストで `-UNIX pwd` コマンドを発行した場合、ディレクトリ変更コマンドが予期したとおりに動作しない場合があります。その一例として、`-UNIX cd /tmp` コマンドを発行した場合、このコマンドが無視されます。オペレーティングシステムコマンドは別の処理として動作するため、処理が終了すると、変数などの環境プロパティが存在しなくなり、ディレクトリの変更は取り消されます。この動作は、処理の実行を別の処理から保護する、サーバ環境の非永続動作に整合しています。

この動作のため、ディレクトリ変更コマンドは通常、複数のオペレーティングシステムコマンドを単一行で実行するリクエストの一部として使用します。最初に `cd` コマンドを発行し、次に別のコマンドを発行することで、これらのコマンドが 1 つのグループとして実行されます。バージョン 7.1 SP05 より前のバージョンから移植されたアプリケーション、または FOCUS から移植されたアプリケーションは、永続動作に基づいてコーディングされている場合があり、その場合はディレクトリ変更コマンドとオペレーティングシステムコマンドが単一行に統合されるようコーディングする必要があります。

## 単一行の複数オペレーティングシステムコマンドの実行

ほとんどのオペレーティングシステムでは、特別な区切り文字または囲み文字の構文を使用して複数のコマンドを単一行で実行することができます。この方法は、オペレーティングシステムによって異なります。次の例は、複数のオペレーティングシステムコマンドを単一行で実行（スタッキング）するために必要な構文を示しています。

Windows の場合

```
! cmd /c "cd %tmp && chdir && if exist mydata.txt echo got it"
! cmd /c "cd %tmp & chdir & if exist mydata.txt echo got it "
! cd %tmp & chdir & if exist mydata.txt echo got it
! dir mydata & if exist mydata.txt echo got it
```

UNIX の場合

```
! cd /tmp ; pwd ; if [-f mydata.txt] ; then echo got it ; fi
! ls mydata.txt ; if [-f mydata.txt] ; then echo got it ; fi
```

### コマンド出力の確認

オペレーティングシステムコマンドの中には、関数を実行後に出力結果を表示しないものがあります。たとえば、次のコマンドは、コマンド実行後に出力結果を表示しません。

```
-UNIX cp a.fex b.fex
-DOS copy a.fex b.fex
```

他のコマンドは、コマンド実行時に出力を返します。たとえば、次のコマンドは、現在のディレクトリ内で末尾が `fex` のファイルのリストを表示します。

```
-UNIX ls -l grep fex
-DOS dir *.fex
```

WebFOCUS は、オペレーティングシステムコマンドの結果をブラウザに表示しません。オペレーティングシステムコマンドの結果を確認するには、出力を別のファイルにリダイレクトする必要があります。

たとえば、次のコマンドは、コマンドの出力結果を `OUTPUT` というファイルに格納します。出力結果は、後からエディタで表示することができます。

```
-UNIX ls -l grep fex > /tmp/output
-DOS dir *.fex > c:\tmp\output
```

## ダイアログマネージャによるパスワードの管理

`-PASS` コマンドを使用して、パスワードの発行および管理を行うことができます。特に、特定のユーザが読み取りまたは書き込みできるファイルやファイルセットを指定する場合に役立ちます。パスワードには、`DBA` 機能を使用して関連付けられた詳細機能が一式備わっています。

パスワードを設定するプロシジャは、暗号化することができます。これにより、プロシジャ自体および設定するパスワードが入力されたり、他人に知られたりすることを防止できます。

変数を `-PASS` に関連付けることにより、パスワード値の入力を要求したり、パスワード値を割り当てたりすることが可能になります。

### 構文      パスワードの指定

```
-PASS password
```

説明

`password`

パスワードまたはパスワードを格納する変数です。

## アプリケーションへのメッセージの送信

-TYPE コマンドを使用して、プロシジャの処理が進行中にアプリケーションにメッセージを送信することができます。メッセージは次の目的に使用します。

- ❑ プロシジャの目的を表示する。
- ❑ プロシジャの結果またはプロシジャのテスト中の計算結果を表示する。
- ❑ 必要な情報を表示する。
- ❑ CGI 変数を WebFOCUS に渡す。詳細は、262 ページの「[複数レポートの制御](#)」を参照してください。

WebFOCUS では、実行するプロシジャに基づいて、メッセージをブラウザに表示したり、HTML ファイルのコメントとして表示したりすることができます。HTML ファイルのコメントを表示するには、たとえば Microsoft Internet Explorer で [表示] の [ソース] を選択するなど、Web ブラウザの所定のオプションを使用します。

## 構文

### メッセージの送信

`-TYPE text-label TYPE text`

または

`-label TYPE text`

説明

`text`

送信するメッセージです。テキストを引用符で囲んだ場合、その引用符もメッセージの一部として表示されます。メッセージの長さは、最大で 495 バイトです。

WebFOCUS では、メッセージは HTML ソースファイルに挿入されます。

`-label`

-GOTO または -IF のターゲットです。

**注意：**label はダイアログマネージャ専用のコマンドで、他のダイアログマネージャコマンドと同一行にのみ配置することができます。

## 例 メッセージの送信

次の例は、-TYPE を使用してクライアントアプリケーションにレポートの内容を通知する方法を示しています。

```
-* Version 1 06/26/00 SLRPT Procedure
-* Component of Retail Sales Reporting Module
-TYPE This report calculates percentage of returns.
TABLE FILE SALES
.
.
.
END
```

## ダイアログマネージャプロシジャのテストとデバッグ

次の方法で、プロシジャのテストとデバッグを行うことができます。

❑ &ECHO 変数は、コマンド行を実行する際に、行の表示を制御します。これにより、プロシジャのテストおよびデバッグを行うことができます。

❑ &STACK 変数を使用して、ダイアログマネージャコマンドのロジックをテストすることができます。この変数を OFF に設定すると、スタックコマンドを実行せずに、プロシジャを実行することができます。これにより、一連のコマンドを表示し、変数値の処理状況を確認することができます。

❑ &RETCODE 変数は、プロシジャの実行後にコードを返します。プロシジャの実行後、正常な出力結果を得られた場合またはレコードが 1 つも取得されなかった場合、&RETCODE の値は 1 になります。プロシジャの解析中にエラーが発生した場合、&RETCODE の値は 8 になります。

&RETCODE を使用して、オペレーティングシステムコマンドの結果をテストすることができます。この変数は、オペレーティングシステムからコードを返します。

❑ &IORETURN 変数は、ダイアログマネージャの -READ および -WRITE コマンドの結果をテストします。-READ または -WRITE の処理後に返された 0 (ゼロ) 以外のリターンコードは、EOF に到達した場合などのエラーが発生したことを表します。

&IORETURN を使用して、次の結果をテストすることができます。

❑ -READ コマンド。&IORETURN が 0 (ゼロ) の場合、外部ファイルから値が正しく読み取られたことを表します。

❑ -WRITE コマンド。&IORETURN が 0 (ゼロ) の場合、外部ファイルに値が正しく書き込まれたことを表します。

- -TYPE コマンドは、アプリケーションにメッセージを書き込みます。-TYPE コマンドを追加することで、複雑なアプリケーションのデバッグが可能になります。-TYPE コマンドは、プロシジャ全体にメッセージを書き込み、プロシジャが正しく処理された領域を特定することができます。プロシジャのデバッグ処理以外では、これらのメッセージの表示が必要ない場合があります。

-TYPE コマンドの出力を省略する場合は、直接 -TYPE コマンドを使用せずに、-TYPE コマンドのコメントを表示または省略するダイアログマネージャ変数を使用する必要があります。このダイアログマネージャ変数は、-TYPE (コメント行を表示) または -\*TYPE (コメントとして処理されるが、コメント行は非表示) として評価されます。以下はその例です。

```
-DEFAULT &SHOWTYPE='NO'
-SET &TYPE = IF &SHOWTYPE= NO THEN '-*TYPE' ELSE '-TYPE';
&TYPE.EVAL MY MESSAGE
```

&SHOWTYPE を YES (または、NO 以外のその他の設定) に設定してクエリを実行した場合、&TYPE.EVAL が -TYPE として評価されます。そうでない場合は -\*TYPE として評価され、コメントとして処理されるが、表示はされません。

## 構文      プロシジャのテストとデバッグ

```
{-DEFAULT|-SET|EX} &ECHO = {ON|ALL|OFF|NONE}
```

### 説明

#### ON

実行するために展開、スタックされた WebFOCUS コマンドを表示します。

#### ALL

実行するために展開、スタックされたダイアログマネージャコマンドおよび WebFOCUS コマンドを表示します。

#### OFF

スタックコマンドとダイアログマネージャコマンドの両方を非表示にします。これがデフォルト値です。

#### NONE

プロシジャコードが非表示になります。&ECHO の値を NONE に設定した場合、セッション中および接続中に値を変更することはできません。

デフォルト設定では、明示的に &ECHO 変数を設定しないプロシジャは、この値が OFF で実行されます。SET DEFECHO コマンドを使用して &ECHO のデフォルト値を変更することができます。詳細は、416 ページの「[&ECHO 変数デフォルト値の設定](#)」を参照してください。

## 構文 **&ECHO 変数デフォルト値の設定**

**SET DEFECHO = {OFF|ON|ALL|NONE}**

### 説明

#### OFF

&ECHO のデフォルト値として OFF を設定します。デフォルト値は OFF です。

#### ON

&ECHO のデフォルト値として ON を設定します。実行するために展開、スタックされた WebFOCUS コマンドを表示します。

#### ALL

&ECHO のデフォルト値として ALL を設定します。実行するために展開、スタックされたダイアログマネージャコマンドおよび WebFOCUS コマンドを表示します。

#### NONE

プロシジャコードが非表示になります。DEFECHO または &ECHO を NONE に設定した場合、セッション中または接続中に変更することはできません。

## 参照 **SET DEFECHO = NONE 使用上の注意**

- ❑ プロシジャで SET DEFECHO=NONE コマンドを発行する場合、設定はそのルーチンの &ECHO には影響しません。このコマンドは、次回実行されるプロシジャの &ECHO の値として機能し、その後で変更することはできません。
- ❑ NONE 値の有効期間内に &ECHO のリセットを試みた場合、-TYPE コマンドを発行すると、設定しようとした値が表示されますが、値は実際には変更されません。

## 例 **プロシジャコードの非表示**

次のプロシジャにより、DEFECHO パラメータの値が検索され、EMPLOYEE データソースに対して TABLE リクエストが発行されます。

```
? SET DEFECHO
-RUN
- TYPE ECHO = &ECHO
TABLE FILE EMPLOYEE
PRINT CURR_SAL CURR_JOBCODE
BY LAST_NAME BY FIRST_NAME
END
-RUN
```

クエリコマンド出力は、DEFECHO が OFF (デフォルト値) であることを示しています。

```
DEFECHO OFF
```

-TYPE コマンドは、&ECHO の値が OFF (デフォルト値) であることを示しています。

```
ECHO = OFF
```

&ECHO が OFF のため、プロシジャの実行時に、TABLE コマンドは表示されません。

```
テーブルのレコード数 = 12 行数 = 12
PAUSE.. PLEASE ISSUE CARRIAGE RETURN WHEN READY
PAGE 1
LAST_NAME FIRST_NAME CURR_SAL CURR_JOBCODE

BANNING JOHN $29,700.00 A17
BLACKWOOD ROSEMARIE $21,780.00 B04
CROSS BARBARA $27,062.00 A17
GREENSPAN MARY $9,000.00 A07
IRVING JOAN $26,862.00 A15
JONES DIANE $18,480.00 B03
MCCOY JOHN $18,480.00 B02
MCKNIGHT ROGER $16,100.00 B02
ROMANS ANTHONY $21,120.00 B04
SMITH MARY $13,200.00 B14
STEVENS RICHARD $9,500.00 A01
 ALFRED $11,000.00 A07
 END OF REPORT
```

DEFECHO=ON に設定し、プロシジャを再度実行します。

クエリコマンド出力は、DEFECHO が ON であることを示しています。

```
DEFECHO ON
```

-TYPE コマンドは、&ECHO の値が ON に変更されたことを示しています。

```
ECHO = ON
```

&ECHO が ON のため、プロシジャの実行時に、TABLE コマンドが表示されます。

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

出力には、次のように表示されます。

```
テーブルのレコード数 = 12 行数 = 12
PAUSE.. PLEASE ISSUE CARRIAGE RETURN WHEN READY
PAGE 1
LAST_NAME FIRST_NAME CURR_SAL CURR_JOBCODE

BANNING JOHN $29,700.00 A17
BLACKWOOD ROSEMARIE $21,780.00 B04
CROSS BARBARA $27,062.00 A17
GREENSPAN MARY $9,000.00 A07
IRVING JOAN $26,862.00 A15
JONES DIANE $18,480.00 B03
MCCOY JOHN $18,480.00 B02
MCKNIGHT ROGER $16,100.00 B02
ROMANS ANTHONY $21,120.00 B04
SMITH MARY $13,200.00 B14
 RICHARD $9,500.00 A01
STEVENS ALFRED $11,000.00 A07
 END OF REPORT
```

さらに、SET DEFECHO = NONE コマンドを発行し、プロシジャを再度実行します。

```
SET DEFECHO = NONE
```

クエリコマンド出力は、DEFECHO の値が NONE に変更されたことを示しています。

```
DEFECHO NONE
```

-TYPE コマンドは、&ECHO の値が NONE であることを示しています。

```
ECHO = NONE
```

DEFECHO の値が NONE であることから、プロシジャの実行時に、TABLE コマンドは表示されません。出力結果は次のとおりです。

```

 テーブルのレコード数 = 12 行数 = 12
PAUSE.. PLEASE ISSUE CARRIAGE RETURN WHEN READY
PAGE 1
LAST_NAME FIRST_NAME CURR_SAL CURR_JOBCODE

BANNING JOHN $29,700.00 A17
BLACKWOOD ROSEMARIE $21,780.00 B04
CROSS BARBARA $27,062.00 A17
GREENSPAN MARY $9,000.00 A07
IRVING JOAN $26,862.00 A15
JONES DIANE $18,480.00 B03
MCCOY JOHN $18,480.00 B02
MCKNIGHT ROGER $16,100.00 B02
ROMANS ANTHONY $21,120.00 B04
SMITH MARY $13,200.00 B14
 RICHARD $9,500.00 A01
STEVENS ALFRED $11,000.00 A07
 END OF REPORT

```

一度 NONE に設定した DEFECHO の値を変更することはできません。次の SET コマンドは、値の「ON」への変更を試みますが、クエリコマンド出力は、引き続き「NONE」であることを示しています。

```

SET DEFECHO=ON
? SET DEFECHO
DEFECHO NONE

```

構文      ダイアログマネージャコマンドロジックのテスト

{-DEFAULT|-SET|EX} &STACK = {ON|OFF}

説明

ON

スタックコマンドを通常どおり実行します。これがデフォルト値です。

OFF

スタックコマンドを実行しません。また、&RECORDS や &LINES などのシステム変数も設定されません。ダイアログマネージャコマンドは実行されるため、プロシジャのロジックをテストすることができます。

例      &RETCODE 変数によるコマンド結果のテスト

次のコマンドは、C:¥MYDATA¥WEEK27.DAT ファイルの存在をテストします。このファイルが存在する場合、&RETCODE の値が 0 (ゼロ) となり、ファイルは消去されます。

```
DOS STATE C:¥MYDATA¥WEEK27.DAT
-IF &RETCODE NE 0 GOTO START;
DOS ERASE C:¥MYDATA¥WEEK27.DAT
-START
```

## アプリケーションデバッグメッセージの表示

メッセージビューアを使用すると、エラーメッセージ、情報メッセージ、ダイアログマネージャのコマンド行 (例、-TYPE、-DEFAULT、-SET &ECHO) などのメッセージを表示することができます。これらのメッセージは、レポート出力の下部に独立したフレームとして表示され、アプリケーションのデバッグに利用されます。

### 構文      メッセージビューアの使用

WebFOCUS を呼び出す URL に、次の WebFOCUS Servlet 変数を追加します。

`IBIWF_msgviewer=option`

説明

`option`

次のいずれかです。

**OFF**

メッセージビューアを無効にします。このオプションを無効に設定した場合でも、ブラウザのソース表示オプションを使用することにより、HTML レポートの情報メッセージやエラーメッセージを表示することができます。

**ON**

レポート出力下部の独立したフレームにメッセージを表示します。

**ECHOON**

レポート出力下部の独立したフレームにメッセージを表示するほか、実行するために展開およびスタックされたコマンド行を表示します。

**ECHOALL**

レポート出力下部の独立したフレームにメッセージを表示するほか、実行するために展開およびスタックされたコマンド行およびダイアログマネージャのコマンドをすべて表示します。

**注意：**ON、ECHOON、ECHOALL の各オプションは、HTML および PDF を含むすべての出力フォーマットで使用することができます。

### 例      メッセージビューアの使用

次の例は、WebFOCUS セルフサービスアプリケーションでメッセージビューアを使用する方法を示しています。以下は、ECHOALL オプションを指定したパラメータの例です。

[http://srv:port/ibi\\_apps/WFServlet?IBIF\\_ex=region&IBIWF\\_msgviewer=ECHOALL](http://srv:port/ibi_apps/WFServlet?IBIF_ex=region&IBIWF_msgviewer=ECHOALL)

以下は、region.fex リクエストの実行に関連するメッセージビューアの出力を示しています。

```
-INCLUDE region
-DEFAULT &VALUE = 'Northeast';
TABLE FILE GGSales
SUM UNITS
BY REGION
WHERE REGION = 'Northeast'
END
0 NUMBER OF RECORDS IN TABLE= 1080 LINES= 1
0 HOLDING HTML FILE ON PC DISK ...
```

## ダイアログマネージャ構文リファレンス

ここでは、ダイアログマネージャのすべてのコマンドについて説明します。

### \* コマンド

\* は、コメント行の開始を示します。

必要に応じて任意の数のコメント行を追加することができますが、それぞれの行は \* で開始する必要があります。コメント行は、プロシジャの先頭または末尾、あるいはコマンドとコマンドの間に配置することができます。ただし、コマンドと同一の行に配置することはできません。

コメント行に情報を記述して、プロシジャの目的および履歴を他のユーザと共有することができます。

構文は次のとおりです。

```
-* text
```

説明

text

コメントです。\* と text の間にブランクを入れる必要はありません。

### ? コマンド

? は、ローカル変数の現在の値を表示します。

構文は次のとおりです。

```
-? &[string]
```

#### 説明

##### `source_string`

変数名です。このパラメータを指定しない場合は、ローカル変数、グローバル変数、システム変数、統計変数の現在の値がすべて表示されます。ただし、-DEFAULTH コマンドを使用して値が非表示にされている変数を除きます。

### -CLOSE

-CLOSE は、-READ または -WRITE コマンドで NOCLOSE オプションを使用して開いた外部ファイルを閉じます。NOCLOSE は、-READ または -WRITE 処理が完了するまでファイルを開いた状態にするオプションです。

構文は次のとおりです。

##### `-CLOSE filename`

#### 説明

##### `filename`

オペレーティングシステムが認識できる物理ファイルに関連付けられた識別名です。

### -DEFAULT

-DEFAULT コマンドは、ローカル変数またはグローバル変数のデフォルト値を設定します。ストアドプロシジャ内の変数にデフォルト値を指定することにより、そのプロシジャを常時正しく実行させることができます。

1 つの変数に対して複数の -DEFAULT コマンドを発行することができます。グローバル変数の場合は、これらの -DEFAULT コマンドをそれぞれ異なるプロシジャで発行することができます。この変数の値を他の方法で設定しない限り、最後に発行した -DEFAULT コマンドが引き続き有効になります。

ただし、-SET コマンドの発行、ユーザが入力した値の取得、ファイルからの値の読み取りなどの方法で変数の値を設定すると、同一変数に対してそれ以降に発行した -DEFAULT コマンドは無視されます。

構文は次のとおりです。

##### `-DEFAULT [&]name=value [...]`

#### 説明

##### `name`

変数名です。

**value**

変数に割り当てるデフォルト値です。

**-DEFAULTH**

-DEFAULTH コマンドは、隠し変数にデフォルト値を設定します。

-DEFAULTH コマンドを使用することにより、変数を初期化し、その変数を WebFOCUS パラメータのプロンプトで使用されることを回避することができます。-DEFAULTH で初期化された変数は、パラメータのプロンプトに使用する XML 記述情報には返されません。これらの変数は、パラメータプロンプト機能に表示されず、-? コマンドでも表示されないため、ユーザには非表示になります。

構文は次のとおりです。

```
-DEFAULTH &[&]name=value [...]
```

**説明****name**

変数名です。この変数は、パラメータのプロンプトに使用されず、ユーザには非表示になります。

**value**

変数に割り当てるデフォルト値です。

**-DOS**

-DOS は、プロシジャから DOS オペレーティングシステムコマンドを実行します。

構文は次のとおりです。

```
-DOS command
```

**説明****command**

DOS コマンドです。

**-EXIT**

-EXIT は、プロシジャを強制終了します。すべてのスタックコマンドが実行されて、プロシジャが終了します。別のプロシジャから呼び出されたプロシジャの場合、呼び出し側のプロシジャは処理を続行します。

タスクを完了する最終分岐を処理した後、プロシジャを終了するには -EXIT を使用します。プロシジャの最終行は、暗黙的な -EXIT になります。

構文は次のとおりです。

**-EXIT**

**注意：**プロシジャにグローバル変数が含まれており、-EXIT コマンドで終了する場合、グローバル変数の値は記録されません。このため、後続のプロシジャの実行を、このグローバル変数の値に依存させることはできません。

## **-GOTO**

-GOTO は、指定されたラベルに制御を転送します。

ダイアログマネージャがラベルを検出すると、これに続く行で処理が続けられます。ダイアログマネージャがラベルを見つけられない場合は、処理が終了してエラーメッセージが表示されます。

構文は次のとおりです。

```
-GOTO label
 :
 :
-label [TYPE text]
```

説明

**label**

-GOTO アクションのターゲットを指定するユーザ定義の名前です。この名前の最大長は、64 バイトです。

この名前にブランクを含めたり、-QUIT および -EXIT 以外のダイアログマネージャコマンドを使用したりすることはできません。関数、数値演算や論理演算などと混同しやすい語句は使用しないでください。

**TYPE text**

クライアントアプリケーションにメッセージを送信します。

## **-HTMLFORM**

-HTMLFORM は、Web ページ用に作成した HTML ファイルにレポートの出力を送信し、そのページをブラウザに表示します。出力を HTML ファイルに保存しておき、後から表示することもできます。

構文は次のとおりです。

```
-HTMLFORM filename [SAVE AS htmlpage]
```

または

```
-HTMLFORM BEGIN
.
.
.
-HTMLFORM END
```

説明

**filename**

WebFOCUS がこのコマンドで作成するレポートのプレースホルダを格納する HTML ファイルです。

```
ON TABLE HOLD HTMTABLE AS report
```

**SAVE**

HTML ファイル filename と WebFOCUS が生成したレポートを組み合わせで作成される HTML ページを保存するよう指定します。

**AS htmlpage**

HTML ファイル filename と WebFOCUS が生成したレポートを組み合わせで作成される HTML ページのファイル名です。このファイルを保存しておき、後から表示することもできます。

```
-HTMLFORM BEGIN
```

プロシジャ内のインライン HTML フォームの開始を示します。

```
-HTMLFORM END
```

プロシジャ内のインライン HTML フォームの終了を示します。

**-IF**

-IF は、指定した式の評価に基づいて、プロシジャの実行を転送します。

式が False となる -IF コマンドを ELSE を指定せずに使用した場合、そのすぐ後に続く行から処理を継続します。

構文は次のとおりです。

```
-IF expression [THEN] GOTO label1; [ELSE GOTO label2;]
 [ELSE IF...;]
```

説明

**label1...label2**

-GOTO アクションのターゲットを指定するユーザ定義の名前です。この名前の最大長は、64 バイトです。

この名前にブランクを含めたり、-QUIT および -EXIT 以外のダイアログマネージャコマンドを使用したりすることはできません。関数および数値演算や論理演算などと混同しやすい語句は使用しないでください。

#### **expression**

有効な式です。リテラルにブランクまたはカンマ (,) が含まれていない場合は、一重引用符 (') で囲む必要はありません。

#### **THEN**

コマンドを読みやすくするために使用する任意指定のコードです。

#### **ELSE GOTO**

-IF テストに失敗した場合にコントロールを label2 に渡します。

#### **ELSE IF**

複合 -IF テストを指定します。

コマンドの末尾にセミコロン (;) を挿入し、後続行をハイフン (-) で開始する必要があります。

### **-INCLUDE**

-INCLUDE は、プロシジャから別のプロシジャの呼び出しを可能にします。プロシジャからは、別のプロシジャをいくつでも呼び出すことができます。最大で 4 つの -INCLUDE コマンドをネストすることができます。

呼び出し先のプロシジャには、完全に実行可能なコード、またはコードの一部を含めることができます。呼び出し側のプロシジャおよび呼び出し先のプロシジャは、互いに他方のラベルに分岐することはできません。

構文は次のとおりです。

**-INCLUDE filename**

説明

#### **filename**

呼び出し先のプロシジャ名です。

完全修飾ファイル名を使用する際の構文についての詳細は、396 ページの「[-INCLUDE コマンドでの完全修飾名の使用](#)」を参照してください。

### **-label**

-label は、-GOTO コマンドまたは -IF 条件のターゲットです。

構文は次のとおりです。

**-label [TYPE message]**

説明

**label**

分岐のターゲットを識別するユーザ定義の名前です。名前の最大長は 64 バイトです。

この名前にブランクを含めたり、-QUIT および -EXIT 以外のダイアログマネージャコマンドを使用したりすることはできません。関数および数値演算や論理演算などと混同しやすい語句は使用しないでください。

**TYPE message**

クライアントアプリケーションにメッセージを送信します。

## -PASS

-PASS は、パスワードの発行および管理を直接実行します。この機能は、特定のユーザが読み取りまたは書き込みできるファイルやファイルセットを指定する場合に特に役立ちます。パスワードには、DBA 機能を使用して関連付けられた詳細機能が一式備わっています。

パスワードを設定するプロシジャは、暗号化することができます。これにより、プロシジャ自体および設定するパスワードが入力されたり、他人に知られたりすることを防止できます。

変数を -PASS に関連付けることができます。これにより、パスワード値の入力を要求したり、パスワード値を割り当てたりすることが可能になります。

構文は次のとおりです。

**-PASS password**

説明

**password**

FOCUS のリテラルパスワード、またはパスワードを格納する変数です。

## -QUIT

-QUIT は、プロシジャを即座に強制終了します。スタックコマンドは実行されません。

プロシジャが別のプロシジャから呼び出されている場合、制御は呼び出し側のプロシジャではなく、クライアントアプリケーションに直接返されます。

-QUIT は、分岐のターゲットにすることもできます。

構文は次のとおりです。

**-QUIT**

## -QUIT FOCUS

-QUIT FOCUS は、プロシジャおよび FOCUS を終了します。制御は、オペレーティングシステムまたは呼び出し側プログラムに返されます。

FOCUS を終了するときに、呼び出し側のプログラムが存在すれば、これに値が返されます。これを使用することで、別の Windows アプリケーションから FOCUS を呼び出した場合に、FOCUS の終了状態を示すステータスコードを返すことができます。

さらに、-QUIT FOCUS を分岐のターゲットにして、同一結果を得ることができます。

構文は次のとおりです。

**-QUIT FOCUS [n]**

説明

**n**

アプリケーションのリターンコードの値です。これは、定数または整数の変数を指定することができます。

n の値を指定しなかった場合、または整数以外の値を指定した場合、リターンコードはデフォルト値の 0 (ゼロ) になります。

## -READ

-READ は、オペレーティングシステムに定義された外部ファイルからデータを読み取ることができます。

-READ が検出される前に、変数リストの長さを特定しておく必要があります。各変数に使用する文字数は、-DEFAULT を使用して設定します。

可変長フォーマットのファイルを読み取る場合、変数のリストが 1 行に収まらないときは、先頭行の末尾にカンマ (,) を入れ、次の行の先頭にハイフン (-) を入れて開始します。

```
-READ EXTFILE, &CITY, &CODE1,
- &CODE2
```

固定フォーマットのファイルを読み取る場合、以下のようにハイフン (-) とカンマ (,) の組み合わせで開始します。

```
-READ EXTFILE &CITY.A8. &CODE1.A3.,
-, &CODE2.A3.
```

構文は次のとおりです。

```
-READ filename[,] [NOCLOSE] &name[.format.][,]...
```

### 説明

#### filename[,]

読み取り先の外部ファイル名です。これは、使用するオペレーティングシステムで定義されたものでなければなりません。filename の後のブランクは固定フォーマットファイルを表し、カンマ (,) は可変長フォーマットファイルを表します。

#### NOCLOSE

-READ 処理が完了するまで、外部ファイルを開いた状態にします。NOCLOSE オプションで開いておいたファイルは、-CLOSE filename コマンドを使用して閉じることができます。

#### &name[,]...

変数のリストです。可変長フォーマットファイルでは、それぞれの変数名をカンマ (,) で区切ることができますが、必須ではありません。

#### format.

変数のフォーマットです。文字 (A) または数値 (I) です。フォーマットはピリオド (.) で区切る必要があります。カンマ区切りファイルでは format は無視されます。

## -READFILE

-READFILE は、マスターファイルフィールドをダイアログマネージャ変数に読み込みます。

構文は次のとおりです。

```
-READFILE [app/]mastername
```

説明

**app**

ファイルが格納されているアプリケーションディレクトリです。

**mastername**

読み取るマスターファイルの名前です。

## -REPEAT

-REPEAT は、ストアドプロシジャ内でループの実行を可能にします。ループは、次のいずれかが発生した場合に終了します。

- ❑ ループが完了した場合。
- ❑ -QUIT または -EXIT コマンドが発行された場合。
- ❑ -GOTO が発行されて、ループ外のラベルに移動する場合。-GOTO を再発行してそのループに戻った場合、ループは先に終了した箇所から続行します。

構文は次のとおりです。

```
-REPEAT label n TIMES
```

または

```
-REPEAT label WHILE condition;
```

または

```
-REPEAT label FOR &variable [FROM fromval] [TO toval] [STEP s]
```

説明

**label**

繰り返し実行するコード (ループ) を指定します。2 つのループのラベル名が異なる場合、一方のラベルに他方のラベルのループを含めることができます。

**n TIMES**

ループを実行する回数を指定します。n の値には、ローカル変数、グローバル変数、定数のいずれかを指定することができます。変数を指定した場合、評価は 1 回に限定されるため、ループの実行回数を変更することはできません。ループを途中で終了するには、-QUIT または -EXIT を使用します。

**WHILE condition;**

ループを実行する条件を指定します。条件は、True または False を返す論理式です。条件が True の場合にループが実行されます。

**FOR &variable**

各ループの実行を開始する際にテストする変数です。fromval および toval の値が指定されている場合は、それらの値と比較されます。STEP が正の値のときは、&variable が toval 以下の場合にのみループが実行されます。STEP が負の値のときは、&variable が toval 以上の場合にのみループが実行されます。toval を使用しない場合は、n TIMES などの別のオプションを代用する必要があります。

**FROM fromval**

ループの実行を開始するたびに &variable と比較する定数です。デフォルト値は 1 です。

**TO toval**

ループの実行を開始するたびに &variable と比較する値です。デフォルト値は 1,000,000 です。

**STEP s**

ループの実行が終了するたびに &variable に累積する定数です。正または負の値を指定することができます。デフォルト値は 1 です。

**注意：**パラメータの FROM、TO、STEP は、任意の順序で使用することができます。

**-RUN**

-RUN は、すべてのスタックコマンドを即座に実行します。

実行後は、-RUN の次の行からストアドプロシジャの処理が続行します。

-RUN は、一般的に次の操作に使用します。

- ❑ リクエストを実行して結果を生成し、その結果をテストおよび分岐に使用します。
- ❑ -READ または -WRITE を使用して開いた外部ファイルを閉じます。-READ の場合、ファイルを閉じると、行ポインタがファイルの先頭に配置されます。-WRITE の場合、ファイルの割り当ておよび定義に基づいて行ポインタが配置されます。

構文は次のとおりです。

**-RUN**

**-SET**

-SET は、数式または論理式で計算された値あるいはリテラル値を変数に割り当てます。

リテラル値にブランク、カンマ (,)、等号 (=) が含まれている場合は、値を一重引用符 (') で囲む必要がありますが、これらの記号が含まれていない場合は一重引用符の使用はオプションです。

構文は次のとおりです。

```
-SET &[&]name= {expression|value};
```

説明

**&name**

値を設定する変数名です。

**expression**

有効な式です。式は複数の行に記述することができます。その場合は、コマンドの末尾にセミコロン (;) を追加します。

**value**

変数に割り当てられたリテラル値です。リテラル値にカンマ (,) またはブランクが含まれている場合は、値を一重引用符 (') で囲む必要があります。値自体に一重引用符が含まれている場合は、一重引用符を 2 つ使用することで一重引用符を表します。

## -TSO

WebFOCUS では、-TSO はプロシジャから TSO オペレーティングシステムコマンドを実行します。これは、コマンドが RUN の場合にのみサポートされます。

構文は次のとおりです。

```
-TSO command
```

説明

**command**

TSO RUN コマンドです。

## -TYPE

-TYPE は、クライアントアプリケーションにメッセージを送信します。

-TYPE コマンドは連続的に使用することができますが、必ず -TYPE で開始する必要があります。

変数は、メッセージに埋め込むことができます。それぞれの変数に現在割り当てられている値が表示されます。

構文は次のとおりです。

```
-TYPE text
```

説明

text

クライアントアプリケーションに送信されるメッセージで、ラインフィードが末尾に追加されます。テキストの前後を引用符で囲んだ場合は、引用符もメッセージの一部として表示されます。

テキストの長さは、最大で 256 バイトです。

## -UNIX

WebFOCUS で、-UNIX はプロシジャから UNIX オペレーティングシステムコマンドを実行します。

構文は次のとおりです。

**-UNIX command**

説明

command

UNIX コマンドです。

## -WINNT

-WINNT は、プロシジャから Windows オペレーティングシステムコマンドを実行します。

構文は次のとおりです。

**-WINNT command**

説明

command

Windows コマンドです。

## **-WRITE**

**-WRITE** は、外部ファイルにデータを書き込みます。

コマンドが複数行にわたる場合は、行の末尾にカンマ (,) を追加し、後続行をそれぞれハイフン (-) で開始します。

**NOCLOSE** オプションを指定していない場合、**-RUN**、**-EXIT**、**-QUIT** コマンドでプロシジャを終了する際に開いた状態のファイルも閉じます。

構文は次のとおりです。

```
-WRITE filename [NOCLOSE] text
```

説明

**filename**

書き込み先の外部の物理ファイル名です。filename は、オペレーティングシステムで識別できる名前であればなりません。

**NOCLOSE**

**-WRITE** 処理が完了するまで外部ファイルを開いた状態にしておきます。**NOCLOSE** オプションを使用して開いた状態にしておいたファイルは、**-CLOSE filename** コマンドを使用して閉じることができます。

**text**

変数とテキストの組み合わせです。

# 6

## プロシジャのテストとデバッグ

---

現在の環境に対してクエリを実行し、バージョン、サーバ情報、JOIN などの情報を表示したり、使用中のファイルを識別したりして、WebFOCUS アプリケーションのデバッグを行うことができます。特別なダイアログマネージャ変数や Web コンソールで実行するトレースなどの他のデバッグ方法も用意されています。

### トピックス

- |                                                      |                                                                     |
|------------------------------------------------------|---------------------------------------------------------------------|
| <input type="checkbox"/> クエリコマンドによるアプリケーションのデバッグ     | <input type="checkbox"/> 国際言語サポートの表示                                |
| <input type="checkbox"/> 結合構造の表示                     | <input type="checkbox"/> エラーメッセージ説明の表示                              |
| <input type="checkbox"/> 一時項目 (DEFINE) の表示           | <input type="checkbox"/> 現行検索パスの表示                                  |
| <input type="checkbox"/> 有効な通貨データソースの表示              | <input type="checkbox"/> バージョン番号の表示                                 |
| <input type="checkbox"/> 使用可能なフィールドの表示               | <input type="checkbox"/> リモート TIBCO WebFOCUS Reporting Server の値の表示 |
| <input type="checkbox"/> ファイルディレクトリテーブルの表示           | <input type="checkbox"/> パラメータ設定の表示                                 |
| <input type="checkbox"/> マスターファイルフィールド情報の表示          | <input type="checkbox"/> グラフパラメータの表示                                |
| <input type="checkbox"/> FILEDEF で割り当てた現行 ddname の表示 | <input type="checkbox"/> 接続サーバサイトコードの表示                             |
| <input type="checkbox"/> 定義済み関数の表示                   | <input type="checkbox"/> スタイルシートパラメータ設定の表示                          |
| <input type="checkbox"/> HOLD フィールドの表示               | <input type="checkbox"/> グローバル変数値の表示                                |
| <input type="checkbox"/> JOIN 構造の表示                  | <input type="checkbox"/> 使用中ファイルの識別                                 |
|                                                      | <input type="checkbox"/> システムテーブルからの動的レポート                          |
- 

### クエリコマンドによるアプリケーションのデバッグ

クエリコマンドを使用すると、メタデータ、データソース、言語環境、WebFOCUS に関する情報を表示することができます。この方法は、アプリケーションの正常な動作を阻害するアプリケーションの要素を特定する場合に役立ちます。現在の環境に対してクエリを実行するには、WebFOCUS でクエリコマンドを使用します。

クエリコマンドの結果は、ブラウザのウィンドウに表示されるか、HTML ファイルのコメントとして表示されます。HTML のコメントを表示するには、Web ブラウザでこの操作に該当する機能を使用します。たとえば、Microsoft Internet Explorer では、[ソースの表示] を使用します。

構文                    クエリコマンドの発行

? query [filename]

説明

query

クエリのコマンド名です。

filename

クエリの対象となるファイルの名前です。このパラメータは、特定のクエリにのみ適用されます。

クエリコマンドをリスト表示するには、プロシジャ内で「?」と入力します。

参照                    クエリコマンドの概要

下表は、クエリコマンドの一覧です。ここには、コマンドごとの詳細情報が含まれています。

クエリコマンド	説明
? COMBINE	結合したファイル構造のリストを表示します。
? DEFINE	DEFINE コマンドまたは属性を使用して作成した現在アクティブな一時項目を表示します。
? SET EUROFILE	現在有効な通貨データソースをリスト表示します。
?F	現在有効なフィールドをリスト表示します。
? FDT	FOCUS データソースの物理的な属性を表示します。
?FF	アクティブなマスターファイルのフィールド名、エイリアス、フォーマットに関する情報をリスト表示します。
? FILEDEF	FILEDEF コマンドで割り当てられた現在の論理名を表示します。

クエリコマンド	説明
? FUNCTION	DEFINE コマンドで作成した関数を表示します。
? HOLD	HOLD マスターファイルで定義されたフィールドを表示します。
? JOIN	データソース間に存在する JOIN 構造を表示します。
? LANG	国際言語サポートに関する情報を表示します。
? MDI	マルチディメンションインデックスの統計および説明を生成します。
? n	エラーメッセージの説明を表示します。n は、エラーメッセージの番号を表します。
? PATH	現在の検索パスを表示します。
? RELEASE	製品のバージョン番号を表示します。
? REMOTE	リモートサーバの値を表示します。
? SET	WebFOCUS を制御するパラメータ設定を表示します。
? SET GRAPH	パラメータ設定を表示します。このパラメータ設定は、GRAPH コマンドで生成するグラフを制御します。
? SET NOT	特定の領域で設定できない SET コマンドをリスト表示します。
? SITECODE	接続サーバのサイトコードを取得します。
? STYLE	スタイルシートパラメータの現在の設定を表示します。
? &&	グローバル変数値を表示します。

## 結合構造の表示

? COMBINE コマンドは、現在の結合構造内に存在するファイルを表示します。

### 構文 結合構造の表示

? COMBINE

### 例 結合構造の表示

次のコマンドを発行します。

? COMBINE

次のような情報が生成されます。

FILE=EMPLJOB COMBINED FROM EMPLOYEE JOBFIL

### 参照 ? COMBINE クエリ情報

ファイル構造リストの形式は、次のとおりです。

```
FILE=combine_name COMBINED FROM
file_1
file_2
.
.
.
file_n
```

説明

combine\_name

COMBINE 構造の名前です。

file\_1...file\_n

結合構造を構成する FOCUS データソースの名前です。

## 一時項目 (DEFINE) の表示

? DEFINE コマンドは、リクエストで使用するアクティブな一時項目 (DEFINE) をリスト表示します。一時項目 (DEFINE) は、DEFINE コマンドまたはマスターファイル内の DEFINE 属性で作成することができます。このコマンドで表示できるフィールド名の最大長は 32 バイトです。フィールド名が 32 バイトを超える場合は、さらに名前が続くことを示す記号が 32 バイト目に表示されます。WebFOCUS では、この記号はアンパサンド (&) になります。

## 構文 一時項目 (DEFINE) の表示

? DEFINE [appname/][filename]

説明

**appname**

アプリケーションディレクトリです。

**filename**

一時項目 (DEFINE) が格納されたデータソースです。filename を省略すると、このコマンドはすべての一時項目 (DEFINE) を表示します。

## 例 一時項目 (DEFINE) の表示

ここでは、EMPLOYEE データベースに対するリクエストで「NEW\_DATE」という名前の一時項目 (DEFINE) を作成した場合を想定します。

次のコマンドを発行します。

? DEFINE

次の情報が生成されます。

```
FILE FIELD NAME FORMAT SEGMENT VIEW TYPE EMPLOYEE NEW_DATE YYMD 2
```

## 参照 ? DEFINE クエリ情報

DEFINE で作成したそれぞれの一時項目に対して、次の情報がリスト表示されます。

オプション	説明
<b>FILE</b>	一時項目 (DEFINE) が格納されたデータソースの名前です。
<b>FIELD NAME</b>	一時項目 (DEFINE) の名前です。
<b>FORMAT</b>	一時項目 (DEFINE) のフォーマットです。この表記は、マスターファイル内の FORMAT 属性で使用する表記と同一です。
<b>SEGMENT</b>	一時項目 (DEFINE) が存在するマスターファイル内のセグメント数です。レポートの作成時、アプリケーションは一時項目 (DEFINE) をこのセグメントのフィールドとして扱います。セグメント番号をセグメント名に関連付けるには、? FDT を使用します。

オプション	説明
VIEW	代替ビューを指定する DEFINE のルートセグメントです。以下はその例です。 <code>DEFINE FILE EMPLOYEE.JOBCODE</code>
TYPE	マスターファイルの DEFINE 属性を使用して一時項目を作成したか、 DEFINE コマンドを使用して一時項目を作成したかを示します。前者の場合は MASTER、後者の場合はブランクが表示されます。

有効な通貨データソースの表示

? SET EUROFILE コマンドは、現在有効な通貨データソースを表示します。

構文 有効な通貨データソースの表示

有効な通貨データソースを表示するには、次のコマンドを発行します。

? SET EUROFILE

例 有効な通貨データソースの表示

次のコマンドを発行します。

? SET EUROFILE

次のような情報が生成されます。

EUROFILE CURRCODE

EUROFILE GBP

使用可能なフィールドの表示

?F コマンドは、現在使用可能なフィールドを表示します。

?F コマンドでは、フィールド名の 66 バイト全体が表示されます。

**構文**      **使用可能なフィールドの表示**`?F filename`

説明

`filename`

データソースの名前です。

**例**      **使用可能なフィールドの表示**

次のコマンドを発行します。

`?F EMPLOYEE`

次の情報が生成されます。

```

FILENAME = EMPLOYEE EMP_INFO.EMP_ID LAST_NAME FIRST_NAME HIRE_DATE
DEPARTMENT CURR_SAL CURR_JOBCODE ED_HRS BANK_NAME BANK_CODE BANK_ACCT
EFFECT_DATE DAT_INC PCT_INC SALARY PAYINFO.JOBCODE TYPE ADDRESS_LN1
ADDRESS_LN2 ADDRESS_LN3 ACCTNUMBER PAY_DATE GROSS DED_CODE DED_AMT
JOBSEG.JOBCODE JOB_DESC SEC_CLEAR SKILLS SKILLS_DESC DATE_ATTEND
ATTENDSEG.EMP_ID COURSE_CODE COURSE_NAME

```

**ファイルディレクトリテーブルの表示**

? FDT コマンドは、FOCUS データソースの物理特性をリスト化したファイルディレクトリテーブルを表示します。

セグメントおよびインデックス (マスターファイル内でキーワードの FIELDTYPE=I で指定されたフィールド) のそれぞれが 1 ページ分を占有します。ファイルディレクトリテーブルには、1 ページにセグメントインスタンスが占有する領域の大きさ、開始ページ番号と終了ページ番号、開始から終了までの全ページ数などの情報がセグメントおよびインデックス別に表示されます。

**構文**      **ファイルディレクトリテーブルの表示**`? FDT filename`

説明

`filename`

データソースの名前です。

**例**      **ファイルディレクトリテーブルの表示**

次のコマンドを発行します。

? FDT EMPLOYEE

次の情報が生成されます。

```
DIRECTORY:C:\¥ibi¥WEBFOCUS¥...¥employee.foc ON 09/25/2006 AT 09.50.28
DATE/TIME OF LAST CHANGE: 03/30/2005 16.19.22 SEGNAME LENGTH PARENT
START END PAGES LINKS TYPE 1 EMPINFO 22 1 1 1 6 2 FUNDTRAN 10 1 2 2 1 2
3 PAYINFO 8 1 3 3 1 3 4 JOBSEG 11 3 4 5 SECSEG 4 4 2 6 SKILLSEG 11 4 2 7
ADDRESS 19 1 4 4 1 2 8 SALINFO 6 1 5 5 1 3 9 DEDUCT 5 8 6 8 3 2 10
ATTNDSEG 7 1 3 11 COURSEG 11 10 2
```

参照

? FDT クエリ情報

ファイルディレクトリテーブルには、次の情報がリスト表示されます。

SEGNAME	ファイルに存在する各セグメントの名前です。各セグメントに割り当てられた連続番号がテーブルの左側に表示されます。テーブルの下部に表示された番号なしの項目はインデックスです。このインデックスは、マスターファイル内で FIELDTYPE=I 属性が指定されたフィールドに付属します。
LENGTH	各セグメントインスタンスを 4 バイト単位で表した長さです。1 ページに収まるインスタンス数を求めるには、992 をこの値で除算します。
PARENT	親セグメントです。この数字は、SEGNAME フィールドに表示されたセグメント名を指定します。
START	セグメントまたはインデックスの開始ページ番号です。
END	セグメントまたはインデックスの終了ページ番号です。
PAGES	セグメントまたはインデックスが占有するページ数です。
LINKS	各セグメントインスタンスのポインタ部分の長さを 4 バイト単位で表した値です。各セグメントインスタンスは、「データ」と「ポインタ」という 2 つの部分で構成されます。ポインタは、他のインスタンスを特定するために使用する内部番号です。
TYPE	インデックスのタイプです。NEW は、バイナリインデックスを示します。OLD は、ハッシュインデックスを示します。このファイルには、KU、LM、DKU、DKM、KL、KLU タイプのセグメントは物理的には存在しません。そのため、この情報はテーブルに表示されません。

## マスターファイルフィールド情報の表示

?FF コマンドは、アクティブなマスターファイルのフィールド名、エイリアス、フォーマットに関する情報を表示します。

### 構文 マスターファイルフィールド情報の表示

```
?FF filename [string]
```

説明

**filename**

マスターファイルの名前です。

**string**

最大長が 66 バイトの文字列です。このコマンドは、ここで指定した文字列で始まるフィールドの情報のみを表示します。このパラメータを省略すると、マスターファイルに存在するすべてのフィールドに関する情報が表示されます。

### 例 マスターファイルフィールド情報の表示

次のコマンドを発行します。

```
?FF EMPLOYEE
```

次の情報が生成されます。

```
FILENAME= EMPLOYEE EMPINFO.EMP_INFO EMPINFO.EID A9 LAST_NAME LN A15
FIRST_NAME FN A10 HIRE_DATE HDT 16YMD DEPARTMENT DPT A10 CURR_SAL CSAL
D12.2M CURR_JOB_CODE CJC A3 ED_HRS OJT F6.2 BANK_NAME BN A20 BANK_CODE BC
I6S BANK_ACCT BA I9S EFFECT_DATE EDATE 16YMD
```

## FILEDEF で割り当てた現行 ddname の表示

? FILEDEF コマンドは、各種入力ファイルおよび出力ファイルで割り当てた論理名 (ddname) を表示します。

現在の処理タイプに応じて、セッション中に WebFOCUS が多数の FILEDEF を作成する場合があります。? FILEDEF コマンドを使用し、システムが作成した 4 つの FILEDEF を確認することができます。ユーザが作成した FILEDEF およびシステムが作成した FILEDEF を合わせて、最大で 250 個の FILEDEF を使用することができます。詳細は、663 ページの「[TIBCO WebFOCUS ファイルの定義と割り当て](#)」を参照してください。

### 構文 現行 ddname の表示

現在の ddname を表示するには、次のコマンドを発行します。

? FILEDEF

## 例 現行 ddname の表示

次のコマンドを発行します。

? FILEDEF

次のような情報が生成されます。

```
Lname Device Lrecl Recfm Append Expl Filename
=====
HOLD2 DISK 0 V N Y C:¥VM¥SMALL¥HOLD2.FTM
```

## 定義済み関数の表示

? FUNCTION コマンドは、定義済みの関数とそのパラメータをすべて表示します。

## 構文 定義済み関数の表示

定義済み関数を表示するには、次のコマンドを発行します。

? FUNCTION

## 例 定義済み関数の表示

次のコマンドを発行します。

? FUNCTION

次のような情報が生成されます。

FUNCTIONS CURRENTLY ACTIVE

Name	Format	Parameter	Format
SUBTRACT	D8.2	VAL1	D8
		VAL2	D8

## HOLD フィールドの表示

? HOLD コマンドは、マスターファイルで定義したフィールドの中で、ON TABLE HOLD コマンドで作成したフィールドをリスト表示します。このリストには、フィールド名、エイリアス、FORMAT (USAGE) 属性で定義したフォーマットが表示されます。? HOLD コマンドで表示されるフィールド名の最大長は 32 バイトです。フィールド名が 32 バイトを超える場合は、さらに名前が続くことを示す記号が 32 バイト目に表示されます。WebFOCUS では、この記号はアンパサンド (&) になります。

? HOLD コマンドは、現在のリクエストで作成した HOLD マスターファイルのフィールドを表示します。

## 構文 HOLD フィールドの表示

? HOLD [filename]

説明

filename

ON TABLE HOLD コマンドの AS 句で割り当てられた名前です。このファイル名を省略すると、ファイル名はデフォルトの HOLD になります。

## 例 HOLD フィールドの表示

次のコマンドを発行します。

? HOLD

次のような情報が生成されます。

```
DEFINITION OF HOLD FILE FIELDNAME ALIAS FORMAT LAST_NAME E01 A15
FIRST_NAME E02 A10
```

## JOIN 構造の表示

? JOIN コマンドは、現在有効な JOIN 構造をリスト表示します。このコマンドで表示されるフィールド名は最大で 12 バイトです。フィールド名が 12 バイトを超える場合は、さらに名前が続くことを示す記号が 12 バイト目に表示されます。WebFOCUS では、この記号はアンパサンド (&) になります。

## 構文 JOIN 構造の表示

JOIN 構造を表示するには、次のコマンドを発行します。

? JOIN

## 例 JOIN 構造の表示

次のコマンドを発行します。

? JOIN

次のような情報が生成されます。

JOINS CURRENTLY ACTIVE			CROSSREFERENCE					
HOST			FIELD	FILE	TAG	AS	ALL	WH
FIELD	FILE	TAG	FIELD	FILE	TAG	AS	ALL	WH
-----	-----	---	-----	-----	---	--	---	--
JOB CODE	EMPLOYEE		JOB CODE	JOB FILE			N	N

参照 ? JOIN クエリ情報

次の JOIN 情報がリスト表示されます。

HOST FIELD	データソースを結合するホストフィールドの名前です。
FILE	ホストデータソースの名前です。
TAG	ホストデータソースのフィールド名に対して一意の識別子として使用するタグ名です。
CROSSREFERENCE FIELD	データソース間の JOIN に使用するクロスリファレンスフィールドの名前です。
FILE	クロスリファレンスデータソースの名前です。
TAG	クロスリファレンスデータソース内のフィールド名に対して一意の識別子として使用するタグ名です。
AS	JOIN 構造の名前です。
ALL	非ユニーク JOIN には Y、ユニーク JOIN には N を表示します。
WH	この JOIN が、条件付き JOIN であるか、等価 JOIN であるかを指定します。

国際言語サポートの表示

? LANG コマンドは、国際言語サポートに関する情報を表示します。

構文 国際言語サポートに関する情報の表示

国際言語サポートに関する情報を表示するには、次のコマンドを発行します。

? LANG

**例 国際言語サポートに関する情報の表示**

次のコマンドを発行します。

? LANG

次のような情報が生成されます。

```
NATIONAL LANGUAGE INFORMATION Language 001/AMENGLISH () Code
Page 437 client Code Page 437 Dollar 24($) Lowcase alphabet YES Decimal
notation OFF(.) Currency symbol $ Date/Time format EDA NLS sort NO NLS
upcase/lowcase NO DBCS Flag OFF(SBCS)
```

**エラーメッセージ説明の表示**

? n コマンドは、エラーメッセージの詳細な説明およびエラーの解決方法を表示します。

この機能を使用して、DB2 および MODEL 204 などの特定のデータアダプタで生成されたエラーメッセージを参照することもできます。

**構文 エラーメッセージ説明の表示**

? n

説明

n

エラーメッセージの番号です。

**例 エラーメッセージ説明の表示**

たとえば、次のようなエラーメッセージが表示された場合を想定します。

(FOC125) RECAP 演算に誤りがあります。

次のコマンドを発行します。

? 125

次のメッセージが生成されます。

```
(FOC125) RECAP 演算に誤りがあります。
RECAP という語のあとには必ず演算式を記述してください。
RECAP という語を削除するか、演算式を記述してください。
```

## 現行検索パスの表示

? PATH クエリは、現在の検索パスを表示します。ファイルの検索は、複数のディレクトリを対象として実行されます。これらのディレクトリの検索順序は、SET DEFAULT コマンドおよび FOCUS ディレクトリ変数の値を使用してそれぞれのユーザが指定します。

### 構文 現行検索パスの表示

```
? PATHquery commands: ? PATH; ? PATH command;
```

### 例 現行検索パスの表示

次のコマンドを発行します。

```
? PATH
```

WebFOCUS Reporting Server が EDAPATH を使用している場合は、次のような情報が生成されます。

```
EDAPATH = D:¥ibi¥apps¥ggdemo¥ D:¥ibi¥apps¥ncp¥ D:¥ibi¥apps¥template¥ C:¥VM¥SMALL¥ TEMPDIR = D:¥ibi¥srv82¥wfs¥edatemp¥ts000019¥ CURRDIR = D:¥ibi¥srv82¥wfs¥edatemp¥ts000019¥ EDASYN = D:¥ibi¥srv82¥wfs¥catalog¥ EDASYNR = D:¥ibi¥srv82¥wfs¥catalog¥
```

WebFOCUS Reporting Server が APP PATH を使用している場合は、次のような情報が表示されます。

```
BASEAPP = C:¥ibi¥apps¥baseapp; APPHOLD = C:¥ibi¥srv82¥wfs¥edatemp¥ts000001; APPPATH = C:¥ibi¥apps¥session; C:¥ibi¥apps¥session
```

