

Oracle® Hyperion Financial Management

Administrator's Guide

リリース 11.1.2.4

著作権情報

Oracle® Hyperion Financial Management Administrator's Guide, 11.1.2.4

Copyright © 2000, Oracle and/or its affiliates. All rights reserved.

著者: EPM 情報開発チーム

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション(人的傷害を発生させる可能性があるアプリケーションを含む)への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

ドキュメントのアクセシビリティについて	13
ドキュメントのフィードバック	14
第1章. Financial Managementのバージョン情報	15
Financial Managementの機能	15
Performance Management Architect	16
EPM Workspace	16
Financial Managementのディメンション	16
シナリオ・ディメンション	16
年ディメンション	17
期間ディメンション	17
エンティティ・ディメンション	17
値ディメンション	17
勘定科目ディメンション	17
内部取引ディメンション	18
ビュー・ディメンション	18
カスタム・ディメンション	18
ユーザー定義の要素	18
第2章. アプリケーションの管理	23
アプリケーションの管理	24
アプリケーション・プロファイルの作成	25
言語の入力	25
カレンダーの定義	26
期間の編集	27
期間単位の編集	28
カスタム・ディメンションの作成	29
アプリケーション・プロファイルの保存	31
アプリケーションの作成	31
アプリケーションの登録	33
アプリケーション・モジュールの構成	33
カスタム・ディメンションを追加するためのアプリケーションの変更	34
アプリケーションを開く	35
アプリケーションを閉じる	35
アプリケーション・サーバー・クラスタの変更	36
アプリケーション・リストの表示	36
アプリケーションの削除	36
アプリケーション要素のロード	37
アプリケーション要素の抽出	38
サンプル・アプリケーションの使用方法	38
システム・メッセージの使用	39
システム・メッセージの表示	39
システム・メッセージの削除	40
システム・メッセージの詳細文字列	40
システム・ユーザーの管理	41

ユーザーの表示	41
ユーザーのログアウト	42
サーバーとアプリケーションの管理	42
接続の使用不可と使用可能	43
管理モードの有効化および無効化	43
サーバーの同期化	44
タスクの監査	44
データの監査	46
実行中のタスクのモニタリング	48
実行中のタスクの表示	48
実行中のタスクの停止	49
実行中のタスクのリフレッシュ	49
無効なレコードのスキャンとクリア	50

第3章. アプリケーション・セキュリティの管理 51

アプリケーション・セキュリティに関する考慮事項	52
Financial ManagementからのShared Services Consoleの起動	52
セキュリティ・クラスを割り当てるユーザーおよびグループの選択	53
セキュリティ・クラスの設定	53
セキュリティ・クラスへのユーザー・アクセス権の割当て	55
電子メール・アラートの設定	57
セキュリティ・レポートの実行	57
アプリケーション・セキュリティのロード	58
セキュリティ情報のクリアとロード	59
セキュリティ情報をクリアする前に行っておくべきタスク	60
セキュリティ情報をクリアした後で行うタスク	60
アプリケーション・セキュリティの抽出	62

第4章. メタデータの管理 63

勘定科目の定義	64
勘定科目の種別の動作	67
動的勘定科目の定義	69
カスタム・メンバーの定義	70
カスタム・ディメンション情報の追加	72
カスタム・ディメンション情報のインポート	72
カスタム・ディメンション情報の手動での編集	73
エンティティ・メンバーの定義	73
シナリオ・メンバーの定義	74
アプリケーション設定の定義	78
期間別組織	81
連結メソッドの定義	81
連結メソッドの使用方法	83
手動での連結メソッドの割当て	83
POWNまたはPOWNMIN連結メソッドの使用	84
通貨の定義	85
セル・テキスト・ラベルの定義	87
システム生成の勘定科目	89

連結勘定科目	89
所有権勘定科目	90
システムで生成された勘定科目の編集	91
内部取引パートナーの設定	91
システムで生成されたICPメンバーの編集	92
システム生成の値メンバーの編集	93
セキュリティに基づくメタデータのフィルタ	93
APPフォーマットのメタデータ・ファイルの作成	94
ファイル・フォーマット	95
バージョン	96
アプリケーション設定	96
通貨	98
メンバー	98
連結メソッド	104
階層	104
メタデータ・ファイルに含まれないディメンション	109
メタデータ・マネージャのビューの使用	109
メタデータ・ファイルのフォーマットの変更	110
ツリー・ビューでのタスク	111
リスト・ビューでのタスク	115
リスト・ビューのメタデータのソート	117
「ファイルのプロパティ」でのメタデータ・レポートの作成	118
メタデータの参照整合性	118
メタデータの参照整合性チェック	119
メタデータ・ログ・ファイルの参照整合性エラー	119
メタデータのマージ・ユーティリティの使用	119
メタデータのロード	120
メタデータ・ロードの変更の表示	123
メタデータの抽出	124
第5章. メンバー・リストの管理	127
メンバー・リスト・ファイルの作成	128
EnumMemberLists	128
EnumMembersInList	129
動的メンバー・リスト	132
動的POVメンバー・リスト	134
メンバー・リストのロード	137
メンバー・リストの抽出	138
システム生成のリストとディメンション	138
第6章. 仕訳の管理	141
仕訳ファイルの作成	141
ファイル・フォーマット・セクション	142
バージョン・セクション	142
仕訳グループ・セクション	143
標準セクション	143
繰返しセクション	143

ヘッダー・セクション	144
仕訳のロード	146
仕訳の抽出	147
第7章. データ・フォームの管理	151
フォーム・デザイナーでのデータ・フォームの作成	151
視点の設定	152
フォームの詳細の指定	153
データ・フォームのオンデマンド・ルールの指定	153
フォームの行および列オプションの指定	154
フォームのヘッダーの指定	155
データ・フォーム・スクリプトの使用	155
AddMember	158
BackgroundPOV	160
Blank	161
Cn	162
Calc1	163
CalcByRow	163
Cell_Link	165
CellText	165
CustomHeader	166
CustomHeaderStyle	167
DynamicPOV	168
FormInputBoxLength	168
FormNumDecimals	168
FormRowHeight	169
FormScale	169
HeaderOption	170
HideInPov	171
指示	172
LineItemDetailSinglePeriod	172
リンク	173
MaxCells	173
MaxColsForSparseRetrievalMethod	174
NoSuppress	174
NumDecimals	175
OnDemandRules	175
上書き	175
POVOrder	177
PrintNumDataColsPerPage	178
PrintNumRowsPerPage	178
PrintRepeatHeadersonAllPages	179
Rn	179
ReadOnly	181
ReportDescription	181
ReportLabel	182
ReportSecurityClass	183
ReportType	183

RowHeaderPct	183
SCalc	184
Scale	185
SelectablePOVList	186
ShowDescriptions	187
ShowLabels	188
文字列	188
Style	188
SuppressColHeaderRepeats	191
SuppressInvalidCols	192
SuppressInvalidRows	192
SuppressNoDataCols	193
SuppressNoDataRows	193
SuppressRowHeaderRepeats	194
SuppressZeroCols	194
SuppressZeroRows	194
相対期間の使用	195
競合する属性の優先順位	196
データ・フォームの編集	197
データ・フォームのロード	197
データ・フォームの抽出	198
データ・フォームの削除	198
第8章. データベースへのデータの抽出	199
データ・ソース名(DSN)の構成	200
スター・スキーマ	200
スター・スキーマのフォーマット	202
接頭辞表	202
データの作成とスター・スキーマへのエクスポート	203
スター・スキーマの更新	205
スター・スキーマの削除	206
スター・スキーマのテンプレートの作成	206
スター・スキーマのテンプレートの削除	207
第9章. レポートの定義	209
仕訳レポート・スクリプトの定義	209
内部取引照合レポート・スクリプトの定義	210
内部取引照合レポートのメンバー・リストの選択	211
内部取引照合レポートの勘定科目の選択	211
内部取引照合レポートの小数点以下の桁数の指定	211
内部取引照合レポート用スタイル・シートの選択	211
内部取引照合レポートの通貨の指定	211
内部取引照合レポートの非表示オプション	212
内部取引照合レポート・スクリプトのキーワード	213
内部取引レポート・スクリプトの定義	223
第10章. ルールの管理	227

ルールのタイプ	227
ルールに関する注意事項	230
計算コマンド	230
現在のディメンション・メンバー	235
勘定科目式	236
関数によるデータの自動クリア	237
エラー・メッセージ	238
連結時のルールの実行	238
デフォルトの換算方法	239
Financial Managementのオブジェクト	240
よく使用するルール	240
データの再使用	241
値の計算による勘定科目の設定	241
条件付きのルール	242
すべての勘定科目の期首残高の設定	244
Notepad ++でのルールの作成	245
Notepad ++の起動	245
Notepad ++の構成	245
ルール・ファイルの作成	246
ルールのロード	248
ルールの抽出	250

第11章. ルール関数 251

関数の概要	254
ABSExp	261
AccountType	262
AccountTypeID	264
AddEntityToList	265
AddEntityToListUsingIDs	266
AddMemberToList	266
AddMemberToListUsingIDs	267
Alloc	267
AllowAdjFromChildren	269
AllowAdjs	271
ApplicationName	272
CalculateExchangeRate	272
CalculateRate	273
CalcStatus	274
CellTextUnitItem	276
Clear	278
Con	279
Consol1、Consol2、Consol3	281
ConsolidateYTD	282
ContainsCellText	283
ContainsCellTextWithLabel	284
Currency	285
CustomTop	286
DataUnitItem	287

Decimal	288
DefaultFreq	289
DefaultParent	290
DefaultTranslate	291
DefaultView	292
DefCurrency	293
Down	295
Dynamic	296
Exp	297
ディメンションの交差に関する考慮事項	299
期間および年のキーワード	299
算術計算	300
Exp内での他の関数の配置	301
複数の勘定科目の同時設定	301
Expおよびディメンションの交差に関する考慮事項	302
GetCell	304
GetCellNoData	305
GetCellRealData	307
GetCellText	308
GetCellTextWithLabel	308
GetCellType	309
GetCustomLabelArray	311
GetItem	311
GetItemIDs2	313
GetItemIDs2ExtDim	314
GetNumItems	315
GetNumLID	316
GetRate	317
GetSubmissionGroup	318
GetSubmissionPhase	318
Holding	319
ICPTopMember	320
ICPWeight	321
IDFromMember	322
ImpactStatus	324
Input	325
IsAlmostEqual	326
IsBase	327
IsCalculated	329
IsChild	330
IsConsolidated	333
IsDescendant	334
IsFirst	336
IsICP	337
IsLast	339
IsTransCur	340
IsTransCurAdj	340
IsValidDest	341

IsZero	342
List	344
Member	346
MemberFromID	347
Method	348
NoInput	349
NoRound	350
NumBase	351
Number	354
NumChild	354
NumCustom	357
NumDescendant	357
NumPerInGen	360
NumPeriods	361
OpenCellTextUnit	362
OpenDataUnit	363
OpenDataUnitSorted	364
Owned	364
Owner	365
PCon	365
PEPU	367
PeriodNumber	368
PlugAcct	369
POwn	370
PVAForBalance	371
PVAForFlow	372
RateForBalance	373
RateForFlow	373
ReviewStatus	374
ReviewStatusUsingPhaseID	376
Round	376
Scale	378
SecurityAsPartner	379
SecurityClass	380
SetCellTextWithLabel	382
SetData	383
SetDataWithPOV	384
SubmissionGroup	385
SupportsProcessManagement	386
SupportsTran	387
SwitchSign	388
SwitchType	390
Trans	391
TransPeriodic	392
UD1...3	393
ValidationAccount	395
ValidationAccountEx	396
XBRLTags	397

第12章. カスタム関数	399
管理レポート関数	400
Average	400
Cumulative	403
Difference	407
DSO - 売上債権回転日数	410
Opening	413
Rate	416
ビジネス・ルール関数	423
Custom_Alloc	424
Increase_Decrease	427
Pro_Rata_Ratio	430
Spread	433
Units_Rates	436
第13章. Calculation Managerを使用したルールの作成	439
Calculation Managerのセキュリティ役割	439
Calculation Managerでのアプリケーションの操作	439
Calculation Managerへのルールの移行	440
関数セレクトでのVB関数のサポート	440
Financial Managementの特別なVB Script関数	442
第14章. 内部取引の管理	445
内部取引の設定	445
内部取引の期間を開く	445
照合許容差を設定する	446
「転記前に照合」オプションの設定	447
内部取引の期間を閉じる	448
理由コードの管理	448
理由コードの追加	448
理由コードの編集	449
理由コードの削除	449
内部取引のモニタリング	449
エンティティのロックとロック解除	451
内部取引の要約の表示	452
第15章. プロセス管理の提出フェーズの管理	453
提出フェーズの定義	453
送信グループの設定	454
提出グループおよびフェーズの例	455
提出グループのフェーズへの割当て	457
未割当ての提出グループの表示	458
第16章. 電子メール・アラートの管理	461
プロセス管理のアラートの設定	461
内部取引のアラートの設定	462

付録A. 構成設定	465
使用可能な構成設定	466
構成設定の変更	470
値の上書き	471
設定表の表示の変更	471
設定の検索	471
有効な設定の表示	472
設定のエクスポート	472
設定の削除	472
付録B. パフォーマンスの最適化	473
パフォーマンスの概要	474
Oracle Hyperion EPM System Performanceの概要	474
一般的なEPMインストール・ディレクトリの参照	474
Financial Managementのレコードおよびサブキューブ	474
Financial Managementのチューニングに関する推奨事項	475
パフォーマンスの問題の診断	475
モニタリング・ツールの使用	475
リモート診断エージェント(RDA)の使用	477
参照アプリケーションの使用	477
オペレーティング・システム・パラメータのチューニング	478
Windowsパラメータのチューニング	478
Webサーバーのチューニング	478
HFM Webのチューニング	479
HFM Webのチューニング・パラメータ	480
Webブラウザの最適化	481
Financial Managementアプリケーションのチューニング	482
一般的にチューニングされたFinancial Managementの設定	482
大規模アプリケーション用のFinancial Managementのメモリー設定	486
アプリケーション固有の設定	486
Financial Managementアプリケーション・サーバーのチューニング	487
アプリケーション・データベースのメンテナンス	488
基本的な設計上の考慮事項	489
Financial Management用のOracle 11gデータベースのチューニング	490
はじめに	490
一般的なパフォーマンスの問題	491
Oracle 11g Databaseのチューニング・ガイドライン	491
Oracle Databaseリリース11gのメモリー設定を決定する方法	496
Oracle Databaseリリース11gのプロセス数を計算する方法	499
その他の考慮事項	500
よくある質問	502

ドキュメントのアクセシビリティについて

Oracleのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWeb サイト <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc> を参照してください。

Access to Oracle Support

Oracleサポート・サービスでは、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> か、聴覚に障害のあるお客様は <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> を参照してください。

ドキュメントのフィードバック

このドキュメントへのフィードバックをお送りください: epmdoc_ww@oracle.com

次のソーシャル・メディア・サイトでEPM情報開発をフォローできます:

LinkedIn - http://www.linkedin.com/groups?gid=3127051&goback=.gmp_3127051

Twitter - <http://twitter.com/hyperionepminfo>

Facebook - <http://www.facebook.com/pages/Hyperion-EPM-Info/102682103112642>

Google+ - <https://plus.google.com/106915048672979407731/#106915048672979407731/posts>

YouTube - <http://www.youtube.com/user/OracleEPMWebcasts>

1

Financial Management のバージョン情報

この項の内容:

Financial Managementの機能	15
Performance Management Architect	16
EPM Workspace	16
Financial Managementのディメンション	16
ユーザー定義の要素	18

Financial Managementの機能

Oracle Hyperion Financial Managementには、次の機能があります:

- 会社の財務情報を統一された形式で表示。様々なソースからの実績情報や指標をスケーラブルなWebベースのアプリケーションにまとめられます。
- ファスト・バーチャル・クローズ機能。Webベースのプロセス管理、Webベースの会社間調整、仕訳の調整、データやビジネス・メジャーの一貫性機能が備わっており、決算にかかる時間が短縮されます。
- パワフルなマルチディメンショナル分析機能。会社、コスト・センター、製品、ブランド、顧客、流通経路の財務とビジネスの主要な傾向、収益性の新しいソースやキャッシュ・フローを確認してレポートにまとめられます。
- 仮定に基づいた柔軟なシナリオ管理機能。実際の結果、予算、予想、プランを動的に連結して、仮定と実際のデータの変化を示すレポートを作成できます。
- フォーマット済の各種レポート。1つのアプリケーションで、内部での管理用、公開用、法的機関への提出用の正確なレポートを簡単に作成できます。
- パッケージをインストールしてそのまま使用できる各種機能。国際レベルでの配賦、複数通貨換算、既存のアプリケーションやERP、CRMシステムとのデータの統合を低コストで行えます。
- カスタマイズ可能なアプリケーション。業界標準のツールを使用して、問題をすばやく低コストで解決できます。
- Web向けの設計。ユーザーは、標準のWebブラウザを使用して、任意の場所から財務のグローバル情報に簡単かつ安全にアクセスできます。リレーショナル・データ・ストレージによって、ミッション・クリティカルなデータをユーザーは1年365日いつでも使用できます。

また、Financial Managementでは次も提供しています:

- Sustainability Reporting、IFRS、日本の法定レポートなど、特定の要件に関する事前構築済のスターター・キット・アプリケーション
- Oracle Essbaseとの統合によりレポートおよび分析機能を拡張
- その他のOracle Hyperion Enterprise Performance Managementアプリケーションとの統合

Performance Management Architect

Oracle Hyperion EPM Architectは、Financial Managementのインストールおよび構成のオプション・コンポーネントです。これを使用して、アプリケーション、ディメンション、同期データの作成や作業を行います。

Performance Management Architectで実行されるタスクについては、『Oracle Hyperion Enterprise Performance Management Architect管理者ガイド』を参照してください。

EPM Workspace

Financial Managementは、Oracle Hyperion Enterprise Performance Management Workspaceで使用できます。「ナビゲート」、「お気に入り」、「管理」または「ツール」の各メニューで使用できるプリファレンスや機能など、Oracle Hyperion Enterprise Performance Management Workspace,で実行できるタスクの詳細は、『Oracle Hyperion Enterprise Performance Management Workspaceユーザーガイド』およびオンライン・ヘルプを参照してください。

Financial Managementのディメンション

ディメンションは、組織のデータを表し、通常は関連するメンバーのグループを含みます。ディメンションの例には、勘定科目、エンティティおよび期間があります。Financial Managementは、システム定義のディメンションを8つ提供し、勘定科目に適用できるカスタム・ディメンションを無制限に追加できます。

ディメンションを構成する要素はメンバーと呼ばれます。たとえば、GrossMarginおよびTotalRevenuesは勘定科目ディメンションのメンバーです。

ディメンション・メンバーは階層に配置されます。上位レベルのメンバーは親メンバーと呼ばれ、親メンバーの直下のメンバーは子と呼ばれます。親の下にいるすべてのメンバーは子孫と呼ばれます。最下位レベルの階層のメンバーは基本レベルのメンバーと呼ばれます。

通常、データはディメンションの基本レベルのメンバーに入力され、親メンバーには入力されません。親レベルのメンバーの値は、親レベルのメンバーの子から集計されます。基本レベルのメンバーのデータが計算される場合もあります。

次の項では、システム定義のディメンションについて説明します。ディメンション属性の設定について、Performance Management Architectを使用している場合は、『Oracle Hyperion Enterprise Performance Management Architect管理者ガイド』を参照し、Financial Managementクラシック・アプリケーション管理を使用している場合は、[63ページの第4章「メタデータの管理」](#)を参照してください。

シナリオ・ディメンション

シナリオ・ディメンションは、実績、予算、予測などの一連のデータを表します。たとえば、実績シナリオは、過去と現在の企業経営を表す総勘定元帳のデータを含むことができます。予算シナリオは、対象とする企業経営を表すデータを含むことができます。予測シナリオは一般に、将来の期間の予測に対応するデータを含みます。リーガル・シナリオは、リーガルGAAPのフォーマットとルールに従って計算されたデータを含むことができます。

1つのアプリケーションに対して任意の数のシナリオを定義し、デフォルトの頻度、デフォルトのビュー、ゼロ・データの設定など、シナリオ・ディメンションのメンバーの属性を定義できます。

年ディメンション

年ディメンションは、データの会計年度またはカレンダー年を表します。アプリケーションは、2年以上のデータを含むことができます。アプリケーションを作成して、年ディメンションからデータを処理する年を選択する場合は、年の範囲を指定します。

期間ディメンション

期間ディメンションは、四半期や月などの期間を表します。期間を階層で表示すると、期間と期間単位が含まれます。たとえば、実績シナリオでデータを毎月管理する場合は、通常、このシナリオで1年に12個のデータ期間を使用できます。Financial Managementは、期間ディメンションに対して年、月および週をサポートします。

エンティティ・ディメンション

エンティティ・ディメンションは、管理構造および法的な報告構造などの会社の組織構造を表します。エンティティは、部門、子会社、工場、地域、国、法的部門、事業部門、部署、その他の組織単位などを表すことができます。任意の数のエンティティを定義できます。

エンティティ・ディメンションは、システムの連結ディメンションです。エンティティ・ディメンションの階層は、データの様々な連結ビューを示します。様々な階層は、地理的な連結、法的な連結またはアクティビティごとの連結に対応できます。組織に存在する各メンバー・コンポーネント間のすべての関係は保管されてこのディメンションで管理されます。ある組織のエンティティは基本エンティティ、従属エンティティまたは親エンティティに分類できます。基本エンティティは、組織構造の下部に存在し、他のエンティティを所有しません。依存エンティティは、組織内の他のエンティティによって所有されます。親エンティティは、1つ以上の依存エンティティを含み、直接報告を受けます。

デフォルト通貨やセキュリティ・クラスなどのエンティティ・ディメンションのメンバーの属性を定義し、エンティティで調整が可能かどうか、および内部取引の詳細を保管するかどうかを指定する属性を定義します。

値ディメンション

値ディメンションは、アプリケーションに保管されたタイプの値を表し、入力通貨、親通貨、調整および比例、消去、貢献の詳細などの連結の詳細を含めます。たとえば、エンティティ通貨のメンバーはエンティティの値をその現地通貨に保管します。親通貨のメンバーは、親エンティティの通貨に換算されたエンティティの値を保管します。値ディメンションは、データに適用される取引の監査証跡を作成する際に便利です。

勘定科目ディメンション

勘定科目ディメンションは、通常の勘定科目の階層を表します。勘定科目は、エンティティとシナリオの財務データをアプリケーションに保管します。各勘定科目には、会計の動作を定義する収益、費用などのタイプがあります。

勘定科目の種別、表示する小数点以下の桁数、勘定科目が計算済勘定科目か、連結済勘定科目か、内部取引パートナーの勘定科目かなど、勘定科目ディメンションのメンバーの属性を定義します。

内部取引ディメンション

内部取引ディメンションは、勘定科目に存在する内部取引のすべての残高を表します。これは、勘定科目ディメンションと任意のカスタム・ディメンションとの組合せに使用される予約ディメンションです。Financial Managementは、すべての勘定科目とエンティティについて、内部取引の詳細を追跡および消去できます。また、内部取引照合レポートを実行して内部取引を表示できます。

ビュー・ディメンション

ビュー・ディメンションは、期別、年次累計、四半期累計などの期間単位のカレンダ・インテリジェンスの様々なモードを表します。ビューを「期別」に設定すると、各月の値が表示されます。ビューを「年次累計」または「四半期累計」に設定すると、年次または四半期の累積値が表示されます。

カスタム・ディメンション

カスタム・ディメンションは、勘定科目に関連付けられたディメンションです。これらのディメンションを使用すると、製品、マーケット、チャネル、貸借対照表の動向または消去のタイプなど、勘定科目に関連付けられているその他の詳細を指定できます。たとえば、カスタム・ディメンションには、製品ライン、地域、チャネル、顧客などがあります。SalesおよびCOGS勘定科目に関連付けられている製品のカスタム・ディメンションでは、売上と費用の詳細を製品ごとに追跡できます。

ユーザー定義の要素

Financial Managementの多くの要素はユーザー定義です。たとえば、仕訳の作成時には、仕訳にラベルと説明を指定します。

次の表に、ユーザー定義の要素、各要素の最小長と最大長、およびその他の制限を示します。この表では、要素がモジュール別にグループ分けされています。

表1 ユーザー定義の要素

要素	最小の長さ	最大の長さ	制限
アプリケーション・プロファイル			
言語	1	20	なし
期間のラベル	1	80	<ul style="list-style-type: none">特殊文字は使用できません。空白、記号、ウムラウトなどの発音区別符号は使用できません。
ビューのラベル	1	10	<ul style="list-style-type: none">特殊文字は使用できません。空白、記号、ウムラウトなどの発音区別符号は使用できません。

要素	最小の長さ	最大の長さ	制限
ビューの説明	0	40	アンパサンド(&)は使用できません。
期間の説明	0	40	アンパサンド(&)は使用できません。
アプリケーションの作成			
アプリケーションのラベル	1	10	<ul style="list-style-type: none"> 特殊文字は使用できません。 1文字目は数字にできません。 空白、記号、ウムラウトなどの発音区別符号、ドイツ語の大文字エスツェットなどの特殊文字は使用できません。 <p>注: アプリケーション・ラベルでは、大文字と小文字は区別されません。たとえば、App1とAPP1は同じアプリケーション・ラベルとみなされます。</p>
アプリケーションの説明	1	255	アンパサンド(&)は使用できません。
メタデータ・マネージャ			
メンバーのラベル	1	80	<p>一意にする必要があります。空白も含め、80文字まで入力できますが、1文字目は空白にできません。次の文字は使用できません:</p> <ul style="list-style-type: none"> アスタリスク(*) アット記号(@) カンマ(,) 中カッコ({}) 二重引用符(") スラッシュ(/) マイナス記号(-) シャープ(#) ピリオド(.) プラス記号(+)

要素	最小の長さ	最大の長さ	制限
			<ul style="list-style-type: none"> セミコロン(;)) 注: エンティティ名を「ALL」にすることはできません。
メンバーの説明	0	40	アンパサンド(&)は使用できません。
別名のラベル	0	80	アンパサンド(&)は使用できません。
セキュリティ			
セキュリティ・クラス	1	80	次の文字は使用できません: <ul style="list-style-type: none"> アスタリスク(*) アット記号(@) カンマ(,) 中カッコ({}) 二重引用符(") マイナス記号(-) シャープ(#) ピリオド(.) プラス記号(+) セミコロン(;)) スラッシュ(/)
仕訳			
仕訳のラベル	1	20	次の文字は使用できません: <ul style="list-style-type: none"> アスタリスク(*) アット記号(@) カンマ(,) 中カッコ({}) 二重引用符(") スラッシュ(/) マイナス記号(-) シャープ(#) ピリオド(.) プラス記号(+) セミコロン(;))
仕訳の説明	0	255	なし
仕訳グループ	0	30	なし

要素	最小の長さ	最大の長さ	制限
仕訳のライン・アイテムの説明	0	50	なし
ロード/抽出			
区切り文字	1	1	<p>次の文字のいずれかを使用する必要がありますが、ファイル内またはファイル名では使用できません:</p> <ul style="list-style-type: none"> • アンパサンド(&) • アット記号(@) • 円記号(バックスラッシュ)(¥) • 脱字記号(^) • コロン(:) • カンマ(,) • ドル記号(\$) • 縦線() • パーセント記号(%) • 疑問符(?) • セミコロン(;) • チルダ(`) <p>注:</p> <p>ファイル全体で同じ区切り文字を使用する必要があります。同じファイル内で異なる区切り文字を使用すると、ファイルのロード時にエラーが発生します。</p>
データ・グリッド			
セルの説明	1	1900	なし
ライン・アイテムの詳細	1	80	なし
注釈	0	255	なし
小数点	1	1	<p>次の文字は、データ・グリッドでは小数点として使用できません:</p> <ul style="list-style-type: none"> • 円記号(バックスラッシュ)(¥) • スラッシュ(/) • マイナス記号(-)

要素	最小の長さ	最大の長さ	制限
			<ul style="list-style-type: none"> ・ プラス記号(+)
ドキュメント			
ドキュメント名 (フォルダおよびレポート名を含む)	1	16	<p>次の文字は、ドキュメント名には使用できません:</p> <ul style="list-style-type: none"> ・ アスタリスク(*) ・ アット記号(@) ・ 円記号(バックslash)(¥) ・ コロン(:) ・ カンマ(,) ・ 中カッコ({}) ・ 二重引用符(") ・ slash(/) ・ 大なり記号(>) ・ 小なり記号(<) ・ 縦線() ・ シャープ(#) ・ ドキュメント名の最後のピリオド(.) ・ プラス記号(+) ・ 疑問符(?) ・ セミコロン(;)) <p>注:</p> <p>ドキュメント名の末尾または先頭を空白文字にすることはできません。</p>

2

アプリケーションの管理

この項の内容:

アプリケーションの管理	24
アプリケーション・プロファイルの作成	25
アプリケーションの作成	31
アプリケーションの登録	33
アプリケーション・モジュールの構成	33
カスタム・ディメンションを追加するためのアプリケーションの変更	34
アプリケーションを開く	35
アプリケーションを閉じる	35
アプリケーション・サーバー・クラスタの変更	36
アプリケーション・リストの表示	36
アプリケーションの削除	36
アプリケーション要素のロード	37
アプリケーション要素の抽出	38
サンプル・アプリケーションの使用方法	38
システム・メッセージの使用	39
システム・ユーザーの管理	41
サーバーとアプリケーションの管理	42
タスクの監査	44
データの監査	46
実行中のタスクのモニタリング	48
無効なレコードのスキャンとクリア	50



注意

アプリケーション管理に関するこの章の情報は、クラシックFinancial Managementのアプリケーション専用です。クラシック・アプリケーション管理で作成したアプリケーションをPerformance Management Architectに変換すると、クラシックFinancial Management管理のアプリケーションを操作できなくなります。Performance Management Architectを使用したアプリケーション管理については、『Oracle Hyperion Enterprise Performance Management Architect管理者ガイド』を参照してください。

アプリケーションは、エンティティ、勘定科目、シナリオおよび同時に使用するその他のディメンションで構成されます。必要な数のアプリケーションを作成できます。たとえば、複数の組織用に税務データをレポートするアプリケーションと、その他の組織用にセキュリティおよび為替歩合データをレポートするアプリケーションを設定できます。

アプリケーション管理では、「ディメンション・エディタ」および「Financial Managementアプリケーション作成者/Financial Managementマネージャ」のセキュリティ役割が必要です。役割の詳細は、*Oracle Enterprise Performance Management Systemユーザー・セキュリティ管理ガイド*を参照してください。

アプリケーションは、アプリケーション・サーバーで実行します。アプリケーション・サーバーのクラスタを設定して、複数のサーバーの負荷を分散します。手順については、『*Oracle Hyperion Enterprise Performance Management Systemインストールおよび構成ガイド*』を参照してください。

アプリケーション・サーバーの構成時に指定した管理者グループのメンバーのみが、次の管理タスクを実行できます。

システムのユーザー

- ユーザーのリスト
- ユーザーのログアウト

サーバーとアプリケーションの管理

- 無効なコンポーネントの表示
- 接続の有効化または無効化
- ユーザーのログアウト

システム・メッセージ

- 表示: 管理者グループのメンバーであることが必要
- 削除: 管理者グループのメンバーであることが必要

ユーザーを管理者グループに割り当てるには、*Oracle Enterprise Performance Management Systemユーザー・セキュリティ管理ガイド*を参照してください。

アプリケーションの管理

アプリケーションの定義は、次の5つの基本的な手順で行います。

1. アプリケーションのカレンダ、言語および期間単位の各定義を含むアプリケーション・プロファイルを作成します。[25ページのアプリケーション・プロファイルの作成](#)を参照してください。
2. アプリケーション・シェルを作成して、アプリケーションの格納場所であるアプリケーション・サーバーを定義し、アプリケーション・ラベルと説明、アプリケーション・プロファイル、およびログ・ファイルの場所を定義します。[31ページのアプリケーションの作成](#)を参照してください。
3. どのユーザーにアプリケーションへのアクセス権を付与するか、各ユーザーにどのタイプのアクセス権を付与するかなど、アプリケーションのセキュリティを定義します。[51ページの第3章「アプリケーション・セキュリティの管理」](#)を参照してください。
4. 勘定科目、エンティティ、シナリオ、カスタム・ディメンション、アプリケーション設定、連結メソッド、通貨などのアプリケーションのメタデータを定義します。[63ページの第4章「メタデータの管理」](#)を参照してください。
5. データ、データ・フォーム、メンバー・リスト、ルールおよび仕訳をアプリケーションにロードします。

アプリケーション・プロファイルの作成

アプリケーション・プロファイルには、アプリケーションのメタデータの説明言語、カレンダー(年および期間)、頻度、およびカスタム・ディメンションの情報が含まれています。

1つのプロファイルを2つ以上のアプリケーションに対して使用できます。作成するアプリケーションごとにアプリケーション・プロファイルを指定する必要があります。

以下のトピックを参照してください。

- [25ページの言語の入力](#)
- [26ページのカレンダーの定義](#)
- [27ページの期間の編集](#)
- [28ページの期間単位の編集](#)
- [29ページのカスタム・ディメンションの作成](#)
- [31ページのアプリケーション・プロファイルの保存](#)

▶ アプリケーション・プロファイルを作成するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「管理タスク」から、「プロファイル・エディタ」をダブルクリックするか、「管理」、「プロファイル・エディタ」の順に選択します。
3. 次のいずれかのオプションを選択します:
 - 「新規アプリケーション・プロファイルの作成」を選択します。
 - 「既存のアプリケーション・プロファイルの選択」を選択し、「参照」をクリックして、ファイルを検索します。
4. 「OK」をクリックします。

言語の入力

アプリケーション全体で使用される説明には、10までの言語を指定できます。様々な言語を使用して、アプリケーションのアイテムごとに説明を作成できます。



注意

アプリケーションを作成した後で、アプリケーションの言語を変更することはできません。

▶ 言語を指定するには:

1. 「言語」画面から、「言語の追加」(+)ボタンをクリックするか、「アクション」、「言語の追加」の順に選択します。



注:

10までの言語を入力できます。各言語には、最大20文字まで使用できます。スペースも1文字としてカウントされることに注意してください。

2. 「次」をクリックするか、上部の進捗状況バーから「カレンダー」をクリックします。

カレンダーの定義

アプリケーションのカレンダーは、年、期間およびビュー・ディメンションを使用して設定します。年ディメンションのメンバーは、アプリケーションの開始年と年の合計を定義します。期間ディメンションのメンバー階層は、アプリケーションの基本レベルの期間と要約レベルの期間を定義します。期間ディメンションの最上位レベルの親は、常に[年]メンバーです。ビュー・ディメンションは、当四半期初めからの累計や年次累計など、アプリケーションの期間累計のビューを定義します。ビュー・ディメンションには、基本期間を表す「期別」という名前のシステム定義メンバーが含まれます。ビュー・メンバーは、期間ディメンションのメンバー階層のレベルに対応している必要があります。

開始年、年数、基本期間、要約期間、期間累計のビューはカスタマイズできます。アプリケーション・プロファイルのカレンダーのタイプと期間を選択すると、アプリケーション・プロファイルのデフォルトの期間単位が作成されます。たとえば、標準カレンダーを選択し、期間として半年、四半期および月を含めた場合、次の期間単位が作成されます: 年次、半年ごと、四半期ごと、および月次。

また、カスタム・カレンダーや手動定義のカレンダーを選択できます。カスタム・カレンダーを定義する場合は、期間数と期間のラベルの接頭辞を定義する必要があります。フラット・リストが作成されます。階層は後で変更できます。

手動カレンダーを定義する場合は、期間単位と期間が空白になります。期間単位と期間は、必要に応じて後で入力できます。



注意

アプリケーションを作成した後で、アプリケーションの年の変更、または期間の説明またはラベルの変更は行えません。

▶ カレンダーを設定するには:

1. 「**カレンダー**」画面の「**期間**」セクションで、カレンダーのタイプを選択します。

- 標準カレンダー
- カスタム・カレンダー
- 手動で定義したカレンダー

2. カレンダー・オプションを選択します:

- 「**標準カレンダー**」を選択した場合:
 - 含める期間(半年、四半期、三半期または月)を選択します。
 - カレンダーに月を含めた場合、「**開始月**」ドロップダウン・リストから、カレンダーの最初の月を選択します。
- 「**カスタム・カレンダー**」を選択した場合:
 - 「**期間のラベルの接頭辞**」に、含める期間の接頭辞を入力します。

ラベルには、スペースも含めて10文字まで使用できます。

- 「**基本期間数**」に、年内の期間数を入力します。

たとえば、期間数に10を、ラベルの接頭辞にNewPeriodを入力すると、これらの期間は、NewPeriod1からNewPeriod10として階層に追加されます。

- 「**手動定義カレンダー**」を選択した場合:
 - 「**開始年**」に、カレンダーの開始年を入力します。

○「年数」に、含める年数の合計を入力します。

3. 「次」をクリックするか、上部の進捗状況バーから「期間単位」をクリックします。

期間の編集

期間階層は、定義した期間と期間単位を組み合わせたものです。この階層には、期間を追加または削除することで、変更を加えることができます。たとえば、4番目の四半期にもう1か月を追加して、13か月の1年を表示できます。

期間を選択したら、期間のラベルや説明を変更できます。[Year]期間のラベルまたは説明は編集できません。



注:

期間のラベルを言語ごとに編集することはできません。ある言語で期間のラベルを変更すると、その変更はアプリケーション・プロファイルで定義された他のすべての言語にも適用されます。ただし、言語ごとに一意の期間の説明を持たせることができます。

兄弟期間と子期間を追加するには、[27ページの兄弟期間と子期間の追加](#)を参照してください。

▶ 期間を入力して編集するには:

1. 期間階層から期間を選択し、期間ラベルまたは説明を入力または編集します。

期間のラベルには、40文字まで入力できます。期間の説明には、80文字まで入力できます。スペースも1文字としてカウントされることに注意してください。

2. アプリケーション・プロファイルを作成するときは、必ず各期間の説明を付けてください。



説明を付けずに後で追加する場合は、アプリケーション・プロファイルを変更して、そのプロファイルを使用するすべてのアプリケーションを再度作成する必要があります。



兄弟期間と子期間の追加

期間の階層に、兄弟期間や子期間を1つ以上追加できます。

▶ 子期間または兄弟期間を1つ追加するには:


1. 1つの子期間または兄弟期間を追加する期間をハイライトします。
2. 次のいずれかのオプションを選択します:

•  「子の挿入」ボタン  をクリックするか、「+」ドロップダウン・リストから「子の挿入」をクリックするか、または「アクション」、「子の挿入」の順に選択します。

•  「兄弟の挿入」ボタン  をクリックするか、「+」ドロップダウン・リストから「兄弟の挿入」をクリックするか、または「アクション」、「兄弟の挿入」の順に選択します。

3. 新しい兄弟期間または子期間の名前を入力します。

▶ 兄弟期間または子期間を複数追加するには:

1. 複数の子期間または兄弟期間を追加する期間をハイライトします。
2.  「多数の挿入」ボタン をクリックするか、「+」ドロップダウン・リストから「多数の挿入」をクリックするか、または「アクション」、「多数の挿入」の順に選択します。
3. 追加する期間の数を入力します。
4. 新しい期間のラベルの接頭辞を入力し、「OK」をクリックします。



ヒント:

期間数に10を、ラベルの接頭辞にNewPeriodを入力すると、これらの期間は、NewPeriod1からNewPeriod10として階層に追加されます。

5. 期間を追加したら、「次」をクリックするか、上部の進捗状況バーから「カスタム・ディメンション」をクリックします。

期間の削除

期間を階層から削除すると、その期間の子孫もすべて削除されます。



注:

[Year]期間は削除できません。

▶ 期間を削除するには:

1. 削除する期間を選択します。
2. 「期間の削除」をクリックするか、「アクション」、「期間の削除」の順に選択します。

「カスタム・ディメンション」セクションに移動するには、「次」をクリックするか、上部の進捗状況バーから「カスタム・ディメンション」をクリックします。

期間単位の編集

期間単位は、月や四半期など、データを入力できる期間のレベルを指定します。期間単位とこれに対応するビューは、カレンダーを定義するときに選択した期間に基づいて作成されます。期間単位は、追加、変更、削除できます。また、各期間単位にわかりやすいラベルを入力したり、あらかじめ定義した各言語で表示したりできます。

年次累計(YTD)の期間単位は、「期間単位」列の最初の行にデフォルトで指定されます。YTDの期間単位の削除、またはラベルの変更はできません。ただし、定義した各言語のYTDの期間単位の説明を入力することはできます。



注:

ユーザーが定義できる期間単位の他に、各アプリケーションに2つのシステム定義の期間単位および対応するビュー(シナリオ・ビューおよび期別)があります。

カレンダーの期間に「手で定義したカレンダー」を選択した場合、期間単位のグリッドは空になるため、必要な期間単位のビューとその説明を入力する必要があります。期間のディメンションのレベルごとに、1つの期間単位を入力する必要があります。



注意

アプリケーションを作成した後で、アプリケーションの期間単位の説明またはラベルを変更することはできません。

▶ 期間単位を定義するには:

1. 「期間単位」画面から、「期間単位の追加」(+)ボタンをクリックするか、「アクション」、「期間単位の追加」の順に選択します。
2. 1つ以上の期間単位および説明を入力します。

デフォルトでは、YTDが最初の期間単位です。定義した期間単位数は、期間ツリー内の生成数と同じである必要があります。



注:

ラベルには最大40文字まで入力できます。説明には、80文字まで入力できます。

ラベルには、次の文字は使用できません: . + - * / # { } ; , @ "



ヒント:

アプリケーションを作成した後で期間単位を変更することはできないため、各言語の各期間単位に必ず説明を含めるようにしてください。

3. 「次」をクリックするか、上部の進捗状況バーから「期間」をクリックします。

カスタム・ディメンションの作成

カスタム・ディメンション情報は、アプリケーション・プロファイルの一部として指定します。

カスタム・ディメンション表を開くと、アプリケーションの最初の2つのカスタム・ディメンションが自動的に表示されます。これらのカスタム・ディメンションは、通貨レートおよび連結メソッド情報に使用され、これらのディメンションのサイズを「大」にする必要があります。1つ目のディメンションは通貨(元)および連結メソッド情報に、2つ目のディメンションは通貨(先)の情報に使用されます。これら2つのエントリのカスタム・ディメンションに対して、ディメンション名(短い説明)とディメンション別名(長い説明)を指定できます。通貨に使用列のサイズまたは情報は変更できません。

次のガイドラインに従って、デフォルトのカスタム・ディメンションの他に、追加のカスタム・ディメンションを作成できます:

- アプリケーションに、一意のカスタム・ディメンション名とディメンション別名を入力する必要があります。名前と別名の両方が一意である必要があります。たとえば、PRODというディメンション名を使用している場合、ディメンション別名をPRODにはできません。

- ・ デイメンション名は最大10文字です。
- ・ デイメンション別名は最大20文字です。
- ・ デイメンション名と別名に、スペースや空白を使用することはできません。
- ・ カスタム・デイメンションに、次のいずれかのサイズを選択する必要があります: 「小」、「中」、「大」。

カスタム・デイメンションの最大数は、データベース・タイプと各カスタム・デイメンションのサイズによって異なります。

- ・ 「大」のカスタム・デイメンションは、20億のメンバーをサポートし、保管用にデータベース列に4バイトのスペースが必要です。
- ・ 「中」は、32,000メンバーをサポートし、2バイトが必要です。
- ・ 「小」は、128メンバーまでをサポートし、1バイトが必要です。

カスタム・デイメンションの最大数の計算方法はデータベース・タイプに関係なく同じですが、データベース・タイプは、使用可能な合計スペースの量を決定します。

物理カスタム列の合計数は、Oracle、Microsoft SQL ServerまたはIBM DB2データベースのカスタム・デイメンション・ストレージで使用可能な合計バイト数を決定します。

- ・ Oracleは、21物理カスタム列までをサポートし、使用可能なスペースの合計は、 $21 \times 8 = 168$ バイトです。
- ・ SQL Serverは、5物理カスタム列をサポートし、使用可能なスペースの合計は、 5×8 バイト = 40バイトです。
- ・ IBM DB2は、900バイトまでの主キーをサポートし、使用可能なスペースの合計は、800バイトです。

少なくとも2つのカスタム・デイメンションが必要です。3つ以上のカスタム・デイメンションの計算に使用する式は次のとおりです:

Migrating applications: $4 \times LD + 2 \times MD + SD \leq \text{MAXBYTES} - 8$

New applications: $4 \times LD + 2 \times MD + SD \leq \text{MAXBYTES}$

LD = 「大」のデイメンション数、 MD = 「中」のデイメンション数、 SD = 「小」のデイメンション数。

MAXBYTES = 40 (Microsoft SQL Serverの場合)、168 (Oracleの場合)、800 (IBM DB2の場合)

実用的には、カスタム・デイメンションに「中」を使用することをお勧めします。

アプリケーションのライフタイムにおいて、128より多いメンバーを持つことが予想される場合、カスタム・デイメンションを「小」サイズとすることはお勧めしません。

アプリケーションのライフタイムにおいて、デイメンションに32,536より多いメンバーを持つ予定である場合のみ、カスタム・デイメンションに「大」サイズを使用することをお勧めします。

次の例は、カスタム・デイメンションのサンプルを示しています。

カスタム・デイメンション名	カスタム・デイメンション別名	カスタム・デイメンション・サイズ	通貨に使用
Prod	Products	大	通貨(元)
Mkt	Markets	大	通貨(先)
Flows	BSFlows	小	N/A

カスタム・ディメンション名	カスタム・ディメンション別名	カスタム・ディメンション・サイズ	通貨に使用
Cust	Customers	大	N/A

▶ カスタム・ディメンションを作成するには:

1. 「カスタム・ディメンション」表で、「カスタム・ディメンションの作成」(+)**ボタンをクリックするか、「アクション」、「カスタム・ディメンションの作成」の順に選択します。**
2. 「ディメンション名」に、一意のディメンション名を入力します。
3. 「ディメンションの別名」に、一意のディメンション別名を入力します。
4. 「ディメンション・サイズ」に、次のいずれかのオプションを選択します: 「小」、「中」または「大」。
5. 追加のディメンションを作成するには、これらの手順を繰り返します。

アプリケーション・プロファイルの保存

プロファイル・エディタのすべての手順を完了したら、プロファイルを保存します。プロファイルは、.PER (バイナリ) またはXMLファイルとして保存できます。アプリケーション・プロファイルのファイルには、デフォルトでPERという拡張子が使用されます。


アプリケーション・プロファイルを保存した後は、プロファイル・エディタから直接アプリケーションを作成できます。

▶ アプリケーション・プロファイルを保存するには:

1. 「保存」をクリックするか、「アクション」、「保存」の順に選択します。
2. ファイル・フォーマットの**選択**からオプションを選択し、「OK」をクリックします。

- xml
- バイナリ

ファイルは「Downloads」ディレクトリに保存されます。

3. オプション: ファイルの名前を変更するには、ファイルの名前を入力するか、または  をクリックして上書きするプロファイルを参照します。
4. オプション: アプリケーションを作成するには、「アプリケーションの作成」をクリックするか、「アクション」、「アプリケーションの作成」の順に選択します。

アプリケーションの作成



注意

この情報は、クラシックFinancial Managementアプリケーション専用です。Performance Management Architectを使用したアプリケーション管理については、『Oracle Hyperion Enterprise Performance Management Architect管理者ガイド』を参照してください。

アプリケーションは、一定の分析要件またはレポート要件を満たすディメンションおよびディメンション・メンバーの集合です。たとえば、あるアプリケーションで税金のデータに関するレポートを作成し、別のアプリケーションでは販売の分析を行えます。

税引当アプリケーションの詳細は、Oracle Hyperion Tax Provisionのドキュメントを参照してください。

アプリケーションを作成するには、次の情報を指定します:

- 新規アプリケーションが稼働するアプリケーション・サーバー・クラスタ
- アプリケーション・ラベル。
- アプリケーションの説明。
- アプリケーションのカレンダー、言語および期間単位の各定義を含むアプリケーション・プロファイル。
- アプリケーションの追加先のOracle Hyperion Shared Servicesプロジェクト。

『Oracle Enterprise Performance Management Systemユーザー・セキュリティ管理ガイド』を参照してください。

- アプリケーション・タイプ



注:

アプリケーションを作成したら、セキュリティを設定する必要があります。

▶ アプリケーションを作成するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「アプリケーション」タブで、「新規」をクリックするか、または「アクション」、「新規」の順に選択します。
3. 「クラスタ」リストから、新規アプリケーションが稼働するアプリケーション・サーバー・クラスタを選択します。

サーバー・クラスタが表示されない場合は、サーバー・クラスタの登録が必要な場合があります。『Oracle Enterprise Performance Management Systemインストールおよび構成ガイド』を参照してください。

4. 「名前」に、新規アプリケーションの名前を入力します。

アプリケーション・ラベルには最大10文字の英数字または12バイトを使用できます。これは、数字で始めたり、空白、アンパサンド(&)記号、特殊文字または5文字を超える国際文字を含むことはできません。また、次の文字を含むこともできません: ドイツ語の大文字のß (Unicode U+1E9E)および小文字のß (Unicode U+00DF)、トルコ語の点のない小文字のİ (Unicode U+0131)。

アプリケーション・ラベルでは、大文字と小文字は区別されません。たとえば、App 1とAPP1は同じアプリケーションとみなされます。HFM、HSVおよびHSXは予約名であるため、アプリケーション・ラベルまたは説明には使用できません。



注意

アプリケーションを別のOracle Hyperion Financial Managementアプリケーション・サーバーに置く場合でも、同じ名前で作成しないでください。同じ名前を持つ複数のアプリケーションは、それぞれが別のFinancial Managementアプリケーション・サーバーにある場合でも、1つのOracle Hyperion Shared Servicesサーバーでは共存できません。

5. 「説明」に、アプリケーションの説明を入力します。

アプリケーションの説明にはスペースを含む最大255文字を使用できます。HFM、HSVおよびHSXは予約名であるため、アプリケーションの説明には使用できません。

6. 「プロファイル」テキスト・ボックスの横にある「参照」をクリックして、使用するアプリケーション・プロファイルを検索します。

アプリケーション・プロファイルのファイルには、デフォルトでPERという拡張子を使用されます。

7. 「ユーザー管理プロジェクト」リストから、アプリケーションを追加するShared Servicesプロジェクトを選択します。



注:

各アプリケーションはプロジェクトに属している必要があります。『Oracle Enterprise Performance Management Systemユーザー・セキュリティ管理ガイド』を参照してください。

8. 「アプリケーション・タイプ」リストから、次のタイプを選択します:

- ・ 連結
- ・ 税引当

9. 「作成」をクリックします。

アプリケーションの登録

インストール・プロセス中にサーバー・クラスタを構成および登録します。アプリケーションの作成後、優先サーバー・クラスタに対して、アプリケーションを登録できます。

▶ アプリケーションを登録するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「管理」、「アプリケーション」の順に選択します。
3. アプリケーション・リストから、アプリケーションを選択します。
4. 「登録」をクリックするか、「アクション」、「登録」の順に選択します。
5. 「クラスタ」リストから、サーバー・クラスタを選択します。
6. 「ユーザー管理プロジェクト」リストから、Oracle Hyperion Shared Servicesプロジェクトを選択します。
7. 「登録」をクリックします。

アプリケーション・モジュールの構成

アプリケーションを作成する場合、すべてのアプリケーション・モジュールはデフォルトで有効になっています。ユーザーは、適切なセキュリティ役割があるモジュールを表示できます。

アプリケーション・タイプによっては、すべてのモジュールが必要ない場合があります。アプリケーション管理者の場合、特定のアプリケーション・モジュールを無効にするよう選択できます。たとえば、特定のアプリケーションの仕訳または内部取引モジュールを無効にすることができます。モジュールを無効にすると、いずれのアプリケーション・ユーザーに対しても、そのモジュールは表示されません。



注:

無効化するモジュールに関連付けられたドキュメント(仕訳や内部取引明細レポートなど)がある場合、それも必ずシステムから削除してください。

モジュールに変更を加えた後、変更を表示するには、アプリケーションを閉じて再度開きます。

アプリケーション・モジュールの構成情報は、アプリケーション名_Moduleconfig.xmlという名前のXMLファイルとして保存されます。「アプリケーション要素のロード」ページおよび「アプリケーション要素の抽出」ページから、モジュールの構成情報をロードおよび抽出できます。

▶ アプリケーション・モジュールを構成するには:

1. 「ナビゲート」、「アプリケーション」、「連結」の順に選択し、アプリケーションを選択します。
2. 「連結」、「メンテナンス」、モジュール構成の順に選択するか、または、「アプリケーション・タスク」ペインから、「メンテナンス」を展開し、モジュール構成を選択します。
3. モジュール構成ページの無効なモジュールリストで、無効化するモジュールを次から選択します:
 - ・ 仕訳
 - ・ 内部取引(ICT)
 - ・ エクイティ・ピックアップ(EPU)
 - ・ 出資比率の管理
 - ・ プロセス管理
 - ・ データ管理
 - ・ 監査タスク
4. 「保存」をクリックするか、「アクション」、「保存」の順に選択します。
5. 変更内容を表示するには、アプリケーションを閉じて再度開きます。

カスタム・ディメンションを追加するためのアプリケーションの変更

新規アプリケーション・プロファイルを作成することによって、既存のアプリケーションにカスタム・ディメンションを追加できます。既存のカスタム・ディメンションを削除または修正することはできず、他のプロファイル設定を変更することもできません。



注:

カスタム・ディメンションを追加し、アプリケーションを作成またはデプロイした後は、カスタム名または別名を変更できません。

カスタム・ディメンションを追加するには、「アプリケーション管理者」のセキュリティ役割が割り当てられている必要があります。

カスタム・ディメンションを追加する前に、次のタスクを完了します:

- ・ CASSecurity、HsxServerおよびHsvDataSourceプロセスが、すべてのサーバーで停止していることを確認します。
- ・ Hyperion Financial Management - 管理サービスが、すべてのサーバーで停止していることを確認します。
- ・ DMEListenerおよびHFMSERVICEプロセスが、すべてのサーバーで停止していることを確認します。
- ・ 続行する前に、アプリケーションをバックアップしてください。

- 必要な場合、元のアプリケーションでデータ監査情報を復元できるように、データ監査表をアーカイブします。
- データ監査情報をアーカイブしたら、データ監査表をクリアします。
- すべてのユーザーがアプリケーションからログアウトしていることを確認してください。
- アプリケーションがいずれのアプリケーション・サーバーでも実行されていないことを確認してください。

変更するアプリケーションを選択したとき、それが以前に変更されていた場合、最終変更の日付、時間およびステータス(完了、エラーで完了など)を示す警告メッセージが表示されます。

▶ カスタム・ディメンションを追加するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「管理」、「アプリケーション」の順に選択するか、または「管理タスク」から「アプリケーション」を選択します。
3. 変更するアプリケーションをハイライトします。
4. 「変更」をクリックするか、または「アクション」、「変更」の順に選択します。
5. アプリケーションのバックアップ・プロシージャを実行したことを確認してから、「変更」をクリックします。
6. クラスタを選択し、追加のカスタム・ディメンションを含む新規プロファイルを選択してから、「変更」をクリックします。
7. 既存および新規のカスタム・ディメンションを表示する新しいダイアログが開きます。アプリケーションへの変更を確認し、「変更」をクリックします。

変更プロセスが開始すると、進行状況ダイアログが表示され、そこで進行状況をモニターしたり、ダイアログ・ボックスを閉じたりできます。変更プロセスが実行されているかぎり、アプリケーションに対して「変更」をクリックすると、同じダイアログが表示されます。

8. 変更プロセスが完了すると、システム・メッセージ・モジュールでメッセージが表示されます。
9. ログオフしてからログオンします。新規カスタム・ディメンションが表示されたことを確認します。

アプリケーションを開く

Financial Managementでは、アプリケーション内のすべてのデータが処理されます。一度に複数のアプリケーションを開いて作業できます。

アプリケーションにアクセスするには、アプリケーションのユーザーとして割り当てられる必要があります。

▶ アプリケーションを開くには:

1. 「ナビゲート」、「アプリケーション」、「連結」の順に選択します。
2. アプリケーションを選択します。



注:

アプリケーションがリストされていない場合、「リフレッシュ」をクリックします。

アプリケーションを閉じる

作業している現在のアプリケーションを閉じたり、複数のアプリケーションを開いている場合は同時にすべて閉じたりすることができます。

▶ アプリケーションを閉じるには、次のいずれかの操作を行います:

- ・ 「ファイル」、「閉じる」、「現在」の順に選択するか、「すべて」を選択します。
- ・ アプリケーションを開いているタブの上部の「X」をクリックします。

アプリケーション・サーバー・クラスタの変更

管理モジュールを使用して、アプリケーションを管理します。デフォルトでは、管理モジュールは、最初の使用可能なクラスタ/サーバーを使用します。管理モジュールを再ロードするこのオプションを使用して、接続したサーバーを変更できます。

▶ サーバーのクラスタを変更するには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「管理」、クラスタの変更の順に選択します。
3. クラスタ・サーバーリストから、クラスタを選択します。
4. 「OK」をクリックします。

アプリケーション・リストの表示

「アプリケーション」表には、システム内の使用可能なすべてのFinancial Managementアプリケーションのリストが含まれます。表には、アプリケーションの名前と説明が表示されます。

▶ アプリケーション・リストを表示するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「管理」、「アプリケーション」の順に選択します。

アプリケーションの削除



注意

この情報は、Financial Managementアプリケーションの管理のみに使用されません。Performance Management Architectを使用したアプリケーション管理については、『Oracle Hyperion Enterprise Performance Management Architect管理者ガイド』を参照してください。

アプリケーションを削除する前に、他のユーザーがアプリケーションを使用していないことを確認してください。

アプリケーションを削除するには、次のセキュリティ役割が必要です:

- ・ Shared Services: Financial Managementマネージャ、またはShared Services: Financial Management管理者およびShared Services: Financial Managementアプリケーション作成者
- ・ アプリケーション: アプリケーション管理者
- ・ アプリケーション: プロビジョニング・マネージャ

▶ 1つのアプリケーションを削除するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「管理」、「アプリケーション」の順に選択します。
3. 削除するアプリケーションを選択します。
4. 「削除」をクリックするか、「アクション」、「削除」の順に選択します。
5. アプリケーションが開いている場合、警告メッセージが表示されます。次のいずれかのオプションを選択します:
 - アプリケーションを強制削除するには、**強制削除**をクリックします。
ログインしているユーザーは、アプリケーションの削除後にエラー・メッセージを受信します。
 - アプリケーション・ユーザーを表示するには、**ユーザーの表示**をクリックします。「システム・ユーザー」ページから、ユーザーをログアウトできます。
 - アプリケーションの削除を取り消すには、「**取消し**」をクリックします。
6. ユーザーがアプリケーションを開いていない場合は、確認のプロンプトが表示されます。「はい」をクリックして、アプリケーションを削除します。

アプリケーション要素のロード

アプリケーションを作成したら、メタデータ、メンバー・リスト、ルール、およびセキュリティ・ファイルをロードします。ファイルを個々にロードすることも、ロードする複数のファイルを選択することも、すべてのファイルを一度にロードすることも可能です。複数のファイルをロードする場合は、正しい順序でロードされます。また、ファイルをスキャンして、ロード前に検証することもできます。

ロード・プロセスごとにログ・ファイルが生成されるため、それぞれのプロセスを個別に確認できます。ロード・プロセスが完了すると、ログのリンクが表示されるため、エラーを表示できます。正常に完了しないロード・ファイルがある場合は、エラーを修正して再ロードできます。

アプリケーション要素ファイルには特定のファイル・フォーマットを使用する必要があり、ロード・プロセスでいくつかのオプションを使用できます。次の項を参照してください:

- [58ページのアプリケーション・セキュリティのロード](#)
- [120ページのメタデータのロード](#)
- [137ページのメンバー・リストのロード](#)
- [248ページのルールのロード](#)
- [33ページのアプリケーション・モジュールの構成](#)

▶ アプリケーション要素をロードするには:

1. アプリケーションを開きます。
2. 「連結」、「ロード」、**アプリケーション要素**の順に選択します。
3. ロードするファイルの名前を入力するか、「参照」をクリックしてロードするファイルを検索します。
4. ロード・プロセスのオプションを指定します。
5. **オプション**: ファイル・フォーマットが正しいかどうかを確認するには、「スキャン」をクリックします。
6. 「ロード」をクリックして個々のファイルをロードするか、「すべてロード」をクリックします。

既存のファイルを再ロードすると、それらのファイルを再度使用かどうかを確認する警告プロンプトが表示されます。行う場合、「はい」をクリックします。



ヒント:

ファイル・オプションをデフォルト値にリセットするには、「リセット」または「すべてリセット」をクリックします。

アプリケーション要素の抽出

アプリケーション要素を抽出し、テキスト・エディタで情報を表示および編集して、アプリケーションにその要素を再ロードできます。これは、複数のファイルを同時に更新する必要がある場合に便利です。個々のファイルを抽出することも、抽出する複数のファイルを選択することも、すべてのファイルを一度に抽出することも可能です。

アプリケーション要素ファイルには特定のファイル・フォーマットを使用する必要があり、抽出プロセスでいくつかのオプションを使用できます。次の項を参照してください:

- [62ページのアプリケーション・セキュリティの抽出](#)
- [124ページのメタデータの抽出](#)
- [138ページのメンバー・リストの抽出](#)
- [250ページのルールの抽出](#)
- [33ページのアプリケーション・モジュールの構成](#)

抽出プロセスが完了すると、ログのリンクが表示されるため、エラーを表示できます。

▶ アプリケーション要素を抽出するには:

1. アプリケーションを開きます。
2. 「連結」、「抽出」、アプリケーション要素の順に選択します。
3. 抽出プロセスのオプションを指定します。
4. 「抽出」をクリックして個々のファイルを抽出するか、「すべて抽出」をクリックします。
5. 「ダウンロード」をクリックして、抽出したファイルをダウンロードします。

サンプル・アプリケーションの使用方法

Financial Managementにはサンプル・アプリケーション・ファイルが付属しています。これらのファイルは、テスト・アプリケーションの移入に使用できます。

テスト・アプリケーションを作成する場合、セキュリティ、メタデータ、データ、ルール、仕訳ファイル、レポート定義、データ・グリッド、データ・フォーム・スクリプトを含むサンプル・アプリケーションからファイルをロードできます。

表2 アプリケーション・ファイルのタイプ

サンプル・ファイル	コンテンツ
メンバー・リスト(.lst)	ディメンションのメンバー・リスト
メタデータ(.ads) (クラシック・アプリケーション管理用は.xml)	メタデータ

サンプル・ファイル	コンテンツ
データ(.dat)	該当するシナリオ、年、データ
ルール(.r1e)	アプリケーション・データのロジックの実行に使用するルール
仕訳(.j1f)	サンプルの仕訳ファイルとテンプレート・ファイル・フォーマット
システム・レポート(.rpt)	データ・グリッド、仕訳または内部取引のシステム・レポート
データ・グリッド・ファイル(*.xml)	データ・グリッド
データ・フォーム(.wdf)	データ・フォームのスクリプト
タスク・リスト(.xml)	タスク・リスト

システム・メッセージの使用

システム・メッセージ・ログには、Financial Managementのシステム・メッセージのリストが表示されます。ログから、メッセージの要約の表示、個々のメッセージの詳細の表示、およびメッセージの印刷を実行できます。エラー・メッセージには**X**が、情報メッセージには**i**が表示されます。システム・メッセージは、ログから削除できます。メッセージは、削除するまでログに残されます。

システム・メッセージを表示するには、アプリケーション・サーバーの構成時に指定する管理者グループのメンバーである必要があります。ユーザーを管理者グループに割り当てるには、*Oracle Enterprise Performance Management System*ユーザー・セキュリティ管理ガイドを参照してください。

次の手順を参照してください:

- [39ページのシステム・メッセージの表示](#)
- [40ページのシステム・メッセージの削除](#)

システム・メッセージの表示

▶ システム・メッセージを表示するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「管理」、「システム」、「メッセージ」の順に選択します。
3. オプション: 「フィルタ」パネルから、フィルタ条件を選択します:
 - アプリケーション
 - サーバー
 - 日付範囲
 - すべての日付を表示するには、「すべて含む」を選択します。
 - データ範囲を指定するには、「範囲」を選択し、「分」、「時間」、「日」または「月」を指定します。
 - 特定の日付を指定するには、「カスタム」を選択して開始および終了の日付を入力するか、カレンダー・アイコンをクリックして日付を選択します。
4. システム・メッセージのリストから、詳細を表示するメッセージを選択します。

5. メッセージをダブルクリックして開くか、「表示」をクリックするか、「アクション」、「表示」の順に選択します。
6. システム・メッセージの詳細を確認したら、「OK」をクリックします。

システム・メッセージの削除

管理者グループに割り当てられている場合、システム・メッセージ・ログからシステム・メッセージを削除できます。

▶ システム・メッセージを削除するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「管理」、「システム」、「メッセージ」の順に選択します。
3. システム・メッセージのリストから、削除するメッセージを1つ以上選択します。
4. 次のいずれかのアクションを行います:
 - 選択したメッセージを削除するには、「選択した項目の削除」をクリックするか、「アクション」、「選択した項目の削除」の順に選択します。
 - 現在のフィルタ設定に基づいてメッセージを削除するには、「現在のフィルタ設定に基づいて削除」をクリックするか、または「アクション」、「フィルタされた項目の削除」の順に選択します。
 - すべてのシステム・メッセージを削除するには、「すべて削除」をクリックするか、「アクション」、「すべて削除」の順に選択します。



ヒント:

メッセージ・リストをリフレッシュするには、「リフレッシュ」をクリックするか、「アクション」、「リフレッシュ」の順に選択します。

システム・メッセージの詳細文字列

一部のプロセスは、エラーなどのシステム・メッセージに関する技術情報の文字列を戻します。この文字列には、エラーを一意に識別する参照番号が含まれていて、その後に様々な情報フィールドが続きます。次の例のように、すべてのフィールドはセミコロンで区切られ、各フィールドにはラベルがあり、ラベルの後にはコロンが付いています。

```
Error Reference Number: {219EB33B-BF50-11D6-A43E-0000863DCCF1}
```

```
Num: 0x800415c6; Type: 1; DTime: 1/3/12 12:20:10 PM; Svr: SERVER1; File: CHsxServerImpl.cpp; Line: 1842; Ver: 3.0.0.196;
```

次の表に、各システム・メッセージ・フィールドの説明を示します。

表3 システム・メッセージの詳細文字列のフィールド

フィールド	説明
Num	エラー番号(16進形式)
Type	システム内部でのみ使用

フィールド	説明
DTime	エラーのタイムスタンプ
Svr	エラーが発生したコンピュータのマシン名
File	エラーに対応するソース・コード・ファイルの名前
Line	エラーに対応するソース・コード・ファイル内の行番号
Ver	エラーに対応するDLLのバージョン番号

システム・ユーザーの管理

システム・ユーザー機能によって、システムのユーザーを表示して、アプリケーションまたはサーバーからユーザーをログオフできます。どのユーザーがどのモジュールを使用しているか、どんなアクティビティが実行されているかを表示できます。[41ページのユーザーの表示](#)および[42ページのユーザーのログアウト](#)を参照してください。



注:

システム上にユーザーを表示するには、「Financial Management管理者」のセキュリティ役割が割り当てられている必要があります。

アプリケーション・サーバーの構成時に指定した管理者グループのメンバーのみが、ユーザーのログアウトを実行できます。ユーザーを管理者グループに割り当てるには、*Oracle Enterprise Performance Management Systemユーザー・セキュリティ管理ガイド*を参照してください。

ユーザーの表示

すべてのアプリケーションおよびサーバーにログオンしているユーザーを表示することや、リストをフィルタ処理して特定のサーバーおよびアプリケーションのユーザーを表示することが可能です。



注:

システム上にユーザーを表示するには、「Financial Management管理者」のセキュリティ役割が割り当てられている必要があります。

ログオンした各ユーザーについて、次の情報を入手できます。

- ユーザー名
- ユーザーが現在使用しているモジュール
- ユーザーの現在の操作
- 操作が開始された時刻
- ステータス
- サーバー名
- アプリケーション名

▶ ログオンしたユーザーを表示するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「管理」、「システム」、「ユーザー」の順に選択します。
3. オプション: 「フィルタ」パネルから、アプリケーションごと、またはサーバーごとにユーザーをフィルタ処理します。

ユーザーのログアウト

バックアップや復元などのシステム全体に影響を与えるプロセスを実行するために、アプリケーションまたはアプリケーション・サーバーからユーザーをログオフできます。たとえば、サーバーにログオンしているユーザーをログオフし、以降サーバーにログオンできないようにできます。[43ページの接続の使用不可と使用可能](#)を参照してください。

アプリケーション・サーバーの構成時に指定した管理者グループのメンバーのみが、ユーザーのログアウトを実行できます。

ユーザーをログアウトしても、すぐには切断されません - ユーザーが実行しているプロセスが終了する間に、5分間の潜在的な遅延があり、その後ユーザーがログアウトされます。

Microsoft Internet Information Services (IIS)のタイムアウト設定を変更するか、EPMコンフィグレータを使用してWebセッションのタイムアウト設定を変更し、ユーザーのセッションのタイムアウトを制御できます。デフォルト・タイムアウト設定は20分です。

システムからユーザーをログオフすると、ユーザーの停止済ステータスが表示されます。さらに、ログオフされたユーザーには、ログアウト後の最初のユーザー・アクションの直後に通知されます。

▶ アプリケーションまたはサーバーからユーザーをログオフするには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「管理」、「システム」、「ユーザー」の順に選択します。
3. ユーザーのリストから、ログアウトするユーザーを1つ以上選択します。
4. 次のいずれかのアクションを行います:
 - 「選択したユーザーのログアウト」をクリックするか、「アクション」、「選択したユーザーのログアウト」の順に選択します。
 - 「すべてのユーザーのログアウト」をクリックするか、「アクション」、「すべてのユーザーのログアウト」の順に選択します。



ヒント:

ユーザー・リストをリフレッシュするには、「リフレッシュ」をクリックするか、「アクション」、「リフレッシュ」の順に選択します。

サーバーとアプリケーションの管理

システムの管理機能を使用して、アプリケーションまたはサーバーへのユーザー接続を無効にすることや、ユーザーをログオフさせることができます。

43ページの接続の使用不可と使用可能および 42ページのユーザーのログアウトを参照してください。

アプリケーション・サーバーの構成時に指定した管理者グループのメンバーのみが、使用不可のコンポーネントを表示したり、接続を使用可能または使用不可にしたり、ユーザーをログアウトしたりできます。ユーザーを管理者グループに割り当てるには、*Oracle Enterprise Performance Management System*ユーザー・セキュリティ管理ガイドを参照してください。

接続の使用不可と使用可能

接続を使用不可にすると、指定したサーバーまたはアプリケーションに新規ユーザーがログオンできなくなります。接続を無効にする機能は、ユーザーのログアウト機能とともに使用できます。たとえば、アプリケーションへのログオンを無効にし、アプリケーションにログオンしているユーザーをログアウトし、メタデータをロードしてから、アプリケーションへの接続を有効にできます。



注:

管理者は、接続を使用不可にしている間にあらゆる管理作業(メタデータのロード、アプリケーションの削除など)を行えます。

- ▶ アプリケーションまたはサーバーへのユーザー接続を使用不可または使用可能にするには:
 1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
 2. 「管理」、「システム」、「管理」の順に選択します。
 3. オプション: リストされた接続の表示を変更するには、「表示」を選択してから、アイテムを展開、縮小、スクロールまたは表示するオプションを選択します。
 4. 次のいずれかのアクションを行います:
 - 「ユーザーのログアウト」をクリックするか、「アクション」、「ユーザーのログアウト」の順に選択します。
 - 「接続を無効にする」をクリックするか、「アクション」、「接続を無効にする」の順に選択します。
 - 「接続を有効にする」をクリックするか、「アクション」、「接続を有効にする」の順に選択します。
 5. 接続のリストをリフレッシュするには、「リフレッシュ」をクリックするか、「アクション」、「リフレッシュ」の順に選択します。

管理モードの有効化および無効化

管理者である場合、管理モードを有効化して、バックアップや復元操作などの管理メンテナンス・タスクの実行中にユーザーがアプリケーションにログインしないようにできます。

アプリケーションを管理モードに変更すると、すべてのユーザーはシステムからログアウトされます。保留中のタスク(連結、データ・ロード、メタデータ・ロードなど)がある場合、アプリケーションは、管理モードに変更されず、保留中のタスクが完了して停止するまで待機します。タスクが完了したら、アプリケーションの管理モードを有効化できます。

管理モードを無効化するまで、ユーザーはアプリケーションにログインできません。

- ▶ 管理モードを有効化および無効化するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「システム」、「アプリケーション」の順に選択します。
3. アプリケーションを選択します。
4. 次のいずれかのアクションを行います:
 - ツールバーから「管理モード」ボタンをクリックします。
 - 「アクション」、「管理モード」の順に選択します。
 - アプリケーション名を右クリックし、「管理モード」を選択します。
5. アプリケーションでの作業が完了したら、アプリケーションを選択し、管理モードを無効化します。

サーバーの同期化

Financial Managementアプリケーション・サーバー間の同期化はシステム時刻に基づきます。クロックの変更は、この同期化に影響します。夏時間(DST)への変更については、まずサーバーを停止してから時刻を変更し、それから再起動することをお勧めします。

タスクの監査

タスク監査機能を使用して、ユーザーが実行するタスクを表示できます。監査されるタスクを日付範囲、アプリケーション・サーバー、ユーザーおよび実行されるタスクでフィルタ処理できます。

「アプリケーション管理者」役割が割り当てられている場合、タスク監査情報を表示、エクスポートおよび削除できます。管理者ではないが、「タスク監査の表示」役割がある場合、タスク監査情報を表示およびエクスポートできます。

タスク監査では、ユーザーが行った次の操作がログに保存されます。

- アイドル
- ルールのロード
- ルールのスキャン
- ルールの抽出
- 連結
- 記帳ロジック(計算)
- 換算
- カスタム・ロジック
- 配賦
- データのロード
- データの抽出
- データの入力
- データの取得
- データのクリア
- データのコピー
- 仕訳の入力

- 仕訳の取得
- 仕訳の転記
- 仕訳の転記の戻し
- 仕訳テンプレートの入力
- メタデータのロード
- メタデータの抽出
- メンバー・リストのロード
- メンバー・リストのスキャン
- メンバー・リストの抽出
- セキュリティのロード
- セキュリティのスキャン
- セキュリティの抽出
- ログオン
- ログオンの失敗
- ログオフ
- メタデータのスキャン
- データのスキャン
- データベースへのデータの抽出のエクスポート
- データベースへのデータの抽出のスキーマ削除
- 取引のロード
- 取引の抽出
- ドキュメントの添付ファイル
- ドキュメントの添付を解除する
- 取引の作成
- 取引の編集
- 取引の削除
- 取引の転記
- 取引の転記の戻し
- 無効なレコードの削除
- データ監査の除去
- タスク監査の除去
- すべての取引の転記
- すべての取引の転記の戻し
- すべての取引の削除
- すべての取引の照合を解除
- IDによる自動照合
- 勘定科目による自動照合
- IDによる内部取引照合レポート
- 勘定科目による内部取引照合レポート

- ・ 内部取引レポート

タスク監査ログには次の情報が含まれます。

- ・ ユーザー名
- ・ 実行した操作
- ・ 操作の開始時刻
- ・ 操作の終了時刻
- ・ サーバー名
- ・ 説明
- ・ 現在のモジュール

タスク監査ログ情報は(*APPNAME_TASK_AUDIT*)表に保管されます。表内の情報をファイルにバックアップまたは抽出してから、表をクリアします。ログ・サイズをモニターして定期的にクリアする必要があります。

▶ タスク監査を実行するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「管理」、「監査」、「タスク」の順に選択します。
3. オプション: 「フィルタ」パネルから、フィルタ条件を選択します:
 - ・ アプリケーション
 - ・ サーバー
 - ・ 日付範囲
 - すべての日付を表示するには、「すべて含む」を選択します。
 - データ範囲を指定するには、「範囲」を選択し、「分」、「時間」、「日」または「月」を指定します。
 - 特定の日付を指定するには、「カスタム」を選択して開始および終了の日付を入力するか、カレンダー・アイコンをクリックして日付を選択します。
 - ・ ユーザー
 - ・ タスク
4. オプション: 監査情報をcsvファイルにエクスポートするには、「エクスポート」をクリックするか、「アクション」、「エクスポート」の順に選択して、ダウンロードの指示に従います。
5. オプション: 現在のフィルタ設定に基づいてエンTRIESを削除するには、「現在のフィルタ設定に基づいて削除」をクリックするか、または「アクション」、「フィルタされた項目の削除」の順に選択します。
6. オプション: ログからすべてのエンTRIESを削除するには、「すべて削除」をクリックするか、「アクション」、「すべて削除」の順に選択します。



注:

ログをクリアすると、クリア処理の記録がログに残り、これをクリアすることはできません。

データの監査

データ監査機能を使用して、ユーザーが実行したデータの変更を確認できます。データの変更を、日付範囲、アプリケーション・サーバー、ユーザーおよびディメンション・メンバー別にフィルタ処理できます。

メタデータ・マネージャでは、データ変更を監査する勘定科目およびシナリオのEnableDataAuditメタデータ属性を有効にできます。シナリオの監査設定は、勘定科目の監査設定を上書きします。シナリオのEnableDataAudit属性が「はい」に設定されていると、EnableDataAuditが「FALSE」に設定されている勘定科目に対しても、シナリオのすべての勘定科目が監査されます。シナリオのEnableDataAuditが「上書き」に設定されている場合、EnableDataAuditが「TRUE」に設定されているすべての勘定科目が監査されます。シナリオおよび勘定科目のメンバーの監査を無効にするには、EnableDataAudit属性を「いいえ」に変更します。

「アプリケーション管理者」役割が割り当てられている場合、データ監査情報を表示、エクスポートおよび削除できません。管理者ではないが、「データ監査の表示」役割がある場合、データ監査情報を表示およびエクスポートできます。

データ監査では、ユーザーが行った次の操作がログに保存されます。

- データの入力
- データのクリア
- データのコピー
- データのロード
- 仕訳の入力

データ監査ログには次の情報が含まれます。

- ユーザー名
- 実行した操作
- 変更の時刻
- サーバー名
- データの視点
- 視点に入力された値

データ監査ログ情報はAPPNAME_DATA_AUDIT表に保管されます。表内の情報をバックアップまたは抽出できます。ログ・サイズをモニターして定期的クリアする必要があります。

▶ データ監査を実行するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「管理」、「監査」、「データ」の順に選択します。
3. オプション: 「視点」から、ディメンション・メンバーを選択するディメンションをクリックします。
4. オプション: 「フィルタ」パネルから、フィルタ条件を選択します:
 - アプリケーション
 - サーバー
 - 日付範囲
 - すべての日付を表示するには、「すべて含む」を選択します。
 - データ範囲を指定するには、「範囲」を選択し、「分」、「時間」、「日」または「月」を指定します。
 - 特定の日付を指定するには、「カスタム」を選択して開始および終了の日付を入力するか、カレンダー・アイコンをクリックして日付を選択します。
 - ユーザー
5. オプション: 監査情報をcsvファイルにエクスポートするには、「エクスポート」をクリックするか、「アクション」、「エクスポート」の順に選択して、ダウンロードの指示に従います。