## バージョン番号の表示

? RELEASE コマンドは、現在インストールされている製品のバージョン番号を表示します。

### 構文 バージョン番号の表示

バージョン番号を表示するには、次のコマンドを発行します。

```
? RELEASE
```

### 例 バージョン番号の表示

次のコマンドを発行します。

```
? RELEASE
```

次のような情報が生成されます。

```
EDA999 Release * Current Software EDA999 Release RELEASE = R729999B
EDA999 Release GEN_NUM = 000.223968 EDA999 Release SOURCE_DATE =
10/16/2002 18:13:18 EDA999 Release BUILD_DATE = 10/17/2002 02:59:24
EDA999 Release INSTALLATION_DATE = 10-17-2002 16:53:16
```

## リモート TIBCO WebFOCUS Reporting Server の値の表示

? REMOTE コマンドは、リモート WebFOCUS Reporting Server に関する情報を表示します。Reporting Server に接続している際に ? REMOTE コマンドを発行すると、リモートターゲット、サービス、ユーザ ID、クラスタノード、Servlet、Reporting Server ノード、Web サーバのユーザ ID およびパスワードに関する情報を取得することができます。

### 構文 リモート TIBCO WebFOCUS Reporting Server の値の表示

? REMOTE

### 例 リモート TIBCO WebFOCUS Reporting Server の値の表示

次のコマンドを発行します。

? REMOTE

次のような情報が生成されます。

```
Remote Destination --> LOOPBACK Remote User ID is not set. Remote User
Password is not set. There is no conversations active.
```

## パラメータ設定の表示

? SET コマンドは、パラメータ設定をリスト表示します。これらのパラメータ設定は、WebFOCUS 環境を制御します。アプリケーションにはパラメータのデフォルト値が設定されますが、SET コマンドを使用してデフォルト値を変更することができます。

### 構文 パラメータ設定の表示

? SET [ALL|[FOR] parameter]

説明

ALL

すべてのパラメータ設定を表示します。

parameter

SET パラメータです。特定のパラメータ設定を表示します。

FOR

パラメータ設定およびそのパラメータの設定元を表示します。

例                    パラメータ設定の表示

次のコマンドを発行します。

? SET

次のような情報が生成されます。

PARAMETER SETTINGS					
ALL.	OFF	FOC2GIGDB	ON	PAUSE	OFF
ASNAMES	FOCUS	FOCALLOC	OFF	POOL	OFF
AUTOINDEX	ON	FOCCREATELOC	OFF	PRINT	ONLINE
AUTOPATH	ON	FOCSTACK SIZE	8	PRINTPLUS	OFF
BINS	64	FOCTRANSFORM	ON	QUALCHAR	.
BLKCALC	NEW	FORMFEED	ASCII	QUALTITLES	OFF
BUSDAYS	_MTWTF_	HDAY		REBUILDMSG	1000
BYPANELING	OFF	HIPERFOCUS	OFF	RECAP - COUNT	OFF
CACHE	0	HOLDATTRS	FOCUS	SAVEMATRIX	OFF
CARTESIAN	OFF	HOLDLIST	ALL	SCREEN	OFF
CDN	OFF	HOLDSTAT	OFF	SHADOW PAGE	OFF
COLUMNSCROLL	OFF	HOTMENU	OFF	SMARTMODE	OFF
DATEDISPLAY	OFF	IBMLE	ON	SPACES	AUTO
DATEFNS	ON	INDEX TYPE	NEW	SQLENGINE	
DATETIME	STARTUP/RESET	LANGUAGE	AMENGLISH	SUMPREFIX	LST
DEFCENT	19	LINES/PAGE	66	TCPIPINT	OFF
DEFECHO	OFF	LINES/PRINT	57	TEMP DISK D:¥ibi¥srv8..	
DUPLICATECOL	ON	MERGEOPT	Y	TERMINAL	IBM3270
EMPTYREPORT	OFF	MESSAGE	ON	TESTDATE	TODAY
EQTEST	WILDCARD	MODE	WINNT	TITLES	ON
EXCELRELEASE	2000	MORE	OFF	VIEWNAME SIZE	60
EXL2KLANG	AMENGLISH	MULTIPATH	COMPOUND	WIDTH	130
EXTAGGR	ON	NFOC	OFF	WINPFKEY	OLD
EXTHOLD	ON	NODATA	.	XFBINS	64 (passive)
EXTRACT	OFF	ONFIELD	ALL	XFC	ON
EXTSORT	OFF	PAGE - NUM	ON	XRETRIEVAL	ON
FIELDNAME	NEW	PANEL	0	YRTHRESH	0

これらのパラメータの中には、SET コマンドで指定した名前と異なる名前で表示されるものがあります。以下はその例です。

SET パラメータ	説明
FOCSTACK SIZE	FOCSTACK パラメータと同一です。
INDEX TYPE	INDEX パラメータと同一です。

SET パラメータ	説明
<a href="#">LINES/PAGE</a>	PAPER パラメータと同一です。
<a href="#">LINES/PRINT</a>	LINES パラメータと同一です。
<a href="#">SHADOW PAGES</a>	SHADOW パラメータと同一です。

## 例 単一パラメータ設定の表示

次のコマンドを発行します。

```
? SET ONLINE-FMT
```

パラメータがデフォルト値に設定されている場合は、次のような情報が生成されます。

```
ONLINE-FMT HTML
```

## 例 有効なパラメータ設定場所の表示

次のコマンドを発行します。

```
? SET FOR EXTSORT
```

次の情報が生成されます。

```
EXTSORT ON
```

```

SETTABLE FROM COMMAND LINE : YES
SETTABLE ON TABLE : YES
SETTABLE FROM SYSTEM-WIDE PROFILE : YES
SETTABLE FROM HLI PROFILE : YES
```

## グラフパラメータの表示

? SET GRAPH コマンドは、GRAPH コマンドで生成されたグラフを制御するためのパラメータ設定をリスト表示します。

## 構文 グラフパラメータの表示

グラフパラメータを表示するには、次のコマンドを発行します。

```
? SET GRAPH
```

例                    グラフパラメータの表示

次のコマンドを発行します。

```
? SET GRAPH
```

次のような情報が生成されます。

GRAPH PARAMETER SETTINGS					
3DGRAPH	ON	GTREND	OFF	PIE	OFF
AUTOTICK	ON	GVIEWERMAX	OFF	TERMINAL	IBM3270
BARNUMB	OFF	GVIEWERPAUSE	ON	VAUTO	ON
BARSPACE	0	HAUTO	ON	VAXIS	66
BARWIDTH	1	HAXIS	130	VCLASS	.00
BSTACK	OFF	HCLASS	.00	VGRID	OFF
FORMATGRAPH	HTML	HISTOGRAM	ON	VMAX	.00
GMISSING	OFF	HMAX	.00	VMIN	.00
GMISSVAL	.00	HMIN	.00	VTICK	.00
GPROMPT	OFF	HSTACK	OFF	VZERO	OFF
GRIBBON	OFF	HTICK	.00		
GRID	OFF	LOOKGRAPH	NONE		

接続サーバサイトコードの表示

edaserve.cfg ファイルには、サーバに関連付けられたサイトコードを識別する「site\_code」と呼ばれるオプションのキーワードが格納されます。

サイトコードをインストールすると、その値は ? SITECODE クエリコマンドを発行して取得することができます。このクエリコマンドをリクエストで発行する場合、そのリクエストが接続するサーバのサイトコードが抽出されます。サイトコードがインストールされていない場合、サイトコードが利用できないことを示すメッセージが表示されます。

構文                    接続サーバサイトコードの検索

```
? SITECODE
```

例                    接続サーバサイトコードの検索

edaserve.cfg ファイルに次の項目が格納されています。

```
site_code = A52709b
```

次のクエリコマンドを発行します。

```
? SITECODE
```

出力結果は次のとおりです。

SITE CODE A52709b

サイトコードがインストールされていない場合、? SITECODE クエリは次のメッセージを返します。

SITE CODE NOT AVAILABLE

## スタイルシートパラメータ設定の表示

? STYLE コマンドは、スタイルシートパラメータの現在の設定を表示します。

### 構文 スタイルシートパラメータ設定の表示

? [SET] STYLE

### 例 スタイルシートパラメータ設定の表示

次のコマンドを発行します。

? STYLE

次のような情報が生成されます。

```

 STYLESHEET PARAMETER SETTINGS
ONLINE-FMT HTML
OFFLINE-FMT STANDARD
STYLESHEET ON
SQUEEZE OFF
LABELPROMPT OFF
STYLEMODE FULL
ORIENTATION PORTRAIT
UNITS INCHES
TOPMARGIN .250
BOTTOMMARGIN .250
LEFTMARGIN .250
RIGHTMARGIN .250
TARGETFRAME ????
FOCEXURL ¥cgi-bin¥ibi_cgi¥ibiweb.exe?IBIMR_drill=X,corp/corp.htm
BASEURL OFF

```

### 参照 ? STYLE クエリ情報

表示されるスタイルシート情報には次のものがあります。

ONLINE-FMT	レポート出力のフォーマットです。HTML (デフォルト)、PDF、Excel 2000、Excel 97 での出力およびスタイルが未設定のテキストベースの出力を生成することができます。
------------	----------------------------------------------------------------------------------------------

OFFLINE-FMT	レポート出力のフォーマットです。HTML での出力およびスタイルが未設定のテキストベースの出力を生成することができます。
STYLESHEET	ページのサイズ、方向、マージンなどのフォーマットオプションを指定するスタイルシートパラメータの使用の有無を切り替えます。
LABELPROMPT	宛名ラベルレポートなどの複数ウィンドウレポートを印刷する際に、先頭ページに使用するラベルを指定します。
STYLEMODE	出力を複数の HTML テーブルに表示し、大規模なレポートの出力を高速化します。この場合、各テーブルにはそれぞれ別のレポートページが表示されます。
ORIENTATION	スタイルを設定したレポートのページの方法です。縦長または横長のいずれかを指定することができます。
UNITS	PDF レポート出力に使用する単位です。インチ、センチメートル、ポイントのいずれかを指定します。
TOPMARGIN	レポート出力のページ上部の境界を設定します。
BOTTOMMARGIN	レポート出力のページ下部の境界を設定します。
LEFTMARGIN	レポート出力のページ左側の境界を設定します。
RIGHTMARGIN	レポート出力のページ右側の境界を設定します。
TARGETFRAME	すべてのドリルダウンハイパーリンクの移動先となるフレームです。
FOCEXURL	WebFOCUS Client からプロシジャを実行する際の、プロシジャおよびメソッドの場所です。
BASEURL	アプリケーションが HTML ドキュメントを作成した際に指定した相対 URL で、ブラウザはこの URL をデフォルトパスとして検索を実行します。

## グローバル変数値の表示

? && コマンドは、ダイアログマネージャのグローバル変数およびその現在値をリスト表示します。グローバル変数は、WebFOCUS Reporting Server に接続中はこれらの値を保持します。

**注意：**ダイアログマネージャのすべての変数 (ローカル、グローバル、システム、統計) に対してクエリを実行するには、次のコマンドを発行します。

-? &

詳細は、ダイアログマネージャのマニュアルを参照してください。

### 構文      グローバル変数値の表示

? &&

サイトによっては、ダイアログマネージャの変数を表すアンパサンド (& または &&) を別の記号で置き換えている場合があります。その場合は、置き換えた記号をクエリコマンドで使用してください。たとえば、ダイアログマネージャの変数をパーセント記号 (%) で表すように変更した場合、グローバル変数をリスト表示するには次のコマンドを発行します。

? %%

### 例      グローバル変数値の表示

次のコマンドを発行します。

? &&

次のような情報が生成されます。

```
&&CITY 'STAMFORD' &&CODE1 'B10' &&CODE2 'B20'
```

## 使用中ファイルの識別

アプリケーションのデバッグを実行する場合は、使用中のファイルを識別する必要があります。識別できるのは、プロシジャ、データソース、マスターファイル、アクセスファイル、スタイルシートのファイル名およびその格納場所です。WebFOCUS は、WebFOCUS Reporting Server のパスに基づいて検索を実行し、WHENCE コマンドを使用してファイルを識別します。

WHENCE コマンドを使用する場合、ファイル名の最大長は 64 バイトです。

## 構文      使用中プロシジャの検索

`WHENCE procedure FOCEXEC`

説明

`procedure`

プロシジャです。

## 例      使用中プロシジャの検索

次のコマンドを発行します。

`WHENCE FINANCE FOCEXEC`

次のような出力が生成されます。

`C:¥ibi¥apps¥session¥layout.fex`

## 構文      使用中マスターファイルの検索

`WHENCE masterfile MASTER`

説明

`masterfile`

マスターファイルです。

## 例      使用中マスターファイルの検索

次のコマンドを発行します。

`WHENCE CENTINV MASTER`

次のような出力が生成されます。

`C:¥ibi¥apps¥session¥centinv.mas`

## 構文      使用中アクセスファイルの検索

`WHENCE accessfile ACCESS`

説明

`accessfile`

アクセスファイルです。

**例            使用中アクセスファイルの検索**

次のコマンドを発行します。

```
WHENCE GKEIVP ACCESS
```

次のような出力が生成されます。

```
C:¥ibi¥srv¥home¥catalog¥gkeivp.acx
```

**構文          使用中スタイルシートの検索**

```
WHENCE stylesheet FOCSTYLE
```

説明

```
stylesheet
```

スタイルシートです。

**例            使用中スタイルシートの検索**

次のコマンドを発行します。

```
WHENCE GGORDERS FOCSTYLE
```

次のような出力が生成されます。

```
C:¥ibi¥apps¥session¥classic.sty
```

**システムテーブルからの動的レポート**

ユーザ環境についての情報 (アプリケーション、ファイル、カラム、ディレクトリ、テーブル、インデックス、キーなど) を動的に収集する一連のシノニムに対してレポートリクエストを発行することができます。また、関数、SET パラメータ、エラーファイルについての情報も取得することができます。このようなシノニムは、サーバの home ディレクトリ下の catalog ディレクトリに格納され、接尾語に FMI (FOCUS メタデータインターフェース) が指定されています。

**システムテーブルシノニムの概要**

FMI シノニムはそれぞれ、ユーザ環境の特定のファイルセットについて情報を取得します。

FMI マスターファイルを検証する場合、REMARKS 属性および DESCRIPTION 属性は、各システムテーブルおよびシステムテーブル内の各カラムによって返されるデータを定義します。

**注意:** システムテーブルシノニムは、今後のバージョンで変更される可能性があります。そのため、システムテーブルシノニムの構造に依存するアプリケーションの設定は行わないでください。

たとえば、次の例は SYSFILES マスターファイルのバージョンですが、これは、ユーザのアプリケーションパスに格納されたマスターファイルの情報をデフォルト設定で取得します。

```

$-----$
$ Copyright (c) 2013 TIBCO, Inc. All rights reserved. @MFSM_NOPROLOG@ $
$-----$
$--CAN BE USED TO RETRIEVE DIRECTORY INFO - USE THE FOLLOWING TO DEFINE DIRECTORY
$--SQL FMI SET SYSFILES EDASYNM
$--SQL FMI SET SYSFILES FOCEXEC
FILE=SYSFILES, SUFFIX=FMI, REMARKS='Metadata: Directory information', $
SEGMENT=FILE, SEGTYPE=S0, $
 FIELD=FILENAME , ,A64 ,A64B, DESC='MEMBER NAME
', $
 FIELD=LGNAME , ,A8 ,A8B , DESC='LOGICAL NAME
', $
 FIELD=PHNAME , ,A80 ,A80B, DESC='PHYSICAL NAME 1ST PART
', $
 FIELD=PHNAME2 , ,A80 ,A80B, DESC='PHYSICAL NAME 2ND PART
', $
 FIELD=PHNAME3 , ,A80 ,A80B, DESC='PHYSICAL NAME ETC..
', $
 FIELD=PHNAME4 , ,A80 ,A80B, DESC='PHYSICAL NAME The whole length up to
512 bytes ', $
 FIELD=PHNAME5 , ,A80 ,A80B, DESC='PHYSICAL NAME The last part is 32 but
can be up ', $
 FIELD=PHNAME6 , ,A80 ,A80B, DESC='PHYSICAL NAME to 44 because of nlscut
of ', $
 FIELD=PHNAME7 , ,A80 ,A80B, DESC='PHYSICAL NAME previous 6 parts (2
bytes per part)', $
 FIELD=VERSION , ,I4 ,I1, DESC='MF:VERSION
', $
 FIELD=MOD , ,I4 ,I1, DESC='MF:MODIFICATION NUMBER
', $
 FIELD=LINECNT , ,I4 ,I2, DESC='MF:CURRENT LINE COUNTER.
', $
 FIELD=DATE , ,A8 ,A8B, DESC='IBI DATE (DD/MM/YY)
', $
 FIELD=TIME , ,A8 ,A8B, DESC='IBI TIME (HH.MM.SS)
', $
 FIELD=USERID , ,A100 ,A100B, DESC='MF: LAST USER WHO CHANGED. UNIX/
NT:owner', $
 FIELD=SIZE , ,I11 ,I4, DESC='UNIX/NT:SIZE IN BYTES.
', $
 FIELD=EXTENSION , ,A3 ,A3B, DESC='ACCEPTED SHORT EXTENSION FOR FILE
', $

```

よく使用する FMI シノニムのリストは次のとおりです。SYSTABLE シノニムに対してリクエストを発行することで、システムテーブルシノニムのリストを作成することができます。

❑ **SYSAPPS** アプリケーションおよびアプリケーション内のファイルについての情報を取得します。

❑ **SYSCOLUMN** テーブルおよびテーブル内のカラムについての情報を取得します。

- ❑ **SYSDEFFN** DEFINE FUNCTION についての情報を取得します。
- ❑ **SYSERR** エラーメッセージファイル情報を取得します。
- ❑ **SYSFILES** ディレクトリ情報を取得します。
- ❑ **SYSIMP** インパクト分析情報を取得します。
- ❑ **SYSINDEX** インデックス情報を取得します。
- ❑ **SYSKEYS** キー情報を取得します。
- ❑ **SYSRPDIR** ユーザのアプリケーションパスで使用可能なすべてのプロシジャについての情報を取得します。
- ❑ **SYSSET** SET コマンドおよびグローバルダイアログマネージャ変数についての情報を取得します。
- ❑ **SYSSQLOP** 関数情報を取得します。
- ❑ **SYSTABLE** テーブル情報を取得します。
- ❑ **SYSVDTF** リレーショナルアダプタのデータタイプ情報を取得します。

## SYSAPPS - アプリケーションおよびアプリケーションファイルについてのレポート

SYSAPPS シノニムは、アプリケーションおよびアプリケーション内のファイルについての情報を取得します。

### 例 アプリケーション情報およびファイル情報の取得

次のリクエストは、ファイル名が「a」から「g」の文字で始まる ibisamp アプリケーション内のマスターファイルについて、アプリケーション名、パスおよびファイル名、拡張子、接尾語、キー、およびセグメント数を取得します。

```
TABLE FILE SYSAPPS
PRINT APPNAME AS App APPLC AS Path
FNAME AS 'File,Name' SUFFIX AS 'File,Type' KEYS
NUMSEG AS '# of,Segments'
WHERE APPNAME EQ 'ibisamp'
WHERE FEXT EQ 'mas'
WHERE FQNAME LT 'C:¥ibi¥apps¥ibisamp¥hday'
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。

App	Path	File Name	File Type	KEYS	# of Segments
ibisamp	C:\ibi\apps\ibisamp	brokers	FOC	BROKER_ID	1
ibisamp	C:\ibi\apps\ibisamp	car	FOC	COUNTRY CAR MODEL BODYTYPE WARRANTY STANDARD	7
ibisamp	C:\ibi\apps\ibisamp	carolap	FOC	COUNTRY CAR MODEL BODYTYPE WARRANTY STANDARD	7
ibisamp	C:\ibi\apps\ibisamp	cashflow	FOC	CASH_DATE	1
ibisamp	C:\ibi\apps\ibisamp	course	FOC	COURSECODE	1
ibisamp	C:\ibi\apps\ibisamp	courses	FOC	COURSE_CODE	1
ibisamp	C:\ibi\apps\ibisamp	educfile	FOC	COURSE_CODE DATE_ATTEND EMP_ID	2
ibisamp	C:\ibi\apps\ibisamp	empdata	FOC	PIN	1
ibisamp	C:\ibi\apps\ibisamp	employee	FOC	EMP_ID DAT_INC TYPE PAY_DATE DED_CODE SKILLS DATE_ATTEND	11
ibisamp	C:\ibi\apps\ibisamp	experson	FOC	SOC_SEC_NO	1
ibisamp	C:\ibi\apps\ibisamp	filemnt	FIX		1
ibisamp	C:\ibi\apps\ibisamp	finance	FOC	YEAR ACCOUNT	1
ibisamp	C:\ibi\apps\ibisamp	ggdemog	FOC	ST	1
ibisamp	C:\ibi\apps\ibisamp	ggorder	FOC	ORDER_NUMBER	2
ibisamp	C:\ibi\apps\ibisamp	ggprods	FOC	PRODUCT_ID	1
ibisamp	C:\ibi\apps\ibisamp	ggsales	FOC	SEQ_NO	1
ibisamp	C:\ibi\apps\ibisamp	ggstores	FOC	STORE_CODE	1

SYSCOLUM - テーブルおよびテーブル内のカラムについてのレポート

SYSCOLUM シノニムはテーブル情報を取得します。これには、テーブル名、作成者名、セグメント名およびセグメント数、ディメンションビューまたはビジネスビューでのセグメントロール、カラム名、カラムのデータタイプが含まれます。マスターファイルで参照されるデータソースについてのレポートとして使用します。

例                    テーブル情報およびカラム情報の取得

次のリクエストは、テーブル名が文字列「wf\_」で始まるテーブルから、テーブル情報およびカラム情報を取得します。

```
TABLE FILE SYSCOLUM
PRINT TBNAME AS Table TBTYPE AS Suffix NAME AS Field,Name COLTYPE AS
Data,Type ACTUAL AS Format
WHERE TBNAME LIKE 'wf_%'
WHERE RECORDLIMIT EQ 20
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。

<u>Table</u>	<u>Suffix</u>	<u>Field Name</u>	<u>Data Type</u>	<u>Format</u>
wf_retail	SQLMSS	ID_SALES	INTEGER	I4
wf_retail	SQLMSS	ID_STORE	INTEGER	I4
wf_retail	SQLMSS	ID_CURRENCY	INTEGER	I4
wf_retail	SQLMSS	ID_CUSTOMER	INTEGER	I4
wf_retail	SQLMSS	ID_DISCOUNT	INTEGER	I4
wf_retail	SQLMSS	ID_PRODUCT	INTEGER	I4
wf_retail	SQLMSS	ID_TIME	INTEGER	I4
wf_retail	SQLMSS	COGS_LOCAL	DOUBLE	D8
wf_retail	SQLMSS	COGS_US	DOUBLE	D8
wf_retail	SQLMSS	DISCOUNT_LOCAL	DOUBLE	D8
wf_retail	SQLMSS	DISCOUNT_US	DOUBLE	D8
wf_retail	SQLMSS	GROSS_PROFIT_LOCAL	DOUBLE	D8
wf_retail	SQLMSS	GROSS_PROFIT_US	DOUBLE	D8
wf_retail	SQLMSS	MSRP_LOCAL	DOUBLE	D8
wf_retail	SQLMSS	MSRP_US	DOUBLE	D8
wf_retail	SQLMSS	QUANTITY_SOLD	INTEGER	I4
wf_retail	SQLMSS	REVENUE_LOCAL	DOUBLE	D8
wf_retail	SQLMSS	REVENUE_US	DOUBLE	D8
wf_retail	SQLMSS	SALE_UNITY	INTEGER	
wf_retail	SQLMSS	ID_SHIPFACT	INTEGER	I4

## SYSDEFFN - DEFINE FUNCTION についてのレポート

SYSDEFFN シノニムは DEFINE FUNCTION 情報を取得します。これには、関数の名前、引数、引数のフォーマット、フィールド、説明が含まれます。

**例** **DEFINE FUNCTION 情報の取得**

次のリクエストは、DEFINE FUNCTION の名前、引数、引数のフォーマットについての情報を取得します。

```
TABLE FILE SYSDEFFN
PRINT DFNAME AS Function,Name ARGNAME AS Argument,Name ARGFORMAT AS
Argument,Format
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。

Function	Argument	Argument
<u>Name</u>	<u>Name</u>	<u>Format</u>
QUOTIENT	DIVIDEND	D8
QUOTIENT	DIVISOR	D8
ADD	VAL1	D8
ADD	VAL2	D8
SUBTRACT	VAL1	D8
SUBTRACT	VAL2	D8

**SYSERR - エラーメッセージファイルについてのレポート**

SYSERR シノニムは、エラーファイル名、各ファイル内の最小メッセージ数と最多メッセージ数、メッセージおよび説明のテキスト、メッセージ数、また警告メッセージかどうか、情報メッセージかどうか、プロシジャに行番号が表示されるかどうかの情報を取得します。

**例 エラーメッセージファイル情報の取得**

次のリクエストは、メッセージのテキストと説明を取得します。

```
TABLE FILE SYSERR
BY ERRNUM NOPRINT SUBHEAD
" <ERRTEXT "
" <ERRLINE1 "
" <ERRLINE2 "
" <ERRLINE3 "
" <ERRLINE4 "
WHERE RECORDLIMIT EQ 7
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。

(FOC001) THE NAME OF THE FILE OR THE WORD 'FILE' IS MISSING  
The request statement is missing the word FILE, or data fieldnames were encountered before the file's identity was provided.

(FOC002) A WORD IS NOT RECOGNIZED: %1  
A word which is not a reserved word begins a phrase. A reserved word may have been misspelled or there is a syntax error. Reserved words are IF, BY, AND, etc.

(FOC003) THE FIELDNAME IS NOT RECOGNIZED: %1  
A word which is assumed to be the name of a data field does not appear on the list of names or aliases for the file. Check the spelling of the fieldname.

(FOC004) THE OPTION ON THE VERB OBJECT IS NOT RECOGNIZED:  
The prefix in front of the fieldname is not valid. Valid options include: MAX., MIN., AVE., ASQ., FST., LST., PCT., RPCT., TOT., SUM., CNT., ALL., SEG., ST., and CT.

(FOC005) THE NUMBER OF VERB OBJECTS EXCEEDS THE MAXIMUM  
Up to 256 fields may be used in a single report request.  
This total does not include sort fields (e.g., BY and ACROSS fields).

(FOC006) THE FORMAT OF THE TEST VALUE IS INCONSISTENT WITH FIELD FORMAT: %1  
The value supplied as a literal in an IF or WHERE test must match the format type of the field being tested. For example, an integer field may not be tested against a non-numeric character string.

(FOC007) THE REQUEST STATEMENT DOES NOT CONTAIN A VERB  
The request statement does not contain a verb, and the heading, if any, does not contain a reference to any data fields, implying a verb.

## SYSFILES - メタデータまたはプロシジャのディレクトリ情報についてのレポート

SYSFILES シノニムは、アプリケーションパス内のマスターファイルまたはプロシジャファイルを取得します。デフォルト設定では、SYSFILES は、マスターファイルおよびそのプロパティのリストを取得します。SET SYSFILES コマンドが、取得するファイルタイプを指定します。

構文は次のとおりです。

```
SQL FMI SET SYSFILES {EDASYNM|FOCEXEC}
```

説明

[EDASYNM](#)

マスターファイルについての情報を取得します。これがデフォルト値です。

[FOCEXEC](#)

プロシジャファイルについての情報を取得します。

例

### マスターファイル情報の取得

次のリクエストは、ibisamp アプリケーションディレクトリ内でファイル名が「a」から「h」の文字で始まるマスターファイルについて、ファイル名、拡張子、パスを取得します。

```
TABLE FILE SYSFILES
PRINT FILENAME AS File
EXTENSION AS Extension
PHNAME AS Path
WHERE PHNAME LIKE 'ibisamp/%'
WHERE FILENAME LT 'i'
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。

<u>File</u>	<u>Extension</u>	<u>Path</u>
brokers	mas	ibisamp/brokers.mas
car	mas	ibisamp/car.mas
carolap	mas	ibisamp/carolap.mas
cashflow	mas	ibisamp/cashflow.mas
course	mas	ibisamp/course.mas
courses	mas	ibisamp/courses.mas
educfile	mas	ibisamp/educfile.mas
empdata	mas	ibisamp/empdata.mas
employee	mas	ibisamp/employee.mas
experson	mas	ibisamp/experson.mas
filemnr	mas	ibisamp/filemnr.mas
finance	mas	ibisamp/finance.mas
ggdemog	mas	ibisamp/ggdemog.mas
ggorder	mas	ibisamp/ggorder.mas
ggprods	mas	ibisamp/ggprods.mas
ggsales	mas	ibisamp/ggsales.mas
ggstores	mas	ibisamp/ggstores.mas
hday	mas	ibisamp/hday.mas

## SYSIMP - インパクト分析情報についてのレポート

SYSIMP シノニムは、ファイルの格納場所および参照先についての情報を取得します。SYSIMP には、呼び出しユーザ情報のセグメントおよび各呼び出しユーザが呼び出したファイルの子セグメントが含まれます。

例                   インパクト分析情報の取得

次のリクエストは、ibisamp アプリケーション内で名前が「s」から「z」の文字で始まる呼び出しユーザのファイルの名前、タイプ、説明、および呼び出されたファイルの名前、タイプ、説明を取得します。

```
TABLE FILE SYSIMP
PRINT CTYPE AS Caller,Type CDESCRIPTION AS Description
RFILE AS Called,Name
RPT_TYPE AS Called,Type
RLINENUM AS Line RUSAGE AS 'Used In'
REXTENSION AS Extension RDESCRIPTION AS Description
BY CFILE/A15 AS Caller,Name
WHERE CAPPLICATION EQ 'ibisamp'
WHERE CFILE GE 's'
ON TABLE SET PAGE NOLEAD
ON TABLE SET SHOWBLANKS ON
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

下図は、出力結果の一部を示しています。

Caller Name	Caller Type	Description	Called Name	Called Type	Line	Used In	Extension	Description
subfoot1	Procedure		car	MFD	2	TABLE	mas	Legacy Metadata Sample: car
wf_retail	Synonym	Cluster Join of Fact Tables Sales, Shipments and Labor for Demo Database	wf_retail_sales	MFD	4	CRFILE	mas	Sales Fact
	Synonym	Cluster Join of Fact Tables Sales, Shipments and Labor for Demo Database	wf_retail_customer	MFD	42	CRFILE	mas	Customer Dimension
	Synonym	Cluster Join of Fact Tables Sales, Shipments and Labor for Demo Database	wf_retail_age	MFD	211	CRFILE	mas	Age Dimension
	Synonym	Cluster Join of Fact Tables Sales, Shipments and Labor for Demo Database	wf_retail_time	MFD	967	CRFILE	mas	Time Dimension
	Synonym	Cluster Join of Fact Tables Sales, Shipments and Labor for Demo Database	wf_retail_time	MFD	935	CRFILE	mas	Time Dimension
	Synonym	Cluster Join of Fact Tables Sales, Shipments and Labor for Demo Database	wf_retail_industry	MFD	872	CRFILE	mas	Industry Dimension
	Synonym	Cluster Join of Fact Tables Sales, Shipments and Labor for Demo Database	wf_retail_occupation	MFD	344	CRFILE	mas	Occupation Dimension
	Synonym	Cluster Join of Fact Tables Sales, Shipments and Labor for Demo Database	wf_retail_marital_status	MFD	339	CRFILE	mas	Marital Status Dimension
	Synonym	Cluster Join of Fact Tables Sales, Shipments and Labor for Demo Database	wf_retail_income	MFD	326	CRFILE	mas	Income Dimension
	Synonym	Cluster Join of Fact Tables Sales, Shipments and Labor for Demo Database	wf_retail_geography	MFD	231	CRFILE	mas	Geography Dimension
	Synonym	Cluster Join of Fact Tables Sales, Shipments and Labor for Demo Database	wf_retail_education	MFD	220	CRFILE	mas	Education Dimension
	Synonym	Cluster Join of Fact Tables Sales, Shipments and Labor for Demo Database	wf_retail_currency	MFD	202	CRFILE	mas	Currency Dimension
	Synonym	Cluster Join of Fact Tables Sales, Shipments and Labor for Demo Database	wf_retail_time	MFD	170	CRFILE	mas	Time Dimension

SYSINDEX - インデックス情報のついてのレポート

SYSINDEX シノニムは、シノニム内で定義されたインデックスについての情報を取得します。

## 例 インデックス情報の取得

次のリクエストは、「gg」の文字列で始まるファイルのインデックスフィールド名を取得します。

```
TABLE FILE SYSINDEX
PRINT NAME AS Index
BY TBNAME AS File
WHERE TBNAME LIKE 'gg%'
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。

<u>File</u>	<u>Index</u>
ggdemog	ST
ggorder	PRODUCT_ID
	VENDOR_CODE
ggprods	PRODUCT_ID
	VENDOR_CODE
ggsales	CATEGORY
	PCD
	REGION
	ST
	STCD
ggstores	STORE_CODE
	STATE

## SYSKEYS - キー情報についてのレポート

SYSKEYS シノニムは、シノニム内で定義されたキーについての情報を取得します。

例            キー情報の取得

次のリクエストは、「gg」の文字列で始まるファイルのキーフィールド名およびソート順を取得します。

```
TABLE FILE SYSKEYS
PRINT IXNAME AS Key ORDERING AS Order
BY TBNAME AS File
WHERE TBNAME LIKE 'gg%'
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。

<u>File</u>	<u>Key</u>	<u>Order</u>
ggdemog	ST	A
ggorder	ORDER_NUMBER	A
ggprods	PRODUCT_ID	A
ggsales	SEQ_NO	A
ggstores	STORE_CODE	A

SYSRPDIR - ストアドプロシジャについてのレポート

SYSRPDIR シノニムは、アプリケーションパスで使用可能なすべてのプロシジャを取得します。

例            ストアドプロシジャ情報の取得

次のリクエストは、「wf\_」の文字列で始まるプロシジャを取得します。

```
TABLE FILE SYSRPDIR
PRINT RPC_TYPE AS Procedure,Type
BY RPC_NAME AS Procedure,Name
WHERE RPC_NAME LIKE 'wf_%'
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。

<u>Procedure Name</u>	<u>Procedure Type</u>
wf_mrauth	4-GL
wf_mrgroups	4-GL
wf_mrusers	4-GL
wf_retail_drop_app	4-GL
wf_retail_drop_sql	4-GL
wf_retail_jschart	4-GL
wf_retail_lite_def	4-GL
wf_retail_report	4-GL
wf_retail_validate	4-GL

## SYSSET - SET パラメータについてのレポート

SYSSET シノニムは、SET パラメータ、およびそれぞれの受容値とデフォルト値についての情報を取得します。

### 例 SET パラメータ情報の取得

次のリクエストは、「D」の文字で始まる SET パラメータおよびそれぞれの説明と値を表示します。

```
TABLE FILE SYSSET
PRINT SETDESC CURR_VALUE VALUE IS_DEFAULT
BY SETNAME AS Set,Name
WHERE SETNAME GE 'D' AND SETNAME LT 'E'
ON TABLE SET PAGE NOLEAD
ON TABLE SET STYLE *
GRID=OFF,$
ENDSTYLE
END
```

下図は、出力結果の一部を示しています。

Set Name	Description	Current Value	Available Value	Default
DATEDISPLAY	Date display value when date functions evaluate as zero.	OFF	OFF	1
	Date display value when date functions evaluate as zero.	OFF	ON	0
	Date display value when date functions evaluate as zero.	OFF	COMP	0
DATEFNS	Activates year 2000-compliant versions of date subroutines.	ON	OFF	0
	Activates year 2000-compliant versions of date subroutines.	ON	ON	1
DATEFORMAT	Format of input string in DT() date/time function.	MDY	MDY	1
	Format of input string in DT() date/time function.	MDY	YMD	0
	Format of input string in DT() date/time function.	MDY	DMY	0
	Format of input string in DT() date/time function.	MDY	MYD	0
DATEOUTPUT	Enable locale-sensitive Date Format	DEFAULT	.	.
DATETIME	Sets time and date in reports and controls the format in which CREATE SYNONYM creates date-time columns in a Master File.	STARTUP	STARTUP	0
	Sets time and date in reports and controls the format in which CREATE SYNONYM creates date-time columns in a Master File.	STARTUP	STARTUP/RESET	0
	Sets time and date in reports and controls the format in which CREATE SYNONYM creates date-time columns in a Master File.	STARTUP	RESET	0
	Sets time and date in reports and controls the format in which CREATE SYNONYM creates date-time columns in a Master File.	STARTUP	CURRENT	0
	Sets time and date in reports and controls the format in which CREATE SYNONYM creates date-time columns in a Master File.	STARTUP	CURRENT/NOW	0
DBACSENSITIV	Sets time and date in reports and controls the format in which CREATE SYNONYM creates date-time columns in a Master File.	STARTUP	NOW	0
	Controls whether password validation is case-sensitive.	OFF	OFF	0
DBAJJOIN	Controls whether password validation is case-sensitive.	OFF	ON	0
	Controls how to add VALUE restriction as JOIN condition or to TABLE request.	OFF	OFF	1
DBASOURCE	Controls how to add VALUE restriction as JOIN condition or to TABLE request.	OFF	ON	0
	Controls the source of access restrictions in a multi-file structure.	HOST	HOST	0
	Controls the source of access restrictions in a multi-file structure.	HOST	ALL	0

SYSSQLOP - 関数情報についてのレポート

SYSSQLOP シノニムは、関数および関数の説明、パラメータ、構文、アダプタカテゴリについての情報を取得します。

例 関数の説明および構文の取得

次のリクエストは、名前が「A」と「B」の文字で始まるレガシー関数の名前、説明および構文を取得します。

```
TABLE FILE SYSSQLOP
SUM FUNCTION_DESC FUNCTION_SYNTAX
BY FUNCTION
WHERE CATEGORY LIKE 'L%'
WHERE FUNCTION LE 'C'
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。

Function name	Function description	Function syntax
ARGLEN	Returns the length of a character string, excluding trailing blanks. Use CHAR_LENGTH instead.	ARGLEN(length, string, output_format)
ATODBL	Converts a character string to double-precision format	ATODBL(string, length, output_format)
AYMDI	Add or subtract days to or from a date	AYMDI(arg1, arg2, arg3, arg4)
AYMI	Add or subtract months to or from dates	AYMI(arg1, arg2, arg3, arg4)
BAR	Produce a bar chart	BAR(barlength, infield, maxvalue, 'char', output_format)
BITSON	Determine if a bit is ON or OFF	BITSON(bitnumber, string, output_format)
BITVAL	Evaluate a bit string as a binary integer	BITVAL(string, startbit, number, output_format)
BNYSRC	Get the number of FOR value	BNYSRC(arg1)
BYTVAL	Translates a character to a decimal value	BYTVAL(character, output_format)

SYSTABLE - テーブル情報についてのレポート

SYSTABLE シノニムは、パス内のシノニム情報を取得します。これには、シノニムのタイプ、作成者、カラム数、キー、レコード長、および説明が含まれます。

例 FMI シノニムリストの取得

次のリクエストは、システムテーブルシノニムの名前、説明、および属性を取得します。

```
TABLE FILE SYSTABLE
PRINT TBTYPE REMARKS COLCOUNT RECLENGTH KEYCOLUMNS
BY NAME
WHERE NAME LIKE 'sys%'
WHERE TBTYPE EQ 'FMI'
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。

NAME	TBTYPE	REMARKS	COLCOUNT	RECLENGTH	KEYCOLUMNS
sysacfg	FMI	Metadata: Server admin information	14	2282	0
sysapps	FMI	Metadata: Applications and Files Information	123	51307	0
syscnv	FMI	SQL Adapter: Type Conversion Information	8	846	0
syscolumn	FMI	Metadata: Column Information	121	8376	0
syscube	FMI	Metadata: Cube Information - Hierarchical	66	8763	0
syscube2	FMI	Metadata: Cube Information - Parent/Child	29	2478	0
sysdeffn	FMI	Metadata: DEFINE FUNCTION Information	17	3738	0
sysdirs	FMI	Metadata: File System Directory and File Information	17	1361	0
sysentty	FMI	Metadata: Entity Information - OBSOLETE	10	1100	0
syserr	FMI	Metadata: Error Files Information	12	826	0
sysfiles	FMI	Metadata: Directory information	17	774	0
sysfkeys	FMI	Metadata: Foreign Key Information	11	324	0
sysflow	FMI	Metadata: FOCEXEC Flow Information	91	9997	0
sysimp	FMI	Metadata For FMI Impact	31	3515	0
sysindex	FMI	Metadata: Index Information	42	986	0
syskeys	FMI	Metadata: Key Information	11	630	0
sysrpdrr	FMI	Metadata: Available Stored Procedures	2	22	0
syssscha	FMI	Scheduler: Agent Information	21	2846	0
sysssche	FMI	Scheduler: Event Information	7	704	0
sysset	FMI	Metadata: SET Information	34	2779	0
sysssqlop	FMI	Functions Optimization Information for SQL Adapters	23	4090	0
systable	FMI	Metadata: Table Information	40	1149	0

## データタイプについてのレポート

SYSVDTP シノニムは、SQL アダプタおよび固定フォーマットのシーケンシャルデータソースのデータタイプ、および対応する USAGE および ACTUAL フォーマットを取得します。これは FMI シノニムではありませんが、区切りシーケンシャルファイルからデータタイプ情報を取得します。

## 例 MySQL アダプタのデータタイプの取得

次のリクエストは、MySQL アダプタのデータタイプ情報を取得します。

```
TABLE FILE SYSVDT
PRINT DATA_TYPE_CATEGORY
VENDOR_DATA_TYPES
TYPE_RANGE
SERVER_USAGE_CATEGORY
SERVER_USAGE
SERVER_ACTUAL
REMARKS
BY ADAPTER
WHERE SUFFIX EQ 'SQLMYSQL'
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。

Adapter	Data Type Category	Vendor Data Types	Type Range	Server Data Type	Server USAGE	Server ACTUAL	Remarks
MySQL	Date-Time	DATE		Date-Time	YYMD	DATE	
	Date-Time	TIME		Date-Time	HHIS	HHIS	
	Date-Time	TIMESTAMP/DATETIME		Date-Time	HYMDMS	HYMDMS	
	Date-Time	TIMESTAMP WITH TIME ZONE			N/A	N/A	
	Date-Time	YEAR		Numeric	I6	I4	
	Numeric	TINYINT/SMALLINT		Numeric	I6	I2	
	Numeric	TINYINT(1)/BIT/BOOL		Numeric	I11	I2	
	Numeric	INTEGER/MEDIUMINT		Numeric	I11	I4	
	Numeric	INTEGER UNSIGNED		Numeric	P11	P8	
	Numeric	BIGINT		Numeric	P20	P11	
	Numeric	DECIMAL(p,s)/NUMERIC(p,s)	p=1..31,s=0	Numeric	Pn	Pk	n=p+1,k=(p/2)+1
	Numeric	DECIMAL(p,s)/NUMERIC(p,s)	p=1..31,s>0	Numeric	Pn.m	Pk	n=p+2,m=min(s,31),k=(p/2)+1
	Numeric	DECIMAL(p,s)/NUMERIC(p,s)	p>32,s=0	Numeric	P32	P16	
	Numeric	DECIMAL(p,s)/NUMERIC(p,s)	p>32,s>0	Numeric	P33.m	P16	m=min(s,31)
	Numeric	REAL/FLOAT/DOUBLE PRECISION		Numeric	D20.2	D8	
	LOB and Other	TEXT/MEDIUMTEXT/LONGTEXT		Text	TX50	TX	
	LOB and Other	TINYTEXT		Alphanumeric	A255V	A255V	
	LOB and Other	BLOB/TINYBLOB/LONGBLOB		BLOB	BLOB	BLOB	
	LOB and Other	VARBINARY(n)	n>16383	BLOB	BLOB	BLOB	
	LOB and Other	ENUM		Alphanumeric	An	An	
	LOB and Other	SET		Alphanumeric	An	An	MySQL JDBC Connector version has to be 5.1.7 or higher
	LOB and Other	GEOMETRY		Text	TX50	TX	Server supports this type as read-only via spatial properties: - Master File: GEOGRAPHIC_ROLE=GEOMETRY_AREA; - Access File: SQL_FLD_OBJ_TYPE=OPAQUE, SQL_FLD_OBJ_PROP=GEOMETRY_SHAPE, SQL_FLD_OBJ_EXPR=DB_EXPR() where DB_EXPR() are vary depends on DBMS
	LOB and Other			Text			
	LOB and Other			Text			
	LOB and Other			Text			
	LOB and Other			Text			
	Character	CHAR(n)		Alphanumeric	An	An	
	Character	VARCHAR(n)		Alphanumeric	AnV	AnV	n>32765 will be truncated
	Character	BINARY(n)		Alphanumeric	Am	Am	m=2*n
	Character	VARBINARY(n)	n<=16383	Alphanumeric	Am	Am	m=2*n
	Character, Unicode	CHAR(n)		Alphanumeric	An	An	
	Character, Unicode	VARCHAR(n)		Alphanumeric	AnV	AnV	n>32765 will be truncated



## FOCUS データソースへのアクセス

---

ここでは、USE コマンドを使用して FOCUS データソースにアクセスする方法について説明します。USE コマンドのいくつかの機能は、マスターファイルの DATASET 属性とともに実行することができます。

ここでは、Windows、UNIX の各オペレーティングシステムでの情報および事例を紹介します。

また、ユニバーサル連結を使用し、1 つのリクエストで複数の異なるデータソースのデータを取得することもできます。

外部インデックスを使用するには、USE コマンドを発行する必要があります。

### トピックス

- ☐ [USE コマンド](#)
  - ☐ [FOCUS データソースの指定](#)
  - ☐ [異なるファイル名の指定](#)
  - ☐ [新しいデータソースの指定](#)
  - ☐ [データソースの保護](#)
  - ☐ [データソースの連結](#)
  - ☐ [現在有効な USE オプションの表示](#)
  - ☐ [USE オプションのクリア](#)
- 

### USE コマンド

TABLE FILE filename などのコマンドを発行して FOCUS データソースにアクセスする場合、WebFOCUS は指定されたファイル名のマスターファイルを検索し、次にそのファイルと同一名のデータソースを検索します。USE コマンドは、FOCUS データソースの名前と場所を指定します。そのため、USE コマンドは次のような場合に役立ちます。

- ☐ デフォルトの命名規則が使用されていない場合。
- ☐ データソースが標準の検索パス上に存在しない場合。

- ❑ 特別なオプションを明示的に指定する必要がある場合。

USE コマンドを使用して FOCUS データソースを指定すると、USE ディレクトリが作成されます。データソース定義のリストは、このディレクトリに格納されます。USE ディレクトリが有効になっている場合、WebFOCUS はデフォルトの名前と検索パスを使用する代わりに、USE ディレクトリ内の情報を使用してデータソースの場所を特定します。USE ディレクトリを使用することにより、最大で 1022 件のデータソースにアクセスすることが可能になります。なお、このディレクトリは、FOCUS データソースにのみ適用されます。USE ディレクトリの作成は、次の方法で簡略化することができます。

- ❑ プロシジャが USE ディレクトリにアクセスする必要がある場合、プロシジャの先頭に - INCLUDE コマンドを使用し、呼び出し対象のすべての USE を 1 つのファイルに追加します。
- ❑ WebFOCUS Reporting Server プロファイルにすべての USE を追加します。このプロファイルが開始されると、この USE コマンドが有効になります。

マスターファイルの DATASET 属性は、USE コマンドが実行可能なタスクをすべて実行することはできませんが、その一部を実行することができます。

### 参照

### その他の TIBCO FOCUS データソースファイル指定

オペレーティングシステムの中には、FOCUS データソースの指定に、その他のファイルを使用するものがあります。

- ❑ Windows および UNIX では、FOCUS データソースにはデフォルトのファイル拡張子である .FOC が付けられます。たとえば、TABLE FILE EMPLOYEE コマンドは、EMPLOYEE.FOC を使用します。
- ❑ TABLE FILE EMPLOYEE コマンドは、ddname の EMPLOYEE に割り当てられたデータソースを使用します。

## FOCUS データソースの指定

FOCUS データソースを指定するには、USE コマンドを使用します。

1 回の USE コマンドの使用で 1 つまたは複数のデータソースを指定することができます。詳細は、479 ページの「[単一データソースの指定](#)」および 479 ページの「[複数データソースの指定](#)」を参照してください。

構文      USE コマンドによるデータソースの指定

```
USE {ADD|CLEAR|REPLACE}
fileid [READ|NEW] [AS mastername] [ON resource]
[fileid [READ|NEW] [AS mastername] [ON resource]]
.
.
.
END
```

説明

ADD

現在のディレクトリに 1 つまたは複数の新しいファイル ID を追加します。ADD オプションを使用せずに USE コマンドを発行した場合、現在の USE ディレクトリが、指定したデータソースのリストで置換されます。

CLEAR

USE ディレクトリをクリアします。指定した他のオプションはすべて無視されます。このオプションには END コマンドを使用する必要はありません。

REPLACE

USE ディレクトリ内の既存のファイル ID を置換します。このオプションを使用すると、ファイル ID およびその ID の後に続くオプションを変更することができます。

fileid

オペレーティングシステムで有効なファイル名です。

下表は、オペレーティングシステムごとの説明です。

オペレーティングシステム	説明
Windows:	drive:¥path¥filename.ext フォーマットで指定する有効なファイル名です。ドライブ名およびパスはオプションです。   <b>注意:</b> アプリケーションネームスペーステクノロジーを使用している場合は、フルパスを指定する必要はありません。app_name[/app_name_a]/filename.ext フォーマットで指定する有効なファイル名です (例、ibisamp/ggsales.foc)。

オペレーティング システム	説明
UNIX:	/directory/filename.ext フォーマットで指定する有効なファイル名です。ディレクトリはオプションとして指定します。   <b>注意:</b>アプリケーションネームスペーステクノロジーを使用している場合は、フルパスを指定する必要はありません。 <code>app_name[/app_name_a]/filename.ext</code> フォーマットで指定する有効なファイル名です (例、ibisamp/ggsales.foc)。

READ

データソースへのアクセスを読み取り専用にします。

NEW

新しいデータソースであることを表します。

AS mastername

ファイル ID に関連付けられたマスターファイルの名前を指定します。

READ および NEW の両オプションは、AS オプションとともに使用することができます。ファイル ID の後にいくつかのオプションの組み合わせを追加しても無効です。

**注意:** ON 句で指定した FOCUS Database Server 上のファイルに対して AS 句を使用する場合は、READ パラメータが必要です。これにより、ファイルの作成時に関連付けられたマスターファイル以外のマスターファイルでファイルにアクセスする際に、ファイルへの書き込みアクセスが防止されます。この場合、READ アクセスのみが許可されます。

ON resource

複数のエンドユーザが使用する FOCUS データソースを同期するよう特別に構成した WebFOCUS Reporting Server を指定します。

**注意:** ON 句で指定した FOCUS Database Server 上のファイルに対して AS 句を使用する場合は、READ パラメータが必要です。これにより、ファイルの作成時に関連付けられたマスターファイル以外のマスターファイルでファイルにアクセスする際に、ファイルへの書き込みアクセスが防止されます。この場合、READ アクセスのみが許可されます。

例          単一データソースの指定

下表は、データソースの指定に使用できる USE コマンドの使用例を示しています。

オペレーティングシステム      USE コマンドの使用例	
Windows:	USE EMP026.FOC AS PRODUCTS END
UNIX:	USE emp026.foc AS PRODUCTS END

例          複数データソースの指定

1 回の USE コマンドで、複数のデータソースを指定することができます。この場合、それぞれのデータソースに対して異なるパラメータを使用します。下表は、USE コマンドの使用例を示しています。

オペレーティングシステム      USE コマンドの使用例	
Windows:	USE EMP024.FOC ON MULTID EMP025.FOC AS PRODUCTS EMP026.FOC READ AS ACCOUNTS END
UNIX:	USE emp024.foc ON MULTID emp025.foc AS PRODUCTS emp026.foc READ AS ACCOUNTS END

異なるファイル名の指定

次のような場合は、FOCUS データソースを指定する際に、USE コマンドを発行して異なるファイル名を指定します。

- ❑ データソースの名前が、それに関連付けられたマスターファイル名と異なる場合。

- ❑ データソースのタイプまたは拡張子が、デフォルトのタイプまたは拡張子と異なる場合。
- ❑ データソースが格納されているディレクトリが、標準の検索パスと異なる場合。

**注意:** USE コマンドを発行して値を変更しない限り、マスターファイルの LOCATION 属性が名前を付けたデータソースにはデフォルト値が割り当てられます。物理ファイルの名前またはパスがデフォルト値と異なる場合は、そのファイルを USE ディレクトリに格納する必要があります。

構文 異なるファイル名の指定

```
USE
fileid AS mastername END
```

説明

**fileid**  
オペレーティングシステムで有効なファイル名です。

**mastername**  
ファイル ID に関連付けられたマスターファイルの名前です。

例 異なるファイル名の指定

EMPLOYEE マスターファイルで定義した EMP026 データソースを読み取るには、USE コマンドを発行します。これにより、TABLE FILE EMPLOYEE コマンドによる EMP026 データソースの読み取りが可能になります。

オペレーティングシステム      USE コマンドの使用例	
Windows:	USE EMP026.FOC AS EMPLOYEE END
UNIX:	USE emp026.foc AS EMPLOYEE END

## 例 異なるファイルタイプまたは拡張子の指定

### Windows:

EMPLOYEE マスターファイルで定義した EMP026 データソース (拡張子 .dat) を読み取るには、次のコマンドを発行します。

```
USE
EMP026.DAT AS EMPLOYEE
END
```

### UNIX:

EMPLOYEE マスターファイルで定義した EMP026 データソース (拡張子 .dat) を読み取るには、次のコマンドを発行します。

```
USE
emp026.dat AS EMPLOYEE
END
```

USE コマンドを発行後、TABLE FILE EMPLOYEE コマンドで EMP026 を読み取ることができます。

## 例 異なるファイルパスの指定

### Windows:

EMPLOYEE マスターファイルで定義し、C:¥ACCOUNTING¥ ディレクトリに格納された EMP026 データソースを読み取るには、次のコマンドを発行します。

```
USE
C:¥ACCOUNTING¥EMP026.FOC AS EMPLOYEE
END
```

### UNIX:

EMPLOYEE マスターファイルで定義し、/accounting/ ディレクトリに格納された emp026 データソースを読み取るには、次のコマンドを発行します。

```
USE
/accounting/emp026.foc AS EMPLOYEE
END
```

## 新しいデータソースの指定

新しいデータソースを指定することができます。新しいデータソースを指定する場合、デフォルトのファイル名をそのまま使用するか、異なる名前を付けることができます。

### 構文 新しいデータソースの指定

```
USE
fileid NEW
END
```

説明

**fileid**

オペレーティングシステムで有効なファイル名です。データソースを作成すると、そのデータソースにファイル ID が割り当てられます。

**NEW**

このデータソースがまだ存在しないことを指定します。NEW パラメータを省略すると、データソースが見つからないことを告げるメッセージが返されます。この場合、USE コマンドは実行されません。

### 例 新しいデータソースの指定

次のコマンドは、WAGES マスターファイルを使用して WAGES データソースを作成します。

**Windows:**

```
USE
C:¥DATA¥WAGES.FOC NEW
END
CREATE FILE WAGES
```

**UNIX:**

```
USE
wages.foc NEW
END
CREATE FILE WAGES
```

## データソースの保護

USE コマンドを発行する際に READ パラメータを使用して、データソースの変更を防止することができます。保護されたデータソースは変更することはできませんが、さまざまなツールおよびコマンドで読み取ることができます。

### 構文 データソースの保護

```
USE
fileid READ
END
```

説明

`fileid`

オペレーティングシステムで有効なファイル名です。

**READ**

データソースへのアクセスを読み取り専用にします。

例

## データソースの保護

次のコマンドは、EMPLOYEE データソースの変更を防止します。

**Windows:**

```
USE
EMPLOYEE.FOC READ
END
```

**UNIX:**

```
USE
/usr/mydata/employee.foc READ
END
```

## データソースの連結

複数の FOCUS データソースが同一のマスターファイルで定義されている場合、これらのデータソースを連結する USE コマンドを発行し、1つのリクエストですべてのデータソースを読み取ることができます。

また、共通ファイルをクロスリファレンスとして使用する複数のファイルを連結することもできます。この連結を実行するには、USE コマンドでホストファイルを指定してから、クロスリファレンスファイルを指定します。

**注意：**連結したファイルを JOIN コマンドのクロスリファレンスファイルとして指定することはできません。

構文

## データソースの連結

```
USE
fileid-1 AS mastername
fileid-2 AS mastername
.
.
fileid-n AS mastername
END
```

### 説明

`fileid-1...fileid-n`

連結する一連のファイルに使用する有効なファイル名です。

`mastername`

データソースを定義するマスターファイルの名前です。

### 例

#### データソースの連結

次のコマンドは、FOCUS データソースの EMP024、EMP025、EMP026 を連結します。これらのデータソースはすべて EMPLOYEE マスターファイルで定義されています。次に、TABLE FILE EMPLOYEE コマンドを 1 回発行し、これら 3 つのデータソースをまとめて読み取ります。

#### Windows:

```
USE
C:¥DATA¥EMP024.FOC AS EMPLOYEE
C:¥DATA¥EMP025.FOC AS EMPLOYEE
C:¥DATA¥EMP026.FOC AS EMPLOYEE
END
```

#### UNIX:

```
USE
/usr/mydata/emp024.foc AS EMPLOYEE
/usr/mydata/emp025.foc AS EMPLOYEE
/usr/mydata/emp026.foc AS EMPLOYEE
END
```

### 例

#### 複数連結の指定

次のコマンドは、EMPLOYEE マスターファイルで定義した EMP01 および EMP02 データソースを連結し、SALES マスターファイルで定義した SALES01 および SALES02 データソースを連結します。

#### Windows:

```
USE
EMP01.FOC AS EMPLOYEE
EMP02.FOC AS EMPLOYEE
SALES01.FOC AS SALES
SALES02.FOC AS SALES
END
```

**UNIX:**

```
USE
emp01.foc AS EMPLOYEE
emp02.foc AS EMPLOYEE
sales01.foc AS SALES
sales02.foc AS SALES
END
```

TABLE FILE EMPLOYEE コマンドを使用し、EMP01 および EMP02 データソースを読み取ることができます。TABLE FILE SALES コマンドを使用し、SALES01 および SALES02 データソースを読み取ることができます。

**構文      データソースと単一クロスリファレンスファイルの連結**

```
USE
fileid-1 AS mastername
fileid-2 AS mastername
.
.
fileid-n AS mastername
reffileEND
```

**説明**

**fileid-1...fileid-n**

連結する一連のファイルに使用する有効なファイル名です。

**mastername**

データソースを定義するマスターファイルの名前です。

**reffile**

連結する一連のファイルに使用するクロスリファレンスファイルです。

**例      データソースと単一クロスリファレンスファイルの連結**

EMPLOYEE は「EMP01」および「EMP02」という 2 つのファイルで構成され、両ファイルには「EDUCFILE」という共通クロスリファレンスファイルがあります。USE コマンドを発行し、これらのファイルをまとめて読み取ります。

**Windows:**

```
USE
EMP01.FOC AS EMPLOYEE
EMP02.FOC AS EMPLOYEE
EDUCFILE.FOC
END
```

### UNIX:

```
USE
/usr/mydata/emp01.foc AS EMPLOYEE
/usr/mydata/emp02.foc AS EMPLOYEE
/usr/mydata/educfile.foc
END
```

## 例

### データソースと複数クロスリファレンスファイルの連結

EMPLOYEE が「EMP01」および「EMP02」という 2 つのファイルで構成され、それぞれのファイルに「ED01」および「ED02」という異なるクロスリファレンスファイルがある場合、1 つのコマンドでこれらの 4 つのファイルをまとめて読み取ることができます。USE コマンドを発行し、それぞれのホストファイルの後にそれに対応するクロスリファレンスファイルを指定します。

### Windows:

```
USE
EMP01.FOC AS EMPLOYEE
ED01.FOC AS EDUCFILE
EMP02.FOC AS EMPLOYEE
ED02.FOC AS EDUCFILE
END
```

### UNIX:

```
USE
/usr/mydata/emp01.foc AS EMPLOYEE
/usr/mydata/ed01.foc AS EDUCFILE
/usr/mydata/emp02.foc AS EMPLOYEE
/usr/mydata/ed02.foc AS EDUCFILE
END
```

## 現在有効な USE オプションの表示

現在有効になっている USE オプションを表示するには、? USE クエリを発行します。このクエリは、USE コマンドで指定したデータソースのリストおよび現在有効になっているオプションを表示します。

## 構文

### 現在有効な USE オプションの表示

```
? USE
```

**例           現在有効な USE オプションの表示**

? USE コマンドの結果は次のようになります。

```
DIRECTORIES IN USE ARE:
C:¥IBI¥apps¥ibisamp¥CENTINV.FOC AS CENTINV
C:¥IBI¥apps¥ibisamp¥CENTHR.FOC AS CENTHR
```

**USE オプションのクリア**

USE コマンドを使用して、USE ディレクトリをクリアすることができます。

**構文           USE ディレクトリのクリア**

```
USE CLEAR
```



# 8

## 環境のカスタマイズ

---

SET コマンドを使用して、WebFOCUS 環境を制御するパラメータを変更することができます。SET コマンドのパラメータは、レポートおよびグラフのコンテンツ、画面表示や印刷時の外観を制御します。これらのパラメータは、エンドユーザからのリクエストに対するシステムレスポンス、パフォーマンスに影響を与えるデータ検索の特性も制御します。また、メタデータの設定や日付などの情報の操作にも SET コマンドを使用することができます。

SET コマンドは、開発においての重要な要素を多面的に制御するコマンドです。

### トピックス

- ❑ [SET コマンドの使用](#)
  - ❑ [SET コマンドの発行方法](#)
  - ❑ [SET コマンドのコーディング](#)
  - ❑ [SET パラメータのタイプ](#)
  - ❑ [SET パラメータ構文](#)
- 

## SET コマンドの使用

アプリケーション開発者は、次のような場合に SET コマンドを使用します。

- ❑ 開発作業を効率的に行い、テストおよびデバッグのさまざまな要求に対応する。
- ❑ デフォルト設定を変更する必要のない安定した最適な実行環境をエンドユーザに提供する。

独自の adhoc レポートを作成する場合は、個別のニーズに合わせてレポートの外観やコンテンツをカスタマイズする際に SET コマンドを使用します。

### SET コマンドの発行方法

SET コマンドを発行するには、SET ツールを使用してストアードプロシジャコマンドを生成するか、ストアードプロシジャ内にコーディングします。SET ツールを使用すると、開発中のレポートに SET コマンドが生成されます。SET コマンドをコーディングする場合は、以下のオプションを使用することができます。

- ❑ WebFOCUS Reporting Server 上でサポートされたプロファイルで発行する。プロファイルで発行する場合、現在のプロシジャを実行するための接続が確立されている間は、コマンドの設定がサーバ環境に適用されます。
- ❑ プロシジャで発行する。
  - ❑ レポートリクエストの前に発行する。
  - ❑ レポートリクエストまたはグラフィクエストで ON TABLE SET または ON GRAPH SET と併せて発行する。プロシジャに複数のレポートリクエストまたはグラフィクエストが存在する場合でも、この方法による SET コマンドは、このコマンドが発行された特定のレポートリクエストまたはグラフィクエストにのみ適用されます。
- ❑ BI Portal 管理者環境の [その他] レポートオブジェクトで発行する。[その他] レポートオブジェクトで発行された設定は、1 つのレポートテンプレートに適用されます。
- ❑ WebFOCUS スクリプトファイルである site.wfs で発行する。site.wfs ファイルで発行すると、設定は WebFOCUS Client から WebFOCUS Reporting Server へのすべての接続に適用されます。

### 参照

### SET コマンドの優先順位

同一のパラメータに対して複数の SET コマンドを発行する場合、次のように優先順位が決定されます。

#### WebFOCUS

1. プロシジャ内の SET
2. site.wfs または ibidir.wfs ファイル内の SET
3. WebFOCUS Reporting Server プロファイル内の SET
  - a. グローバルプロファイル
  - b. グループプロファイル
  - c. ユーザプロファイル

## SET コマンドのコーディング

SET コマンド構文には、次のガイドラインが適用されます。

- ❑ 1 つのコマンドに複数のパラメータを設定することができます。その場合は、各パラメータをカンマ (,) で区切ります。
- ❑ 1 行に記述できるパラメータ数に制限はありません。1 行に収まらない場合は、新しい行に再度 SET コマンドを発行します。
- ❑ パラメータの多くは、リクエスト内の ON TABLE SET または ON GRAPH SET を使用して設定することができます。この方法で設定できないパラメータは、詳細情報の項目で説明しています。

パラメータに固有の構文および有効な値についての詳細は、517 ページの「[SET パラメータ構文](#)」を参照してください。

### 構文      パラメータの設定

```
SET parameter = option[, parameter = option,...]
```

説明

**parameter**

変更する設定です。

**option**

パラメータの有効な値です。

### 例      単一パラメータの設定

次の例は、PAGE-NUM パラメータを使用してデフォルトのページ番号を非表示にする方法を示しています。

```
SET PAGE-NUM = OFF
```

```
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
END
```

出力結果は次のとおりです。

<u>LAST_NAME</u>	<u>FIRST_NAME</u>
------------------	-------------------

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

注意：この例の出力には、罫線を非表示にするフォーマット設定が適用されています。

例 複数パラメータの設定

次の例は、プロシジャで 1 つのコマンドを使用して 2 つのパラメータを設定する方法を示しています。最初の NODATA パラメータは、ミッシングデータに使用するデフォルト文字をピリオド (.) から「NONE」に変更します。2 つ目の PAGE-NUM パラメータは、デフォルトのページ番号を非表示にします。

```
SET NODATA = NONE, PAGE-NUM = OFF
TABLE FILE EMPLOYEE
PRINT CURR_SAL BY EMP_ID
ACROSS DEPARTMENT
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
END
```

この出力では、給与情報が存在しない従業員には NONE が表示されます。その従業員が、ここで参照する部署に勤務していないためです。また、出力の上部にページ番号は表示されません。

出力結果は次のとおりです。

DEPARTMENT		
EMP_ID	MIS	PRODUCTION
071382660	NONE	\$11,000.00
112847612	\$13,200.00	NONE
117593129	\$18,480.00	NONE
119265415	NONE	\$9,500.00
119329144	NONE	\$29,700.00
123764317	NONE	\$26,862.00
126724188	NONE	\$21,120.00
219984371	\$18,480.00	NONE
326179357	\$21,780.00	NONE
451123478	NONE	\$16,100.00
543729165	\$9,000.00	NONE
818692173	\$27,062.00	NONE

注意：この例の出力には、罫線を非表示にするフォーマット設定が適用されています。

## 構文 レポートリクエストでのパラメータの設定

```
ON TABLE SET parametervalue [AND parametervalue ...]
```

説明

**parameter**

変更する設定です。

**value**

パラメータの有効な値です。

## 例 レポートリクエストでのパラメータの設定

次の例は、ON TABLE SET コマンドを使用してミッシングデータのデフォルト文字をピリオド (.) から「NONE」に変更し、ページ番号を非表示にする方法を示しています。

```
TABLE FILE EMPLOYEE
PRINT CURR_SAL BY EMP_ID
ACROSS DEPARTMENT
ON TABLE SET NODATA NONE AND PAGE-NUM OFF
ON TABLE SET STYLE *
TYPE=REPORT, GRID=OFF,$
END
```

この出力では、給与情報が存在しない従業員には NONE が表示されます。また、出力の上部にページ番号は表示されません。

出力結果は次のとおりです。

	DEPARTMENT	
EMP_ID	MIS	PRODUCTION
071382660	NONE	\$11,000.00
112847612	\$13,200.00	NONE
117593129	\$18,480.00	NONE
119265415	NONE	\$9,500.00
119329144	NONE	\$29,700.00
123764317	NONE	\$26,862.00
126724188	NONE	\$21,120.00
219984371	\$18,480.00	NONE
326179357	\$21,780.00	NONE
451123478	NONE	\$16,100.00
543729165	\$9,000.00	NONE
818692173	\$27,062.00	NONE

注意：この例の出力には、罫線を非表示にするフォーマット設定が適用されています。

## 構文 グラフクエストでのパラメータの設定

```
ON GRAPH SET parametervalue [AND parametervalue ...]
```

説明

**parameter**

変更する設定です。

**value**

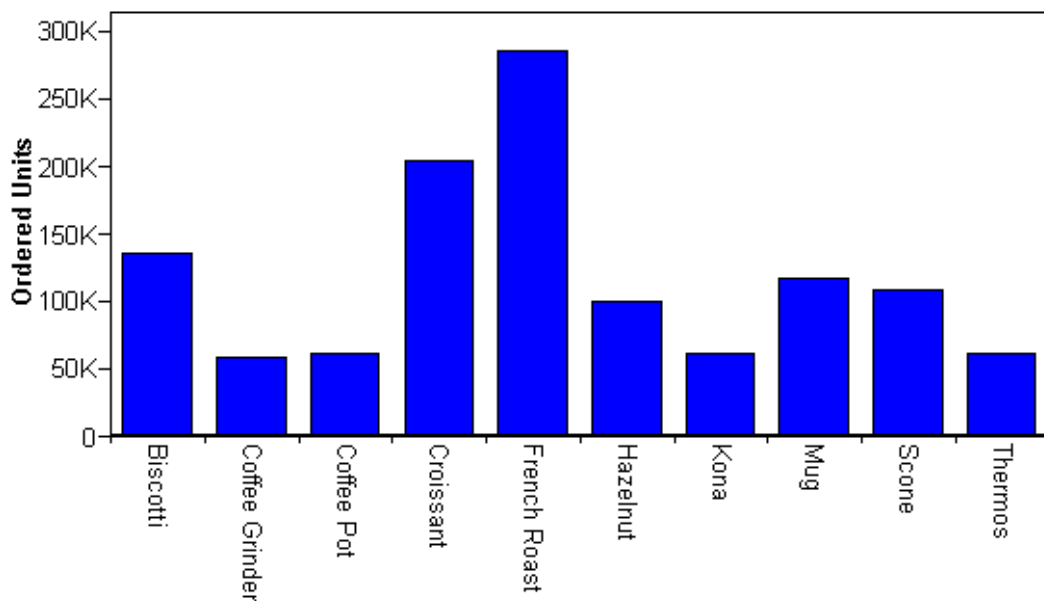
パラメータの有効な値です。

## 例 グラフクエストでのパラメータの設定

次の例は、ON GRAPH SET コマンドを使用して、3D パラメータのデフォルト設定を OFF に変更する方法を示しています。

```
GRAPH FILE GGORDER
SUM QUANTITY
ACROSS PRODUCT_DESC
ON GRAPH SET 3D OFF
END
```

出力結果は次のとおりです。



## SET パラメータのタイプ

ここでは、実行可能なタスクタイプおよび各タスクタイプを実行するための SET パラメータについて説明します。1 つのパラメータが複数の実行タスクに該当する場合は、それぞれのカテゴリに表示されます。各パラメータの説明および構文についての詳細は、517 ページの「[SET パラメータ構文](#)」を参照してください。

下表は、SET パラメータを使用して実行できるタスクタイプの一覧です。

496 ページの「 <a href="#">演算</a> 」	FOCUS で実行する演算方法に影響します。
497 ページの「 <a href="#">データとメタデータ</a> 」	データの格納方法および処理方法を指定します。
499 ページの「 <a href="#">Data Migrator タスク</a> 」	データフローのデータ処理を制御します。
500 ページの「 <a href="#">日付操作タスク</a> 」	レポート内での日付の処理方法および表示方法を制御します。
501 ページの「 <a href="#">WebFOCUS 固有のタスク</a> 」	アプリケーションで Web 固有のタスクを実行する際の動作に影響します。
502 ページの「 <a href="#">グラフタスク</a> 」	グラフの処理および表示を制御します。
504 ページの「 <a href="#">メモリ設定および最適化タスク</a> 」	アプリケーションのメモリおよび最適化に影響します。
505 ページの「 <a href="#">レポートコード、コンテンツ、処理のタスク</a> 」	リクエストの内容および処理方法を指定します。
510 ページの「 <a href="#">レポートレイアウトと表示タスク</a> 」	レポートの表示に影響します。

516 ページの「 <a href="#">セキュリティタスク</a> 」	データソースおよびプロシジャへのユーザアクセスを制御します。
---------------------------------------	--------------------------------

演算

次のパラメータは、WebFOCUS で実行する演算の動作を制御します。

CDN

数字の表記に使用する区切り文字を指定します。

COMPUTE

式のコンパイルを制御します。

DMPRECISION

ダイアログマネージャの -SET コマンドで正確な数値変数の値を計算する際の数値精度を指定します。

FLOATMAPPING

浮動小数点数のすべての数値処理にパック 10 進数ベースの桁数を使用します。

MISS\_ON

一時項目 (DEFINE) および一時項目 (COMPUTE) に、SOME、ALL、MISSING ON のいずれかでデフォルト値を設定します。

MISSINGTEST

IF-THEN-ELSE テストの IF 式でミッシング値のテストを行うかどうかを指定します。

NEG-ZERO

負の小数値を端数処理した結果の 0 (ゼロ) を負の値として表示します。

PARTITION\_ON

統計関数のパーティションサイズを設定します。

USERFCHK

DEFINE FUNCTION で作成した関数の引数および、WebFOCUS が提供する関数の引数 (WF)、FOCUS が提供する関数の引数 (FOCUS) に適用する検証レベルを制御します。

USERFNS

WebFOCUS に付属の関数 (WF)、FOCUS に付属の関数 (FOCUS)、またはローカルで定義した同一名の関数のどちらを使用するかを指定します。

%STRICTMATH

WebFOCUS で、値を 0 (ゼロ) で除算した場合の処理方法を定義します。

## データとメタデータ

次のパラメータは、データの格納方法と処理方法を指定します。

### ACCBLN

マスターファイルで ACCEPT コマンドを指定したフィールドに、ブランクまたは 0 (ゼロ) の値を受容します。

### ASNAMES

HOLD マスターファイルの FIELDNAME 属性を制御します。

### COUNTWIDTH

COUNT フィールドのデフォルトフォーマットを整数の 5 バイトから整数の 9 バイトに拡張します。

### DATEFORMAT

日付の値をフォーマット済みまたは変換済み文字列フォーマットで入力した際の、日付要素 (月/日/年) の順序を指定します。

### DEFINES

パフォーマンスを向上させるために一時項目 (DEFINE) をマシンコードにコンパイルします。

### DIRECTHOLD

FOCUS フォーマットの HOLD ファイルを直接作成するかどうかを制御します。

### DTSTRICT

日付時間フィールドに対する厳密処理の使用を制御します。

### EQTEST

選択条件で「\$」および「\$\*」の記号をワイルドカード文字として扱うか、通常の文字として扱うかを制御します。

### EUROFILE

変換する通貨の情報が格納されたデータソースを有効にします。

### FIELDNAME

長い修飾フィールド名の使用を制御します。

### FOCHTMLURL

/ibi\_apps/ibi\_html 以外のエイリアスを使用したりソースへのアクセスを可能にします。

### HLDCOM\_TRIMANV

区切りフォーマットで出力した場合に末尾のグランクを保持するかどうかを制御します。

### HNODATA

HOLD ファイルに継承するミッシング値を制御します。

### **HOLDFORMAT**

HOLD ファイルのデフォルトフォーマットを指定します。

### **HOLDLIST**

レポートリクエストのフィールドの中で、HOLD ファイルに格納するフィールドを指定します。

### **HOLDMISS**

HOLD ファイル内で、ミッシングデータとデフォルトデータ (ゼロまたはブランク) の両者を区別します。

### **HOLDSTAT**

コメントおよび DBA 情報を HOLD マスターファイルに含めるかどうかを指定します。

### **HTMLARCHIVE**

HTML および DHTML レポートを、イメージファイルとともに単一 Web アーカイブドキュメント (.mht ファイル) にパッケージ化します。

### **HTMLENCODE**

HTML 出力内のデータをエンコードします。

### **KEEPDEFINES**

JOIN コマンドを実行後、ホストまたは JOIN 構造用に作成した一時項目 (DEFINE) を保持するかどうかを制御します。

### **MAXDATAEXCPT**

セッション終了時までには発生可能なデータ例外の個数を変更します。

### **MDICARDWARN**

ディメンションのカーディナリティが特定の値を超えた場合に警告メッセージを表示します。

### **MDIENCODING**

データソースを読み込まずに、MDI ファイルから出力結果を取得できるようにします。

### **MDIPROGRESS**

MDI ビルドの進行状況に関するメッセージを表示します。

### **NULL**

ミッシング値とブランク文字列または 0 (ゼロ) 値の両者を区別するため、可変長カンマ区切りまたはタブ区切りの HOLD ファイルを作成できるようにします。

### **OLDSTYRECL**

レコード長の LRECL を、現在の設定である LRECL=0 または以前の設定である LRECL=512 のどちらに設定するかを指定します。

**PCOMMA**

PC アプリケーションまたは HOLD FORMAT COM コマンドが作成するカンマ区切りファイルを取得できるようにします。

**QUALCHAR**

修飾フィールド名に使用する修飾文字を指定します。

**RANK**

複数のデータ値が同一ランキングに分類される際のリクエストでのランキング番号の割り当て方法を指定します。

**SAVEDMASTERS**

リクエストで使用したマスターファイルをメモリに保存します。

**SHADOW**

シャドウライティング機能を有効にします。

**SHIFT**

シフト文字列の使用を制御します。

**TEMPERASE**

WebFOCUS の接続を閉じた後で、接続中に作成された一時ファイルを保持するかどうかを指定します。

**WEBARCHIVE**

複数の EXL2K ファイルを単一ファイルにパッケージ化します。

**WEEKFIRST**

週の開始日とする曜日を指定します。

**WPMINWIDTH**

WP フォーマットの出力ファイルの最小幅を指定します。

**XRETRIEVAL**

レポートのプレビュー時のデータ取得を制御します。

**XFOCUSBINS**

XFOCUS データソースのバッファとして使用するメモリのページ数を定義します。

## Data Migrator タスク

次のパラメータは、データフローのデータ処理を制御します。

**CNVERR**

データフローでのフォーマット変換エラーを拒否するか、デフォルト値を使用して処理するかを制御します。

### CNVERRLIMIT

データフローのフォーマット変換エラー数の上限を設定します。

## 日付操作タスク

次のパラメータは、レポートに表示する日付の処理方法および表示方法を制御します。

### BUSDAYS

営業日および営業日以外の日付を指定します。

### DATE\_ORDER

日付構成要素の順序を指定します。

### DATE\_SEPARATOR

日付構成要素に使用する区切り文字を指定します。

### DATEDISPLAY

日付フォーマットのフィールドに 0 (ゼロ) の値が含まれる場合の表示を制御します。

### DATEFNS

2000 年対応バージョンの日付サブルーチンを有効にします。

### DATETIME

レポートの日付と時間を設定します。また、CREATE SYNONYM がマスターファイルに作成する日付時間フィールドのフォーマットを制御します。

### DEFCENT

アプリケーションで使用するデフォルトの西暦を定義します。

### EXL2KTXDATE

変換された日付をテキスト値ではなく、フォーマットマスクが適用された日付値として送信するかどうかを制御します。

### FILECASE

UNIX および WebFOCUS では、大文字、小文字、大文字小文字混在のファイル名およびディレクトリがサポートされます。UNIX 環境のデフォルト設定が小文字であるため、WebFOCUS が自動的にファイル名およびディレクトリを小文字に変換します。

### HDAY

休日として指定された日付を取得するための休日ファイルを指定します。

### LEADZERO

先頭ゼロの切り捨てを回避します。

### SUPPRESSDRILLDT

日付時間パラメータのドリルダウンで DT 関数を追加しません。

### TIME\_SEPARATOR

&TOD 変数の時間構成要素に使用する区切り文字を指定します。

**TESTDATE**

一時的にシステムの日付を変更して動的ウィンドウをテストします。

**YRTHRESH**

世紀の開始を定義します。

**WebFOCUS 固有のタスク**

次のパラメータは、WebFOCUS で実行する Web 固有のタスクの動作に影響します。

**&&APP\_PERMIT**

APPROOT ディレクトリ配下でのアプリケーションとファイルの作成および更新を可能にします。

**BASEURL**

WebFOCUS が作成する HTML ドキュメントで参照される相対 URL のデフォルトパスを指定します。ブラウザは、このパスに基づいてファイルを検索します。

**FOCEXURL**

ドリルダウンをリモートで起動および実行します。

**JSURLS**

HTML レポートに JavaScript ファイルまたは VBScript ファイルを追加します。

**DRILLMETHOD**

ドリルダウンリクエストで使用するメソッドを制御します。DRILLMETHOD を POST に設定すると、パラメータと値は URL に含まれられず、ログにも格納されません。DRILLMETHOD を GET に設定すると、パラメータと値は URL に含まれられ、ログに格納されます。

**WEBVIEWALLPG**

WebFOCUS ビューアに [すべてのページ] ボタンを表示するかどうかを制御します。

**WEBVIEWCLOSE**

WebFOCUS ビューアに [閉じる] ボタンを表示するかどうかを制御します。

**WEBVIEWCLMSG**

WebFOCUS ビューアで [閉じる] オプションをクリックした際に、メッセージを表示するかどうかを制御します。

**WEBVIEWER**

WebFOCUS で、Web ビューアを有効にして起動します。

**WEBVIEWHELP**

WebFOCUS ビューアに [ヘルプ] ボタンを表示するかどうかを制御します。

**WEBVIEWHOME**

WebFOCUS ビューアを閉じる際に HTML ページが表示されるようにします。

### WEBVIEWTARG

ユーザがターゲットフレームで WebFOCUS ビューアを開けるようにします。

### WEBVIEWTITLE

WebFOCUS ビューアウィンドウのタイトルテキストを定義します。

## グラフタスク

次のパラメータは、グラフの処理および表示を制御します。

### 3D

グラフを 3 次元または 2 次元のいずれかに設定します。

### ARGRAPHENGINE

Active テクノロジーレポートでグラフの描画に使用するグラフエンジンを設定します。この設定は、FORMAT AHTML または他の Analytic Document フォーマット (APDF、FLEX) で使用した場合にのみ有効になります。

### AUTOFIT

HTML5 グラフがコンテナ内に収まるようサイズ変更するかどうかを制御します。

### AUTOTICK

グラフの目盛り間隔を設定します。

### BARNUMB

棒グラフの棒の先端または円グラフの円項目に合計値を配置します。

### BSTACK

棒グラフの棒を積み重ねるか、横に並べるかを指定します。

### EMBEDDABLE

HTML ページに追加する HTML5 グラフ出力を、ドキュメントレベルの HTML タグを使用して生成するか、使用せずに出力するかを制御します。

### EMBEDHEADING

見出しテキストまたは脚注テキストを、グラフィイメージの一部として表示するか、グラフィイメージとは別に表示するかを制御します。

### GRAPH-PPI

サーバーサイド HTML5 グラフを描画する際の解像度を制御します。

### GRAPHDEFAULT

WebFOCUS デフォルト設定を有効にするか、グラフ API デフォルト設定を有効にするかを制御します。

### GRAPHEDIT

WebFOCUS バージョン 8.0 以降では、このパラメータは廃止されました。

**GRAPHENGINE**

使用する WebFOCUS グラフエンジンのバージョンを指定します。

**GRAPHSERVURL**

HTTP リクエストを Servlet に送信し、グラフの GIF ファイルを作成します。

**GRMERGE**

リクエストに複数のソートフィールドが含まれている場合に、複数のグラフを作成するか、単一のグラフを作成するかを制御します。

**GRMULTIGRAPH**

GRMERGE を ADVANCED に設定した場合に、複数グラフの生成に使用するソートフィールド数を指定します。

**GRID**

グラフ上の縦の目盛りの位置で、平行に伸びる横の罫線を描きます。

**GRWIDTH**

セル単位で指定した幅の HTML テーブルに出力したグラフを配置します。

**GTREND**

基本線形回帰を指定し、SCATTER グラフの X 軸および Y 軸の値を変更します。

**HAUTO**

表示する値に合わせて横軸のスケールを自動的に変更します。

**HAXIS**

横軸の幅をバイト数で指定します。

**HCLASS**

AUTOTICK が OFF の場合に、横の目盛り間隔を指定します。

**HISTOGRAM**

横軸の値が数値以外の場合は、曲線ではなくヒストグラムを描きます。

**HMAX**

自動スケール変更機能を使用しない場合に (HAUTO=OFF)、横軸に表示する最大値を設定します。

**HMIN**

自動スケール変更機能を使用しない場合に (HAUTO=OFF)、横軸に表示する最小値を設定します。

**HSTACK**

ヒストグラムで棒グラフを横に並べる代わりに、棒グラフを積み重ねます。

**HTICK**

AUTOTICK が OFF の場合に、横軸の目盛り間隔を設定します。

### LOOKGRAPH

グラフのスタイルを指定します。

### OLAPGRMERGE

グラフを OLAP として表示する場合に、BY および ACROSS の両方の句を含むリクエストから、すべての出力データを反映した 3 次元グラフを生成します。

### VAUTO

表示する値に合わせて縦軸のスケールを自動的に変更します。

### VAXIS

縦軸の長さを行数で指定します。

### VCLASS

AUTOTICK が OFF の場合に、縦軸の目盛り間隔を指定します。

### VGRID

グラフの横軸または縦軸の目盛りの位置で罫線を描きます。

### VMAX

自動スケール変更機能を使用しない場合に (VAUTO=OFF)、縦軸に表示する最大値を設定します。

### VMIN

自動スケール変更機能を使用しない場合に (VAUTO=OFF)、縦軸に表示する最小値を設定します。

### VTICK

AUTOTICK が OFF の場合に、縦軸の目盛り間隔を設定します。

### VZERO

縦軸のミッシング値を 0 (ゼロ) として取り扱います。

## メモリ設定および最適化タスク

次のパラメータは、アプリケーションのメモリおよび最適化を制御します。

### AUTOINDEX

FOCUS データソースのインデックスフィールドを自動的に利用することにより、データ検索を高速化します。

### AUTOPATH

最適な検索パスを動的に選択します。

### AUTOSTRATEGY

WHERE または IF テストで指定されたキーフィールドの検索を、WebFOCUS がどの時点で停止するかを指定します。

**BINS**

データソースのバッファに使用するメモリのページ数を指定します。

**CACHE**

FOCUS データソースのページをメモリに格納し、データソースと BINS 間でバッファします。

**COMPUTE**

式のコンパイルを制御します。

**DEFINES**

パフォーマンスを向上させるために一時項目 (DEFINE) をマシンコードにコンパイルします。

**DMH\_LOOPLIM**

ダイアログマネージャで使用可能なループ反復数を制御します。

**DMH\_STACKLIM**

FOCSTACK に配置可能な行数を制御します。

**ESTRECORDS**

リクエストに格納する推定レコード数を渡します。

**FIXRETRIEVE**

HOLD ファイルなどの固定フォーマットのシーケンシャルファイルからのキー検索を可能にします。

**FOCSTACK**

実行待ちの FOCUS コマンドが使用する空き容量を 1000 バイト単位で指定します。

**RECORDLIMIT**

取得するレコード数を制限します。

**SQLTOPTTF**

SQL トランスレータが、TABLE コマンドの代わりに TABLEF コマンドを生成できるようになります。

**TEMP**

WebFOCUS で、一時ファイルを特定のディレクトリに割り当てます。

## レポートコード、コンテンツ、処理のタスク

次のパラメータは、レポートのコンテンツまたは処理に影響します。

**ALL**

レポート内のミッシングセグメントを処理します。

### ALLOWCVTERR

無効な日付フォーマットを含むデータ行の表示を制御します。

### ARCFGU

Analytic Document (IDA) 構成ファイルの標準検索パスを上書きすることができます。

### AREXPIRE

Analytic Document の期日を指定します。

### ARPASSWORD

Analytic Document 出力の表示に必要なパスワードを指定します。

### ASNAMES

HOLD マスターファイルの FIELDNAME 属性を制御します。

### AUTOTABLEF

クエリ実行時に使用する機能により内部マトリックスが作成されることを回避します。

### BLANKEMPTY

数値以外の XLSX データセル内の NULL 値とブランク値を区別することができます。これは、XLSX 出力フォーマットにのみ適用されます。

### BUSDAYS

営業日を指定します。

### CARTESIAN

PRINT または LIST コマンドを使用したマルチパスリクエストで、非相関データインスタンスのすべての組み合わせを含むレポートを作成します。

### CDN

数字の表記に使用する区切り文字を指定します。

### CENT-ZERO

小数部のみの数字に先頭のゼロを表示します。

### COLLATION

文字値の照合順序を制御します。

### COMPMISS

レポートリクエストでフォーマットが再設定されたフィールドに MISSING 属性を継承するかどうかを制御します。

### COMPUTE

式のコンパイルを制御します。

### DATEDISPLAY

日付フォーマットのフィールドに 0 (ゼロ) の値が含まれる場合の表示を制御します。

**DATEFNS**

2000 年対応バージョンの日付サブルーチンを有効にします。

**DATETIME**

レポート内の日付を設定します。また、CREATE SYNONYM がマスターファイルに作成する日付フィールドのフォーマットを制御します。

**DBAJJOIN**

DBA 制限をレポートのフィルタとして扱うか、JOIN 条件に追加するかを制御します。

**DB\_INFILE**

DB\_INFILE 関数で生成された式を、リレーショナルデータソースに対して使用できるように最適化するかどうかを制御します。

**DEFCENT**

アプリケーションで使用するデフォルトの西暦を定義します。

**DEFECHO**

アプリケーションに使用する &ECHO 変数のデフォルト値を定義します。

**DRILLFOCMISSING**

MISSING 値のドリルダウン値として \_FOC\_MISSING 文字列を渡すか、ピリオド (.) を渡すかを制御することができます。

**EMPTYCELLS**

数値フィールドで、XLSX 出力フォーマットのフィールドの MISSING オプションを処理して、数式バーに表示されるローデータとして、値が存在しない (空の) セルの代わりに、MISSING の値 0 (ゼロ) を表示できます。

**EMPTYREPORT**

レポートリクエストが 0 (ゼロ) のレコードを取得した際の出力を制御します。

**ERROROUT**

エラーが発生した場合、リクエストを終了し、メッセージを返します。

**ESTRECORDS**

リクエストに格納する推定レコード数を渡します。

**EXCELSERVURL**

EXCEL 2007 ファイル (.xlsx) を構成するファイルコンポーネントの圧縮に使用する Application Server を指定します。

**EXL2KLANG**

Microsoft Excel リクエストに使用する言語を指定します。この言語は、ブラウザがインストールされたマシンの Excel 言語と一致させる必要があります。

### EXTAGGR

外部ソートでの集計を有効にします。

### EXTHOLD

外部ソートを使用した HOLD ファイルの作成を可能にします。

### EXTRACT

リクエストの構造化 HOLD ファイルを有効にします。

### EXTSORT

外部ソート機能を有効にします。

### FIELDNAME

長い修飾フィールド名の使用を制御します。

### FILECOMPRESS

SET FILECOMPRESS=ON コマンドは、PDF 出力ファイルを圧縮します。圧縮された PDF ファイルは、元の PDF ファイルの 25 パーセントのサイズになります。

### FILENAME

コマンド内でデフォルトとして使用するファイルを指定します。

### FILTER

宣言したフィルタを有効にします。

### FOC144

警告メッセージ (FOC144) を非表示にします。

### FORMULTIPLE

FML マトリックスの複数の行に、FOR フィールドの値と同一の値を使用できるようにします。

### HNODATA

HOLD ファイルに継承するミッシング値を制御します。

### HOLDATTR

元のマスターファイルの TITLE 属性および ACCEPT 属性を HOLD マスターファイルに追加します。

### JOINLM

レコード指向型のアダプタに対して厳密な等価性が必要か、部分キーの JOIN がサポートされるかを制御します。JOINLM は、JOIN\_LENGTH\_MODE と同義です。

### JOINOPT

ミッシング値を修正し、レポート出力を正しい位置に揃えます。また、種類の異なる数値データタイプのフィールド間の JOIN を有効にします。

**JSURLS**

HTML レポートに JavaScript ファイルまたは VBScript ファイルを追加します。

**KEEPDEFINES**

JOIN コマンドを実行後、ホストまたは JOIN 構造用に作成した一時項目 (DEFINE) を保持するかどうかを制御します。

**LANG[UAGE]**

LANG[UAGE] パラメータは、国際言語サポート (NLS) の環境を指定します。このパラメータは、サーバのエラーメッセージに使用する言語を設定します。また、マスターファイルの説明に別の言語の TITLE 属性が含まれている場合に、レポートタイトルに使用する言語をこのパラメータで設定することもできます。

**LEADZERO**

先頭ゼロの切り捨てを回避します。

**NODATA**

レポート内でミッシングデータであることを表す文字列を指定します。

**ONFIELD**

リクエストで参照されていないフィールドの ON 句を無視するかどうかを制御します。

**PARTITION\_ON**

統計関数のパーティションサイズを制御します。

**PDFLINETERM**

Windows と UNIX の間で正常にファイル転送を行えるように、PDF 出力ファイルの各レコードに余分なスペースを追加するかどうかを指定します。

**PHONETIC\_ALGORITHM**

PHONETIC 関数で使用する音声アルゴリズムを指定します。

**PPTXGRAPHTYPE**

PowerPoint (PPTX) のスライドに埋め込まれたグラフの画質を向上させます。

**PRINTDST**

複数の DST 演算子とともに PRINT コマンドを使用するレポートの処理を制御します。

**QUALCHAR**

修飾フィールド名に使用する修飾文字を指定します。

**SAVEMATRIX**

ダイアログマネージャコマンドを使用する場合に、リクエストからマトリックスを保存し、上書きされないように保護します。

**SHORTPATH**

LEFT OUTER JOIN でのクロスリファレンスセグメントのミッシングインスタンスに対するテストの処理方法を制御します。

### **SORTMATRIX**

SORTMATRIX パラメータは、外部メモリの使用を抑えてインメモリのソートを使用するかどうかを制御します。

### **SORTMEMORY**

ソートに使用可能な内部メモリ量を制御します。

### **SUMMARYLINES**

1 つのリクエストの集計行で演算接頭語を持つフィールドと持たないフィールドとの結合を可能にします。

### **SUMPREFIX**

外部ソートを使用して文字フォーマットまたは SmartDate フォーマットで集計を実行する際に、ユーザがアンサーセットの表示順序を選択できるようにします。

### **TITLES**

レポート出力のフィールドタイトルに、マスターファイルで定義したフィールドタイトルを使用します。

### **WARNING**

警告メッセージを OFF にします。

### **XLSXLOCALZIP**

EXCEL07 (XLSX) ファイルの圧縮と解凍時に、ローカル C ベースの ZIP モードを有効にします。

## レポートレイアウトと表示タスク

次のパラメータは、レポートのレイアウトと表示方法に影響します。

### **ACCESSHTML**

セクション 508 に準拠する HTML レポート出力を生成します。

### **ACCESSIBLE**

セクション 508 に準拠する HTML コードでレポート出力を生成します。

### **ACCESSPDF**

セクション 508 に準拠する PDF レポート出力を生成します。

### **ACROSSLINE**

レポート出力の列タイトルの下線を制御します。TITLELINE は同義語です。

### **ACROSSPRT**

リクエストで PRINT コマンドと ACROSS 句が使用されている場合に各ソートグループ内のレポート行数を圧縮します。

### **ACROSSTITLE**

ACROSS タイトルを ACROSS 値の上側または左側のどちらに表示するかを制御します。

**ACRSVRBTITL**

ACROSS グループに 1 つのフィールドのみが表示される場合の ACROSS フィールドタイトルの表示を制御します。

**ALTBACKPERLINE**

ALTBACKPERLINE 属性は、位置決めドライバを使用するレポート (例、PDF、DHTML、PPT、PPTX) で、1 行おきに異なる背景色を表示します。

**AUTODRILL**

レポートの本文内で、1 つのディメンションから次のレベルへの自動ドリルダウンを可能にします。

**AUTOFIT**

HTML レポート出力がウィンドウ内に収まるようサイズ変更するかどうかを制御します。

**BASEURL**

WebFOCUS が作成する HTML ドキュメントで参照される相対 URL のデフォルトパスを指定します。ブラウザは、このパスに基づいてファイルを検索します。

**BLANKINDENT**

FML 階層の関係を明確にするため、階層レベル別にインデントを設定して値のタイトルを表示します。

**BOTTOMMARGIN**

スタイルが設定されたレポートで、各ページのレポートコンテンツに下部境界を設定します。

**BYDISPLAY**

レポートの各行、列、または両方にソートフィールドを表示します。

**BYPANEL**

BY フィールドの表示を制御します。

**BYSCROLL**

レポートのコンテンツの移動に合わせて、レポートの見出しと脚注をスクロールします。

**CENT-ZERO**

小数部のみの数字に先頭のゼロを表示します。

**COLUMNSCROLL**

レポートが画面の幅より広い場合に、レポートのウィンドウで列単位のスクロールを可能にします。

**COMPOUND**

複数のレポートを単一の PDF ファイルに統合して複合レポートを作成します。

### CSSURL

HTML レポートを外部カスケードスタイルシート (CSS) ファイルにリンクして、レポートのスタイルを変更します。

### CURRENCY\_DISPLAY

通貨値の位置を基準に通貨記号の相対位置を定義します。

### CURRENCY\_ISO\_CODE

使用する通貨記号の ISO コードを定義します。

### CURRENCY\_PRINT\_ISO

現在有効なコードページに従って通貨記号を表示できない場合の処理方法を定義します。

### CURRSYMB

数値フォーマット指定に表示オプションの M または N が使用されている場合に、レポート出力に表示する通貨記号を設定します。

### CURSYM\_D

数値フォーマット指定に表示オプションの :D または :d が使用されている場合に、レポート出力に表示する文字を設定します。

### CURSYM\_E

数値フォーマット指定に表示オプションの :E または :e が使用されている場合に、レポート出力に表示する文字を設定します。

### CURSYM\_F

数値フォーマット指定に表示オプションの :F が使用されている場合に、レポート出力に表示する文字を設定します。

### CURSYM\_G

数値フォーマット指定に表示オプションの :G が使用されている場合に、レポート出力に表示する文字を設定します。

### CURSYM\_L

数値フォーマット指定に表示オプションの :L または :l が使用されている場合に、レポート出力に表示する文字を設定します。

### CURSYM\_Y

数値フォーマット指定に表示オプションの :Y または :y が使用されている場合に、レポート出力に表示する文字を設定します。

### CUSTOM-PAGE-LENGTH

PAGESIZE=CUSTOM のページの長さを設定します。

### CUSTOM-PAGE-WIDTH

PAGESIZE=CUSTOM のページの幅を設定します。

**DISPLAYROUND**

バイナリから 10 進数への変換で発生する端数処理エラーを修正するために、浮動小数点数および倍精度浮動小数点数に若干数値を追加して表示します。

**DROPBLNKLINE**

レポート出力からブランク行を除外します。

**DUPLICATECOL**

複数の表示コマンドの各フィールドを、それぞれ別の列に表示するか、同一列内に積み重ねて表示するかを制御します。

**EXPANDABLE**

列単位で展開されるアコーディオンレポートを作成します。

**EXPANDBYROW**

同一列内で展開されるアコーディオンレポートを作成します。

**EXPANDBYROWTREE**

強化されたインターフェースを使用して、アコーディオンレポートを作成します。

**EXTENDNUM**

レポートの表示値のオーバフローを回避します。

**FOCFIRSTPAGE**

出力の先頭ページにページ番号を割り当てます。

**HIDENULLACRS**

Null 値のみを含む ACROSS 列を非表示にします。

**HTMLCSS**

レポート出力を表示する HTML ページで、インラインカスケードスタイルシートのコマンドを作成します。

**HTMLEMBEDIMG**

イメージおよびグラフを HTML または DHTML .html ファイルに直接埋め込むかどうかを指定します。

**LAN[GUAGE]**

国際言語サポート (NLS) 環境を指定します。サーバエラーメッセージの言語を設定します。[マスターファイルの説明] に別の言語の TITLE 属性が指定されている場合は、レポートタイトルの言語設定に使用することもできます。

**LAYOUTGRID**

レポートを設計する際に、レポート出力にグリッドを表示し、データやオブジェクトを正確に配置することができます。このオプションは、PDF、DHTML レポート出力を使用する場合にのみ適用されます。

### LAYOUTRTL

レポート出力を右から左へ表示します (この表示方法がサポートされているフォーマットの場合)。

### LEFTMARGIN

スタイルが設定されたレポートで、各ページのレポートコンテンツに左側境界を設定します。

### LINES

上部の見出しから下部の脚注までを含めて、ページに表示する行数の最大値を設定します。OFFLINE-FMT パラメータは、リクエストから生成されたレポート出力のフォーマットを指定します。

### OFFLINE-FMT

OFFLINE-FMT パラメータは、リクエストから生成されたレポート出力のフォーマットを指定します。

### ONLINE-FMT

レポート出力のフォーマットを指定します。このパラメータは、WebFOCUS にのみ適用されます。

### ORIENTATION

スタイルが設定されたレポートのページ方向を指定します。

### OVERFLOWCHAR

数値レポート列に値を表示するための十分な領域がない場合に列に表示する文字を変更します。

### PAGE[ -NUM]

出力ページのページ番号を制御します。

### PAGE-SCALE

ページ幅に合わせて PDF レポート出力の幅を拡大縮小します。

### PANEL

レポートパネルの最大幅を指定します。

### PCTFORMAT

PCT.、RPCT.、PCT.CNT. の演算接頭語を含むフィールドで、元のフィールドフォーマットを使用して表示するか、パーセント記号を使用して表示するかを制御します。

### PRFTITLE

レポートの演算接頭語を含むフィールドに、読みやすい翻訳可能なフィールドタイトルを生成します。

### PRINTPLUS

印刷の詳細設定を指定します。

**POPUPDESC**

POPUPDESC パラメータを使用すると、HTML レポートのフィールドタイトルにマウスポインタを置いたときに、フィールド説明のポップアップを表示させることができます。

**QUALTITLES**

マスターファイルに同一のフィールド名が存在する場合は、修飾フィールドタイトルをレポート出力に使用します。

**RECAP - COUNT**

出力ページあたりの行数を集計する際に、RECAP が生成した値の行を行数に含めます。

**RIGHTMARGIN**

各ページのレポートコンテンツに右側境界を設定します。

**SHOWBLANKS**

HTML および EXL2K レポート出力で、文字列の先頭および中間に存在するブランクを保持します。

**SPACES**

レポートの列間のブランク数を設定します。

**SQUEEZE**

レポート出力の列幅を指定します。

**STYLEMODE**

大規模なレポート出力の場合、レポート出力を複数の HTML テーブルに表示します。この場合、各テーブルにはそれぞれ別のレポートページが表示されます。

**STYLE[SHEET]**

スタイルシートパラメータを使用または未使用にすることにより、レポート出力のフォーマットを制御します。

**SUBTOTALS**

集計行をデータの上側または下側のどちらに表示するかを制御します。

**TARGETFRAME**

ブラウザに表示する HTML ファイルの見出しに、HTML コードの BASE TARGET="frameName" を追加します。

**TITLELINE**

列タイトルの下線を制御します。ACROSSLINE は同義語です。

**TOPMARGIN**

レポート出力の各ページに上部境界を設定します。

**UNITS**

ページマージン、フィールドの位置、列幅に使用する長さの単位を指定します。

### VISBARORIENT

ACROSS フィールドのピアグラフの方向を指定します。

### XLSXLOCALZIP

EXCEL07 (XLSX) ファイルの圧縮と解凍時に、ローカル C ベースの ZIP モードを有効にします。

### XLSXPAGEBRKIGNORE

WebFOCUS の改ページを XLSX フォーマットの Excel の改ページと同期します。

## セキュリティタスク

次のパラメータは、データソースおよびプロシジャへのユーザアクセスを指定します。

### &&APP\_PERMIT

APPROOT ディレクトリ配下でのアプリケーションとファイルの作成および更新を可能にします。

### DBACSENSITIV

パスワードの検証で大文字と小文字を区別するかどうかを制御します。

### DBASOURCE

マルチファイル構造でアクセスを制限するソースを制御します。

### PASS

DBA セキュリティにより保護されたデータソースまたはプロシジャへのユーザアクセスを可能にします。

### PERMPASS

PERMPASS パラメータは、セッション中または接続中は継続して有効になるパスワードを作成します。

### USER

WebFOCUS で、DBA セキュリティで保護されたデータソースまたはプロシジャへのユーザアクセスを有効にします。

## SET パラメータ構文

ここでは、環境を制御する SET パラメータの説明および構文をパラメータのアルファベット順に記載しています。

A	B	C	D
3D	BARNUMB	CACHE	DATE_ORDER
ACCBLN	BASEURL	CARTESIAN	DATE_SEPARATOR
ACCESSHTML	BINS	CDN	DATEDISPLAY
ACCESSIBLE	BLANKEMPTY	CENT-ZERO	DATEFORMAT
ACCESSPDF	BLANKINDENT	CNOTATION	DATETIME
ACROSSLINE	BOTTOMMARGIN	CNVERR	DB_INFILE
ACROSSPRT	BUSDAYS	CNVERRLIMIT	DBACSENSITIV
ACROSSTITLE	BYDISPLAY	COLLATION	DBAJJOIN
ACRSVRBTITL	BYPANEL	COMPMISS	DBASOURCE
ALL		COMPOUND	DEFCENT
ALLOWCVTERR		COMPUTE	DEFECHO
ALTBACKPERLINE		COUNTWIDTH	DEFINES
ARCFGU		CSSURL	DIRECTHOLD
AREXPIRE		CURRSYMB	DISPLAYROUND
ARGGRAPHENGIN		CURSYM_D	DMH_LOOP LIM
ARPASSWORD		CURSYM_E	DMH_STACK LIM
ASNAMES		CURSYM_F	DMPRECISION
AUTODRILL		CURSYM_G	DRILLFOCMISSING
AUTOFIT		CURSYM_L	DRILLMETHOD
AUTOINDEX		CURSYM_Y	DROPBLNKLINE
AUTOPATH		CUSTOM_PAGE	DTSTRICT
AUTOSTRATEGY		_LENGTH	DUPLICATECOL
AUTOTABLEF		CUSTOM-PAGE-	
AUTOTICK		WIDTH	

E	F	G	H
EMBEDDABLE	FIELDNAME	GRAPH-PPI	HAUTO
EMBEDHEADING	FILECASE	GRAPHDEFAULT	HAXIS
EMPTYCELLS	FILECOMPRESS	GRAPHEDIT	HCLASS
EMPTYREPORT	FILENAME	GRAPHENGINE	HDAY
EQTEST	FILTER	GRAPHSERVURL	HIDENULLACRS
ERROROUT	FIXRETRIEVE	GRID	HISTOGRAM
ESTRECORDS	FOC144	GRMERGE	HLDCOM_TRIMANV
EUROFILE	FOCEXURL	GRMULTIGRAPH	HMAX
EXCELSERVURL	FOCFIRSTPAGE	GRWIDTH	HMIN
EXL2KLANG	FOCHTMLURL	GTREND	HNODATA
EXL2KTXDATE	FOCSTACK		HOLDATTRS
EXPANDABLE	FORMULTIPLE		HOLDFORMAT
EXPANDBYROW			HOLDLIST
EXPANDBYROWTREE			HOLDMISS
EXTAGGR			HOLDSTAT
EXTENDNUM			HSTACK
EXTHOLD			HTMLARCHIVE
EXTRACT			HTMLCSS
EXTSORT			HTMLEMBEDIMG
			HTMLENCODE

I	J	K	L
INDEX	JOIN_LENGTH_MODE JOINOPT JPEGENCODE JPEGQUALITY JSURLS	KEEPDEFINES KEEPFILTERS	LANGUAGE LAYOUTGRID LAYOUTURL LEADZERO LEFTMARGIN LINES LOOKGRAPH

M	N	O	P
MATCHCOLUMNORDER	NEG-ZERO	OFFLINE-FMT	PAGE-NUM
MAXDATAEXCPT	NODATA	OLAPGRMERGE	PAGE-SCALE
MAXLRECL	NULL	OLDSTYRECLN	PAGESIZE
MDICARDWARN		ONFIELD	PANEL
MDIENCODING		ONLINE-FMT	PARTITION_ON
MDIPROGRESS		ORIENTATION	PASS
MESSAGE		OVERFLOWCHAR	PCOMMA
MISS_ON			PCTFORMAT
MISSINGTEST			PDFLINETERM
MULTIPATH			PERMPASS
			PHONETIC_ALGORIT HM
			POPUPDESC
			PPTXGRAPHTYPE
			PRFTITLE
			PRINT
			PRINTDST
			PRINTPLUS

Q	R	S	T
QUALCHAR	RANK	SAVEDMASTERS	TARGETFRAME
QUALTITLES	RECAP-COUNT	SAVEMATRIX	TEMP
	RECORDLIMIT	SHADOW	TEMPERASE
	RIGHTMARGIN	SHIFT	TESTDATE
	RPAGESET	SHORTPATH	TITLELINE
		SHOWBLANKS	TITLES
		SORTMATRIX	TOPMARGIN
		SORTMEMORY	
		SPACES	
		SQLTOPTTF	
		SQUEEZE	
		STRICTMATH	
		STYLEMODE	
		STYLESHEET	
		SUBTOTALS	
		SUMMARYLINES	
		SUMPREFIX	
		SUPPRESSDRILLDT	

U	V	W	X-Y-Z
UNITS	VAUTO	WARNING	XLSXLOCALZIP
USER	VAXIS	WEBARCHIVE	XLSXPAGEBRK
USERCHK	VCLASS	WEBVIEWALLPG	IGNORE
USERFNS	VGRID	WEBVIEWCLMSG	XRETRIEVAL
	VMAX	WEBVIEWCLOSE	YRTHRESH
	VMIN	WEBVIEWER	
	VTICK	WEBVIEWHELP	
	VZERO	WEBVIEWHOME	
		WEBVIEWTARG	
		WEBVIEWTITLE	
		WEEKFIRST	
		WPMINWIDTH	

3D

3D パラメータは、グラフを 3 次元または 2 次元のいずれかに設定します。

構文は次のとおりです。

SET 3D = {ON|OFF}

説明

ON

3 次元グラフを生成します。デフォルト値は ON です。

OFF

2 次元グラフを生成します。

ACCBLN

ACCBLN パラメータは、マスターファイルで ACCEPT コマンドを指定したフィールドに対して  
ブランクまたは 0 (ゼロ) を許容します。

構文は次のとおりです。

**SET ACCBLN = {ON|OFF}**

説明

**ON**

許容値リストにブランクまたは 0 (ゼロ) の値が明示的に記述されていない場合は、ACCEPT コマンドを指定したフィールドにブランクまたはゼロを許容します。デフォルト値は ON です。

**OFF**

許容値リストにブランクまたは 0 (ゼロ) の値が明示的に記述されていない場合は、ACCEPT コマンドを指定したフィールドにブランクまたはゼロを許容しません。

## ACCESSHTML

ACCESSHTML パラメータは、セクション 508 アクセシビリティ要件に準拠する HTML レポートを出力します。ACCESSHTML パラメータは、ACCESSIBLE パラメータ (現在でもサポート対象) より優先されます。これは、ACCESSHTML パラメータが、今後追加されるアクセシビリティ標準をサポートするための拡張性を備えているためです。

セクション 508 準拠の HTML コードで、ページ番号を自動的にオフにします。列タイトルおよびページ見出しはレポートの先頭にのみ表示され、ページ脚注はレポートの末尾にのみ表示されます。