6. オプション: 現在のフィルタ設定に基づいてエントリを削除するには、「現在のフィルタ設定に基づいて削除」をクリックするか、または「アクション」、「フィルタされた項目の削除」の順に選択します。
7. オプション: ログからすべてのエントリを削除するには、「すべて削除」をクリックするか、「アクション」、「すべて削除」の順に選択します。



注:

ログをクリアすると、タスク監査ログにクリア処理の記録が残り、これをクリアすることはできません。

実行中のタスクのモニタリング

実行中のタスク・モジュールを使用して、実行中のタスクを表示および終了できます。実行中のタスクをアプリケーション、サーバー、ユーザー、実行されたタスクおよびタスク・ステータス別にフィルタ処理できます。

デフォルトで、実行中のタスクはデータベースに900秒(15分)間残ります。AutoClearDeadTasksAfterSecondsの設定を変更して、デフォルトの時間を変更できます。[465ページの付録A「構成設定」](#)を参照してください。

- [48ページの実行中のタスクの表示](#)
- [49ページの実行中のタスクの停止](#)
- [49ページの実行中のタスクのリフレッシュ](#)

実行中のタスクの表示

▶ 実行中のタスクを表示するには:

1. 次のいずれかのアクションを行います:
 - アプリケーションから、「連結」、「メンテナンス」、「タスクの実行」の順に選択します。
 - 「ナビゲート」、「管理」、「連結管理」、「管理」、「システム」、「タスクの実行」の順に選択します。
2. オプション: 「フィルタ」パネルから、フィルタ条件を選択します:
 - アプリケーション
 - サーバー
 - ユーザー
 - タスク
 - ステータス

次のタスクを表示および終了できます。

- 連結
- データのロード
- データの抽出
- データベースへのデータの抽出のレポート
- すべての取引の転記

- すべての取引の転記の戻し
- すべての取引の削除
- すべての取引の照合を解除
- IDによる自動照合
- 勘定科目による自動照合
- 内部取引照合レポート
- IDによる内部取引照合レポート
- 勘定科目による内部取引照合レポート
- 内部取引レポート
- 仕訳レポート

実行中のタスクを次のステータス・タイプ別にフィルタ処理できます:

- 初期化中
- 実行中
- 一時停止
- 初期化解除中
- 停止済
- 中止済
- 完了
- 応答していません
- 中止(スケジュールによる)
- 開始(スケジュールによる)

実行中のタスクの停止

実行中のタスクを終了できるのは、そのタスクを開始したユーザーまたは「管理者」の役割を割り当てられたユーザーに限られます。

▶ タスクを停止するには:

1. 「タスクの実行」モジュールから、停止するタスクを選択します。
2. 「タスクの中止」をクリックするか、「アクション」、「タスクの中止」の順に選択します。

実行中のタスクのリフレッシュ

データや内部取引ファイルを複数または大量にロードする場合、実行中のタスク・モジュールに、タスクがすぐに表示されないことがあります。実行中のタスクを開くと、データや内部取引のロードでファイルの転送に遅延が発生した場合には、待機するようメッセージが表示され、画面が自動的にリフレッシュされて、タスクが表示されます。デフォルトでは、「タスクの実行」画面が5秒間隔で自動的にリフレッシュされ、自動リフレッシュ・プロセスが最大120回実行されます。120回実行しても実行中のタスクが検出されない場合は、リフレッシュ・サイクルが停止し、手動でのリフレッシュが必要になります。リフレッシュ・サイクルは、実行中のタスクがすべて完了するまで続行されます。



注:

Financial Managementのシステム管理者である場合は、デフォルトのリフレッシュ件数とリフレッシュ間隔を構成できます。[470ページの構成設定の変更](#)を参照してください。

▶ 「タスクの実行」リストをリフレッシュするには:

1. タスク・リストから、「リフレッシュ」をクリックするか、「アクション」、「リフレッシュ」の順に選択します。
2. ドロップダウン・リストから、リフレッシュ・モードを選択します:

- デフォルト
- 手動
- 5秒
- 10秒
- 15秒
- 30秒
- 60秒



注:

リフレッシュ・モードの設定は、次回アプリケーションにログオンするまで保持されます。

無効なレコードのスキャンとクリア

無効なレコードのクリア機能を使用して、アプリケーションに無効なレコードがないかスキャンし、削除できます。無効なレコードをクリアするには、管理者のセキュリティ・アクセス権が必要です。

無効なレコードのクリア・プロセスを実行すると、データベース、ネットワークおよびFinancial Management環境に影響があり、性能上の問題を引き起こす場合があります。このプロセスは、ユーザーがFinancial Management環境にアクセスしないメンテナンス時に実行する必要があります。

▶ 無効なレコードをスキャンして削除するには:

1. 無効なレコードを削除するアプリケーションを開きます。
2. 「連結」、「データ」、「管理」の順に選択します。
3. 「データの管理」ページから、「無効なレコードのクリア」を展開します。



注:

この機能を使用できるのは、管理者のセキュリティ・アクセス権のあるユーザーのみです。

4. 次のいずれかのアクションを選択します:

- 「スキャン」をクリックし、無効なレコードをスキャンします。
- 「無効なレコードのクリア」をクリックして、レコードをクリアします。

3

アプリケーション・セキュリティの管理

この項の内容:

アプリケーション・セキュリティに関する考慮事項	52
Financial ManagementからのShared Services Consoleの起動	52
アプリケーション・セキュリティのロード	58
アプリケーション・セキュリティの抽出	62

セキュリティおよびアクセス権限を使用すると、Financial Managementのアプリケーションとアプリケーション要素へのアクセスを制御できます。セキュリティを設定すると、データを保護し、許可されていないユーザーによるデータの変更を防止できます。たとえば、アプリケーション内の特定のデータ要素およびフォームへのアクセスを制限できます。

セキュリティには次の2つのレベルがあります。

- 外部プロバイダによる認証。
- Financial Managementセキュリティ。このレベルでは、ユーザーやグループがアプリケーションに割り当てられ、アプリケーション要素はセキュリティ・クラスに割り当てられます。

Financial Managementアプリケーションのセキュリティは、次の2つの方法で設定できます。

- セキュリティ・ファイルをアプリケーションにロードします。58ページの[アプリケーション・セキュリティのロード](#)を参照してください。
- Oracle Hyperion Shared Services Consoleを使用して、セキュリティ情報を設定します。52ページの[Financial ManagementからのShared Services Consoleの起動](#)を参照してください。

アプリケーション管理には、次のセキュリティ役割が必要です。役割の割当てについては、*Oracle Enterprise Performance Management Systemユーザー・セキュリティ管理ガイド*を参照してください。

セキュリティ役割	説明
Financial Management管理者	管理機能にアクセスするために必要です。
ディメンション・エディタ	手動でディメンションを作成および管理するだけでなく、ディメンション作成用のインポート・プロファイルを作成および管理します。 クラシック・アプリケーション管理オプションにアクセスする必要があります。
Financial Managementアプリケーション作成者/Financial Managementマネージャ	アプリケーションを作成します。この役割を持つユーザーはアプリケーションを作成できますが、アクセス権限のあるディメンションしか変更できません。 「アプリケーション作成者」の役割を持つユーザーがPerformance Management Architectからアプリケーションを配置する場合、その

セキュリティ役割	説明
	<p>ユーザーは自動的に、該当するアプリケーションのアプリケーション管理者とプロビジョニング・マネージャになります。「アプリケーション作成者」はすべてのアプリケーションを作成できます。</p> <p>「Financial Managementアプリケーション作成者」は、連結アプリケーションと汎用アプリケーションを作成できます。アプリケーションを作成するには、構成ユーティリティで指定されたアプリケーション作成者グループのメンバーであることも必要です。</p>

アプリケーション・セキュリティに関する考慮事項

Financial Managementセキュリティでは、アプリケーション要素とタスクのセキュリティに柔軟性を実現しています。セキュリティ・クラスは作成時にアプリケーション要素に割り当てられるため、アプリケーションを設定する前にセキュリティ・システムを設計する必要があります。

あるアプリケーションのセキュリティ・システムを設計したら、バックアップまたは別のアプリケーションにロードするために、セキュリティ要素を抽出できます。[58ページのアプリケーション・セキュリティのロード](#)および[62ページのアプリケーション・セキュリティの抽出](#)を参照してください。

Financial Managementで、セキュリティを設定する前に、次の問題を検討する必要があります。

- Financial Managementのタスクとアプリケーション要素をどのように分類しますか？
- ユーザーをどのようにグループ化しますか？
- ユーザーやグループに、どのレベルのアクセス権を設定しますか？
- アプリケーションの要素を作成するときに、どのセキュリティ・クラスを割り当てますか？

Financial ManagementからのShared Services Consoleの起動

Financial Managementのアプリケーションのセキュリティを設定する前に、次のタスクを実行する必要があります。

1. Financial Managementのアプリケーションを作成します。クラシックFinancial Managementアプリケーションについては、[31ページのアプリケーションの作成](#)を参照してください。Performance Management Architectアプリケーションについては、『Oracle Hyperion Enterprise Performance Management Architect, Fusion Edition管理者ガイド』を参照してください。
2. ユーザーをプロビジョニングするには、ユーザーおよびグループをアプリケーションに割り当て、ユーザーに役割を割り当てます。『Oracle Enterprise Performance Management Systemユーザー・セキュリティ管理ガイド』を参照してください。

次に、Shared Services Consoleを使用して、Financial Managementのアプリケーションのセキュリティを設定します。コンソールで、アプリケーションの次のタスクを実行できます。

- ユーザーとグループの割当て
- セキュリティ・クラスへのユーザー権限の割当て

- セキュリティ・レポートの実行

▶ Shared Services Consoleを起動するには:

1. Financial Managementから、「ナビゲート」、「管理者」の順に選択します。
2. 「**Shared Services Console**」を選択します。

セキュリティ・クラスを割り当てるユーザーおよびグループの選択

プロビジョニング・マネージャの役割に割り当てられたユーザーのみがユーザーおよびグループを定義できます。『Oracle Enterprise Performance Management Systemユーザー・セキュリティ管理ガイド』を参照してください。

デフォルトで、「アクセス・コントロール」ページにプロビジョニングされたグループおよびユーザーの最初の100個が表示されます。表内で、最初にグループが表示され、続いてユーザーが表示されます。ユーザーとグループは、表内のアイコンで識別できます。

▶ アプリケーションのユーザーとグループを選択するには:

1. Shared Services Consoleで、「アプリケーション・グループ」を展開し、アプリケーション名を右クリックして「アクセス権の割当て」を選択します。
2. 「ユーザー/グループ」を選択するか、または「アクション」、「ユーザー/グループ」の順に選択します。
3. 「ユーザー」または「グループ」のいずれかのオプションを選択します。
4. 「使用可能なユーザー」または「使用可能なグループ」で、アプリケーションに割り当てるユーザーおよびグループを選択し、矢印キーを使用して、それらを「選択したユーザー」または「選択したグループ」列に移動します。
5. 「OK」をクリックします。

セキュリティ・クラスの設定

セキュリティ・クラスによって、アプリケーション要素に対するユーザーとグループのアクセス権が決まります。



注意

このセクションの情報は、クラシックFinancial Managementアプリケーション専用です。Performance Management Architectを使用したセキュリティの設定については、『Oracle Hyperion Enterprise Performance Management Architect管理者ガイド』を参照してください。

次の手順を実行できます:

- [54ページのセキュリティ・クラスの作成](#)
- [54ページのセキュリティ・クラスの削除](#)
- [55ページのセキュリティ・クラスの選択](#)



注:

「プロビジョニング・マネージャ」の役割に割り当てられているユーザーのみが、アプリケーションのセキュリティ・クラスを定義できます。

アプリケーションのセキュリティ・クラスを定義すると、これらのセキュリティ・クラスを勘定科目やエンティティなどのアプリケーション要素に割り当てることができるようになります。

ユーザーやグループがアプリケーション要素にアクセスできるかどうかは、ユーザーやグループが属しているセキュリティ・クラスやアプリケーション要素に関連付けられているセキュリティ・クラスによって異なります。

システム生成のセキュリティ・クラスは[Default]と呼ばれ、アプリケーションの一部として作成されます。ユーザーが削除することや、変更することはできません。[Default]セキュリティ・クラスにはアクセス権を割り当てられます。セキュリティ・クラスを割り当てられていないメンバーは、Defaultセキュリティ・クラスを持っているものとして処理されます。

セキュリティ・クラスの作成

▶ セキュリティ・クラスを作成するには:

1. Shared Services Consoleで、「アプリケーション・グループ」を展開し、アプリケーション名を右クリックして「アクセス権の割当て」を選択します。
2. 「セキュリティ・クラスの追加」を選択するか、または「アクション」、「セキュリティ・クラスの追加」の順に選択します。
3. 「名前」に、セキュリティ・クラスの名前を入力します。

名前は、一意で、最大80文字までにする必要があります。空白を含めることができますが、空白で開始することはできません。次の文字は使用できません:

- アスタリスク(*)
- アット記号(@)
- カンマ(,)
- 中カッコ({})
- 二重引用符(")
- マイナス記号(-)
- シャープ(#)
- ピリオド(.)
- プラス記号(+)
- セミicolon(;)
- スラッシュ(/)

4. 「OK」をクリックします。

セキュリティ・クラスの削除

セキュリティ・クラスが不要になったら、セキュリティ・クラス・ディメンションから削除できます。セキュリティ・クラスをアプリケーションから削除する前に、そのセキュリティ・クラスとアプリケーション要素の関連付けを解除する必要があります。

エンティティ、勘定科目またはシナリオの関連付けをセキュリティ・クラスから解除するには、メタデータ・ファイルのセキュリティ・クラスを変更します。仕訳の関連付けをセキュリティ・クラスから解除するには、仕訳ファイルを変更するか、仕訳の処理モジュールで仕訳のセキュリティ・クラスを更新します。

▶ セキュリティ・クラスを削除するには:

1. Shared Services Consoleで、「アプリケーション・グループ」を展開し、アプリケーション名を右クリックして「アクセス権の割当て」を選択します。
2. セキュリティ・クラスのリストから、表のビューに基づくセキュリティ・クラスの行/列ヘッダーを選択します。



ヒント:

複数のセキュリティ・クラスを削除するには、[Ctrl]キーを使用して行/列ヘッダーを選択します。

3. 「セキュリティ・クラスの削除」を選択するか、または「アクション」、「セキュリティ・クラスの削除」の順に選択します。
4. 「はい」をクリックして、削除を確認します。

セキュリティ・クラスの選択

デフォルトで、「セキュリティ・クラス」ページに、使用可能なセキュリティ・クラスが表示され、最初の100個のセキュリティ・クラスが表示されます。

▶ アプリケーションのセキュリティ・クラスを選択するには:

1. Shared Services Consoleで、「アプリケーション・グループ」を展開し、アプリケーション名を右クリックして「アクセス権の割当て」を選択します。
2. 「セキュリティ・クラス」を選択するか、または「アクション」、「セキュリティ・クラス」の順に選択します。
3. 「使用可能なセキュリティ・クラス」で、アプリケーションに割り当てるセキュリティ・クラスを選択し、矢印キーを使用して、それらを「選択されたセキュリティ・クラス」列に移動します。
4. 「OK」をクリックします。

セキュリティ・クラスへのユーザー・アクセス権の割当て

ユーザーとグループを定義し、セキュリティ・クラスを作成したら、アプリケーションの各セキュリティ・クラスに対する各ユーザーとグループのアクセス・タイプを指定できます。

ユーザーに、次の5つのアクセス・タイプのいずれかを割り当てられます: なし、メタデータ、読取り、レベルを上げる、またはすべて。ピボット機能を使用して、アクセス権の割当て表の2つのビューを切り替えられます。たとえば、ユーザーとグループが行にあり、セキュリティ・クラスが列にある場合、「ピボット」をクリックすると、ユーザーとグループが列に、セキュリティ・クラスが行に移動します。

ユーザーにセキュリティ・クラスに対するアクセス権を付与したら、電子メール・アラートを有効にして、内部取引レポートに使用することや、そのセキュリティ・クラスを使用するエンティティやシナリオのプロセス単位のステータスが変更されたことをユーザーに通知することができます。プロセス・コントロールの電子メール・アラートを受信するには、ユーザーにセキュリティ・クラスに対するすべてのアクセス権またはレベルを上げるアクセス権が必要です。



注:

「アプリケーション管理者」の役割に割り当てられているユーザーは、アプリケーションのすべての情報にアクセスできます。

表4 ユーザーのアクセス・レベル

アクセス・レベル	説明
なし	セキュリティ・クラスに割り当てられた要素にアクセスできません。
メタデータ	指定したメンバーをリストに表示できますが、メンバーのデータを表示したり変更したりすることはできません。
読取り	セキュリティ・クラスに割り当てられた要素のデータを表示できますが、レベルを上げたり却下したりできません。
レベルを上げる	セキュリティ・クラスに割り当てられた要素のデータを表示し、レベルを上げたり却下したりできます。
すべて	セキュリティ・クラスに割り当てられた要素のデータを変更し、レベルを上げたり却下したりできます。

▶ セキュリティ・クラスに対するユーザー・アクセス権を割り当てるには:

1. Shared Services Consoleで、「アプリケーション・グループ」を展開し、アプリケーション名を右クリックして「アクセス権の割当て」を選択します。
2. アクセス権を割り当てるセルを選択します。



ヒント:

複数のセルを選択するには、[Shift]キーと[Ctrl]キーを使用します。1つの列または行を選択するには、列または行ヘッダーをクリックします。列と行の表示を変更するには、「ピボット」をクリックします。

3. 右クリックし、割り当てるアクセス・レベルを選択します。



注:

56ページの表 4を参照してください。

- ・ なし
 - ・ メタデータ
 - ・ 読取り
 - ・ レベルを上げる
 - ・ すべて
4. オプション: 電子メール・アラートを追加するには、表内のセルを選択し、右クリックして「電子メール・アラートの有効化」を選択します。



注意

アラート・プロセスでは、MSAD、LDAP、Native Directoryなどの認証ファイルに保管された電子メール・アドレスを使用します。



注:

電子メール・アラートを除去するには、セルを選択して「電子メール・アラートの無効化」をクリックします。

5. 「保存」をクリックします。

電子メール・アラートの設定

内部取引とプロセス管理確認プロセスで電子メール・アラートを使用できます。電子メール・アラートを使用すると、主要なイベントまたはシステム内でのデータの変更をハイライトできます。たとえば、内部取引が不一致であるため一致させる必要があること、またはプロセス単位を次のレベルに上げられることを電子メール・アラートで送信できます。



注:

アラート・プロセスでは、LDAP、MSAD、Native Directoryなど、外部の認証プロバイダに保管された電子メール・アドレスが使用されます。

アプリケーション管理の役割を持つユーザーは、電子メール・アラートを自動的に受信しません。アプリケーション管理者の役割を持つユーザーが電子メール・アラートを受信するには、そのユーザーを個別ユーザーとして設定し、アラートを受信するセキュリティの役割を割り当てます。

セキュリティ・レポートの実行

アプリケーションに対するセキュリティの設定時に選択した情報に基づいて、セキュリティ・レポートを実行できます。ユーザー別のクラス、ユーザー別の役割、ユーザー別のクラスと役割、およびグループ別のユーザーに対してレポートを実行できます。レポートはオンラインで表示できますが、ファイルへのエクスポートもできます。

▶ セキュリティ・レポートを作成するには:

1. Oracle Hyperion Shared Services Consoleで、「アプリケーション・グループ」を展開し、アプリケーション名を右クリックして「アクセス権の割当て」を選択します。
2. 「セキュリティ・レポート」を選択するか、または「アクション」、「セキュリティ・レポート」の順に選択します。
3. レポート・オプションを選択します。
 - 「アクセス権」を選択し、次のオプションを選択します:
 - クラス(ユーザー別)
 - 役割(ユーザー別)
 - 「グループ別のユーザー」
4. 「レポートのフォーマット」で、次から選択します: PDF、RTF、HTML、XLS、XLSX
5. オプション: 「テンプレート」を選択します。
6. 次のいずれかのオプションを選択します:

- 新規ページでレポートを開くには、「レポートの開始」
- 選択したレポート・ファイル・フォーマットでレポートを保存するには、「ファイルへエクスポート」

アプリケーション・セキュリティのロード



注意

セキュリティ・クラスをロードできるのは、クラシックFinancial Managementアプリケーションに対してのみです。

アプリケーションに情報をロードする前に、アプリケーション・セキュリティをロードする必要があります。複数のアプリケーション要素を一度にロードする場合は、まず、セキュリティ・ファイルがロードされます。

セキュリティ情報ロード・ファイルには、ASCIIまたはUnicodeフォーマットを指定できます。セキュリティ情報ロード・ファイルのデフォルトのファイル名拡張子はSECです。

ユーザー、セキュリティ・クラス、役割のアクセス権およびセキュリティ・クラスのアクセス権をロードできます。

ユーザーまたはグループから役割を除去するには、Shared Services Consoleで役割を変更する必要があります。『Oracle Enterprise Performance Management System-ユーザー・セキュリティ管理ガイド』を参照してください。



注:

Webのポップアップ・ブロックの例外にFinancial Managementを追加することをお勧めします。データのロードなどのいくつかのタスクを実行すると、ステータス・ウィンドウにタスクのステータスが表示されます。コンピュータでポップアップ・ブロックを使用可能にしている場合、このステータス・ウィンドウは表示されません。

▶ アプリケーション・セキュリティをロードするには:

1. アプリケーションを開きます。
2. 「連結」、「ロード」、アプリケーション要素の順に選択します。
3. 「セキュリティ」セクションにロードするファイル名を入力するか、「参照」をクリックしてファイルを検索します。



注:

デフォルトでは、アプリケーション・セキュリティ情報ファイルにSECファイル拡張子が使用されます。ロード・プロセスでは、TXTやCSVなどの他のファイル拡張子も使用できますが、SECファイル拡張子の使用をお勧めします。

4. オプション: 新規セキュリティ情報をロードする前に、アプリケーションのセキュリティ情報をクリアするには、「すべて選択解除」を選択します。



注意

「すべてクリア」オプションを使用できるのは、アプリケーション管理者およびプロビジョニング・マネージャの役割が割り当てられている場合のみです。また、クリアを行うユーザーも含め、すべてのユーザーがこのプロセスで除去されるため、このオプションを使用する場合は、ユーザーを再プロビジョニングする必要があります。ユーザーのプロビジョニングについては、*Oracle Enterprise Performance Management System ユーザー・セキュリティ管理ガイド*を参照してください。クリアオプションを選択する前に、[59ページのセキュリティ情報のクリアとロード](#)を再確認してください。

5. 「区切り文字」から、ファイル内の情報を区切るために使用する文字を選択します。次の文字が有効です:

, ~ @ \$ % ^ & | : ; ? \



注:

ファイル名や、ファイル内のその他の要素で使用されていない文字を使用してください。たとえば、エンティティの説明にカンマを使用する場合、カンマを区切り文字として使用できません。

6. 「フィルタ」セクションで、ロードするセキュリティ情報のタイプを選択します。



ヒント:

フィルタの選択をリセットするには、「リセット」をクリックします。

7. 「ロード」をクリックします。

セキュリティ情報のクリアとロード

セキュリティ・ファイルのロードを開始する前に、アプリケーションからセキュリティ情報をクリアして、新しいセキュリティ情報をロードできます。たとえば、セキュリティのロード時にセキュリティ・クラスClass1をClass2に変更する場合は、Class1のセキュリティ・クラスを参照するすべてのアプリケーション要素を変更する必要があります。

セキュリティ・クラス情報を使用しているアプリケーション要素に対して新しいセキュリティ参照が生成されるため、新しいセキュリティ情報をロードする前に事前の手順を実行し、ロード後に事後の手順を実行する必要があります。

▶ セキュリティ情報をクリアして、新しいセキュリティ・ファイルをロードするには:

1. アプリケーションからアプリケーション要素を抽出します。[60ページのセキュリティ情報をクリアする前にしておくべきタスク](#)を参照してください。
2. 既存のセキュリティ情報をクリアして、新しいセキュリティ・ファイルをロードするように指定します。
3. アプリケーションにアプリケーション要素をロードします。[60ページのセキュリティ情報をクリアした後で行うタスク](#)を参照してください。



注:

これらの手順を行うには、「アプリケーション管理者」のセキュリティ役割が割り当てられている必要があります。

セキュリティ情報をクリアする前に行っておくべきタスク

セキュリティ情報をクリアして、セキュリティ・ファイルをロードする前に、セキュリティ・クラス情報を使用する指定のアプリケーション要素に対して次のタスクを実行する必要があります。

メタデータ

- ▶ セキュリティ情報をクリアしてロードする前にメタデータを更新するには:
 1. アプリケーションのすべてのメタデータ要素を抽出します。
 2. 必要に応じて、メタデータ要素のセキュリティ・クラス情報を変更します。

仕訳

- ▶ セキュリティ情報をクリアしてロードする前に仕訳を更新するには:
 1. アプリケーション内の転記済仕訳の転記をすべて戻します。
 2. 承認済の仕訳をすべて却下し、仕訳ステータスを「作業中」に戻します。
 3. すべての仕訳を抽出します。
 4. 必要に応じて、仕訳のセキュリティ・クラス情報を変更します。

グリッド

- ▶ セキュリティ情報をクリアしてロードする前にグリッドを更新するには:
 1. セキュリティ・クラスが割り当てられているすべてのグリッドを抽出します。
 2. 必要に応じて、グリッドのセキュリティ・クラス情報を変更します。

データ・フォーム

- ▶ セキュリティ情報をクリアしてロードする前にデータ・フォームを更新するには:
 1. セキュリティ・クラスが割り当てられているデータ・フォームをすべて抽出します。
 2. 必要に応じて、データ・フォームのセキュリティ・クラス情報を変更します。

セキュリティ情報をクリアした後で行うタスク

セキュリティ情報をクリアして、セキュリティ・ファイルをロードした後は、セキュリティ・クラス情報を使用する指定のアプリケーション要素に対して次のタスクを実行する必要があります。

メタデータ

▶ メタデータを更新するには:

1. メタデータ情報がクリアされていることを確認します。
2. 更新されたメタデータ・ファイルをアプリケーションにロードします。

仕訳

▶ セキュリティ情報をクリアしてロードした後に仕訳を更新するには:

1. 更新された仕訳ファイルをロードします。
2. セキュリティ情報をクリアしてロードする前に転記を戻した仕訳を転記します。
3. セキュリティ情報をクリアしてロードする前に却下した仕訳を承認します。

グリッド

▶ セキュリティ情報をクリアしてロードした後にグリッドを更新するには:

1. 更新されたグリッド・ファイルをロードします。
2. 既存のドキュメントを上書きするオプションを選択します。

データ・フォーム

▶ セキュリティ情報をクリアしてロードした後にデータ・フォームを更新するには:

1. 更新されたデータ・フォーム・ファイルをロードします。
2. 既存のドキュメントを上書きするオプションを選択します。

フォルダ

▶ セキュリティ情報をクリアしてロードした後にフォルダを更新するには:

1. 間違ったセキュリティ・クラスが割り当てられている可能性があるフォルダを削除します。
2. 新しいフォルダをアプリケーションに追加します。

レポート

▶ セキュリティ情報をクリアしてロードした後にレポートを更新するには:

1. セキュリティ・クラスが割り当てられているすべてのレポートを再ロードします。
2. 必要に応じて、新しいセキュリティ・クラスを割り当てます。

タスク・リスト

▶ セキュリティ情報をクリアしてロードした後にタスク・リストを更新するには:

1. セキュリティ・クラスが割り当てられているすべてのタスク・リストを再ロードします。
2. 必要に応じて、新しいセキュリティ・クラスを割り当てます。

アプリケーション・セキュリティの抽出

アプリケーション・セキュリティを抽出して、テキスト・エディタで表示または変更できます。アプリケーションからアプリケーション・セキュリティを抽出する際は、マルチバイト文字セット(MBCS)をサポートするフォーマットでファイルを保存します。デフォルトでは、アプリケーション・セキュリティ・ファイルでは、ファイル拡張子SECが使用されます。

次のタイプのセキュリティ情報を抽出できます。

- ユーザーとグループ
- セキュリティ・クラス
- 役割アクセス
- セキュリティ・クラスのアクセス権



注:

セキュリティをバックアップ・ファイルに定期的に抽出することをお勧めします。セキュリティ情報のバックアップについては、『Oracle Enterprise Performance Management Systemインストールおよび構成ガイド』を参照してください。

▶ アプリケーション・セキュリティを抽出するには:

1. アプリケーションを開きます。
2. 「連結」、「抽出」、アプリケーション要素の順に選択します。
3. 「セキュリティ」セクションで、「区切り文字」から、ファイル内の情報を区切るために使用する文字を選択します。

次の文字が有効です:

, ~ @ \$ % ^ & | : ; ? \



注:

ファイル名や、ファイル内のその他の要素で使用されていない文字を使用してください。たとえば、エンティティの説明にカンマを使用する場合、カンマを区切り文字として使用できません。

4. 「フィルタ」で、抽出するセキュリティのタイプを選択します。



ヒント:

選択をリセットするには、「リセット」をクリックします。

5. 「抽出」をクリックします。
6. ブラウザに表示されるダウンロードの手順に従います。

使用しているWebブラウザによって、手順が異なります。ファイルは、設定したWebディレクトリに保存してください。

4

メタデータの管理

この項の内容:

勘定科目の定義	64
カスタム・メンバーの定義	70
カスタム・ディメンション情報の追加	72
エンティティ・メンバーの定義	73
シナリオ・メンバーの定義	74
アプリケーション設定の定義	78
期間別組織	81
連結メソッドの定義	81
通貨の定義	85
セル・テキスト・ラベルの定義	87
システム生成の勘定科目	89
内部取引パートナーの設定	91
セキュリティに基づくメタデータのフィルタ	93
APPフォーマットのメタデータ・ファイルの作成	94
メタデータ・マネージャのビューの使用方法	109
メタデータの参照整合性	118
メタデータのマージ・ユーティリティの使用	119
メタデータのロード	120
メタデータ・ロードの変更の表示	123
メタデータの抽出	124



注意

この章の情報は、クラシックFinancial Managementのアプリケーション専用です。Oracle Hyperion EPM Architectを使用して作成されたアプリケーションのメタデータの管理の詳細は、『Oracle Hyperion Enterprise Performance Management Architect, Fusion Edition管理者ガイド』を参照してください。

メタデータは、ディメンション名、メンバー名、プロパティ、為替レート、セキュリティなど、データを説明して格納するアプリケーションの構造要素として定義されます。クラシックFinancial Managementのアプリケーションのメタデータは、2つの方法で定義できます:

- XMLまたはAPPのメタデータ・ファイルを作成して、アプリケーションにロードします。
- メタデータ・マネージャを使用してXMLまたはAPPのメタデータ・ファイルを作成して、アプリケーションにロードします。



注:

メタデータをロードする前に、アプリケーションのセキュリティを設定する必要があります。

サンプル・メタデータ・ファイルは、Financial Managementのサンプル・アプリケーションをインストールするときに取り込まれます。これらのファイルは、Financial Managementをインストールしたディレクトリの「Sample Applications」フォルダに配置されます。

勘定科目の定義

勘定科目ディメンションは、アプリケーションの勘定科目のチャートを定義します。勘定科目は、[64ページの表 5](#)に記載されている属性を指定して定義します。

表5 勘定科目メンバーの属性

属性	説明
AccountType	(必須)次のいずれかの値です。 <ul style="list-style-type: none">• ASSET - 会社の資産の値を保管します。• LIABILITY - 会社の負債のある時点での残高を保管します。• REVENUE - 期別の値および年次累計値を保管します。値が正の場合は純利益が増えます。 注: リリース4.1以前のFinancial Managementでは、この勘定科目の種別はINCOMEと呼ばれていました。 <ul style="list-style-type: none">• EXPENSE - 期別の値および年次累計値を保管します。値が正の場合は、純利益が減ります。• FLOW - 期別値および累計値を保管します。• BALANCE - 特定の時点での符号なしの値を保管します。• BALANCERECURRING - 特定の時点での符号なしの値を保管します。将来の期間で繰り返し発生します。• CURRENCYRATE - 換算レート情報を保管します。• GROUPLABEL - 勘定科目をグループ化します。• DYNAMIC - 勘定科目の値が表示されているデータから動的に計算されることを示します。 67ページの勘定科目の種別の動作 を参照してください。
CalcAttribute	この勘定科目のルール・ファイルで実行される計算の説明です

属性	説明
	この情報は、データ・フォームおよびデータ・グリッドにセル情報の一部として表示されます。空白を含み、80文字まで入力できます。
CustomTop	<p>カスタム・ディメンションの階層のどのTopMemberを、この勘定科目で有効にするかを指定します。ここで指定したメンバーとそのすべての子孫のみが有効になります。</p> <p>注:</p> <p>属性の数は、アプリケーションに定義されたカスタム・ディメンションの数に基づいています。カスタム・ディメンション別名を反映するため、属性名が変更されます。たとえば、属性名のカスタムがカスタム・ディメンション別名で置換されます。</p>
DefaultParent	勘定科目のデフォルトの親です。
Description	<p>勘定科目の説明です。</p> <p>説明には、空白を含み、80文字まで入力できますが、アンパサンド(&)または円記号(バックスラッシュ)(¥)は使用できません。</p>
EnableCustomAggr	<p>カスタム・ディメンション・データが現在の勘定科目に集約されるかどうかを指定します。この属性は、連結ではなく特別合計に使用されます。勘定科目がカスタム・ディメンションで集約できるようにする場合はYを指定し、集約できないようにしない場合はNを指定します。</p> <p>注:</p> <p>属性の数は、アプリケーションに定義されたカスタム・ディメンションの数に基づいています。カスタム・ディメンション別名を反映するため、属性名が変更されます。たとえば、属性名のカスタムがカスタム・ディメンション別名で置換されます。</p>
EnableDataAudit	<p>勘定科目を監査できるかどうかを指定します。</p> <p>勘定科目の監査を使用可能にするにはYを指定し、使用不可にするにはNを指定します。デフォルトはNです。この属性が勘定科目またはシナリオに割り当てられている場合は、この属性によって、何を監査できるかが決まります。</p>
ICPTopMember	<p>勘定科目の内部取引PartnerTopMember</p> <p>指定したメンバーおよびその子孫のすべてが勘定科目で有効です。エンティティ・ディメンションのその他のメンバーすべてが勘定科目で無効です。</p>
IsCalculated	勘定科目を計算できるかどうかを指定します。

属性	説明
	基本レベルの勘定科目のみを計算できます。基本レベルの勘定科目を計算する場合は、手動で値を入力できません。勘定科目を計算する場合はYを指定し、計算しない場合はNを指定します。
IsConsolidated	勘定科目の値が親エンティティに連結されるかどうかを指定します。親勘定科目に連結しないと、勘定科目は連結時に無視されます。連結の実行時に勘定科目を連結する場合はYを指定し、連結しない場合はNを指定します。
IsICP	勘定科目が内部取引勘定科目かどうかを指定します。勘定科目が内部取引勘定科目の場合は、次のいずれかの値で調整勘定科目を指定する必要があります： <ul style="list-style-type: none"> • Y: 自己ICP取引を含め、ICP取引を計上できる場合 • N: ICP取引を計上できない場合 • R: ICP取引を計上できるが、勘定科目自体でICP取引が制限されている場合
Member	勘定科目の名前を指定します。この属性は必須です。勘定科目名は一意である必要があります。空白を含み、80文字まで入力できますが、1文字目は空白にできません。勘定科目名には次の文字は使用しないでください。 <ul style="list-style-type: none"> • アスタリスク(*) • アット記号(@) • カンマ(,) • 中カッコ({ }) • 二重引用符(") • 大なり記号(>) • 小なり記号(<) • マイナス記号(-) • シャープ(#) • ピリオド(.) • プラス記号(+) • セミコロン(;) • スラッシュ(/)
NumDecimalPlaces	勘定科目値に表示される小数点の右側の桁数を指定します。この属性は必須です。0-9から値を指定します。
PlugAcct	内部取引の矛盾の識別に使用される勘定科目名を指定します。勘定科目にIsICP属性が選択されている場合は、PlugAcct属性は必須です。空白であるか、有効な勘定科目の名前であることが必要です。空白の場合、勘定科目の内部取引消去は処理されません。

属性	説明
SecurityClass	勘定科目データにアクセスできるユーザーを定義するセキュリティ・クラスを指定します。セキュリティ・クラス名には、80文字まで入力できます。セキュリティ・アクセス権は、勘定科目データにのみ適用されます。
Submission Group	フェーズ送信を使用するアプリケーションの送信グループを指定します。 1から9の間の数値を入力して送信グループを指定するか、ゼロを入力して勘定科目をプロセス・コントロールから除外します。デフォルト値は1です。
UserDefined1、UserDefined2、UserDefined3	勘定科目のカスタム情報を保管します。最大256文字まで入力できます。関数 UserDefined1、UserDefined2、UserDefined3は、この属性に保管されているテキストを取得します。
UsesLineItems	勘定科目がライン・アイテムを持てるかどうかを指定します。選択されている場合、勘定科目では、ライン・アイテムが有効になっているシナリオでライン・アイテムが使用されます。勘定科目がライン・アイテムを使用する場合はYを指定し、使用しない場合はNを指定します。 注意 ライン・アイテムの詳細の入力後にこの属性を変更すると、保管されたライン・アイテムの詳細が勘定科目に有効ではなくなる場合があります。次の動作が発生します。 <ul style="list-style-type: none"> この属性をYからNに変更した場合は、データベースに保管されたライン・アイテムの詳細が有効ではなくなり、合計のみが表示されます。 この属性をNからYに変更した場合は、合計金額は表示されますが、該当するライン・アイテムの詳細情報は表示されません。合計を抽出し、合計がライン・アイテムの詳細情報と一致するようにライン・アイテムの詳細データとしてロードできます。
XBRLタグ	勘定科目のXBRLタグを指定します。入力できる最大文字数は225文字です。

勘定科目の種別の動作

各勘定科目には勘定科目の種別があります。勘定科目の種別は、子勘定科目を親勘定科目に集約する方法や、勘定科目残高を累計する方法を決定します。システムでの勘定科目の種別の動作は、68ページの表 6で説明されています。基本レベル勘定科目にデータを入力すると、結果は、階層を介して自動的にロール・アップされます。

勘定科目の種別は、子の値を親の値に加算するか、親の値から減算するかを決定します。これにより、財務の計算を勘定科目のチャートに直接作成できます。たとえば、ASSETという勘定科目の種別では、期間全体は合計されま

せん。借方がASSET勘定科目の場合、入力した値はその勘定科目に加算されます。貸方にすると、値は減算されます。この勘定科目の種別のデフォルト換算は、DefaultRateForBalance勘定科目フィールドの値です。

REVENUE勘定科目では、年次累計が合計されます。DYNAMICという勘定科目の種別は、カスタム・ディメンション、期間および期間累計のビューの親の値を正しく計算するために必要です。合計に集約する必要のない関連勘定科目をグループ化するには、GROUPLABELという勘定科目の種別を使用できます。たとえば、貸借対照表勘定科目をグループ化するBalance Sheet Accountsという名前の最上位レベルの勘定科目を作成できます。GROUPLABEL以外のすべての勘定科目の種別には、データが格納されます。

表6 勘定科目の種別の動作

種別	年次累計	借方	貸方	デフォルトの換算方法
ASSET(資産)	いいえ	加算	減算	DefaultRateForBalance 勘定科目
LIABILITY(負債)	いいえ	減算	加算	DefaultRateForBalance 勘定科目
REVENUE(収益)	はい	減算	加算	DefaultRateForFlow 勘定科目
EXPENSE(費用)	はい	加算	減算	DefaultRateForFlow 勘定科目
FLOW(フロー)	はい	加算	減算	なし
BALANCE(残高)	いいえ	加算	減算	なし
BALANCE RECURRING(経常 残高)	いいえ	加算	減算	なし
CURRENCYRATE(通貨 レート)	いいえ	N/A	N/A	N/A
GROUPLABEL(グ ループラベル)	N/A	N/A	N/A	N/A
DYNAMIC(動的)	N/A	N/A	N/A	N/A