構文は次のとおりです。

**SET ACCESSHTML = {508|OFF}**

説明

**508**

セクション 508 アクセシビリティ要件に準拠する HTML レポートを出力します。

**OFF**

セクション 508 アクセシビリティ要件に準拠するレポートを出力しません。デフォルト値は OFF です。

## ACCESSIBLE

ACCESSIBLE パラメータは、セクション 508 アクセシビリティ要件に準拠する HTML コードでレポートを出力します。

セクション 508 準拠の HTML コードで、ページ番号を自動的にオフにします。列タイトルおよびページ見出しはレポートの先頭にのみ表示され、ページ脚注はレポートの末尾にのみ表示されます。

構文は次のとおりです。

```
SET ACCESSIBLE = {508|OFF}
```

説明

**508**

セクション 508 アクセシビリティ要件に準拠する HTML コードでレポートを出力します。

OFF

セクション 508 アクセシビリティ要件に準拠するレポートを出力しません。デフォルト値は OFF です。

### ACCESSPDF

ACCESSPDF パラメータは、セクション 508 アクセシビリティ要件に準拠する PDF レポートを出力します。

構文は次のとおりです。

```
SET ACCESSPDF = {508|OFF}
```

説明

**508**

セクション 508 アクセシビリティ要件に準拠する PDF レポートを出力します。

OFF

セクション 508 アクセシビリティ要件に準拠するレポートを出力しません。デフォルト値は OFF です。

### ACROSSLINE

ACROSSLINE パラメータは、レポート出力の列タイトルの下線を制御します。TITLELINE は ACROSSLINE と同義です。

構文は次のとおりです。

```
SET {ACROSSLINE|TITLELINE} = {ON|OFF|SKIP}
```

説明

ON

レポート出力の列タイトルに下線を配置します。デフォルト値は ON です。

**OFF**

下線の代わりにブランク行を挿入します。

**SKIP**

下線およびブランク行を挿入しません。

**ACROSSPRT**

ACROSSPRT パラメータは、リクエストで PRINT コマンドと ACROSS 句が使用されている場合に各ソートグループ内のレポート行数を圧縮します。

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

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

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

構文は次のとおりです。

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

**説明****NORMAL**

レポート行を圧縮しません。

**COMPRESS**

ソートグループ内でデータ値を上に移動することでレポート行を圧縮します。

**ACROSSTITLE**

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

PDF、HTML、EXL2K、EXL07 レポート出力で ACROSSTITLE パラメータを使用すると、ACROSS タイトルと ACROSS 値の両方を 1 行に表示することができます。その場合は、SET ACROSSTITLE = SIDE コマンドを発行します。このコマンドは、ACROSS 値の左側に ACROSS タイトルを配置します。タイトルは、BY フィールドタイトルの上部に右揃えで表示されます。デフォルト設定で作成される、ACROSS タイトルを表示するための見出し行は生成されません。

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

構文は次のとおりです。

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

説明

[ABOVE](#)

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

[SIDE](#)

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

## ACRSVRBTITL

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

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

構文は次のとおりです。

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

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

## 説明

HIDEONE

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

## ON

表示フィールドが 1 つの場合でもタイトルを常に表示します。

## OFF

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

## ALL

ALL パラメータは、レポート内のミッシングセグメントを処理します。

SET ALL = ON により、LEFT OUTER JOIN が指定されます。LEFT OUTER JOIN の場合、ホストファイルのレコードは、すべてレポート出力上に表示されます。クロスリファレンスセグメントインスタンスがホストセグメントインスタンスに存在しない場合、レポート出力にはクロスリファレンスセグメントのフィールドのミッシング値が表示されます。

従属セグメントに選別条件が設定されている場合、選別条件を満たさないセグメントインスタンスおよび対応するホストセグメントインスタンスはレポート出力から除外されます。

構文は次のとおりです。

**SET ALL = {ON|OFF|PASS}**

## 説明

## ON

リクエストに WHERE または IF 条件を使用してセグメント内のフィールドを選別していない場合は、ミッシングセグメントをレポートに表示します。フィールドのミッシング値は、NODATA パラメータで設定した NODATA 文字で表記されます。詳細は、615 ページの「**NODATA**」を参照してください。

OFF

ミッシングセグメントをレポートに表示しません。デフォルト値は OFF です。

PASS

リクエストでの WHERE または IF 条件の使用に関係なく、ミッシングセグメントをレポートに表示します。

ALLOWCVTERR

ALLOWCVTERR パラメータは、FOCUS 以外のデータソースを対象とし、格納された日付 (ACTUAL 属性) をフォーマットが設定された日付 (FORMAT または USAGE 属性) に変換する場合に適用されます。

このパラメータは、無効な日付フォーマットを含むデータ行の表示を制御します。以前は、「SmartDate」と呼ばれていました。このパラメータを ON に設定すると、MISSING および DATEDISPLAY パラメータの設定に基づいて、無効な日付フォーマットには基準日またはブランクが返されます。

**注意：**ALLOWCVTERR パラメータは、一時項目 (DEFINE) ではサポートされません。

構文は次のとおりです。

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

説明

ON

無効な日付フォーマットを含むデータ行を表示します。ALLOWCVTERR を ON に設定すると、無効な日付の表示が MISSING 属性および DATEDISPLAY コマンドの設定に基づいて指定されます。

下表は、それぞれの結果の説明です。

DATEDISPLAY	MISSING	結果
OFF	OFF	ブランクが返されます。
	ON	NODATA 文字の値が返されます (デフォルト設定ではピリオド)。詳細は、615 ページの「 <a href="#">NODATA</a> 」を参照してください。

DATEDISPLAY	MISSING	結果
ON	OFF	基準日が返されます (YMD または YYMD フォーマットの日付では 1900 年 12 月 31 日、YM、YYM、YQ、YYQ フォーマットの日付では 1901 年 1 月)。
	ON	NODATA 文字の値が返されます (デフォルト設定ではピリオド)。

**OFF**

無効な日付フォーマットを含むデータ行を表示せず、エラーメッセージを生成します。デフォルト値は OFF です。

**ALTBACKPERLINE**

ALTBACKPERLINE 属性は、位置決めドライバを使用するレポート (例、PDF、DHTML、PPT、PPTX) で、1 行おきに異なる背景色を表示します。これにより、長いフィールド値を折り返した場合に、境界とは関係なく、各値内の折り返し行ごとに交互に異なる背景色を表示することができます。1 行おきに異なる背景色を適用するには、WRAP を使用するプロシジャに SET ALTBACKPERLINE=ON コマンドを明示的に追加する必要があります。

構文は次のとおりです。

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

**説明****ON**

各値内の折り返し行ごとに交互に異なる背景色を表示します。

**OFF**

フィールド値ごとに交互に異なる背景色を表示します。これがデフォルト値です。

**ARCFGU**

SET ARCFGU コマンドを使用すると、ユーザが作成した irpcfgu.json ファイルを検索して使用することにより、Analytic Document (IDA) 構成ファイルの標準検索パスを上書きすることができます。SET コマンドを使用して、すべてのディレクトリ (DEFAULT) または単一の特定のディレクトリ (app\_name) に設定することも、この上書きの検索をしない (NONE) 設定にすることもできます。SET ARCFGU=NONE は、最も高速な検索を提供します。

構文は次のとおりです。

**SET ARCFGU={[DEFAULT](#) | [app\\_name](#) | [NONE](#)}**

説明

**[DEFAULT](#)**

アプリケーションパス内のすべてのディレクトリから `irpcfgu.json` ファイルを検索します。これはデフォルト値であり、実行速度が最も低下します。

**[app\\_name](#)**

アプリケーションの明示的な名前を指定します。この名前が APP PATH に存在する必要はありません。

**[NONE](#)**

ユーザがカスタマイズしたバージョンを使用しません。値が `NONE` の場合は、ユーザ構成ファイルは検索されません。この設定が、最も高速な実行オプションです。

## AREXPIRE

AREXPIRE パラメータを使用すると、Analytic Document を期限切れにし、レポート出力を不可にする期日を設定することができます。

AREXPIRE パラメータは必要に応じて使用します。デフォルト設定では、Analytic Document に期日は設定されません。期日をリセットするにはリクエストに戻り、期日のない新しい出力を作成します。

構文は次のとおりです。

**SET AREXPIRE = {[yymmdd](#) | [xxx](#)DAYS}**

説明

**[yymmdd](#)**

YYMMDD フォーマットの期日です。たとえば、レポートを 2009 年 1 月 1 日に期限切れにするには、090101 と表します。

**[xxx](#)**

現在の日付からレポートの期限切れまでの日数です。有効な値は 1 から 999 です。

リクエスト内からコマンドを発行することもできます。

**ON TABLE SET AREXPIRE {[yymmdd](#) | [xxx](#)DAYS}**

## ARGRAPHENGIN

Active テクノロジレポートでグラフの描画に使用するグラフエンジンを設定します。この設定は、FORMAT AHTML または他の Analytic Document フォーマット (APDF、FLEX) で使用した場合にのみ有効になります。

構文は次のとおりです。

```
SET ARGRAPHENGIN = JSCHART
```

## ARPASSWORD

ARPASSWORD パラメータを使用すると、Analytic Document 出力を表示するためのパスワードを設定することができます。レポート出力を開く前に、レポートのロックを解除するパスワードの入力が要求されます。

ARPASSWORD パラメータは必要に応じて使用します。デフォルト設定では、Analytic Document のアクセスにパスワードは必要ありません。

構文は次のとおりです。

```
SET ARPASSWORD = password
```

説明

`password`

長さ 32 バイト以内の任意の文字列です。

リクエスト内からコマンドを発行することもできます。

```
ON TABLE SET ARPASSWORD password
```

## ASNAMES

ASNAMES パラメータは、HOLD マスターファイルの FIELDNAME 属性を制御します。TABLE リクエストに AS 句を使用すると、指定したリテラルが HOLD ファイルのフィールド名に使用されます。また、作成された HOLD ファイルの ACROSS フィールド値に指定するフィールド名も制御します。

構文は次のとおりです。

```
SET ASNAMES = {ON|OFF|MIXED|FOCUS|FLIP}
```

### 説明

#### OFF

AS 句で指定したりテラルを HOLD ファイルのフィールド名として使用しないため、ACROSS フィールドの名前には影響しません。

#### ON

AS 句で指定したりテラルを大文字に変換し、それをフィールド名として HOLD マスターファイルに格納します。ACROSS フィールド値の先頭に AS 名を連結した ACROSS フィールド名を作成し、すべてのフォーマットの HOLD ファイルを対象として ACROSS フィールドの名前を制御します。

#### MIXED

AS 句で指定したりテラルをフィールド名として使用し、AS 名の大文字と小文字を区別したまま、ACROSS フィールド値の先頭に AS 名を連結した ACROSS フィールド名を作成します。

#### FOCUS

AS 句で指定したりテラルをフィールド名として使用し、FOCUS フォーマットの HOLD ファイルのみを対象として ACROSS フィールドの名前を制御します。デフォルト値は FOCUS です。

#### FLIP

元のマスターファイルのフィールド名を HOLD マスターファイルのエイリアス名に継承し、元のマスターファイルのエイリアス名を HOLD マスターファイルのフィールド名に継承します。

## AUTODRILL

AUTODRILL パラメータは、OLAP に適用されます。

このパラメータを使用すると、現在のレポートの本文内で、1つのディメンションから次の階層に自動的にドリルダウンすることができます。つまり、マスターファイルの WITHIN 属性で定義されたすべての出力データ項目の HTML コードに、サーバが自動的にハイパーリンク (<A HREF タグ) を生成します。このハイパーリンクの生成は、AUTODRILL が ON の場合に実行されます。

他に階層レベルがなく、選択したフィールドが BY フィールドの場合は、そのディメンションのフィールドを含めずにレポートが生成されます。他に階層レベルがなく、選択したフィールドが ACROSS フィールドの場合は、そのディメンションの行を含めずにレポートが生成されます。AUTODRILL は、マスターファイルでディメンションとして定義されたソートフィールドで動作します。

OLAP コントロールパネルが開いている場合は、画面上で選択した出力データ項目がパネルの [選択条件] リストに追加されるため、ドリルダウンレポートを実行する前にエンドユーザが複数の項目を選択することができます。

OLAP コントロールパネルが開いていない場合は、画面上で選択した出力データ項目に基づいて自動的にドリルダウンレポートが実行されます。

構文は次のとおりです。

```
SET AUTODRILL = {ON|OFF|ALL}
```

説明

**ON**

マスターファイルの WITHIN 属性で定義したすべての出力データ項目の HTML ソースコードにハイパーリンクを自動的に生成します。デフォルト値は ON です。

**OFF**

ハイパーリンクの自動生成を無効にします。

**ALL**

メジャーも含めてレポート内のすべての値をハイパーリンクとして作成します。これにより、レポート内のすべての値でドリルダウンが可能になります。メジャーでドリルダウンすると、最下位のドリルダウン (BY) フィールドを基準にさらに詳細なレポートが生成されます。

## AUTOFIT

AUTOFIT パラメータを使用すると、HTML レポートの場合は、レポート出力がウィンドウまたはフレーム内に収まるよう自動的にサイズ変更され、HTML5 グラフの場合は、グラフがコンテナ内に収まるよう自動的にサイズ変更されます。

構文は次のとおりです。

```
SET AUTOFIT = {OFF|ON|RESIZE}
```

```
ON {GRAPH|TABLE} SET AUTOFIT {OFF|ON|RESIZE}
```

説明

**OFF**

HTML レポートの場合、データおよびスタイルで指定されたサイズが使用されます。HTML5 グラフの場合、HAXIS および VAXIS パラメータで指定されたサイズが使用されます。

### ON

HTML レポートの場合、レポート出力がウィンドウまたはフレーム内に収まるよう常にサイズ変更されます。HTML5 グラフの場合、グラフがコンテナ内に収まるよう常にサイズ変更されます。

### RESIZE

HTML5 グラフにのみ適用されます。最初は HAXIS および VAXIS パラメータで指定されたサイズに固定されますが、コンテナのサイズが変更された場合はグラフ出力のサイズも変更されます。

## AUTOINDEX

AUTOINDEX パラメータは、インデックスフィールドを自動的に利用してデータ取得を高速化します。たいていの場合、TABLE リクエストにはこれらのフィールドに対する等価条件または範囲条件のテストが含まれています。FOCUS データソースにのみ適用されます。

TABLE リクエストに代替ファイルビューが含まれている場合 (例、TABLE FILE filename.fieldname)、AUTOINDEX は実行されません。TABLE リクエストに BY HIGHEST または BY LOWEST 句が含まれ、AUTOINDEX が ON に設定されている場合、インデックス検索は実行されません。

構文は次のとおりです。

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

### 説明

#### ON

インデックス検索の使用が可能な場合にそれを使用します。デフォルト値は ON です。

#### OFF

インデックスビューで明示的に指定している場合に限り、インデックス検索を使用します (例、TABLE filename.indexed-fieldname)。

## AUTOPATH

AUTOPATH パラメータは、データベース構造および参照フィールドを分析し、最下位に属するセグメントをエントリポイントとして選択することにより、FOCUS データソースにアクセスするための最適な検索パスを動的に選択します。AUTOPATH は、インデックスフィールド以外のフィールドでのみ使用します。

構文は次のとおりです。

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

説明

ON

最適な検索パスを動的に選択します。デフォルト値は ON です。

OFF

連続データ検索を使用します。エンドユーザは、filename.segname を操作して検索パスを制御します。

## AUTOSTRATEGY

AUTOSTRATEGY パラメータは、WHERE または IF テストで指定したキーフィールドを検索する際に、FOCUS がその検索を終了する時期を指定します。ON に設定した場合、キーフィールドが見つかった時点で検索が終了します。これにより、検索速度が最適化されます。OFF に設定した場合、データソースの最後まで検索が実行されます。

構文は次のとおりです。

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

説明

ON

一致する値が見つかった時点で検索を終了します。デフォルト値は ON です。

OFF

データソース全体を検索します。

## AUTOTABLEF

AUTOTABLEF パラメータは、クエリで使用する機能が内部マトリックスを作成されることを回避します。内部マトリックスの作成を回避すると、内部オーバーヘッドが減少し、パフォーマンスが向上します。

構文は次のとおりです。

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

説明

ON

内部マトリックスを作成しません。デフォルト値は ON です。

### OFF

内部マトリックスを作成します。

## AUTOTICK

AUTOTICK パラメータは、グラフの目盛り間隔を自動的に設定します (596 ページの「[HTICK](#)」 および 653 ページの「[VTICK](#)」 も参照してください)。

構文は次のとおりです。

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

説明

### [ON](#)

目盛り間隔を自動的に設定します。デフォルト値は ON です。

### OFF

目盛り間隔を設定しません。目盛り間隔は、HTICK および VTICK などのパラメータで設定する必要があります。

## BARNUMB

BARNUMB パラメータは、棒グラフの棒の先端または円グラフの円項目に合計値を配置します。

構文は次のとおりです。

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

説明

### ON

棒グラフの棒の先端または円グラフの円項目に合計値を配置します。

### [OFF](#)

合計値を配置しません。デフォルト値は OFF です。

## BASEURL

BASEURL パラメータは、WebFOCUS が作成する HTML ドキュメントで参照される相対 URL のデフォルトパスを指定します。ブラウザは、このパスに基づいてファイルを検索します。このパラメータを使用すると、完全な URL を指定しなくても、ファイル名のみを使用してファイル、イメージ、Java ファイルへのハイパーリンクを設定することができます。

構文は次のとおりです。

```
SET BASEURL = url
```

説明

[url](#)

ドキュメントに使用する HTML ファイル、イメージファイル、Java アプレット CLASS ファイルが格納された完全修飾ディレクトリです。この URL が Web サーバのアドレスを表す場合は、先頭に「http://」を、末尾にスラッシュ (/) を追加する必要があります。

## BINS

BINS パラメータは、データソースのバッファに使用するメモリのページ数 (4,096 バイト単位のプロック) を指定します。

構文は次のとおりです。

```
SET BINS = n
```

説明

[n](#)

データソースのバッファに使用するコアページ数です。有効値は 13 から 64 です。デフォルト値は、開始後の残存するコアの 3 分の 2 で、およそ 64 ページになります。

## BLANKEMPTY

BLANKEMPTY パラメータを使用すると、数値以外の XLSX データセル内の NULL 値とブランク値を区別することができます。これは、XLSX 出力フォーマットにのみ適用されます。

構文は次のとおりです。

```
SET BLANKEMPTY = {ON|OFF}
ON TABLE SET BLANKEMPTY {ON|OFF}
```

説明

[ON](#)

単一のブランク値を空のセルと見なします。これは従来の動作です。

[OFF](#)

単一のブランク値を含むセルを単一のブランクとして処理します。デフォルト値は OFF です。

### 注意

- ❑ 数値以外のフォーマットに設定されたフィールドの場合、ブランクを含む通常のデータフィールドは、セル内のブランクを保持します。MISSING としてフラグされたセルは、XLSX 出力でミッシングセルの生成が継続されます。
- ❑ EMPTYCELLS パラメータは、XLSX 出力フォーマットのフィールドの MISSING オプションを処理して、数式バーに表示されるローデータとして、値が存在しない (空の) セルの代わりに、MISSING の値 0 (ゼロ) を表示できるようにします。

## BLANKINDENT

FML 階層では、階層間の関係を明確にするため、各レベルで値のキャプション (タイトル) がインデントされます。HTML と PDF のいずれかのレポートに BLANKINDENT パラメータを使用し、階層の各レベルにインデントを設定することができます。インデントを設定する場合、各階層レベルにデフォルトのインデントを使用することも、任意のインデント値を指定することも可能です。キャプションにインデントを設定した HTML レポートを出力するには、BLANKINDENT パラメータを ON に設定するか、特定の値を指定する必要があります。

PDF レポートでは、インデントを正しく表示するために桁数の調整が必要となる場合があります。

構文は次のとおりです。

```
SET BLANKINDENT = {ON|OFF|n}
```

### 説明

#### ON

FML 階層でキャプションの先頭に標準的に表示されるブランク 1 つに対して、0.125 単位分のインデントを設定します。FML 階層の子レベルでは、対象の行とその 1 つ前の行との間に標準的に表示されるブランク 1 つに対して、0.125 単位分のインデントを設定します。

#### OFF

HTML レポートでは、FML 階層のキャプションに適用するインデントをオフにします。他のフォーマットでは、デフォルトのインデントであるブランク 2 つ分を使用します。デフォルト値は OFF です。

n

UNITS パラメータで定義した単位に基づいて明示的に指定する値です。キャプションの先頭に標準的に表示されるブランク数にこの値を乗算します。FML 階層内の子レベルでは、対象の行とその 1 つ前の行と間に標準的に表示されるブランク 1 つに対して、n 単位分のインデントを設定します。デフォルトのブランク数は 2 です。0 (ゼロ) を指定すると、OFF を指定した場合と同一のレポート出力が生成されます。n には、負の数は使用できません。負の値を指定した場合は、次のメッセージが生成され、リクエストは BLANKINDENT=OFF の設定として処理されます。

VALID VALUES ARE OFF, ON OR A POSITIVE NUMBER (IN CURRENT UNITS)

## BOTTOMMARGIN

BOTTOMMARGIN パラメータは、ページに表示するレポートコンテンツにスタイルシートの下部境界を設定します。

このパラメータは、PDF レポートのフォーマットにのみ適用されます。

構文は次のとおりです。

SET BOTTOMMARGIN = {n|\_250}

説明

n

ページ上のレポートコンテンツに適用する下マージンをインチ単位で指定します。デフォルト値は 0.25 インチです。

## BUSDAYS

BUSDAYS パラメータは、月曜から金曜までを営業日とする通常の曜日指定を採用しない場合に、営業日および営業日以外の曜日をそれぞれ指定します。

構文は次のとおりです。

SET BUSDAYS = {week|\_MTWTF\_}

説明

week

SMTWTFS を指定します。これは、一週間の曜日 (日、月、火、水、木、金、土) を表しています。営業日として指定しない日は、その曜日をアンダースコア(\_) に置換する必要があります。

曜日の文字の位置が正しくない場合、または文字をアンダースコア ( \_ ) 以外の記号に置換した場合は、エラーメッセージが表示されます。デフォルト値は `_MTWTF_` ( \_月火水木金 \_ ) です。

### BYDISPLAY

通常、ソートグループでは、ソートフィールドの値はそのソートグループの先頭行または左端の列にのみ表示されます。ただし、SET BYDISPLAY コマンドを使用すると、レポートのすべての行に BY または ACROSS フィールドの値を表示することができます。SET BYDISPLAY コマンドはすべての出力フォーマットでサポートされますが、特に Excel フォーマットに出力する場合は、後からデータ操作をしやすいレポート出力を作成するためにこのコマンドを使用すると便利です。それは、Excel 上で特定の列を基準にソートを行う場合、その列にブランク行がいくつも存在すると、ソートが正しく実行されないためです。

また、この機能を使用すると、1つのソートフィールドを2回指定する必要がなくなります。つまり、1つは表示フィールド用、もう1つはソートフィールド用 (NOPRINT オプションの使用) です。

構文は次のとおりです。

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

#### 説明

##### OFF

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

##### **ON または BY**

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

##### **ACROSS**

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

##### **ALL**

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

### BYPANEL

BYPANEL パラメータを使用すると、複数パネルで構成されるレポートの各パネルの左側に BY フィールドを表示することができます。

構文は次のとおりです。

**SET BYPANEL = option**

説明

**option**

次のいずれかです。

**ON** - 各レポートパネルにソートフィールドの値を繰り返し表示します。

**OFF** - 各レポートパネルにソートフィールドの値を繰り返し表示しません。フィールドは最初のページにのみ表示されます。また、フィールドが複数のパネルに分割されて表示される場合があります。これがデフォルト値です。

**0** - 各レポートパネルにソートフィールドの値を繰り返し表示しません。また、フィールドは複数のパネルに分割されて表示されません。

**n** - 各レポートパネルにソートフィールドの **n** 桁分を繰り返し表示します。n の値は、リクエストで指定したソートフィールドの総数以下にすることができます。

## CACHE

CACHE パラメータは、割り当てるキャッシュページ数を制御します。このコマンドは、ON TABLE SET では使用することはできません。

FOCUS データソースの 4 キロバイトページをメモリに格納しておき、データソースと BINS 間でバッファします。

プロシジャがデータソースページの読み込みを指示すると、FOCUS は最初に BINS を検索し、続いてキャッシュメモリ、ディスクのデータソースの順に検索します。キャッシュメモリにページが見つかった場合、FOCUS はディスク I/O を実行する必要はありません。

プロシジャがデータソースページの書き込みを指示すると、そのページが BINS からディスクに書き込まれます。また、ページが更新されると、キャッシュメモリとディスクのバージョンが一致するように、更新したページがキャッシュメモリにもコピーされます。読み込みの場合とは異なり、書き込み処理ではキャッシュメモリにはディスク I/O は保存されません。

FOCSORT もキャッシュに書き込まれます。キャッシュの空き容量がなくなると、ディスクに書き込まれます。最適な結果を得るには、データソース全体を格納するキャッシュと、リクエストに使用する FOCSORT のサイズを合計したキャッシュを設定します。特定のリクエストに対する FOCSORT のサイズを推定するには、? STAT コマンドを発行し、リスト表示された SORTPAGES の数をメモリ内のデータソースページの数に追加します。その合計に対して SET CACHE コマンドを発行します。キャッシュを 50 に設定すると、4 キロバイトページの 50 倍が連続した格納領域としてキャッシュに割り当てられます。

CACHE 設定をクリアするには、SET CACHE = n コマンドを発行します。このコマンドでバッファがクリアされます。キャッシュメモリ内の情報がすべて失われます。

構文は次のとおりです。

**SET CACHE = {0|n}**

説明

**0**

キャッシュに領域を割り当てません。キャッシュは無効になります。デフォルト値は 0 (ゼロ) です。

**n**

キャッシュメモリに割り当てる連続した格納領域の 4 キロバイトページの数です。この数の最小値は 2 で、最大値は利用可能なメモリ容量により決まります。

## CARTESIAN

CARTESIAN パラメータは、PRINT または LIST を含むリクエストに適用されます。

このパラメータは、マルチパスリクエストにおいての非相関データインスタンスのすべての組み合わせを含むレポートを作成します。このパラメータは、ACROSS を使用してキャンセルします。

構文は次のとおりです。

**SET CARTESIAN = {ON|OFF}**

説明

**ON**

非相関レコードで構成されたレポートを生成します。

**OFF**

直積を無効にします。デフォルト値は OFF です。

## CDN

CDN パラメータは、数字の表記に使用する区切り文字を指定します。

TABLE リクエストの出力では、コンチネンタル 10 進表記 (CDN) がサポートされます。DEFINE および COMPUTE コマンドではサポートされません。

構文は次のとおりです。

**SET CDN = option**

## 説明

## option

次のいずれかです。

**DOTS\_COMMA** または **ON** CDN を使用します。小数点にはカンマ (,)、3 桁ごとの区切りにはピリオド (.) を使用します。たとえば、「3,045,000.76」は「3.045.000,76」と表記されます。ドイツ、デンマーク、イタリア、スペイン、ブラジルでは ON を使用します。

**注意:** CDN ON を使用する複数の数値パラメータは、カンマ (,) とそれに続くブランクで区切る必要があります。

**COMMAS\_DOT** または **OFF** CDN をオフにします。たとえば、「3,045,000.76」は「3,045,000.76」と表記されます。デフォルト値は OFF です。日本、米国、カナダ、メキシコ、イギリスでは OFF を使用します。

**SPACES\_COMMA** または **SPACE** 3 桁ごとの区切り文字にブランク、小数点にカンマ (,) を使用します。たとえば、「3,045,000.76」は「3 045 000,76」と表記されます。フランス、ノルウェー、スウェーデン、フィンランドでは SPACE を使用します。

**SPACES\_DOT** または **SPACEP** 3 桁ごとの区切り文字にブランク、小数点にピリオド (.) を使用します。たとえば、「3,045,000.76」は「3 045 000.76」と表記されます。

**QUOTES\_COMMA** または **QUOTE** 3 桁ごとの区切り文字に一重引用符 (')、小数点にカンマ (,) を使用します。たとえば、「3,045,000.76」は「3'045'000,76」と表記されます。スイスでは QUOTE を使用します。

**QUOTES\_DOT** または **QUOTEP** 3 桁ごとの区切り文字に一重引用符 (')、小数点にピリオド (.) を使用します。たとえば、「3,045,000.76」は「3'045'000.76」と表記されます。

**注意:** レポートの表示フォーマットが Excel 2000 以降の場合、CDN はユーザのコンピュータの設定により制御されます。つまり、レポート出力上の数値フォーマットは、地域に設定されたロケール (場所) またはブラウザの言語オプションの規則に基づいて、設定されます。

## CENT-ZERO

CENT-ZERO パラメータは、小数部のみの数字に先頭のゼロを表示します。小数点の表記をピリオド (.) にするか、カンマ (,) にするかは、CDN の設定で決まります。

構文は次のとおりです。

**SET CENT-ZERO = {ON|OFF}**

### 説明

#### ON

数字の先頭にゼロを追加します。CDN の設定に基づいて、数字の前に小数点を表すピリオド (.) またはカンマ (,) が追加されます。

#### OFF

先頭のゼロを表示しません。CDN の設定に基づいて、数字の前に小数点を表すピリオド (.) またはカンマ (,) が追加されます。デフォルト値は OFF です。

## CNOTATION

CNOTATION パラメータは、リクエストに対して作成された内部マトリックスの各列に連続列番号を割り当てます。このパラメータを COMPUTE および RECAP コマンドで使用すると、リクエスト内でそのフィールドを参照することができます。

列番号は内部マトリックスの各列を参照する番号のため、データの取得と集計が完了した段階で割り当てられます。レポート出力に実際にフィールドが表示されなくても、そのフィールドが内部マトリックスに存在している場合があります。たとえば、リクエストで計算値を使用する場合、その計算値用として内部マトリックスに 1 つまたは複数のフィールドが生成されます。フィールドに NOPRINT オプションを指定した場合、そのフィールド用として内部マトリックスのフィールドが 1 つ分占有されます。また、フィールドのフォーマットを再設定した場合、そのフィールドの値に使用するフィールドが新しく生成されます。FORECAST または REGRESS などの特定の RECAP 演算を実行する場合は、内部マトリックスに複数のフィールドが生成されます。

内部マトリックスでは BY フィールドに列番号は割り当てられませんが、デフォルト設定でその他のすべてのフィールドに列番号が割り当てられます。つまり、リクエストで列番号を参照する場合は、内部的に生成されたすべてのフィールドを念頭に置いて正しい列番号を指定する必要があります。このデフォルト設定の列割り当てを変更するには、次の 2 つのコマンドを使用します。SET CNOTATION=PRINTONLY コマンドを使用すると、レポート出力に表示するフィールドにのみ列番号が割り当てられます。SET CNOTATION=EXPLICIT コマンドを使用すると、リクエストで参照されているフィールドにのみ列番号が割り当てられます。

構文は次のとおりです。

**SET CNOTATION={ALL | PRINTONLY | EXPLICIT}**

### 説明

#### ALL

内部マトリックスのすべての列に列番号を割り当てます。デフォルト値は ALL です。

**PRINTONLY**

レポート出力に表示するフィールドにのみ列番号を割り当てます。

**EXPLICIT**

フィールドの表示、非表示に関係なく、リクエストで参照されるすべてのフィールドに列参照番号を割り当てます。

**注意：**この設定は ON TABLE 句ではサポートされません。

**CNVERR**

フォーマット変換エラーは、数値 USAGE フォーマットで記述されたフィールドに文字値が含まれる場合に発生します。

CNVERR パラメータは、データの切り捨てや変換エラーの原因となる値をデフォルト値で置き換えるようデータエンジンに指示を出します (フィールドが NULL 値可の場合は NULL、それ以外の場合は 0 (ゼロ))。

構文は次のとおりです。

```
SET CNVERR = {REJECT|IGNORE}
```

説明

**REJECT**

無効なデータを含む行を処理せず、警告メッセージを発行します。これがデフォルト値です。

**IGNORE**

無効なデータを含む行を受容し、デフォルト値で置換します (フィールドが NULL 値可の場合は NULL、それ以外の場合は 0 (ゼロ))。

**CNVERRLIMIT**

フォーマット変換エラーは、数値 USAGE フォーマットで記述されたフィールドに文字値が含まれる場合に発生します。

CNVERRLIMIT パラメータは、データ変換エラーに関する警告のレコード数に対して、最大 999999 個までの上限を設定します。

構文は次のとおりです。

```
SET CNVERRLIMIT = {0|nnnnnn}
```

### 説明

**nnnnnn**

最大 999999 までの整数値です。上限に達した場合、処理が終了し、システム変数 FOCERRNUM が 1346 に設定されます。デフォルト値 (0) では、フォーマット変換エラーが上限なく許容されます。

## COLLATION

COLLATION パラメータは、2 つの文字値の比較に関連するすべての言語要素の順位付けと一致を制御します。

構文は次のとおりです。

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

### 説明

**BINARY**

バイナリ値に基づいて、照合順序を設定します。

**SRV\_CI**

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

**SRV\_CS**

言語の設定に基づいて、照合順序を設定します。大文字と小文字は区別されます。

**CODEPAGE**

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

## COMPMISS

リクエストでフィールドのフォーマットを再設定すると (例、SUM フィールドのフォーマット)、フォーマットを再設定したフィールドの値を格納するための内部 COMPUTE フィールドが作成され、レポート出力に表示されます。元のフィールドにミッシング値が含まれている場合は、COMPMISS パラメータを ON に設定することで、そのミッシング値を内部フィールドに継承することができるようになります。ミッシング値が内部フィールドに継承されない場合は、ゼロ (数字フィールド) またはブランク (文字フィールド) が表示されます。ミッシング値が内部フィールドに継承される場合は、レポート出力にミッシングデータシンボルが表示されます。

構文は次のとおりです。

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

説明

**ON**

ミッシング値をフォーマットが再設定されたフィールドに継承させます。デフォルト値は ON です。

**OFF**

フォーマットが再設定されたフィールドにブランクまたは 0 (ゼロ) を表示します。

## COMPOUND

COMPOUND パラメータは、複合レポートの作成に使用し、複数のレポートを 1 つの PDF ファイルに統合します。COMPOUND を使用すると、スタイルが設定されたレポートフォーマット (PDF、HTML、Analytic Document、PowerPoint、Excel) のレポートを連結することができます。複合レポート内に、イメージとして保存したグラフを含むイメージファイルも埋め込むことができます。

複合レポートにタイプの異なるレポートが含まれている場合、構文は次のとおりです。

```
SET COMPOUND = {OPEN|CLOSE} [NOBREAK]
```

または

```
ON TABLE SET COMPOUND {OPEN|CLOSE}
```

この構文を使用する場合、連結する各タイプのレポートの表示フォーマットを識別するために、次のコードも含める必要があります。

```
ON TABLE {PCHOLD|HOLD|SAVE} [AS name] FORMAT formatname
```

複合レポート内のすべてのレポートが同一タイプ (PDF) の場合、構文は次のとおりです。

```
ON TABLE {PCHOLD|HOLD|SAVE} [AS name] FORMAT {PDF|PS} {OPEN|CLOSE} [NOBREAK]
```

説明

**name**

ファイルの名前です。この名前は、複合レポートの最初のレポートから取得されます。最初のレポートに名前が指定されていない場合は、「HOLD」という名前が使用されます。

**formatname**

スタイルが設定されたレポートフォーマットの名前です。有効なフォーマットは、PDF、HTML、AHTML、PPT、EXL2K です。

**OPEN**

最初のレポートで指定され、連結プロセスを開始します。OPEN 属性を含むレポートは、PDF フォーマットである必要があります。デフォルト値は OPEN です。

### CLOSE

最後のレポートで指定され、連結プロセスを終了します。

### NOBREAK

改ページさせない場合に使用するオプションです。デフォルト設定では、各レポートは別ページに表示されます。NOBREAK を使用すると、同じページにレポートを表示することができます。

### 注意

- ❑ 複合レポートからの出力を保存することができます。
- ❑ 複合レポートをネストすることはできません。
- ❑ 複数枠レポートは、複合レポートでは使用できません。

## COMPUTE

COMPUTE パラメータは、リクエスト実行時の演算のコンパイルを制御します。

構文は次のとおりです。

```
SET COMPUTE = {COMPILED|OLD}
```

### 説明

#### COMPILED

リクエスト実行時に式をコンパイルし、リクエストで使用する式のみをコンパイルします。COMPILED がデフォルト値です。

#### OLD

OLD の値は廃止され、COMPILED として機能します。

## COUNTWIDTH

COUNTWIDTH パラメータは、COUNT フィールドのデフォルトフォーマットを 5 バイトの整数から 9 バイトの整数、またはユーザのオペレーティング環境でサポートされる特定の整数フォーマットに拡張します。

構文は次のとおりです。

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

説明

**ON**

COUNT フィールドのデフォルトフォーマットを 5 バイトの整数から 9 バイトの整数に拡張します。

**OFF**

COUNT フィールドのデフォルトフォーマットを 5 バイトの整数から 9 バイトの整数に拡張しません。デフォルト値は OFF です。

**n**

オペレーティング環境でサポートされる最大整数フォーマットにまで、COUNT フィールドの幅を指定可能にします。

## CSSURL

CSSURL パラメータは、HTML レポートを外部カスケードスタイルシート (CSS) ファイルにリンクして、レポートのスタイルを設定します。

構文は次のとおりです。

**SET CSSURL = link**

説明

**link**

カスケードスタイルシート (CSS) ファイルの URL です。この URL には、絶対リンクまたは相対リンクのいずれかを指定することができます。

## CURRENCY\_DISPLAY

このパラメータは、通貨値の位置を基準に通貨記号の相対位置を定義します。

構文は次のとおりです。

**SET CURRENCY\_DISPLAY = pos**

説明

**pos**

数値の位置を基準にして通貨記号の相対位置を定義します。デフォルト値は default です。この設定では、現在有効になっているフォーマットと通貨記号の位置が使用されます。有効な値には、次のものがあります。

❑ **LEFT\_FIXED** 通貨記号を数値の前に左揃えで配置します。

- ❑ **LEFT\_FIXED\_SPACE** 通貨記号を数値の前に左揃えで配置し、記号と数値の間に少なくとも 1 つ空白を挿入します。
- ❑ **LEFT\_FLOAT** 通貨記号を数値の前に配置し、記号と数値の間に空白を挿入しません。
- ❑ **LEFT\_FLOAT\_SPACE** 通貨記号を数値の前に配置し、記号と数値の間に空白を 1 つ挿入します。
- ❑ **TRAILING** 通貨記号を数値の後に配置し、記号と数値の間に空白を挿入しません。
- ❑ **TRAILING\_SPACE** 通貨記号を数値の後に配置し、記号と数値の間に空白を 1 つ挿入します。

**注意：**この設定は、FORMAT EXL2K レポート出力ではサポートされません。

### CURRENCY\_ISO\_CODE

このパラメータは、使用する通貨記号の ISO コードを定義します。

構文は次のとおりです。

```
SET CURRENCY_ISO_CODE = iso
```

説明

*iso*

標準の 3 文字通貨コードです (例、米ドルは USD、日本円は JPY)。デフォルト値は default です。この設定では、構成済み言語コードの通貨コードが使用されます。

**注意：**この設定は、FORMAT EXL2K レポート出力ではサポートされません。

### CURRENCY\_PRINT\_ISO

このパラメータは、表示されるフィールドのフォーマットに !C または :C 拡張通貨記号が含まれる場合に、現在有効なコードページに従って通貨記号を表示できない場合の処理方法を定義します。

構文は次のとおりです。

```
SET CURRENCY_PRINT_ISO = {DEFAULT|ALWAYS|NEVER}
```

説明

DEFAULT

現在のコードページで通貨記号を表示できない場合は、ISO コードに置換されます。これがデフォルト値です。

**ALWAYS**

通貨記号を通貨の ISO コードに常に置換します。

**NEVER**

通貨記号を通貨の ISO コードに置換しません。現在のコードページで通貨記号を表示できない場合は、通貨記号を表示しません。

**注意**

- ☐ Unicode 環境を使用すると、すべての通貨記号を表示することができますが、それ以外の環境ではこの設定が必要です。
- ☐ このパラメータは、FORMAT EXL2K レポート出力ではサポートされません。

**CURRSYMB**

CURRSYMB パラメータは、数値フォーマットに表示オプションの M または N が指定されている場合に、通貨記号を設定します。デフォルトの通貨記号はコードページによって異なります。

構文は次のとおりです。

**SET CURRSYMB = symbol**

**説明****symbol**

表示可能な任意の文字またはサポートされている通貨コードです。

**注意：**ドル記号 (\$) を文字として指定するには、一重引用符 (') で囲む必要があります。

- ☐ USD または '\$' - 米ドルを指定します。
- ☐ GBP - イギリスポンドを指定します。
- ☐ JPY - 日本円を指定します。
- ☐ EUR - ユーロを指定します。
- ☐ NIS - イスラエル新シェケルを指定します。

## CURSYM\_D

CURSYM\_D パラメータは、数値フォーマット指定で !D、:D、!d、:d 表示オプションのいずれかが使用されている場合に、通貨記号に使用する文字を指定します。これらの表示オプションを使用した場合、デフォルト設定で、浮動 (D) ドル記号または固定 (d) ドル記号が数字の左側に表示されます。

構文は次のとおりです。

```
SET CURSYM_D = currsym
```

説明

`currsym`

最大で 4 バイトの表示可能文字を指定します。

## CURSYM\_E

CURSYM\_E パラメータは、数値フォーマット指定で !E、:E、!e、:e 表示オプションのいずれかが使用されている場合に、通貨記号に使用する文字を指定します。これらの表示オプションを使用した場合、デフォルト設定で、浮動 (E) ユーロ記号または固定 (e) ユーロ記号が数字の左側に表示されます。

構文は次のとおりです。

```
SET CURSYM_E = currsym
```

説明

`currsym`

最大で 4 バイトの表示可能文字を指定します。

## CURSYM\_F

CURSYM\_F パラメータは、数値フォーマット指定で !F または :F 表示オプションが使用されている場合に、通貨記号に使用する文字を指定します。これらの表示オプションを使用した場合、デフォルト設定で、浮動ユーロ記号が数字の右側に表示されます。このコマンドでは、数字と通貨記号の間にブランクを追加することができます。

構文は次のとおりです。

```
SET CURSYM_F = currsym
```

説明

`currsym`

最大で 4 バイトの表示可能文字を指定します。これらの文字にブランクが含まれている場合、文字全体を一重引用符 (') で囲む必要があります。

## CURSYM\_G

CURSYM\_G パラメータは、数値フォーマット指定で !G または :G 表示オプションが使用されている場合に、通貨記号に使用する文字を指定します。これらの表示オプションを使用した場合、デフォルト設定で、浮動ドル記号が数字の右側に表示されます。このコマンドでは、数字と通貨記号の間にブランクを追加することができます。

構文は次のとおりです。

`SET CURSYM_G= currsym`

説明

`currsym`

最大で 4 バイトの表示可能文字を指定します。これらの文字にブランクが含まれている場合、文字全体を一重引用符 (') で囲む必要があります。

## CURSYM\_L

CURSYM\_L パラメータは、数値フォーマット指定で !L、:L、!l、:l 表示オプションのいずれかが使用されている場合に、通貨記号に使用する文字を指定します。これらの表示オプションを使用した場合、デフォルト設定で、浮動 (L) ポンド記号または固定 (l) ポンド記号が数字の左側に表示されます。

構文は次のとおりです。

`SET CURSYM_L = currsym`

説明

`currsym`

最大で 4 バイトの表示可能文字を指定します。

## CURSYM\_Y

CURSYM\_Y パラメータは、数値フォーマット指定で !Y、:Y、!y、:y 表示オプションのいずれかが使用されている場合に、通貨記号に使用する文字を指定します。これらの表示オプションを使用した場合、デフォルト設定で、浮動 (Y) または固定 (y) の日本円記号または中国元記号が数字の左側に表示されます。

構文は次のとおりです。

```
SET CURSYM_Y = currsym
```

説明

`currsym`

最大で 4 バイトの表示可能文字を指定します。

### CUSTOM-PAGE-LENGTH

CUSTOM-PAGE-LENGTH パラメータは、DHTML、PDF、PPTX 出力レポートのページの長さを設定します。この SET パラメータは、PAGESIZE=CUSTOM で使用します。

構文は次のとおりです。

```
CUSTOM-PAGE-LENGTH = number
```

説明

`number`

UNITS パラメータ (インチ、センチメートル、ポイント) によって異なります。デフォルト値は 11 インチです。

### CUSTOM-PAGE-WIDTH

CUSTOM-PAGE-WIDTH パラメータは、DHTML、PDF、PPTX 出力レポートのページの幅を設定します。この SET パラメータは、PAGESIZE=CUSTOM で使用します。

構文は次のとおりです。

```
CUSTOM-PAGE-WIDTH = number
```

説明

`number`

UNITS パラメータ (インチ、センチメートル、ポイント) によって異なります。デフォルト値は 8.5 インチです。

### DATE\_ORDER

このパラメータは、日付構成要素の表示順序を定義します。

構文は次のとおりです。

```
SET DATE_ORDER = {DEFAULT|DMY|MDY|YMD}
```

## 説明

DEFAULT

日付構成要素の元の順序を優先します。これがデフォルト値です。

## DMY

すべての日付は日/月/年と表示されます。

## MDY

すべての日付は月/日/年と表示されます。

## YMD

すべての日付は年/月/日と表示されます。

## 注意

- ❑ `DATE_ORDER` は、すべての日付表示および日付時間表示に対して指定されている日付順序を上書きします。使用範囲を `DATE_ORDER` を使用したリクエストのみに制限するには、`ON TABLE SET` 句を使用します。
- ❑ この設定をダイアログマネージャシステム変数 (例、`&DATE`、`&TOD`、`&YMD`、`&DATEfmt`、`&DATXfmt`) とともに使用するには、システム変数に `.DATE_LOCALE` 接尾語を追加します。これにより、ローカライズされているシステム変数と、ローカライズされていないシステム変数の共存が可能になります。
- ❑ このパラメータは、`FORMAT EXL2K` レポート出力ではサポートされません。

**DATE\_SEPARATOR**

このパラメータは、日付構成要素を表示する際の区切り文字を定義します。

構文は次のとおりです。

`SET DATE_SEPARATOR = separator`

## 説明

*separator*

次のいずれかの値です。

- ❑ **DEFAULT** フィールドの `USAGE` フォーマットで定義された区切り文字を使用します。
- ❑ **SLASH** 日付構成要素の区切り文字にスラッシュ (/) を使用します。
- ❑ **DASH** 日付構成要素の区切り文字にハイフン (-) を使用します。

- ❑ **BLANK** 日付構成要素の区切り文字にブランクを使用します
- ❑ **DOT** 日付構成要素の区切り文字にピリオド (.) を使用します。
- ❑ **NONE** 日付構成要素に区切り文字を使用しません。

注意

- ❑ **DATE\_SEPARATOR** は、すべての日付表示および日付時間表示の日付区切り文字を上書きします。ただし、これらのパラメータに変換表示オプション (T、Tr、t、tr) が含まれている場合は、指定された区切り文字で出力されます。
- ❑ この設定をダイアログマネージャシステム変数 (例、&DATE、&TOD、&YMD、&DATEfmt、&DATXfmt) とともに使用するには、システム変数に **.DATE\_LOCALE** 接尾語を追加します。これにより、ローカライズされているシステム変数と、ローカライズされていないシステム変数の共存が可能になります。
- ❑ このパラメータは、**FORMAT EXL2K** レポート出力ではサポートされません。

DATEDISPLAY

DATEDISPLAY パラメータは、基準日の表示を制御します。従来は、ファイルから読み込んだ日付が基準日と一致した場合、または SmartDate フォーマットのフィールド値が 0 (ゼロ) の場合、TABLE は常にブランクを表示していました。下表は、サポートされている日付フォーマットの基準日を示しています。

フォーマット	基準日
YMD および YYMD	1900/12/31
YM および YYM	1901/01
YQ および YYQ	1901/Q1
JUL および YYJUL	00/365 および 1900/365

**注意：** ON TABLE コマンドを使用して DATEDISPLAY を設定することはできません。

構文は次のとおりです。

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

説明

**ON**

日付が基準日の値と一致する場合は基準日を表示します。

**OFF**

日付が基準日の値と一致する場合はブランクを表示します。デフォルト値は OFF です。

## DATEFNS

DATEFNS パラメータは、2000 年対応バージョンの日付関数を有効にします。

構文は次のとおりです。

**SET DATEFNS = {ON|OFF}**

説明

**ON**

2000 年の日付をサポートしている日付関数を有効にします。デフォルト値は ON です。

**OFF**

この値は使用できなくなり、ON として動作します。

## DATEFORMAT

DATEFORMAT パラメータは、日時の値がフォーマット済みまたは変換済み文字列フォーマットで入力された場合の、日付要素 (月/日/年) の順序を指定します。このパラメータを使用すると、値の入力フォーマットが、その値に適用される変数のフォーマットに依存しなくなります。

構文は次のとおりです。

**SET DATEFORMAT = datefmt**

説明

**datefmt**

MDY、DMY、YMD、MYD のいずれかを指定することができます。U.S. English フォーマットでは、デフォルト値は MDY です。

## DATETIME

DATETIME パラメータは、レポート内の日付時間を設定し、CREATE SYNONYM がマスターファイルに作成する日付時間フィールドのフォーマットを制御します。

構文は次のとおりです。

**SET DATETIME = option**

説明

**option**

次のいずれかです。

- ❑ **STARTUP** セッションを開始した日時です。デフォルト値は STARTUP です。
- ❑ **CURRENT|NOW** データを取得するたびに値が変わります。たとえば、深夜 12 時直前の 11:59 P.M. にバッチジョブを開始する場合、そのジョブは翌日まで継続します。DATETIME を NOW|CURRENT に設定した場合、その変数を参照している値には、ジョブが開始した日付ではなく、現在の日付が出力されます。
- ❑ **RESET** セッションの残りの期間または SET DATETIME コマンドを新たに発行するまでの期間は、現在の実行日時で固定されます。

## DB\_INFILE

SET DB\_INFILE コマンドは、DB\_INFILE 関数で生成された式を、リレーショナルデータソースに対して使用できるよう最適化するかどうかを制御します。

構文は次のとおりです。

**SET DB\_INFILE = {DEFAULT|EXPAND\_ALWAYS|EXPAND\_NEVER}**

説明

DEFAULT

サブクエリの作成が可能と分析された場合に、DB\_INFILE によるサブクエリの作成を有効にします。これがデフォルト値です。

EXPAND\_ALWAYS

DB\_INFILE によるサブクエリの作成を無効にします。代わりに、式がメモリ内の IF および WHERE 句に拡張されます。

EXPAND\_NEVER

DB\_INFILE により、式がメモリ内の IF および WHERE 句に拡張されないようにします。代わりに、サブクエリの作成を試みます。サブクエリを作成できない場合、FOC32585 メッセージが生成され、処理が停止します。

## DBACSENSITIV

DBA またはユーザが SET USER、SET PERMPASS、SET PASS コマンドのいずれかを発行すると、DBA 属性を含むマスターファイルすべてのデータソースへのアクセスを許可する前に、このユーザ ID の認証情報が確認されます。このパスワードは、プロシジャを暗号化または復号化する際にも確認されます。

SET DBACSENSITIV コマンドは、確認の前にパスワードを大文字に変換するかどうかを指定します。

構文は次のとおりです。

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

説明

### ON

パスワードを大文字に変換しません。ユーザ定義のパスワードと、マスターファイルまたはプロシジャのパスワードを比較する際は、常に大文字と小文字が区別されます。

### OFF

確認前にパスワードを大文字に変換します。ユーザ定義のパスワードと、マスターファイルまたはプロシジャのパスワードを比較する際は、大文字と小文字は区別されません。デフォルト値は OFF です。

## DBAJJOIN

DBAJJOIN パラメータは、DBA 制限をレポートリクエストの WHERE 条件として処理するか、JOIN 条件として追加するかを制御します。

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

説明

### OFF

レポートリクエストで DBA 制限を WHERE フィルタとして処理します。デフォルト値は OFF です。

### ON

DBA 制限を JOIN 条件として処理します。

### DBASOURCE

DBASOURCE パラメータにより、マルチファイル構造へのアクセスを許可するセキュリティ属性が指定されます。デフォルト設定では、アクセス制限は JOIN 構造のホストファイル、または COMBINE 構造の最後のファイルに基づいて決定されます。DBASOURCE パラメータを「ALL」に設定した場合、JOIN または COMBINE 構造のファイルすべてのアクセス制限が適用されます。

JOIN 構造および COMBINE 構造のファイルには、すべて同一の DBA パスワードが設定されている必要があります。DBA 属性が同一でない場合、構造にアクセスする方法はありません。

SET DBASOURCE コマンドは 1 回のセッションまたは接続につき、一度だけ発行することができます。それ以上コマンドを発行しようとしても、2 回目以降は無視されます。値がプロファイルで設定されている場合、セッション中にユーザが変更することはできません。

DBASOURCE=ALL の場合は、次のようになります。

- ❑ JOIN 構造に対する TABLE リクエストでは、ユーザが構造内の各ファイルに最低読み取り権限を所有している場合に限り、クロスリファレンスファイルおよびクロスリファレンスセグメントへのアクセスが許可されます。

DBASOURCE=HOST の場合は、次のようになります。

- ❑ TABLE リクエストでは、ユーザは JOIN 構造のホストファイルの読み取り権限が必要です。セキュリティ制限は、すべてホストファイルから適用されます。構造内のファイルのセキュリティ制限を適用するには、DBAFILE を作成して有効にすることもできます。

構文は次のとおりです。

```
SET DBASOURCE = {HOST|ALL}
```

説明

#### [HOST](#)

DBAFILE を使用して、構造内の別ファイルのアクセス制限を適用する場合を除き、JOIN 構造のホストファイル、または COMBINE 構造の最後のファイルに、アクセス制限を限定します。デフォルト値は HOST です。

#### [ALL](#)

ユーザには、JOIN 構造および COMBINE 構造のファイルすべてに読み取り権限が必要です。このファイルに INCLUDE、DELETE、または UPDATE コマンドを発行する場合、ユーザに WRITE、UPDATE、または READ/WRITE アクセス権限が必要です。

## DEFCENT

DEFCENT パラメータは、デフォルトの西暦をグローバルに定義するか、明示的に西暦を指定していないアプリケーションのフィールドレベルで定義します。DEFCENT は YRTHRESH と併用し、指定した値に基づいて現在の西暦を解析します。グローバルに割り当てる場合、ファイルレベルまたはフィールドレベルの値を指定しない限り、これらのパラメータで作成された時間間隔が、アプリケーションが使用する 2 桁の年にそれぞれ適用されます。詳細は、662 ページの「[YRTHRESH](#)」を参照してください。

**注意：**マスターファイルに FDEFCENT および FYRTHRESH 属性を追加した場合でも、同一の結果を得ることができます。

構文は次のとおりです。

```
SET DEFCENT = {cc|19}
```

説明

[cc](#)

デフォルトの西暦です。値を指定しない場合、デフォルト値は 19 になります。20 世紀では、cc のデフォルト値は 19 です。

## DEFECHO

DEFECHO パラメータは、&ECHO 変数のデフォルト値を設定します。

構文は次のとおりです。

```
SET DEFECHO = {OFF|ON|ALL|NONE}
```

説明

[OFF](#)

&ECHO のデフォルト値として OFF を設定します。デフォルト値は OFF です。

[ON](#)

&ECHO のデフォルト値として ON を設定します。

[ALL](#)

&ECHO のデフォルト値として ALL を設定します。

[NONE](#)

プロシジャコードが非表示になります。DEFECHO または &ECHO を NONE に設定した場合、セッション中または接続中に変更することはできません。

### DEFINES

DEFINES パラメータは、一時項目 (DEFINE) をマシンコードにコンパイルすることにより、一時項目での演算を高速化します。

このパラメータは、Windows、UNIX オペレーティングシステムにのみ適用されます。

構文は次のとおりです。

```
SET DEFINES = {COMPILED|OLD}
```

説明

#### COMPILED

リクエストの実行時に式をコンパイルします。リクエストで使用する DEFINE のみがコンパイルされます。COMPILED がデフォルト値です。

#### OLD

OLD の値は廃止され、COMPILED として機能します。

### DIRECTHOLD

DIRECTHOLD パラメータは、内部的に生成される MODIFY プロシジャおよび中間シーケンシャルファイルを作成せずに、HOLD ファイルを FOCUS フォーマットで直接作成します。

このパラメータは、Windows および UNIX オペレーティングシステムにのみ適用されます。

構文は次のとおりです。

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

説明

#### ON

中間ファイルおよび MODIFY プロシジャを使用せずに、FOCUS HOLD ファイルを生成します。デフォルト値は ON です。この値のみがサポートされます。OFF は、構文の下位互換性をサポートするために提供されていますが、動作は ON と同一です。

#### OFF

OFF は、構文の下位互換性をサポートするために提供されていますが、動作は ON と同一です。

## DISPLAYROUND

DISPLAYROUND パラメータは、浮動小数点数および倍精度浮動小数点数の表示時の端数処理を制御します。デフォルト値の ON は、表示する前に非常に小さい数値 (最小精度単位、ULP) を追加します。これにより、2 進数値の小数部が  $1/2$  (.1) になり、小数部が .49999... (.50000 ではなく) に変換されるため、切り上げではなく切り捨てられた数値の表示が補正されます。これは数値の表示のみに影響し、演算または格納時には影響しません。

構文は次のとおりです。

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

説明

### ON

浮動小数点数および倍精度浮動小数点数の表示時の端数処理エラーを補正します。これがデフォルト値です。

### OFF

浮動小数点数および倍精度浮動小数点数の表示時の端数処理エラーを補正しません。

## DMH\_LOOPLIM

DMH\_LOOPLIM パラメータは、-REPEAT または -GOTO コマンドで作成されるダイアログマネージャループ反復の最大許容数を設定します。

構文は次のとおりです。

```
SET DMH_LOOPLIM = n
```

説明

*n*

ループ反復の最大許容数を設定します。デフォルト値は 0 (ゼロ) です。この設定では、ループ反復数に制限はありません。

DMH\_LOOPLIM は、セッション全体でエラーを発生させずに、既存のレポートおよびプロシジャを実行するために必要十分な値に設定する必要があります。このパラメータを設定する場合は、プロファイルで設定する方法をお勧めします。

## DMH\_STACKLIM

DMH\_STACKLIM パラメータは、FOCSTACK に配置可能な最大行数を設定します。

構文は次のとおりです。

```
SET DMH_STACKLIM = n
```

### 説明

*n*

FOCSTACK に配置可能な最大行数を設定します。デフォルト値は 0 (ゼロ) です。この設定では、スタックコマンド数に制限はありません。

DMH\_STACKLIM は、セッション全体でエラーを発生させずに、既存のレポートおよびプロシジャを実行するために必要十分な値に設定する必要があります。このパラメータを設定する場合は、プロファイルで設定する方法をお勧めします。

## DMPRECISION

DMPRECISION パラメータは、ダイアログマネージャの -SET コマンドでの数値精度を指定します。

倍精度演算を使用する場合でも、この演算精度を設定していないと、数値計算の結果は整数値として返されます。この演算精度を設定せずに 10 進数精度で数値を返すには、FTOA サブルーチンへの入力として演算を入力する必要があります。FTOA サブルーチンには、戻り値の小数点以下の桁数を設定することができます。

構文は次のとおりです。

```
SET DMPRECISION = {OFF|n}
```

### 説明

OFF

小数点以下の切り捨て (端数処理を行わない) を指定します。デフォルト値は OFF です。

*n*

端数処理の桁数を指定する 0 (ゼロ) から 9 の正の数です。n に 0 (ゼロ) を指定すると、端数処理された整数値が表示されます。

## DRILLFOCMISSING

DRILLFOCMISSING パラメータを使用すると、MISSING 値のドリルダウン値として \_FOC\_MISSING 文字列を渡すか、ピリオド (.) を渡すかを制御することができます。

構文は次のとおりです。

```
SET DRILLFOCMISSING = {ON|OFF}
ON TABLE SET DRILLFOCMISSING {ON|OFF}
```

説明

**ON**

MISSING データのドリルダウン値として `_FOC_MISSING` 文字列を渡します。デフォルト値は ON です。

**OFF**

MISSING データのドリルダウン値としピリオド (.) を渡します。

## DRILLMETHOD

DRILLMETHOD パラメータは、ドリルダウンリクエストで GET メソッドの代わりに POST メソッドを使用することを可能にします。URL およびログにパラメータと値を表示しないため、POST は GET よりも安全です。また、URL 上の 2048 バイト (2K) の文字制限も適用されません。

構文は次のとおりです。

```
SET DRILLMETHOD = {HREF|POST|GET}
ON TABLE SET DRILLMETHOD {HREF|POST|GET}
```

説明

**HREF**

WebFOCUS Servlet を呼び出し、プロシジャ名を送信します。デフォルト値は HREF です。

**POST**

パラメータと値を URL に表示せず、ログに保存しません。

**GET**

パラメータと値を URL に表示し、ログに保存します。

**注意**

- ☐ DRILLMETHOD パラメータは、HTML および DHTML レポートのみでサポートされます。
- ☐ POST および GET メソッドの両方で、ページソースを表示する際は、パラメータと値が表示されます。

## DROPBLNKLINE

DROPBLNKLINE パラメータは、レポート出力のフォーマット設定時に、中間合計、中間見出し、中間脚注の周囲のブランク行を削除します。また、特定のデータ行は、ブランクにした場合、レポートにブランク行として表示されます。SET DROPBLNKLINE=ON を使用して、これらのブランク行を削除することができます。

この設定は、出力フォーマット HOLD/PCHOLD/SAVE フォーマットの ALPHA、INTERNAL、BINARY、COM、COMT、COMMA、TAB、TABT、FIX、DFIX、すべての DBMS、VSAM、LOTUS、SYLK、DIF、FOCUS、XFOCUS には適用されません。

構文は次のとおりです。

**SET DROPBLNKLINE = {OFF|ON|BODY|HEADING|ALL}**

説明

### OFF

システムが生成するブランク行および空のデータ行を挿入します。デフォルト値は OFF です。

### ON|BODY

レポート本文内のシステム生成のブランク行を削除します (例、中間見出し前後のブランク行)。また、値が存在しないためにレポート出力でブランク行になる特定のデータ行を出力結果から削除します。BODY は ON の同義語です。

### HEADING

見出しとタイトル間のブランク行、およびレポート本文と脚注間のブランク行を削除します。リクエストでレポート内の任意の位置に境界線または背景色スタイルシート属性が使用されている、固定レイアウトのフォーマット (PDF、DHTML、PPT、PPTX) で動作します。

### ALL

ON と HEADING の両方の動作が適用されます。

## DTSTRICT

DISTRICT パラメータは、厳密処理の使用を制御します。厳密処理は、エンドユーザの入力、トランザクションからの読み取り、表示、サブルーチンからの取得時に、日付時間値をチェックし、これらが有効な日付時間であることを確認します。たとえば、数字の月は 1 から 12 の間で、日は特定の月の日数内でなければなりません。

構文は次のとおりです。

**SET DTSTRICT = {ON|OFF}**

説明

### ON

厳密処理を呼び出します。デフォルト値は ON です。

**OFF**

厳密処理を呼び出しません。日付時間構成要素には、フィールドで使用可能な桁数の制限値内であれば、任意の値を指定することができます。たとえば、フィールドが 2 桁の月の場合、値 12 および 99 は有効ですが、115 は無効です。

**DUPLICATECOL**

DUPLICATECOL パラメータは、複数の表示コマンドを使用するリクエストのフォーマットを再設定し、集計フィールドを同一列内の表示フィールドの上部に配置します。

構文は次のとおりです。

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

説明

**ON**

フィールドごとに新しい列が作成されてレポートに表示されます。デフォルト値は ON です。

**OFF**

共通フィールドはすべて、レポートの同一列内に表示されます。

**EMBEDDABLE**

EMBEDDABLE パラメータは、HTML5 グラフ出力でのドキュメントレベルの HTML タグ (例、<html>、<head>、<body>) の生成を制御します。これにより、HTML の 1 ページ内に複数の HTML5 グラフを埋め込むことが可能になります。

構文は次のとおりです。

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

説明

**OFF**

ドキュメントレベルの HTML タグを含む完全な HTML レポート出力を生成します。これがデフォルト値です。

**ON**

ドキュメントレベルのタグを使用せずに、HTML フォーマットでレポート出力を生成します。-HTMLFORM で使用される HTML5 グラフを作成する場合は、この設定を使用する必要があります。

**注意：**SET EMBEDDABLE=ON は、HTML レポート出力および Java ベースのグラフフォーマットにも影響します。これらのフォーマットは、HOLD FORMAT HTMTABLE の使用に相当します。

### EMBEDHEADING

EMBEDHEADING パラメータは、見出しテキストまたは脚注テキストを、グラフィイメージの一部として表示するか、グラフィイメージとは別に表示するかを制御します。

構文は次のとおりです。

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

説明

ON

見出しと脚注を、グラフィイメージの一部として表示します。

OFF

見出しと脚注を、グラフィイメージとは別のテキスト要素として表示します。これがデフォルト値です。

### EMPTYCELLS

数値フィールドで EMPTYCELLS パラメータを使用すると、XLSX 出力フォーマットのフィールドの MISSING オプションを処理して、数式バーに表示されるローデータとして、値が存在しない (空の) セルの代わりに、MISSING の値 0 (ゼロ) を表示できます。

構文は次のとおりです。

```
SET EMPTYCELLS = {ON|ZEROVALUE|XLSXOFF}
ON TABLE SET EMPTYCELLS {ON|ZEROVALUE|XLSXOFF}
```

説明

ON

数値フィールドの場合、値が MISSING であるセルにブランクのセルを作成します。デフォルト値は ON です。

ZEROVALUE

数値フィールドの場合、値が MISSING であるセルに 0 (ゼロ) のローデータ値を挿入します。これは、Excel 2007 以降 (.xlsx 出力フォーマット) に適用されます。

## XLSXOFF

値が MISSING の場合、文字フィールドに対してはブランクのセルを生成し、数値フィールドに対しては値 0 (ゼロ) のセルを生成します。

## EMPTYREPORT

EMPTYREPORT パラメータは、TABLE リクエストで取得したレコード数がゼロ件の場合に生成する出力を制御します。

EMPTYREPORT は、TABLEF および Excel ではサポートされません。TABLEF または Excel リクエストが取得したレコード数がゼロ件の場合は、空のレポートが生成されます。

**注意:** EMPTYREPORT を OFF に設定して IF TOTAL または WHERE TOTAL を使用し、その TOTAL 条件を満たすデータが存在しない場合は、空のレポートが作成されます。これは、TOTAL 条件が適用される前に、レポート行に対して EMPTYREPORT のテストが適用されるためです。

構文は次のとおりです。

```
SET EMPTYREPORT={ANSI|ON|OFF}
```

### 説明

#### ANSI

COUNT がリクエストされている場合は、単一行レポートを生成し、ミッシングデータ文字またはゼロを表示します。いずれの場合でも、&RECORDS が 0 になり、&LINES は 1 になります。

SQL トランスレータを呼び出す場合は、EMPTYREPORT のデフォルト設定が OFF から ANSI に自動的に置換されます。

#### ON

空のレポートを生成します。つまり、列見出しのみでコンテンツは何も表示されません。これは、以前のバージョンでのデフォルトの動作です。

#### OFF

レポート出力を生成しません。デフォルト値は OFF です。ただし、SQL トランスレータのリクエストは例外です。SQL トランスレータを呼び出す場合は、EMPTYREPORT パラメータのデフォルト設定が OFF から ANSI に自動的に置換されるため、結果は ANSI 設定と同一になります。

リクエスト内からコマンドを発行することもできます。

```
ON TABLE SET EMPTYREPORT ANSI
```

### EQTEST

EQTEST パラメータは、IF テスト、および 1 つまたは複数の IF テストに変換される WHERE テストで、「\$」と「\$\*」をワイルドカード文字として扱うか、通常の文字として扱うかを制御します。

構文は次のとおりです。

```
SET EQTEST = {WILDCARD|EXACT}
```

説明

#### WILDCARD

「\$」および「\$\*」を、ワイルドカード文字として扱います。デフォルト値は「WILDCARD」です。

#### EXACT

IF テスト、および IF テストに変換される WHERE テストで、「\$」と「\$\*」をワイルドカードではなく通常の文字として扱います。

### ERROROUT

ERROROUT パラメータは、バッチ FOCUS ジョブがプロシ ज्याで検出されたエラーに応答する方法を制御します。このパラメータは、ON TABLE SET コマンドで設定することはできません。

ERROROUT を ON に設定した場合、エラーメッセージが生成されるとジョブが終了し、リターンコードの 8 が返されます。警告メッセージの場合は、この処理は実行されません。

ERROROUT を OFF に設定した場合、それぞれのメッセージに応じて、FOCUS がプロシ ज्याの処理を続行するかどうかを決定します。ユーザは &FOCERRNUM などのダイアログマネージャ変数を確認し、次のコマンドを発行して FOCUS を停止するとともに、リターンコードに n を設定することができます。

```
-QUIT FOCUS n
```

```
exit rc
```

**注意：**ERROROUT の設定は、対話型セッションでは無視されます。

構文は次のとおりです。

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

説明

**ON**

バッチ FOCUS ジョブでエラーメッセージが生成されると、リターンコードを 8 に設定し、ジョブを終了します。

また、次のメッセージを表示し、プログラムが終了した理由をユーザに通知します。

`Exiting due to Exit on Error`

**OFF**

エラーメッセージの発生に応答し、リターンコードを設定したり、ジョブまたはプロシジヤを自動的に終了したりしません。デフォルト値は OFF です。

## ESTRECORDS

ESTRECORDS パラメータは、リクエストでソートするレコード数の推定値を渡します。

TABLE、MATCH、GRAPH リクエストで ESTRECORDS を設定するには、ON TABLE SET コマンドを使用する必要があります。

構文は次のとおりです。

`ON TABLE SET ESTRECORDS n`

説明

**n**

ソートするレコード数の推定値です。

## EUROFILE

EUROFILE パラメータは、換算する通貨の情報が格納されたデータソースを有効にします。セッション中にこの設定を変更し、別の通貨データソースにアクセスすることができます。このパラメータは、レポートリクエストで発行することはできません。

**注意：**EUROFILE と同一行に他のパラメータを設定することはできません。同一行で指定した他のパラメータは無視されます。

構文は次のとおりです。

`SET EUROFILE = {ddname|OFF}`

### 説明

#### ddname

使用する通貨データソースのマスターファイル名です。ddname は、FOCUS がアクセス可能な読み取り専用のデータソースを参照する必要があります。デフォルト値は設定されていません。

#### OFF

現在の通貨データソースを無効にしてメモリから削除します。

## EXCELSERVURL

EXCELSERVURL パラメータは、EXCEL 2007 ファイル (.xlsx) を構成するファイルコンポーネントの圧縮に使用する Application Server を指定します。

ファイルを開くマシンには、Excel 2007 または Excel 2003 が Microsoft Office 変換ユーティリティとともにインストールされている必要があります。Excel 2003 では、XLSX ファイルタイプを定義するとともに、WebFOCUS 管理コンソールの [出力先変更設定] 下で [保存レポート] 設定を [はい] に変更する必要があります。出力先変更設定を変更する方法についての詳細は、『TIBCO WebFOCUS セキュリティ管理ガイド』を参照してください。

構文は次のとおりです。

```
SET EXCELSERVURL = url
```

### 説明

#### url

WebFOCUS Client がインストールされている Web サーバでの WebFOCUS アプリケーションコンテキストルートの URL です (最大 256 バイト)。

以下はその例です。

```
http://hostname/ibi_apps//IBIEXCELSERVURL
```

### 説明

#### hostname

WebFOCUS Client がインストールされている Application Server 名です。

## EXL2KLANG

EXL2KLANG パラメータを nlscfg.err ファイルに追加すると、このパラメータで指定した言語が Microsoft Excel リクエストに使用されます。出力を正しく表示するには、この言語を Excel の言語と一致させる必要があります。

プロファイルまたはプロシジャで SET EXL2KLANG コマンドを記述し、エラーファイル内の設定を上書きすることができます。

構文は次のとおりです。

**EXL2KLANG** = {language|ENG}

説明

#### language

Excel で使用する言語です。有効な値には、次のものがあります。

- ☐ **ENG** - 英語 (デフォルト値は ENG です。)
- ☐ **FRE** - フランス語
- ☐ **GER** - ドイツ語
- ☐ **JPN** - 日本語
- ☐ **KOR** - 韓国語
- ☐ **SPA** - スペイン語

## EXL2KTXTDATE

EXL2KTXTDATE パラメータを使用すると、変換済み日付をテキストではなく、フォーマットマスクを適用した日付値として転送するよう指定することができます。

構文は次のとおりです。

**SET EXL2KTXTDATE** = {TEXT|VALUE}

説明

#### TEXT

テキストを含む日付値を、書式設定されたテキストとして Excel 2000 に転送します。デフォルト値は TEXT です。

#### VALUE

変換済み日付値にテキストが含まれ、その日付値が Excel でサポートされている日付フォーマットの場合に、テキストフォーマットマスクが適用された標準の日付値として Excel 2000 に転送します。

### EXPANDABLE

列単位で展開するアコーディオンレポートを生成します。

構文は次のとおりです。

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

説明

[OFF](#)

アコーディオンレポートを作成しません。デフォルト値は OFF です。

[ON](#)

列単位で展開されるアコーディオンレポートを作成します。

### EXPANDBYROW

同一列内で展開するアコーディオンレポートを作成します。

構文は次のとおりです。

```
SET EXPANDBYROW = {OFF|ON|ALL|n}
```

説明

[OFF](#)

アコーディオンレポートを作成しません。デフォルト値は OFF です。

[ON](#)

アコーディオンレポートを作成し、最上位のソートフィールドレベルのみを最初に表示します。下位レベルの行を表示するには、表示されたいずれかのソートフィールド値の横にあるプラス記号 (+) をクリックします。

[ALL](#)

アコーディオンレポートを作成し、最初にすべてのソートフィールドレベルを展開した状態で表示します。ソートフィールドレベルを折りたたむには、そのレベルのいずれかのソートフィールド値の横にあるマイナス記号 (-) をクリックします。

[n](#)

アコーディオンレポートを作成し、最初に n 個のソートフィールドレベルを展開した状態で表示します。展開されたソートフィールドレベルを折りたたむには、そのレベルのいずれかのソートフィールド値の横にあるマイナス記号 (-) をクリックします。

## EXPANDBYROWTREE

強化されたインターフェースを使用して、アコーディオンレポートを生成します。

構文は次のとおりです。

```
SET EXPANDBYROWTREE = {OFF|ON|ALL|n}
ON TABLE SET EXPANDBYROWTREE {OFF|ON|ALL|n}
```

説明

### OFF

強化されたインターフェースで、アコーディオンレポートを作成しません。デフォルト値は OFF です。

### ON

強化されたインターフェースで、アコーディオンレポートを作成します。この設定では、初期状態で最上位のソートフィールドレベルのみを表示します。下位レベルの行を表示するには、表示されたいずれかのソートフィールド値の横にあるプラス記号 (+) をクリックします。

### ALL

強化されたインターフェースで、アコーディオンレポートを作成します。この設定では、初期状態ですべてのソートフィールドレベルが展開されて表示されます。ソートフィールドレベルを折りたたむには、そのレベルのいずれかのソートフィールド値の横にあるマイナス記号 (-) をクリックします。

### n

強化されたインターフェースで、アコーディオンレポートを作成します。この設定では、初期状態で n 個のソートフィールドレベルが展開されて表示されます。展開されたソートフィールドレベルを折りたたむには、そのレベルのいずれかのソートフィールド値の横にあるマイナス記号 (-) をクリックします。

## EXTAGGR

EXTAGGR パラメータは、外部ソートを使用して集計を実行します。

構文は次のとおりです。

```
SET EXTAGGR = {ON|OFF|NOFLOAT}
```

説明

### ON

外部ソートによる集計を使用可能にします。デフォルト値は ON です。

OFF

外部ソートによる集計を使用不可にします。

NOFLOAT

浮動小数点データフィールドが存在しない場合に集計を使用可能にします。

EXTENDNUM

EXTENDNUM パラメータは、レポート出力の割り当て領域に表示値が収まらない場合にアスタリスク (\*) を表示するか、値が表示されるようレポートの列を拡張するかを制御します。

構文は次のとおりです。

SET EXTENDNUM = {ON|OFF|AUTO}

説明

ON

定義された USAGE フォーマットに関係なく、すべての値を省略せずに表示します。

OFF

USAGE フォーマットで割り当てられた列幅に値が収まらない場合にアスタリスク (\*) を表示します。これが従来の動作です。

AUTO

下表のように、出力フォーマットおよび SQUEEZE 設定に基づいて ON または OFF 設定を適用します。

フォーマット	SQUEEZE 設定	EXTENDNUM
PDF、DHTML、PPT、PPTX	ON	ON
	OFF	OFF
HTML、EXL2K、XLSX	なし	ON
WP、その他の区切りフォーマット	なし	OFF

デフォルト値は AUTO です。

## EXTHOLD

EXTHOLD パラメータを使用すると、外部ソートを使用して HOLD ファイルを作成することができます。

構文は次のとおりです。

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

説明

### ON

外部ソートを使用して HOLD ファイルを作成します。デフォルト値は ON です。

### OFF

外部ソートを使用して HOLD ファイルを作成しません。

## EXTRACT

EXTRACT パラメータは、リクエストに使用する構造化 HOLD ファイルを有効にします。

このパラメータは、ON TABLE 句を使用した TABLE または TABLEF リクエストでのみサポートされます。

構文は次のとおりです。

```
ON TABLE SET EXTRACT = {ON|*|OFF}
```

説明

### ON

このリクエストの構造化 HOLD ファイルを有効にし、リクエストに記述されたすべてのフィールドを抽出します。

★

このリクエストの構造化 HOLD ファイルを有効にし、その後の一連の抽出オプションが続くことを示します。たとえば、構造化 HOLD ファイルから特定のフィールドを除外することができます。

### OFF

このリクエストの構造化 HOLD ファイルを無効にします。デフォルト値は OFF です。

## EXTSORT

EXTSORT パラメータは、外部ソート機能を有効にし、TABLE、MATCH、GRAPH コマンドで使用可能にします。

レポート全体をメモリで処理できる場合、外部ソートは使用されません。

メモリ内でレポートを処理できるかどうか特定するには、TABLE、MATCH、GRAPH コマンドの後に ? STAT クエリを発行し、SORT USED パラメータの値を確認します。

スタイルシートが使用されている場合は、外部ソートは動作しません。

構文は次のとおりです。

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

説明

**ON**

レポートをソートする際に外部ソートを選択的に使用可能にします。デフォルト値は ON です。

**OFF**

レポートをソートする際に内部ソートプロシジャを使用します。

## FIELDNAME

FIELDNAME パラメータは、長いフィールド名および修飾フィールド名のサポートを制御します。

このコマンドは、ON TABLE SET で使用することはできません。

構文は次のとおりです。

```
SET FIELDNAME = {NEW|NOTRUNC|OLD}
```

説明

**NEW**

長いフィールド名および修飾フィールド名をサポートします。デフォルト値は NEW です。

**NOTRUNC**

長いフィールド名および修飾フィールド名はサポートしますが、一意の省略名はサポートしません。

**OLD**

このパラメータ値は、現在使用できません。この値は、NEW として機能します。

## FILECASE

UNIX および WebFOCUS では、大文字、小文字、大文字小文字混在のファイル名およびディレクトリがサポートされます。UNIX 環境のデフォルト設定が小文字であるため、WebFOCUS が自動的にファイル名およびディレクトリを小文字に変換します。

構文は次のとおりです。

```
SET FILECASE = {ACTUAL|LOWER}
```

説明

### ACTUAL

大文字、小文字、大文字と小文字の混在のいずれかで入力したファイル名をそのまま保持します。この設定を使用すると、WebFOCUS は入力した通りの名前でファイル名を検索します (大文字、小文字、大文字と小文字の混在)。

### LOWER

大文字または大文字と小文字の混在の名前を小文字に変換します。この設定を使用すると、WebFOCUS は小文字でファイル名を検索します。

## FILECOMPRESS

SET FILECOMPRESS=ON コマンドは、PDF 出力ファイルを圧縮します。

構文は次のとおりです。

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

説明

### ON

PDF 出力ファイルを圧縮します。

### OFF

PDF 出力ファイルを圧縮しません。デフォルト値は OFF です。

このコマンドは、PDF 出力にのみ適用されます。HTML や Excel などの他の出力フォーマットでは無視されます。

## FILE[NAME]

FILE[NAME] パラメータは、デフォルト設定でコマンドに使用するファイル名を指定します。デフォルトのファイル名を設定しておく、そのファイルを使用する際に名前の指定を省略することができます。

構文は次のとおりです。

```
SET FILE[NAME] = filename
```

説明

`filename`

コマンドで使用するデフォルトのファイルです。

## FILTER

FILTER パラメータは、レポートの作成に使用する選別条件をデータソースに割り当てます。このパラメータを使用し、フィルタを有効または無効にします。

SET FILTER コマンドの使用は 1 行に制限されます。複数のフィルタを 1 行内で有効にするには、SET FILTER コマンドを繰り返し発行します。

構文は次のとおりです。

```
SET FILTER= {*_|xx[yy zz]} IN {file|*} {ON|OFF}
```

説明

\*

宣言されたすべてのフィルタを示します。デフォルト値は \* です。

`xx, yy, zz`

FILTER FILE コマンドの NAME = 構文で宣言するフィルタの名前です。

`file`

選択条件を割り当てるデータソースの名前です。

**ON**

特定のデータソースまたはすべてのデータソース (\*) を対象に、すべてのフィルタ (\*) または指定したフィルタを有効にします。1 つのデータソースに対して有効にできるフィルタの最大数は、単一レポートリクエストの WHERE/IF 条件の制限を超えない範囲で、WHERE/IF 句の数に限定されます。

**OFF**

特定のデータソースまたはすべてのデータソース (\*) を対象に、すべてのフィルタ (\*) または指定したフィルタを無効にします。デフォルト値は OFF です。

## FIXRET[RIEVE]

FIXRETRIEVE パラメータを使用すると、HOLD ファイルなどの固定フォーマットのシーケンシャルファイルからのキー検索が可能になります。つまり、キーについての等価テストまたは範囲テストが True の場合に検索処理を停止します。

構文は次のとおりです。

```
SET FIXRET[RIEVE] = {OFF|ON}
```

説明

**ON**

キー検索を有効にします。デフォルト値は ON です。

**OFF**

キー検索を無効にします。

## FLOATMAPPING

SET FLOATMAPPING を使用すると、DB2 および Oracle で使用可能なパック 10 進数ベースの桁数を活用し、この機能を浮動小数点数のすべての数値処理にまで拡張することができます。この処理により、精度の向上と (端数処理を含む)、パフォーマンスの向上の両方が期待できます。

構文は次のとおりです。

```
SET FLOATMAPPING = {D|M|X}
```

説明

**D**

標準の倍精度処理を使用します。これがデフォルト値です。

**M**

新しい内部フォーマットを使用します。このフォーマットでは、倍精度小数点数の 10 進数精度が最大 16 桁まで提供されます。

**X**

新しい内部フォーマットを使用します。このフォーマットでは、倍精度小数点数の 10 進数精度が最大 34 桁まで提供されます。

**注意：**フィールドが HOLD ファイルに転送された場合、内部データタイプの X または M は、HOLD マスターファイルの USAGE および ACTUAL フォーマットに継承されます。

### FOC144

FOC144 パラメータは、次の FOC144 警告メッセージを非表示にします。

警告。無関連のオカレンスのテストです。

構文は次のとおりです。

```
SET FOC144 = {NEW|OLD}
```

説明

**NEW**

FOC144 警告メッセージを表示します。デフォルト値は NEW です。

**OLD**

FOC144 警告メッセージを表示しません。

### FOCEXURL

FOCEXURL パラメータは、ドリルダウンをリモートで実行します。ドリルダウンプログラムは、ローカルマシンまたはリモートマシンのどちらにでも常駐させることができます。

構文は次のとおりです。

```
SET FOCEXURL = path
```

説明

**path**

WebFOCUS Servlet のパスです。Servlet のデフォルトパスは /ibi\_apps/WFServlet です。

### FOCFIRSTPAGE

FOCFIRSTPAGE パラメータは、出力の先頭ページにページ番号を割り当てます。

構文は次のとおりです。

```
SET FOCFIRSTPAGE = {n|1|&FOCNEXTPAGE}
```

説明

**n**

出力の先頭ページに割り当てる番号です。有効値は 1 から 6 バイトの整数です。デフォルト値は 1 です。

**&FOCNEXTPAGE**

最後のレポートで使用する最終ページ番号で値が決定する変数です。この値は、最後のレポートで使用する最終ページ番号より 1 つだけ大きい数字になります。

**FOCHTMLURL**

FOCHTMLURL パラメータを使用すると、/ibi\_apps/ibi\_html 以外のエイリアスを使用してリソースにアクセスすることができます。

WebFOCUS Reporting Server で /ibi\_apps/ibi\_html 以外のエイリアスを生成するには、SET FOCHTMLURL コマンドを使用して、/ibi\_apps/ibi\_html の代わりに生成するエイリアスを設定します。

たいていの場合、このコマンドはサーバプロファイル (EDASPROF.PRF) で使用するか、インストールのデフォルト値を設定するために WFS ファイル (site.wfs) で使用します。

構文は次のとおりです。

```
SET FOCHTMLURL=/my_html
```

説明

**my\_html**

WebFOCUS Reporting Server で生成される /ibi\_apps/ibi\_html 以外のエイリアスです。  
260 バイト以内で指定することができます。

**FOCSTACK**

この設定は不要になりましたが、この設定が使用されている既存のアプリケーションが継続して動作するように製品に付属されています。FOCSTACK パラメータは、実行待ちの FOCUS コマンドのスタックである FOCSTACK が使用するメモリ容量を 1000 バイト単位で指定します。

このコマンドは、ON TABLE SET で使用することはできません。

構文は次のとおりです。

```
SET FOCSTACK [SIZE] = {n|8}
```

説明

**n**

FOCSTACK が使用できる最大容量を 1000 バイト単位で指定します。最大値は領域のサイズにより異なります。

### 8

FOCSTACK は 8000 バイトを使用することができます。デフォルト値は 8 です。

## FORMULTIPLE

次の構文を FML リクエストの前または中に記述することにより、FOR フィールドのある値を複数行に使用することができます。この値は、値のみを指定、範囲の一部として使用、または計算に含めることができます。

構文は次のとおりです。

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

説明

### ON

FML リクエストの複数行で FOR フィールドの特定の値を参照することができます。

FORMULTIPLE を ON に設定すると、データソースから抽出した値は、タグが一致するレポート出力の各行に挿入されます。

### OFF

1 つの値を複数行に指定することはできません。デフォルト値は OFF です。

FORMULTIPLE を OFF に設定すると、複数のタグは、その参照方法 (OR 句、TO 句、マスク文字) に関わらず、まずは完全に一致するタグまたは範囲の終了位置に対して評価され、次にマスク文字、最後に範囲内の値に対して評価されます。たとえば、値がまず完全一致として、次に範囲の一部として指定されている場合は、完全一致が表示されます。優先順位が同じ (例、完全一致と範囲の終了位置) であるタグを含む行が複数あり、これらの行に値が一致する場合の表示結果は予想できません。

## GRAPH-PPI

GRAPH-PPI パラメータは、サーバサイド HTML5 グラフ出力 (XLSX、PPTX、PDF、DHTML、PNG フォーマット) に適用する解像度を指定します。この場合、GRAPHENGINE=JSCHART 設定が必要です。構文は次のとおりです。

```
SET GRAPH-PPI = n
```

説明

### n

出力の解像度をインチ当たりのピクセル数で指定する 72 から 1200 までの整数です。デフォルト値は 192 です。数字が大きいくほど出力ファイルのサイズが大きくなります。

## GRAPHDEFAULT

GRAPHDEFAULT パラメータは、WebFOCUS でグラフを描画する際に、WebFOCUS のデフォルトグラフスタイル (例、デフォルトの色やデータラベル配置) を使用するか、グラフ API のデフォルトグラフスタイルを使用するかを制御します。

構文は次のとおりです。

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

説明

[ON](#)

WebFOCUS のデフォルトグラフスタイルを使用します。デフォルト値は ON です。

[OFF](#)

グラフ API のデフォルトスタイルを使用します。

## GRAPHEDIT

WebFOCUS バージョン 8.0 以降では、このパラメータは廃止されました。グラフの編集には、InfoAssist を使用することができます。

構文は次のとおりです。

```
SET GRAPHEDIT = SERVER
```

説明

[SERVER](#)

サーバサイドグラフを生成します。Web サーバで GIF グラフが作成され、そのイメージがクライアントのブラウザに返されます。デフォルト値は SERVER です。

## GRAPHENGINE

GRAPHENGINE パラメータは、使用する Java ベース WebFOCUS グラフエンジンのバージョンを指定します。構文は次のとおりです。

```
SET GRAPHENGINE = {GRAPH53|NEW|JSCHART}
```

説明

[GRAPH53](#) | [NEW](#)

バージョン 5.3 以降の WebFOCUS グラフエンジンです。デフォルト値は GRAPH53 です。NEW は GRAPH53 と同義です。

### JSCHART

LOOKGRAPH 値が PPTX、XLSX、PDF、DHTML、PNG のいずれかの場合に、サーバサイド HTML5 グラフ出力を生成します。この設定は、グラフリクエストにグラフ属性構文が使用されている場合は、デフォルト設定で有効になります。

### GRAPHSEVURL

GRAPHSEVURL パラメータは、HTTP リクエストを Servlet に送信し、グラフのイメージファイルを作成します。イメージファイルは WebFOCUS Reporting Server に保存され、そこで PDF レポートに埋め込むイメージの入力ファイルとして使用されます。この処理では、WebFOCUS Reporting Server で Java を実行する必要はなく、Servlet を実行するリモートマシン上でのみ必要になります。この方法は、TCP/IP をサポートするすべてのプラットフォームで 사용할 ことができます。

構文は次のとおりです。

**SET GRAPHSEVURL = url**

説明

**url**

GIF ファイルを作成するグラフ Servlet の URL です (最大 256 バイト)。

以下はその例です。

**[http://hostname/ibi\\_apps/IBIGraphServlet](http://hostname/ibi_apps/IBIGraphServlet)**

説明

**hostname**

サーバが常駐するホストの名前です。

### GRID

GRID パラメータは、グラフ上の横および縦の目盛りの位置に罫線を描きます。

構文は次のとおりです。

**SET GRID = {ON|OFF}**

説明

**ON**

グラフ上の横および縦の目盛りの位置に罫線を描きます。

**OFF**

グラフ上の横および縦の目盛りの位置に罫線を描きません。デフォルト値は OFF です。

**GRMERGE**

GRMERGE パラメータは、リクエストに複数のソートフィールドが含まれている場合に、すべての出力データを反映した 3 次元グラフを作成するか、すべての出力データを個別に反映した複数のグラフを作成するかを制御します。GRMERGE が OFF の場合、BY および ACROSS の両方の句を含むリクエストは 2 つのグラフを生成します。GRMERGE が ON の場合、グラフは 1 つに集約されます。

構文は次のとおりです。

**SET GRMERGE = {ON|OFF|ADVANCED}**

**説明****ON**

すべての出力データを反映した 1 つの 3 次元グラフを生成します。デフォルト値は ON です。

**OFF**

すべての出力データを反映した複数の個別グラフを生成します。

**ADVANCED**

高度な統合オプションを有効にします。このオプションを選択すると、次の 3 つのパラメータに基づいて複数グラフの統合方法が決定されます。

- ☐ **GRMULTIGRAPH** 複数グラフの作成に使用するソートフィールド数を指定します。
- ☐ **GRLEGEND** グラフ凡例に配置するソートフィールド数を指定します。
- ☐ **GRXAXIS** X 軸に表示するソートフィールド数を指定します。グラフを描画するには、GRXAXIS の値を少なくとも 1 に設定する必要があります。1 より大きい値を指定すると、ネストされた X 軸が作成されます。

**注意：**GRMULTIGRAPH、GRLEGEND、GRXAXIS で使用するソートフィールドの総数は、グラフリクエストで使用するソートフィールド数に一致させる必要があります。

**GRMULTIGRAPH**

GRMULTIGRAPH パラメータは、GRMERGE を ADVANCED に設定した場合に、複数グラフの作成に使用するソートフィールド数を指定します。

構文は次のとおりです。

**SET GRMULTIGRAPH = n**

説明

**n**

出力を複数のグラフに分割する際に使用するソートフィールド数 (0 から 2) を指定します。グラフの分割には、最も外側のソートフィールドが使用されます。n が 0 (ゼロ) より大きい場合、**GRMERGE=OFF** と同様に機能しますが、この設定ではソートフィールドを追加することが可能です。

## GRWIDTH

**GRWIDTH** パラメータは、OLAP グラフおよび標準のレポートリクエストに適用されます。ただし、このパラメータを OLAP コントロールパネルで制御することはできません。

構文は次のとおりです。

**SET GRWIDTH = {n|0}**

説明

**n**

グラフを表示する HTML テーブルの幅です。セル単位で指定します。有効値は 0 (ゼロ) から 512 です。このパラメータを 0 に設定すると、Java アプレットが生成され、1 ページに複数のグラフが上下に並んで表示されます。デフォルト値は 0 (ゼロ) です。

## GTREND

**GTREND** パラメータは、基本線形回帰の使用を有効にし、**SCATTER** グラフで X 軸および Y 軸の値を変更します。基本線形回帰は、X 軸および Y 軸の値の総和の平均を使用し、散布図の傾向を表す一次方程式を求めます。

構文は次のとおりです。

**SET GTREND={ON|OFF}**

説明

**ON**

基本線形回帰を使用し、X 軸および Y 軸の値を変更します。

**OFF**

X 軸および Y 軸の値を変更しません。デフォルト値は **OFF** です。

## HAUTO

HAUTO パラメータは、表示する値に合わせて横軸のスケールを自動的に変更します。HAUTO が OFF の場合、横軸のスケールを変更するには、HMAX および HMIN パラメータを使用する必要があります。詳細は、592 ページの「[HMAX](#)」および 592 ページの「[HMIN](#)」の説明を参照してください。

構文は次のとおりです。

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

説明

[ON](#)

横軸のスケールを自動的に変更します。デフォルト値は ON です。

[OFF](#)

横軸のスケールを自動的に変更しません。この場合、エンドユーザが HMAX および HMIN の値を指定する必要があります。

## HAXIS

HAXIS パラメータは、横軸の幅をバイト数で指定します。このパラメータは、オフラインで生成したグラフに適用されます。オンライン表示では、WebFOCUS がページ幅に合わせてグラフの幅を自動調整するため、HAXIS の設定は無視されます。

構文は次のとおりです。

```
SET HAXIS = {n|770}
```

説明

[n](#)

横軸の幅です。バイト数で指定します。許容値は 20 から 770 までの整数です。デフォルト値は 770 です。

## HCLASS

HCLASS パラメータは、AUTOTICK が OFF の場合に横軸の目盛り間隔を指定します。関連情報は、596 ページの「[HTICK](#)」を参照してください。

構文は次のとおりです。

```
SET HCLASS = {n|_00}
```

説明

**n**

横軸の目盛り間隔の値です。デフォルト値は .00 です。

### HDAY

HDAY パラメータは、休日として指定した日付を格納する休日ファイルを指定します。このファイルから取得した日付は、日付関数の DATEDIF、DATEMOV、DATECVT、DATEADD で使用されます。このファイル名は、先頭の「HDAY」という語句とそれに続く任意の 2 から 4 文字で構成します。

休日ファイルをクリアするには、次のように指定します。

**SET HDAY = OFF**

構文は次のとおりです。

**SET HDAY = xxxx**

説明

**xxxx**

休日ファイル名の HDAYxxxx に含める文字列です。この文字列は、2 から 4 バイトで指定する必要があります。

このパラメータには、デフォルト設定はありません。

### HIDENULLACRS

HIDENULLACRS パラメータは、Null 列のみを含む ACROSS グループを非表示にします。

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

構文は次のとおりです。

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

説明

**ON**

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

### OFF

列を非表示にしません。デフォルト値は OFF です。

## HISTOGRAM

HISTOGRAM パラメータは、横軸の値が数値以外の場合に、曲線の代わりにヒストグラムを描きます。

構文は次のとおりです。

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

説明

### ON

横軸の値が数値以外の場合は、曲線ではなくヒストグラムを描きます。デフォルト値は ON です。

### OFF

横軸の値が数値以外の場合、曲線を描きます。

## HLDCOM\_TRIMANV

HLDCOM\_TRIMANV パラメータは、区切り文字付き出力ファイルの AnV フィールドに末尾のブランクを保持するかどうかを制御します。

構文は次のとおりです。

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

説明

### OFF

区切りフォーマットで出力された場合に、AnV フィールドの末尾のブランクを保持します。デフォルト値は OFF です。

### ON

区切りフォーマットで出力された場合に、AnV フィールドの末尾のブランクを削除します。

### HMAX

HMAX パラメータは、自動スケール変更機能を使用しない場合に (HAUTO=OFF)、横軸に表示する最大値を設定します。

構文は次のとおりです。

```
SET HMAX = {nnn|.00}
```

説明

**nnn**

HAUTO=OFF の場合に横軸に表示する最大値です。デフォルト値は .00 です。

### HMIN

HMIN パラメータは、自動スケール変更機能を使用しない場合に (HAUTO=OFF)、横軸に表示する最小値を設定します。

構文は次のとおりです。

```
SET HMIN = {nnn|.00}
```

説明

**nnn**

HAUTO=OFF の場合に横軸に表示する最小値です。デフォルト値は .00 です。

### HNODATA

HNODATA パラメータは、ミッシングデータの文字列を制御します。MISSING=ON 属性が指定されたフィールドのミッシング値は、ここで指定した文字列として HOLD FORMAT ALPHA ファイルに継承されます。MISSING=ON 属性が指定されていないフィールドのミッシング値は、ブランク (文字フィールド) またはゼロ (数値フィールド) として HOLD ファイルに継承されます。

構文は次のとおりです。

```
SET HNODATA = {charstring|,$}
```

説明

**charstring**

MISSING=ON 属性が指定されたフィールドのミッシング値を、HOLD FORMAT ALPHA ファイルに継承する際の文字列です。最大長は 12 バイトです。ピリオド (.) がデフォルト値は です。

指定した文字列がフィールド長を超える場合は、次の値が格納されます。

□ 文字フィールドの場合は、文字列の左端の文字。

□ 数値フィールドの場合は、ブランク。

文字フィールドのミッシング値としてデフォルト値 (ピリオド) 以外の文字を使用すると、データを表示する際にフォーマットエラーの発生を回避するため、そのフィールドにはブランクが挿入されます。デフォルトの HNODEATA 値を使用する場合、数値フィールドにはデフォルト値が挿入されます。このようにすると、HOLD ファイルに対するリクエストで、HOLD ファイルに継承されたミッシングデータを識別することができます。

小数点以下の桁数を含む値を HNODEATA に指定し、ミッシングデータが格納されたフィールドのフォーマットが整数の場合、その値は整数に端数処理された上で挿入されます。小数点以下の桁数をサポートする数値フィールドでは、値は小数点以下の正しい桁数に端数処理された上で挿入されます。

, \$

ミッシングデータが存在する際に、フィールドに何も挿入しないことを示します。

## HOLDATTR

HOLDATTR パラメータは、元のマスターファイルのどの属性を HOLD マスターファイルで使用するかを制御します。この設定は、HOLD マスターファイルのフィールド名の名付け方法には影響しません。

**注意：** HOLDATTRS は HOLDATTR と同義です。

構文は次のとおりです。

**SET HOLDATTR = {ON|OFF|FOCUS|CUBE}**

説明

### ON

HOLD ファイルのフォーマットに関係なく、元のマスターファイルの TITLE 属性を HOLD マスターファイルに追加します。この場合、PROPERTY 属性も継承されます。HOLD ファイルが FOCUS フォーマットの場合は、HOLD マスターファイルに ACCEPT 属性を追加します。

### OFF

元のマスターファイルの TITLE 属性および ACCEPT 属性を HOLD マスターファイルに追加しません。

### FOCUS

HOLD ファイルが FOCUS フォーマットの場合は、TITLE 属性および ACCEPT 属性を HOLD マスターファイルに追加します。この場合、PROPERTY 属性も継承されます。デフォルト値は FOCUS です。

### CUBE

フォルダおよび DV\_ROLE 属性を継承します。また、TITLE 属性が HOLD マスターファイルに継承されます。フィールド名は、エイリアス値として継承されます。

## HOLDFORMAT

HOLDFORMAT パラメータは、HOLD ファイルに使用するデフォルトフォーマットを設定します。この値は、リクエストで ON TABLE SET HOLD FORMAT コマンドを発行することにより、それぞれの HOLD ファイルで上書きすることができます。

構文は次のとおりです。

```
SET HOLDFORMAT = {BINARY|ALPHA}
```

説明

### BINARY

HOLD ファイルをバイナリフォーマットで作成します。デフォルト値は BINARY です。

### ALPHA

HOLD ファイルを ALPHA フォーマットで作成します。

## HOLDLIST

HOLDLIST パラメータは、HOLD ファイルまたは PCHOLD ファイルに格納するフィールドを表示フィールドのみに限定するか、すべてのフィールドを対象にするかを制御します。

構文は次のとおりです。

```
SET HOLDLIST = {PRINTONLY|ALL|ALLKEYS|EXPLICIT}
```

説明

### PRINTONLY

レポートリクエストで指定したフィールドのみを HOLD ファイルまたは PCHOLD ファイルに格納します。

### ALL

レポートリクエストで参照されるすべてのフィールドを HOLD ファイルまたは PCHOLD ファイルに格納します。この場合、一時項目 (COMPUTE) および COMPUTE コマンドで参照するフィールドも含まれます。デフォルト値は ALL です。なお、ALL の同義語として OLD を使用することもできます。

**注意:** リクエストで指定された BY フィールドに NOPRINT オプションが使用されている場合、その BY フィールドは SET HOLDLIST=ALL に設定した場合でも HOLD ファイルに格納されません。

### ALLKEYS

すべてのフィールドを HOLD または PCHOLD ファイルに格納します (NOPRINT オプションが指定された BY フィールドも含む)。

### EXPLICIT

リクエストで NOPRINT オプションを使用してレポート出力から明示的に除外したフィールドを HOLD または PCHOLD ファイルに格納します。ただし、NOPRINT が暗示的に指定されているフィールドは格納されません。たとえば、リクエストでフィールドのフォーマットを再設定した場合、新しいフォーマットのフィールドと、元のフォーマットのフィールド (NOPRINT が暗示的に指定された) の 2 つのフィールドが存在します。

## HOLDMISS

HOLDMISS パラメータを使用すると、デフォルト値のブランク (文字データ) またはゼロ (数値データ) とミッシングデータの両者を HOLD ファイル内で区別することができます。

構文は次のとおりです。

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

説明

### OFF

ミッシングデータを HOLD ファイルに格納することはできません。デフォルト値は OFF です。

### ON

ミッシングデータを HOLD ファイルに格納することができます。TABLE は、データが見つからない場合にミッシングデータのデフォルト値を生成します。

### HOLDSTAT

HOLDSTAT パラメータは、コメントおよび DBA 情報を HOLD および PCHOLD マスターファイルに追加します。この情報は、HOLDSTAT ERRORS ファイルから取得することも、ユーザ指定のファイルから取得することもできます。

構文は次のとおりです。

```
SET HOLDSTAT = {ON|OFF|name}
```

説明

#### ON

コメントおよび DBA 情報を取得します。取得元は、UNIX および Windows では holdstat.mas または errors.mas ファイルです。

#### OFF

HOLDSTAT ファイルの情報を HOLD または PCHOLD マスターファイルに追加しません。デフォルト値は OFF です。

#### name

エンドユーザが作成した HOLDSTAT ファイルを指定します。この情報は、HOLD または PCHOLD マスターファイルに追加されます。

### HSTACK

HSTACK パラメータは、ヒストグラムで棒グラフを横に並べずに積み重ねます。

構文は次のとおりです。

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

説明

#### ON

ヒストグラムで棒グラフを積み重ねます。

#### OFF

ヒストグラムで棒グラフを横に並べます。デフォルト値は OFF です。

### HTICK

HTICK パラメータは、AUTOTICK が OFF の場合に、横の目盛り間隔を設定します。詳細は、589 ページの「[HCLASS](#)」を参照してください。

構文は次のとおりです。

```
SET HTICK = {nnn|.00}
```

説明

**nnn**

AUTOTICK=OFF の場合の横軸の目盛り間隔です。デフォルト値は .00 です。

## HTMLARCHIVE

HTMLARCHIVE パラメータにより、HTML レポートまたは DHTML レポートは、イメージファイルとともに単一の Web アーカイブドキュメント (.mht ファイル) にパッケージ化されます。

HTML のこのフォーマットは、Internet Explorer のみでサポートされています。

構文は次のとおりです。

**SET HTMLARCHIVE = {ON|OFF}**

説明

**ON**

HTML レポートまたは DHTML レポートを、イメージファイルとともに単一 Web アーカイブドキュメント (.mht ファイル) にパッケージ化します。

**OFF**

複数ファイルの単一ドキュメントへのパッケージ化は行われません。デフォルト値は OFF です。

## HTMLCSS

HTMLCSS パラメータは、HTML 表示ページ内に内部カスケードスタイルシートコマンドを作成します。

構文は次のとおりです。

**SET HTMLCSS = {ON|OFF}**

説明

**ON**

レポート出力を表示する HTML ページに内部スタイルシートコマンドを作成します。

**OFF**

レポート出力を表示する HTML ページに内部スタイルシートコマンドを作成しません。デフォルト値は OFF です。

### HTMLEMBEDIMG

HTMLEMBEDIMG パラメータを使用すると、HTML または DHTML .htm ファイルにイメージおよびグラフを直接埋め込むエンコードメカニズムが有効になります。これにより、任意のブラウザからすべての WebFOCUS レポートへのアクセスが可能になります。

構文は次のとおりです。

```
SET HTMLEMBEDIMG = {OFF|ON|AUTO}
```

説明

#### OFF

デフォルト動作に影響しません。HTMLARCHIVE が ON に設定されている場合、.mht ファイルが生成されます。

#### ON

.htm ファイル内でイメージをエンコードします。

#### AUTO

イメージの処理方法は、呼び出し元クライアントのブラウザに基づいて決定されます。Internet Explorer を使用している場合、HTMLARCHIVE が使用され、イメージが .mht ファイルに埋め込まれます。その他のすべてのブラウザでは、HTMLEMBEDIMG が使用され、イメージ情報がエンコードされて .htm ファイルに埋め込まれます。表示ブラウザが不明な場合、ReportCaster から生成されたレポートと同様に、現在有効な HTMLARCHIVE 設定が使用されます。

### HTMLENCODE

HTMLENCODE パラメータは、HTML タグが実データ内に格納されている場合や DEFINE または COMPUTE コマンドを使用して作成されている場合に、これらのタグをエンコードするかどうかを制御します。

WebFOCUS レポートで HTMLENCODE=ON に設定すると、文字列内で設定されたテキストが送信の際に暗号化され、レポートに表示される際に復号化されます。これにより、セキュリティが確保されるとともに、特殊文字が正しく表示されるようになります。

構文は次のとおりです。

```
SET HTMLENCODE {ON|OFF}
```

## 説明

**ON**

HTML 出力のデータをエンコードします。この設定は、HTML タグが実データ内に格納されている場合や DEFINE または COMPUTE コマンドを使用して作成されている場合に、これらのタグの表示を無効にします。

**OFF**

HTML エンコードを無効にします。デフォルト値は OFF です。

**注意：**新しい XLSX ファイルフォーマットが ZIP 圧縮形式のため、キャレット (<) などのネイティブ HTML 記号はタグ文字としてサポートされません。XLSX では、他の出力フォーマットと異なり、HTMLENCODE のデフォルト値が ON に設定されています。HTMLENCODE を OFF に設定すると、HTML タグ文字が含まれたデータがセルから除外されます。

**INDEX**

INDEX パラメータは、インデックスに使用するインデックススキーマを指定します。インデックスとは、マスターファイルで FIELDTYPE=I キーワードが指定されたフィールドです。このパラメータは、以前のバージョンの WebFOCUS を使用して FOCUS データソースに作成したインデックスに適用されます。

構文は次のとおりです。

```
SET INDEX[TYPE] = {NEW|OLD}
```

## 説明

**NEW**

バイナリツリーインデックスを作成します。デフォルト値は NEW です。

**OLD**

ハッシュインデックスを作成します。

## JOIN\_LENGTH\_MODE (JOINLM)

JOIN\_LENGTH\_MODE (JOINLM) パラメータは、レコード指向型のアダプタ (例、VSAM、DFIX、FIX) に対して、結合されたフィールドの組み合わせの等価性処理を制御します。互換性はあるが同一ではない JOIN フィールドの処理では、次の 2 つのモードがサポートされます。

- ❑ **SQL 互換モード** JOIN コマンドプロセッサは、結合されたフィールドの値の等価性を厳密に保証します。ホストからクロスリファレンスへの変換時に、有意な文字列の末尾の切り捨てが検出された場合は、「ファイルが見つかりません」というエラーが発生します。ホストフィールドの値が短い場合は、データタイプに従って有意でない文字列を含むクロスリファレンスフィールドの長さに拡張されます。
- ❑ **FOCUS レポートモード** JOIN コマンドプロセッサは、結合されたフィールドの値の等価性を部分的に保証します。
  - ❑ 短いフィールドを長いフィールドに結合する場合は、検索範囲が作成され、ホストの値を前に持つすべてのクロスリファレンスの値を検索可能にします (部分キー結合)。
  - ❑ 長いフィールドを短いフィールドに結合する場合は、クロスリファレンスフィールドの長さに合わせて、ホストの値の末尾が無条件に切り捨てられます。

構文は次のとおりです。

**SET JOIN\_LENGTH\_MODE = {[SQL](#)|[RANGE](#)}**

### 説明

#### [SQL](#)

SQL 互換モードを設定します。ホストとクロスリファレンスのフィールド間の厳密な等価性を保証します。これがデフォルト値です。

#### [RANGE](#)

FOCUS レポートモードを設定します。部分キー JOIN をサポートします。

## JOINOPT

JOINOPT パラメータには、2 つの機能があります。

- ❑ **ユニーク JOIN でのラギング値の補正** 親セグメントに 2 つ以上のユニーク子セグメントが存在し、それぞれが複数の子を持つ場合、レポートでのミッシング値の表示が不正確になる可能性があります。その結果、それ以外の子の値が、レポートに正しく整列されなくなります。これらの不整列値は「ラギング値」と呼ばれます。JOINOPT パラメータは、ラギング値を補正することで、出力を正しく整列します。

- **データタイプ変換ありの JOIN の有効化** 異なる数値データタイプを含む複数のデータソースを結合することができます。たとえば、短いパック 10 進数フォーマットのフィールドを長いパック 10 進数フォーマットのフィールドに結合したり、整数フォーマットのフィールドをパック 10 進数フォーマットのフィールドに結合したりすることができます。これにより、JOIN 構造のデータソースからレポートを柔軟に作成することが可能になります。