表13は、特定のタイプの親勘定科目に合計された場合に勘定科目の種別がどのように動作するかを示しています。列は、親勘定科目の勘定科目の種別を表します。たとえば、集約時に、資産勘定科目の値は親の資産勘定科目および費用勘定科目に加算され、親の負債勘定科目および収益勘定科目から減算されます。



注:

略語は、勘定科目の種別の最初の1文字または2文字を表しています。列に「いいえ」と表示されている場合、勘定科目の種別が親勘定科目に集約されないことを示します。

表7 親勘定科目への集約時の勘定科目の種別の動作

勘定科目の種別	親勘定科目									
	A	L	R	E	F	B	BR	C	G	D
ASSET(資産)	加算	減算	減算	加算	加算	加算	加算	いいえ	いいえ	いいえ
LIABILITY(負債)	減算	加算	加算	減算	加算	加算	加算	いいえ	いいえ	いいえ
REVENUE(収益)	減算	加算	加算	減算	加算	加算	加算	いいえ	いいえ	いいえ
EXPENSE(費用)	加算	減算	減算	加算	加算	加算	加算	いいえ	いいえ	いいえ
FLOW(フロー)	加算	加算	加算	加算	加算	加算	加算	いいえ	いいえ	いいえ
BALANCE(残高)	加算	加算	加算	加算	加算	加算	加算	いいえ	いいえ	いいえ
BALANCE RECURRING(経常残高)	加算	加算	加算	加算	加算	加算	加算	いいえ	いいえ	いいえ
CURRENCYRATE(通貨レート)	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
GROUP LABEL(グループラベル)	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ
DYNAMIC(動的)	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ	いいえ

この例では、勘定科目の種別が親勘定科目にどのように集約されるかを示しています。

```

Total Assets 80
├ Fixed Assets 100
└ Amortization 20
    
```

この例では、Total Assetsは資産勘定科目で、Fixed Assets(資産勘定科目)とAmortization(負債勘定科目)の親です。これらの勘定科目が親勘定科目に集約される際に、固定資産値100が加算され、資産償却値20が減算され、総資産の結果値は80になります。

動的勘定科目の定義

動的勘定科目とは、データの要求時に動的に計算される値を持つ勘定科目です。動的勘定科目の値は保管されません。最も一般的な動的計算は、比率計算です。

▶ 動的勘定科目および計算を定義するには:

1. 動的勘定科目の種別を使用する勘定科目を設定します。

動的にできるのは、これらの勘定科目のみです。



注:

動的勘定科目では、勘定科目属性
IsCalculated、IsConsolidated、EnableCustomAggr、UsesLineItemsは無視されます。

2. ルール・ファイルに、Sub Dynamic ()セクションを作成します。
3. ルール・ファイルで計算を定義します。

計算の作成の詳細は、ルールの作成のガイドラインを参照してください。

カスタム・メンバーの定義

カスタム・ディメンションは勘定科目ディメンションと関連づけられており、勘定科目に関する追加の詳細を提供します。[70ページの表 8](#)に記載されている属性を使用して、カスタム・メンバーを定義します。

表8 カスタム・メンバーの属性

属性	説明
DefaultParent	カスタム・ディメンションのメンバーのデフォルトの親を指定します。
説明	カスタム・メンバーの説明を指定します。説明には、空白を含み、80文字まで入力できます。
IsCalculated	基本レベルのカスタム勘定科目を計算するかどうかを指定します。基本レベルのカスタム勘定科目を計算する場合は、値を手動入力できません。カスタム勘定科目を計算する場合はYを指定し、計算しない場合はNを指定します。
Member	カスタム・メンバーの名前を指定します。この属性は必須です。空白を含み、80文字まで入力できますが、1文字目は空白にできません。 注: カスタム・ディメンション・メンバーの名前を連結メソッドの名前と同じにすることはできません。 カスタム・メンバー名には、次の文字は使用しないでください: <ul style="list-style-type: none">• アスタリスク(*)• アット記号(@)• カンマ(,)• 中カッコ({})• 二重引用符(")• マイナス記号(-)• シャープ(#)

属性	説明
	<ul style="list-style-type: none"> • ピリオド(.) • プラス記号(+) • セミコロン(;) • スラッシュ(/)
SecurityClass	<p>カスタム・ディメンション・データにアクセスできるユーザーを定義するセキュリティ・クラス名を指定します。セキュリティ・クラス名には、80文字まで入力できます。セキュリティ・アクセス権は、データにのみ適用されます。</p>
Submission Group	<p>提出グループを指定します。有効な値は0-99の数値です。</p> <p>デフォルトは空白です。空白にすると、デフォルトの値1になります。</p> <p>提出グループを0に設定すると、勘定科目が確認プロセスに含まれなくなります。</p>
SwitchSignForFlow	<p>次のルールを使用するFLOW勘定科目の符号変更(借方/貸方)を指定します。</p> <ul style="list-style-type: none"> • ASSET勘定からLIABILITY勘定へ • LIABILITY勘定からASSET勘定へ • EXPENSEからREVENUEへ • REVENUEからEXPENSEへ • BALANCE勘定からFLOW勘定へ • FLOW勘定からBALANCE勘定へ <p>この勘定科目の符号が切り替えられる場合はY、切り替えられない場合はNを指定します。</p>
SwitchTypeForFlow	<p>次のルールを使用するFLOW勘定の勘定科目の種別の変更を指定します。</p> <ul style="list-style-type: none"> • ASSET勘定からEXPENSE勘定へ • EXPENSE勘定からASSET勘定へ • LIABILITYからREVENUEへ • REVENUEからLIABILITYへ • BALANCE勘定からFLOW勘定へ • FLOW勘定からBALANCE勘定へ <p>この勘定科目の種別が切り替えられる場合はY、切り替えられない場合はNを指定します。</p>
UserDefined1、UserDefined2、UserDefined3	<p>ディメンション・メンバーのカスタム情報を保管します。最大256文字まで入力できます。関数UserDefined1、UserDefined2、UserDefined3は、この属性に保管されているテキストを取得します。</p>

カスタム・ディメンション情報の追加

アプリケーション・メタデータ・ファイルを作成する際には、カスタム・ディメンション情報を追加できます。この情報は、次のいずれかの方法で追加できます：

- アプリケーション・プロファイルからカスタム・ディメンション情報をインポート
- メタデータ・マネージャでカスタム・ディメンション情報を手動で編集

アプリケーションにメタデータ・ファイルをロードすると、メタデータ・ファイルの情報が、アプリケーション・プロファイルにより前に作成されたアプリケーションのカスタム・ディメンション情報に対して検証されます。エラーがある場合には、エラー・メッセージが表示され、メタデータ・ロード・プロセスを続行できません。

カスタム・ディメンション情報のインポート

アプリケーション・メタデータ・ファイルを作成したら、プロファイルに指定したカスタム・ディメンション情報をメタデータ・マネージャが参照できるように、アプリケーション・プロファイル情報をインポートできます。プロファイルをインポートしたら、メタデータ・マネージャでカスタム・ディメンションを選択できます。メタデータ・ディメンション・リストには、カスタム・ディメンションのディメンション別名(長い説明)が表示されます。

システムにより、置換アクションを使用してプロファイル情報がインポートされます。そのため、アプリケーション・プロファイル情報をインポートすると、メタデータ・リスト内のカスタム・ディメンションに対する追加、削除または並替えが行われます。

必要な場合には、カスタム・ディメンション情報をインポートする前に、メタデータ・ファイルの既存のディメンションを変更してください。カスタム・ディメンション別名は、別の一意の名前に変更できます。たとえば、MarketsをGlobalMktに変更できます。

また、メタデータ・ファイルのカスタム・ディメンションを削除することも可能です。グリッドからカスタム・ディメンションを削除する際には、システムにより、対応するメンバー階層情報も削除されます。



注：

カスタム・ディメンション表の最初の2つのエントリは削除できません。既存のエントリを削除するには、名前を変更する必要があります。少なくとも2つのカスタム・エントリが必要です。

メタデータ・ファイルには、カスタム・ディメンションを追加できます。新しいディメンションは、メタデータ・リストの最後に追加されます。リストの途中にディメンションを挿入するには、追加してから並べ替える必要があります。

▶ カスタム・ディメンションを編集するには：

1. メタデータ・マネージャから、**カスタムの編集**をクリックします。
2. **オプション**：カスタム・ディメンションの名前の変更、削除または追加を行うには、「**名前変更**」、「**削除**」または「**追加**」をクリックします。

▶ アプリケーション・プロファイルからカスタム・ディメンション情報をインポートするには：

1. メタデータ・マネージャから、「**インポート**」をクリックします。
2. インポートするアプリケーション・プロファイルの名前を入力し、「**OK**」をクリックします。

カスタム・ディメンション情報の手動での編集

新しいメタデータ・ファイルには、アプリケーション・プロファイルからカスタム・ディメンションをインポートするかわりに、手動でカスタム・ディメンションを追加できます。また、「編集」オプションを使用して、メタデータ・ファイルに格納されたカスタム・ディメンション情報を変更できます。

これまでにカスタム・ディメンション情報を入力したことがない場合は、カスタムの編集ページを開くと、メタデータ・マネージャで唯一必要な情報であるカスタム・ディメンション別名を入力するための空白のグリッドが表示されます。これまでに、アプリケーション・プロファイルからカスタム・ディメンション情報をインポートしたことがある場合や、手動で情報を入力したことがある場合は、表に情報が表示されます。

プロファイル情報を使用してアプリケーションを作成する際に、実際の順序はアプリケーションにより決定されるため、カスタム・ディメンション別名の情報は任意の順序で入力できます。

▶ カスタム・ディメンションを手動で変更するには:

1. メタデータ・マネージャから、**カスタムの編集**をクリックします。
2. 必要に応じて、カスタム・ディメンション別名を入力します。
3. 「OK」をクリックします。

エンティティ・メンバーの定義

エンティティは、部門、子会社、工場、地域、国、法的部門、事業部門、部署、その他の組織単位など、企業の組織構造を表します。エンティティにより、データの連結パスが定義されます。エンティティ・ディメンションに対して連結を実行すると、ディメンション階層の定義に従って、子から親へデータがロール・アップされます。複数の親の下位にエンティティ子メンバーを配置することで、複数の連結パスを作成できます。[73ページの表 9](#)に記載されている属性を使用して、エンティティ・メンバーを定義します。

表9 エンティティ・メンバーの属性

属性	説明
AllowAdjFromChildren	親エンティティに対して子からの仕訳の転記を許可するかどうかを指定します。複数の親にロール・アップするエンティティの場合は、親エンティティでこの属性を有効にできます。子からの仕訳の転記を許可する場合はYを指定し、許可しない場合はNを指定します。
AllowAdjs	このエンティティに対して仕訳の転記を許可するかどうかを指定します。エンティティに対して仕訳の転記を許可する場合はYを指定し、許可しない場合はNを指定します。
DefaultParent	エンティティのデフォルトの親を指定します。
DefCurrency	エンティティのデフォルト通貨を指定します。この属性は必須です。
Description	エンティティの説明を指定します。説明には、空白を含み、80文字まで入力できます。
HoldingCompany	エンティティ・メンバーの所有者を識別する、エンティティの持株会社を指定します。エンティティの名前を指定するか、または空白のままにできます。

属性	説明
IsICP	内部取引でエンティティをパートナーとするかどうかを指定します。エンティティが内部取引エンティティの場合はYを指定し、内部取引エンティティではない場合はNを指定します。ICPを選択するメンバーは、ICPディメンションのメンバーとして自動的に表示されます。
Member	<p>エンティティの名前を指定します。この属性は必須です。エンティティ名は一意にする必要があります。空白を含み、80文字まで入力できますが、1文字目は空白にできません。エンティティ名には、次の文字は使用しないでください。</p> <ul style="list-style-type: none"> • アスタリスク(*) • アット記号(@) • カンマ(,) • 中カッコ({ }) • 二重引用符(") • マイナス記号(-) • シャープ(#) • ピリオド(.) • プラス記号(+) • セミicolon(;) • スラッシュ(/) <p>注： エンティティ名を「ALL」にすることはできません。</p>
SecurityAsPartner	この属性を使用すると、内部取引パートナーとして機能しているエンティティのセキュリティ・クラスを指定できます。ICPエンティティに対して有効なセキュリティ・クラスの名前を指定します。
SecurityClass	エンティティのデータにアクセスできるユーザーの有効なセキュリティ・クラスの名前を指定します。セキュリティ・クラス名には、80文字まで入力できます。
UserDefined1、UserDefined2、UserDefined3	エンティティのカスタム情報を保管します。最大256文字まで入力できます。関数UserDefined1、UserDefined2、UserDefined3は、この属性に保管されているテキストを取得します。

シナリオ・メンバーの定義

シナリオ・ディメンションは、予算、実績、予測などの一連の関連データを表します。75ページの表 10に記載されている属性を使用して、シナリオ・メンバーを定義します。

シナリオの期間単位は、データを入力できる期間レベルを指定します。期別データ値または年次累計データ値のいずれかを入力して表示できます。たとえば、データ・ビューとして「期別」を選択した際に、データを年次累計値として入力した場合、期別値が年次累計値から自動的に導出されます。

各シナリオに対して、欠落しているデータの表示方法を指定できます。Financial Managementでは、レポートに表示する際や要約期間を計算する際に、欠落している値はゼロとして解釈されます。欠落しているデータのゼロを現在の期間(Periodic)のゼロとして解釈するか、年次累計(YTD)のゼロとして解釈するかを指定できます。

また、シナリオごとにプロセス管理オプションを有効にできます。プロセス管理を有効にするか、プロセス単位に最大レベルの確認を選択するか、フェーズ送信の開始年を選択するかどうかを選ぶことができます。

表10 シナリオ・メンバーの属性

属性	説明
ConsolidateYTD	次のいずれかの連結のデータ・ビューを指定します - 年次累計または期別。この属性は必須です。YTDの場合はYを指定し、期別の場合はNを指定します。 注: ConsolidateYTDをNに設定した場合、ZeroViewForAdjおよびZeroViewforNonAdjオプションの両方とも期別に設定する必要があります。
DefaultFreq	シナリオに対してデータ入力がある有効な期間タイプを指定します。この属性は必須です。 たとえば、Monthlyという値は、四半期期間や年次期間ではなく、月次期間でのみ入力データを抽出できることを示します。期間単位は、アプリケーション・プロファイルで定義する必要があります。
DefaultParent	シナリオのデフォルトの親を指定します。
DefaultView	視点バーでシナリオ・ビューが選択された場合に使用するデータ・ビュー(年次累計または期別)を指定します。この属性は必須です。「YTD」または「期別」を指定します。 シナリオのデフォルト・ビューを変更し、ライン・アイテムの詳細が入力された場合は、まずライン・アイテムの詳細を抽出して保存してください。次に、シナリオからライン・アイテムの詳細を削除してから、ビューを変更します。再ロードする前に、抽出されたライン・アイテムの詳細を、新しいデフォルト・ビューと一致するように変更する必要があります。
DefFreqForICTrans	内部取引データのデフォルト期間単位を指定します。この属性は有効な期間単位である必要があり、80文字まで入力できます。この属性のデフォルトは空白です。
Description	シナリオの説明を指定します。説明には、空白を含み、80文字まで入力できます。

属性	説明
EnableDataAudit	<p>シナリオのデータに対する変更をデータ監査ログで追跡するかどうかを指定します。勘定科目またはシナリオのこの属性によって、何を監査できるかが決まります。次のいずれかの値を指定します。</p> <ul style="list-style-type: none"> • Y: すべての勘定科目を自動的に監査します。EnableDataAuditをFALSEに設定させる勘定科目も監査されます。 • O: EnableDataAuditがTRUEに設定されている勘定科目のみを監査します。 • N: すべての勘定科目を監査しません。
MaximumReviewLevel	<p>シナリオのプロセス単位に最大レベルの確認を指定します。各プロセス単位に、最大10レベルの確認を指定できます。1から10の確認レベルを指定します。この属性は必須です。</p>
Member	<p>シナリオの名前を指定します。この属性は必須です。空白を含み、80文字まで入力できますが、1文字目は空白にできません。シナリオ名には、次の文字は使用しないでください。</p> <ul style="list-style-type: none"> • アスタリスク(*) • アット記号(@) • カンマ(,) • 中カッコ({}) • 二重引用符(") • マイナス記号(-) • シャープ(#) • ピリオド(.) • プラス記号(+) • セミコロン(;)) • スラッシュ(/)
PhasedSubmissionStartYear	<p>プロセス管理のフェーズ送信が有効になっているアプリケーションで、フェーズ送信の開始年を指定します。</p>
SecurityClass	<p>シナリオのデータにアクセスできるユーザーを定義する有効なセキュリティ・クラスの名前を指定します。セキュリティ・クラス名には、80文字まで入力できます。たとえば、シナリオへのアクセス権がないユーザーは、シナリオの仕訳期間を開くことができます。</p>
SupportsProcessManagement	<p>シナリオでプロセス管理をサポートするかどうかを指定します。次のいずれかの値を指定します。</p>

属性	説明
	<ul style="list-style-type: none"> • Y: 電子メール・アラートなしのプロセス管理を使用可能にします • N: プロセス管理を使用不可にします。 • A: プロセス管理および電子メール・アラートを使用可能にします
UserDefined1、UserDefined2、UserDefined3	シナリオのカスタム情報を保管します。最大256文字まで入力できます。関数UserDefined1、UserDefined2、UserDefined3は、この属性に保管されているテキストを取得します。
UsesLineItems	<p>勘定科目でシナリオにライン・アイテムの詳細を使用可能にするかどうかを指定します。ライン・アイテムを使用できるようにする場合はYを指定し、使用できないようにする場合はNを指定します。</p> <p>注:</p> <p>ライン・アイテムの詳細の入力後にこの属性を変更すると、保管されたライン・アイテムの詳細がシナリオに有効ではなくなる場合があります。次の動作が発生します。</p> <ul style="list-style-type: none"> • この属性をYからNに変更した場合は、データベースに保管されているライン・アイテムの詳細が有効ではなくなります。合計のみが表示されます。 • この属性をNからYに変更した場合は、合計金額は表示されますが、該当するライン・アイテムの詳細情報は表示されません。合計を抽出し、合計がライン・アイテムの詳細情報と一致するようにライン・アイテムの詳細データとしてロードできます。
ZeroViewForAdj	<p>欠落している調整済の期間データ値を解釈する方法を指定します。この属性は必須です。「YTD」または「期別」を指定します。</p> <p>注:</p> <p>ConsolidateYTDをNに設定した場合、ZeroViewForAdjおよびZeroViewforNonAdjオプションの両方とも期別に設定する必要があります。</p>
ZeroViewForNonadj	<p>欠落している未調整の期間データ値を解釈する方法を指定します。この属性は必須です。「YTD」または「期別」を指定します。</p> <p>注:</p> <p>ConsolidateYTDをNに設定した場合、ZeroViewForAdjおよびZeroViewforNonAdjオプションの両方とも期別に設定する必要があります。</p>

アプリケーション設定の定義

アプリケーション設定は、Financial Managementアプリケーション全体に適用されます。アプリケーション設定によって、次のアプリケーションの情報が決まります。

- 組織が動的かどうか、期間別組織を使用するかどうか?
- どのディメンションにセキュリティを設定するか?
- どのデフォルトの換算レートとメソッドを使用するか?
- ICPの割合はどの程度か?
- 連結ルールを適用するかどうか?
- デフォルト通貨は何か?

78ページの表 11に記載されている属性を使用して、アプリケーション設定を定義します。

表11 アプリケーション設定の属性

属性	説明
ConsolidationRules	<p>連結ルールをサポートするかどうかを指定します。次のいずれかの値を指定します。</p> <p>Y: ユーザー定義ルールのConsolidate()ルーチンに記述されているルールを使用します。</p> <p>R: 値ディメンションの比例値を導出します。比例データは保管されないことに注意してください。</p> <p>N: デフォルトの連結と消去を使用します。</p>
DefaultCurrency	アプリケーションのデフォルト通貨を指定します。この属性は必須です。
DefaultRateForBalanceAccounts	ASSET勘定やLIABILITY勘定で使用する換算レートが含まれている勘定科目。この属性は必須です。
DefaultRateForFlowAccounts	REVENUE勘定またはEXPENSE勘定で使用する換算レートが含まれている勘定科目。この属性は必須です。
DefaultValueForActive	アクティブ勘定科目のデフォルト値を指定します。この属性は必須です。子を非アクティブとみなし、親に連結しない場合は0を指定します。子をアクティブとみなし、親に連結する場合は1を指定します。
EnableMetadataSecurityFiltering	<p>ユーザーにすべてのディメンション・メンバーを表示するか、アクセス権を持っているディメンション・メンバーのみを表示するかを指定します。次のディメンション・メンバーがフィルタ処理されます。</p> <ul style="list-style-type: none"> • シナリオ • エンティティ • 内部取引パートナー(ICP) • 勘定科目

属性	説明
	<ul style="list-style-type: none"> ・ カスタム <p>ユーザーがアクセス権を持っていないディメンション・メンバーを除外するにはYを指定します。この属性のデフォルトはNです。</p>
FDMApName	Oracle Hyperion Financial Data Quality Managementアプリケーションの名前
ICPEntitiesAggregationWeight	値ディメンションの[ICP Top]メンバーに集約する内部取引パートナ・エンティティ[ICP Entities]の金額の割合を指定します。この属性は必須です。百分率の値を、小数(1.0が100パーセント)で表します。
MaxCellTextSize	セル・テキストに使用できる最大文字数を指定します。無制限または1900以上2,147,483,646までの場合は-1を指定します。0から1899までの値は無効です。デフォルト値は8,000です。
MaxNumDocAttachments	ユーザーごとのドキュメントの添付ファイルの最大数を指定します。無制限または2,147,483,647までの正数の場合は-1を指定します。デフォルト値は-1です。
MaxDocAttachmentSize	ドキュメントの添付ファイルのサイズの最大バイト数を指定します。無制限または2,147,483,646までの正数の場合は-1を指定します。デフォルト値は-1です。
NodeSecurity	ノードのセキュリティ・アクセスのタイプを指定します。この属性は必須です。エンティティのセキュリティ・アクセスに基づいてノード・データをチェックするには「エンティティ」を選択し、親のセキュリティ・アクセスに基づいてノード・データをチェックするには「親」を選択します。
OrgByPeriodApplication	アプリケーションで、新規連結構造を以前の連結構造と共存できるようにするかどうかを指定します。新規組織構造を許可するにはYを指定し、現在の組織構造のみを許可するにはNを指定します。
SupportSubmissionPhaseforAccounts	プロセス管理のフェーズ提出を、アプリケーションの勘定科目でサポートするかどうかを指定します。 有効な値はYまたはNです。デフォルトはNです。
SupportSubmissionPhaseforCustom	プロセス管理のフェーズ提出を、アプリケーションのカスタム・メンバーでサポートするかどうかを指定します。有効な値はYまたはNです。デフォルトはNです。 注: 属性の数は、アプリケーションに定義されたカスタム・ディメンションの数に基づいています。カスタム・ディメンション別名を反映するため、属性名が変更されます。たとえば、属性

属性	説明
	名のカスタムがカスタム・ディメンション別名で置換され ます。
SupportSubmissionPhaseforICP	プロセス管理のフェーズ提出を、アプリケーションのICPメン バーでサポートするかどうかを指定します。 有効な値はYまたはNです。デフォルトはNです。
UsePVAForBalanceAccounts	BALANCE勘定のデフォルト換算メソッドを指定します。 期別値(PVA)換算メソッドを使用する場合はYを指定し、 為替レート(VAL)換算メソッドの値を使用する場合はNを 指定します。
UsePVAForFlowAccounts	FLOW勘定のデフォルト換算メソッドを指定します。期別 値(PVA)換算メソッドを使用する場合はYを指定し、為替 レート(VAL)換算メソッドの値を使用する場合はNを指定し ます。
UseSecurityForAccounts	アプリケーションの勘定科目をセキュリティで保護するかど うかを指定します。セキュリティで勘定科目を保護する場 合はYを指定し、セキュリティで保護しない場合はNを指定 します。
UseSecurityForCustom	アプリケーションのカスタム・ディメンションをセキュリティ で保護するかどうかを指定します。セキュリティでカスタム・ ディメンションを保護する場合はYを指定し、セキュリティで 保護しない場合はNを指定します。 注： 属性の数は、アプリケーションに定義されたカスタム・ディメ ンションの数に基づいています。カスタム・ディメンション別 名を反映するため、属性名が変更されます。たとえば、属性 名のカスタムがカスタム・ディメンション別名で置換されま す。
UseSecurityForEntities	アプリケーションのエンティティをセキュリティで保護するか どうかを指定します。セキュリティでエンティティを保護する 場合はYを指定し、セキュリティで保護しない場合はNを指 定します。
UseSecurityForICP	アプリケーションのICPメンバーをセキュリティで保護する かどうかを指定します。セキュリティでICPメンバーを保護 する場合はYを指定し、セキュリティで保護しない場合はN を指定します。
UseSecurityForScenarios	シナリオをセキュリティで保護するかどうかを指定します。 セキュリティでシナリオを保護する場合はYを指定し、セ キュリティで保護しない場合はNを指定します。
UseSubmissionPhase	プロセス管理のフェーズ提出をアプリケーションで使用する かどうかを指定します。

属性	説明
	有効な値はYまたはNです。デフォルトはNです。
ValidationAccount	<p>検証に使用する勘定科目名を指定します。検証に使用する勘定科目は既存の勘定科目であることが必要です。</p> <p>プロセス管理では、検証勘定科目は、プロセス単位を次の確認レベルに上げる前に、値が必ずゼロに等しくなるようにするために使用されます。検証勘定科目1は送信フェーズ1で使用され、検証勘定科目2から9は、送信フェーズ2から9で使用されます。</p>

期間別組織

組織構造は、買収、売却、合併および更生などの多くの理由で変化する可能性があります。期間別組織機能を使用すると、同じアプリケーション内で最新の連結構造と過去の構造を共存させることができます。

組織の変化に対応するために、Financial Managementでは、エンティティ階層のフィルタとしてアクティブなシステム勘定科目が使用されます。アクティブ勘定科目は、親レベルでデータを保管し、ICPディメンションを使用して子に関する情報を保管する内部取引勘定科目です。親への子エンティティの連結ステータスがアクティブであるか非アクティブであるかを指定します。

親の子に相当するICPメンバーの場合は、アクティブ勘定科目は、子を現在の年、シナリオ、期間のアクティブな連結メンバーとみなすかどうかをシステムに示します。アクティブ勘定科目が0であるICPメンバーに相当する子は、非アクティブな子とみなされ、連結されません。アクティブ勘定科目が1であるICPメンバーに相当する子は、アクティブな子とみなされ、連結されます。アクティブな子データに対する変更は親に影響を与えます; 非アクティブな子データへの変更は親に影響を与えません。アクティブな勘定科目の値はデータ・グリッドで表示または変更できます。

DefaultValueForActive属性は、アクティブ勘定科目が空白である子のステータスを制御します。そのため、すべての親と子の交差をアクティブまたは非アクティブと指定する必要はありません。デフォルトでは、特に指定がないかぎり、すべての子は、その親との関係においてアクティブです。

連結メソッドの定義

81ページの表 12に記載されている属性を使用して、アプリケーションの連結メソッドを定義します。

表12 連結メソッドの属性

属性	説明
ConsolMethod	<p>連結メソッドの名前を指定します。この属性は必須です。通貨名は一意にする必要があります。空白を含み、80文字まで入力できます。名前には、次の文字は使用しないでください:</p> <ul style="list-style-type: none"> • アスタリスク(*) • アット記号(@) • カンマ(,) • 中カッコ({ })

属性	説明
	<ul style="list-style-type: none"> • 二重引用符(") • マイナス記号(-) • シャープ(#) • ピリオド(.) • プラス記号(+) • セミコロン(;) • スラッシュ(/)
Control	<p>計算ルーチンによって使用されるコントロール・タイプに対応するしきい値を指定します。この属性に次のいずれかの値を指定します:</p> <ul style="list-style-type: none"> • Blank • No • Limited • Full
Description	<p>連結メソッドの説明を指定します。説明には、空白を含み、80文字まで入力できます。</p>
IsHoldingMethod	<p>持株会社に連結メソッドを使用するかどうかを指定します。この属性はオプションです。持株会社にこのメソッドを使用するにはYを指定し、別のメソッドを使用する場合はNを指定します。</p>
PercentConsol	<p>出資比率計算プロセスによって適用される連結比率を指定します。パーセントの値(たとえば100)または、次のいずれかのキーワードを指定します。</p> <ul style="list-style-type: none"> • POWN • POWNMIN <p>注:</p> <p>POWNおよびPOWNMINの詳細は、84ページのPOWNまたはPOWNMIN連結メソッドの使用を参照してください。</p>
ToPercentControl	<p>PercentControlの範囲の上限を指定します。出資比率計算ルーチンに使用されます。0-100の値を指定します。</p> <p>注:</p> <p>メソッドのレコードの1つで、100に指定する必要があります。</p>
ToPercentControlComp	<p>コントロール比率の範囲の上限を、範囲に含めるかどうかを指定します。ToPercentControl属性と組み合わせて出資比率計算ルーチンに使用されます。この属性</p>

属性	説明
	は、UsedByCalcRoutine属性がNの場合はオプションです。この属性には、<または<=を指定します。
UsedByCalcRoutine	このメソッドを自動出資比率計算に使用するかどうかを指定します。出資比率計算にこのメソッドを使用する場合はYを指定し、使用しない場合はNを指定します。

連結メソッドの使用方法

連結メソッドは、連結および出資比率の計算プロセスで使用されます。

メタデータで連結メソッドを定義すると、通貨(元)ディメンションの[ConsolMethod]システム・リストが自動的に生成され、このリストに連結メソッド・セクションに定義されているすべてのメソッドが含まれます。

連結メソッドをエンティティに割り当てて、連結時に使用できるようにするには、2つの方法があります：

- データ・ロードまたはデータ入力によってメソッドを手動で割り当てられます。
- エンティティに割り当てられた最終的な支配比率に基づく出資比率の計算プロセスで、メソッドを自動的に割り当てられます。出資管理計算の詳細は、『Oracle Hyperion Financial Managementユーザー・ガイド』を参照してください。

手動での連結メソッドの割当て

連結メソッド情報を手動で入力するために、次の情報が含まれているデータ・グリッドを作成できます。

POV: Scenario, Year, Period, View, Entity, Value, Account, Custom

Scenario: Applicable scenario

Year: Applicable year

Period: Applicable period

Entity: A parent entity

Value: [None]

Account: [Method] system-generated account

Custom: ToCurrency

Row: ICP entities. For parent entities, you use the ICP Entities system list, or a user-defined list of selected ICP entities.

Column: FromCurrency. You should use the ConsolMethods system-generated list.

メソッド割当て情報は、親エンティティのデータ・ファイルの勘定科目メソッドに保管されます。親のそれぞれの子の場合は、連結メソッド割当てはICPディメンションに保管されます。割り当てられたメソッドは、子を親に連結する際に使用されます。

グリッドの交差の場合は、1を使用して、ICPエンティティへのメソッド割当てを指定します。たとえば、親グループに2つの子AとBがあり、AにGlobalメソッドを割り当て、BにEquityメソッドを割り当てる場合は、GlobalメソッドとエンティティAの交差に1を入力し、EquityメソッドとエンティティBの交差に1を入力します。

POWNまたはPOWNMIN連結メソッドの使用

出資比率計算プロセスでは、連結メソッド表の設定を使用して支配比率と出資比率の最終パーセンテージが計算され、連結比率と連結メソッドが自動的に割り当てられます。

EQUITYプロセスに対応する連結メソッドには、連結メソッド表のPOWNMINキーワードを使用します。POWNMINを使用すると、EQUITY企業に割り当てられる連結比率は、ステージングで実行される連結プロセスに使用される比率に対応します。

POWNMIN計算

$$\text{POWNMIN} = \text{POWN} + \text{Sum of (Percent Minority of Entity Owners * Direct Percentage of Ownership in the Entity)}$$

ここで:

- 少数比率 = 連結比率 - 出資比率
- エンティティ所有者とは、処理されるエンティティの株式を所有する現在の親の子孫のエンティティです
- エンティティの直接出資比率は、Shares%Ownedシステム勘定科目から取得されます

例:

- BのAによる出資: 80%
- CのAによる出資: 70%
- DのBによる出資: 20%

- DのCによる出資: 20%

親エンティティGROUPはエンティティA、B、C、Dを子孫として持ちます(Aは持株会社です)。出資比率は次のように計算されます:

- A: 100%
- B: 80%
- C: 70%
- D: 30%

Dの(連結ロジック表による)出資比率はPOWNMINであるとします。

連結プロセスがステージングで実行された場合、POWNMINプロセスは次のようになります:

1. 直接出資率を使用したDのBへの連結: 20%
2. 直接出資率を使用したDのCへの連結: 20%
3. それぞれの比率を使用したBおよびCのAへの連結: (80%および70%)

次のように計算されます:

$$\begin{aligned} \text{Entity D's Percent consolidation} &= 30\% + (100\% - 80\%) * 20\% + (100\% - 70\%) * 20\% \\ &= 40\% \end{aligned}$$

このステージングされた連結プロセスを使用し、エンティティDは合計比率40%で連結されます。

下位持株会社BとCがAに連結されると、10%の少数株主持ち分がエンティティDの資本で計算されます。

POWN計算

ただし、連結がフラット階層で行われる場合は通常、その処理では最終出資比率(POWN)を資本企業の連結比率として使用します。この場合、Dのグループへの連結比率は30%になります。エンティティDからの資本では、少数株主持ち分は計算されません。

要約:

- POWNを使用すると、エンティティDに割り当てられる連結比率は30% (最終出資比率)になります。
- POWNMINを使用すると、エンティティDに割り当てられる連結比率は40% (ステージングされた連結プロセスを使用)になります。

通貨の定義

通貨は、エンティティの換算値を格納します。すべてのアプリケーションに通貨ディメンションが必要です。通貨ディメンションには、エンティティ・ディメンションでエンティティに割り当てられているデフォルトの各通貨に通貨が含まれている必要があります。通貨ディメンションに追加された各通貨は、システム生成メンバーとして値ディメンションに表示されます。値ディメンションから通貨を選択し、通貨に換算されたデータ値を表示できます。

通貨ディメンション種別を使用してディメンションを作成し、86ページの表 13に記載されている属性を使用して、アプリケーションの通貨を定義します。アプリケーションに必要な各通貨の通貨ディメンションにメンバーを作成します。

表13 通貨の属性

属性	説明
Currency	<p>通貨の名前を指定します。この属性は必須です。通貨名は一意にする必要があります。空白を含み、80文字まで入力できます。通貨名には、次の文字は使用しないでください。</p> <ul style="list-style-type: none"> • アスタリスク(*) • アット記号(@) • カンマ(,) • 中カッコ({ }) • 二重引用符(") • マイナス記号(-) • シャープ(#) • ピリオド(.) • プラス記号(+) • セミコロン(;) • スラッシュ(/)
Description	<p>通貨の説明を指定します。説明には、空白を含み、80文字まで入力できます。</p>
DisplayInICT	<p>内部取引モジュールのドロップダウン・リストに通貨を表示するかどうかを指定します。通貨を表示する場合はYを指定し、表示しない場合はNを指定します。デフォルトはYです。</p>
Scale	<p>小数点の位置を指定して、通貨の金額を表示および保管するときの単位を指定します。この属性は必須です。また、為替レートの入力方法も決定します。たとえば、データが千の単位の調整される場合、データ・フォームに入力された1という値は、データベースでは1,000として保存されます。Scaleはエンティティの属性ではなく、通貨の属性です。この属性に次のいずれかの値を指定します：</p> <ul style="list-style-type: none"> • 空白 = なし • 0 = 1単位 • 1 = 十 • 2 = 百 • 3 = 千 • 4 = 万 • 5 = 十万

属性	説明
	<ul style="list-style-type: none"> • 6 = 百万 • 7 = 千万 • 8 = 億 • 9 = 十億
TranslationOperator	<p>内部取引を使用している場合は、ローカル通貨を為替レートで乗算または除算するかどうかを指定します。デフォルトは空白です。次のいずれかの値を指定します。</p> <ul style="list-style-type: none"> • D: 取引通貨をレートで除算して現地通貨を計算します。 • M: 取引通貨をレートで乗算して現地通貨を計算します。 • 空白: デフォルトでDが使用されます。

[Currencies]システム・メンバー・リストは、通貨(元)および通貨(先)ディメンションに使用できます。アプリケーションに追加した通貨は、[Currencies]メンバー・リストに追加されます。[Currencies]リストによって、通貨の組合せに対する通貨換算レートの入力と、通貨以外のメンバーの除外が可能になります。

セル・テキスト・ラベルの定義

データ・グリッドまたはフォームの有効なセルにセル・テキストを追加できます。視点の交差のさまざまなタイプの情報を格納するため、複数のセル・テキスト・エントリが必要な場合があります。複数のセル・テキスト・エントリを作成し、セル・テキスト・ラベルを定義して、セル・テキストを表示する際に情報のタイプを簡単に識別できます。セル・テキスト・ラベルは、データ・グリッドおよびデータ・フォームで使用できます。

ユーザーが複数のセル・テキスト・エントリを入力する前に、管理者はセル・テキスト・ラベルを定義する必要があります。ラベルはメタデータとしてロードされます。アプリケーションのすべての勘定科目のセルに適用され、ユーザーがセル・テキスト情報を入力する際に選択できます。『Oracle Hyperion Financial Managementユーザー・ガイド』を参照してください。

セル・テキスト・ラベルを定義するガイドラインは、次のとおりです:

- ラベル名は最大80文字です。データ・セル内で一意である必要があります。
- 空白を含めることができますが、空白で開始することはできません。Oracleデータベースを使用している場合、ラベルに空白を含めることはできません。
- ラベル名に次の文字を含めることはできません:
 - アンパサンド(&)
 - アスタリスク(*)
 - アット記号(@)
 - カンマ(,)
 - 中カッコ({})
 - 二重引用符(")
 - スラッシュ(/)
 - 小なり記号(<)
 - マイナス記号(-)

- シャープ(#)
- ピリオド(.)
- パイプ文字(|)
- プラス記号(+)
- セミコロン(;)
- チルダ文字(~)

ロード・ファイルの「セル・テキスト・ラベル」セクションのメタデータ・ロードの一部として、セル・テキスト・ラベルをロードできます。次の例は、セル・テキスト・ラベルのサンプルのロード・ファイル・セクションを示しています：

```
<MISC Name="CellTextLabel">
```

```
<MISCENTRY>
```

```
<LABEL>MaturityDate</LABEL>
```

```
</MISCENTRY>
```

```
<MISCENTRY>
```

```
<LABEL>CouponRate</LABEL>
```

```
</MISCENTRY>
```

```
<MISCENTRY>
```

```
<LABEL>ExchangeRate</LABEL>
```

```
</MISCENTRY>
```

```
<MISCENTRY>
```

```
<LABEL>Rating</LABEL>
```

```
</MISCENTRY>
```


<MISCENTRY>

<LABEL>InterestRate</LABEL>

</MISCENTRY>

</MISC>

システム生成の勘定科目

アプリケーションを作成すると、そのアプリケーションの連結と出資比率のシステム勘定科目が自動的に作成されます。



注:

変更できるのは、システム勘定科目の説明、セキュリティ・クラス、および小数点位置のみです。システム勘定科目のその他の属性は、すべて事前に定義されていて、変更はできません。

連結勘定科目

89ページの表 14 に記載されているシステム勘定科目は、エンティティ・ディメンションの個々の親に必要で、連結プロセスで使用されます。



注:

連結に使用されるシステム勘定科目は、アクティブ勘定科目を除き、すべて残高勘定です。アクティブ勘定科目は、経常残高勘定です。

表14 連結に使用されるシステム勘定科目

勘定科目	説明
アクティブ	子から親への連結のステータス。子を親に連結する場合は「はい」、親に連結しない場合は「いいえ」です。
[PCON]	連結比率。エンティティの親に連結するエンティティの値の比率。-100から100までの正数または負数で、0も含まれます。デフォルト値は100です。 注: 後続の期間では0と導出されます。そのため、すべての後続の期間に比率を入力する必要があります。

勘定科目	説明
[POWN]	他のエンティティによって所有されているエンティティの株式数に基いて計算した出資比率。0-100の正数。デフォルト値は100です。
[DOWN]	直接出資比率。0-100の正数。デフォルト値は100です。
[PCTRL]	他のエンティティに所有されている議決権株式に基づいて計算された支配比率。0-100の正数。デフォルト値は100です。
Method	エンティティに割り当てられる連結メソッド。「なし」を指定、または使用可能なメソッドのリストから選択。
Consol1、Consol2、Consol3	連結メソッド。0-255の数値。

所有権勘定科目

90ページの表 15に記載されているシステム勘定科目は、出資比率計算に使用されます。



注:

出資比率計算に使用されるシステム勘定科目は、すべてBALANCE勘定です。

表15 出資比率のシステム勘定科目

勘定科目	説明
SharesOwned	所有株式の総数。正数または0。デフォルトは0。 注: 所有株式の総数は、未処理株式の総数以下であることが必要です。
VotingOwned	所有議決権株式の数。正数または0。デフォルト値は0です。 注: 所有議決権株式の総数は、未処理議決権株式の総数以下であることが必要です。
SharesOutstanding	未処理株式の総数または割合。正数または0。デフォルト値は0です。 注: 未処理株式の数または割合を入力します。割合はパーセントで入力します。
VotingOutstanding	未処理議決権株式の数。正数または0。デフォルト値は0です。

勘定科目	説明
	注: 未処理議決権株式の数または割合を入力します。割合はパーセントで入力します。
Shares%Owned	システムによって計算されます
Voting%Owned	システムによって計算されます

システムで生成された勘定科目の編集

アプリケーションを作成するときに、システム勘定科目のメンバーが自動的に作成されます。



注:

変更できるのは、システム勘定科目メンバーの説明、セキュリティ・クラス、および小数点位置のみです。その他の属性は、すべて事前に定義されていて、変更はできません。

▶ システム勘定科目を編集するには:

1. メタデータ・マネージャで、システムで生成された勘定科目のメンバーが含まれているファイルを開きます。
2. 「リスト・ビュー」タブを選択します。
3. リストからシステム勘定科目を選択し、説明、セキュリティ・クラス、または小数点の位置を必要に応じて変更します。
4. その他のシステム勘定科目メンバーを編集するには、必要に応じて手順3を繰り返します。
5. 「ファイルの保存」をクリックします。



注:

変更を有効にするには、更新したメタデータ・ファイルをアプリケーションにロードする必要があります。

内部取引パートナーの設定

内部取引は、内部取引パートナー(ICP)ディメンション全体で管理されます。ICPディメンションは、1つの勘定科目に対して存在するすべての内部取引残高を含んでいます。ICPは、内部取引の詳細を追跡および消去するために、勘定科目ディメンションおよびカスタム・ディメンションとともに使用する予約済のディメンションです。

内部取引用にアプリケーションを設定するには、次のアクションを行う必要があります。

- 内部取引が計上される勘定科目と、各内部取引勘定の調整勘定科目を指定します(それぞれ、勘定科目のメタデータのIsICP属性とPlugAcct属性を指定します)。
- 内部取引を実行するエンティティを指定します(エンティティ・メタデータのIsICP属性)。

内部取引を作成する場合は、各グループが内部取引勘定科目と調整勘定科目を少なくとも1つずつ持っている必要があります。メタデータ・マネージャで勘定科目のIsICP属性を選択し、勘定科目を内部取引として指定します。勘定科目を内部取引として指定して、内部取引を入力すると、連結プロセスによって、消去仕訳または逆仕訳が[Elimination]値ディメンション・メンバーに作成されます。

調整勘定科目は、消去の完了時に[Elimination]値ディメンションの2つの内部取引勘定科目間の差異を保管する勘定科目です。調整勘定科目は、ICP勘定科目として設定できます。調整勘定科目をICP勘定科目として設定するには、IsICPメタデータ属性をYまたはRに設定し、消去が該当のICPメンバーに書き込まれるようにします。調整勘定科目をICP勘定科目として設定しない場合は、IsICP属性をNに設定します。その結果、消去は[ICP None]に書き込まれます。

連結時に、有効な内部取引エンティティ間の取引が消去されます。[73ページのエンティティ・メンバーの定義](#)を参照してください。

次の表に、システムで生成されるICPの要素を示します。

表16 システム生成のICP要素

ICPの要素	説明
[ICP Top]	最上位の内部取引メンバーを指定します
[ICP None]	内部取引メンバーが使用されないことを指定します。
[ICP Entities]	内部取引が発生するエンティティを示します。

システムで生成されたICPメンバーの編集

アプリケーションを作成すると、そのアプリケーションに対して内部取引パートナー(ICP)メンバーが自動的に作成されます。ICPメンバーは、IsICP属性が選択されているエンティティ・メンバーごとに作成されます。



注:

変更できるのは、ICPメンバーの説明とセキュリティ・クラスのみです。その他の属性は、すべて事前に定義されていて、変更はできません。

▶ 内部取引メンバーを変更するには:

1. メタデータ・マネージャで、システムで生成された内部取引パートナーのメンバーが含まれているファイルを開きます。
2. リストからICPメンバーを選択し、必要に応じて説明とセキュリティ・クラスを変更します。
3. その他のICPメンバーを編集するには、必要に応じて手順2を繰り返します。
4. 「ファイルの保存」をクリックします。



注:

変更を有効にするには、更新したメタデータ・ファイルをアプリケーションにロードする必要があります。

システム生成の値メンバーの編集

アプリケーションを作成すると、そのアプリケーションの値メンバーが自動的に作成されます。



注:

変更できるのは、値メンバーの説明のみです。その他の属性は、すべて事前に定義されていて、変更はできません。

メタデータをロードしたら、アプリケーションの通貨ごとに3つの値ディメンション・メンバーが自動的に作成されます:

- *CurrencyName*
- *CurrencyName Adjs*
- *CurrencyName Total*

CurrencyName は通貨のラベルです。

たとえば、通貨USDの場合は、次の値ディメンション・メンバーが作成されます: USD、USD Adjs、USD Total。



注:

メタデータ・ファイルには、<Currency>値メンバーに対して指定した説明が含まれている必要があります。メタデータ・ファイルで通貨の説明が指定されていない場合に、メタデータをロードすると、通貨の説明は表示されません。

▶ 値メンバーの適用を変更するには:

1. メタデータ・マネージャで、システム生成の値メンバーが含まれているファイルを開きます。
2. 「メンバー属性」タブで、階層内のメンバーを選択し、説明属性を変更します。
3. 必要に応じて、手順2を繰り返し、その他の値メンバーの説明を追加します。
4. 「ファイルの保存」をクリックします。



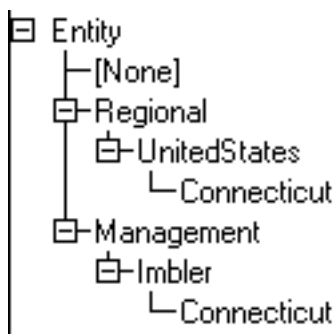
注:

変更を有効にするには、更新したメタデータ・ファイルをアプリケーションにロードする必要があります。

セキュリティに基づくメタデータのフィルタ

セキュリティに基づいてメタデータをフィルタすると、ユーザーがアクセス権を持っているシナリオ、エンティティ、ICP、勘定科目およびカスタム・ディメンションのみがユーザーに表示されます。AppSettings EnableMetadataSecurityFilteringメタデータ属性をYに設定し、アプリケーション・レベルでメタデータのフィルタ処理を設定します。ユーザーが階層表示できる要素については、セキュリティ・クラスを割り当て、ユーザーにセキュリティ・クラスに対するメタデータ・アクセス権を割り当てます。

ユーザーは、アクセス権を持っているメンバーの親および祖先への暗黙のアクセス権を持っています。暗黙のアクセス権によって、ユーザーには祖先および親が階層ツリー構造で表示されますが、アクセスはできません。たとえば、次のツリー構造では、親(UnitedStatesとImbler)と祖先(ManagementとRegional)がツリーに表示されますが、ユーザーがアクセスできるのはConnecticutのみです。



APPフォーマットのメタデータ・ファイルの作成

APPフォーマットのメタデータ・ファイルを使用し、アプリケーションにメタデータを追加できます。メタデータ・ファイル・セクションは任意の順序で配置できます; ただし、セクションは次の順序で自動的に処理されます:

- 通貨
- エンティティ・ディメンション
- シナリオ・ディメンション
- カスタム・ディメンション
- 勘定科目ディメンション
- 値ディメンション
- 内部取引パートナー・ディメンション
- アプリケーション設定
- 連結メソッド

各種のディメンション関連メタデータの場合は、セクションは次の順序で処理されます。

- メンバー
- 階層



注:

カスタム・ディメンションの場合、これらのセクションは、カスタム・ディメンションの別名を使用する必要があります。

エンティティ・ディメンション、シナリオ・ディメンション、勘定科目ディメンションおよびカスタム・ディメンションのメタデータは、メンバー・セクションおよび階層セクションに配置されます。カスタム・ディメンションにディメンション・セクションが含まれる場合があります。各種のメタデータ・セクションは、1つのメタデータ・ファイルで一度のみ使用できません。

区切り文字には、次の文字を使用できます。

```
, ~ @ $ % ^ | : ; ? \
```



注:

ファイル名や、ファイル内のその他の要素で使用されていない文字を使用してください。区切り文字は、ファイル拡張子がAPPのASCIIファイルにのみ必要です。拡張マークアップ言語(XML)ファイルには必要ありません。

感嘆符(!)で始まる行は、メタデータ・ファイルの新規セクションの開始を示しており、その後、通貨、メンバー、階層などの有効なセクション名を続ける必要があります。TRUEとFALSEは、TRUEの場合はY、FALSEの場合はNです。アポストロフィ(')で始まる行は、コメント行とみなされ無視されます。

次のセクションはメタデータ・ファイルで使用できます。

- ファイル・フォーマット
- バージョン
- アプリケーション設定
- 通貨
- デイメンション
- メンバー
- 階層
- 連結メソッド

ファイル・フォーマット

メタデータ・ファイルのこのセクションは、ファイルのバージョン番号を示します。バージョン番号は、ファイル・フォーマットが変更された場合のみ変更できます。ファイル・フォーマットはメタデータの抽出時に自動的に生成されます。したがって、ロードするファイルを定義している場合は、有効なファイル・フォーマットを含める必要があります。次の構文は、ファイル・フォーマットを指定します。

```
!FILE_FORMAT =  
majorNumber  
.  
minorNumber
```

majorNumberとminorNumberは、1つまたは2つの桁から構成されます。majorNumberには先行ゼロを含めることができ、minorNumberには後続ゼロを含めることができます。majorNumberとminorNumberを含める必要があり、小数点にはピリオド(.)以外は使用しないでください。次の例は、有効なファイル・フォーマットを示しています:

```
!FILE_FORMAT = 11.12
```

```
!FILE_FORMAT = 11.120
```

```
!FILE_FORMAT = 011.120
```

```
!FILE_FORMAT = 011.12
```

バージョン

メタデータ・ファイルのこのセクションは、メタデータの抽出に使用されたFinancial Managementのバージョンを示します。バージョン番号は、メタデータの抽出時に自動的に生成されます。したがって、ロードするメタデータ・ファイルを作成している場合は、バージョンの指定は必要ありません。次の構文は、バージョンを示しています。

```
!VERSION =  
major version  
.  
minor version  
.  
build version
```

この例では、有効なバージョン値を示しています。

```
!VERSION = 11.1.4749
```

アプリケーション設定

メタデータ・ファイルのこのセクションは、Financial Managementアプリケーション全体に適用される設定を定義します。アプリケーション設定の属性については、[78ページのアプリケーション設定の定義](#)を参照してください。

この例では、アプリケーション設定の属性を指定します。

```
!APPLICATION_SETTINGS
```

```
DefaultCurrency=USD
```

```
DefaultRateForBalanceAccounts=Rate1
```


DefaultRateForFlowAccounts=Rate2

UsePVAForBalanceAccounts=Y

UsePVAForFlowAccounts=Y

ICPEntitiesAggregationWeight=1

DefaultValueForActive=1

ConsolidationRules=N

OrgByPeriodApplication=N

NodeSecurity=Entity

UseSecurityForAccounts=N

UseSecurityForEntities=Y

UseSecurityForScenarios=Y

UseSecurityForFlows=Y

UseSecurityForMarket=Y

UseSecurityForRegion=N

UseSecurityForCostCenter=N

UseSecurityForICP=N

EnableMetadataSecurityFiltering=N

```
SupportSubmissionPHaseForFlows=Y
```

```
SupportSubmissionPhaseForMarket=Y
```

```
SupportSubmissionPHaseForRegion=N
```

```
SupportSubmissionPhaseForCostCenter=N
```

通貨

メタデータ・ファイルのこのセクションは、通貨を定義します。次の構文は、通貨を指定します。

```
Label; Scale; Descriptions
```

[85ページの通貨の定義](#)を参照してください。

この例では、通貨の属性を指定します。

```
!CURRENCIES
```

```
EURO;0;English=European Euro
```

```
GBR;0;English=Great Britain Pounds
```

```
USD;0;English=United Stated Dollars
```

メンバー

メタデータ・ファイルのこのセクションは、ディメンションのメンバーを定義します。区切り文字を使用すると、欠落した値を空白として表せます。次の構文を使用して、ディメンション・メンバーを入力します。

```
!MEMBERS=Customers
```



```
!Custom_Order=Product;Customers;Channel;UnitsFlows
```

次に、AdminExpensesとCapitalStockという2つの勘定科目の属性を指定した例を示します。

```
!MEMBERS=Account
```

```
AdminExpenses;EXPENSE;N;Y;Y; ;AllCustom3;AllCustom1;AllMarket;AllFlows;2;N;Y;Y;Y;Y; ;  
; ; ; ; ;N;DefaultParent=NetIncome
```

```
CapitalStock;LIABILITY;N;Y;N; AllCustom3;  
[None];AllMarket;AllFlows;6;N;Y;Y;Y;Y;  
; ; ; ; ;N;DefaultParent=TotalEquity;English=Capital Stock
```

シナリオ

シナリオ・ディメンション・メンバーの構文:

```
'Label, DefaultFreq, DefaultView, ZeroViewForNonadj, ZeroViewForAdj,  
ConsolidateYTD, UserDefined1, UserDefined2, UserDefined3, SupportsProcessManagement,  
SecurityClass, MaximumReviewLevel, UsesLineItems, EnableDataAudit,  
EnableJournalsAutoLabel, DefFreqForPostingFlowTrans, DefaultParent, Descriptions
```

[74ページのシナリオ・メンバーの定義](#)を参照してください。

次に、ActualとBudgetという2つのシナリオの属性を指定した例を示します。

```
!MEMBERS=Scenario
```

```
Actual;MTD;Periodic;Periodic;Periodic;N; ; ; ;N; ;10;Y;N;N;MTD ;DefaultParent=#root
```

```
Budget;MTD;Periodic;Periodic;Periodic;N; ; ; ;Y; ;10;Y;N;N;MTD ;DefaultParent=#root
```

エンティティ

エンティティ・ディメンション・メンバーの構文:

```
'Label, DefCurrency, AllowAdjs, IsICP, AllowAdjFromChildren, SecurityClass,
UserDefined1, UserDefined2, UserDefined3, HoldingCompany, SecurityAsPartner,
DefaultParent, Descriptions
```

[73ページのエンティティ・メンバーの定義](#)を参照してください。

次に、California、CanadaおよびConnecticutという3つのエンティティの属性を指定した例を示します。

```
!MEMBERS=Entity
```

```
California;USD;Y;Y;Y;US;;;;;DefaultParent=Imbler;English=State of
California;French=California
```

```
Canada;USD;Y;N;N;N;;;;;DefaultParent=Regional
```

```
Connecticut;USD;Y;Y;N;US;Northeast;;;;;DefaultParent=Imbler
```

カスタム

カスタム・ディメンション・メンバーの構文:

```
'Label, IsCalculated, SwitchSignForFlow, SwitchTypeForFlow, UserDefined1,
UserDefined2, UserDefined3, SecurityClass, DefaultParent, Descriptions
```

[70ページのカスタム・メンバーの定義](#)を参照してください。

次に、カスタム3ディメンションのメンバーの属性を指定した例を示します。

```
!MEMBERS=Customers
```

```
[None];N;N;N;N;;;;;DefaultParent=#root
```

```
AllCustomers;Y;N;N;N;;;;;DefaultParent=#root
```

```
Customer2;N;N;N;N;;;;;DefaultParent=AllCustomers
```

```
Customer3;N;N;N;;;;DefaultParent=AllCustomers
```

```
Customer4;N;N;N;;;;DefaultParent=AllCustomers
```

```
Customer5;N;N;N;;;;DefaultParent=AllCustomers
```

値

メンバー・セクションを使用して、値ディメンションのシステム定義メンバーに、説明を定義できます。また、通貨用にシステムで生成される値ディメンションのメンバーの場合、通貨の説明に追加される説明を定義できます。

値メンバーの構文:

```
Label  
;  
Descriptions
```

システム定義の値メンバーのラベルを指定できます。また、これらのラベルを使用して、これに対応するユーザー定義の通貨用にシステムで生成される値メンバーの説明に追加する説明を作成できます。

```
<Currency Total>
```

```
<Currency Adjs>
```

```
<Currency>
```

たとえば、USDとEURのそれぞれの通貨を"USドル"および"ユーロ"と定義したとします。また、ロードされたメタデータ・ファイルに、これらの値メンバーの説明が定義されているとします。

```
[None];English=ValueNone
```

```
<Currency Total>;English=Total
```

```
<Currency Adjs>;English=Adjs
```

```
<Currency>;English=Base
```

103ページの表 17に、USDおよびユーロの通貨用にシステムで生成される、値ディメンションの3つのメンバーを示します。

表17 値ディメンションの説明

値メンバー	説明
USD Total	USDドル合計
USD Adjs	USDドル調整
USD	USDドル基本
EUR Total	ユーロ合計
EUR Adjs	ユーロ調整
EUR	ユーロ基本



注:

メタデータ・ファイルには、値メンバー<Currency>に対して指定した説明が含まれている必要があります。メタデータ・ファイルで通貨の説明が指定されていない場合に、メタデータをロードすると、通貨の説明は表示されません。

内部取引パートナー

メンバー・セクションを使用して、内部取引パートナー(ICP)ディメンションの次のシステム定義メンバーに、セキュリティ・クラスおよび説明を定義できます。

- [ICP Top]
- [ICP None]
- [ICP Entities]

内部取引パートナー・メンバーの構文:

```
Label  
;  
SecurityClass  
;  
Descriptions
```

この例では、セキュリティ・クラスを指定せずに[ICP Top]、[ICP None]、[ICP Entities]の説明を定義する方法を示します。

```
[ICP Top];;English=Top ICP
```

```
[ICP None];;English=No ICP
```

```
[ICP Entities];;English=Entities ICP
```

連結メソッド

メタデータ・ファイルのこのセクションは、連結メソッドを定義します。

連結メソッドの構文:

```
Label;UsedByCalcRoutine;IsHoldingMethod;ToPercentControlComp;  
ToPercentControl;PercentConsol;Control;Descriptions
```

[81ページの連結メソッドの定義](#)を参照してください。

この例では、連結メソッドの属性を指定します。

```
!CONSOLIDATION_METHODS
```

```
M2;Y;N;<=;20;0;No
```

```
M3;Y;N;<;50;POWN;Limited
```

```
M4;Y;N;<=;50;50;Limited
```

```
M1;Y;Y;<=;100;100;Full
```

```
M5;Y;N;<=;100;100;Full
```

階層

メタデータ・ファイルのこのセクションは、親子関係を定義します。親子関係はノードとも呼ばれます。ノードには、独自の属性値セットを指定できます。

ノード・レコードはカンマで区切られたリストです。リストの各行にある最初の2つのアイテムは、親と子を示します。区切り文字を使用すると、欠落した値を空白として表せます。階層の最上位のすべてのメンバーは、空白の親の子として表示されます。



ヒント:

カスタム・ディメンションのノード・レコードには、3番目の属性が含まれています。[108ページのカスタム階層](#)を参照してください。

階層セクションは、次の行で開始します。<>の中に実際のディメンション名を入力してください。

```
!HIERARCHIES=<>
```

カスタム・ディメンションのセクションを開始するときにスペースを含めないでください。たとえば、FLOWディメンションの階層セクションは、次の行で開始します。

```
!HIERARCHIES=Customers
```

```
;[None];1
```

```
;AllCustomers;0
```

```
AllCustomers;Customer2;1
```

```
AllCustomers;Customer3;1
```

```
AllCustomers;Customer4;1
```

```
AllCustomers;Customer5;1
```



注:

カスタム・ディメンションの場合、このセクションは、カスタム・ディメンションの別名を使用する必要があります。

次の各トピックでは、勘定科目、シナリオ、エンティティおよびカスタムの各ディメンションの階層セクションのフォーマットを説明します。

勘定科目階層

勘定科目ディメンションの階層の構文:

```
parentmemberlabel  
;  
childmemberlabel
```

この例では、勘定科目ディメンションの階層を指定します。

```
!HIERARCHIES=Account
```

```
:[None]
```

```
;ExchangeRates
```

```
ExchangeRates;Rate1
```

```
ExchangeRates;Rate2
```

```
;Plug
```

```
;NetProfit
```

```
NetProfit;NetIncome
```

```
NetIncome;GrossMargin
```

```
GrossMargin;Sales
```

```
GrossMargin;TotalCosts
```

```
TotalCosts;Purchases
```

```
TotalCosts;Salaries
```

```
TotalCosts;OtherCosts
```

```
NetIncome;AdminExpenses
```

```
NetIncome;InterestCharges
```

```
NetProfit;Taxes
```

シナリオ階層

シナリオ・ディメンションの階層の構文:

```
parentmemberlabel  
;  
childmemberlabel
```

この例では、シナリオ・ディメンションの階層を指定します。

```
!HIERARCHIES=Scenario
```

```
;Actual
```

```
;Budget
```

エンティティ階層

エンティティ・ディメンションの階層の構文:

```
parentmemberlabel  
;  
childmemberlabel
```

この例では、エンティティ・ディメンションの階層を指定します。

```
!HIERARCHIES=Entity
```

```
;[None]
```

```
;Regional
```

```
Regional;UnitedStates
```

```
UnitedStates;California
```

```
California;Sunnyvale
```

```
California;FosterCity
```

カスタム階層

カスタム・ディメンションの階層の構文:

```
parentmemberlabel  
;  
childmemberlabel  
;  
AggregationWeight
```

この例は、カスタム・ディメンションの階層を指定しています:

```
!HIERARCHIES=Products
```

```
;[None];1
```

```
;AllProducts;0
```

```
AllProducts;Golf;1
```

```
Golf;GolfBalls;1
```

```
Golf;GolfShoes;1
```

```
Golf;GolfTees;1
```

メタデータ・ファイルに含まれないディメンション

年、期間およびビューの各ディメンションは、メタデータ・ファイルには含まれません。これらのディメンションは、アプリケーションを定義するときに指定するアプリケーション・プロファイルで定義します。

値および内部取引パートナーのディメンション・メンバーは、そのほとんどがシステムで定義されます。ただし、値ディメンションのメンバーの説明、および内部取引パートナー・ディメンションのセキュリティ・クラスや説明は定義できます。

- 値: 標準メンバーは自動的に生成されます。また、メタデータをロードした後、ロードされた各通貨に使用される値ディメンションの次の3つのメンバーが自動的に作成されます: *CurrencyName*、*CurrencyName Adjs*および*CurrencyName Total*。ここで、*CurrencyName*は、通貨ラベルです。たとえば、通貨USDの場合は、Financial Managementにより次の値ディメンション・メンバーが作成されます: USD、USD Adjs、USD Total。システム生成メンバーの説明や、ユーザー定義の通貨用にシステムで生成される値メンバーに追加される説明を定義できます。
- 内部取引パートナー: このディメンションは自動生成されます。内部取引パートナー・ディメンションのメンバーは、IsICP属性がTRUEに設定されているエンティティ・ディメンションのメンバーごとに生成されます。103ページの内部取引パートナーで説明されているように、いくつかの内部取引パートナー・メンバーのセキュリティ・クラスおよび説明を定義できます。

メタデータ・マネージャのビューの使用方法

メタデータ・ファイルの編集および作成には、メタデータ・マネージャを使用します。メタデータには、ディメンションのメンバー属性や通貨に関する情報が含まれています。たとえば、メタデータ・マネージャを使用して、勘定科目をアプリケーションに追加できます。



注:

メタデータ・マネージャを使用できるのは、Financial ManagementのWindowsクライアントのみです。

メタデータ・マネージャでは、Extensible Markup Language(XML)およびAPPのファイル形式でファイルを開くことができます。メタデータ・マネージャを使用して、XMLまたはAPPファイルの作成や、Financial Managementアプリケーションから抽出したXMLまたはAPPファイルの編集ができます。メタデータ・マネージャで作成したメタデータ・ファイルは、Little Endianバイト・オーダーによって自動的にUnicodeフォーマットにエンコードされます。メタデータ・ファイルの作成または編集が終了したら、ファイルに含まれているメタデータをアプリケーションにロードできます。

メタデータをアプリケーションで直接編集することはできません。メタデータ・マネージャでメタデータを変更すると、変更されるのはファイルのみで、アプリケーションのメタデータは変更されません。ファイルのメタデータを抽出して変更したら、変更が有効になるようにアプリケーションにメタデータ・ファイルを再ロードする必要があります。

110ページの表 18に、メタデータ・マネージャのビューを示します。

表18 メタデータ・マネージャのビュー

タブ	説明
ツリー・ビュー	階層ビューを使用して、メンバーを追加または変更します。 注： AppSettings、ConsolMethodおよびCurrenciesでは、リスト・ビューを使用する必要があります。
リスト・ビュー	フラット・リスト・フォーマットを使用して、メンバーを追加または変更します。
ファイルのプロパティ	メタデータ・レポートを作成するために使用します。

大半のメタデータでは、ツリー・ビューまたはリスト・ビューでメンバーを追加したり、変更したりできます。



注:

AppSettings、ConsolMethodsおよびCurrenciesのメタデータでは、リスト・ビューを使用する必要があります。

ツリー・ビューでは、階層ビューのメンバーを追加または変更できます。各メンバーは、メンバーの子または兄弟として追加します。各メンバーの属性は、「メンバー属性」タブで入力します。データを変更するには、メンバーを選択し、メンバーまたはその属性を変更します。[111ページのツリー・ビューでのタスク](#)を参照してください。

リスト・ビューでは、フラット・リストのメンバーおよびメンバー属性を追加および変更します。階層のメンバーを並べ替えるには、ツリー・ビューに切り替えて、メンバーを階層にドラッグします。[115ページのリスト・ビューでのタスク](#)を参照してください。



注:

ツリー・ビューに加えられた変更はリスト・ビューに適用され、リスト・ビューに加えられた変更はツリー・ビューに適用されます。

メタデータ・ファイルのフォーマットの変更

メタデータ・マネージャでは、ファイルを必要な拡張子で保存することで、APPファイルとXMLファイルを相互に変換できます。

▶ XMLファイルをAPPファイルとして保存するには:

1. 「ファイルを開く」をクリックし、変換するXMLファイルを選択します。
2. 「ファイルの保存」をクリックします。
3. 「保存のタイプ」ドロップダウン・リストで、「APPファイル(*.app)」を選択します。
4. ファイル名を変更し、ファイルの拡張子を「APP」に変更します。
5. 「保存」をクリックします。



注:

メタデータ・マネージャで変換した直後のAPPファイルを開くと、区切り文字を入力するよう求めるメッセージが表示されます。区切り文字は、APPファイルにのみ必要です。

ツリー・ビューでのタスク

このトピックのすべての手順は、メタデータ・マネージャでメタデータ・ファイルを開いていることが前提となっています。メタデータ・ファイルを変更したら、必ずファイルを保存してください。次の手順を参照してください:

[111ページのメンバーの追加と変更](#)

[112ページのカスタム・ディメンションのノード属性の変更](#)

[113ページのメンバー・リストからのメンバーの追加](#)

[113ページのメンバーの移動](#)

[113ページのメンバーのレベルを上げる](#)

[114ページのメンバーの位置の変更](#)

[114ページのメンバーの削除と除去](#)

[114ページの孤児メンバーの追加](#)

[115ページの孤児メンバーの除去](#)

[115ページの階層の展開と縮小](#)

メンバーの追加と変更

新しい階層にメンバーを追加する場合、このメンバーはTopMemberの子として追加します。たとえば、メタデータ・ファイルを作成して勘定科目を設定する場合、デフォルトでTopMemberが「勘定科目」という名前になります。子勘定科目は、この「勘定科目」に対してのみ追加できます。TopMemberを右クリックすると名前を変更できます。



注:

ツリーに同じ親が複数ある場合は、親のすべてのインスタンスに子メンバーが追加されます。

ツリー・ビューでメンバーを追加するか、既存のメンバーを選択すると、メタデータ・マネージャのワークスペースの右側にそのメンバーの属性が表示されます。



ツリー・ビューの右側には、[112ページの表 19](#)で説明するその他のタブがあります。

表19 ツリー・ビューのタブ

タブ	説明
メンバー属性	選択したメンバーの属性を表示、編集、入力します。
メンバー・リスト	リストから階層にメンバーをドラッグします。
ノード属性	カスタム・ディメンションに使用します。このタブには、現在選択されているカスタム・ディメンションのメンバーとその親との関係に適用できる属性が含まれています。

▶ メンバーを追加または変更するには:

1. 「ツリー・ビュー」タブが選択されていることを確認し、**メタデータ・アイテム**・ドロップダウン・リストからディメンションを選択します。
2. 次のいずれかのタスクを行います:

- ・ 特定のメンバーに子メンバーを追加するには、メンバーを選択して、をクリックします
- ・ 特定のメンバーに兄弟メンバーを追加するには、メンバーを選択して、をクリックします
- ・ メンバーを変更するには、メンバーを選択します。



注:

子を追加できるのは、階層内のTopMemberのみです。

3. 「**メンバー属性**」タブで、メンバーの属性を入力または変更します。
4. 新しいエントリを確定するには、**[Tab]**キーを押します。
5. メンバーをすべて追加または更新するまで、これらの手順を繰り返します。

カスタム・ディメンションのノード属性の変更

集約加重プロパティは、親に集約されるカスタム・メンバーの割合を指定し、1は100%を意味します。選択したカスタム・メンバーとその親との関係を変更するには、「ノード属性」タブで集約加重の属性を変更します。たとえば、集約加重を.5に設定すると、メンバーの値の50%のみが親に集約されます。[70ページのカスタム・メンバーの定義](#)を参照してください。

カスタム・ディメンションの集約加重には、任意の値(正または負の値)および分率(たとえば、1.5は有効な値)を指定できます。デフォルト値は0(集約なし)および1(集約あり)です。値が0または1以外の場合は、乗数の値を使用して子メンバーが親に集約されます。

▶ ノード属性を変更するには:

1. 「ツリー・ビュー」タブが選択されていることを確認し、「**メタデータ・アイテム**」ドロップダウン・リストからカスタム・ディメンションを選択します。
2. 階層内でカスタム・メンバーをハイライトします。
3. **メタデータ・マネージャ**のワークスペースの右側で、「**ノード属性**」タブを選択します。
4. AggrWeight属性を変更します。

メンバー・リストからのメンバーの追加

メンバーを「メンバー・リスト」タブからドラッグして階層構造にドロップすることで、階層にメンバーを追加できます。親メンバーに子メンバーを追加し、階層内に同じ親が複数存在する場合は、親メンバーのすべてのインスタンスに子メンバーが追加されます。複数のメンバーを同時に追加できます。

▶ 「メンバー・リスト」タブからメンバーを追加するには:

1. 「ツリー・ビュー」タブが選択されていることを確認し、メタデータ・アイテム・ドロップダウン・リストからディメンションを選択します。
2. 「メンバー・リスト」タブから、追加するメンバーを選択します。



注:

複数のメンバーを選択するには、[Ctrl]キーを押しながら、メンバーを選択していきます。

3. 選択したメンバーを階層内にドラッグします。



注:

メンバーは、階層にコピーされるだけで、「メンバー・リスト」タブからは除去されません。

メンバーの移動

階層内のメンバーをドラッグして移動できます。複数のメンバーを同時に移動できます。

▶ メンバーを移動するには:

1. 「ツリー・ビュー」タブが選択されていることを確認し、メタデータ・アイテム・ドロップダウン・リストからディメンションを選択します。
2. 階層内でメンバーを選択します。




注:

複数のメンバーを選択するには、[Ctrl]キーを押しながら、メンバーを選択していきます。

3. 選択したメンバーを別の場所にドラッグします。

メンバーのレベルを上げる

ツールバーの「レベルを上げる」ボタン  を使用して、階層内のメンバーのレベルを上げることができます。メンバーのレベルを上げると、メンバーが階層内の1つ上のレベルに移動します。



▶ メンバーのレベルを上げるには:

1. 「ツリー・ビュー」タブが選択されていることを確認し、メタデータ・アイテム・ドロップダウン・リストからディメンションを選択します。
2. 階層内で、レベルを上げるメンバーを選択します。

3. をクリックします。

メンバーの位置の変更

ツールバーの「上へ移動」および「下へ移動」ボタンを使用して、階層内のメンバーの位置を変更します。位置の変更では、メンバーが新しいレベルに上がるわけではなく、階層の現在のレベル内で上下に移動します。

- ▶ メンバーを兄弟に対して位置変更するには:
1. 「ツリー・ビュー」タブが選択されていることを確認し、メタデータ・アイテム・ドロップダウン・リストからディメンションを選択します。
 2. 階層内で、位置を変更するメンバーを選択します。
 3. またはをクリックして、選択したメンバーの位置を変更します。

メンバーの削除と除去

メンバーをメタデータから完全に削除したり、メンバーを親から除去しても「メンバー・リスト」タブには残したりできません。メンバーをその親から除去したときに、そのメンバーに他の親がない場合は、「メンバー・リスト」タブで孤児メンバーとして表示されます。階層からメンバーを完全に削除すると、そのメンバーはリスト・ビューからも削除されます。

- ▶ メンバーを削除または除去するには:
1. 「ツリー・ビュー」タブが選択されていることを確認し、メタデータ・アイテム・ドロップダウン・リストからディメンションを選択します。
 2. 階層内でメンバーを選択します。
 3. メニュー・オプションを右クリックして選択します。
 - ・ 親から除去: メンバーはその親から除去されますが、「メンバー・リスト」タブには残ります。
 - ・ ディメンションから削除: メンバーはディメンションから削除されます。

孤児メンバーの追加

孤児メンバーは、階層に含まれていないメンバーです。そのため、このメンバーには親または兄弟のメンバーはありません。リスト・ビューで追加したメンバーは、ツリー・ビューで階層に追加するまで、孤児になります。



注意

孤児メンバーが含まれるメタデータ・ファイルは、スキャンしたり、アプリケーションにロードしたりできません。

- ▶ 孤児メンバーを階層に追加するには:
1. 「ツリー・ビュー」タブが選択されていることを確認し、メタデータ・アイテム・ドロップダウン・リストからディメンションを選択します。
 2. メタデータ・マネージャのワークスペースの右側で、「メンバー・リスト」タブを選択します。
 3. 「孤児メンバーのみを表示」を選択して、ディメンションのすべての孤児メンバーを表示します。

4. メンバーをハイライトして、階層内の新しい場所にドラッグします。

孤児メンバーの除去

孤児メンバーのみを表示すると、メンバー・リストから孤児メンバーを簡単に除去できます。



注意

孤児メンバーが含まれるメタデータ・ファイルは、スキャンしたり、アプリケーションにロードしたりできません。

▶ 孤児メンバーを除去するには:

1. 「ツリー・ビュー」タブが選択されていることを確認し、メタデータ・アイテム・ドロップダウン・リストからディメンションを選択します。
2. メタデータ・マネージャのワークスペースの右側で、「メンバー・リスト」タブを選択します。
3. 「孤児メンバーのみを表示」を選択して、選択したディメンションの孤児メンバーを表示します。
4. メンバーをハイライトし、ハイライトしたメンバーの削除を右クリックして選択します。





注:

すべての孤児メンバーを除去するには、右クリックして「すべて選択」を選択します。

階層の展開と縮小

ツールバーの「展開」ボタンおよび「縮小」ボタンを使用して、階層を展開または縮小できます。階層全体を展開または縮小したり、個々の親メンバーを展開または縮小できます。

▶ ツリーを展開または縮小するには:

1. 「ツリー・ビュー」タブが選択されていることを確認し、「メタデータ・アイテム」ドロップダウン・リストからディメンションを選択します。
2. 階層を展開するには、メンバーを選択し、をクリックして、階層内の現在のメンバーとその下位にあるすべてのメンバーを展開します。
3. 階層を縮小するには、メンバーを選択し、をクリックして、階層内の現在のメンバーとその下位にあるすべてのメンバーを縮小します。

リスト・ビューでのタスク

このトピックのすべての手順は、メタデータ・マネージャでメタデータ・ファイルを開いていることが前提となっています。メタデータ・ファイルを変更したら、必ずファイルを保存してください。



注:

リスト・ビューでメンバーを追加した後は、ツリー・ビューでメンバーを階層に追加します。

次の手順を参照してください:

[116ページのメンバーの追加と変更](#)

[116ページのメンバーのコピー、切り取りおよび貼付け](#)

[117ページのメンバーの削除](#)


[117ページのメタデータの検証](#)

[117ページの列の表示または非表示](#)

メンバーの追加と変更

リスト・ビューを使用してメンバーを追加すると、フラット・リストにメンバーとその属性を入力できます。追加または変更した情報は、ツリー・ビューなどの別のビューに変更するか、手動でデータを検証するまで検証されません。

▶ メンバーを追加または変更するには:

1. 「リスト・ビュー」タブが選択されていることを確認し、**メタデータ・アイテム**・ドロップダウン・リストからディメンションを選択します。
2.  をクリックし、新しい行にメンバー名と属性を入力します。
3. メンバーを変更するには、変更したいメンバーを選択して、メンバーまたはその属性を変更します。
4. メンバーの追加および変更がすべて完了するまで、これらの手順を繰り返します。

メンバーのコピー、切り取りおよび貼付け

メンバーを1つのセルから別のセルに、または複数のセルから複数のセルに切り取り、コピーおよび貼付けできます。また、ドロップダウン・リストのセルを使用不可にすることもできます。ドロップダウン・リストのセルを使用不可にすると、複数のセルのコピー、切り取り、貼付けが簡単にできます。リスト・ビューで加えられた変更は、ツリー・ビューに反映されます。

▶ メンバー、アプリケーション設定または通貨の切り取り、コピーまたは貼付けを行うには:



1. 「リスト・ビュー」タブが選択されていることを確認し、**メタデータ・アイテム**・ドロップダウン・リストからディメンションを選択します。
2. オプション: 「コンボボックスを使用不可にする」を選択します。
3. 1つのセル、複数のセル、または連続したセルを選択します。




注:

行全体または列全体を選択するには、行番号または列ヘッダーを選択します。

4. 次のいずれかのアクションを実行します:


- セルの情報を切り取ってクリップ・ボードに保管するには、 をクリックします。
- セルの情報をコピーしてクリップ・ボードに保管するには、 をクリックします。

クリップ・ボードの情報を選択したセルに貼り付けるには、をクリックします。

メンバーの削除

メンバーを削除するときは、そのメンバーに関連付けられているすべての属性も削除する必要があります。そのため、メンバーを削除するには、メンバー行全体を選択する必要があります。リスト・ビューで加えられた変更は、ツリー・ビューに反映されます。

▶ メンバーを削除するには:

1. 「リスト・ビュー」タブが選択されていることを確認し、**メタデータ・アイテム**・ドロップダウン・リストからディメンションを選択します。
2. 行番号をクリックして、行をハイライトします。
3. をクリックします。

メタデータの検証

メンバーに対して行った変更は、ツリー・ビューなどの別のビューに変更するか、ツールバーの「検証」ボタンを選択するまで検証されません。検証プロセス中にエラーが見つかった場合は、エラーが発生した行番号が表示されます。

列の表示または非表示

列を表示または非表示にすることにより、表示する情報を含む列を指定できます。

▶ 1列のみを非表示にするには:

1. 「リスト・ビュー」タブが選択されていることを確認し、「**メタデータ・アイテム**」ドロップダウン・リストからディメンションを選択します。
2. 列を右クリックし、「**現在の列の非表示**」を選択します。

▶ 複数の列の表示と非表示を切り替えるには:

1. 「リスト・ビュー」タブが選択されていることを確認し、「**メタデータ・アイテム**」ドロップダウン・リストからディメンションを選択します。
2. グリッドを右クリックし、「**列の表示/非表示**」を選択します。
3. 表示する列を選択します。
4. 「**OK**」をクリックします。

リスト・ビューのメタデータのソート

リスト・ビューのメタデータを列でソートできます。列にある情報は、昇順または降順にソートできます。

▶ 列をソートするには:

1. 「リスト・ビュー」タブが選択されていることを確認し、「**メタデータ・アイテム**」ドロップダウン・リストからディメンションを選択します。

2. 列ヘッダーをダブルクリックし、列にある情報をソートします。



ヒント:

逆の順序で列をソートするには、もう一度列ヘッダーをダブルクリックします。

「ファイルのプロパティ」でのメタデータ・レポートの作成

メタデータをHTMLフォーマットに変換するときにXSLスタイル・シートを適用すると見やすくなります。

Financial Managementでは、メタデータ・レポート用にデフォルトで2つのレポートXSLスタイル・シートが用意されています。たとえば、階層のメタデータを表示するレポートの場合は、HFM_MetadataWithHierarchy.XSLスタイル・シートを使用します。

インストール・プロセス中にスタイル・シートがインストールされます。スタイル・シートは、デフォルトで、Financial Managementのインストール先ディレクトリのReport Style Sheets\Metadataフォルダに配置されます。用意されているスタイル・シートを使用せず独自のスタイル・シートを作成する場合、XSLのサポートについては、管理者に問い合わせてください。


▶ メタデータ・レポートを作成するには:

1. メタデータ・マネージャで、「ファイルのプロパティ」タブを選択し、メタデータ・ファイルを開きます。



注:

メタデータ・ファイルには、デフォルトでXMLまたはAPP拡張子が使用されます。

2. オプション: レポートを生成する前にメタデータを表示するには、「XMLの表示」をクリックします。
3. 「XSLスタイル・シート・ファイル名」テキスト・ボックスに、メタデータ・ファイルに適用するスタイル・シートの名前を入力するか、をクリックしてファイルを検索します。



注:

メタデータ・レポート・スタイルシートには、デフォルトでXSLというファイル拡張子が使用されます。

4. フォーマット済のレポートを表示するには、「メタデータの表示」をクリックします。

メタデータの参照整合性

Financial Managementでは、アプリケーションで参照整合性の問題が発生しないように、メタデータを変更する前に、変更がアプリケーションの現在の状態に合っているかどうかを確認されます。

メタデータをロードすると、メタデータのロード・ファイルとアプリケーションのメタデータ要素が比較されます。変更はすべて記録され、いくつかの変更は既存のデータに対して確認されます。参照整合性の問題が発生する変更はできません。

メタデータの参照整合性チェック

ログ・ファイルには、ロードしているメタデータ・ファイルに関連した既存データを確認する必要があるメタデータ属性の、特定の変更に関する情報が含まれています。

また、ロード・ファイルとアプリケーションのメタデータとの間の無効な視点についても確認されます。ディメンションのメンバーがロード・ファイルにはないがアプリケーションの仕訳に存在する場合、メタデータのロードは行われません。

メタデータ・ログ・ファイルの参照整合性エラー

参照整合性エラーは、メタデータのログ・ファイルの、次のセクションに表示されます。

```
Metadata referential integrity check started at
```

参照整合性チェック・セクションの各行は、ロード・ファイルのメタデータの整合性エラーを示します。整合性チェックで見つかったエラーは、次のフォーマットで表示されます。

```
Journals::SINGLECA1 Scenario::Actual Year::2014
Value::[Contribution Adjs]
Period::January has 1 occurrences of
Changed::[SCENARIO::Actual::ZeroViewForAdj: Periodic to YTD]
```

この例では、次の視点において、SINGLECA1仕訳でメタデータの整合性エラーが発生しています: シナリオがActual、年が2014、値が[Contribution Adjs]、期間がJanuary。エラーは、ActualシナリオのZeroViewForAdj属性が、PeriodicからYTDに変更されたことを示しています。Actualシナリオの仕訳が存在しているため、このような変更はできません。

メタデータのマージ・ユーティリティの使用

既存のリリースから新規リリースにアップグレードする場合、メタデータのマージ・ユーティリティを使用して、既存のメタデータ・ファイルを最新バージョンのファイルとマージできます。これにより、既存のメタデータ・ファイルで行われた変更を保持できます。このユーティリティは、Financial Management¥Utilitiesフォルダにあります。



注:

メタデータのマージ・ユーティリティは、XMLフォーマットのメタデータでのみ機能します。APPフォーマットでは機能しません。

使用方法

```
MetadataMerge.bat -b<Base File>[i<Ignore descriptions>]-l <Latest file>-m
<Modified file>[-o <Output File>]
```


ここで:

`-b,-base <Base File>` = ベース・バージョンのメタデータ・ファイルのフル・パス(拡張子を持つファイル名を含む)

`-l,-latest <Latest File>` = リリース・バージョンのメタデータ・ファイルのフル・パス(拡張子を持つファイル名を含む)

`-m,-modified <Modified File>` = 変更されたメタデータ・ファイルのフル・パス(拡張子を持つファイル名を含む)

`-o,-output <Output File>` = 出力メタデータ・ファイルのパス。ここに、更新されたメタデータおよびメタデータの差分レポートが保存されます

`-i,-ignoredesc <Ignore Descriptions>` = メンバーの説明の変更を無視します

例

`-b c:\temp\MetadataMerge\Comma_V1_B.xml`

`-l c:\temp\MetadataMerge\Comma_V2_R.xml`

`-m c:\temp\MetadataMerge\Comma_Customer_M.xml` (Comma_V1_B.xmlファイルに基づいてユーザーが変更したメタデータ)

`-i true`

コマンド: `MetadataMerge.bat -b c:\temp\MetadataMerge\Comma_Metadata_B.xml -l c:\temp\MetadataMerge\Comma_Metadata_R.xml -m c:\temp\MetadataMerge\Comma_Metadata_M.xml -i true`

▶ メタデータのマージ・ユーティリティを使用するには:

1. ファイル・エクスプローラまたはコマンドラインから `MetadataMerge.bat` を実行します。
2. マージ処理中にメタデータの競合が存在する場合は、警告メッセージが表示されます。次のいずれかのオプションを選択します。
 - Y - はい。最新のリリース・ファイルの変更を適用し、その変更を既存のファイルにマージします。
 - N - いいえ。最新のリリース・ファイルの変更を適用しません。
 - MA - すべてマージ。すべての変更が最新のリリース・ファイルから適用されます。他の競合については要求されません。
 - MN - マージしない。変更は適用されません。他の競合については要求されません。
3. `Images` フォルダにある2つのイメージを、メタデータの差分レポートの.htmlファイルが生成されるパスにコピーします。これらのイメージは、メタデータの差分レポートのツリー用の「展開」および「縮小」アイコンです。

メタデータのロード

メタデータ・ファイルをロードすると、Financial Managementにより、メタデータがロード・ファイルの新しいメタデータに置換されます。置換は、勘定科目の追加などメタデータに簡単な変更を行うときに便利です。たとえば、アプリケー

ションにNorth Americaエンティティが含まれており、エンティティをメタデータ・ファイルからロードする場合、ファイルのNorth Americaエンティティの属性により、アプリケーションのNorth Americaエンティティが置換されます。



注:

メタデータ・ファイルでアンパサンド文字(&)を使用しないでください。そのようにした場合、エラーが発生します。

メタデータ・ファイルをロードする場合、ファイル・ロードを続行する前に、連結、データ入力またはその他のロード・プロセスなどその他のタスクが終了するまで待機する必要があります。メタデータのロードは、長時間にわたる連結中などではなく、サーバー・クラスタ全体で小規模のアクティビティ実行中に行うことをお勧めします。「タスクの実行」ページでは、どの連結ロードまたはデータ・ロードが処理中かなどをチェックできます。

大きなメタデータ・ファイルをロードすると、プロキシのタイムアウト・エラーが発生することがあります。このエラーが発生した場合は、Webプロキシのタイムアウト設定を大きくしてください。

アプリケーションにメタデータ・ファイルがロードされた後、そのアプリケーションを使用しているユーザーに対し、システムが変更されたのでアプリケーションからログオフして再びログオンする必要があることが通知されます。



注意

メタデータをロードする前に、孤児メンバーを削除する必要があります。孤児メンバーが削除されていないと、メタデータは更新されません。

マージまたは置換ロード・オプションを選択する必要があります。新しいメタデータをロードする前にメタデータをすべてクリアすることができ、データの整合性も確認できます。121ページの表 20では、ロード・オプションを説明します。

表20 メタデータのロード・オプション

ロード・オプション	説明
マージ	ディメンション・メンバーがロード・ファイルとアプリケーションのデータベースに存在する場合、データベースのメンバーはロード・ファイルのメンバーに置換されます。ロード・ファイルに参照されない他のディメンション・メンバーがデータベースにある場合、データベースのメンバーは変更されません。 たとえば、データベースにCT、MAおよびCAのエンティティがあるとして、ロード方法としてマージを選択して、CAの新しい情報のみを含むメタデータ・ファイルをロードします。データベースでは、CAは新しい情報で更新され、MAとCTはそのままデータベースに残され、変化しません。
置換	アプリケーション・データベース内のすべてのディメンション・メンバーが削除され、ロード・ファイルのメンバーがデータベースに入れられます。 たとえば、データベースにCT、MAおよびCAのエンティティがあるとして、ロード方法として置換を選択して、CAの

ロード・オプション	説明
	新しい情報のみを含むメタデータ・ファイルをロードします。データベースでは、CTとMAは削除され、エンティティは、ロード・ファイルの新しい情報で更新されたCAのみになります。
ロードする前にメタデータをすべてクリア	アプリケーション・データベース内のすべてのディメンション・メンバーと、対応するデータ、仕訳および内部取引が削除されます。 注： このオプションを選択すると、マージおよび置換のロード方法より優先されます。
整合性チェック	メタデータとデータをチェックして、整合性を確認します。 118ページのメタデータの参照整合性 を参照してください。



注:

Webのポップアップ・ブロックの例外にFinancial Managementを追加することをお勧めします。データのロードなどのいくつかのタスクを実行すると、ステータス・ウィンドウにタスクのステータスが表示されます。コンピュータでポップアップ・ブロックを使用可能にしている場合、このステータス・ウィンドウは表示されません。

▶ メタデータをロードするには:

1. アプリケーションを開きます。
2. 「連結」、「ロード」、アプリケーション要素の順に選択します。
3. 「メタデータ」セクションにロードするファイル名を入力するか、「参照」をクリックしてファイルを検索します。



注:

メタデータ・ファイルには、デフォルトでXMLまたはAPP拡張子が使用されます。ロード・プロセスでは、TXTやCSVなどの他のファイル拡張子も受け入れますが、APPまたはXMLファイル拡張子の使用をお勧めします。

4. オプション: 「メタデータとデータのクリア」を選択します。



注:

このオプションを選択する場合は、「メタデータ・オプション」セクションの要素は選択できません。

5. オプション: 「整合性の確認」を選択して、データ・ファイルを現在のアプリケーション内のメタデータと照合します。

アプリケーションがロード・ファイルのメタデータの悪影響を受けないようにするために、このオプションを選択することを強くお勧めします。



注:

整合性エラーが発生した場合は、ログ・ファイルにエラーが記録され、メタデータ・ファイルはアプリケーションにロードされません。

この手順を続行する前に、エラーを修正する必要があります。[119ページのメタデータ・ログ・ファイルの参照整合性エラー](#)を参照してください。

6. 「ロード・オプション」セクションで、ロード方法を選択します:

- マージ
- 置換

7. 「区切り文字」から、ファイル内のメタデータを区切るために使用する文字を選択します。

区切り文字は、APPファイル拡張子を持つASCIIファイルにのみ必要になります。区切り文字は、XMLファイルでは必要ありません。次の文字が有効です:

```
, ~ @ $ % ^ | : ; ? \
```



注:

ファイル内の別の場所やファイル名で使われていない文字を使用します。たとえば、エンティティの説明にカンマを使用する場合、カンマを区切り文字として使用できません。

8. 「フィルタ」セクションで、ロードするメタデータのタイプを選択します。



ヒント:

フィルタの選択をリセットするには、「リセット」をクリックします。

9. オプション: ファイル・フォーマットが正しいかどうかを確認するには、「スキャン」をクリックします。

10. 「ロード」をクリックします。

メタデータ・ロードの変更の表示

メタデータ・ファイルが正常にロードされた後で、ロード・プロセス中に行われたメタデータの変更に関する詳細情報を表示できます。たとえば、勘定科目ディメンションで属性が変更された場合、古い属性と新しい属性の値が表示されます。タスク監査モジュールから、この情報のレポートを表示できます。

メタデータ・ロードの変更レポートには、次の詳細が含まれます:

- 追加されたメンバー
- 削除されたメンバー
- Oracle Hyperion EPM Architectを使用して名前が変更されたメンバー
- メンバー属性の変更(説明の変更は含まない)

- 新規の親/子関係
- 削除された親/子関係
- カスタム・ディメンションでの親子関係の集約加重の変更

メタデータ・ロードのレポートのサンプル出力

**** Entity ****

Added member WestRegion.

Changed IsICP for China from Y to N.

Added Parent/Child Regional/Asia.

Removed Parent/Child Europe/Bulgaria.


**** Account ****

Renamed member SalesIC to SalesInterco.

Deleted member EastSales.

**** Scenario ****

Added member Forecast.

- ▶ メタデータ・ロードの変更を表示するには:
1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
 2. 「管理」、「監査」、「タスク」の順に選択します。
 3. タスク監査の「処理状態」列で、表示するメタデータ・ロード・プロセスの横にあるリンク・アイコン  をクリックします。
 4. 「開く」をクリックし、テキスト・エディタでレポートを表示します。

メタデータの抽出

メタデータを抽出して、これをメタデータ・マネージャで表示または変更できます。メタデータを抽出するときは、ファイルをXMLまたはAPPファイルとして保存し、ファイル名とファイルの場所を指定します。メタデータを変更したら、変更が有効になるように、変更したファイルをアプリケーションに再ロードする必要があります。

値ディメンションなどのシステム定義のディメンションのメンバーは、抽出できません。また、アプリケーション・プロファイルで定義された年や期間などのディメンションのメンバーも抽出できません。

▶ メタデータを抽出するには:

1. アプリケーションを開きます。
2. 「連結」、「抽出」、アプリケーション要素の順に選択します。
3. 「メタデータ」セクションで、「区切り文字」から、ファイル内のメタデータを区切るために使用する文字を選択します。

区切り文字は、.appファイル拡張子を持つASCIIファイルにのみ必要になります。区切り文字は、XMLファイルでは必要ありません。次の文字が有効です:

, ~ @ \$ % ^ | : ; ? ¥



注:

ファイル内の別の場所やファイル名で使われていない文字を使用します。たとえば、エンティティの説明にカンマを使用する場合、カンマを区切り文字として使用しないでください。

4. ファイル・フォーマットを選択します:
 - **Financial Management (.app)**
 - **Financial Management (.xml)**
 - **EPM Architect (.ads)**
5. 「フィルタ」で、抽出するメタデータのタイプを選択します。
6. 「抽出」をクリックします。
7. ブラウザに表示されるダウンロードの手順に従います。

使用しているWebブラウザによって、手順が異なります。ファイルは、設定したWebディレクトリに保存してください。

5

メンバー・リストの管理

この項の内容:

メンバー・リスト・ファイルの作成	128
メンバー・リストのロード	137
メンバー・リストの抽出	138
システム生成のリストとディメンション	138

メンバー・リストを使用すると、ディメンション内のメンバーのサブセットを指定できるようになり、データ・グリッドやデータ・フォームでメンバー階層を参照する時間を短縮できます。たとえば、エンティティ・ディメンションの視点で Italy、France および UK が頻繁に使用される場合、頻繁に使用されるメンバーを含む European region という名前のメンバー・リストを作成できます。エンティティ・ディメンションの階層から参照するのではなく、リストからメンバーを選択できます。

メンバー・リストは、ルール内でも使用できます。たとえば、メンバー・リストのすべてのメンバーを計算するルールを作成できます。

Financial Management では、2つのタイプのメンバー・リストが使用されます:

- システム定義メンバー・リスト
- ユーザー定義メンバー・リスト

アプリケーションを作成してメタデータをロードすると、システム定義メンバー・リストが生成されます。これらのメンバー・リストでは、指定された親のすべての子、または指定されたディメンションのすべての基本メンバーなど、共通のプロパティに基づいてメンバーがグループ化されます。システム定義メンバー・リストは、[Descendants] のように、大カッコで囲まれています。

ユーザー指定のディメンション・メンバーを含む、ユーザー定義メンバー・リストを作成できます。たとえば、データ・グリッドまたは仕訳で視点を設定するとき、またはデータベース管理でデータをコピーまたはクリアするときに、メンバー・リストを選択できます。Financial Management のすべてのディメンションのメンバー・リストを作成できます。

静的および動的メンバー・リストを作成できます。静的メンバー・リストには、ユーザー指定のディメンション・メンバーが含まれます。たとえば、次の勘定科目のみが含まれる、ProfitAndLoss と呼ばれる勘定科目の静的メンバー・リストを作成できます: Sales、Purchases、Salaries、OtherCosts、TotalCosts。静的リストにメンバーを追加するには、メンバーを手動で追加する必要があります。

動的メンバー・リストには、実行時に、指定した条件に一致するメンバーが含まれます。リストは動的に作成されるため、更新は取得時に行われます。シナリオ、年、期間およびエンティティ・ディメンションでは、POV の現在のメンバーをリストの開始点として使用できます。



注:

静的メンバー・リストと動的メンバー・リストを、同じファイルで組み合わせることができます。

サンプル・メンバー・リスト・ファイルは、Financial Managementのサンプル・アプリケーションをインストールするときに取り込まれます。これらのファイルは、Financial Managementをインストールしたディレクトリの「Sample Applications」フォルダに配置されます。

メンバー・リスト・ファイルの作成

Notepad ++などのテキスト・エディタを使用してメンバー・リスト・ファイルを作成し、そのファイルをアプリケーションにロードできます。[245ページのNotepad ++でのルールの作成](#)を参照してください。

Microsoft Visual Basicのスクリプト構文を使用して、静的リストと動的リストのそれぞれにメンバーを追加できます。

ユーザー定義メンバー・リスト名には引用符は使用できません。デフォルトでは、メンバー・リスト・ファイルにLSTファイル拡張子が使用されます。

メンバー・リスト・ファイルを作成するには、次のサブルーチンを使用します。

- Sub [128ページのEnumMemberLists](#) (): どのディメンションにメンバー・リストがあるかを指定し、各ディメンションのメンバー・リストを定義します。各ディメンションのメンバー・リストの数やそのディメンション内にある各メンバー・リストの名前を定義します。
- Sub [129ページのEnumMembersInList](#) (): 各メンバー・リスト内のメンバーを定義します。



注:

HS.Dimension関数は、ディメンションのディメンション別名を戻します。ディメンション名およびディメンション別名(ロング名)を指定できるカスタム・ディメンションで、HS.Dimension関数は、ディメンション別名(ロング名)を戻します。

スクリプトへの動的メンバー・リストの追加については、[132ページの動的メンバー・リスト](#)を参照してください。

EnumMemberLists

各メンバー・リスト・ファイルにEnumMemberListsサブルーチンを含めて、メンバー・リストを含めるディメンション、各ディメンションのリスト数、および各メンバー・リストの名前を指定する必要があります。EnumMemberLists ()サブルーチンでは、次の表に示す構文および関数を使用して、メンバー・リストを定義します。

構文	説明
Dim <i>ElementLists</i> (<i>n</i>)	各文のリスト数を指定します <i>Element</i> はメンバー・リストを作成しているディメンションの名前で、 <i>n</i> はそのディメンションに定義しているメン

構文	説明
	<p>バー・リストの合計数です。たとえば、ファイルにエンティティ・ディメンションの3つのメンバー・リストが含まれている場合、構文は次のようになります:</p> <pre style="text-align: center;">Dim EntityLists(3)</pre>
HS.Dimension = " <i>Element</i> "	<p>ここで、<i>Element</i>はディメンション名です。例:</p> <pre style="text-align: center;">If HS.Dimension = "Entity" Then</pre>
<i>ElementLists</i> (<i>n</i>) = " <i>ListName</i> "	<p>各リストの名前と数値IDを指定します。ここで、<i>Element</i>はディメンション名、<i>n</i>はメンバー・リストのID番号、<i>ListName</i>はリストの名前です。 @POVキーワードを使用して、現在POVに設定されているディメンション・メンバーに基づいた動的リストを作成できます。エンティティ・リストに示されるエンティティは、レポートのPOVで選択されるシナリオ、年、期間およびエンティティを基本にできます。例:</p> <pre style="text-align: center;">EntityLists(1) = "NewEngland"</pre> <pre style="text-align: center;">EntityLists(2) = "Alloc"</pre> <pre style="text-align: center;">EntityLists(3) = "AllEntities(@POV)"</pre>
HS.SetMemberLists <i>ElementLists</i>	<p>指定された名前および数値IDを設定します。ここで、<i>Element</i>はディメンション名です。例:</p> <pre style="text-align: center;">HS.SetMemberLists EntityLists</pre>

EnumMembersInList

EnumMembersInListサブルーチンを使用して、リストにメンバーを追加します。静的メンバー・リストの場合は、スク립トにリストのすべてのメンバーをリストします。EnumMembersInList ()サブルーチンでは、次の表に示す構文および関数を使用して、各メンバー・リストのメンバーを定義します。

エンティティ以外のすべてのディメンションでは、HS.AddMemberToList文を使用して、メンバーを指定する必要があります。エンティティ・ディメンションの場合は、HS.AddEntityToList文を使用して、メンバーとその親を指定する必要があります。

表21 EnumMembersInListの構文

構文	説明
HS.Dimension = " <i>Element</i> "	<i>Element</i> はディメンションです。例:

構文	説明
	<pre>If HS.Dimension = "Entity" Then</pre>
HS.MemberListID= <i>n</i>	<p>数値IDでメンバー・リストを指定します。ここで、<i>n</i>は、EnumMemberListsサブルーチンのメンバー・リストに割り当てられている数値IDです</p> <pre>HS.MemberListID = 1</pre>
HS.AddEntityToList <i>Member</i> HS.AddMemberToList <i>Member</i>	<p>エンティティ以外のディメンションのリストにメンバーを追加します。ここで、<i>Member</i>はメンバー名です</p> <pre>HS.AddEntityToList "UnitedStates", "Maine" HS.AddMemberToList "July"</pre>
HS.AddEntityToList <i>Parent, Member</i>	<p>エンティティ・ディメンションのリストにメンバーを追加します。ここで、<i>Parent</i>は追加しているメンバーの親で、<i>Member</i>はエンティティ・ディメンションのメンバーです。</p> <pre>HS.AddEntityToList "UnitedStates", "California"</pre>
HS.Entity.List	<p>エンティティ・ディメンションのリストにメンバーを追加します</p> <pre>HS.Entity.List("", "[Base])</pre>
HS.MemberListEntity HS.MemberListScenario HS.MemberListYear HS.MemberListPeriod	<p>動的メンバー・リストの指定に使用します。</p>

ここでは、ファイルのEnumMembersInListセクションのサンプルを示します。この例では、3つのエンティティ・リストのエンティティが定義されています。また、勘定科目リストのメンバーも定義されています。

```
Sub EnumMembersInList()
```

```
If HS.Dimension = "Entity" Then
```

```
    If HS.MemberListID = 1 Then
```

```
        HS.AddEntityToList "UnitedStates", "Connecticut"
```

```
        HS.AddEntityToList "UnitedStates", "Massachusetts"
```

```
        HS.AddEntityToList "UnitedStates", "RhodeIsland"
```

```
        HS.AddEntityToList "UnitedStates", "Maine"
```

```
    ElseIf HS.MemberListID = 2 Then
```

```
        HS.AddEntityToList "UnitedStates", "Connecticut"
```

```
    ElseIf HS.MemberListID = 3 Then
```

```
        HS.AddEntityToList "UnitedStates", "California"
```

```
    End If
```

```
ElseIf HS.Dimension = "Account" Then
```

```
    If HS.MemberListID = 1 Then
```

```
        HS.AddMemberToList "Sales"
```

```
        HS.AddMemberToList "Purchases"
```

```
        HS.AddMemberToList "Salaries"
```

```
HS.AddMemberToList "OtherCosts"
```

```
HS.AddMemberToList "TotalCosts"
```

```
HS.AddMemberToList "GrossMargin"
```

```
HS.AddMemberToList "HeadCount"
```

```
HS.AddMemberToList "AdminExpenses"
```

```
HS.AddMemberToList "InterestCharges"
```

```
HS.AddMemberToList "NetIncome"
```

```
HS.AddMemberToList "Taxes"
```

```
HS.AddMemberToList "NetProfit"
```

```
End If
```

```
End If
```

```
End Sub
```



注:

各ディメンションで必要に応じてメンバー・リストをいくつでも作成できます。すべてのディメンションにメンバー・リストを作成する必要はありません。

動的メンバー・リスト

動的メンバー・リストの場合は、メンバー・リストのすべてのメンバーをリストするかわりに、指定した条件に一致するメンバーを選択するルールを入力します。条件は、通貨や勘定科目の種類などのメンバー・プロパティです。リストは、ユーザーがアクセスするたびに動的に生成されます。

メンバー・リストを構築するには、Financial Managementの関数および引数を使用できません。[245ページのNotepad ++でのルールを作成](#)を参照してください

次の構文では、USDエンティティをすべて取得する動的メンバー・リストを作成します。

```
If HS.Dimension = "Entity" Then
```

```
  If HS.MemberListID=1 Then
```

```
    ELi=HS.Entity.List(" ", " ")
```

```
    'Entities are read into an array.
```

```
    For i=Lbound(ELi) to Ubound(ELi)
```

```
      'Loops through all entities.
```

```
        If (StrComp(HS.Entity.DefCurrency(ELi(i)),  
"USD",vbTextCompare)=0) Then
```

```
          HS.AddEntityToList " ",ELi(i)
```

```
          'String compares default currency for entity to USD. If there is a match, the  
entity is added to the member list.
```

```
        End If
```

```
      Next
```

```
    End If
```

```
  End If
```

この例では、エンティティのリストは配列に受け入れられます。配列の各エンティティでは、DefaultCurrencyプロパティの値がUSDの優先値と比較されます。値がUSDと同一である場合、エンティティがリストに追加されます。その後、配列の次のエンティティが処理されます。

動的POVメンバー・リスト

動的POVメンバー・リストは、1つ以上のディメンションの現在のPOVメンバーに基づいて動的に作成されます。

次のメンバー・リスト・ファイルの例では、動的POVのセクションを太字で示しています。

```
Sub EnumMemberLists()
```

```
Dim EntityLists(5)
```

```
If HS.Dimension = "Entity" Then
```

```
EntityLists(1) = "AllEntities"
```

```
EntityLists(2) = "AppCur"
```

```
EntityLists(3) = "NoAppCur"
```

```
EntityLists(4) = "Global(@POV)"
```

```
EntityLists(5) = "POWN(@POV)"
```

```
HS.SetMemberLists EntityLists
```

```
End If
```

```
End Sub
```

```
Sub EnumMembersInList()
```

```
If HS.Dimension = "Entity" Then
```

```
If HS.MemberListID = 1 Then
```

```
HS.AddEntityToList "", "Corp_Ops"
```

```
HS.AddEntityToList "", "China"
```

```
HS.AddEntityToList "", "Colombia"
```

```
HS.AddEntityToList "", "Germany"
```

```
HS.AddEntityToList "", "Spain"
```

```
HS.AddEntityToList "", "UK"
```

```
End If
```

```
EntList=HS.Entity.List("", "")
```

```
AppCur=HS.AppSettings.Currency
```

```
For each Ent in EntList
```

```
  If HS.Entity.DefCurrency(Ent)=AppCur Then
```

```
    If HS.MemberListID = 2 Then HS.AddEntityToList "", Ent
```

```
  ElseIf Ent<>"[None]" Then
```

```
    If HS.MemberListID = 3 Then HS.AddEntityToList "", Ent
```

```
  End If
```

```
Next
```

```
ScenPOV=HS.MemberListScenario
```

```
YearPOV=HS.MemberListYear
```

```
PerPOV=HS.MemberListPeriod
```

```
EntPOV=HS.MemberListEntity
```

```
If HS.MemberListID = 4 Or HS.MemberListID = 5 Then
```

```
  If ( EntPOV <> "" ) Then
```

```
    EntList=HS.Node.List("E#" & EntPOV, "[Descendants]", "S#" & ScenPOV &  
    ".Y#" & YearPOV & ".P#" & PerPOV)
```

```
    If IsArray(EntList) Then
```

```
      For each Ent in EntList
```

```
        If Ent <> "[None]" Then
```

```
          If HS.Node.Method("S#" & ScenPOV & ".Y#" & YearPOV & ".P#" &  
          PerPOV & ".E#" & EntPOV & "." & Ent)="GLOBAL" Then
```

```
            If HS.MemberListID = 4 Then HS.AddEntityToList "",Ent
```

```
          End If
```

```
          If HS.Node.POwn("S#" & ScenPOV & ".Y#" & YearPOV & ".P#" & PerPOV  
          & ".E#" & EntPOV & "." & Ent) > 0.5 Then
```

```
            If HS.MemberListID = 5 Then HS.AddEntityToList "",Ent
```


End If

End If

Next

End If

End If

End If

End Sub

メンバー・リストのロード

メンバー・リスト・スクリプト・ファイルを作成したら、アプリケーションにロードします。ファイルをロードする前にスキャンして、正しくフォーマットされているか検証できます。ファイルを編集する際は、アプリケーションに再ロードする必要があります。

メンバー・リスト・ファイルをロードする場合、ファイル・ロードを続行する前に、連結、データ入力またはその他のロード・プロセスなどのその他のタスクが終了するまで待機する必要があります。メンバー・リストのロードは、長時間にわたる連結中などではなく、サーバー・クラスタ全体で小規模のアクティビティ実行中に行うことをお勧めします。「タスクの実行」ページでは、どの連結ロードまたはデータ・ロードが処理中かなどをチェックできます。

アプリケーションにメンバー・リスト・ファイルがロードされた後、そのアプリケーションを使用しているユーザーに対し、システムが変更されたのでアプリケーションからログオフして再びログオンする必要があることが通知されます。算出ステータスもOK SC (システムの変更)に変更されます。



注:

Webのポップアップ・ブロックの例外にFinancial Managementを追加することをお勧めします。データのロードなどのいくつかのタスクを実行すると、ステータス・ウィンドウにタスクのステータスが表示されます。コンピュータでポップアップ・ブロックを使用可能にしている場合、このステータス・ウィンドウは表示されません。

▶ メンバー・リストをロードするには:

1. アプリケーションを開きます。
2. 「連結」、「ロード」、アプリケーション要素の順に選択します。

3. 「メンバー・リスト」セクションにロードするファイル名を入力するか、「参照」をクリックしてファイルを検索します。



注:

デフォルトでは、メンバー・リスト・ファイルにLSTファイル拡張子が使用されます。ロード・プロセスでは、TXTやCSVなどの他のファイル拡張子も受け入れますが、LSTファイル拡張子の使用をお勧めします。

4. オプション: ファイル・フォーマットが正しいかどうかを確認するには、「スキャン」をクリックします。
5. 「ロード」をクリックします。



注:

無効なメンバー・リストがロードされた場合、Webユーザーがデータ・グリッドにアクセスできないことがあります。次のエラー・メッセージが表示される場合は、メンバー・リストを修正してファイルをロードすることが必要になる場合があります:

```
"Type Mismatch /hfm/Data/ExploreData.asp. Error Number 13. Error Source: Microsoft VBScript runtime error."
```

メンバー・リストの抽出

アプリケーションからメンバー・リストを抽出できます。抽出されたメンバー・リストは、ASCIIファイルとして保存されます。デフォルトでは、メンバー・リスト・ファイルにLSTファイル拡張子が使用されます。メンバー・リストを抽出した後は、テキスト・エディタでメンバー・リストの情報を表示および変更できるようになります。

▶ メンバー・リストを抽出するには:

1. アプリケーションを開きます。
2. 「連結」、「抽出」、アプリケーション要素の順に選択します。
3. 「メンバー・リスト」セクションで、「抽出」をクリックします。
4. ブラウザに表示されるダウンロードの手順に従います。

使用しているWebブラウザによって、手順が異なります。ファイルは、設定したWebディレクトリに保存してください。

システム生成のリストとディメンション

次の表に、システム生成のリストの名前と、各リストがそれぞれのディメンションで使用できるかどうかを示します。

表22 システム生成のリストとディメンション

システム生成のリスト	シナリオ	エンティティ	勘定科目	ICP	カスタム	値	年	期間
[Hierarchy]	X	X	X	X	X	X	X	X

システム生成のリスト	シナリオ	エンティティ	勘定科目	ICP	カスタム	値	年	期間
[Descendants]	X	X	X	X	X	X	X	X
[Children]		X	X	X	X			X
[Base]		X	X	X	X			
[Parents]		X						
[Ancestors]		X						
[System]			X	X				
[Currencies]					C1, C2			
[ConsolMethod]					C1			
[Inputs]						X		
[Adjustments]						X		
[Totals]						X		
[Default Currencies]						X		
[First Generation]								X
[Second Generation]								X
[Third Generation]								X
[Fourth Generation]								X
[Fifth Generation]								X
[Sixth Generation]								X

6

仕訳の管理

この項の内容:

仕訳ファイルの作成	141
仕訳のロード	146
仕訳の抽出	147

外部の総勘定元帳システムの多くは、仕訳情報を含むASCIIテキスト・ファイルを生成します。ASCIIテキスト・ファイルは、Financial Managementのアプリケーションにロードできます。必要な場合、ASCIIファイルはロードする前にテキスト・エディタを使用して編集できます。

サンプルの仕訳帳ファイルは、Financial Managementのサンプル・アプリケーションをインストールするときに取り込まれます。これらのファイルは、Financial Managementをインストールしたディレクトリの「Sample Applications」フォルダに配置されます。

仕訳ファイルの作成

仕訳ファイルは、マルチバイト文字セット(MBCS)対応のASCIIフォーマットを使用するか、Little Endianバイト・オーダーによってUnicodeフォーマットにエンコードされたファイルを使用して作成できます。デフォルトでは、仕訳ファイルにはJLFファイル拡張子が使用されます。

セキュリティ情報ファイルには、次のセクションを含めることができます。

- ファイル・フォーマット
- バージョン
- 仕訳グループ
- 標準
- 繰返し
- ヘッダー - シナリオ、年、期間

感嘆符(!)で始まる行は、仕訳ファイルの新しいセクションの開始を示しており、その後に有効なセクション名(Yearなど)を続ける必要があります。アポストロフィ(')で始まる行は、コメント行とみなされ無視されます。

以下の特殊文字は、ファイル内で別の用途で使用されていないかぎり、ファイル内の情報を区切るために使用できます。

文字	説明
&	アンパサンド
@	アット・マーク

文字	説明
¥	円記号(バックスラッシュ)
^	脱字記号
:	コロソ
,	カンマ
\$	ドル記号
#	シャープ
%	パーセント記号
	パイプ記号
?	疑問符
;	セミコロソ
~	チルド



注:

ファイル全体で同じ区切り文字を使用する必要があります。同じファイル内で異なる区切り文字を使用すると、ファイルのロード時にエラーが発生します。

ファイル・フォーマット・セクション

このファイル・セクションには、バージョン番号が含まれます。この番号は、ファイル・フォーマットに変更があった場合にのみ変更されます。ファイル・フォーマットは、仕訳を抽出すると自動的に生成されます。



注:

このセクションは、必須ではありません。

この例では、ファイル・フォーマットを指定した例を示します。

```
!File_Format = 1.0
```

バージョン・セクション

このファイル・セクションには、Financial Managementのバージョン番号が含まれます。バージョン番号は、仕訳を抽出すると自動的に生成されます。



注:

このセクションは、必須ではありません。

この例では、バージョンを指定します。

```
!Version = 11.1
```

仕訳グループ・セクション

このファイル・セクションでは、次の構文を使用して仕訳グループを定義します。

```
!GROUP=<journal group>;<journal group description>
```

たとえば、次の例では2つの仕訳グループを定義しています：

```
!GROUP=Allocations;Allocations Journals Group  
!GROUP=Tax;Tax Journals Group
```

標準セクション

標準テンプレートは、アプリケーションのすべてのシナリオ、年および期間に適用されます。このテンプレートは、特定の組合せのシナリオ、期間および年に基づいたものではありません。

次に、標準テンプレートの構文を示します。

```
!STANDARD = <label>, <balancing attribute>, <type>, <journal group>,  
<securityclass>, <SingleParent.SingleEntity>
```

```
!DESC=<journal description>
```

```
<parent.entity>, <account>, <ICP>, <CustomDimensionName>, <amount type>,  
<amount>, <line item desc>
```

繰返しセクション

繰返しテンプレートは、アプリケーションのすべてのシナリオ、年および期間に適用されます。このテンプレートは、Value Adjsに基づいていますが、特定の組合せのシナリオ、期間および年に基づいたものではありません。



注：

自動逆仕訳テンプレート用に繰返しテンプレートを作成することはできません。タイプ属性の値は、通常のRにする必要があります。

次に、繰返しテンプレートの構文を示します。

```
!RECURRING = <label>, <balancing attribute>, <type>, <value>, <journal  
group>, <securityclass>, <SingleParent.SingleEntity>
```

```
!DESC=<journal description>
```

```
<parent.entity>, <account>, <ICP>, <CustomDimensionName>, <amount type>,  
<amount>, <line item desc>
```

ヘッダー・セクション

このファイル・セクションには、シナリオ、年および期間に関する情報が含まれます。仕訳ファイルのヘッダー情報とこれに対応する明細行がヘッダー・セクションの後に続きます。ここでは、特定のシナリオ、年および期間に基づいて仕訳情報が編成されます。仕訳ファイルには、複数のヘッダー・セクションを指定できます。

この構文では、シナリオ、年および期間を指定します：

```
!SCENARIO= Actual
```

```
!YEAR = 2014
```

```
!PERIOD = January
```

ヘッダー・セクションの次に、Actualというシナリオの、2014年のJuly(1月)の仕訳の詳しい情報を入れます。

ヘッダー・セクションの仕訳サブセクションでは、指定したシナリオ、期間および年の仕訳詳細情報を定義します。

次の [144ページの表 23](#)では、仕訳サブセクションで使用される属性の説明を示します。これらの属性は、通常の仕訳、および繰返しテンプレートと標準テンプレートで使用されます。