構文は次のとおりです。

```
SET JOINOPT = {NEW|GNTINT|OLD}
```

説明

**NEW**

親セグメントに複数のユニーク子セグメントが存在する場合に、ラギング値を補正します。また、データタイプ変換ありの JOIN を有効にします。

**GNTINT**

親セグメントに複数のユニーク子セグメントが存在する場合に、ラギング値を補正します。また、データタイプ変換ありの JOIN を有効にします。

**OLD**

ラギング値は補正されず、データタイプ変換ありの JOIN もサポートされません。これがデフォルト値です。

## JPEGENCODE

SET JPEGENCODE コマンドを使用して、圧縮方法を選択することができます。

構文は次のとおりです。

```
SET JPEGENCODE=[FLATE|DCT]
```

説明

**FLATE**

定義済み固定画質

WebFOCUS バージョン 7.7 SP03 で導入された圧縮方法です。この方法では、PDF ファイル内のイメージを個別に圧縮することができます。この方法は、WebFOCUS バージョン 7.7 SP03 からバージョン 7.7 SP06 までのデフォルトです。

**DCT**

離散コサイン変換

WebFOCUS バージョン 8.2 SP01 で導入された圧縮方法です。この方法では、保持する画質をパーセントで指定することができます。この方法は、WebFOCUS バージョン 8.2 SP01 以降でのデフォルト設定です。

**注意：**イメージによっては、高い圧縮率を適用した場合でも、画質の低下が顕著に現れない場合があります。通常、高い圧縮率を適用すると、イメージの画質が低下します。画質の値を設定するには、DCT 設定とともに、SET JPEGQUALITY を使用します。

### JPEGQUALITY

SET JPEGQUALITY コマンドを使用して、画質を選択することができます。

構文は次のとおりです。

```
SET JPEGQUALITY=n
```

説明

*n*

DCT 設定で使用する画質のパーセントです。1 から 100 までの値を指定できます。デフォルト値は 100 です。この場合、画質は低下せず、最小の圧縮が適用されます。

### JSURLS

JSURLS パラメータは、HTML 出力を生成する TABLE リクエストまたは GRAPH リクエストに JavaScript または VBScript ファイルを追加します。これにより、任意の JavaScript または VBScript 関数を使用して WebFOCUS HTML レポートまたはグラフの表示をカスタマイズすることができます。JavaScript および VBScript ファイルは最後にロードされ、また、リストに記述された順にロードされるため、HTML ページのカスタマイズに制約を与えることはありません。

また、WebFOCUS レポートまたはグラフの実行時に、定義済みの JavaScript 関数セットが呼び出されます。これらの関数には、すべての WebFOCUS レポートに適用される関数もあれば、OLAP、Web サービス、目次の各レポートにのみ適用される関数もあります。JSURLS を使用し、これらのデフォルト関数を無効にしたり、編集したりすることができます。定義済み JavaScript 関数の一覧を確認するには、/ibi/WebFOCUSxx/webapps/WEB-INF/jsp/global/ibigl.js を参照してください。

構文は次のとおりです。

```
SET JSURLS='/file1 [/file2] [/file3]...'
```

### 説明

`/file1 [/file2] [/file3]...`

JavaScript または VBScript が格納されたファイルです。複数の js ファイルを指定する場合は、それぞれをブランクで区切り、値を一重引用符 (') で囲む必要があります。これらのファイルは、Web サーバがアクセス可能な場所に保存しておく必要があります。

これらのファイルは URL で参照することができます。

## KEEPDEFINES

KEEPDEFINES パラメータは、作成した一時項目 (DEFINE) をホストまたは JOIN 構造に使用する場合、JOIN コマンドの実行後にその一時項目を保持するかどうかを制御します。このパラメータは、DEFINE コマンドを JOIN コマンドより前に実行する場合に適用されます。

構文は次のとおりです。

`SET KEEPDEFINES = {ON|OFF|ALL}`

### 説明

#### ON

JOIN コマンドの実行後に一時項目 (DEFINE) を保持します。

#### OFF

JOIN コマンドの実行後に一時項目 (DEFINE) をクリアします。デフォルト値は OFF です。

#### ALL

JOIN コマンドの実行後、リクエスト内でそれ以前および以降の一時項目 (DEFINE) のすべてを保持します。

## KEEPFILTERS

デフォルト設定では、ホストデータソースで定義したフィルタは、JOIN コマンドの実行後にクリアされます。ただし、SET KEEPFILTERS=ON コマンドを発行することにより、JOIN コマンドを発行した後もフィルタを保持することができます。

KEEPFILTERS を ON に設定すると、JOIN コマンドの実行後に、フィルタ定義およびその定義を宣言したときのステータスが元の状態に戻ります。JOIN を発行する前に定義した一連のフィルタおよび一時項目 (DEFINE) は、「コンテキスト」と呼ばれます。なお、一時項目に関連するコンテキストについての詳細は、SET KEEPDEFINES および DEFINE FILE SAVE についての説明を参照してください。JOIN または DEFINE FILE コマンドを発行するたびに、新しいコンテキストが作成されます。

JOIN コマンドを発行した後で新しいフィルタを定義する場合、FILTER FILE コマンドの CLEAR オプションを発行しない限り、以前に定義したフィルタと同一名のフィルタを作成することはできません。CLEAR オプションを使用すると、すべてのコンテキストから指定したマスターファイルのフィルタ定義がすべてクリアされます。

JOIN をクリアすると、その JOIN コマンドを発行する前に有効であったフィルタ定義およびクリアされなかったフィルタ定義がすべて元のステータスに戻ります。JOIN CLEAR join\_name コマンドを発行して JOIN をクリアすると、JOIN join\_name コマンドを発行した後で作成したコンテキストおよびフィルタ定義がすべて削除されます。

構文は次のとおりです。

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

説明

OFF

JOIN の前に発行されたフィルタは保持されません。これがデフォルト値です。

ON

JOIN 間でのフィルタは保持されます。

## LANG[UAGE]

LANG[UAGE] パラメータは、国際言語サポート (NLS) の環境を指定します。このパラメータは、サーバのエラーメッセージに使用する言語を設定します。また、マスターファイルに別の言語の TITLE 属性が含まれている場合に、レポートタイトルに使用する言語をこのパラメータで設定することもできます。

構文は次のとおりです。

```
SET LANG[UAGE] = [LNG|ln]
```

説明

LNG

言語の指定に使用する 3 文字の省略形です。

ln

言語の指定に使用する 2 文字の ISO コードです。

下表は、言語の指定に使用する省略形および ISO コードの一覧です。

言語名 (コード)	表示言語 (GUI)	言語の省略形	言語の ISO コード
AMENGLISH or ENGLISH or UKENGLISH	English	AME or ENG or UKE	en
ARABIC	Arabic	ARB	ar
BALTIC	Lithuanian	BAL	lt
CZECH	Czech	CZE	cs
DANISH	Danish	DAN	da
DUTCH	Dutch	DUT	nl
FINNISH	Finnish	FIN	fi
FRENCH	French - Standard or Canadian	FRE	fr fc
GERMAN	German - Standard or Austrian	GER	de at
GREEK	Greek	GRE	el
HEBREW	Hebrew	HEB or HEW	iw
ITALIAN	Italian	ITA	it
JAPANESE	Japanese-JIS or EUC	JPN or JPE	ja or je
KOREAN	Korean	KOR	ko
POLISH	Polish	POL	po
PORTUGUESE	Portuguese- Brazil or Portugal	POR	br pt

言語名 (コード)	表示言語 (GUI)	言語の省略形	言語の ISO コード
RUSSIAN	Russian	RUS	ru
S-CHINESE	Chinese-Simplified GB	PRC	zh
SPANISH	Spanish	SPA	es
SWEDISH	Swedish	SWE	sv
T-CHINESE	Chinese-Traditional Big-5	ROC	tw
THAI	Thai	THA	th
TURKISH	Turkish	TUR	tr

LAYOUTGRID

レポートを設計する際に、レポート出力にグリッドを表示し、データやオブジェクトを正確に配置することができます。このオプションは、PDF、DHTML レポート出力を使用する場合にのみ適用されます。

構文は次のとおりです。

SET LAYOUTGRID = {ON|OFF}

説明

ON

レポート出力に、グリッドを表示します。

OFF

レポート出力に、グリッドを表示しません。デフォルト値は OFF です。

LAYOUTRTL

レポート出力を右から左へ表示します (この表示方法がサポートされているフォーマットの場合)。

構文は次のとおりです。

SET LAYOUTRTL = {ON|OFF}

説明

**ON**

レポート出力を右から左へ表示します。

**OFF**

レポート出力を左から右へ表示します。デフォルト値は OFF です。

## LEADZERO

ダイアログマネージャの文字列では、先頭のゼロは切り捨てられます。WebFOCUS の関数には、ダイアログマネージャで呼び出した場合、結果として数値を返すものがあります。結果のフォーマットが YMD で、年を表す値に 00 が存在する場合、その 00 は切り捨てられます。

構文は次のとおりです。

**SET LEADZERO = {ON|OFF}**

説明

**ON**

先頭のゼロが存在する場合に、その値を表示します。

**OFF**

先頭のゼロが存在する場合に、その値を切り捨てます。デフォルト値は OFF です。

## LEFTMARGIN

LEFTMARGIN パラメータは、ページに表示するレポートコンテンツにスタイルシートの左側境界を設定します。このパラメータは、PDF レポートに適用されます。

構文は次のとおりです。

**SET LEFTMARGIN = {value|.250}**

説明

**value**

ページに表示するレポートコンテンツの左側の境界です。デフォルト値は 0.250 インチです。

## LINES

LINES パラメータは、上部の見出しから下部の脚注までを含めて、1 ページに表示する出力行の最大数を設定します。

このパラメータは、上部の見出しから下部の脚注までを含めて、論理ページに表示する行の最大数を設定します。LINES の許容値は 1 から 999999 です。スタイルフォーマットの出力では、連続用紙を生成するには 999 以上の値を指定します。連続用紙が指定され、出力フォーマットのページに物理サイズが使用される場合 (例、PDF 出力)、列タイトルが物理ページの上部に繰り返し表示されます。この場合、ページ番号は表示されません。スタイルなしフォーマットの出力では、連続用紙に 999999 の値を指定します。

WebFOCUS は、SET パラメータで定義した行グループに続いて HEADING および FOOTING テキストを生成します。

構文は次のとおりです。

```
SET LINES = {n|57}
```

説明

**n**

論理ページに表示する出力行の最大数です。デフォルト値は 57 です。

## LOOKGRAPH

LOOKGRAPH パラメータは、グラフのスタイルを指定します。このパラメータを使用することで、散布図、棒グラフ、ヒストグラム、円グラフをはじめとするグラフスタイルでさまざまなバリエーションを作成することができます。

構文は次のとおりです。

```
SET LOOKGRAPH = option
```

説明

**option**

グラフのスタイルを指定します。WebFOCUS には、さまざまなグラフスタイルが用意されています。

## MATCHCOLUMNORDER

MATCHCOLUMNORDER パラメータは、MATCH 出力での MATCH FILE リクエストのフィールド順序を制御します。新しいデフォルト設定の方法では、MATCH リクエスト内での位置に関係なく、複数のファイルのフィールドを結果の HOLD ファイルのソートフィールド値でグループ化します。従来の方法では、リクエスト内で参照されるとおりに HOLD ファイルにフィールドが追加されます。

構文は次のとおりです。

```
SET MATCHCOLUMNORDER = {GROUPED|UNGROUPED}
```

## 説明

**GROUPED**

複数ファイルの動詞オブジェクトを、最上位の共通ソートキーでグループ化します。その結果、従来の処理とは異なる順序でフィールドが HOLD ファイルに継承されます。これは、HOLD マスターファイルで生成されたデフォルト設定のエイリアス名が後続リクエストに使用された場合、後続リクエストの MATCH の結果に影響します。通常、キーとして使用されない同一名のフィールドが MATCH で統合された場合にこの問題が発生します。この方法の利点は、関連するマスターファイルを生成する HOLD ファイルフォーマット (例、FORMAT FOCUS) がサポートされることです。

**UNGROUPED**

複数ファイルの動詞オブジェクトを、結果の HOLD ファイルのレコードを表示する際にソートキーでグループ化しません。フィールドは、リクエストで参照されるとおりに HOLD ファイルのレコードに追加されます。

**MAXDATAEXCPT**

データに数値のみを含めることができる場合に、このデータをオペレーティング環境のアーキテクチャでサポートされていない方法で操作すると、データ例外が発生します。SET MAXDATAEXCPT コマンドを使用することで、セッション終了時まで発生可能なデータ例外の個数を変更することができます。

**注意：**SET MAXDATAEXCPT は、メインフレームプラットフォームでのみ機能します。その他のすべてのプラットフォームでこの構文を使用することはできませんが、この機能はサポートされません。

ON TABLE SET 句を使用する TABLE リクエストでこのコマンドを発行すると、そのリクエストに対して新しいカウントが作成されます。実行中セッションのカウントは保存され、リクエストの実行後に復元されます。

構文は次のとおりです。

```
SET MAXDATAEXCPT={@|maxexcpt}
```

## 説明

**maxexcpt**

セッション終了時まで発生可能なデータ例外の個数を表す 1 から 4 までの数字です。この値に 0 (ゼロ) を指定すると、データ例外の個数が無制限になります。これがデフォルト値です。この値に 1 を指定すると、最初のデータ例外の発生でセッションが終了します。

リクエストで MAXDATAEXCPT を変更した場合、新しいカウントが作成され、実行中セッションのカウントは保存されて、リクエストの実行後に復元されます。このコマンドを TABLE リクエストの外部で発行した場合、実行中のカウンタは 0 (ゼロ) にリセットされます。

### MAXLRECL

MAXLRECL パラメータは、OCCURS セグメントを含む外部ファイルにレコード長の最大値を定義します。デフォルト値は 0 (ゼロ) です。ただし、デフォルト設定では、WebFOCUS は 12 キロバイトのレコード長を読み取ることができます。この値は、最大で 64 キロバイトまで設定することができます。データフィールド内部メモリ領域の最大の長さは 32 キロバイトであり、この値は変更されていません。

**注意：**MAXLRECL は文字ベースです。バイトベースではありません。Unicode 環境では、UNIX および Windows の 1 文字に対して 3 バイトを使用します。2 バイト文字セットを使用する場合は、1 文字に 2 バイトを使用します。

構文は次のとおりです。

```
SET MAXLRECL = {n|0}
```

説明

n

OCCURS セグメントを含む外部ファイルに適用するレコード長の最大値です。デフォルト値は 0 (ゼロ) です。

### MDICARDWARN

MDICARDWARN パラメータは、ディメンションのカーディナリティが特定の値を超えるたびに警告メッセージを表示します。これにより、MDI ビルドの状況を確認することができます。ディメンションのデータの同一値の個数が所定の割合に達すると、警告メッセージが発行されます。MDICARDWARN の信頼性を高めるには、データソースに 10 万件以上のレコードが必要です。

**注意：**警告メッセージ以外に、大括弧 ([ ]) で囲まれた数字が表示されます。この数字は、警告メッセージのテキストに提示されたディメンションのデータ値の個数の最小値を示しています。

構文は次のとおりです。

```
SET = MDICARDWARN = n
```

説明

**n**

0 から 50 までのパーセント値です。

## MDIENCODING

MDIENCODING パラメータを使用すると、データソースを読み込まずに MDI ファイルから出力を取得することができます。

MDI のフィールドまたはディメンションに MAXVALUES 属性が指定されている場合、または親子関係に関連している場合は、WebFOCUS は常にインデックス値をエンコードします。エンコードした値は再構築時に MDI ファイルに格納され、MTABLE リクエストに MDIENCODING コマンドを指定して抽出してデコードすることができます。MDIENCODING コマンドを使用すると、データソースを読み取らずに MDI ファイル自体から出力を取得することができます。

MDIENCODING を使用する TABLE リクエストのフィールドには、次の規則が適用されます。

- ❑ 一度に参照できる MDI は 1 つです。
- ❑ 1 つのリクエストで同時に使用できるディメンションは、同一の親子階層に属するディメンションのみです。同一の親子階層に属さないディメンションでも、MDIENCODING 属性が指定されている場合は、リクエスト内のフィールドとして使用することができます。

構文は次のとおりです。

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

説明

**ON**

データソースを読み取らずに MDI ファイルから出力を取得することができます。

**OFF**

MDI の値を取得するには、データソースにアクセスする必要があります。

**注意：**このコマンドは、ON TABLE 句でのみ発行することができます。このパラメータにはデフォルト値はありません。

## MDIPROGRESS

MDIPROGRESS パラメータは、MDI ビルドの進行状況に関するメッセージを表示します。このメッセージには、MDI に n 件のデータレコードが挿入されるたびに、累積されたレコード数が表示されます。

構文は次のとおりです。

```
SET MDIPROGRESS = {n|0}
```

説明

n

1000 より大きい整数です。MDI ビルドに n 件のレコードが累積されるたびに進行状況のメッセージを表示します。デフォルト値は 100,000 です。

0

進捗メッセージを無効にします。

## MESSAGE

MESSAGE パラメータは、Web ブラウザのソース表示機能においての情報メッセージの表示を制御します。このパラメータは、ON TABLE SET と併用することはできません。

構文は次のとおりです。

```
SET {MESSAGE|MSG} = {ON|OFF}
```

説明

ON

情報メッセージを表示します。デフォルト値は ON です。

OFF

FOCUS がプロシジャでコマンドを実行する場合は、メッセージおよびキャレットはともに非表示になります。エラーメッセージおよび入力を要求するキャレットは通常どおり表示されます。

## MISS\_ON

一時項目 (DEFINE) または一時項目 (COMPUTE) にミッシング値が含まれるよう設定されている場合、DEFINE または COMPUTE の作成式に使用されるフィールド値のすべて (ALL) がミッシング値の場合に結果フィールドをミッシングとして処理するか、フィールド値の一部 (SOME) がミッシング値の場合に結果フィールドをミッシングとして処理するかを制御することができます。MISSING ON が設定された DEFINE または COMPUTE で ALL または SOME のどちらも指定しない場合、デフォルト値は SOME になります。

SET パラメータの MISS\_ON を使用して、MISSING ON が設定された DEFINE または COMPUTE で ALL または SOME のいずれも指定されていない場合に、どちらを使用するかを指定することができます。

構文は次のとおりです。

```
SET MISS_ON = {SOME|ALL}
```

説明

#### SOME

式のフィールドの少なくとも 1 つに値がある場合、一時項目には値が存在することを示します。計算でフィールドのミッシング値は 0 (ゼロ) またはブランクとして評価されます。式のすべてのフィールドの値が欠落している場合、一時項目の値は欠落します。デフォルト値は **SOME** です。

#### ALL

式のすべてのフィールドに値がある場合、一時項目に値が存在することを示します。式で指定されたフィールドの少なくとも 1 つにミッシング値が存在する場合にも、一時項目にミッシング値があると解釈します。

## MISSINGTEST

デフォルト設定では、IF-THEN-ELSE 式を使用して結果を計算する際に、その式の結果が 0 (ゼロ) (数値式の場合) またはブランク (文字式の場合) と評価された場合、その式の左側に MISSING ON が存在するかどうかを確認されます。その場合、式の結果が true または false ではなく MISSING となり、フィールドに何らかのミッシング値が必要な場合も返される結果は MISSING で、THEN 式または ELSE 式を評価した結果にはなりません。SET MISSINGTEST コマンドを使用して、IF 式のミッシングテストを除外することができます。これにより、THEN 式または ELSE 式が評価されて結果として返されます。

構文は次のとおりです。

```
SET MISSINGTEST = {NEW|OLD|SPECIAL}
```

説明

#### NEW

IF 式をミッシング値の評価から除外します。これにより、結果が MISSING ではなく、true または false になります。評価が true の場合、THEN 式を使用して結果が計算されます。結果が false の場合、ELSE 式を使用して結果が計算されます。これがデフォルト値です。

#### OLD

IF 式をミッシング値の評価に含めます。IF 式の評価結果が MISSING の場合、ミッシングフィールドに何らかのミッシング値が必要な場合は結果も MISSING になります。

### SPECIAL

パラメータを RStat に渡すために必要です。

## MULTIPATH

MULTIPATH パラメータは、独立パスで実行するテストを制御します。

構文は次のとおりです。

**SET MULTIPATH = {[SIMPLE](#)|[COMPOUND](#)}**

説明

### SIMPLE

次の場合、レポート出力に親セグメントが表示されます。

- ☐ 親セグメントに属する子セグメントの中で、選別条件を満たす子セグメントが少なくとも 1 つ存在する。
- ☐ 親セグメントが参照する子セグメントはパス上に存在しないが、オプションの子セグメントが存在する。

マルチパスレポートのデータを選別するリクエストの場合、警告メッセージ (FOC144) が表示されます。

([FOC144](#)) 警告。無関連のオカレンスのテストです。

### COMPOUND

必要となるすべての子セグメントが親セグメントに存在する場合は、レポート出力に親セグメントが表示されます。COMPOUND 設定は、([FOC144](#)) 警告メッセージを生成しません。デフォルト値は COMPOUND です。

セグメントの規則は、FOCUS がデータソース階層またはビュー階層を降順に移動する際にレベルごとに適用されます。データソースのツリー全体では、親セグメントの存在は子セグメントの存在の上に、子セグメントの存在は孫セグメントの存在の上にそれぞれ成立します。

## NEG-ZERO

SET NEG-ZERO パラメータは、負の小数値を端数処理した結果の 0 (ゼロ) を負の値として表示します。

構文は次のとおりです。

**SET NEG-ZERO = {[OFF](#)|[ALWAYS](#)|[EXCEL](#)}**

## 説明

### OFF

負のゼロ値を表示しません。これがデフォルト設定の従来の動作です。

### ALWAYS

負の数値を端数処理した結果の 0 (ゼロ) に対し、マイナス符号を含む負のフォーマットを適用します。

### EXCEL

負の数値を端数処理した結果の 0 (ゼロ) に対し、負の形式 B、R、および CR を適用します。

Excel では、NEG-ZERO で設定した値にかかわらず、このスキーマを使用して常に負のゼロ値を表示します。

## 注意

- ❑ NEG-ZERO パラメータは、指数フォーマット (例、E10.2) には適用されません。
- ❑ NEG-ZERO パラメータは、表示以外の目的で端数処理が実行される場合は適用されません。たとえば、演算の一部として実行される浮動小数点数からパック 10 進数または整数への変換には適用されません。

## NODATA

NODATA パラメータは、レポート内でミッシングデータであることを表す文字列を指定します。

構文は次のとおりです。

```
SET {NODATA|NA} = {string|.}
```

## 説明

### string

レポート内でミッシングデータであることを表す文字列です。ピリオド (.) がデフォルト値です。

## NULL

NULL パラメータを使用すると、可変長カンマ区切りまたはタブ区切りの HOLD ファイルを作成し、ブランク文字列または 0 (ゼロ) 値とミッシング値の両者を区別することができます。

SET NULL=ON でサポートされる HOLD フォーマットは、COM、COMT、TAB、TABT です。レコードのミッシング値は、連続した 2 つの区切り記号で明示されます。ミッシング値で始まるレコードでは、先頭に区切り文字が挿入されます。ミッシング値で終了するレコードでは、末尾に区切り文字が挿入されます。

構文は次のとおりです。

**SET NULL = {ON|OFF}**

説明

**ON**

マスターファイルでフィールドに MISSING=ON を指定している場合、ミッシング値を区切りテキスト形式の HOLD ファイルに継承します。

**OFF**

数値のミッシング値には 0 (ゼロ) の値、文字のミッシング値にはブランク (" ") を区切りテキスト形式の HOLD ファイルに継承します。デフォルト値は OFF です。

## OFFLINE-FMT

OFFLINE-FMT パラメータは、リクエストから生成されたレポート出力のフォーマットを指定します。

構文は次のとおりです。

**SET OFFLINE-FMT = option**

説明

**option**

次のいずれかです。

**STYLED** - 現在有効なスタイルシートに基づいてレポートのフォーマットを設定します。

**STANDARD** - スタイルが未設定の文字ベースの出力としてレポートを生成します。デフォルト値は STANDARD です。

## OLAPGRMERGE

複数の BY フィールドを含む、または BY フィールドと ACROSS フィールドを含む OLAP グラフで、これらのグラフを単一グラフに統合するかどうかを指定します。

構文は次のとおりです。

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

説明

**ON**

グラフの統合オプションをオンにします。この設定では、グラフの AUTODRILL が無効になります。

**OFF**

グラフの統合オプションをオフにします。外部ソートフィールドの値ごとに、それぞれ別のグラフが作成されます。デフォルト値は OFF です。

## OLDSTYRECLEN

OLDSTYRECLEN パラメータは、レコード長の LRECL を現在の設定である LRECL=0 に設定するか、WebFOCUS バージョン 5.3 SP04 以前に使用していた LRECL=512 に設定するかを指定します。

構文は次のとおりです。

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

説明

**ON**

LRECL=512 を使用します。

**OFF**

LRECL=0 を使用します。デフォルト値は OFF です。

## ONFIELD

ONFIELD パラメータは、リクエストに存在しないフィールドを参照する ON 句を無視するか、リクエストを終了するかを決定します。存在しないフィールドの ON 句を有効にすると、実行時のユーザ選択で、リクエストの実行ごとにどの要素を含めるかを指定することができます。

使用されるフィールドは、すべてデータソースのマスタファイルに存在する必要があり、存在しない場合は、処理が停止します。

構文は次のとおりです。

```
SET ONFIELD = {ALL|IGNORE}
```

```
ON TABLE SET ONFIELD {ALL|IGNORE}
```

説明

ALL

ON 句で参照されているフィールドがリクエストに存在しない場合、メッセージが発行されて処理が停止します。デフォルト値は ALL です。

IGNORE

リクエストに存在しないフィールドを参照する ON 句、および指定したフィールドのタイプでサポートされていないオプションを含む ON 句を無視します。

## ONLINE-FMT

ONLINE-FMT パラメータは、レポート出力のフォーマットを指定します。このパラメータは、WebFOCUS にのみ適用されます。

構文は次のとおりです。

**SET ONLINE-FMT = option**

説明

option

次のいずれかです。

**HTML** - レポートを HTML ページとして出力します。デフォルト値は HTML です。

**PDF** - レポートを PDF ドキュメントとして出力します。

**Excel 2000 (EXL2K)** - レポートを Excel 2000 ワークシートとして出力します。

**Excel 97 (EXL97)** - レポートを Excel 97 ワークシートとして出力します。

**STANDARD** - 文字ベースおよび行ベースの固定幅フォントのレガシーフォーマットでレポートを出力します。

## ORIENTATION

ORIENTATION パラメータは、スタイルシートでスタイルを設定したレポートのページの方角を指定します。

構文は次のとおりです。

**SET ORIENTATION = {PORTRAIT|LANDSCAPE}**

説明

PORTRAIT

縦長スタイルでページを表示します。デフォルト値は PORTRAIT です。

## LANDSCAPE

横長スタイルでページを表示します。

## OVERFLOWCHAR

OVERFLOWCHAR パラメータは、数値レポート列に値を表示するための十分な領域がない場合に、表示される文字を制御します。表示されるオーバーフロー文字の数は、フィールドに割り当てられた長さと同じです。デフォルト設定では、オーバーフロー文字としてアスタリスク (\*) が表示されます。

構文は次のとおりです。

```
SET OVERFLOWCHAR = 'char'
```

説明

**char**

1 バイトの表示可能文字です。指定する文字には、一重引用符 (') で囲む必要がないものがあります。

数字、カンマ (,), ピリオド (.), アポストロフィ ('), パーセント記号 (%), マイナス記号 (-), ブランク、現在の通貨記号、ドル記号 (\$), 円記号 (¥), ポンド記号、ユーロ記号は、オーバーフロー文字としてサポートされません。また、オペレーティング環境によっては、その他の記号が有効な場合があります。

## PAGE[-NUM]

PAGE[-NUM] パラメータは、出力ページのページ番号を制御します。

構文は次のとおりです。

```
SET PAGE[-NUM] = option
```

説明

**option**

次のいずれかです。

**ON** - ページの左上隅にページ番号を表示します。デフォルト値は ON です。

**OFF** - ページ番号を非表示にします。

**NOPAGE** - 改ページを非表示にします。その結果、レポートが連続的な論理ページとして表示されます。PAGE を NOPAGE に設定した場合、フィールドタイトルの表示位置は LINES パラメータで制御します。NOPAGE の代わりに NOLEAD を使用することもできます。

### PAGE-SCALE

PAGE-SCALE パラメータは、新しいパネルを開かずに、ページ幅に合わせて PDF レポート出力を拡大縮小して表示します。

デフォルト設定では、PDF レポート出力の幅が広すぎて単一ページに収まらない場合、収まりきらない列を表示するために同一ページに複数のパネルが生成されます。ページ番号は、ページ番号とパネル番号を指定します。たとえば、ページ番号が 1.1 および 1.2 の場合、ページ 1/パネル 1、ページ 1/パネル 2 をそれぞれ表します。

構文は次のとおりです。

```
SET PAGE-SCALE = {OFF|AUTO}
```

説明

#### [OFF](#)

PDF レポート出力でのページの拡大縮小表示を無効にします。これがデフォルト値です。

#### [AUTO](#)

PDF レポート出力でのページの拡大縮小表示を有効にします。

### PAGESIZE

PAGESIZE パラメータは、レポートページの用紙サイズを指定します。

構文は次のとおりです。

```
SET PAGESIZE = size
```

説明

#### [size](#)

用紙サイズを指定します。実際の用紙サイズが PAGESIZE 設定に一致しない場合は、レポートはトリミングされるか、レポートに余分なスペースが埋め込まれます。

用紙サイズのオプションは次のとおりです。

LETTER - 8.5 x 11 インチ (21.6 x 27.9 cm)

ENVELOPE-PERSONAL - 3.625 x 6.5 インチ (9.2 x 16.5 cm)

ENVELOPE-MONARCH - 3.875 x 7.5 インチ (9.8 x 19.1 cm)

ENVELOPE-9 - 3.875 x 8.875 インチ (9.8 x 22.5 cm)

ENVELOPE-10 - 4.125 x 9.5 インチ (10.5 x 24.1 cm)

ENVELOPE-12 - 4.5 x 11 インチ (11.4 x 27.9 cm)

ENVELOPE-DL - 4.3 x 8.6 インチ (10.9 x 21.8 cm)

ENVELOPE-ITALY - 4.3 x 9.1 インチ (10.9 x 23.1 cm)

ENVELOPE-B4 - 9.8 x 13.9 インチ (24.9 x 35.3 cm)

ENVELOPE-B5 - 6.9 x 9.8 インチ (17.5 x 24.9 cm)

ENVELOPE-B6 - 6.9 x 4.9 インチ (17.5 x 12.4 cm)

ENVELOPE-C3 - 12.75 x 18 インチ (32.4 x 45.7 cm)

ENVELOPE-C4 - 9 x 12.75 インチ (22.9 x 32.4 cm)

ENVELOPE-C5 - 6.4 x 9 インチ (16.3 x 22.9 cm)

ENVELOPE-C6 - 4.5 x 6.375 インチ (11.4 x 16.2 cm)

ENVELOPE-C65 - 4.5 x 9 インチ (11.4 x 22.9 cm)

STATEMENT - 5.5 x 8.5 インチ (14.0 x 21.6 cm)

EXECUTIVE - 7.5 x 10.5 インチ (19.1 cm x 26.7 cm)

GERMAN-STANDARD-FANFOLD - 8.5 x 12 インチ (21.6 x 30.5 cm)

GERMAN-LEGAL-FANFOLD - 8.5 x 13 インチ (21.6 x 33.0 cm)

FOLIO - 8.5 x 13 インチ (21.6 x 33.0 cm)

LEGAL - 8.5 x 14 インチ (21.6 cm x 35.6 cm)

10X14 - 10 x 14 インチ (25.4 cm x 35.6 cm)

TABLOID - 11 x 17 インチ (27.9 cm x 43.2 cm)

CUSTOM - DHTML、PDF、PPTX レポートのカスタムページサイズ設定を有効にします。

A3 - 11.7 x 16.8 インチ (29.7 x 42.7 cm)  
A4 - 8.25 x 11.7 インチ (21.0 x 29.7 cm)  
A5 - 5.8 x 8.25 インチ (14.7 x 21.0 cm)  
B4 - 9.8 x 13.9 インチ (24.9 x 35.3 cm)  
B5 - 7.2 x 10.1 インチ (18.3 x 25.7 cm)  
C - 17 x 22 インチ (43.2 x 55.9 cm)  
D - 22 x 34 インチ (55.9 x 86.4 cm)  
E - 34 x 44 インチ (86.4 x 111.8 cm)  
US-STANDARD-FANFOLD - 14.875 x 11 インチ (37.8 x 27.9 cm)  
LEDGER - 17 x 11 インチ (43.2 x 27.9 cm)  
QUARTO - 8.5 x 10.8 インチ (21.6 x 27.4 cm)

### PANEL

PANEL パラメータは、レポートパネルの最大幅をバイト数で指定します。レポート出力が最大幅を超える場合、出力は複数のパネルに分割されます。たとえば、PANEL を 80 に設定した場合、レコードの先頭の 80 文字が最初のパネルに表示され、次の 80 文字が 2 つ目のパネルに表示されます。

レポートを画面に表示する場合、PANEL パラメータの理想的な値は、使用する画面の幅になります (通常は 80)。レポートをプリンタに出力する場合、PANEL パラメータの理想的な値は、使用するプリンタの印刷幅になります (通常は 132)。PANEL の値がこの値を超える場合、または 0 (ゼロ) に設定されている場合、長いレポート行は画面またはページの周囲に回り込んで表示されます。

BYPANEL パラメータが OFF の場合、レポートを最大で 4 つのパネルに分割することができます。SET BYPANEL を OFF 以外の値に設定する場合は、レポートを 99 個のパネルに分割することが可能です。

STYLESHEET パラメータが有効になっている場合、PANEL は無視されます。

構文は次のとおりです。

**SET PANEL = {*n*|*n*}**

説明

***n***

レポートパネルの最大幅をバイト数で指定します。

***n***

レポートを複数のパネルに分割しません。長いレポート行は、画面またはページの周囲に回り込みます。デフォルト値は 0 (ゼロ) です。

**注意:** PANEL パラメータは、文字ベースの出力フォーマットのみに反映されます。文字ベース以外のすべての出力フォーマットでは、PANEL パラメータは無視されるか (HTML、AHTML、EXL2K)、独自のパネル化が実行されます (PDF)。PANEL パラメータを有効にするには、SET ONLINE-FMT=STANDARD または SET ONLINE-FMT=STYLED に設定する必要があります。

## PARTITION\_ON

統計関数を使用する際に、リクエストにソートフィールドが含まれる場合は、関数を計算するパーティションのサイズを設定する必要があります。この場合、PARTITION\_ON コマンドを使用して設定することができます。

構文は次のとおりです。

```
SET PARTITION_ON = {FIRST|PENULTIMATE|TABLE}
```

説明

### FIRST

リクエストの 1 つ目のソートフィールド (主ソートフィールド) を使用して、値を分割します。

### PENULTIMATE

一時項目 (COMPUTE) が評価される最後のソートフィールドの 1 つ前のソートフィールドを使用して、値を分割します。これがデフォルト値です。

### TABLE

内部マトリックス全体を使用して、統計関数を計算します。

## PASS

PASS パラメータは、DBA のセキュリティで保護されたデータソースまたはプロシジャへのユーザアクセスを有効にします。

このコマンドは、ON TABLE SET と併用することはできません。

構文は次のとおりです。

```
SET PASS = password [IN filename]
```

説明

### password

DBA のデータベースセキュリティで保護されたデータソースへのアクセスを許可するパスワードです。

### filename

セキュリティで保護された特定の FOCUS データソースまたはプロシジャです。

## PCOMMA

PCOMMA パラメータは、カンマ区切りファイルの取得を制御します。

デフォルト設定では、マスターファイルで SUFFIX=COM を指定している場合、文字入力値は二重引用符 (") で囲まれていません。また、各レコードは、カンマとドル記号の組み合わせ記号 (,\$) で終了しています。このフォーマットでは、PC アプリケーションで作成したほとんどのカンマ区切りファイルの取得はサポートされません。

構文は次のとおりです。

SET PCOMMA = option

### 説明

#### option

次のいずれかの値です。

- ❑ **ON** PC アプリケーションで作成したカンマ区切りデータソースの取得を有効にします。これは、文字データが二重引用符 (") で囲まれていること、および各レコードが 1 行内に完全に収まり、キャリッジリターン (CR) およびラインフィード (LF) で終了していることを示します。また、文字データが二重引用符 (") で囲まれておらず、各レコードがカンマ (,) およびドル記号 (\$) で終了している場合も、カンマ区切りデータソースを取得することができます。
- ❑ **OFF** PC アプリケーションで作成したカンマ区切りデータソースの取得を無効にします。これは、文字データが二重引用符 (") で囲まれていないこと、および各レコードがカンマとドル記号の組み合わせ記号 (,\$) で終了していることを示します。デフォルト値は OFF です。
- ❑ **DFIX** SUFFIX=COM、COMT、TAB、TABT が指定された区切りファイルが DFIX アダプタ経由で処理されます。この処理により、1 行に 2 つの区切り文字が検知された際に、より完全で意味のあるメッセージが提供されるとともに、ミッシング値の処理の一部が変更されます。DFIX 処理では、ミッシング値がフィールドに割り当てられます。

区切りファイルに DFIX 処理を使用するには、区切りファイルが次の要件を満たす必要があります。

- ❑ 各レコード全体が 1 行に収まり、レコード行が CRLF (キャリッジリターン/ラインフィード) 文字で終了する必要があります。

- ❑ 囲み文字は、COM (新規) および COMT レコードの区切り文字直後の先頭位置にのみ配置することができます。それ以外の位置では認識されません。
- ❑ 1 行あたりのフィールド数は、マスターファイルで定義されたフィールドの個数を超えることはできません。

## PCTFORMAT

PCTFORMAT パラメータは、演算接頭語 PCT.、RPCT.、PCT.CNT. を含むフィールドにパーセント記号で表示するか、元のフィールドに指定されたフォーマットで表示するかを制御します。

構文は次のとおりです。

```
SET PCTFORMAT = {OLD|PERCENT}
```

説明

### OLD

演算接頭語 PCT.、RPCT.、PCT.CNT. を含むフィールドを元のフィールドに指定されたフォーマットで表示します。

### PERCENT

演算接頭語 PCT.、RPCT.、PCT.CNT. を含むフィールドをパーセント記号で表示します。また、演算接頭語付きフィールドのフォーマットが再設定されます。これがデフォルト値です。

PCT.CNT. フィールドでは、フォーマットを再設定しない限り、常に小数点以下 2 桁で表示されます。PCT.CNT. および RPCT. フィールドで、SET PCTFORMAT = PERCENT に設定されている場合、元のフィールドのフォーマットにより次のように表示が異なります。

- ❑ 精度ベースフォーマット (F、D、M、X) の場合、フィールドの長さは 7 バイト、小数点以下 2 桁で表示されます。
- ❑ パック 10 進数フォーマットの場合、フィールドは元の小数点以下の桁数で表示されます。
- ❑ 整数フォーマットの場合、フィールドに小数点以下の桁は表示されません。

## PDFLINETERM

PDFLINETERM パラメータは、Windows と UNIX の間で正常にファイル転送を行えるように、PDF 出力ファイルの各レコードに余分なスペースを追加するかどうかを指定します。

Windows システムでは、各 PDF の最後に、バイトオフセットを保持しているテーブルがあります。この中には、2 つ 1 組の終端文字である、キャリッジリターン (CR) とラインフィード (LF) が記録されています。UNIX では、各ファイルの終端文字は 1 文字、つまりラインフィード (LF) のみです。Windows と UNIX のシステム間でファイルを転送するには、PDFLINETERM パラメータを正しく使用する必要があります。

構文は次のとおりです。

**SET PDFLINETERM = {[STANDARD](#)|[SPACE](#)}**

説明

### [STANDARD](#)

余分な文字を使用せずに PDF ファイルを作成します。このファイルは、Windows マシンにテキストモードで転送すると有効な PDF ファイルになりますが、UNIX マシンへの転送では無効です。その後に UNIX マシンから Windows マシンにテキストモードで転送された場合は、Windows マシン上で有効な PDF ファイルになります。

### [SPACE](#)

各レコードにブランクを 1 つ追加して PDF ファイルを作成します。このファイルは、UNIX マシンにテキストモードで転送すると有効な PDF ファイルになりますが、Windows マシンへの転送では無効です。その後に ASCII UNIX マシンから Windows マシンにバイナリモードで転送された場合は、Windows マシン上で有効な PDF ファイルになります。

## PERMPASS

PERMPASS パラメータは、セッション中または接続中は継続して有効になるパスワードを作成します。この設定は、サポートされているすべてのプロファイルで発行できますが、特にユーザプロファイルで発行すると、特定のユーザに限定してパスワードを作成できるため便利です。このパラメータは、ON TABLE 句で設定することはできません。このパラメータがすべてのユーザに適用されることを回避するため、EDASPROF では設定しないことをお勧めします。

PERMPASS が有効な場合は、既存のマスターファイルの DBA セクションで設定したすべてのセキュリティルールが適用されます。ユーザは、SET PASS または SET USER コマンドを発行して、別のセキュリティルールに従うユーザパスワードに変更を加えることはできません。この変更を加えようとすると、次のメッセージが表示されます。

(FOC32409) 恒久 PASS が有効化されています。これ以外は無視されます。  
値は変更されていません。

1 つのセッションで設定できる永続パスワードは 1 つです。永続パスワードを設定すると、セッション中はそのパスワードを変更することはできません。

構文は次のとおりです。

```
SET PERMPASS=userpass
```

説明

**userpass**

データソースに関連付けられたマスターファイルで DBA セキュリティルールが確立されている場合に、そのデータソースへのアクセスに使用するユーザパラメータです。

## PHONETIC\_ALGORITHM

PHONETIC\_ALGORITHM パラメータは、PHONETIC 関数で使用する音声アルゴリズムを指定します。PHONETIC 関数は、名前などの文字値について発音に基づいたインデックスを作成し、綴りが異なるが発音が同じ単語を同じグループに分類します。

構文は次のとおりです。

```
SET PHONETIC_ALGORITHM = {METAPHONE|SOUNDEX}
```

説明

**METAPHONE**

インデックス化に Metaphone アルゴリズムを使用します。Metaphone は、名前に使用する以外に、ほとんどの英単語に適しています。Metaphone アルゴリズムは、よく使用される多くのスペルチェッカーの基礎になっています。METAPHONE は、デフォルト設定のアルゴリズムです。

**注意：**Metaphone は、リレーショナル DBMS に送信される SQL では最適化されません。そのため、SQL DBMS のリクエストを最適化する必要がある場合は、SOUNDEX 設定を使用する必要があります。

**SOUNDEX**

Soundex は、英語発音の音声に基づいて名前をインデックス化する、従来の音声アルゴリズムです。

## POPUPDESC

POPUPDESC パラメータを使用すると、HTML レポートのフィールドタイトルにマウスポインタを置いたときに、フィールド説明のポップアップを表示させることができます。

構文は次のとおりです。

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

### 説明

#### ON

フィールド説明のポップアップを有効にします。

#### OFF

フィールド説明のポップアップを無効にします。デフォルト値は OFF です。

## PPTXGRAPHTYPE

PPTXGRAPHTYPE 属性を使用すると、PowerPoint (PPTX) のスライドに埋め込まれたグラフの画質が向上します。PNG 出力フォーマットを使用して、イメージおよびテキストの画質を向上させるとともに、透明度をサポートすることができます。この出力フォーマットは、背景色が定義されたテンプレートを使用する場合や、グラフに他のコンポーネントや描画オブジェクトを重ね合わせる場合など、多くの重要なシナリオで役立ちます。

構文は次のとおりです。

**SET PPTXGRAPHTYPE={PNG|PNG\_NOSCALE|JPEG}**

### 説明

#### PNG

画質を大幅に改善するために、PNG イメージを元のサイズの 2 倍に拡大します。絶対ポイントサイズのテキストなど (埋め込みスケール見出しを含む)、拡大縮小できない項目がグラフに含まれている場合、このオプションを選択すると問題が発生することがあります。また、ビットマップよりサイズが大きくなるため、出力サイズも増加します。グラフ内のテキストは、従来の JPEG フォーマットに比べてかなり鮮明になります。

PNG では、解像度の増加に伴いフォントサイズが内部的に拡大された場合でも、グラフ内のフォントサイズは保持されます。スタイルシート (\*GRAPH\_SCRIPT) で設定された絶対フォントサイズは、仮想座標で表現されるサイズに変換されて (グラフのサイズに相対的なサイズ)、仮想座標で埋め込み見出しおよび脚注のフォントサイズが生成されます。

#### PNG\_NOSCALE

PNG で描画されますが、拡大はされません。この設定では、JPEG より画質が若干向上します。JPEG を PNG\_NOSCALE に変更すると、グラフがより鮮明になりますが、テキストには若干の効果があるのみです。

#### JPEG

従来のフォーマットです。これがデフォルト値です。

## PRFTITLE

PRFTITLE パラメータは、演算接頭語付きフィールドに説明的な列タイトルを生成します。これらの列タイトルには、演算接頭語の読み取り可能、翻訳可能な説明が表示されます。

構文は次のとおりです。

```
SET PRFTITLE = {SHORT|LONG}
```

説明

### SHORT

フィールド名の上側に演算接頭語の名前を追加して列タイトルを生成します。

### LONG

フィールド名の上側に演算接頭語の長い名前を追加して、説明的な列タイトルを生成します。サーバで他言語が構成されている場合、演算接頭語の長い名前は、その言語に翻訳されて表示されます。

## PRINT

PRINT パラメータは、レポートの出力先を指定します。

レポート出力を画面またはプリンタのどちらに送信するかを指定します。

ONLINE および OFFLINE を独立したコマンドとして入力することもできます。これらのコマンドは、PRINT 設定の一部として ONLINE および OFFLINE を指定した場合と同一の効果があります。

構文は次のとおりです。

```
SET PRINT = {ONLINE|OFFLINE}
```

説明

### ONLINE

レポート出力をブラウザに送信します。デフォルト値は ONLINE です。

### OFFLINE

レポート出力をシステムプリンタに送信します。また、FILEDEF コマンドを使用して「online ddname」というラベルが付いたファイルにレポート出力を送信することもできます。

## PRINTDST

DST 演算子の処理が強化され、同一リクエストで複数の DST 演算子がサポートされるとともに、DST を ACROSS で使用できるようになりました。

これらの強化により、PRINT コマンドに複数の DST 演算子を指定して、独立した DST 値を取得するリクエストの動作を制御することが可能になります。この機能を実装するには、PRINTDST パラメータを NEW に設定する必要があります。

構文は次のとおりです。

```
SET PRINTDST = {OLD|NEW}
```

説明

### OLD

PRINT リクエスト内の複数の DST 演算子を、ネストした BY フィールドとして処理します。これにより、これらの DST 演算子が互いに依存した関係になります。デフォルト値は OLD です。

### NEW

PRINT リクエスト内の複数の DST 演算子を、完全に独立したオブジェクトとして処理します。

## PRINTPLUS

PRINTPLUS パラメータは、印刷の詳細設定を指定します。PRINTPLUS は、スタイルシートではサポートされません。

構文は次のとおりです。

```
SET {PRINTPLUS|PRTPLUS} = {ON|OFF}
```

説明

### ON

改ページを内部的に処理し、ページ間のブランクを正しく表示します。指定済み条件が一致するところで、RECAP を実行できます。また、SUBFOOT は、脚注の下部ではなく、上部に表示されます。

### OFF

スタイルシートをサポートしません。デフォルト値は OFF です。

## QUALCHAR

QUALCHAR パラメータは、修飾フィールド名に使用する修飾文字を指定します。

構文は次のとおりです。

```
SET QUALCHAR = {character|.}
```

## 説明

## character

有効な修飾文字です。次の値を指定することができます。

.	ピリオド	(hex 4B)
:	コロン	(hex 7A)
!	感嘆符	(hex 5A)
%	パーセント記号	(hex 6C)
	縦線	(hex 6A)
¥	円記号	(hex E0)

ピリオド (.) がデフォルト値です。上記リストのピリオド (.) 以外の修飾文字の使用には制限があります。それらの文字を、66 バイトのフィールド名に使用することはできません。

修飾文字がピリオド (.) の場合は、上記リストのピリオド以外の任意の文字をフィールド名の一部として使用することができます。デフォルトの修飾文字をピリオド (.) 以外の文字に変更した場合、その文字をフィールド名に使用することはできません。

## QUALTITLES

QUALTITLES パラメータは、マスターファイル内に同一のフィールド名が存在する場合に、レポート出力に修飾フィールドタイトルを使用します。修飾フィールドタイトルは、セグメント名を含めることにより同一のフィールド名を区別します。

構文は次のとおりです。

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

## 説明

## ON

同一のフィールド名が存在し、FIELDNAME が NEW に設定されている場合、修飾フィールドタイトルを使用します。

## OFF

修飾フィールドタイトルの使用を無効にします。デフォルト値は OFF です。

### RANK

RANK パラメータは、リクエストに [RANKED] BY [HIGHEST|LOWEST] *n* 句が含まれ、複数のデータ値が同一順位に分類される場合の順位番号の割り当て方法を指定します。次の値グループの順位が連番の次の整数値の場合、この順位付け方法は「DENSE」(連番) と呼ばれます。次の値グループの順位が前の順位に複数のデータ値の個数を加算した整数値の場合、この順位付け方法は「SPARSE」(非連番) と呼ばれます。

構文は次のとおりです。

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

説明

#### DENSE

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

#### SPARSE

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

### RECAP-COUNT

RECAP-COUNT パラメータは、出力する論理ページの単位ページあたりの行数を集計する際に、その集計値に RECAP で作成した値の行も含めます。

1 ページあたりの行数は、LINES パラメータで指定します。

構文は次のとおりです。

```
SET RECAP-COUNT = {ON|OFF}
```

説明

#### ON

RECAP で作成した値の行を行数の集計に含めます。

#### OFF

RECAP で作成した値の行を行数の集計に含めません。デフォルト値は OFF です。

### RECORDLIMIT

RECORDLIMIT パラメータは、取得または表示するレコード数を制限します。

構文は次のとおりです。

```
SET RECORDLIMIT = {RECORDLIMIT|OUTPUTLIMIT}
```

説明

#### RECORDLIMIT

RECORDLIMIT フィルタ (WHERE RECORDLIMIT EQ *n* または IF RECORDLIMIT EQ *n*) が設定されたリクエストで、レポート出力に表示するレコード数を、このフィルタで指定した読み取り数に制限します。デフォルト値は RECORDLIMIT です。

#### OUTPUTLIMIT

RECORDLIMIT フィルタ (WHERE RECORDLIMIT EQ *n* または IF RECORDLIMIT EQ *n*) が設定されたリクエストで、出力結果に表示するレコード数に RECORDLIMIT フィルタを適用します。

## RIGHTMARGIN

RIGHTMARGIN パラメータは、ページに表示するレポートコンテンツにスタイルシートの右側境界を設定します。このパラメータは、PDF レポートに適用されます。

構文は次のとおりです。

```
SET RIGHTMARGIN = {value|.250}
```

説明

#### value

ページに表示するレポートコンテンツの右側の境界です。デフォルト値は 0.250 インチです。

## RPAGESET

RPAGESET パラメータは、SUBFOOT で作成したテキストおよび RECAP で作成したフィールド値を出力に含める場合に、論理ページの単位ページあたりに表示する行数を制御します。

構文は次のとおりです。

```
SET RPAGESET = {NEW|OLD}
```

説明

#### NEW

論理ページの単位ページあたりの行数には、LINES パラメータの値に 2 を加算し、さらに SUBFOOT を使用した最上位 BY フィールドの個数を加算した値を設定します。

### [OLD](#)

論理ページの単位ページあたりの行数を、**LINES** パラメータの値に設定します。詳細は、607 ページの「[LINES](#)」の説明を参照してください。デフォルト値は **OLD** です。

## SAVEDMASTERS

**SAVEDMASTERS** パラメータは、リクエストで使ったマスターファイルをメモリに保存します。マスターファイルをメモリに保存すると、その後のリクエストで参照する際に、そのマスターファイルを再び解析する必要がなくなるためパフォーマンスが向上します。

マスターファイルは 99 個までメモリに保存することができます。

このパラメータは、**ON TABLE SET** コマンドで設定することはできません。

構文は次のとおりです。

```
SET SAVEDMASTERS = n
```

説明

**n**

**SAVEDMASTERS** リストに表示するマスターファイルの最大数を 0 (ゼロ) から 99 までの整数で指定します。デフォルト値は 10 です。

なお、**SAVEDMASTERS** を 0 (ゼロ) に設定した場合でも、最後に使ったマスターファイルは常にメモリに保存されます。ただし、0 (ゼロ) に設定した場合は、保存済みマスターファイルのリストは生成されません。

## SAVEMATRIX

**SAVEMATRIX** パラメータは、リクエストからの内部マトリックスを保存し、ダイアログマネージャコマンドを使用する際に上書きされないように保護します。

構文は次のとおりです。

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

説明

**ON**

最終のリクエストから内部マトリックスを保存し、上書きを防止します。

[OFF](#)

リクエストごとに内部マトリックスを上書きします。デフォルト値は **OFF** です。

## SHADOW

SHADOW パラメータは、FOCUS データベースファイルのシャドウライティング機能を有効にする場合にのみ機能します。

構文は次のとおりです。

```
SET SHADOW [PAGE] = {ON|OFF|OLD}
```

説明

**ON**

シャドウライティング機能を有効にします。シャドウライティングの最大ページ数は 256 キロバイトです。

**OFF**

シャドウライティング機能を無効にします。デフォルト値は OFF です。

**OLD**

使用する FOCUS ファイルが、バージョン 7.0 より前のバージョンで作成されたことを指定します。これは、シャドウライティングの最大ページ数が 63,551 であることを意味します。

## SHIFT

SHIFT パラメータは、シフト文字列の使用を制御します。

構文は次のとおりです。

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

説明

**ON**

ヘブライ語または 2 バイト文字サポート (DBCS) に使用するシフト文字列を指定します。

**OFF**

SHIFT が無効であることを示します。デフォルト値は OFF です。

## SHORTPATH

SHORTPATH パラメータは、LEFT OUTER JOIN でクロスリファレンスセグメントのミッシングインスタンスに対する選択条件の処理方法を制御します。

WebFOCUS では、SET ALL = ON または JOIN LEFT\_OUTER コマンドにより、LEFT OUTER JOIN を指定します。LEFT OUTER JOIN の場合、ホストファイルのレコードは、すべてレポート出力上に表示されます。クロスリファレンスセグメントインスタンスがホストセグメントインスタンスに存在しない場合(「ショートパス」と呼ばれます)、レポート出力に、クロスリファレンスセグメントのフィールドのミッシング値が表示されます。ただし、テスト目的でフィールドにミッシング値を割り当てることはできません。

従属セグメントに選別条件が設定されている場合、選別条件を満たさないセグメントインスタンスおよび対応するホストセグメントインスタンスはレポート出力から除外されます。セグメントインスタンスがミッシングの場合、セグメントのフィールドにミッシング値が割り当てられていないため、ミッシング値のテストは失敗します。

LEFT OUTER JOIN をリレーショナルエンジンで実行し、次の条件が両方とも適用される場合、ミッシングのクロスリファレンスセグメントインスタンスが存在するホストレコードは、WebFOCUS とは多少異なる方法で処理されます。

- ❑ クロスリファレンスセグメントに選別条件が設定されている。
- ❑ ホストのセグメントインスタンスに、対応するクロスリファレンスセグメントインスタンスが存在しない。

これらの条件に該当する場合、WebFOCUS ではレポート出力からホストレコードが除外されますが、リレーショナルエンジンでは従属セグメントのフィールドにミッシング値を割り当て、選別条件を適用します。ミッシング値が選別条件をパスする場合は、レコード全体がレポート出力に保持されます。この種の処理は、クロスリファレンスファイルに一致レコードが存在しないホストレコードをすべて検索または集計する場合や、ミッシングインスタンスを含むクロスリファレンスセグメントから別の従属セグメントへの DEFINE ベースの JOIN を作成する場合に役立ちます。

LEFT OUTER JOIN が有効な場合に、WebFOCUS でミッシングセグメントインスタンスのフィールドにミッシング値を割り当てるには、SET SHORTPATH=SQL コマンドを発行します。

**SET SHORTPATH = {[FOCUS](#)|SQL}**

説明

### **[FOCUS](#)**

対応するクロスリファレンスセグメントが存在せず、レポートにクロスリファレンスセグメントの選別条件が設定されている場合に、レポート出力のホストセグメントを除外します。デフォルト値は FOCUS です。

## SQL

OUTER JOIN のミッシングクロスリファレンスセグメントフィールドに、ミッシング値を割り当てます。このレコードに対して選別条件を適用し、選別テストにパスした場合、レコードをレポート出力に保持します。

**注意：**SET ALL=ON コマンドまたは JOIN LEFT\_OUTER コマンド (マスターファイル内の記述、またはそれ以外のいずれも) の結果として、有効な OUTER JOIN が存在する必要があります。

## SHOWBLANKS

SHOWBLANKS パラメータは、HTML および EXL2K レポート出力で、文字列の先頭または中間にあるブランクを保持します。

構文は次のとおりです。

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

説明

### [OFF](#)

HTML および EXL2K レポート出力で、文字列の先頭および中間に存在するブランクを削除します。デフォルト値は OFF です。

### [ON](#)

HTML および EXL2K レポート出力で、文字列の先頭および中間に存在するブランクを保持します。

## SORTMATRIX

SORTMATRIX パラメータは、外部メモリの使用を抑えてインメモリのソートを使用するかどうかを制御します。構文は次のとおりです。

```
SET SORTMATRIX = {SMALL|LARGE}
```

### 説明

#### SMALL

最大 2048 行の単一ソートマトリックスを作成し、挿入ソートに基づくバイナリサーチを取得時の集計に使用します。このマトリックスの最大行数は、このタイプのソートで最適パフォーマンスとなるよう指定されています。ソートマトリックスが最大行数に達すると、ディスク上の FOCSORT というファイルに書き込まれ、インメモリのマトリックスを空にして取得を継続します。FOCSORT への書き込みは必要に応じて何回でも行われます。データの最後が検出されると、残りの行が FOCSORT に書き込まれ、集計処理を完了すると同時に、マージルーチンによって、ソート文字列がすべて FOCSORT に結合されます。場合によっては、複数の結合フェーズが必要になります。

#### LARGE

SORTMEMORY パラメータで指定した適切なメモリ量が使用可能な場合に、メモリ内に大規模なマトリックスまたは複数の小規模なマトリックスを作成します。デフォルト値は LARGE です。この方法を使用する目的は、ディスクにレコードを書き込む前に、できるだけ多くのソートを内部メモリで実行することです。ディスク I/O がソート処理に必要かどうかは、ソートに割り当てられたメモリ量およびリクエスト出力のサイズによります。SORTMEMORY の容量が十分でなく、LARGE を有意義に利用できない場合、ソートのデフォルトは SMALL に設定されます。LARGE を使用することで、ディスク I/O の要件を大幅に抑えることができます。そのため、最終的にディスク I/O が必要な場合 (出力が非常に大きい場合) に、複数の結合フェーズを使用する必要が実質的になくなります。

## SORTMEMORY

SORTMEMORY パラメータは、ソートに利用可能な内部メモリ量を制御します。構文は次のとおりです。

```
SET SORTMEMORY = {n|512}
```

### 説明

**n**

ソートに利用可能なメモリのメガバイト数 (正の値) です。デフォルト値は 512 です。

## SPACES

SPACES パラメータは、レポート内の列間に挿入するブランク数を設定します。

このパラメータは、HTML、PDF、またはスタイルが設定されたレポートでは機能しません。

構文は次のとおりです。

```
SET SPACES = {AUTO|n}
```

説明

#### AUTO

列間に 1 つまたは 2 つのブランクを自動的に挿入します。デフォルト値は AUTO です。

#### n

レポートの列間に挿入するブランク数です。有効値は 1 から 8 の整数です。

## SQLTOPTTF

SQLTOPTTF パラメータを使用すると、SQL トランスレータが TABLE コマンドの代わりに TABLEF コマンドを生成することができます。

構文は次のとおりです。

**SET SQLTOPTTF = {ON|OFF}**

説明

#### ON

TABLEF コマンドの生成が可能な場合にこのコマンドを生成します。たとえば、JOIN または GROUP BY コマンドが使用されていない場合に TABLEF コマンドを生成することができます。デフォルト値は ON です。

#### OFF

常に TABLE コマンドを生成します。

## SQUEEZE

SQUEEZE パラメータは、スタイルシート機能にのみ適用されます。

このパラメータは、レポート出力の列幅を指定します。列幅は、データ値またはフィールドタイトルを基準にするか、マスターファイルで定義したフィールドフォーマットを基準にします。

構文は次のとおりです。

**SET SQUEEZE = {ON|OFF|n}**

説明

#### ON

最大幅のデータ値またはフィールドタイトルのいずれか大きい方を基準に列幅を割り当てます。

### OFF

マスターファイルで指定したフィールドフォーマットを基準に列幅を割り当てます。この値を指定すると、フィールドタイトルまたはフィールドフォーマット定義のいずれか長い方の幅まで列幅が拡張されます。デフォルト値は OFF です。

### n

列幅の設定に使用可能な UNITS パラメータの設定を基準にして特定の値を指定します。ただし、これは PDF でのみ有効です。

## %STRICTMATH

%STRICTMATH パラメータは、WebFOCUS で値を 0 (ゼロ) で除算した場合の処理方法を指定します。

構文は次のとおりです。

```
SET %STRICTMATH = {'OLD'|'NEW'}
```

説明

### OLD

値を 0 (ゼロ) で除算した場合、ゼロを返します。この値は、一重引用符 (') で囲む必要があります。

### NEW

値を 0 (ゼロ) で除算した場合、ゼロを返します。レポート出力の HTML ソースに次のエラーメッセージが生成されます。

(FOC201) ゼロによる除算が発生しました。

デフォルト値は NEW です。この値は、一重引用符 (') で囲む必要があります。

## STYLEMODE

STYLEMODE パラメータは、レポート出力として生成する HTML のタイプを指定します。

**注意：**複数のページでデータ列の位置を揃えるには、SQUEEZE パラメータを OFF に設定する必要があります。

構文は次のとおりです。

```
SET STYLEMODE = {FULL|FIXED|PAGED}
```

説明

### FULL

標準の HTML 出力を表示します。デフォルト値は FULL です。

**FIXED**

HTML に <PRE タグを生成し、ここでフォーマット設定したテキストを HTML として取り扱わないことを示します。

**PAGED**

WebFOCUS では、レポート出力を複数の HTML テーブルに表示します。この場合、各テーブルには、それぞれ別のレポートページが表示されます。Web サーバからファイルを取得する場合、大規模なファイルを 1 つ取得するよりも、このように小規模ファイルを複数取得する方が迅速に処理されます。

**STYLE[SHEET]**

STYLE[SHEET] パラメータは、スタイルシートパラメータの使用の有無を切り替えることにより、レポート出力のフォーマットを制御します。これらのパラメータを使用し、ページのサイズ、方向、マージンなどのフォーマットオプションを指定します。

構文は次のとおりです。

```
SET STYLE[SHEET] = {stylesheet|ON|OFF}
```

**説明****stylesheet**

スタイルシートファイルの名前です。UNIX および Windows では、ファイル拡張子 .sty を除外したスタイルシートファイルの名前を指定します。

PDF レポートでは、UNITS、TOPMARGIN、BOTTOMMARGIN、LEFTMARGIN、RIGHTMARGIN、PAGESIZE、ORIENTATION、SQUEEZE で指定したページレイアウト設定を使用します。また、LINES、PAPER、PANEL、WIDTH の設定は無視されます。

**ON**

エンドユーザのブラウザで定義されたプロポーショナルフォントを使用して HTML テーブルを作成します。デフォルト値は ON です。

PDF レポートでは、UNITS、TOPMARGIN、BOTTOMMARGIN、LEFTMARGIN、RIGHTMARGIN、PAGESIZE、ORIENTATION、SQUEEZE で指定したページレイアウト設定を使用します。また、LINES、PAPER、PANEL、WIDTH の設定は無視されます。

**OFF**

エンドユーザのブラウザで定義されたデフォルトの固定フォントを使用してフォーマット設定済みのレポートを作成します。

PDF レポートでは、LINES および WIDTH の設定が使用されます。UNITS、TOPMARGIN、BOTTOMMARGIN、LEFTMARGIN、RIGHTMARGIN、PAGESIZE、ORIENTATION、SQUEEZE の設定は無視されます。

このオプションは、ON TABLE コマンドで STYLE パラメータが設定されている場合に使用します。これは、この後に StyleSheet コマンドが続いて、それがプロシジャに埋め込まれることを示します。

### SUBTOTALS

SUBTOTALS パラメータは、集計行をレポートの詳細行の上部に表示するか、下部に表示するかを指定します。影響を受ける集計コマンドには、SUBTOTAL、SUB-TOTAL、RECOMPUTE、SUMMARIZE、COMPUTE、RECAP、COLUMN-TOTAL などがあります。

構文は次のとおりです。

```
SET SUBTOTALS {ABOVE|BELOW}
```

説明

**ABOVE**

集計行を詳細行の上部に配置し、レポート出力の各詳細行にソートフィールド値を表示します。

**BELOW**

集計行を詳細行の下部に配置します。デフォルト値は **BELOW** です。

### SUMMARYLINES

SUMMARYLINES パラメータを使用すると、1つのリクエストの集計行で演算接頭語を使用するフィールドと使用しないフィールドを組み合わせることができます。演算接頭語の処理は、すべての集計行に使用されます。演算接頭語を使用しないフィールドは、SUM 演算子を指定した場合のように処理されます。

このコマンドは、ON TABLE SET で使用することはできません。

構文は次のとおりです。

```
SET SUMMARYLINES = {NEW|OLD|EXPLICIT}
```

説明

**NEW**

すべての集計演算が総合計行に継承されます。すべての集計コマンドに対して演算接頭語の処理を使用します (演算接頭語を使用しないすべての集計フィールドは、SUM. 演算接頭語を使用した場合のように処理されます)。集計コマンドで指定したフィールドは、集計コマンドで作成された集計行および集計コマンドの継承で作成された集計行においてのみ値が生成されます。集計行では、文字フィールドの表示がサポートされます。デフォルト値は **NEW** です。

**OLD**

この値は使用できなくなりました。今後は **NEW** として処理されます。

**EXPLICIT**

**SUBTOTAL** および **RECOMPUTE** は、総合計行に継承されません。すべての集計コマンドに対して演算接頭語の処理を使用します (演算接頭語を使用しないすべての集計フィールドは、**SUM**. 演算接頭語を使用した場合のように処理されます)。集計コマンドで指定したフィールドは、集計コマンドで作成された集計行および集計コマンドの継承で作成された集計行においてのみ値が生成されます。集計行では、文字フィールドの表示をサポートします。

**注意：**このコマンドは、**ON TABLE SET** 構文を使用したリクエストではサポートされません。

**SUMPREFIX**

**SUMPREFIX** パラメータを使用すると、外部ソートを使用して文字または **SmartDate** フォーマットで集計を実行する際に、ユーザがアンサーセットの表示順序を選択することができます。

構文は次のとおりです。

```
SET SUMPREFIX = {FST|LST|MIN|MAX}
```

説明

**FST**

文字または日付データタイプ (**SmartDate**) を集計する場合、先頭の値が表示されます。

**LST**

文字または日付のデータタイプ (**SmartDate**) を集計する場合、最終の値が表示されます。デフォルト値は **LST** です。

**MIN**

文字データタイプまたは **SmartDate** データタイプを集計する場合、サーバのコードページおよび構成で設定されたソート順の最小値を表示します。

**MAX**

文字データタイプまたは **SmartDate** データタイプを集計する場合、サーバのコードページおよび構成で設定されたソート順の最大値を表示します。

### SUPPRESSDRILLDT

ドリルダウンで使用する DATETIME フィールドが、DT 関数を使用して渡されます。呼び出されたプロシジャでも DT 関数が使用されると、無効な日付時間値のエラーが発生します。これは、値が 2 度指定されるためです。SUPPRESSDRILLDT = ON パラメータは、DT 関数の二重使用を無効にします。

構文は次のとおりです。

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

説明

**ON**

DT 接頭語の二重使用を無効にします。

**OFF**

DT 接頭語の二重使用を無効にしません。デフォルト値は OFF です。

### TARGETFRAME

TARGETFRAME パラメータは、ブラウザに表示する HTML ファイルの見出しに、HTML コードの <BASE TARGET="frameName"> を追加します。基本レポートまたはグラフのすべてのドリルダウンハイパーリンクは、ここで指定したフレームに移動します。ただし、スタイルシートの TARGET 属性でこの設定を上書きした場合は例外です。

構文は次のとおりです。

```
SET TARGETFRAME = frameName
```

説明

**frameName**

ドリルダウンハイパーリンク (プロシジャまたは URL のいずれか) からの出力を表示する Web ページのフレームです。ここでは、\_blank、\_self、\_parent、\_top などの標準 HTML フレーム名またはユーザ定義の名前を指定することができます。

## TEMP

TEMP パラメータは、WebFOCUS で一時ファイルを特定のディレクトリに割り当てます。

構文は次のとおりです。

```
SET TEMP = directory
```

説明

**directory**

すべての一時ファイルを割り当てるディレクトリです。

## TEMPERASE

TEMPERASE パラメータは、WebFOCUS の接続を閉じた後で、接続中に作成された一時ファイルを保持するかどうかを指定します。このパラメータは、HOLD ファイルおよびセッション中に作成されるその他のファイルに適用されます。

構文は次のとおりです。

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

説明

**ON**

WebFOCUS の接続を閉じた後で一時ファイルをすべて削除します。デフォルト値は ON です。

**OFF**

WebFOCUS 接続中に作成された一時ファイルをすべて保持します。

### TESTDATE

TESTDATE パラメータは、一時的にシステム日付を変更し、動的ウィンドウをテストします。これにより、日付設定をシミュレーションし、プログラムの動作を確認することができます。TESTDATE は、テストデータを使用してテストを実行する場合にのみ使用します。TESTDATE の値は、システムから現在の日付を取得するすべての予約済み変数に影響を与えます。また、TESTDATE を設定することにより、日付を使用する FOCUS 内のすべての場所で影響を受けませんが、システムから直接参照する日付は影響を受けません。

TESTDATE には、TODAY の日付または YYYYMMDD フォーマットの日付のいずれかを指定することができます。これ以外の値を指定すると、次のメッセージが表示されます。

**TESTDATE MUST BE YYYYMMDD OR TODAY**

構文は次のとおりです。

**SET TESTDATE = {yyyymmdd|TODAY}**

説明

**yyyymmdd**

YYYYMMDD フォーマットの 8 桁の日付です。

**TODAY**

現在の日付です。デフォルト値は TODAY です。

### TIME\_SEPARATOR

このパラメータは、&TOD システム変数の時間構成要素に使用する区切り文字を定義します。

構文は次のとおりです。

**SET TIME\_SEPARATOR = {DOT|COLON}**

説明

**DOT**

時間構成要素の区切り文字にピリオド (.) を使用します。これがデフォルト値です。

**COLON**

時間構成要素の区切り文字にコロン (:) を使用します。

### TITLELINE

TITLELINE パラメータは、レポート出力の列タイトルの下線を制御します。

構文は次のとおりです。

`SET {TITLELINE|ACROSSLINE} = {ON|OFF|SKIP}`

説明

**ON**

レポート出力の列タイトルに下線を配置します。デフォルト値は ON です。

**OFF**

下線の代わりにブランク行を挿入します。

**SKIP**

下線およびブランク行を挿入しません。

## TITLES

TITLES パラメータは、マスターファイルで定義したフィールドタイトルをレポート出力の列見出しに使用するかどうかを制御します。

構文は次のとおりです。

`SET TITLES = {ON|OFF|NOPREFIX}`

`ON TABLE SET TITLES {ON|OFF|NOPREFIX}`

説明

**ON**

マスターファイルに TITLE 属性が存在する場合に、TITLE 属性の値をレポート出力の列見出しとして表示します。レポートリクエストのフィールドに演算接頭語が含まれている場合、演算接頭語と TITLE 属性の両方を使用して列見出しを作成します。TITLE 属性が存在しない場合、フィールド名が代わりに使用されます。デフォルト値は ON です。

**OFF**

フィールド名をレポート出力の列見出しとして表示します。レポートリクエストのフィールドに演算接頭語が含まれている場合、演算接頭語とフィールド名の両方を使用して列見出しを作成します。

**NOPREFIX**

マスターファイルに TITLE 属性が存在する場合に、TITLE 属性の値をレポート出力の列見出しとして表示します。TITLE 属性が存在しない場合、フィールド名が代わりに使用されます。レポートリクエストのフィールドに演算接頭語が含まれている場合、演算接頭語とフィールド名の両方を使用して列見出しを作成します。

### TOPMARGIN

TOPMARGIN パラメータは、ページに表示するレポートコンテンツにスタイルシートの上部境界を設定します。

このパラメータは、PDF レポートに適用されます。

構文は次のとおりです。

```
SET TOPMARGIN = {value|.250}
```

説明

**value**

レポート出力のページに設定する上部の境界です。デフォルト値は 0.250 インチです。

### UNITS

UNITS パラメータは、PDF レポートに適用されます。

このパラメータは、ページマージン、フィールドの位置、列幅に使用する長さ単位を指定します。

構文は次のとおりです。

```
SET UNITS = {INCHES|CM|PTS}
```

説明

**INCHES**

単位としてインチを使用します。デフォルト値は INCHES です。

**CM**

単位としてセンチメートルを使用します。

**PTS**

単位としてポイントを使用します。ここで、1 インチ = 72 ポイント、1 センチメートル = 28.35 ポイントです。

### USER

USER パラメータは、DBA のセキュリティで保護されたデータソースまたはプロシジャへのユーザアクセスを有効にします。

構文は次のとおりです。

```
SET USER = user
```

## 説明

## user

DBA のセキュリティで保護されたデータソースおよびプロシジャにパスワードを使用してアクセスできるユーザ ID です。

## USERFCHK

USERFCHK パラメータは、DEFINE FUNCTION で作成した関数の引数および WebFOCUS から提供された関数の引数に適用する検証レベルを制御します。パラメータの個数の検証には影響しませんが、常に正しい数を指定する必要があります。

USERFNS=SYSTEM 設定が有効になっている必要があります。詳細は、650 ページの「[USERFNS](#)」を参照してください。

プロファイルまたはプロシジャで次のコマンドを発行します。

```
SET USERFCHK = setting
```

## 説明

## setting

次のいずれかの値です。

**ON** - リクエスト内のパラメータを検証しますが、マスターファイルの DEFINE で使用する関数のパラメータは検証しません。パラメータの長さが妥当でない場合は、問題の解決を試みます。問題を解決できない場合は、メッセージを生成し、その問題に関係する式の評価を終了します。デフォルト値は ON です。

マスターファイルで指定した関数のパラメータは検証されないため、DEFINE フィールドを使用したリクエストを次に送信するまで、これらの関数のエラーは生成されません。問題が発生した場合は、次のメッセージが生成されます。

(FOC003) フィールド名に誤りがあります。

**OFF** - 次の場合を除き、パラメータは検証されません。

- ☐ パラメータが長すぎるため、演算用コードが格納されたメモリ領域を上書きする可能性がある場合は、サイズを自動的に小さくします。この場合、メッセージは発行されません。

**注意：**OFF 設定は、将来廃止される予定です。

- ☐ 文字パラメータが短すぎる場合は、ブランクを追加して長さを修正します。

**注意：**パラメータの検証を無効にすると予期しない問題が発生する可能性があるため、このオプションは使用しないでください。

**FULL** - ON と同一ですが、マスターファイルの **DEFINE** で使用する関数のパラメータも検証します。

**ALERT** - リクエスト内のパラメータを検証します。問題が検知されても処理は中断されません。マスターファイルの **DEFINE** で使用する関数のパラメータは検証しません。パラメータの長さが妥当でない場合は、問題の修正をバックグラウンドで試みます。問題が修正されてもメッセージは表示されません。問題を修正できない場合は、警告メッセージが生成されます。この設定を **OFF** にした場合と同様に、処理が続行されます。

## USERFNS

**USERFNS** パラメータは、WebFOCUS から提供された関数とローカルで作成した関数の名前が重複する場合に、どちらの関数を使用するかを指定します。

パラメータの検証は、**DEFINE FUNCTION** で作成した関数および WebFOCUS から提供された関数の両方に対して有効にすることができます。

構文は次のとおりです。

```
SET USERFNS= {SYSTEM|LOCAL}
```

説明

### SYSTEM

WebFOCUS から提供された関数と **DEFINE FUNCTION** コマンドで作成した関数に優先権を与えます。デフォルト値は **SYSTEM** です。

この設定は、パラメータの検証を有効にするために必要です。詳細は、649 ページの「**USERFCHK**」を参照してください。

### LOCAL

ローカルで作成した関数を優先します。この設定を有効にした場合、パラメータの検証は実行されません。

## VAUTO

**VAUTO** パラメータは、表示する値に合わせて縦軸のスケールを自動的に変更します。**VAUTO** が **OFF** の場合、縦軸のスケールを変更するには、**VMAX** および **VMIN** パラメータを使用する必要があります。詳細は、653 ページの「**VMAX**」および 653 ページの「**VMIN**」を参照してください。

構文は次のとおりです。

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

説明

**ON**

縦軸のスケールを自動的に変更します。デフォルト値は ON です。

**OFF**

縦軸のスケールを自動的に変更しません。エンドユーザは、VMAX および VMIN の値を指定する必要があります。

## VAXIS

VAXIS パラメータは、オフラインで生成するグラフに適用されます。

このパラメータは、縦軸の長さを行数で指定します。オンライン表示では、WebFOCUS がグラフの長さを端末の高さに合わせて自動調整するため、VAXIS の設定は無視されます。

構文は次のとおりです。

**SET VAXIS = {n|405}**

説明

**n**

ページの長さを行数で指定します。有効値は 20 から 405 までの整数です。デフォルト値は 405 です。

## VCLASS

VCLASS パラメータは、AUTOTICK が OFF の場合に、縦軸の目盛り間隔を指定します。詳細は、653 ページの「**VTICK**」を参照してください。

構文は次のとおりです。

**SET VCLASS = {nnn|0}**

説明

**nnn**

縦軸の目盛り間隔です。

**0**

縦軸の目盛り間隔として 0 (ゼロ) を設定します。デフォルト値は 0 です。

### VGRID

VGRID パラメータは、グラフの横軸または縦軸の目盛りの位置で罫線を描きます。詳細は、586 ページの「[GRID](#)」を参照してください。

構文は次のとおりです。

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

説明

**ON**

グラフの横軸または縦軸の目盛りの位置で罫線を描きます。

**OFF**

罫線を描きません。デフォルト値は OFF です。

### VISBARORIENT

VISBARORIENT パラメータを使用すると、ACROSS フィールドのピアグラフの方向を縦方向または横方向に設定することができます。

このパラメータは、HTML 出力でのみサポートされます。

構文は次のとおりです。

```
SET VISBARORIENT = {H|V}
```

説明

**H**

ピアグラフの方向に縦方向を指定します。

**V**

ピアグラフの方向に横方向を指定します。これがデフォルト値です。

## VMAX

VMAX パラメータは、自動スケール変更機能を使用しない場合に (VAUTO=OFF)、縦軸に表示する最大値を設定します。

構文は次のとおりです。

```
SET VMAX = {nnn|.00}
```

説明

nnn

VAUTO=OFF の場合の縦軸の最大値です。デフォルト値は .00 です。

## VMIN

VMIN パラメータは、自動スケール変更機能を使用しない場合に (VAUTO=OFF)、縦軸に表示する最小値を設定します。

構文は次のとおりです。

```
SET VMIN = {nnn|.00}
```

説明

nnn

VAUTO=OFF の場合に縦軸に表示する最小値です。デフォルト値は .00 です。

## VTICK

VTICK パラメータは、AUTOTICK が OFF の場合に、縦軸の目盛り間隔を設定します。詳細は、651 ページの「[VCLASS](#)」を参照してください。

構文は次のとおりです。

```
SET VTICK = {nnn|.00}
```

説明

nnn

縦軸の目盛り間隔です。デフォルト値は .00 です。

### VZERO

VZERO パラメータは、縦軸のミッシング値を 0 (ゼロ) として扱います。VZERO が OFF の場合、ミッシング値は無視されます。

構文は次のとおりです。

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

説明

#### ON

ミッシング値を 0 (ゼロ) として扱います。デフォルト値は ON です。

#### OFF

ミッシング値を無視します。

### WARNING

WARNING パラメータは、(FOC441) 警告「すでにファイルが存在します。CREATE しますか?」を非表示にします。

構文は次のとおりです。

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

説明

#### ON

警告メッセージを ON にします。デフォルト値は ON です。

#### OFF

警告メッセージを OFF にします。

### WEBARCHIVE

WEBARCHIVE パラメータは、EXL2K レポートを、関連ファイル (例、EXL2K PIVOT フォーマットのピボットキャッシュ) とともに単一 Web アーカイブドキュメント (.xmlh ファイル) にパッケージ化します。このフォーマットは、Excel 2002 以降でのみ使用することができます。

構文は次のとおりです。

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

説明

ON

EXL2K レポートを関連ファイルとともに単一 Web アーカイブドキュメント (.xmh ファイル) にパッケージ化します。デフォルト値は ON です。

OFF

複数のファイルを単一ドキュメントにパッケージ化しません。EXL2K レポートは .xht ファイルとして作成されます。

## WEBVIEWALLPG

WEBVIEWALLPG パラメータは、WebFOCUS ビューアに [すべてのページ] ボタンを表示するかどうかを制御します。

構文は次のとおりです。

SET WEBVIEWALLPG = {OFF|ON}

または

ON TABLE SET WEBVIEWALLPG {OFF|ON}

説明

ON

[すべてのページ] ボタンを表示します。デフォルト値は ON です。

OFF

[すべてのページ] ボタンを表示しません。

## WEBVIEWCLMSG

WEBVIEWCLMSG パラメータは、[閉じる] オプションをクリックした際に、Web ビューアを閉じるときのメッセージを表示するかどうかを制御します。

構文は次のとおりです。

SET WEBVIEWCLMSG = {ON|OFF|CLOSE}

説明

ON

WebFOCUS 言語翻訳ファイル (XXwebfoc\_strings.lng、XX は 2 文字の言語の略名) の「閉じる」メッセージを表示します。

たとえば、ENwebfoc\_strings.lng は、英語の翻訳ファイルです。デフォルト値は ON です。

### OFF

「閉じる」メッセージを表示しません。表示ページは WEBVIEWHOME パラメータで指定され、指定しない場合は、ブランクページが表示されます。

### CLOSE

「閉じる」メッセージを表示せずに、Web ビューアポートが表示されたブラウザウィンドウを閉じます。

## WEBVIEWCLOSE

WEBVIEWCLOSE パラメータを使用して、WebFOCUS のビューアコントロールバーに、[閉じる] オプションを表示するかどうかを指定します。

構文は次のとおりです。

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

説明

### ON

[閉じる] オプションを表示します。デフォルト値は ON です。

### OFF

[閉じる] オプションを表示しません。

## WEBVIEWER

WEBVIEWER パラメータは、Web ビューアを有効にして起動します。レポート出力の先頭ページは WebFOCUS ビューアに表示されますが、残りのページはユーザから要求されるまで Web サーバまたは Application Server に保持されます。

構文は次のとおりです。

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

説明

### ON

Web ビューアを有効にします。レポート出力の先頭ページは WebFOCUS ビューアに表示されますが、残りのページはユーザから要求されるまで Web サーバまたは Application Server に保持されます。

### OFF

すべてのレポート出力をブラウザに送信します。レポート出力が、標準のブラウザのウィンドウに表示されます。デフォルト値は OFF です。

## WEBVIEWHELP

WEBVIEWHELP パラメータは、WebFOCUS ビューアに [ヘルプ] ボタンを表示するかどうかを制御します。

構文は次のとおりです。

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

または

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

説明

### ON

[ヘルプ] ボタンを表示します。デフォルト値は ON です。

### OFF

[ヘルプ] ボタンを表示しません。

## WEBVIEWHOME

WEBVIEWHOME パラメータを使用して、WebFOCUS ビューアを閉じた際に HTML ページを表示することができます。

構文は次のとおりです。

```
SET WEBVIEWHOME = {home_URL|OFF}
```

説明

home\_URL

WebFOCUS ビューアを閉じた際に HTML ページを表示する有効な URL です。

### OFF

WebFOCUS ビューアを閉じた際にブランクのブラウザウィンドウを表示します。別のレポートを実行するには、URL を別途入力する必要があります。デフォルト値は OFF です。

**注意：**アクセシビリティの有効化についての詳細は、523 ページの「[ACCESSIBLE](#)」を参照してください。

### WEBVIEWTARG

WEBVIEWTARG パラメータを使用すると、ユーザが WebFOCUS ビューアをターゲットフレームで開くことができます。

構文は次のとおりです。

```
SET WEBVIEWTARG = {target_frame|OFF}
```

説明

#### target\_frame

ブラウザ内の既存フレームの名前、または予約済み HTML ターゲットフレームのいずれかの名前です。

\_blank WebFOCUS ビューア を新しいブラウザウィンドウで開きます。アクセシビリティを有効にしていないレポートでは、これがデフォルト値です。

\_self WebFOCUS ビューアをアンカーと同一のフレームで開きます。

\_parent WebFOCUS ビューアをアンカーを含む直接上の親フレームで開きます。

\_top レポートを実行したフレームで WebFOCUS ビューアを開きます。

#### OFF

レポートを実行したフレームで WebFOCUS ビューアを開きます。アクセシビリティを有効にしたレポートでは、これがデフォルト値です。

**注意：** SET ACCESSIBLE=508 に設定されている場合、WebFOCUS ビューアのデフォルト設定は WEBVIEWTARG=OFF になります。アクセシビリティの有効化についての詳細は、523 ページの「[ACCESSHTML](#)」および 33 ページの「[セクション 508 アクセシビリティ](#)」を参照してください。

### WEBVIEWTITLE

WEBVIEWTITLE パラメータは、WEBVIEWER インスタンスを含むウィンドウのタイトルテキストを定義します。

構文は次のとおりです。

```
SET WEBVIEWTITLE= 'text'
```

または

```
ON TABLE SET WEBVIEWTITLE 'text'
```

説明

*text*

Web ビューアウィンドウのタイトルテキストです。一重引用符 (') で囲みます。

## WEEKFIRST

WEEKFIRST パラメータは、曜日のいずれかを週の開始日に指定します。このパラメータは、HDIFF、HNAME、HPART、HYYWD、HSETPT 関数による週の演算で使用されます。

HPART および HNAME サブルーチンは、日付時間値から週番号を抽出します。週番号を特定するために、ISO 8601 標準の週番号を使用できます。ここでは、1 年の最初の週を 4 日以上の日付がある 1 月の最初の週として定義しています。1 月の最初の週の前に日付がある場合、前年の第 52 週または第 53 週に属することになります。

これらの関数は、WEEKFIRST の値に基づいて、1 年の最初の週を、7 日間の日付がある 1 月の第 1 週として定義することもできます。

WEEKFIRST パラメータは、週の開始日を指定しますが、それぞれの曜日に対応する日付は変更されません。

構文は次のとおりです。

```
SET WEEKFIRST = {value|7}
```

説明

*value*

次のいずれかの値を指定します。

標準外の週番号を使用して、日曜日から土曜日を 1 から 7 で表します。デフォルト値は 7 です。

または

ISO1 から ISO7 ISO 標準の週番号を使用して、日曜日から土曜日を表します。

**注意：**ISO は ISO2 と同義です。

ISO 標準は先頭の曜日を月曜日に設定するため、ISO 基準に完全に準拠するには、WEEKFIRST パラメータを ISO または ISO2 に設定する必要があります。

### WPMINWIDTH

WP フォーマットの出力ファイルを後から処理するために、バージョンが変更された場合でもレポート幅を固定した状態にするには、SET WPMINWIDTH コマンドを使用して幅を設定します。このパラメータでは、出力ファイルの最小幅を指定します。リクエストの出力ファイルに継承されたフィールドがこの設定幅に収まらない場合は、幅が自動的に広がります。レコードの長さは WPMIDWIDTH の値になります。

構文は次のとおりです。

```
SET WPMINWIDTH = {0|nnn}
```

```
ON TABLE SET WPMINWIDTH {0|nnn}
```

説明

nnn

出力ファイルの最小幅です。nnn に 0 (ゼロ) を指定した場合、幅はレポートリクエストに基づいて自動的に計算されます。出力ファイルに継承されたフィールドが、指定した幅に収まらない場合、それらのフィールドが収まるよう幅が自動的に拡張されます。

### XLSXLOCALZIP

XLSXLOCALZIP パラメータは、EXCEL07 (XLSX) ファイルの圧縮と解凍時に、ローカル C ベースの ZIP モードを有効にします。

XLSXLOCALZIP 設定には、次のことが適用されます。

- ☐ XLSX 出力フォーマットのみに適用されます。
- ☐ EXCELSERVURL 設定より優先されます。
- ☐ 次の機能では、サポートされません。
  - ☐ バーストおよび配信
  - ☐ グラフ
- ☐ PPTX 出力フォーマットでは、WebFOCUS Servlet および JSCOM3 モードが引き続き使用されます。

構文は次のとおりです。

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

説明

**ON**

EXCEL07 (XLSX) ファイルで、ローカル C ベースの ZIP モードを有効にします。

**OFF**

EXCEL07 (XLSX) ファイルで、ローカル C ベースの ZIP モードを有効にしません。デフォルト値は OFF です。

## XLSXPAGEBRKIGNORE

XLSXPAGEBRKIGNORE パラメータは、WebFOCUS フォーマットの XLSX レポート出力で、Excel の改ページと同じ場所に改ページを挿入するかどうかを制御します。

構文は次のとおりです。

**SET XLSXPAGEBRKIGNORE = {OFF|ON}**

説明

**OFF**

XLSX レポート出力フォーマットで、WebFOCUS の改ページと Excel の改ページを同期します。これがデフォルト値です。

**ON**

XLSX レポート出力フォーマットで、WebFOCUS の改ページと Excel の改ページを同期しません。この値は、以前のバージョンの動作に適合します。

## XRETRIEVAL

XRETRIEVAL パラメータは、実際のデータにアクセスせずにレポートのフォーマットをプレビューします。このパラメータを使用すると、TABLE、TABLEF、MATCH のいずれかのリクエストを実行し、レポートの処理を省略して HOLD マスターファイルを生成することができます。

構文は次のとおりです。

**SET XRETRIEVAL = {ON|OFF}**

説明

**ON**

レポートのプレビュー時にデータを取得します。デフォルト値は ON です。

**OFF**

データを取得しないことを指定します。

### YRTHRESH

YRTHRESH パラメータは、グローバルまたはフィールドレベルで世紀の開始日を定義します。このパラメータは、DEFCENT と併せて使用し、指定した値に基づいて現在の西暦を解析します。YRTHRESH 以上の 2 桁の年は、デフォルト世紀の値と見なされます。YRTHRESH より小さい 2 桁の年は、デフォルト世紀より 1 つ先の世紀の値と見なされます。詳細は、561 ページの「[DEFCENT](#)」を参照してください。

**注意：**マスターファイルに FDEFCENT および FYRTHRESH 属性を追加した場合でも、同一の結果を得ることができます。

構文は次のとおりです。

```
SET YRTHRESH = {[-]yy|0}
```

説明

[yy](#)

世紀 (100 年) に対するしきい値です。デフォルト値は 0 (ゼロ) です。

yy が正の値の場合、その値が世紀の開始日になります。しきい値以上の 2 桁の年は、デフォルト世紀の値と見なされます。しきい値より小さい 2 桁の年は、デフォルト世紀より 1 つ先の世紀の値と見なされます。

yy が負の値 (-yy) の場合、その値が現在の年から差し引かれて世紀の開始日となり、デフォルト世紀が自動的に計算されます。開始日は、それぞれ連続する年のはじめに 1 日ずつ増加されます。

# 9

## TIBCO WebFOCUS ファイルの定義と割り当て

---

WebFOCUS では、さまざまな種類のファイルを使用します。これらのファイルには、開発者が作成するアプリケーションファイル、アプリケーションのリクエストに基づいて WebFOCUS が作成する抽出ファイル、アプリケーションの要求に基づいて WebFOCUS が作成する作業ファイルがあります。

ここでは、WebFOCUS ファイル割り当ての要件、およびこの機能をサポートするオペレーティングシステムで論理名 (ddname) をファイルまたはデバイスに割り当てる方法について説明します。

### トピックス

- ❑ [TIBCO WebFOCUS ファイルの割り当て](#)
  - ❑ [Windows 環境でのアプリケーションファイル](#)
  - ❑ [Windows 環境での抽出ファイル](#)
  - ❑ [Windows 環境での作業ファイル](#)
  - ❑ [Windows 環境でのファイルの有無の特定](#)
  - ❑ [UNIX 環境でのアプリケーションファイル](#)
  - ❑ [UNIX 環境での抽出ファイル](#)
  - ❑ [UNIX 環境での作業ファイル](#)
  - ❑ [UNIX 環境でのファイルの有無の特定](#)
- 

### TIBCO WebFOCUS ファイルの割り当て

WebFOCUS ファイルは、「アプリケーションファイル」、「抽出ファイル」、「作業ファイル」という 3 つのカテゴリに分類されます。ここでは、Windows 環境、UNIX 環境で使用するファイル、およびこれらのファイルの定義方法について説明します。

いくつかのファイルは WebFOCUS が自動的に割り当てますが、ユーザが明示的にファイルおよびその格納場所を定義しなければならない場合があります。FILEDEF コマンドにより、アプリケーション名を使用した FILEDEF 構文が作成され、移植先プラットフォームのいずれにも依存しないファイルパスが生成されます。

**注意：**FILEDEF コマンド内のフィールド名またはディレクトリ名にブランクを使用するには、パス全体を二重引用符 (") で囲む必要があります。以下はその例です。

FILEDEF TEMPFILE DISK "C:¥dir1¥dir2¥directory space¥tempfle.ftm"

構文 Windows 環境でのファイルの参照

filename.filetype

説明

filename

ファイルの名前です。

filetype

ファイルのタイプです。これは、FAT ファイルシステムに基づいて Windows 環境で使用する 3 文字の拡張子です。

参照 Windows および UNIX 環境での TIBCO WebFOCUS ファイル

下表は、Windows および UNIX 環境で使用する WebFOCUS ファイルを示しています。WebFOCUS は、ファイル拡張子に基づいてファイルタイプを識別します。

アプリケーションファイル

拡張子	説明
.mas	マスターファイル
.acx	アクセスファイル
.fex	プロシジャ
.foc	FOCUS データソースおよび外部インデックス
.dat	シーケンシャルデータソース

拡張子	説明
<code>.sty</code>	WebFOCUS スタイルシートファイル
<code>.err</code>	エラーメッセージファイルまたはヘルプテキスト
<code>.exe or .dll</code>	Windows 環境での関数ライブラリ
<code>.prf</code>	プロファイル
<code>.htm</code> <code>.html</code> <code>.jpeg</code> <code>.gif</code> <code>.css</code> <code>.js</code> <code>.class</code> <code>.jar</code>	Web ブラウザに表示するファイル

### 抽出ファイル

ファイル	説明
<code>HOLD</code>	HOLD コマンドで保存するデータを格納します。
<code>SAVB</code>	SAVB コマンドで保存するデータを格納します。
<code>SAVE</code>	SAVE コマンドで保存するデータを格納します。
<code>HOLDMAST</code>	HOLD ファイルに使用する一時マスターファイルです。
<code>.FTM</code>	HOLD、SAVB、SAVE コマンドで保存するデータを格納します。

**注意：**ダイアログマネージャの出力ファイルは、FILEDEF コマンドを使用してシステムプロファイルまたはユーザプロファイルで割り当てる必要があります。

## 作業ファイル

ファイル	説明
FOCSTACK	定義済みのダイアログマネージャプロシジャを格納します。
FOCSORT	ソート時に使用します。
FOCPOST	POST コマンドで保存したシーケンシャル出力ファイルです。このファイルを読み戻すには、PICKUP コマンドを使用します。
FOCSML	FML (Financial Modeling Language) で使用します。
OFFLINE	SET PRINT パラメータが OFFLINE に設定されている場合に使用します。
SYSIN	入力を指示します。
SYSPRINT	出力を画面に移動します。

## Windows および UNIX 環境でのファイルの動的定義

アプリケーションファイルの大部分は、ファイルを明示的に定義しなくてもファイルを参照することができます。WebFOCUS は、特定のアプリケーションファイルを動的に割り当てます。また、WebFOCUS は、セッション中にすべての抽出ファイルおよび一時作業ファイルを動的に定義し、オペレーティングシステムに割り当てます。

WebFOCUS は、次の抽出ファイル、出力ファイル、作業ファイルを定義します。

- ☐ HOLD、SAVB、SAVE ファイル。
- ☐ FOCUS データソース。
- ☐ FOCSORT ファイル。
- ☐ キーボードから入力する SYSIN。SYSIN は、TERM に割り当てられます。これは、FILEDEF SYSIN TERM コマンドに相当します。
- ☐ 画面表示用に出力する SYSPRINT。SYSPRINT は、TERM に割り当てられます。これは、FILEDEF SYSPRINT TERM に相当します。

- ❑ FOCSTACK.FTM ファイルへの FOCSTACK。これは、FILEDEF FOCSTACK DISK FOCSTACK.FTM コマンドに相当します。FOCSTACK についての詳細は、307 ページの「[アプリケーションの制御フロー管理](#)」を参照してください。

## FILEDEF コマンドによる論理名の割り当て

固定長ファイルやカンマ区切りデータファイルなど、オペレーティングシステムが管理するファイルの場合、オペレーティングシステムで使用する実際のファイル名が物理ファイル名です。この物理ファイル名を指す省略名が論理名 (ddname) です。長い物理ファイル名の代わりに短い論理名を使用することにより、コーディングが簡略化されます。

FILEDEF コマンドは、物理ファイル名に論理名を割り当て、ファイル属性を指定します。FILEDEF コマンドを使用すると、WebFOCUS に対してファイルおよびその場所を明示的に定義することができます。これにより、アプリケーション名を使用した FILEDEF 構文が作成され、移植先プラットフォームのいずれにも依存しないファイルパスが生成されます。FILEDEF コマンドはプロシジャで発行することができますが、発行したコマンドは 1 つのリクエストでのみ有効になります。

プロシジャごとにファイルの割り当てを含めるのではなく、すべての FILEDEF コマンドを 1 つのファイルに格納しておき、各プロシジャの開始位置で、-INCLUDE コマンドを使用してそのファイルを呼び出すことをお勧めします。これにより、プロシジャごとにファイル割り当て情報を変更する代わりに、FILEDEF コマンドをグローバルに変更することが可能になります。

FILEDEF コマンドをサポートするオペレーティングシステムでは、このコマンドは一般的に次の場合に使用します。

- ❑ **データソースの指定** 使用するすべてのデータソースに対して、FILEDEF 割り当て (FOCUS データソースの場合は USE ディレクトリ) を作成する必要があります。
- ❑ **入出力のリダイレクト** 入出力には、3 つの ddname が使用されます。これらの ddname を再割り当てすることにより、入出力をリダイレクトすることができます。
  - ❑ 入力用の SYSIN
  - ❑ 画面表示用として出力する SYSPRINT

また、UNC (Universal Naming Convention) を使用して、サーバに格納されたファイルに論理名を割り当てることもできます。UNC を利用するには、使用するサーバに最初に接続しておく必要があります。サーバへの接続またはネットワークドライブの割り当てについての詳細は、ネットワーク管理者に問い合わせてください。

## 構文 FILEDEF コマンドによる論理名の割り当て

```
FILEDEF ddname DISK appname[/appnamea...]/filename [(APPEND) [LRECL n]
[RECFM F]
```

または

```
FILEDEF SYSIN TERM [LOWER]
```

または

```
FILEDEF ddname PRINTER
```

説明

### ddname

ファイル、入力、出力に使用する論理名です。ddname として許容される長さは、1 から 8 バイトです。ddname を使用してデータソースをマスターファイルに関連付ける場合は、ddname をマスターファイルの名前に一致させる必要があります。

### DISK

指定した ddname をファイルに関連付けます。

### appname[/appnamea...]

APPROOT 下で物理ファイルを格納するアプリケーション名、またはネストされたアプリケーション名です。

### filename

appname 下のファイルの物理名です。

### APPEND

ファイルの末尾にレコードを追加します。このオプションを指定しない場合、ファイルは上書きされます。

### LRECL n

レコード長を指定します。n は整数です。

ファイルが SAVB ファイル、またはデータ管理で使用するバイナリ値を含む固定フォーマットのトランザクションファイル (FIXFORM ファイル) の場合、LRECL の値を指定する必要があります。

### RECFM F

固定長のレコードを指定します。

ファイルが SAVB ファイル、またはデータ管理で使用するバイナリ値を含む固定フォーマットのトランザクションファイル (FIXFORM ファイル) の場合、RECFM の値を指定する必要があります。

**TERM**

入力ソースおよび出力先としてキーボードとモニタを指定します。

**LOWER**

キーボードからの入力を、入力されたとおりに WebFOCUS に送信します。

デフォルトの名前の規則に準拠していない FOCUS データソース (拡張子が .foc) は、FILEDEF コマンドではなく、USE コマンドで指定します。USE コマンドについての詳細は、475 ページの「[FOCUS データソースへのアクセス](#)」を参照してください。

**例 サーバ上のファイルへの論理名の割り当て**

サーバに存在するファイルに論理名を割り当てるには、UNC (Universal Naming Convention) を使用します。

```
FILEDEF DATFILE DISK ¥¥SERVER2¥DISK1¥MAYSPLES.DAT
```

**例 検索パスの設定**

検索パス上のすべてのディレクトリを対象として NEW\_EMPS.DAT ファイルを検索するには、次のコマンドを発行します。

```
FILEDEF SYSIN DISK *:NEW_EMPS.DAT
```

**例 他のコンテンツへのレポート抽出の追加**

LIBRARY データソースから抽出したレポートを LIB03.FTM ファイルの現在のコンテンツに追加するには、次のコマンドを発行します。

```
FILEDEF SAVE DISK C:¥LIBRARY¥LIB03.FTM (APPEND
```

**例 LRECL による標準テキストエディタファイルの読み取り**

リクエストには、LRECL の暗黙の長さ以上の LRECL を指定することができます。たとえば、テキストファイルに存在する行の最大長が 7 バイトの場合は、次のコマンドを発行します。

```
FILEDEF BIGLINE DISK BIGLINE.FTM (LRECL 7
```

**FILEDEF で割り当てた現行 ddname の表示**

? FILEDEF コマンドは、さまざまなファイル、入力、出力で割り当てた ddname を表示します。

### 構文 現行 ddname の表示

? FILEDEF

### 例 現行 ddname の表示

次のコマンドを発行します。

? FILEDEF

次のような情報が生成されます。

Lname	Device	Lrecl	Recfm	Append	Expl	Filename
HOLD2	DISK	0	V	N	Y	C:¥VM¥SMALL¥HOLD2.FTM

### ファイル割り当てのクリア

ファイルの割り当ては、WebFOCUS 構文でクリアすることができます。

### 構文 構文による論理名のクリア

FILEDEF ddname CLEAR

説明

ddname

論理名です。この名前として許容される長さは、1 から 8 バイトです。

CLEAR

指定した論理名をクリアします。

## Windows 環境でのアプリケーションファイル

WebFOCUS を使用して、アプリケーションファイルの参照および検索を実行することができます。

デフォルトのファイルタイプを変更しない場合は、FILEDEF コマンドを発行したり、他の方法を使用したりしなくてもレポートを作成することができます。FOCUS データソースのデフォルトのファイルタイプを変更する場合、またはデータソースが標準の検索パスに存在しない場合は、USE コマンドを発行する必要があります。USE コマンドについての詳細は、475 ページの「[FOCUS データソースへのアクセス](#)」を参照してください。

## Windows 環境でのマスターファイル

マスターファイルにはデータソースに関するメタデータが格納されています。マスターファイルは、ファイル名と拡張子で識別します (例、filename.mas)。マスターファイルは、データソースを定義するさまざまな属性で構成されます。通常、マスターファイルの名前およびマスターファイルで定義するデータソースの名前は一致します。マスターファイルを構成するレコードは、80 バイトの固定長にする必要があります。

データソースを参照する場合は、そのデータソースの定義およびデータソースのクロスリファレンスとなるすべてのファイルが、使用できる状態でなければなりません。

## Windows 環境でのマスターファイルの場所の特定

WebFOCUS は、EDAPATH または APP PATH ロジックのいずれかを使用して、WebFOCUS Reporting Server の検索パス上にあるマスターファイルを検索します。

**注意：**WebFOCUS は、EDAPATH または APP PATH を検索する前に、HOLD マスターファイルが作成された現在の一時ディレクトリを検索します。

## 参照

### Windows 環境でのマスターファイル名の規則

次の規則が適用されます。

- ❑ ファイル名の最大長は 64 バイトです。名前の先頭には文字を使用する必要があります。それ以外は、文字、数字、アンダースコア (\_) を任意に組み合わせて使用することができます。他の特殊文字も使用できますが、問題の発生原因となる可能性があるため、使用しないことをお勧めします。
- ❑ デフォルト設定では、WebFOCUS はファイル拡張子に基づいてマスターファイルを検索するため、拡張子は常に .mas にする必要があります。

## Windows 環境でのアクセスファイル

データソースの中には、アクセスファイルを使用してマスターファイルを補足するものもあります。アクセスファイルには、データソース記述を補足する追加情報が記述されています。これには、データソースのフルネームやフルパス名などがあります。データソースを記述するには、マスターファイルが 1 つと、データソースによってはアクセスファイルが 1 つ必要です。

WebFOCUS は、EDAPATH または APP PATH ロジックのいずれかを使用して、WebFOCUS Reporting Server の検索パス上にあるアクセスファイルを検索します。

## 参照 Windows 環境でのアクセスファイル名の規則

次の規則が適用されます。

- ❑ ファイル名の最大長は 64 バイトです。名前の先頭には文字を使用する必要があります。それ以外は、文字、数字、アンダースコア (\_) を任意に組み合わせて使用することができます。他の特殊文字も使用できますが、問題の発生原因となる可能性があるため、使用しないことをお勧めします。
- ❑ デフォルト設定では、WebFOCUS はファイル拡張子に基づいてアクセスファイルを検索するため、拡張子は常に .acx にする必要があります。
- ❑ アクセスファイルの名前は、それに対応するマスターファイルの名前と一致させる必要があります。

## Windows 環境でのプロシジャ

プロシジャは、実行可能なテキストファイルで、この中にレポートリクエストが格納されています。1 行につき最大で 3024 バイトを記述することができます。プロシジャは、ファイル名と拡張子で識別します (例、filename.fex)。

WebFOCUS は、EDAPATH または APP PATH ロジックのいずれかを使用して、WebFOCUS Reporting Server の検索パス上にあるプロシジャを検索します。

## 参照 Windows 環境でのプロシジャ名の規則

次の規則が適用されます。

- ❑ ファイル名の最大長は 64 バイトです。名前の先頭には文字を使用する必要があります。それ以外は、文字、数字、アンダースコア (\_) を任意に組み合わせて使用することができます。他の特殊文字も使用できますが、問題の発生原因となる可能性があるため、使用しないことをお勧めします。
- ❑ デフォルト設定では、WebFOCUS はファイル拡張子に基づいてプロシジャを検索するため、拡張子は常に .fex にする必要があります。

## Windows 環境での TIBCO FOCUS データソース

FOCUS データソースには、FOCUS フォーマットで記述したデータが格納されています。すべての FOCUS データソースは、レコード長が 4096 バイトで固定長レコードのフォーマットで統一されます。

FOCUS データソースは、ファイル名と拡張子で識別します (例、filename.foc)。通常、FOCUS データソースの名前とそれに対応するマスターファイルの名前は一致します。たとえば、マスターファイルの名前が ledger.mas の場合、FOCUS データソースの名前は ledger.foc になります。このデフォルト設定は、USE コマンドを使用して変更することができます。詳細は、475 ページの「[FOCUS データソースへのアクセス](#)」を参照してください。

WebFOCUS は、WebFOCUS Reporting Server または SUB Server が常駐するプラットフォームで FOCUS データソースを検索します。

## 参照 Windows 環境での TIBCO FOCUS データソース名の規則

次の規則が適用されます。

- ❑ ファイル名の最大長は 64 バイトです。名前の先頭には文字を使用する必要があります。それ以外は、文字、数字、アンダースコア (\_) を任意に組み合わせて使用することができます。他の特殊文字も使用できますが、問題の発生原因となる可能性があるため、使用しないことをお勧めします。
- ❑ デフォルト設定では、WebFOCUS はファイル拡張子に基づいて FOCUS データソースを検索するため、拡張子は常に .foc にする必要があります。

## Windows 環境での TIBCO FOCUS データソースの外部インデックス

外部インデックスは、指定した 1 つまたは複数の FOCUS データソースのインデックス、フィールド、セグメント情報で構成される FOCUS データソースです。外部インデックスは、それに関連付けられた FOCUS データソースから独立して存在し、データ取得時のパフォーマンス向上を目的として使用されます。Windows では、REBUILD を使用して外部インデックスを作成すると、自動的に永続ファイルとして割り当てられます。

## Windows 環境でのサポート対象データソース

WebFOCUS は、数多くのデータソースタイプをサポートします。

WebFOCUS は、WebFOCUS Reporting Server または SUB Server が常駐するプラットフォームでデータソースを検索します。

## 参照 Windows 環境でのサポート対象データソース名の規則

WebFOCUS は、標準のデータソース名の付け方の規則をサポートします。

## Windows 環境でのシーケンシャルデータソース

WebFOCUS は、次のフォーマット設定のシーケンシャルデータソースを認識します。

- ❑ 固定フォーマット - 各フィールドはレコード内で定義された位置に配置されます。
- ❑ 可変長フォーマット - カンマ区切りフォーマットとも呼ばれますが、フィールドはレコード内の任意の位置に配置することができます。

## Windows 環境での WebFOCUS スタイルシートファイル

WebFOCUS スタイルシートを使用してレポートのスタイルを設定し、重要な情報を強調するレポートを作成することができます。スタイルシートを使用することにより、1つのレポートにさまざまなスタイル特性を持たせたり、個々のレポートコンポーネントに異なるスタイルを設定することができます。また、任意のレポートオブジェクトからハイパーリンクを定義することも可能です。スタイルシートは、外部で作成したり、レポートリクエスト内で作成したりすることができます。

なお、WebFOCUS は、EDAPATH または APP PATH ロジックのいずれかを使用して、WebFOCUS Reporting Server の検索パス上にある WebFOCUS スタイルシートを検索します。

## 参照

### Windows 環境でのスタイルシートファイル名の規則

次の規則が適用されます。

- ❑ ファイル名の最大長は 64 バイトです。名前の先頭には文字を使用する必要があります。それ以外は、文字、数字、アンダースコア ( \_ ) を任意に組み合わせて使用することができます。他の特殊文字も使用できますが、問題の発生原因となる可能性があるため、使用しないことをお勧めします。
- ❑ デフォルト設定では、WebFOCUS はファイル拡張子に基づいてスタイルシートを検索するため、拡張子は常に .sty にする必要があります。