表23 仕訳の属性の説明

属性	値
<ラベル>	ユーザー定義の仕訳のラベル(最大20文字まで)
<調整属性>	<ul style="list-style-type: none">• U = 貸借不一致• B = 貸借一致• E = エンティティで貸借一致
<タイプ>	<ul style="list-style-type: none">• R = 通常仕訳• A = 自動逆仕訳• V = 承認済自動逆仕訳 <p>注： 逆仕訳の転記後にシステムで生成された逆仕訳をロードすることはできませんが、抽出することはできます。</p>
<ステータス>	<ul style="list-style-type: none">• W = 作業中

属性	値
	<ul style="list-style-type: none"> • S = 提出済 • A = 承認済 • P = 転記済 • R = 却下済
<値ディメンション>	<ul style="list-style-type: none"> • [Contribution Adjs] • [Parent Adjs] • <Entity Curr Adjs> • <Parent Curr Adjs>
<仕訳グループ>	<p>オプション: ユーザー定義パラメータ(最大30文字)</p> <p>注:</p> <p>あらかじめグループがプレロードされている必要があります。</p>
<セキュリティ・クラス>	<p>オプション: 仕訳に関連付けられた有効なセキュリティ・クラス。</p> <p>注:</p> <p>セキュリティ・クラスを割り当てない場合、仕訳では、デフォルトのセキュリティ・クラスとみなされます。このセキュリティ・クラスにアクセス権を持っているユーザーのみが、仕訳にアクセスできます。</p>
<singleparent.singleentity>	<p>オプション: 仕訳のすべてのライン・アイテムで使用される有効な親/エンティティの組合せ。仕訳全体で単一の親/エンティティの組合せを指定する場合、parent.entity属性は使用されません。</p>
<仕訳の説明>	<p>仕訳の説明(最大255文字)。各行が「!DESC=」構文で始まる場合は、複数行の説明をロードできます。</p>
<parent.entity>	<p>エンティティ・ディメンションの有効なメンバー。値ディメンションのContribution Adjs, Parent AdjsおよびParentCurrency Adjsの各メンバーにのみ、親が必要です。この属性は、Single Parent.Single Entity属性が使用されていない場合にのみ使用されます。</p>
<勘定科目>	<p>勘定科目ディメンションの有効なメンバー。標準仕訳では、入力勘定になります。勘定科目の種別は、REVENUE(収益)、EXPENSE(費用)、ASSET(資産)、LIABILITY(負債)、FLOW(フロー)またはBALANCE(残高)です。</p>
<ICP>	<p>オプション: 内部取引パートナー・ディメンションの有効なメンバー。この属性はオプションですが、少なくとも「ICP None」と指定する必要があります。</p>
<custom>	<p>オプション: カスタム・ディメンションの有効なメンバー</p>

属性	値
<amount type>	<ul style="list-style-type: none"> • D = 借方 • C = 貸方
<金額>	金額タイプが借方か貸方かにかかわらず、正の金額を指定します
<ライン・アイテムの説明>	オプション: 仕訳詳細の説明(最大50文字)。

次の構文によって、通常仕訳を指定します。

```
!JOURNAL = <label>, <balancing attribute>, <type>, <status>, <value dimension>, <journal group>, <SecurityClass>, <SingleParent.SingleEntity>
```

```
!DESC=<journal description>
```

```
<parent.entity>, <account>, <ICP>, <CustomDimensionName>, <amount type>, <amount>, <line item desc>
```

仕訳のロード

作業中、却下済、提出済、承認済および転記済の仕訳と、標準および繰返しの仕訳テンプレートをロードできます。連結プロセスによって作成されるため、自動連結仕訳はロードできません。

仕訳をロードする前に、最初に仕訳をロードする期間を開いておく必要があります。『Oracle Hyperion Financial Management ユーザー・ガイド』の期間の管理に関する項を参照してください。

仕訳は、置換モードでロードされます。これにより、新しい仕訳データがロードされる前に、特定仕訳ラベルのデータすべてがクリアされます。開いている期間には、転記済の仕訳をロードする必要があります。転記済の自動逆仕訳をロードする場合、次の期間に承認済の逆仕訳が自動的に生成され、生成された逆仕訳を手動で転記する必要があります。

デフォルトでは、仕訳ファイルにはJLFファイル拡張子が使用されます。ロード・プロセスでは、TXTやCSVなどのファイル拡張子を受け入れますが、JLFファイル拡張子の使用をお勧めします。

デフォルトのロード・オプションを変更すると、オプションがすべての行に対して更新されます。特定の行を有効にしてその行のオプションを更新する上書きオプションを使用できます。

ロード・プロセスが完了すると、ログのリンクが表示されるため、エラーを表示できます。



注:

Webのポップアップ・ブロッカの外例にFinancial Managementを追加することをお勧めします。データのロードなどのいくつかのタスクを実行すると、ステータス・ウィンドウにタスクのステータスが表示されます。コンピュータでポップアップ・ブロッカを使用可能にしている場合、このステータス・ウィンドウは表示されません。

▶ 仕訳をロードするには:

1. 「連結」、「ロード」、「仕訳」の順に選択します。
2. 「区切り文字」に、ファイルの仕訳データを区切るために使用する区切り文字を入力します。次の文字が有効です:

, ~ @ \$ % & ^ | : ; ? \



注:

ファイル内の別の場所やファイル名で使われていない文字を使用します。たとえば、エンティティの説明にカンマを使用する場合、カンマを区切り文字として使用できません。

3. オプション: Firefoxをブラウザとして使用している場合は、複数のファイルをロードできます。まだ選択していない場合は「複数の選択」を選択し、「参照」をクリックしてファイルを検索します。

一度に最大10個のファイルを選択できます。次に、システムによって選択したファイルが詳細行に移入されません。

また、ファイルをファイル選択コンテンツ領域にドラッグ・アンド・ドロップできます。

4. オプション: 「追加」をクリックして、仕訳をロードするために行をさらに追加します。



ヒント:

行を削除するには、行を選択して「削除」をクリックします。

5. オプション: 行のデフォルトのファイル設定を上書きするには、「上書き」をクリックします。



ヒント:

ロード・オプションをデフォルト値にリセットするには、「リセット」をクリックします。

6. 「ロード」をクリックします。



注:

既存のファイルを再ロードすると、それらのファイルを再度使用するかどうかを確認する警告プロンプトが表示されます。行う場合、「はい」をクリックします。

仕訳の抽出

アプリケーションから、仕訳、繰返し仕訳および仕訳テンプレートを抽出できます。仕訳を抽出するシナリオ、年、期間、エンティティおよび値ディメンションを選択できます。また、抽出する仕訳ステータス、仕訳タイプおよび仕訳の残高の種類を選択できます。

仕訳を抽出すると、マルチバイト文字セット(MBCS)をサポートするASCIIファイルに保存されます。デフォルトでは、仕訳ファイルにはJLFファイル拡張子が使用されます。メンバー・リストを抽出した後は、テキスト・エディタでメンバー・リストの情報を表示および変更できるようになります。

自動連結仕訳を外部システムに抽出できますが、その仕訳をFinancial Managementに再インポートできません。これらの仕訳を抽出する場合、自動連結仕訳の残高の種類は空白です。

抽出プロセスが完了すると、ログのリンクが表示されるため、エラーを表示できます。

▶ 仕訳を抽出するには:

1. 「連結」、「抽出」、「仕訳」の順に選択します。
2. 仕訳を抽出するシナリオおよび年を指定します。
3. オプション: 仕訳を抽出する期間を選択します。
4. オプション: 抽出するエンティティおよび値ディメンションを、1つ以上選択します。

自動連結仕訳を抽出するには、値メンバーとして[Proportion]または[Elimination]、あるいはその両方を選択する必要があります。



注:

期間、エンティティおよび値ディメンションでは、特定のメンバーを選択しないと、ディメンションのすべてのメンバーを抽出するとみなされます。ただし、メンバーを明示的に選択すると、ディメンションの横に複数の選択があることを示すプラス記号(+)が表示されます。

5. 「区切り文字」に、ファイルの仕訳データを区切るために使用する区切り文字を入力します。次の文字が有効です:

, ~ @ \$ % & ^ | : ; ? \



注:

ファイル内の別の場所やファイル名で使われていない文字を使用します。たとえば、エンティティの説明にカンマを使用する場合、カンマを区切り文字として使用できません。

6. 抽出する仕訳の種類を選択します:
 - テンプレート
 - 繰返しテンプレート
 - 仕訳
7. オプション: 抽出する仕分けのラベルおよびグループを入力します。

ワイルドカードとしてパーセント記号(%)を使用できます。

自動連結仕訳の場合、連結ルールで指定した監査取引の種類値に対して「ラベル」フィールドで問合せを実行できます。ラベルには番号を使用できません。

8. 抽出する仕訳のステータス、種類および残高の種類を選択するか、「すべて」を選択します。
9. 「抽出」をクリックします。
10. ブラウザに表示されるダウンロードの手順に従います。

使用しているWebブラウザによって、手順が異なります。ファイルは、設定したWebディレクトリに保存してください。

7

データ・フォームの管理

この項の内容:

フォーム・デザイナーでのデータ・フォームの作成	151
データ・フォーム・スクリプトの使用	155
相対期間の使用	195
競合する属性の優先順位	196
データ・フォームの編集	197
データ・フォームのロード	197
データ・フォームの抽出	198
データ・フォームの削除	198

データ・フォームは、特定のデータを表示および入力するために設定された、事前定義済の行および列が含まれるテンプレートです。ユーザーは、データ・フォームへの行および列の追加や削除を行うことはできません。変更できるのは、フォームの設定時に指定されているディメンションの視点の選択のみです。あるフォームから別のフォームへのリンクを定義し、リンクされたフォーム内のデータをユーザーがドリル・スルーして表示できるようにすることが可能です。

データ・フォームを作成するには、管理者またはデータ入力フォームの管理のセキュリティ役割に割り当てる必要があります。データ・フォームのロード、抽出および削除には、「データ入力フォームの管理」役割が割り当てられている必要があります。

データ・フォームは2つの方法で作成できます:

- フォーム・デザイナーのデータ・フォーム・オプションの使用
- スクリプト・ビューのスクリプトの書込み

デフォルトでは、データ・フォームのファイル拡張子はWDFです。

デザイナーおよびスクリプト・ビューを容易に切り替えることができます。スクリプトにエラーが含まれる場合、スクリプト・ビューで表示されます。スクリプト・ビューから、データ・フォームを印刷することもできます。

データ・フォームのデザイナー・ビューから「フォームを開く」に切り替えることができます。フォームの保存を求めるプロンプトが表示され、データ・フォームが表示されます。

フォーム・デザイナーでのデータ・フォームの作成

フォーム・デザイナーを使用して、視点を設定し、フォームの詳細、行、列およびヘッダー・オプションを指定します。

データ・フォームの作成後、スキャンして妥当性を確認できます。フォーム・スクリプトが生成および検証されます。エラーがある場合、スクリプト・ビューに表示されます。[155ページのデータ・フォーム・スクリプトの使用](#)を参照してください。

フォームの詳細の設定については、次の項目を参照してください:

- [152ページの視点の設定](#)
- [153ページのフォームの詳細の指定](#)
- [154ページのフォームの行および列オプションの指定](#)
- [155ページのフォームのヘッダーの指定](#)

▶ データ・フォームを作成するには:

1. 「連結」、「ドキュメント」の順に選択します。
2. 「新規」、「データ・フォーム」の順にクリックするか、「アクション」、「新規」、「データ・フォーム」の順に選択します。

デフォルトでは、フォーム・デザイナーが表示されます。



ヒント:

「デザイナー」ページが自動的に表示されない場合、「デザイナー」をクリックするか、「アクション」、「デザイナー」の順に選択します。

3. データ・フォームの視点を設定します。
4. フォームの詳細を指定します。
5. 行および列オプションを指定します。
6. オプション: 行または列を追加するには、「新規行の追加」、新規列の追加の順にクリックするか、「アクション」、「新規行の追加」または新規列の追加の順に選択します。



ヒント:

行または列を削除するには、「列/行の削除」をクリックするか、「アクション」、「列/行の削除」の順に選択します。

7. ヘッダー・オプションを指定します。



ヒント:

フォームをデフォルト値または最後に保存した定義にリセットするには、「リセット」をクリックします。

8. オプション: 「スキャン」をクリックして、フォームの妥当性を確認します。
9. 「保存」をクリックしてフォームを保存し、フォームの情報を入力して、「保存」をクリックします。

視点の設定

POVバーを使用して、バックグラウンドPOVおよび選択可能POVを定義できます。フォームのバックグラウンドPOVは、各ディメンションに、データ・フォームに表示される初期値を指定します。フォームのSelectable POVは、ユーザーが選択できるメンバーのディメンションで構成されます。

POVバーのディメンションには、メンバーを選択してバックグラウンドPOVメンバーを指定します。メンバー・リストを選択すると、データ・フォーム・ユーザーがメンバーを選択できる選択可能POVとして使用されます。2つを超えるアイテムおよび1つを超えるメンバーまたはメンバー・リストは選択できません。

動的POVメンバー・リストを定義した場合、それを使用して、選択したディメンションに対して有効なメンバーを迅速に選択できます。「メンバー選択」ダイアログ・ボックスで、選択したディメンションの有効な動的POVメンバー・リストが、ダイアログ・ボックスの下部に接尾辞@POVを付けて表示されます。

POVを使用する場合は、次のルールが適用されます。

- メンバーまたはメンバー・リストが選択されず、行または列でディメンションが使用されていない場合は、ディメンションの初期値としてユーザーPOVのメンバーが使用されます。
- ディメンションのバックグラウンドおよび選択可能POVに値を指定し、バックグラウンドPOVのメンバーが選択可能POVのメンバー・リストに含まれていない場合は、選択可能POVのメンバー・リストにある最初のメンバーがバックグラウンドPOVメンバーとして使用されます。
- 階層内のすべてのメンバーを操作できるように、ユーザーには、メンバー階層へのフル・アクセス権限が必要です。たとえば、ユーザーがカスタム4のすべてのメンバーにアクセスする場合、このケースでは、親エンティティ CustomTopにアクセスできるようにする必要があります。

フォーム・デザイナー・グリッドは、行および列を含みます。新しいフォームの場合、デフォルトでは、シナリオ・ディメンションがグリッド列に使用され、期間ディメンションがグリッド行に使用されます。列または行をフォームに追加し、POVバーからグリッドの行および列にディメンションをドラッグ・アンド・ドロップしてデータ・フォーム・レイアウトを定義できます。同じ行または列に1つを超えるディメンションをドラッグ・アンド・ドロップする場合、ディメンションの順序を変更できます。ディメンションをクリックすると、「メンバー選択」ページからメンバーを選択してグリッドまたはPOVに含めることができます。

グリッドの行または列を選択すると、対応する行および列のオプションが「フォームの詳細」プロパティ・ペインに表示されます。行または列に計算式を含めることができます。この場合、行または列にディメンションをドラッグ・アンド・ドロップできないので、代わりにプロパティ・ペインを使用して、行または列に計算式を入力する必要があります。行に使用されるディメンションは、列に使用できません。この逆の場合も同様です。

フォームの詳細の指定

「フォームの詳細」セクションでは、データ・フォームのグリッド、印刷、表示および非表示オプションなどのフォーム・プロパティを指定します。データ・フォームのユーザーに指示を含め、データ・フォームで使用可能なオンデマンド・ルールを指定できます。



注:

「フォームの詳細」セクションのアイテムは、デフォルト設定を変更した場合のみスクリプトで更新されます。デフォルト設定を変更しないでおくと、これらのアイテムのキーワードはスクリプトで表示されません。

各オプションの有効な値については、[155ページのデータ・フォーム・スクリプトの使用](#)を参照してください。

データ・フォームのオンデマンド・ルールの指定

データ・フォームから実行できるオンデマンド・ルールを作成できます。オンデマンド・ルールは、計算のサブセットのみを実行して、結果をデータ・フォームで迅速に確認する場合に便利です。たとえば、特定のオンデマンド・ルールを実

行するよう設定されたデータ・フォームで作業している場合、データを入力し、オンデマンド・ルールを実行して、計算の結果を迅速に確認できます。

アプリケーション・ルール・ファイルでオンデマンド・ルールを作成します。ルールは新しいサブルーチンに作成され、接頭辞OnDemand (例: OnDemand_Calculation)で識別されます。[246ページのルール・ファイルの作成](#)を参照してください。

▶ データ・フォームのオンデマンド・ルールを指定するには:

1. 「連結」、「ドキュメント」の順に選択します。
2. 「新規」、「データ・フォーム」の順にクリックするか、「アクション」、「新規」、「データ・フォーム」の順に選択します。

デフォルトでは、フォーム・デザイナーが表示されます。



ヒント:

「デザイナー」ページが自動的に表示されない場合、「デザイナー」をクリックするか、「アクション」、「デザイナー」の順に選択します。

3. 「フォームの詳細」パネルの「オン・デマンド・ルール」で、次のいずれかの方法を使用します:
 - ルール名がわかっている場合、カンマ区切りリストのテキスト・ボックスに、名前を入力します。
 - 使用可能なルールを検索するには、テキスト・ボックスの横にある「編集」ボタンをクリックし、「使用可能なルール」ポップアップ・ダイアログ・ボックスから、フォームで使用する1つ以上のルールを選択し、「OK」をクリックします。
4. 「OK」をクリックします。
5. データ・フォームを保存します。

フォームの行および列オプションの指定

行および列に指定するディメンション要素は、バックグラウンドPOVまたは選択可能POVに設定される要素を上書きします。行または列ごとに1つのメンバー・リストのみを指定できます。

各オプションの有効な値については、[155ページのデータ・フォーム・スクリプトの使用](#)を参照してください。

フォームの行が100行を超える場合は、「スクリプト」タブを使用して、追加の行を指定できます。使用する構文については、[179ページのRn](#)を参照してください。

フォームの列が24列を超える場合は、「スクリプト」タブを使用して、追加の列を指定できます。使用する構文については、[162ページのCn](#)を参照してください。

また、行および列に計算された行または列および計算式を指定できます。「計算された行」オプションを選択すると、選択した行または列が計算された行および列になるため、計算式を入力する必要があります。



注:

「その他」を使用して、BlankやReadOnlyなどの「オプション」セクションに表示されない行または列のキーワードの構文を指定できます。さらに、スクリプト内の行または列の定義に廃止された構文または無効な構文がある場合は、その構文が「その他」に表示されます。

フォームのヘッダーの指定

「ヘッダー」セクションでは、ラベルの表示、説明または両方などのディメンションごとのヘッダー・プロパティ、スタイル、メンバーおよびメンバー・リストのラベルまたは説明の最大長、長さが固定かどうかを指定できます。ディメンションごとに別の長さを指定できます。

「ヘッダー」セクションにも「その他」ボックスがあり、次のように使用できます。

- 構文が [170ページの手順](#) キーワードに追加されると、その構文を「その他」で指定できます。
- ディメンションのHeaderOptionキーワードに無効な構文がフォームに含まれている場合は、「その他」ボックスにその構文が表示されます。

各オプションの有効な値については、[155ページのデータ・フォーム・スクリプトの使用](#)を参照してください。

データ・フォーム・スクリプトの使用

スクリプト機能を使用して、スクリプトを表示、変更および印刷します。「スクリプト」ページのテキスト・ボックスに構文を入力します。

データ・フォームのスクリプトを作成する場合は、3つのタイプの構文要素(キーワード、値およびオプション)を使用できます。キーワードは、スクリプト内の個々の行にあり、等号の左側です。値は等号の直後に置かれ、これで行が完成します。オプションは、スクリプトの行に追加できます。各オプションは、カンマで区切ります。



注:

値が必須の場合は、オプションの前に指定する必要があります。オプションは必須ではなく、任意の順序で指定できます。

データ・フォームのサンプル・スクリプトは、Financial Managementのサンプル・アプリケーションとともにインストールされます。これらのファイルは、Financial Managementをインストールしたディレクトリの「Sample Applications」フォルダに配置されます。



注:

データ入力フォームのスクリプトの要素では、大文字と小文字が区別されません。

表24 データ・フォームのスクリプトの構文

スクリプトの構文	説明
158ページのAddMember	行の定義内で使用すると、ユーザーは、以前にデータがなかったか、ゼロのデータのみであったために、フォームで非表示にされていたメンバーのデータを追加できます。オプションを使用すると、フォームにアイコンが追加されます。このアイコンをクリックして、ユーザーはメンバーをフォームに追加できます。
160ページのBackgroundPOV	フォームのバックグラウンド・ディメンションのメンバーを指定します。

スクリプトの構文	説明
161ページのBlank	フォームに空白の行、列またはセルを挿入します。
162ページのCn	フォームの各列を定義します。
163ページのCalcByRow	セルに交差する列の計算式がある場合に、行の計算式を使用するかどうかを指定します。
165ページのCell_Link	別のデータ入力フォームとリンクするため、行定義内で 173ページのリンク とともに使用します。
165ページのCellText	行または列でセル・テキスト入力を可能にするかどうかを指定するために使用します。
166ページのCustomHeader	<p>メンバー・ラベルまたは説明のかわりに表示するカスタム・ヘッダー・テキストを指定するために使用します。行または列の定義に使用します。</p> <p>注:</p> <p>次のキーワードはカスタム・ヘッダーには使用できません:</p> <ul style="list-style-type: none"> • <pre> • <textarea> • <script> • <javascript> • <jscript> • <vbs> • <vbscript> • <XonX=X>などの文字列(Xは任意の文字列です)
167ページのCustomHeaderStyle	行または列のヘッダーにカスタム・スタイル属性を割り当てます。
168ページのDynamicPOV	廃止されています。使用しないでください。
168ページのFormInputBoxLength	フォーム上の入力ボックスの幅を指定します。
168ページのFormNumDecimals	フォームの小数点以下の桁数を指定します。このキーワードは、セル通貨の小数点設定を上書きしません。 175ページのNumDecimals を使用すると、行、列またはセルのこの設定が上書きされます。
169ページのFormRowHeight	フォーム内のすべての行の高さを指定します。
169ページのFormScale	フォームの位取りを指定します。
170ページのHeaderOption	フォームでディメンション・ヘッダーをどのように表示するかを指定します。ラベルまたは説明、あるいはその両方を表示し、スタイル属性、最大幅または固定幅を設定します。
172ページの指示	ユーザーへの指示をHTMLフォーマットのテキストおよびリンクで作成します。

スクリプトの構文	説明
172ページのLineItemDetailSinglePeriod	選択したセルについてのみライン・アイテムの詳細を表示するか、すべての期間の詳細を表示するかを指定します。
173ページのリンク	別のデータ入力フォームとリンクするため、 165ページのCell_Link とともに使用します。
173ページのMaxCells	データ・フォームのセルの最大数を指定する場合に使用します。
174ページのMaxColsForSparseRetrievalMethod	疎なフォームのパフォーマンスを最適化します。11列以上含むフォームで使用します。
174ページのNoSuppress	1つ以上の行または列の非表示を無効にします。この設定は、フォーム内の次の他の非表示の設定を上書きします： 192ページのSuppressInvalidRows 、 193ページのSuppressNoData
175ページのNumDecimals	行、列またはセルの小数点以下の桁数を指定します。このキーワードは、セル通貨の小数点設定と、フォームの小数点設定 168ページのFormNumDecimals を上書きします。
175ページのOnDemandRules	データ・フォームで使用可能なオンデマンド・ルールを指定するために使用します。
175ページの上書き	1つ以上の行または列に別のPOVまたは計算式を指定するか、スタイル属性の追加または位取りを設定します。行または列の定義に使用します。
171ページのHideInPov	POVでディメンションを非表示にするかどうかを指定するために使用します。
177ページのPOVOrder	POVでのディメンション名の順序を指定するために使用します。
178ページのPrintNumDataColsPerPage	各ページに印刷する列数を指定します。
178ページのPrintNumRowsPerPage	各ページに印刷する行数を指定します。
179ページのPrintRepeatHeadersonAllPages	各ページにヘッダーを印刷します。
179ページのRn	フォームの各行を定義します。
181ページのReadOnly	読取り専用の行、列またはセルを指定します。
181ページのReportDescription	フォームの説明を指定します。 説明にアンパサンド(&)を含めることはできません。
182ページのReportLabel	フォームのラベルを指定します。次の文字は、データ・フォーム・ラベルではサポートされていません： アンパサンド(&)、アスタリスク(*)、円記号(バックスラッシュ)(¥)、コロン(:)、カンマ(,)、中カッコ({ })、二重引用符(")、スラッシュ(/)、小なり記号と大なり記号(< >)、シャープ(#)、丸カッコ()、ピリオド(.)、パイプ()、プラス記号(+)、疑問符(?)、セミコロン(;)、および末尾のアンダースコア(_)

スクリプトの構文	説明
183ページのReportSecurityClass	フォームのセキュリティ・クラスを指定します。
183ページのReportType	フォーム・タイプを設定します。値にはWebFormを設定する必要があります。
183ページのRowHeaderPct	フォームの合計幅に合わせて、行ヘッダーの幅のサイズを変更するために使用します。
184ページのSCalc	行、列またはセルのサーバー側の計算式を指定します。
185ページのScale	行、列またはセルの位取りを指定します。有効な値は-12から12です。この設定は、フォームのスケール設定を上書きします。169ページのFormScaleを参照してください。
186ページのSelectablePOVList	フォーム用に選択できるディメンションのメンバーを指定します。
187ページのShowDescriptions	ディメンションのメンバーの説明を表示します。
188ページのShowLabels	ディメンションのメンバーのラベルを表示します。
188ページの文字列	列、行またはセルにテキスト文字列を追加します。
188ページのStyle	行、列、セルまたはディメンション・ヘッダーのスタイル属性を指定します。
191ページのSuppressColHeaderRepeats	繰り返される列ヘッダーを非表示にします。
192ページのSuppressInvalidCols	無効なセルの入った列を非表示にします。
192ページのSuppressInvalidRows	無効なセルの入った行を非表示にします。
193ページのSuppressNoDataCols	データのない列を非表示にします。
193ページのSuppressNoDataRows	データのない行を非表示にします。
194ページのSuppressRowHeaderRepeats	繰り返される行ヘッダーを非表示にします。
194ページのSuppressZeroCols	ゼロの入った列を非表示にします。
194ページのSuppressZeroRows	ゼロの入った行を非表示にします。

AddMember

このオプションを行の定義に使用すると、アイコンがフォームに追加されます。このアイコンをクリックして、ユーザーは、フォームに追加するメンバーを選択できます。新規メンバーをフォームに追加したら、ユーザーはこれらのメンバーのデータを入力できます。

AddMemberオプションは、勘定科目、ICPおよびカスタム・ディメンションで使用できます。



注:

NoDataセルは、193ページのSuppressNoDataRowsキーワードを使用するか、ユーザーがフォームで該当するチェック・ボックスを選択することで非表示にできます。

構文

```
AddMember :  
MemberList
```

MemberListには、ユーザーがフォームにメンバーを追加できるリスト名を指定します。



注意

参照できるメンバー・リストは、AddMember行ごとに1つのみです。

このオプションを使用するには、次の2つの行定義が必要です。

- メンバー・リストの合計を示し、AddMemberオプションを使用した要約行



注:

要約行は非表示にできません。

- 要約行に指定されたPOVと同じものを含むリスト行。ただし、リスト行は、メンバーを追加中のディメンションのリストを含みます。リストは少なくとも、要約行リストのメンバーを含む必要があります。



注:

リスト行は、要約行の前後どちらにでも入力できます。

例

```
R1=A#SalesInterco.I{[Base]}
```

```
R2=A#SalesInterco.I#[ICP Entities],  
AddMember:I{[Base]}
```

この例では、Row 1に内部取引パートナー・ディメンションの[Base]メンバー・リストを指定します。Row 2には、メンバー・リストの合計とAddMemberのアイコンで使用するメンバー・リストを指定します。



注:

フォーム内の複数の行でAddMemberオプションを使用できますが、要約行ごとに対応するリスト行を入力する必要があります。

BackgroundPOV

このキーワードは、フォームのバックグラウンド・ディメンションのメンバーを指定します。バックグラウンドPOVに指定されていないディメンションは、動的とみなされ、フォームを開いたときにユーザーのPOVから取得されます。

階層内のすべてのメンバーを操作できるように、ユーザーには、メンバー階層への完全なセキュリティ・アクセス権限が必要です。たとえば、ユーザーがカスタム4のすべてのメンバーにアクセスする場合、このケースでは、親エンティティCustomTopにアクセスできるようにする必要があります。



注:

バックグラウンドPOVのメンバーが、そのディメンションの [186ページの手順](#) に準じていない場合は、リストの最初のメンバーがデフォルトで使用されます。

構文

```
BackgroundPOV=S#
Scenario
.W#
View
.E#
Parent
.
Entity
.V#
Value
.A#
Account
.I#
ICP
.
C1#
Custom1
.C2#
Custom2
.C3#
Custom3
.C4#
Custom4
```

表25 BackgroundPOVキーワードの構文

パラメータ	説明
Scenario	有効なシナリオ・メンバーの名前。
View	有効なビュー。

パラメータ	説明
Parent	有効な親メンバーの名前。このパラメータはオプションです。
Entity	有効なエンティティ・メンバーの名前。
Value	有効な値メンバーの名前。
Account	有効な勘定科目メンバーの名前。
ICP	有効なICPメンバーの名前。
Custom	有効なカスタム・メンバーの名前。



注:

Background POV内のディメンションのすべてのメンバーを指定する必要はありません。メンバーを指定していないディメンションの場合、ユーザーの視点からのディメンション・メンバーが使用されます。

例

```
BackgroundPOV=S#Actual.Y#2014.P#January.W#Periodic.V#<Entity
Currency>.A#Sales.I#[ICPNone].C4#[None]
```

Blank

フォーム内の空白の行、列またはセルを指定します。このオプションは、行または列の定義内、またはセルの上書き定義内で使用します。空白の行、列またはセルは空でかつ読取り専用であり、データまたはPOVが含まれていないため、セルのコンテキスト・メニューは使用不可になっています。Blankオプションはサーバー側の計算式とみなされるため、SCalcを使用する任意の箇所で使用できます。

例

Cn

フォーム内の列を定義します。C1、C2、C3などのキーワードは、指定された順に各列の定義を行います。列の識別子は1から順に続きます。

列に指定したディメンション要素は、[160ページのBackgroundPOV](#)または[186ページのSelectablePOVList](#)に設定されている要素を上書きします。列の定義にメンバー・リストやシステム・リストを使用できます。



注:

列ごとに1つのリストしか使用できません。

列の定義内では、次の値およびオプションを使用できます。

- [161ページのBlank](#)
- [163ページのCalc1](#)
- [165ページのCellText](#)
- [166ページのCustomHeader](#)
- [175ページのNumDecimals](#)
- [175ページの上書き](#)
- [181ページのReadOnly](#)
- [184ページのSCalc](#)
- [185ページのScale](#)
- [188ページの文字列](#)
- [188ページのStyle](#)

構文

```
C
n
=
CalcExpression
```

```
C
n
=
POVExpression
```

表26 列のキーワードの構文

パラメータ	説明
n	列番号
CalcExpression	S <code>Calc</code> 、BlankまたはStringを使用します。184ページのS <code>Calc</code> 、161ページのBlankおよび188ページの文字列を参照してください。
POVExpression	ディメンションの有効な交差またはメンバー・リスト

例

```
C1=S#Actual.P#July,CustomHeader:Actual_July
```

```
C2=S#Actual.P#August,(Override 2,3, P#July)
```

```
C3=S#Budget.P#September
```

```
C4=SCalc(col(1)+col(3)),numdecimals:4,scale:1,readonly
```

```
C5=Blank
```

```
C6=S#Budget.P#October,Style:font-style:bold
```

```
C7=C1{TotalProducts.[Hierarchy]}
```

Calc1

この値は非推奨です。このキーワードを使用するフォームがある場合、手動で編集して、184ページのS`Calc`を使用する必要があります。

CalcByRow

デフォルトの計算順序(列が先)を、行が先になるよう変更するために使用されるキーワード。このキーワードはフォーム全体に適用されます。

構文

```
CalcByRow=  
Boolean
```

Booleanは、行の計算式を使用する場合はTRUE、列の計算式を使用する場合はFALSEです。

例

```
ReportType=WebForm
```

```
ReportLabel=CalcByCol
```

```
ReportDescription=Demonstrate CalcByRow
```

```
BackgroundPOV=S#Actual.Y#2014.P#January.w#<Scenario  
View>.E#UnitedStates.Connecticut.V#USD.A#Sales.I#[ICP  
NONE].C1#Golfballs.C2#Customer2.C3#[None].C4#[None]
```

```
C1=S#Actual
```

```
C2=S#Budget
```

```
C3=S#Calc(Col(2)-Col(1)),CustomHeader:Variance
```

```
C4=S#Calc(Col(3)/Col(1)),CustomHeader:Variance %
```

```
R1=A#Sales
```

```
R2=A#Purchases
```

```
R3=A#OtherCosts
```

```
R4=S#Calc(Row(1)+Row(2)+Row(3)),CustomHeader:Total
```

```
ShowLabels=True
```

```
CalcByRow=False
```

Cell_Link

別のデータ・フォームへのリンクを指定します。リンクは行の定義で参照されます。[173ページのリンク](#)とともに使用します。64個までのリンク(Link1-Link64)を指定できます。リンクは番号順にする必要はありません。



注:

リンクするフォーム名では、大文字と小文字が区別されます。

構文

```
Cell_Link  
x
```

```
Link  
x  
=  
FormName
```

xには、リンクの番号を指定します。FormNameには、リンク先のフォーム名を指定します。

例

```
R1=A#Salaries, Cell_Link1
```

```
Link1=Dynamic
```

CellText

行、列またはセルのセル・テキストを表示します。入力されたセル・テキストの最初の69文字をタイトルの入力とみなします。



注:

セル・テキストが含まれているデータ・フォームを抽出すると、セル・テキストのエントリのタイトル(最初の69文字)のみが抽出されます。

構文

```
CellText:<cell text label>
```

例

```
R1=A#Salaries, CellText:[Default]
```

```
R2=A#Salaries, CellText:Validation
```

CustomHeader

列または行のカスタム・ヘッダーを指定します。このオプションは、計算された列または行があり、特定の数式を非表示にして、Varianceなどの説明で置き換える場合に便利です。行または列でディメンションがネストしている場合、カスタム・ヘッダーはそのディメンションのみではなく、その行または列のヘッダー全体に適用されます。

セミコロンを区切り記号に使用して、後続するセルのカスタム・ヘッダーを指定できます。たとえば、この構文を使用すると、行内の3つのディメンション・ヘッダーにシナリオ、年および月のカスタム・ヘッダーを指定できます。

```
R2=S#Actual.Y#2014.P#January, CustomHeader:Scenario;Year;Month
```

一部のヘッダー・セルのみを置き換えるには、ピリオド(.)を使用して、元のヘッダーを表示する必要があることを示します。また、ピリオドを省略することによってヘッダーを非表示にすることもできます。たとえば、次の構文は、最初のディメンションの元のヘッダーを表示し、2番目のディメンションのヘッダーを非表示にし、3番目のディメンションの元のヘッダーを表示します。

```
R2=S#Actual.Y#2014.P#January, CustomHeader:.;;.;
```



注意

計算される行または列にCustomHeaderオプションを使用する場合は、ヘッダーの最初のセルのみにテキストを設定できます。

構文

```
CustomHeader :  
HeaderName
```

HeaderNameには、使用するヘッダーを指定します。



注:

カスタム・ヘッダーでは、カンマ(,)、コロン(:)、アンパサンド(&)は使用できません。

例

この例では、列3のカスタム・ヘッダーをVarianceにします。

```
C3=SCalc(C1-C2), customheader:Variance
```

CustomHeaderStyle

このオプションは、ある行または列のヘッダーにカスタム属性を割り当てます。このオプションは、すべてのヘッダーにわたる単一ディメンションではなく、ある行または列の全体に対応するヘッダーに適用される点でキーワードHeaderOptionのStyleオプションとは異なります。競合が存在する場合、CustomHeaderStyleオプションは、キーワードHeaderOptionのStyleオプションより優先的に使用されます。CustomHeaderStyleを使用するかわりに2つのスタイルを混在させるには、以下の例に示すように、CustomHeaderStyleオプションの前にセミコロンを挿入します。[188ページのStyle](#)を参照してください。

構文

このオプションは、W3Cがサポートする標準を使用します。

```
CustomHeaderStyle :  
Property  
:  
Value
```

例

```
C1=S#Actual, CustomHeaderStyle:font-style:italic;font-family:arial;font-size:12px;font-color:red
```

DynamicPOV

このキーワードは廃止されています。

DynamicPOVキーワードを使用したフォームがある場合は、次の手順に従ってこのキーワードを削除してください:

1. BackgroundPOVキーワードから、DynamicPOVキーワードで指定されたすべてのディメンションを除去します。
2. DynamicPOVキーワードを削除します。

FormInputBoxLength

入力ボックスの幅を指定します。デフォルトは20文字の幅です。



注:

このキーワードによって、入力ボックスに入力できる文字数が決まるわけではありません。

構文

```
FormInputBoxLength=  
InputLength
```

InputLengthには、入力ボックスの幅を文字数で指定します。

例

```
FormInputBoxLength=20
```

FormNumDecimals

フォーム全体の小数点以下の桁数を指定します。このキーワードを指定すると、セルに設定された小数点以下の桁数が上書きされます。このキーワードを指定しない場合は、セルの小数点以下の桁数が使用されます。

[175ページ](#)のNumDecimalsを使用して、列、行またはセルについて小数点以下の桁数の設定を上書きできます。

構文

```
FormNumDecimals=  
Decimals
```


Decimalsには、0-9の値を指定します。



注:

9を使用すると精度が最高になり、実際には小数点以下14桁まで表示されます。

例

```
FormNumDecimals=0
```

FormRowHeight

このキーワードは、フォーム内のすべての行の高さを指定します。デフォルトは16pxです。

構文

```
FormRowHeight=  
Pixels  
px
```

Pixelsには、行の高さをピクセル数で指定します。



注意

Pixelsの後にpxを指定する必要があります。pxを省略すると、フォームの印刷時に行または列が正しくは配置されない場合があります。

例

```
FormRowHeight=16 px
```

FormScale

このキーワードは、フォームのデフォルトの位取りを指定します。

フォームにこのキーワードを指定すると、このキーワードによってエンティティの通貨の位取りが上書きされます。指定しない場合は、エンティティに割り当てられた通貨に定義されている位取りが使用されます。

行または列の定義、またはセルの上書きで [185ページのScale](#) オプションを使用することにより、フォームの位取り設定を上書きできます。

構文

```
FormScale=  
n
```

nには、-12から12の値を指定します。

例

```
FormScale=0
```

HeaderOption

このキーワードは、各ディメンションのヘッダーのデフォルトを指定するために使用します。現在サポートしているオプションは次のとおりです:

- Length - 行ヘッダーの最大長。最大長と最小長を同一にする場合は、数値を使用しても、単語「Fixed」を使用してもかまいません。



注:

行ヘッダーが指定された長さより長い場合は、省略記号に置換されます。

- ShowDescription - メンバーの説明を表示
- ShowLabel - メンバー・ラベルを表示
- Style

構文

```
HeaderOption  
Dimension  
=Length:  
n
```

```
HeaderOption  
Dimension  
=ShowDescription
```

```
HeaderOption  
Dimension  
=ShowLabel
```

```
HeaderOption
Dimension
=Style:
Property
:
Value
;
Property
:
Value
...
```

Dimensionには行ディメンションの名前、*n*にはヘッダーの長さまたは「Fixed」、Propertyにはスタイルのプロパティ、Valueにはプロパティの値をそれぞれ指定します。[188ページのStyle](#)を参照してください。

例

```
HeaderOptionPeriod=Length:4
```

```
HeaderOptionScenario=ShowDescription
```

```
HeaderOptionsAccount=ShowLabel,Style:font-style:italic
```