## Windows 環境での関数ライブラリ

外部関数ライブラリは、WebFOCUS 以外の場所に格納されています。

## Windows 環境でのプロファイル

WebFOCUS は、次のような異なるレベルのプロファイルをサポートするため、柔軟にアプリケーションを設計、実行することができます。

- ❑ **グローバルプロファイル** サーバのインストールおよび構成時に自動的に作成される起動ファイルです。グローバルプロファイルには、サーバの処理を正しく実行するために必要なデフォルト環境の設定が格納されています。
- ❑ **ユーザプロファイル** ユーザ ID に基づいて作成されます。サーバは、ユーザプロファイルを使用してサーバ環境に適用する設定を指定しますが、この設定は特定のユーザ ID に対してのみ適用されます。

## Windows 環境での WebFOCUS の Web ページ

Web ページおよびその関連ファイルは、アプリケーションのプレゼンテーションロジックを提供します。これらのファイルは、ユーザのブラウザに表示されます。たとえば、HTML ファイル、グラフィカルイメージ、Java クラスおよびアーカイブファイル、JavaScript ファイル、CSS ファイルがあります。

Web サーバが Web ページまたはその関連ファイルを検索する場合、Web サーバのホームディレクトリまたは Web サーバエイリアスにあるファイルを検索します。HTMLFORM で呼び出す HTML ファイルは、WebFOCUS Reporting Server で定義したパスに存在していなければなりません。

### 参照 Windows 環境での Web ページ名の規則

WebFOCUS では、標準の名前規則がサポートされます。Web ページには、拡張子の .HTM を付ける必要があります。

## Windows 環境での抽出ファイル

WebFOCUS は、すべての抽出ファイルを WebFOCUS Reporting Server 上の一時ディレクトリ、または WebFOCUS Reporting Server 上でユーザが定義した場所に作成します。

### 参照 Windows 環境での抽出ファイルの場所の特定

WebFOCUS は、EDASERVE.CFG ファイルの EDATEMP 環境変数で指定したディレクトリ内の一時ディレクトリにすべての抽出ファイルを作成します。

## Windows 環境での HOLD ファイル

HOLD 抽出ファイルは、レポートリクエストの結果を格納したシーケンシャルデータソースです。このファイルに対応して、HOLD マスターファイルが作成される場合があります。

FORMAT オプションで拡張子を変更しない限り、抽出ファイルの拡張子は .ftm です。マスターファイルが作成された場合は、抽出ファイルと同一の名前に、拡張子の .mas が付けられます。

HOLD に FORMAT FOCUS オプションを使用すると、WebFOCUS は抽出ファイルとマスターファイルを作成し、それぞれに「FOC\$HOLD」という名前を付けます。次に、これらのファイルがプロシジャの入力用に使用され、最終の FOCUS ファイルが作成されます。新しい FOCUS ファイルおよびマスターファイルは、ユーザの一時ディレクトリに作成されます。

### 構文

#### Windows 環境での HOLD ファイルの作成

```
ON TABLE HOLD [AS filename]
```

説明

**filename**

HOLD ファイルの名前です。ファイル名を指定しない場合、デフォルト名の「HOLD」が使用されます。HOLD コマンドを繰り返し使用する場合、最新の HOLD コマンドが前回の HOLD ファイルを上書きするため、リクエストごとに異なるファイル名を記述し、抽出したデータをそれぞれ個別のファイルに保存すると、前回のファイルの上書きを防止することができます。

## Windows 環境での SAVB ファイル

SAVB ファイルは、レポートリクエストの結果を格納する抽出ファイルです。この場合、すべての数値フィールドはバイナリで格納され、すべての文字フィールドには 4 バイトごとにブランクが追加されます。このファイルを表示することはできません。

SAVB が作成されると、WebFOCUS はマスターファイルを作成しません。

### 構文

#### Windows 環境での SAVB ファイルの作成

```
ON TABLE SAVB [AS filename]
```

説明

**filename**

ファイルの名前です。デフォルト値は SAVB です。デフォルトの拡張子は .ftm です。

## Windows 環境での SAVE ファイル

SAVE ファイルは、レポートのデータを保存するための抽出ファイルですが、見出しまたは中間合計の保存およびマスターファイルの作成は実行されません。このファイルは、単純なシーケンシャル文字データファイルのため、他のプログラムで使用したり、データ管理リクエストで他のデータファイルに統合したりすることができます。デフォルトのフォーマットは単純文字ですが、他のさまざまなソフトウェア製品に対応したフォーマットを指定することができます。WebFOCUS は、レポートのすべての列を表示可能な文字フォーマットで保存します。このとき、列間にはブランクは挿入されません。

SAVE ファイルには、SAVB と同等の外部文字フォーマットが含まれています。このコマンドの構文および割り当ては、SAVB と同一です。ただし、数字は表示可能な文字フォーマットで、ブランクは挿入されません。

### 構文 Windows 環境での SAVE ファイルの作成

```
ON TABLE SAVE [AS filename]
```

説明

**filename**

ファイルの名前です。デフォルト値は SAVE です。デフォルトの拡張子は .ftm です。

## Windows 環境での HOLDMAST ファイル

HOLDMAST ファイルは、一時マスターファイルです。このファイルは、HOLD コマンドで作成されます。作成先は、APP HOLDMETA コマンドで指定します。

### 構文 Windows 環境での一時マスターファイルの場所の指定

```
APP HOLDMETA appname
```

説明

**appname**

有効なアプリケーション名です。

### 例 Windows 環境での一時マスターファイルの場所の指定

次の例は、一時マスターファイルの場所を指定し、物理データソースに論理名を割り当てる方法を示しています。

```
1. APP HOLDMETA app1
2. FILEDEF MKTSALES DISK C:¥TMP¥MKTSALES.FTM
 TABLE FILE GGSales
 SUM DOLLARS BY REGION
 ON TABLE HOLD AS MKTSALES FORMAT ALPHA
 END
```

コード内の番号は、次の説明の番号に対応しています。

1. マスターファイルの場所を app1 アプリケーションディレクトリに指定します。  
WebFOCUS は、2 行目で指定する論理名 MKTSALES と同一の名前を一時マスターファイルに割り当てます。
2. 物理データソースの論理名として MKTSALES を指定します。HOLD コマンドが、一時マスターファイルの mktsales.mas およびデータソースの mktsales.ftm を作成します。

## Windows 環境での作業ファイル

WebFOCUS は、アプリケーションの必要に応じて、作業ファイルを作成します。WebFOCUS は、すべての抽出ファイルを WebFOCUS Reporting Server 上の一時ディレクトリ、または WebFOCUS Reporting Server 上でユーザが定義した場所に作成します。

利用可能な作業ファイルには次のものがあります。

- ☐ **FOCSTACK** このファイルは、定義済みのダイアログマネージャプロシジャを格納します。
- ☐ **FOCSORT** このファイルは、FOCSORT.FOC として割り当てられ、TABLE、TABLEF、MATCH コマンドでソートを実行する際に必要となります。
- ☐ **FOCPOST** このファイルは、シーケンシャル出力を格納します。このファイルの保存には POST コマンド、読み出しには PICKUP コマンドを使用します。
- ☐ **FOCSML** このファイルは、FML (Financial Modeling Language) 機能に使用します。

## Windows 環境でのファイルの有無の特定

STATE コマンドを使用すると、システム変数の &RETCODE に、指定したファイルが存在するかどうかを示す値が設定されます。システム変数の &EXITRC の値も、指定したファイルが存在するかどうかを示します。

### 構文 Windows 環境でのファイルの有無の特定

[opsys] STATE filename

## 説明

### opsys

オペレーティングシステムを特定します。Windows の場合、この値は WINNT または DOS です。

### filename

存在するかどうかを特定するファイルです。現行パス上のファイル名またはフルパス名を指定するか、アプリケーション名とファイル名を appname/filename 形式で指定することができます。

コマンドの発行後、&RETCODE 変数に格納される値は、ファイルが存在する場合は 0 (ゼロ)、ファイルが存在しない場合は -1 になります。&EXITRC 変数に格納される値は、ファイルが存在する場合は 0 (ゼロ)、ファイルが存在しない場合は 2 になります。ファイルが存在しない場合、次のメッセージが生成されます。

File doesn't exist

## UNIX 環境でのアプリケーションファイル

ここでは、WebFOCUS がアプリケーションファイルを参照および検索する方法について説明します。

デフォルトのファイルタイプを変更しない場合は、FILEDEF コマンドを発行したり、他の方法を使用したりしなくてもレポートを作成することができます。FOCUS データソースのデフォルトのファイルタイプを変更する場合、またはデータソースが標準の検索パスに存在しない場合は、USE コマンドを発行する必要があります。USE コマンドについての詳細は、475 ページの「[FOCUS データソースへのアクセス](#)」を参照してください。

## UNIX 環境でのマスターファイル

マスターファイルにはデータソースに関するメタデータが格納されています。マスターファイルは、ファイル名と拡張子で識別します (例、filename.mas)。マスターファイルは、データソースを定義するさまざまな属性で構成されます。通常、マスターファイルの名前およびマスターファイルで定義するデータソースの名前は一致します。マスターファイルを構成するレコードは、80 バイトの固定長にする必要があります。

データソースを参照する場合は、そのデータソースの定義およびデータソースのクロスリファレンスとなるすべてのファイルが、使用できる状態でなければなりません。

WebFOCUS は、EDAPATH または APP PATH ロジックのいずれかを使用して、WebFOCUS Reporting Server の検索パス上にあるマスターファイルを検索します。

## 参照 UNIX 環境でのマスターファイル名の規則

次の規則が適用されます。

- ❑ ファイル名の最大長は 64 バイトです。名前の先頭には文字を使用する必要があります。それ以外は、文字、数字、アンダースコア ( \_ ) を任意に組み合わせて使用することができます。他の特殊文字も使用できますが、問題の発生原因となる可能性があるため、使用しないことをお勧めします。
- ❑ デフォルト設定では、WebFOCUS はファイル拡張子に基づいてマスターファイルを検索するため、拡張子は常に .mas にする必要があります。

## UNIX 環境でのアクセスファイル

データソースの中には、アクセスファイルを使用してマスターファイルを補足するものもあります。アクセスファイルには、データソース記述を補足する追加情報が記述されています。これには、データソースのフルネームやフルパス名などがあります。データソースを記述するには、マスターファイルが 1 つと、データソースによってはアクセスファイルが 1 つ必要です。

WebFOCUS は、EDAPATH または APP PATH ロジックのいずれかを使用して、WebFOCUS Reporting Server の検索パス上にあるアクセスファイルを検索します。

## 参照 UNIX 環境でのアクセスファイル名の規則

次の規則が適用されます。

- ❑ ファイル名の最大長は 64 バイトです。名前の先頭には文字を使用する必要があります。それ以外は、文字、数字、アンダースコア ( \_ ) を任意に組み合わせて使用することができます。他の特殊文字も使用できますが、問題の発生原因となる可能性があるため、使用しないことをお勧めします。
- ❑ デフォルト設定では、WebFOCUS はファイル拡張子に基づいて FOCUS データソースを検索するため、拡張子は常に .acx にする必要があります。

## UNIX 環境でのプロシジャ

プロシジャにはレポートリクエストが含まれます。各行に記述できる最大長は 80 バイトですが、制御文字を使用することはできません。プロシジャは、ファイル名と拡張子で識別します (例、filename.fex)。

プロシジャは、すべてのユーザが同時にアクセスし、実行することができます。サーバがプロシジャの場所を特定できるように、ファイル名と拡張子の前にデバイスおよびディレクトリを指定する必要があります。ディスクおよびディレクトリを指定する場合は、プロシジャを呼び出す際のファイル名とファイル拡張子 (.fex) の両方を含める必要があります。

WebFOCUS は、EDAPATH または APP PATH ロジックのいずれかを使用して、WebFOCUS Reporting Server の検索パス上にあるプロシジャを検索します。

## 参照

### UNIX 環境でのプロシジャ名の規則

次の規則が適用されます。

- ❑ ファイル名の最大長は 64 バイトです。名前の先頭には文字を使用する必要があります。それ以外は、文字、数字、アンダースコア (\_) を任意に組み合わせて使用することができます。他の特殊文字も使用できますが、問題の発生原因となる可能性があるため、使用しないことをお勧めします。
- ❑ デフォルト設定では、WebFOCUS はファイル拡張子に基づいてプロシジャを検索するため、拡張子は常に .fex にする必要があります。

## UNIX 環境での TIBCO FOCUS データソース

FOCUS データソースには、FOCUS フォーマットで記述したデータが格納されています。すべての FOCUS データソースは、レコード長が 4096 バイトで固定長レコードのフォーマットで統一されます。

FOCUS データソースは、ファイル名と拡張子で識別します (例、filename.foc)。通常、FOCUS データソースの名前とそれに対応するマスターファイルの名前は一致します。たとえば、マスターファイルの名前が ledger.mas の場合、FOCUS データソースの名前は ledger.foc になります。このデフォルト設定は、USE コマンドを使用して変更することができます。詳細は、475 ページの「[FOCUS データソースへのアクセス](#)」を参照してください。

WebFOCUS は、WebFOCUS Reporting Server または SUB Server が常駐するプラットフォームで FOCUS データソースを検索します。

## 参照

### UNIX 環境での TIBCO FOCUS データソース名の規則

次の規則が適用されます。

- ❑ ファイル名の最大長は 64 バイトです。名前の先頭には文字を使用する必要があります。それ以外は、文字、数字、アンダースコア (\_) を任意に組み合わせて使用することができます。他の特殊文字も使用できますが、問題の発生原因となる可能性があるため、使用しないことをお勧めします。

- ❑ デフォルト設定では、WebFOCUS はファイル拡張子に基づいて FOCUS データソースを検索するため、拡張子は常に .foc にする必要があります。

### UNIX 環境での TIBCO FOCUS データソースの外部インデックス

外部インデックスは、指定した 1 つまたは複数の FOCUS データソースのインデックス、フィールド、セグメント情報で構成される FOCUS データソースです。外部インデックスは、それに関連付けられた FOCUS データソースから独立して存在し、データ取得時のパフォーマンス向上を目的として使用されます。UNIX では、REBUILD を使用して外部インデックスを作成すると、自動的に永続ファイルとして割り当てられます。

### UNIX 環境でのシーケンシャルデータソース

WebFOCUS は、次のフォーマット設定のシーケンシャルデータソースを認識します。

- ❑ 固定フォーマット - 各フィールドは事前に定義されたレコード内の位置に格納されます。
- ❑ 可変長フォーマット - カンマ区切りフォーマットとも呼ばれますが、フィールドはレコード内の任意の位置に配置することができます。

### UNIX 環境での WebFOCUS スタイルシート

WebFOCUS スタイルシートを使用してレポートのスタイルを設定し、重要な情報を強調するレポートを作成することができます。スタイルシートを使用することにより、1 つのレポートにさまざまなスタイル特性を指定したり、それぞれのレポートコンポーネントに個別にスタイルを設定したりすることができます。また、スタイルシートを使用して、任意のレポートオブジェクトからハイパーリンクを定義することができます。

スタイルシートは、外部で作成することも、レポートリクエスト内で作成することも可能です。

WebFOCUS は、EDAPATH または APP PATH ロジックのいずれかを使用して、WebFOCUS Reporting Server の検索パス上にある WebFOCUS スタイルシートを検索します。

## 参照

### UNIX 環境でのスタイルシート名の規則

次の規則が適用されます。

- ❑ ファイル名の最大長は 64 バイトです。名前の先頭には文字を使用する必要があります。それ以外は、文字、数字、アンダースコア ( \_ ) を任意に組み合わせて使用することができます。他の特殊文字も使用できますが、問題の発生原因となる可能性があるため、使用しないことをお勧めします。

- デフォルト設定では、WebFOCUS はファイル拡張子に基づいてスタイルシートを検索するため、拡張子は常に .sty にする必要があります。

## UNIX 環境でのプロファイル

WebFOCUS は、次のような異なるレベルのプロファイルをサポートするため、柔軟にアプリケーションを設計、実行することができます。

- **グローバルプロファイル** サーバのインストールおよび構成時に自動的に作成される起動ファイルです。グローバルプロファイルには、サーバの処理を正しく実行するために必要なデフォルト環境の設定が格納されています。
- **サーバは、「グループプロファイル」と呼ばれるファイルを使用してサーバ環境に適用する設定を指定しますが、この設定は特定のセキュリティグループのユーザに対してのみ適用されます。**
- **ユーザプロファイル** ユーザ ID に基づいて作成されます。サーバは、ユーザプロファイルを使用してサーバ環境に適用する設定を指定しますが、この設定は特定のユーザ ID に対してのみ適用されます。

## UNIX 環境での抽出ファイル

WebFOCUS は、すべての抽出ファイルを WebFOCUS Reporting Server 上の一時ディレクトリ、または WebFOCUS Reporting Server 上でユーザが定義した場所に作成します。

## UNIX 環境での HOLD ファイル

HOLD 抽出ファイルは、レポートリクエストの結果を格納したシーケンシャルデータソースです。このファイルに対応して、HOLD マスターファイルが作成される場合があります。

FORMAT オプションで指定しない場合、HOLD ファイルの拡張子は .ftm です。マスターファイルが作成された場合は、抽出ファイルと同一の名前に、拡張子の .mas が付けられます。

HOLD に FORMAT FOCUS オプションを使用すると、WebFOCUS は抽出ファイルとマスターファイルを作成し、それぞれに「FOC\$HOLD」という名前を付けます。次に、これらのファイルがプロシジャの入力用に使用され、最終の FOCUS ファイルが作成されます。新しい FOCUS ファイルおよびマスターファイルは、ユーザの一時ディレクトリに作成されます。

## 構文 UNIX 環境での HOLD ファイルの作成

```
ON TABLE HOLD [AS filename]
```

### 説明

#### filename

HOLD ファイルの名前です。ファイル名を指定しない場合、デフォルト名の「HOLD」が使用されます。HOLD コマンドを繰り返し使用する場合、最新の HOLD コマンドが前回の HOLD ファイルを上書きするため、リクエストごとに異なるファイル名を記述し、抽出したデータをそれぞれ個別のファイルに保存すると、前回のファイルの上書きを防止することができます。

## UNIX 環境での SAVB ファイル

SAVB ファイルは、レポートリクエストの結果を格納する抽出ファイルです。この場合、すべての数値フィールドはバイナリで格納され、すべての文字フィールドには 4 バイトごとにブランクが追加されます。このファイルを表示することはできません。

### 構文      UNIX 環境での SAVB ファイルの作成

```
ON TABLE SAVB [AS filename]
```

### 説明

#### filename

ファイルの名前です。デフォルト値は SAVB です。デフォルトの拡張子は .ftm です。

## UNIX 環境での SAVE ファイル

SAVE ファイルは、レポートのデータを保存するための抽出ファイルですが、見出しまたは中間合計の保存およびマスターファイルの作成は実行されません。このファイルは、単純なシーケンシャル文字データファイルで、他のプログラムで使用したり、データ管理リクエストで他のデータファイルに統合したりすることができます。デフォルトのフォーマットは単純文字ですが、他のさまざまなソフトウェア製品に対応したフォーマットを指定することができます。WebFOCUS は、レポートのすべての列を表示可能な文字フォーマットで保存します。このとき、列間にはブランクは挿入されません。

SAVE ファイルには、SAVB と同等の外部文字フォーマットが含まれています。このコマンドのフォーマットと割り当ては、SAVB と同一です。ただし、数字は表示可能な文字フォーマットで、ブランクは挿入されません。

### 構文      UNIX 環境での SAVE ファイルの作成

```
ON TABLE SAVE [AS filename]
```

説明

`filename`

ファイルの名前です。デフォルト値は **SAVE** です。デフォルトの拡張子は **.ftm** です。

## UNIX 環境での HOLDMAST ファイル

HOLDMAST ファイルは、一時マスターファイルです。このファイルは、**HOLD** コマンドで作成されます。作成先は、**APP HOLDMETA** コマンドで指定します。

**HOLD** ファイルがデフォルト名の「**HOLD**」で作成された場合、**APP HOLDMETA** 構文と併せて **FILEDEF** コマンドを使用することで、**HOLD** ファイルおよび対応するマスターファイルに論理名を指定することができます。

## 構文 UNIX 環境での HOLDMAST ファイルの場所の指定

**APP HOLDMETA** `appname`

説明

`appname`

有効なアプリケーション名です。

## 例 UNIX 環境での HOLDMAST ファイルの論理名と場所の割り当て

次の例では、**FILEDEF** を使用して論理名を設定し、**HOLDMAST** ファイルの場所を指定します。

```
1. APP HOLDMETA app1
2. FILEDEF SALES DISK /tmp/sales.ftm
 TABLE FILE GGSALES
 PRINT GGSALES
 ON TABLE HOLD AS SALES
 END
```

コード内の番号は、次の説明の番号に対応しています。

1. マスターファイルの場所を **app1** アプリケーションディレクトリに指定します。  
WebFOCUS は、2 行目で指定する論理名 **SALES** と同一の名前を一時マスターファイルに割り当てます。
2. 物理データソースの論理名として **SALES** を指定します。**HOLD** コマンドが、一時マスターファイルの **mktsales.mas** およびデータソースの **mktsales.ftm** を作成します。

## UNIX 環境での作業ファイル

WebFOCUS は、アプリケーションの必要に応じて、作業ファイルを作成します。WebFOCUS は、すべての抽出ファイルを WebFOCUS Reporting Server 上の一時ディレクトリ、または WebFOCUS Reporting Server 上でユーザが定義した場所に作成します。

利用可能な作業ファイルには次のものがあります。

- ❑ **FOCSTACK** このファイルは、定義済みのダイアログマネージャプロセスを格納します。
- ❑ **FOCSORT** このファイルは、FOCSORT.FOC として割り当てられ、TABLE、TABLEF、MATCH コマンドでソートを実行する際に必要となります。
- ❑ **FOCPOST** このファイルは、シーケンシャル出力を格納します。このファイルの保存には POST コマンド、読み出しには PICKUP コマンドを使用します。
- ❑ **FOCSML** このファイルは、FML (Financial Modeling Language) 機能に使用します。

## UNIX 環境でのファイルの有無の特定

STATE コマンドを使用すると、システム変数の &RETCODE に、指定したファイルが存在するかどうかを示す値が設定されます。システム変数の &EXITRC の値も、指定したファイルが存在するかどうかを示します。

### 構文 UNIX 環境でのファイルの有無の特定

[opsys] STATE filename

説明

opsys

オペレーティングシステムを特定します。UNIX の場合、この値は UNIX です。

filename

存在するかどうかを特定するファイルです。現行パス上のファイル名またはフルパス名を指定するか、アプリケーション名とファイル名を appname/filename 形式で指定することができます。

コマンドの発行後、&RETCODE 変数に格納される値は、ファイルが存在する場合は 0 (ゼロ)、ファイルが存在しない場合は -1 になります。&EXITRC 変数に格納される値は、ファイルが存在する場合は 0 (ゼロ)、ファイルが存在しない場合は 2 になります。ファイルが存在しない場合、次のメッセージが生成されます。

`File doesn't exist`



# 10

## ユーロ通貨サポート

---

ここでは、通貨データソースを作成、使用して、新しいユーロ通貨と各国の現地通貨を換算する方法について説明します。

### トピックス

- ❑ [ユーロ通貨の統合](#)
  - ❑ [通貨の換算](#)
  - ❑ [通貨データソースの作成](#)
  - ❑ [通貨データを格納するフィールドの識別](#)
  - ❑ [通貨データソースの有効化](#)
  - ❑ [通貨データの処理](#)
  - ❑ [有効な通貨データソースのクエリ](#)
  - ❑ [数値区切り文字](#)
  - ❑ [拡張通貨記号の選択](#)
- 

### ユーロ通貨の統合

ユーロ通貨の導入により、ビジネスにおいては 2 種類の通貨で帳簿を付けることが要求され、データソースに新しいフィールドを追加し、新しいタイプの通貨換算を実行することが必要となります。通貨の換算は、欧州連合 (EU) が指定した規則に基づいて実行することができます。手順は次のとおりです。

1. 通貨データソースを作成し、使用する通貨の通貨 ID および為替レートを格納します。詳細は、691 ページの「[通貨データソースの作成](#)」を参照してください。
2. 作成したデータソースで、通貨データを表すフィールドを識別します。詳細は、694 ページの「[通貨データを格納するフィールドの識別](#)」を参照してください。
3. 使用する通貨データソースを有効にします。詳細は、696 ページの「[通貨データソースの有効化](#)」を参照してください。
4. 通貨の換算を実行します。詳細は、697 ページの「[通貨データの処理](#)」を参照してください。

## 通貨の換算

ユーロ通貨は、2002 年 1 月 1 日に加盟諸国で導入され、2002 年 7 月 1 日に単一の法定通貨になりました。現在、すべての金融取引はユーロ通貨で行われます。

EU は、12 加盟国の従来の現地通貨とユーロ通貨の間に固定為替レートを設定しました。EU では 12 以上の通貨がユーロを使用しますが、国際的には 100 以上の通貨が認定されています。また、異なる目的でカスタム通貨を定義しなければならない場合があります。

ユーロ通貨加盟諸国では固定為替レートを使用しますが、ユーロ加盟国とユーロ以外の国々との間の為替レートは自由に変動し、実際には、一度だけ使用するレートもあります (例、実質レートおよび予算レート)。

通貨コードおよび為替レートを識別するには、通貨データソースを作成します。詳細は、691 ページの「[通貨データソースの作成](#)」を参照してください。

## 参照

### 通貨換算の規則

EU により、通貨変換に関して、次のような規則が制定されています。

- ❑ 為替レートは、6 桁の有効数字で構成される 10 進数の値  $r$  で指定します。

この為替レートにより、各国の現地通貨とユーロとの間で次の関係が成立します。

1 ユーロ =  $r$  現地通貨単位

- ❑ ユーロから現地通貨に換算するには、ユーロに  $r$  を乗算し、その結果を小数点以下 2 桁に端数処理します。
- ❑ 現地通貨からユーロに換算するには、現地通貨を  $r$  で除算し、その結果を小数点以下 2 桁に端数処理します。
- ❑ 特定の国の現地通貨から別の現地通貨に換算するには、はじめに現地通貨からユーロに換算し、その結果を小数点以下 3 桁に端数処理します (使用するアプリケーションが正確に小数点以下 3 桁の数字に端数処理します)。次に、ユーロから別の現地通貨に換算し、その結果を小数点以下 2 桁の数字に端数処理します。この 2 段階のプロセスは「三角換算」と呼ばれます。

## 例 三角換算の実行

次の例は、三角換算を実行する方法を示しています。ここでは、10 米ドル (USD) をフランスフラン (FRF) に換算します。米ドル (USD) からユーロ (EUR) に換算する為替レートは 1.17249 で、フランスフラン (FRF) からユーロ (EUR) に換算する為替レートは 6.55957 です。

- ❑ 10 USD を EUR に換算するには、10 USD を EUR の為替レート 0.8840 で除算します。

$$\text{EUR} = 10 / 0.8840$$

結果は、11.3122 ユーロになります。

- ❑ EUR を FRF に換算するには、上記の結果に EUR に対する FRF の為替レート 6.55957 を乗算します。

$$\text{FRF} = 11.3122 * 6.55957$$

結果は、74.26 FRF になります。つまり、10 USD は 74.26 FRF に相当します。

## 通貨データソースの作成

必要な通貨の種類それぞれに対して、値を入力します。

通貨データソースの次の値を入力する必要があります。

- ❑ 通貨を識別する 3 文字のコード (例、米ドルは USD、ベルギーフランは BEF)。代表的な認定済み通貨コードを確認するには、693 ページの「[サンプル通貨コード](#)」を参照してください。
- ❑ 通貨に使用する 1 つまたは複数の為替レート。

通貨データソースには、必要に応じて任意の数の通貨を追加することができます。また、正式な通貨として定義できる通貨に制限はありません。そのため、使用するアプリケーションに応じて、通貨データソースを自由にカスタマイズすることができます。

通貨データソースは、使用するアプリケーションがアクセスできる限り、任意のデータソースタイプを選択することができます (例、FOCUS、FIX、DB2)。通貨マスターファイルには、各通貨 ID を識別するためのフィールドが 1 つと、為替レートを指定するための 1 つまたは複数のフィールドが必要になります。

通貨フィールドを既存のデータソースに追加するのではなく、通貨データ専用のデータソースを新しく作成することを強く推奨します。専用の通貨データソースを作成することにより、パフォーマンスが向上するとともに、リソースの負荷を最小限に抑えることができます。これは、通貨換算を実行する前に通貨データソースがメモリにロードされるためです。

## 構文 通貨データソースを作成するには

```
FILE = name, SUFFIX = suffix,$
FIELD = CURRENCY_ID,, FORMAT = A3, [ACTUAL = A3 ,]$
FIELD = rate_1,, FORMAT = {D12.6|numeric_format1}, [ACTUAL = A12,]$
.
.
FIELD = rate_n,, FORMAT = {D12.6|numeric_formatn}, [ACTUAL = A12,]$
```

### 説明

#### name

通貨データソースの名前です。

#### suffix

通貨データソースの接尾語です。通貨データソースは、アプリケーションがアクセスできる限り、任意のデータソースタイプを選択することができます。

#### CURRENCY\_ID

必須のフィールド名です。このフィールドには、各通貨を識別する 3 文字のコードが格納されます (例、米ドルは USD)。通貨 ID には、国際的に認定されたコードまたはユーザが定義したコードのいずれかを指定することができます。

**注意：**コードの EUR は自動的に認識されるため、このコードを通貨データソースに追加する必要はありません。共通の通貨コードの一覧は、693 ページの「[サンプル通貨コード](#)」を参照してください。

#### rate\_1...rate\_n

通貨の換算に使用するレートタイプです (例、BUDGET、FASB、ACTUAL)。各レートは、1 ユーロに相当する現地通貨単位です。

#### numeric\_format1...numeric\_formatn

為替レートの表示フォーマットです。すべてのフォーマットを数値フォーマットにする必要があります。このフォーマットには、D12.6 を使用することをお勧めします。これは、このフォーマットでは EU の換算規則で要求される 6 桁の有効数字で為替レートを表示できるためです。整数フォーマット (I) は使用しないでください。

#### ACTUAL An

FOCUS 以外のデータソースでのみ指定する必要があります。

**注意：**通貨データソースに追加するフィールド数の最大値は 255 です。つまり、CURRENCY\_ID フィールドが 1 つと最大で 254 個の通貨換算用フィールドです。

## 参照

## サンプル通貨コード

ユーロを採用している欧州諸国は、1999 年 1 月 1 日、ユーロと他の通貨間には為替レートを設定しました。この時点でユーロを採用していた国名には、アスタリスク (\*) が付いています。これは固定為替レートで、変動することはありません。ただし、他の国の為替レートは時間とともに変動します。

通貨名	通貨コード	レート
米国 ドル	USD	.974298
オーストリア シリング	ATS	13.7603
ベルギー フラン*	BEF	40.3399
イギリス ポンド	GBP	.625152
カナダ ドル	CAD	1.54504
デンマーク クローネ	DKK	7.42659
オランダ ギルダー*	NLG	2.20371
ドイツ マルク*	DEM	1.95583
ユーロ	EUR	1
フィンランド マルッカ	FIM	5.94573
フランス フラン*	FRF	6.55957
ギリシャ ドラクマ*	GRD	340.750
アイルランド ポンド*	IEP	0.787564
イタリア リラ*	ITL	1936.27
日本円または中国元	JPY	118.377
ルクセンブルグ フラン*	LUF	40.3399

通貨名	通貨コード	レート
ノルウェー クローネ	NOK	7.34864
ポルトガル エスクード*	PTE	200.482
スペイン ペセタ*	ESP	166.386
スウェーデン クローナ	SEK	9.20906
スイス フラン	CHF	1/4634

## 例

## マスターファイルでの通貨コードおよびレートの指定

次の例は、カンマ区切りの通貨データソースのマスターファイルで 2 つの通貨 (ACTUAL および BUDGET) を指定する方法を示しています。

```
FILE = CURRCODE, SUFFIX = COM,$
FIELD = CURRENCY_ID,, FORMAT = A3, ACTUAL = A3,$
FIELD = ACTUAL, ALIAS =, FORMAT = D12.6, ACTUAL = A12,$
FIELD = BUDGET, ALIAS =, FORMAT = D12.6, ACTUAL = A12,$
```

以下は、このマスターファイルで定義する通貨データソースのサンプルデータです。

```
FRF, 6.55957, 6.50000,$
USD, 0.974298, 1.00000,$
BEF, 40.3399, 41.00000,$
```

## 通貨データを格納するフィールドの識別

通貨データソースを作成した後、そのデータソースに通貨の値を表すフィールドを識別する必要があります。フィールドを通貨建ての値 (特定の通貨単位を表す値) として指定するには、次のいずれかに CURRENCY 属性を追加します。

- ☐ マスターファイルで指定するフィールド
- ☐ DEFINE または COMPUTE の左辺

## 構文 通貨の値の識別

次の構文を使用して、通貨建ての値を識別します。

### マスターファイル

```
FIELD = currfield,, FORMAT = numeric_format, ..., CURR =
{curr_id|codefield} ,$
DEFINE currfield/numeric_format CURR curr_id = expression ;$
DEFINE FILE filename
currfield/numeric_format CURR curr_id = expression ;
END
COMPUTE currfield/numeric_format CURR curr_id = expression ;
```

### マスターファイルの DEFINE

```
DEFINE currfield/numeric_format CURR curr_id = expression ;$
```

### DEFINE FILE コマンド

```
DEFINE FILE filename
currfield/numeric_format CURR curr_id = expression ;
END
```

### COMPUTE コマンド

```
COMPUTE currfield/numeric_format CURR curr_id = expression ;
```

### 説明

#### currfield

通貨建てフィールドの名前です。

#### numeric\_format

数値フォーマットです。換算の対象となる通貨建てにより、小数点以下の桁数は 2 または 0 (ゼロ) にすることをお勧めします。フォーマットの I または F は使用しないでください。

#### CURR

フィールドの値が、通貨建ての値であることを示します。CURR は、完全な属性名である CURRENCY の省略形です。

#### curr\_id

フィールドに関連付けられた 3 文字の通貨 ID です。通貨換算を実行するには、この ID の値を EUR にするか、通貨データソースの CURRENCY\_ID の値にする必要があります。

**codefield**

必要な場合に指定するフィールドの修飾名で、currfield に関連付けられた通貨 ID を格納します。このコードフィールドには、A3 以上の長さのフォーマットを指定する必要があります。このフィールドの最初の 3 バイトには、通貨 ID が格納されていると解釈されます。以下はその例です。

```
FIELD = PRICE,, FORMAT = P12.2C, ..., CURR = TABLE.FLD1,$
.
.
.
FIELD = FLD1,, FORMAT = A3, ..., $
```

フィールド名 FLD1 には、フィールド名 PRICE の通貨 ID が格納されます。

**filename**

このフィールドを定義するファイルの名前です。

**expression**

有効な式です。

**例****通貨建てフィールドの識別**

次の例は、「PRICE」という名前の米ドル通貨建てフィールドを定義したマスターファイルを示しています。

```
FILE=CURRDATA, SUFFIX=COM, $
FIELD=PRICE, FORMAT=P17.2 , ACTUAL=A5, CURR=USD, $
.
.
.
```

**通貨データソースの有効化**

通貨換算を実行するには、SET コマンドに EUROFILE パラメータを設定して、通貨データソースを指定しておく必要があります。デフォルトの状態では、EUROFILE パラメータは設定されていません。

プロシジャまたはサポートされているプロファイルで SET コマンドを発行します。このコマンドは、レポートリクエスト内で設定することはできません。

データソースを有効にすると、SET コマンドを再発行することにより、異なる通貨データソースへのアクセスが可能になります。

**注意：**EUROFILE パラメータは、単独で設定する必要があります。たとえば、SET パラメータをさらに追加すると、追加したパラメータの設定が失われます。

## 構文 通貨データソースの有効化

```
SET EUROFILE = {ddname|OFF}
```

### 説明

#### ddname

通貨データソースのマスターファイルの名前です。この ddname は、読み取り専用モードでアプリケーションがアクセス可能な既知のデータソースを参照する必要があります。

#### OFF

通貨データソースを無効にして、メモリから削除します。

## 参照 EUROFILE のエラーメッセージと注意

- ❑ 通貨データソースのマスターファイルが存在しない場合に SET EUROFILE コマンドを発行すると、次のエラーメッセージが生成されます。

(FOC205) マスターファイルがありません

- ❑ 通貨マスターファイルが FOCUS データソースを指定し、それに関連する FOCUS データソースが存在しない場合に SET EUROFILE コマンドを発行すると、次のエラーメッセージが生成されます。

(FOC036) 次の FOCUS ファイルのデータが見つかりません

## 通貨データの処理

通貨データソースの作成、データソースでの通貨建てフィールドの識別、通貨データソースの有効化をすべて完了後、通貨換算を実行することができます。

通貨データソースの各通貨 ID には、通貨 ID と同一名の仮想換算関数が生成されます。たとえば、通貨データソースにベルギーフラン (BEF) を追加すると、仮想 BEF 通貨換算関数が生成されます。

ユーロ関数の EUR は、自動的にアプリケーションに追加されます。通貨 ID の EUR を通貨データソースに追加する必要はありません。

プラットフォームにより異なりますが、換算結果は小数点以下 31 桁から 36 桁という高精度で計算されます。最終結果の精度は、常に小数点以下 2 桁に端数処理されます。換算結果を正しい精度で表示するには、小数点以下を 2 桁以上にフォーマット設定する必要があります。

## 構文 通貨データの処理

### プロシジャ

```
DEFINE FILE filename
result/format [CURR curr_id] = curr_id(infield, rate1 [,rate2]);
END
```

または

```
COMPUTE result/format [CURR curr_id] = curr_id(infield, rate1 [,rate2]);
```

### マスターファイル

```
DEFINE result/format [CURR curr_id] = curr_id(infield, rate1 [,rate2]);$
```

### 説明

#### filename

このフィールドを定義するファイルの名前です。

#### result

換算先通貨の値です。

#### format

数値フォーマットです。換算の対象となる通貨建てにより、小数点以下の桁数は 2 または 0 (ゼロ) にすることをお勧めします。小数点以下 2 桁以上を表示するフォーマット設定の場合、結果は常に小数点以下 2 桁に端数処理されます。整数および浮動小数点数は使用しないでください。

#### curr\_id

result フィールドの通貨 ID です。この ID は、EUR 値または通貨データソースの通貨 ID に一致する値でなければなりません。それ以外の値を指定すると、次のエラーメッセージが生成されます。

(FOC263) 外部関数またはロードモジュールが見つかりません

**注意：**DEFINE または COMPUTE の左側に CURR 属性を追加して、result フィールドを通貨建ての値として識別します。この値は、次に実行する通貨演算の通貨関数に引数として渡すことができます。この属性を DEFINE または COMPUTE の左側に追加しても、演算結果のフォーマットおよび値は変換されません。

#### infield

通貨建ての値です。この入力値が、元の通貨から curr\_id 通貨建てに換算されます。infield と result の通貨が同一の場合、演算は実行されず、result の値と infield の値が等しくなります。

#### rate1

通貨データソースの rate フィールドの名前です。infield の値がその通貨の rate1 の値で除算され、ユーロに相当する値が生成されます。

通貨演算に **rate2** が指定されておらず、三角換算を行う必要がある場合、この中間結果に **result** 通貨の **rate1** の値が乗算されて換算が実行されます。

場合によっては、特別な目的で別のレートを指定する必要があります。この場合、**rate1** に対して任意のフィールドまたは数値定数を指定することができます。ただし、それが 1 ユーロに相当する **infield** 通貨建ての価値を表していることが前提です。

#### rate2

通貨データソースの **rate** フィールドの名前です。この引数は、三角換算を実行する場合にのみ使用されます。三角換算では、**infield** および **result** の通貨に対して異なる **rate** フィールドを指定する必要があります。この演算の対象となる通貨のいずれかがユーロの場合、この値は無視されます。

**rate1** から算出されたユーロの値に、**result** 通貨の **rate2** の値を乗算して、換算が実行されます。

場合によっては、特別な目的で別のレートを指定する必要があります。この場合、**rate2** に対して任意のフィールドまたは数値定数を指定することができます。ただし、それが 1 ユーロに相当する **infield** 通貨建ての価値を表していることが前提です。

## 参照 通貨演算のエラーメッセージ

通貨コードを指定したマスターファイルに対してレポートリクエストを発行する場合、その通貨コードが現在アクティブな通貨データソースに含まれていないとエラーメッセージが生成されます。

(FOC1911) ファイル定義内の **CURRENCY** がデータに見つかりません。

構文エラーが発生した場合、または通貨換算式に未定義のフィールド名が存在する場合、次のメッセージが生成されます。

(FOC1912) 通貨ステートメントの解析エラーが発生しました。

## 例 通貨換算関数の使用

ここでは、通貨データソースに米ドル (USD) とベルギーフラン (BEF) という 2 つの通貨 ID が定義され、次のように **PRICE** を **BEF** の通貨建てに指定する場合を想定します。

```
FIELD = PRICE, ALIAS=, FORMAT = P17.2, CURR=BEF,$
```

- ❑ 次の例は、通貨 ID の **BEF** に **BUDGET** を替レートを使用して **PRICE** をユーロに換算し、その結果を **PRICE2** に格納する方法を示しています。

```
COMPUTE PRICE2/P17.2 CURR EUR = EUR(PRICE, BUDGET);
```

- ❑ 次の例は、三角換算規則を使用して、BEF の PRICE を USD に換算する方法を示しています。

```
DEFINE PRICE3/P17.2 CURR USD = USD(PRICE, ACTUAL);$
```

はじめに、BEF の ACTUAL レートで PRICE を除算し、小数点以下 3 桁に端数処理したユーロの値を求めます。次に、USD の ACTUAL レートをこの中間値に乗算し、その結果を小数点以下 2 桁に端数処理します。

- ❑ 次の例は、為替レートに数値定数を使用する方法を示しています。

```
DEFINE PRICE4/P17.2 CURR EUR = EUR(PRICE, 5);$
```

- ❑ 次の例は、BEF の除算に ACTUAL レートを使用し、USD の乗算に BUDGET レートを使用する方法を示しています。

```
DEFINE PRICE5/P17.2 CURR USD = USD(PRICE, ACTUAL, BUDGET);$
```

## 例

### 米ドルのユーロ、フランスフラン、ベルギーフランへの換算

次の例は、米ドル (USD) をユーロ (EUR)、フランスフラン (FRF)、ベルギーフラン (BEF) に換算する方法を示しています。

1. 通貨データソースを作成し、通貨 ID および 1 つまたは複数の為替レートを識別します。詳細は、691 ページの「[通貨データソースの作成](#)」を参照してください。次のサンプルデータソースの名前は CURRCODE です。

```
FILE = CURRCODE, SUFFIX = COM,$
FIELD = CURRENCY_ID,, FORMAT = A3, ACTUAL = A3,$
FIELD = ACTUAL, ALIAS =, FORMAT = D12.6, ACTUAL = A12,$
FIELD = BUDGET, ALIAS =, FORMAT = D12.6, ACTUAL = A12,$
```

2. データソースを作成し、換算する値を格納します。詳細は、694 ページの「[通貨データを格納するフィールドの識別](#)」を参照してください。次のサンプルデータソースの名前は CURRDATA です。

```
FILE=CURRDATA, SUFFIX=COM,$
FIELD=PRICE, FORMAT=P17.2, ACTUAL=A5, CURR=USD,$
```

3. 通貨データソースを使用するリクエストを作成し、データソースに格納された通貨の値を換算します。次のプロシジャは、PRICE を EUR、FRF、BEF に換算します。左側の番号は、そのコードの説明に対応しています。

```

 - * THE FOLLOWING ALLOCATIONS ARE FOR RUNNING UNDER z/OS
1. - * DYNAM ALLOC FILE CURRCODE DA USER1.FOCEXEC.DATA(CURRCODE) SHR REU
2. - * DYNAM ALLOC FILE CURRDATA DA USER1.FOCEXEC.DATA(CURRDATA) SHR REU
 - * THE FOLLOWING ALLOCATIONS ARE FOR RUNNING UNDER WINDOWS NT
1. FILEDEF CURRCODE DISK GGDEMO/CURRCODE.COM
2. FILEDEF CURRCODE DISK GGDEMO/CURRDATA.COM
3. SET EUROFILE = CURRCODE
 DEFINE FILE CURRDATA
4. PRICEEUR/P17.2 CURR EUR = EUR(PRICE, ACTUAL);
 END
 TABLE FILE CURRDATA
 PRINT PRICE PRICEEUR AND COMPUTE
5. PRICEFRF/P17.2 CURR FRF = FRF(PRICE, ACTUAL);
 PRICEBEF/P17.2 CURR BEF = BEF(PRICE, ACTUAL);
 END

```

このレポートリクエストは、次の順序で実行されます。

1. FILEDEF コマンドが、オペレーティングシステムに CURRCODE データソースの場所を指示します。
2. FILEDEF コマンドが、オペレーティングシステムに CURRDATA データソースの場所を指示します。
3. SET コマンドが、CURRCODE を通貨データソースとして指定します。
4. この行は、USD を EUR に換算する EUR 関数を呼び出します。
5. この 2 行は、EUR を FRF および BEF に換算する換算関数を呼び出します。

出力結果は次のとおりです。

PRICE	PRICEEUR	PRICEFRF	PRICEBEF
-----	-----	-----	-----
5.00	4.26	27.97	172.01
6.00	5.12	33.57	206.42
40.00	34.12	223.78	1376.20
10.00	8.53	55.95	344.06

算出されたユーロ値の PRICEEUR を、USD から BEF への換算に使用することはできません。それは、PRICEEUR には、三角換算で要求される小数点以下 3 桁ではなく、2 桁 (P17.2) の値が格納されているためです。

## 有効な通貨データソースのクエリ

クエリを発行して、現在有効な通貨データソースを特定することができます。この操作を実行するには、? SET ALL または ? SET EUROFILE コマンドを発行します。

## 構文 有効な通貨データソースの特定

? SET EUROFILE

## 例 有効な通貨データソースの特定

ここでは、「CURRCODE」という通貨データソースを想定します。

次のコマンドを発行します。

```
SET EUROFILE = CURRCODE
? SET EUROFILE
```

出力結果は次のとおりです。

EUROFILE	CURRCODE
----------	----------

## 数値区切り文字

数字の区切り方法は各国で異なります。区切り方法の違いをレポートに反映させるには、CDN SET パラメータで指定するコンチネンタル 10 進表記 (CDN) を使用します。CDN SET パラメータを使用すると、数値区切り文字として、カンマ (,)、ピリオド (.)、ブランク、一重引用符 (') を組み合わせて選択することができます。

CDN SET パラメータは、レポートリクエストで使用することはできますが、DEFINE または COMPUTE コマンドではサポートされていません。

**注意：** CDN パラメータで指定した区切り文字は、CENT-ZERO SET パラメータの影響を受ける数字にも使用されます。

## 構文 数値区切り文字の指定

SET CDN = option

説明

option

数字の表記に使用する区切り文字を指定します。次のオプションがあります。

- ☐ **ON** CDN を使用します。たとえば、「3,045,000.76」は「3.045.000,76」と表記されます。
- ☐ **OFF** CDN をオフにします。たとえば、「3,045,000.76」は「3,045,000.76」と表記されます。この値がデフォルト値です。

- ❑ **SPACE** 有効数字 3 桁ごとの区切り文字にはカンマ (,) の代わりにブランクを使用し、小数点にはピリオド (.) の代わりにカンマ (,) を使用します。たとえば、「3,045,000.76」は「3 045 000,76」と表記されます。
- ❑ **QUOTE** 有効数字 3 桁ごとの区切り文字にはカンマ (,) の代わりに一重引用符 (') を使用し、小数点にはピリオド (.) の代わりにカンマ (,) を使用します。たとえば、「3,045,000.76」は「3'045'000,76」と表記されます。
- ❑ **QUOTEP** 有効数字 3 桁ごとの区切り文字にはカンマ (,) の代わりに一重引用符 (') を使用し、小数点にはピリオド (.) を使用します。たとえば、「3,045,000.76」は「3'045'000.76」と表記されます。

## 例 コンチネンタル 10 進表記による数値表示

下表は、CDN の設定に基づく 1234.56 の表示方法を示しています。

CDN 設定	結果
OFF	1,234.56
ON	1.234,56
SPACE	1 234,56
QUOTE	1'234,56
QUOTEP	1'234.56

## 例 数値区切り文字の指定

次のリクエストでは、CDN を ON に設定し、数値区切り文字として千の位にはピリオド (.)、小数点にはカンマ (,) を使用します。

```
SET CDN = ON
TABLE FILE EMPLOYEE
PRINT LAST_NAME FIRST_NAME SALARYEND
```

出力結果は次のとおりです。

LAST_NAME	FIRST_NAME	SALARY
STEVENS	ALFRED	\$11.000,00
SMITH	MARY	\$13.200,00
JONES	DIANE	\$18.480,00
JONES	DIANE	\$17.750,00
BANNING	JOHN	\$29.700,00
IRVING	JOAN	\$26.862,00
IRVING	JOAN	\$24.420,00
ROMANS	ANTHONY	\$21.120,00
MCCOY	JOHN	\$18.480,00
BLACKWOOD	ROSEMARIE	\$21.780,00
MCKNIGHT	ROGER	\$16.100,00
MCKNIGHT	ROGER	\$15.000,00
CROSS	BARBARA	\$27.062,00
CROSS	BARBARA	\$25.775,00

拡張通貨記号の選択

国際言語サポート (NLS) 用に構成されたデフォルトの通貨記号に関わらず、レポートに表示する通貨記号を選択することができます。その場合は、表示オプションの [通貨記号表示 (浮動) - M] または [通貨記号表示 (固定) - N] の代わりに拡張通貨記号フォーマットを使用します。表示オプションの [通貨記号表示 (浮動) - M] または [通貨記号表示 (固定) - N] を使用する場合は、デフォルトのコードページに関連付けられた通貨記号が表示されます。たとえば、英語 (米国) コードページを使用する場合は、ドル記号 (\$) が表示されます。

**注意：**SET CURRSYMB コマンドを使用して、M と N のオプションで表示する記号を制御することができます。

一方、拡張通貨記号フォーマットを使用すると、ドル記号 (\$) 以外の記号を表示することができます。たとえば、米国ドル、イギリスポンド、日本円、またはユーロの記号を表示することができます。拡張通貨記号は、数値フォーマット (I、D、F、P) でサポートされます。

数値表示フォーマットの最後の 2 文字として、次の文字の組み合わせを使用することができます。

表示オプション	説明	例
!d	ドル記号 (固定)	D12.2!d
:d	ドル記号 (固定)	D12.2:d
ID	ドル記号 (浮動)	D12.2!D

表示オプション	説明	例
:D	ドル記号 (浮動)	D12.2:D
!e	ユーロ記号 (固定)	F9.2!e
:e	ユーロ記号 (固定)	F9.2:e
!E	ユーロ記号 (浮動) (左側)	F9.2!E
:E	ユーロ記号 (浮動) (左側)	F9.2:E
!F	ユーロ記号 (浮動) (右側)	F9.2!F
:F	ユーロ記号 (浮動) (右側)	F9.2:F
!!	イギリスポンド記号 (固定)	D12.1!!
:!	イギリスポンド記号 (固定)	D12.1:!
!L	イギリスポンド記号 (浮動)	D12.1!L
:L	イギリスポンド記号 (浮動)	D12.1:L
!y	日本円記号 (固定)	I9!y
:y	日本円記号 (固定)	I9:y
!	日本円記号 (浮動)	I9!Y
:	日本円記号 (浮動)	I9:Y

**注意：**コロン (:) と感嘆符 (!) は同じ指定ですが、すべてのコードページで変化しないことから、コロンの使用をお勧めします。

## 参照

### 拡張通貨記号フォーマット

次のガイドラインが適用されます。

- ❑ フォーマット指定の最大長は 8 バイトです。
- ❑ 拡張通貨オプションは、フォーマットの最後のオプションとして指定する必要があります。
- ❑ 拡張通貨記号フォーマットに、表示オプションの [通貨記号表示 (浮動) - M] または [通貨記号表示 (固定) - N] を含めることはできません。

- ❑ 固定通貨記号は、レポートページの最初の行にのみ表示されます。(次の例のように) レポートの列に複数の通貨記号を表示するために、フィールドベースのフォーマット再設定を使用する場合、その 1 行目に関連した記号のみが表示されます。この場合は、固定通貨記号を使用することはできません。
- ❑ 端末 I/O プロシジャにより、小文字は大文字として送信されます。そのため、固定の拡張通貨記号はプロシジャでのみ指定することができます。
- ❑ 拡張通貨記号フォーマットは、単精度浮動小数点数、倍精度浮動小数点数、パック 10 進数、整数のフォーマットが設定されたフィールドに使用することができます。文字フォーマット、動的フォーマット、可変長フォーマットを使用することはできません。

## 構文

## 拡張通貨記号の選択

`numeric_format{[:!]}option`

説明

`numeric_format`

有効な数値フォーマットです (データタイプ I、D、F、P)。

`{[:!]}`

コロン (:) か感嘆符 (!) のいずれかが必要ですが、すべてのコードページで変化しない文字は、コロン (:) のみです。

`option`

表示する通貨記号、および浮動、固定のいずれかを指定します。利用可能な値には、次のものがあります。

**d** - ドル記号 (固定) を表示します。

**D** - ドル記号 (浮動) を表示します。

**e** - ユーロ記号 (固定) を表示します。

**E** - ユーロ記号 (浮動) を左側に表示します。

**F** - ユーロ記号 (浮動) を右側に表示します。

**l** - イギリスポンド記号 (固定) を表示します。

**L** - イギリスポンド記号 (浮動) を表示します。

**y** - 日本円記号 (固定) を表示します。

Y - 日本円記号 (浮動) を表示します。

## 例 拡張通貨記号の表示

次のリクエストは、ユーロ記号を表示します。

```
SET PAGE-NUM = OFF
TABLE FILE CENTORD
PRINT PRODDNAME QUANTITY PRICE/D10.2:E
BY ORDER_DATE
WHERE QUANTITY GT 700;
ON TABLE SET STYLE *
TYPE = REPORT, GRID = OFF,$
ENDSTYLE
END
```

出力結果は次のとおりです。

<u>Date Of Order:</u>	<u>Product Name:</u>	<u>Quantity:</u>	<u>Price:</u>
2001/10/16	R5 Micro Digital Tape Recorder	726	€89.00
	ZT Digital PDA - Commercial	726	€499.00
2002/03/20	R5 Micro Digital Tape Recorder	751	€89.00
	ZT Digital PDA - Commercial	751	€499.00
2002/04/03	ZC Digital PDA - Standard	751	€299.00
2002/06/07	R5 Micro Digital Tape Recorder	751	€89.00
	ZT Digital PDA - Commercial	751	€499.00
	ZC Digital PDA - Standard	751	€299.00
	R5 Micro Digital Tape Recorder	702	€89.00
2002/06/18	ZT Digital PDA - Commercial	702	€499.00
	ZC Digital PDA - Standard	702	€299.00
	R5 Micro Digital Tape Recorder	751	€89.00
	ZT Digital PDA - Commercial	751	€499.00
2002/10/16	R5 Micro Digital Tape Recorder	798	€89.00
	ZT Digital PDA - Commercial	798	€499.00
2002/12/19	2 Hd VCR LCD Menu	701	€179.00



# Legal and Third-Party Notices

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE 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.

TIBCO, the TIBCO logo, the TIBCO O logo, ibi, ibi logo, ActiveMatrix BusinessWorks, TIBCO Administrator, BusinessConnect, TIBCO Designer, Enterprise Message Service, Hawk, and Maporama are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

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.

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.

This 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 this software version 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 SOFTWARE GROUP, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

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 "READ ME" FILES.

This and other products of Cloud Software Group, Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2023. Cloud Software Group, Inc. All Rights Reserved.