HideInPov

このキーワードは、各ディメンションのHideInPOVオプションを指定するために使用します。この値が存在する場合、ディメンションはデータ・フォームのPOVバーに表示されません。この値が見つからない場合、ディメンションはPOVバーに表示されます。

構文

```
HideinPOV=
Dimension
```

ディメンション名は、ショート名またはロング名で表示されており、カンマ区切りのリストです。



注:

選択可能なすべてのディメンションがPOVバーに表示されるため、ディメンションが選択可能なPOVの一部である場合、このディメンションは無視されます。

例

次の例では、データ・フォームが開いている場合、シナリオ、年、期間がPOVバーで非表示です。

```
HideinPov=S,Y,P
```

指示

フォームのユーザーに対する指示を指定します。このキーワードを指定しない場合は、指示ウィンドウが開いて、メッセージ「このフォームに定義された詳細な指示はありません。」が表示されます。

構文

```
Instructions=HTMLInstructions
```

HTMLInstructionsには、HTMLフォーマットのテキストおよびリンクを指定します。

例

```
Instructions=Please enter your cost center budgets for the year.Any questions,  
please contact the Budget Administrator.
```

LineItemDetailSinglePeriod

ライン・アイテムの詳細を、選択したセルのみに対して表示するか、すべての入力期間に対して表示するかを指定します。デフォルトはTRUEです。これは、選択したセルのみに対して詳細を表示します。

構文

```
LineItemDetailSinglePeriod=  
Boolean
```

Booleanには、TRUE(選択したセルのライン・アイテムの詳細のみを表示する場合)、またはFALSE(すべての入力期間のライン・アイテムの詳細を表示する場合)を指定します。

例

```
LineItemDetailSinglePeriod=True
```

リンク

別のデータ・フォームへのリンクを指定します。リンクは行の定義で参照されます。[165ページ](#)のCell_Linkとともに使用します。64個までのリンク(Link1-Link64)を指定できます。リンクは番号順にする必要はありません。



注:

リンクするフォーム名では、大文字と小文字が区別されます。

構文

```
Cell_Link  
x
```

```
Link  
x  
=  
FormName
```

xには、リンクの番号を指定します。FormNameには、リンク先のフォーム名を指定します。

例

```
R1=A#Salaries, Cell_Link1
```

```
Link1=Dynamic
```

MaxCells

データ・フォームで許されるセルの最大数を指定します。フォームがMaxCells値を超えると、エラーが発生します。デフォルトは25000です。

構文

```
MaxCells=  
n
```

nには、フォームのセルの数を指定します。



注:

値に含まれるのは、計算済セルなどのフォーム上の非表示でないセルの数です。非表示のセルは含まれません。

例

```
MaxCells=500
```

MaxColsForSparseRetrievalMethod



注:

このキーワードは非推奨です。このキーワードを使用するスクリプトがある場合、フォームは正常にロードを行いますが、値を編集することはできません。

フォーム内の列の数を指定して、疎なデータ・フォームのパフォーマンスを最適化します。11列以上含むフォームに指定します。10列以下のフォームの場合は、自動的に最適化が行われます。

構文

```
MaxColsForSparseRetrievalMethod=  
n
```

*n*には、フォームの列数を指定します。

例

```
MaxColsForSparseRetrievalMethod=11
```

NoSuppress

1つ以上の行または列の非表示を無効にします。これにより、フォームの非表示オプションに関係なく、行または列が表示されます。このオプションは、行または列の定義内で使用します。

例

```
R4=A#Inventory, NoSuppress
```

NumDecimals

計算済または未計算の行または列に表示する小数点以下の桁数、またはセルの上書きで表示する小数点以下の桁数を指定します。このキーワードを指定すると、セルに設定された小数点以下の桁数または [168ページのFormNumDecimals](#) に設定された小数点以下の桁数が上書きされます。

構文

```
NumDecimals:  
n
```

nには、0-9の値を指定します。



注:

9を使用すると精度が最高になり、実際には小数点以下14桁まで表示されます。

例

```
C4=A#Inventory, NumDecimals:1
```

OnDemandRules

このオプションは、データ・フォームで使用可能なオンデマンド・ルールを指定するために使用します。ルール名のカンマ区切りリストを使用します。 [153ページのデータ・フォームのオンデマンド・ルールの指定](#) を参照してください。

例

```
OnDemandRules=Calculation, Tax, Tax2
```

上書き

このオプションは、POVの様々なディメンション・メンバー、式の計算式または1つ以上連続する列または行のテキストを指定します。また、スタイルの変更にも使用します。このオプションは、行または列の定義内で使用します。



注:

連続しないセルを上書きするには、フォーム・オプションの「その他」フィールドに上書きを入力します。複数の上書きを入力するには、各上書きをカンマで区切ります。次の例は、個々の3つのセルを上書きします。

```
Override(1,1,string("455")),Override(3,3,string("23")),
Override(5,5,string("2234"))
```

Overrideオプションでは、メンバー・リストを使用できません。競合が存在する場合は最後の値を使用することで、複数の上書きを1つのセルに適用できます。ほとんどの場合、1つのセルに複数の上書きを混在させることもできます。たとえば、ある行の上書きがNumDecimalsの値を指定し、ある列の上書きがScaleの値を指定する場合、同じディメンションが使用されている場合を除き、競合は存在しません。先行するセミコロンを使用すると、Styleの値を混在させることができます。

上書きが交差する場合は、Overrideキーワードの後にセミコロンを含むことで上書きを混在させることもできます。セミコロンがない場合、フォーム内で上書きが競合すると、行の値は列の値よりも優先するため、行に対して定義されたスタイルが使用されます。行と列の定義の上書きを混在させるには、行のキーワードにセミコロンを含む必要があります。

構文

```
Override(
  StartCell
  ,
  EndCell
  ,
  Override
)
```

パラメータ	説明
StartCell	上書きの開始位置を表す整数値。行の上書きが定義されている場合、このパラメータは、上書きが適用される開始列を示します。この例では、上書きは列2から開始します。 <pre>R2=A#Sales, Override(2,3,A#SalesTP)</pre>
EndCell	上書きの終了位置を表す整数値。前述の例では、上書きは列3で終了します。
Override	POV。たとえば、別の勘定科目で上書きするには、A#newacctを指定します。別のシナリオと勘定科目で上書きするには、A#newacct.S#newsenarioを指定します。次の値またはオプションをPOVの上書きとともに使用したり、単独で使用することもできます。

パラメータ	説明
	<ul style="list-style-type: none"> • 161ページのBlank • 165ページのCellText • 175ページのNumDecimals • 181ページのReadOnly • 184ページのSCalc • 185ページのScale • 188ページの文字列 • 188ページのStyle

例

次の例では、行2の列2および列3が、期間にJanuary、月の売上金額のかわりにPriorSalesの金額を使用して上書きされます:

```
R2=A#Sales, Override(2,3,Y#2014.P#January.A#PriorSales)
```

次の例では、行2の列2および列3が、Sales1、Sales2、Sales3の平均を計算する式で上書きされます。

```
R2=A#Sales, Override(2,3,SCalc((A#Sales1+A#Sales2+A#Sales3)/3))
```

次の例では、行2の列2および列3が、読取り専用オプションで上書きされます。

```
R2=A#Sales, Override(2,3,readonly)
```

POVOrder

このキーワードは、POVでのディメンション名の順序を指定するために使用します。ディメンション名は、ショート名またはロング名のいずれかを指定することができ、表示する順序で、カンマ区切りリストに示されます。リストに指定されないディメンションは、ディメンション番号順に、リストの末尾に追加されます。

例

次の例では、POVバーでのディメンションの順序は、勘定科目、期間、年、シナリオです。



注:

勘定科目ディメンションが非表示として指定された場合、期間がPOVバーに最初に表示されません。

POVOrder=A,P,Y,S

PrintNumDataColsPerPage

このキーワードは、各ページに印刷する列数を指定します。デフォルトは6です。

ユーザーは、フォームの印刷用の表示画面でこの設定を上書きできます。



注:

このキーワードに値を設定するときは、行の高さなどのフォーム固有の条件、使用するプリンタ、および解像度や用紙の向きなどのプリンタ設定も考慮する必要があります。

構文

```
PrintNumDataColsPerPage=  
n
```

nには、1ページに印刷する列数を指定します。

例

```
PrintNumColsPerPage=6
```

PrintNumRowsPerPage

このキーワードは、各ページに印刷する行数を指定します。デフォルトは20です。

ユーザーは、フォームの印刷用の表示画面でこの設定を上書きできます。



注:

このキーワードに値を設定するときは、行の高さなどのフォーム固有の条件、使用するプリンタ、および解像度や用紙の向きなどのプリンタ設定も考慮する必要があります。

構文

```
PrintNumRowsPerPage=
```

n

*n*には、1ページに印刷する行数を指定します。

例

```
PrintNumRowsPerPage=20
```

PrintRepeatHeaderonAllPages

このキーワードは、ヘッダーをすべてのページに印刷するかどうかを指定します。デフォルトはTRUEです。

ユーザーは、フォームの印刷用の表示画面でこの設定を上書きできます。

構文

```
PrintRepeatHeaderonAllPages=  
Boolean
```

*Boolean*には、TRUEまたはFALSEを指定します。

例

```
PrintRepeatHeaderonAllPages=True
```

Rn

このキーワードは、フォーム内の行を定義します。R1、R2、R3などのキーワードは、指定された順に各行の定義を行います。行の識別子は1から順に続きます。メンバー・リストとシステム・リストを行の定義に使用できます。



注:

1つの行で複数のリストを使用できます。

行に指定したディメンション要素は、[160ページのBackgroundPOV](#)または [186ページのSelectablePOVList](#)に設定されている要素を上書きします。

行の定義内では、次の値およびオプションを使用できます。

- [158ページのAddMember](#)
- [161ページのBlank](#)
- [165ページのCell_Link](#)
- [165ページのCellText](#)
- [166ページのCustomHeader](#)
- [174ページのNoSuppress](#)
- [175ページのNumDecimals](#)
- [175ページの上書き](#)
- [181ページのReadOnly](#)
- [184ページのSCalc](#)
- [185ページのScale](#)
- [188ページの文字列](#)
- [188ページのStyle](#)

構文

```
R
n
=
CalcExpression
```

```
R
n
=
POVExpression
```

表27 行のキーワードの構文

パラメータ	説明
n	行番号。
CalcExpression	SCalc、BlankまたはStringを使用します。 184ページのSCalc 、 161ページのBlank および 188ページの文字列 を参照してください。
POVExpression	ディメンションの有効な交差またはメンバー・リスト

例

```
R1=A#Sales.I#[ICP Entities],AddMember:I{[Base]}
```

```
R2=A#Sales.I{[Base]}
```

```
R3=A#HeadCount.I#[ICP None],NoSuppress
```

```
R4=A#Purchases.I#[ICP None],CustomHeader:ABC
```

```
R5=SCalc(Row(2)*100),numdecimals:1,scale:0
```

```
R6=A{OperatingIncome.[Descendants]}
```

ReadOnly

フォーム内の読取り専用の行、列またはセルを指定します。このオプションは、行または列の定義内、またはセルの上書き定義内で使用します。読取り専用のセルは他のすべてのセルと同様に見えますが、内容を編集できません。読取り専用の行、列またはセルのスタイルを変更して、編集可能な行、列およびセルと区別することは可能です。[188ページのStyle](#)を参照してください。



注:

Excelにエクスポートする場合は、読取り専用のフォーマットが維持されます。

例

```
C4=S#Actual.Y#2014,ReadOnly
```

ReportDescription

このキーワードは、フォームの説明を指定します。

構文

```
ReportDescription=  
Description
```

Descriptionには、フォームの説明を指定します。説明には、255文字まで入力できます。

例

```
ReportDescription=Intercompany Detail
```

ReportLabel

このキーワードは、フォームの名前を指定します。このキーワードは必須です。

構文

```
ReportLabel=  
Label
```

Labelには、フォームの名前を指定します。名前には、40文字まで入力できます。次の文字は名前には使用できません:

- アスタリスク(*)
- アット記号(@)
- 円記号(バックスラッシュ)(¥)
- コロン(:)
- カンマ(,)
- 中カッコ({})
- スラッシュ(/)
- 小なり記号と大なり記号(< >)
- シャープ(#)
- カッコ()
- ピリオド(.)
- パイプ(|)
- プラス記号(+)
- 疑問符(?)
- 引用符(")
- セミコロン(;))



注:

レポート・ラベルにアンダースコア(_)を使用できます。ただし、2つの文字の間で使用する必要があります。ラベル名として単独で使用したり、ラベル名の末尾に使用したりできません。

例

```
ReportLabel=ICP Detail
```

ReportSecurityClass

このキーワードは、フォームに割り当てるセキュリティ・クラスを指定します。デフォルトは[Default]です。

構文

```
ReportSecurityClass=  
SecurityClass
```

SecurityClassには、有効なセキュリティ・クラスの名前を指定します。

例

```
ReportSecurityClass=Corporate
```

ReportType

このキーワードは、レポート・タイプを指定します。このキーワードは必須です。ファイルをデータ・フォーム・スクリプトとしてロードするには、値にWebFormを設定する必要があります。

構文

```
ReportType=WebForm
```

RowHeaderPct

このオプションは、フォームの合計幅に合わせて、行ヘッダーの幅のサイズを変更するために使用します。行ヘッダーの実際の幅が指定された比率を超えている場合にのみ適用されます。

たとえば、実際の行ヘッダーの幅がフォームの合計幅の25%で、指定された値が40%の場合、コンテンツが25%で表示されるため、行ヘッダーは40%にはなりません。ただし、指定された値が10%の場合、行ヘッダーの幅は実際の

25%の幅から、指定された10%の幅にスクロール・バー付きで収まるように小さくなります。行ヘッダーに複数の列がある場合は、すべての列の幅が加算され、フォームの合計幅と比較されます。

構文

```
RowHeaderPct:  
n
```

nには、1-100の値を指定します。

例

```
RowHeaderPct=30
```

SCalc

SCalc関数を使用して、列および行に、標準の算術演算子を使用するカスタム式を作成できます。この値は、行または列の定義内、またはセルの上書き定義内で使用します。指定された計算式がアプリケーション・サーバーで実行されます。次の例では、列1から列2を減算します：

```
SCalc(C1-C2)
```

この値を使用して、フォーム内のテキストを作成することもできます。

構文

```
SCalc(  
<expression> [<operator> <expression>]  
)
```



注：

SCalcには、[<operator> <expression>]の複数のペアを指定できます。

パラメータ	説明
operator	計算式の算術演算子。次の演算子をサポートします。 + - * /
expression	計算式の値。数値以外に、セルの参照、行の参照、列の参照および様々なタイプのアイテムを指定できます。

パラメータ	説明

SCalcの計算式では、次のタイプのアイテムを使用できます。

- デイメンション・メンバーの参照。勘定科目デイメンション・メンバーのPurchasesおよびOtherCostsを参照します。 $R6=SCalc(((A\#Purchases)-(A\#OtherCosts))*100)$
- 構文Cell(rowIndex, columnIndex)を使用したセル参照。次の例は、フォーム内の4行2列目のセルを参照します。 $R1=SCalc(Cell(4,2))$
- 構文Row(rowIndex)を使用した行の参照。次の例は、4行目を2行目で除算します。 $R3=SCalc(Row(4)/Row(2))$



注:

行または列にメンバー・リストが含まれる場合は、リストのメンバーの合計を使用して計算が行われます。

- 構文Col(columnIndex)を使用した列の参照。次の例は、1列目と3列目を加算します。 $C4=SCalc(Col(1)+Col(3))$
- ネストされた計算式。括弧を使用してネストします。
- SCalc計算を使用したセルへの前方参照。



注:

SCalc計算は、データが保存されるまで行われません。計算結果は、フォームをリフレッシュするまでフォームには表示されません。

SCalc行または列は、計算内で別のSCalc行または列を参照できますが、SCalc行または列で別のScalc行または列を前方参照することはできません。たとえば、次のSCalcの前方参照は許されます。

```
C1=A#Sales
C2=A#COGS
C3=SCalc(Col(1)-Col(2))
C4=SCalc(Col(3)/Col(1)*100)
```

Scale

このオプションは、計算されない列、行またはセルの位取りを指定します。

構文

Scale:

n

*n*には、-12から12の値を指定します。

例

```
C4=A#Inventory, Scale:2
```

SelectablePOVList

このキーワードは、ユーザーが選択できるディメンションのメンバーを指定します。

選択可能なディメンションの初期値は、そのディメンションに初期値が指定されていれば、バックグラウンドPOVから取得します。初期値が指定されていなければ、初期値はユーザーのPOVから取得します。リストの初期値が無効な場合は、リストの最初のメンバーがデフォルトになります。

ユーザーが新しいメンバーを選択すると、選択されたメンバーはユーザーの視点に組み込まれます。

構文

```
SelectablePOVList=  
Dimension  
{  
MemberList  
}
```

```
SelectablePOVList=  
Dimension  
{ [  
SystemList  
]}
```

```
SelectablePOVList=  
Dimension  
{  
Parent  
.[  
SystemList  
]}
```

表28 SelectablePOVListキーワードの構文

パラメータ	説明
	選択可能なディメンションを表す、次のいずれかの文字

パラメータ	説明
Dimension	<ul style="list-style-type: none"> • S(シナリオの場合) • W(ビューの場合) • E(エンティティの場合) • V(値の場合) • P(期間の場合) • A(勘定科目の場合) • I(ICPの場合) • C(カスタムの場合)
Parent	ディメンションの有効な親
MemberList	有効なメンバー・リストの名前
SystemList	有効なシステム・リストの名前

例

```
SelectablePOVList=S{ActualBudget}.Y{Years}.P{Months}.E{[Hierarchy]}.
C1{ProductsAllocate}.C2{Customers}.C3{AllChannels}
```

ShowDescriptions

このキーワードは、フォームのディメンション要素の説明を表示するかどうかを指定します。デフォルトはFALSEです。



注:

説明が存在しない場合は、ラベルが表示されます。エンティティの説明が存在しない場合は、エンティティ・ラベルのみが表示されます; 親ラベルは含まれません。

構文

```
ShowDescriptions=
Boolean
```

Booleanには、TRUEまたはFALSEを指定します。

例

```
ShowDescriptions=True
```

ShowLabels

このキーワードは、フォームのディメンション要素のラベルを表示するかどうかを指定します。デフォルトはTRUEです。

[188ページのShowLabels](#)および [187ページの手順](#)にTRUEが設定されている場合は、ラベルおよび説明がハイフンで区切られます。両方ともにFALSEが設定されている場合は、ラベルが表示されます。

構文

```
ShowLabels=  
Boolean
```

*Boolean*には、TRUEまたはFALSEを指定します。

例

```
ShowLabels=False
```

文字列

このオプションは、列、行またはセルにテキスト文字列を指定します。このオプションは、行または列の定義内、またはセルの上書き定義内で使用します。Stringオプションは、サーバー側の計算式とみなされるため、SCalcを使用する任意の箇所で使用できます。

例

```
C4=String("Show this read-only string")
```

Style

このオプションは、行、列、セル、カスタム・ヘッダーまたはディメンション・ヘッダーのスタイル属性を指定します。このオプションは、行、列、セルの上書き、カスタム・ヘッダーまたはヘッダー・オプションの定義内で使用します。



注意

フォームにスタイル属性を適用する場合は注意が必要です。Styleオプションは強力な機能であるため、間違って使用した場合、フォームの外観が大幅に変更される可能性があります。



注:

Excelにエクスポートする場合は、スタイルのフォーマットが維持されます。

構文

各スタイル・プロパティは、プロパティ名、コロンおよび値で構成されます。次に示すように、値はスペースで区切られた1つ以上の単語です:

```
Style:color:red;text-align:right
```

Styleオプションは、Financial Managementではなく、ブラウザによって解釈されるため、ブラウザのサポート・レベルによってのみ制限されます。Styleオプションは、World Wide Web Consortium(W3C)がサポートする標準を使用します。



注:

Styleオプション・キーワードを1行で1回使用して、複数のプロパティ:値ペアを追加できます。例:

```
C3=S#Actual.Y#@CUR(-1),Style:font-color:blue;background-color:red;font-weight:bold
```

Style Example表では、使用できるプロパティと値を示します。完全なリストは、W3Cの「Property Index」を参照してください。

表29 Styleの例

プロパティ	値
色(フォアグラウンド)背景色	色名または標準の16進RGB表記。例: Style: color: red Style: color: rgb(255,0,0) Style: background-color: #ff0000 Style: background-color: yellow
Font-family	フォント名。例:

プロパティ	値
	Style: font-family: Arial
フォント	フォント・スタイル。例: Style: font-style: italic 注: 1つのStyle値でフォントのプロパティを6つまで組み合わせることができます。例: Style: font-style:italic;font-family:Arial;font-size:12px
Font-weight	フォントの太さ。値は、demi-bold、demi-light、extra-bold、light、extra-light、demi-lightを含みます。例: Style: font-weight: extra-bold
Font-size	フォントのポイント・サイズ。例: Style: font-size: 12px
整列	テキストの配置。値は、left、right、centerおよびjustifyを含みます。例: <div style="text-align: center; background-color: #f0f0f0; padding: 5px;">Style: text-align: center</div>

行に対して、「スタイル」テキスト・ボックスまたは「カスタム・ヘッダーのスタイル」テキスト・ボックスに最初の文字としてセミコロンを入力することで、行および列の交差のスタイルを組み合わせることができます。Styleキーワードの後にセミコロンがない場合、フォーム内でスタイルが競合すると、行の値は列の値よりも優先するため、行に対して定義されたスタイルが使用されます。

グリッド内の各セルは、右と下の枠線のみを表示します。そのため、隣接するセルが1つのピクセルによって区切られます。Styleを使用して、色を変更したり、右また下の枠線を破線に設定したりできます。ただし、上または左の枠線を使用可能にした場合は、上側のセルの下の枠線と左側のセルの右の枠線に加えて、上または左の枠線がそれぞれ存在することになります。これは、2つの枠線を持つことを意味します。2つの枠線を避けるには、隣接する枠線を無効にします。

また、ヘッダーがグリッドに揃うようにするために、軸のCustomHeaderStyleも設定する必要があります。

例

次の例では、行の定義内でStyleキーワードの後にセミコロンを追加することで、Row 1(ピンクの背景色)とColumn 1(ピンクの背景色と太字、緑のテキスト)のスタイルが組み合わされています。191ページの[SuppressColHeaderRepeats](#)の例には、この構文が使用された場合のサンプルのデータ入力フォームが表示されています。

```
R1=Blank, CustomheaderStyle: font-weight: bold, CustomHeader: Balance sheet accounts,
```

```
Style:; Background-color: pink
```

```
C1=S#Actual.Y#@CUR,  
Style: Background-color: pink; color: green;  
font-weight: bold
```

	<i>Actual</i>	<i>Budget</i>
Balance sheet accounts		
Cash		1,746,137.00
Short Term Receivables		9,951.32
Inventories		-10,746.57
Short Term Investments		2,831.47
Total Short Term Assets		1,748,173.21
Computers	33,333.00	13,382.68
Buildings		4,351.48
Office Equipment	50,000.00	-948.88
Transportation	80,000.00	4,719.12
Fixtures	800,000.00	931.97
Tangible Assets	963,333.00	22,436.37
Accumulated Depreciation	22,222.00	-87,002.34

SuppressColHeaderRepeats

このキーワードは、繰り返される列ヘッダーを非表示にするかどうかを指定します。デフォルトはTRUEです。

この例では、繰り返される列ヘッダーが非表示の場合のデータ・フォームを示します。非表示が無効な場合は、列ヘッダーActualもAugustの上に表示されます。

	Actual	
	July	August
Sales	500,000.00	600,000.00
Purchases	250,000.00	300,000.00

構文

```
SuppressColHeaderRepeats=  
Boolean
```

Booleanには、TRUEまたはFALSEを指定します。

例

```
SuppressColHeaderRepeats=False
```

SuppressInvalidCols

このキーワードは、無効なセルが入った列をフォームで非表示にするかどうかを指定します。デフォルトはFALSEです。



注:

この設定に関係なく、管理者以外のユーザーには無効な列は表示されません。

構文

```
SuppressInvalidCols=  
Boolean
```

Booleanには、TRUE(無効な列を非表示にする場合)またはFALSE(無効な列を表示する場合)を指定します。

例

```
SuppressInvalidCols=False
```

SuppressInvalidRows

このキーワードは、無効なセルが入った行をフォームで非表示にするかどうかを指定します。デフォルトはFALSEです。



注:

この設定に関係なく、管理者以外のユーザーには無効な行は表示されません。

構文


```
SuppressInvalidRows=  
Boolean
```

*Boolean*には、TRUE(無効な行を非表示にする場合)またはFALSE(無効な行を表示する場合)を指定します。

例

```
SuppressInvalidRows=False
```

SuppressNoDataCols

このキーワードは、データのないセルが入った列をフォームで非表示にするかどうかを指定します。デフォルトはFALSEです。

構文

```
SuppressNoDataCols=  
Boolean
```

*Boolean*には、TRUE(データのない列を非表示にする場合)またはFALSE(データのない列を表示する場合)を指定します。

例

```
SuppressNoDataCols=False
```

SuppressNoDataRows

このキーワードは、データのないセルが入った行をフォームで非表示にするかどうかを指定します。デフォルトはFALSEです。

構文

```
SuppressNoDataRows=  
Boolean
```

*Boolean*には、TRUE(データのない行を非表示にする場合)またはFALSE(データのない行を表示する場合)を指定します。

例

```
SuppressNoDataRows=False
```

SuppressRowHeaderRepeats

このキーワードは、繰り返される行ヘッダーを非表示にするかどうかを指定します。デフォルトはTRUEです。

構文

```
SuppressRowHeaderRepeats=  
Boolean
```

Booleanには、TRUEまたはFALSEを指定します。

例

```
SuppressRowHeaderRepeats=False
```

SuppressZeroCols

このキーワードは、ゼロが入った列をフォームで非表示にするかどうかを指定します。デフォルトはFALSEです。

構文

```
SuppressZeroCols=  
Boolean
```

Booleanには、TRUE(ゼロの列を非表示にする場合)またはFALSE(ゼロの列を表示する場合)を指定します。

例

```
SuppressZeroCols=False
```

SuppressZeroRows

このキーワードは、ゼロが入った行をフォームで非表示にするかどうかを指定します。デフォルトはFALSEです。

構文

```
SuppressZeroRows=  
Boolean
```

Booleanには、TRUE(ゼロの行を非表示にする場合)またはFALSE(ゼロの行を表示する場合)を指定します。

例

```
SuppressZeroRows=False
```

相対期間の使用

行と列に次のキーワードを使用して、現在のPOVに関連する期間を選択できます：

- @Cur
- @First
- @Last
- @Next
- @Prev



注：

年と期間の値では、大文字と小文字は区別されません。

たとえば、POVで選択されている期間が1月の場合、この式を使用して作成された列または行には、1月のデータが表示されます：

```
P#CUR+1
```

期間が現在の年以降に延長されたとき、または前の年を参照する必要があるときにシステムで相対的な年を適用するには、Y#@Curを行または列の定義の一部として指定する必要があります。

たとえば、現在の期間が最後の期間である場合は、C1=Y#@Cur.P#@Cur(+1)を指定して、正しい年を戻す必要があります。現在のPOVが2013 Decemberの場合、C1=Y#@Cur.P#@Cur(+1)は2014 Januaryを戻します。

現在のPOVが2013Decemberの場合：C2=P#@Cur(+1)は2014 Januaryを戻します(Januaryが年の最初の期間の場合、Decemberは最後です)。したがって、年の情報を正しく反映するには、列の定義に年を指定します。

例：C2=Y#@Cur.P#@Cur(+1)

@CUR関数を他のディメンションに使用して現在のPOVを取得できます。ただし、他のディメンションの@CURは、順序どおりではないため、+2などのオフセットを使用できません。たとえば、現在のPOVが実績シナリオを持つ場

合、列の有効な例は、C1=S#@CURを使用して"実績"を戻すことになります。POVシナリオを予算に変更すると、この同じ列の定義が"予算"を戻します。

競合する属性の優先順位

データ・フォームの小数点以下の桁数、スケールおよびPOVを設定する方法は複数あります。たとえば、小数点以下の桁数の設定は次のように行います。

- フォームの場合 - FormNumDecimals=3
- 列の場合 - C4=A#Inventory, NumDecimals:1
- 行の場合 - R4=A#Sales, NumDecimals:2
- セルの上書きの場合 - R2=S#Actual.P#August, Override(2,7,P#July, NumDecimals:4)

これらの属性のいずれかの設定が、同じ属性の競合する設定と交差する場合があります。たとえば、ある列がスケール設定1を持ち、交差する行がスケール設定2を持つ場合があります。小数点以下の桁数およびスケール属性に適用される値はこの優先順位に基づいていることを理解することは重要です。

- セルの上書き
- 行
- 列
- フォーム
- デフォルト - 小数点以下の桁数の場合、デフォルトは勘定科目から取得します。スケールの場合、デフォルトはエンティティの通貨から取得します。



注:

POVでは、各ディメンションの値は個別に解決されます。たとえば、勘定科目ディメンションは、フォーム・レベルで設定され、シナリオ・ディメンションは行レベルで設定される場合があります。

競合する属性は、優先順位に従って解決されます。たとえば、小数点以下の桁数属性がセルの上書きで定義されている場合、この値が、フォーム、行、列またはデフォルトのかわりに使用されます。

同様に、セルに競合するデータと計算式が指定されている場合、次の優先順位が適用されます。

- SCalc
- データ



注:

フォーム内のSCalc式は左から右、上から下の順番に評価されます。たとえば、行1内のすべての列が順番に評価されると、次に行2のすべての列が評価されます。SCalc式は、評価順序が優先する別のSCalcセルを参照できます。

データ・フォームの編集

データ・フォームは、フォーム・デザイナーを使用して編集できます。一度に編集できるデータ・フォームは1つのみです。



注:

データ・フォームを編集したら、更新済のスクリプトをアプリケーションに再ロードし、変更を確認する必要があります。

▶ データ・フォームを編集するには:

1. 「連結」、「ドキュメント」の順に選択します。
2. データ・フォームを開きます。
3. 「編集」をクリックするか、「アクション」、「編集」の順に選択します。
4. 必要に応じてフォームを編集します。
5. 「保存」をクリックします。

データ・フォームのロード

データ・フォームのスクリプトを作成したら、アプリケーションにロードします。



注:

Webのポップアップ・ブロックの例外にFinancial Managementを追加することをお勧めします。データのロードなどのいくつかのタスクを実行すると、ステータス・ウィンドウにタスクのステータスが表示されます。コンピュータでポップアップ・ブロックを使用可能にしている場合、このステータス・ウィンドウは表示されません。

▶ データ・フォームをロードするには:

1. 「連結」、「ロード」、「ドキュメント」の順に選択します。
2. 「ドキュメント・タイプ」から、「データ・フォーム」を選択します。
3. ファイルの名前を入力するか、「参照」をクリックしてファイルを検索します。



注:

デフォルトでは、データ・フォームのスクリプトにWDFファイル拡張子が使用されます。ロード・プロセスでは、TXTやCSVなどの他のファイル拡張子も受け入れますが、WDFファイル拡張子の使用をお勧めします。

4. オプション: 次のいずれかの手順を実行します:

- ロードするファイルで指定されているセキュリティ・クラスを上書きするには、「セキュリティ・クラスのオーバーライド」を選択して、「セキュリティ・クラス」からセキュリティ・クラスを選択します。
- このフォームを自分専用にするには、「個人用」チェック・ボックスを選択します。このオプションを選択すると、「上書き」オプションおよび「セキュリティ・クラス」リストは使用できません。

- ロードするフォームを追加するまで、これらの手順を繰り返します。
- オプション: データ・フォームを更新中であれば、「既存のドキュメントを上書きします。」を選択します。
- 「ロード」をクリックします。



注:

ロードするすべてのデータ・フォームは、有効なReportLabelを含む必要があります。複数のフォームをロードするように選択し、そのうちの1つに無効なReportLabelがあると、選択したフォームはすべてロードされません。

データ・フォームの抽出

データ・フォームのスクリプトをアプリケーションから抽出できます。スクリプトを抽出してもスクリプトはフォルダまたはアプリケーションから削除されません。選択した場所にスクリプトのコンテンツのみを抽出します。

データ・フォームのスクリプトを抽出したら、これを変更してアプリケーションに再ロードできます。新規スクリプトのモデルを使用することもできます。



注:

スクリプトをアプリケーションに再ロードする場合は、「既存のドキュメントを上書き」オプションを選択して、アプリケーション内の古いファイルを置換する必要があります。

- ▶ データ・フォームを抽出するには:
 - アプリケーションを開きます。
 - 「連結」、「ドキュメント」の順に選択します。
 - 「ドキュメントの抽出」をクリックするか、「アクション」、「ドキュメントの抽出」の順に選択します。
 - ファイルの名前を入力するか、「参照」をクリックしてファイルを検索します。
 - 「抽出」をクリックします。
 - 「保存」をクリックし、ファイルを保存する場所を指定します。

データ・フォームの削除

データ・フォームを削除するには、データ入力フォームの管理セキュリティ役割を持つ管理者であることが必要です。フォルダを削除する場合は、その中のすべてのデータ・フォームを先に削除する必要があります。

- ▶ データ・フォームを削除するには:
 - アプリケーションを開きます。
 - 「連結」、「ドキュメント」、「データ・フォーム」の順に選択します。
 - データ・フォームを選択して「削除」をクリックするか、「アクション」、「削除」の順に選択します。

8

データベースへのデータの抽出

この項の内容:

データ・ソース名(DSN)の構成	200
スター・スキーマ	200
データの作成とスター・スキーマへのエクスポート	203
スター・スキーマの更新	205
スター・スキーマの削除	206
スター・スキーマのテンプレートの作成	206
スター・スキーマのテンプレートの削除	207

この章の手順を使用して、データベースにデータを抽出します。



注:

データをフラット・ファイルとして抽出するには、『Oracle Hyperion Financial Management, Fusion Editionユーザー・ガイド』のデータの抽出に関する項を参照してください。

データを抽出し、Oracle Essbaseデータベースを使用して、データを分析し、レポートを作成できます。スター・スキーマを使用して、データをEssbaseデータベースに送信します。また、スター・スキーマはサードパーティ製のツールでも使用できます。



注:

データをスター・スキーマにエクスポートしたら、データベースを作成します。データベースを作成するには、使用しているリリースのデータベースのドキュメントを参照してください。

データベースにデータを抽出する前に、スター・スキーマを保管するデータベースのデータ・ソース名(DSN)を設定する必要があります。スター・スキーマを複数のデータベースに保管するには、データベースごとにDSNを作成します。手順については、『Oracle Hyperion Enterprise Performance Management Systemインストールおよび構成ガイド』を参照してください。

データベースにデータを抽出するには、管理者または拡張分析のセキュリティ役割が割り当てられている必要があります。

拡張分析のセキュリティ役割が割り当てられていない場合、データの抽出に次の制限事項が適用されます:

- ICPおよびすべてのカスタム・ディメンションの基本レベルのデータのみを抽出できます。
- 1つのシナリオ・ディメンションおよび1つの年ディメンションのみを選択できます。

- ・ シナリオのデフォルトの期間単位の期間のみを選択できます。

データ・ソース名(DSN)の構成

「データベースへのデータの抽出」機能を使用するには、データ・ソース名(DSN)を構成して、データベース・サーバーとアプリケーション・サーバー間の接続を指定する必要があります。DSNはデータベース・サーバー名と、データベース・サーバーへのフル・アクセス権を持つユーザーのユーザー名とパスワードなどの、その他のデータベース関連の情報を指定します。

▶ DSNを作成するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「管理タスク」から「**DSNの構成**」をダブルクリックします。
3. 「データ・ソースの作成」をクリックします。
4. 宛先データベースのデータ・ソース名を入力します。
5. 「データベース・タイプ」ドロップダウン・リストから、データベース・タイプ「**Oracle**」、「**SQL Server**」または「**DB2**」を選択します。
6. ユーザー名とパスワードを入力します。



注:

ユーザーに表を作成、更新および削除する権限があることを確認します。

7. サーバーのホスト名を入力します。
8. ポート番号を入力します。
9. オプション: デフォルトの表領域またはファイルグループ名を入力します。
10. オプション: デフォルトのインデックス表領域またはファイルグループ名を入力します。
11. 「接続のテスト」をクリックします。
12. 「成功」の確認メッセージが表示されたら、「保存」をクリックしてDSNを保存します。
13. 「リフレッシュ」をクリックして更新されたDSN情報を表示します。

接続を作成して保存すると、「DSNの構成」ページに表示されます。後で編集する必要がある場合は「DSNの編集」をクリックして編集できます。また、不要になった場合は削除できます。

スター・スキーマ

アプリケーションごとに複数のスター・スキーマを作成できます。各スター・スキーマの表名は、ユーザーが指定する接頭辞で始まります。すべてのディメンションから複数のメンバーを選択して、エクスポートする情報を最適に表すスター・スキーマを作成できます。



注:

セル・テキストとライン・アイテムの詳細はスター・スキーマにエクスポートされません。

スター・スキーマ内のデータの組合せは、エクスポートするために選択するディメンションのメンバーに基づいて作成されます。選択するディメンションのメンバーが多くなると、スター・スキーマの作成に必要なデータの組合せの数が多く

なるため、エクスポート・プロセスの完了に必要な時間が長くなります。データの組合せの数を計算するには、選択したメンバーの数をディメンションごとに乗算します。



注意

すべてのディメンションのすべてのメンバーをエクスポートする選択はしないでください。エクスポートするデータのセグメントを選択します。アプリケーションの大きさや、データの組合せ数によって、エクスポート時間が非常に長くなる場合があります。

たとえば、次のデータをエクスポートできます。

- ・ シナリオ - 実績
- ・ 年 - 2014
- ・ 期間 - 1月
- ・ ビュー - 期別
- ・ エンティティ - 地域、米国、フロリダ、コネチカット
- ・ 値 - USD
- ・ 勘定科目 - 粗利益、売上、総費用
- ・ ICP - [なし]
- ・ C1 - ボール、テニス・ボール、ゴルフ・ボール
- ・ C2 - 全顧客、顧客2、顧客3、顧客4、顧客5
- ・ C3 - [なし]
- ・ C4 - [なし]

次に、作成されるスター・スキーマをOracle Essbase Integration Servicesで使用して、情報を参照および使用する必要がある対象ユーザーを表すために、1つ以上のデータ・キューブを作成できます。スター・スキーマには、これらのメンバーのデータの組合せが180含まれます($1 * 1 * 1 * 1 * 4 * 1 * 3 * 1 * 3 * 5 * 1 * 1 = 180$)。



注:

Oracle Essbase Integration Servicesを介してメタデータをOracle Essbaseにエクスポートする場合、データベースへのデータの抽出機能によって、Financial Managementアプリケーションで表示されているものとは異なる順序で、メタデータが使用されます。

必要な場合は、現地通貨のデータのみを抽出できます。現地通貨のデータのみを抽出するには、視点の値ディメンションからエンティティの通貨メンバーを使用します。エンティティの通貨は、通貨のプレースホルダとして機能し、選択されている各エンティティのデフォルトの通貨を抽出します。

選択したシナリオがYTDの場合、データ抽出により期別の派生データが抽出されます。たとえば、シナリオがZeroView=YTDであるとし、最初の期間の支出勘定科目に、値100があります。2番目の期間では、その勘定科目にデータは報告されていません。2番目の期間の期別の派生データは-100で、YTDが0になります。この視点に対して、期間ベースでデータベースへのデータの抽出が実行されている場合、最初の3つの期間の値は、それぞれ100、-100および0になります。

スター・スキーマのフォーマット

スター・スキーマを作成する場合は、抽出フォーマットの次のオプションのいずれかを選択します。

- 標準
- メタデータのみ
- 選択したメタデータのみ
- Essbase
- データ・ウェアハウス

選択する抽出フォーマットのオプションによって、使用するスキーマ・フォーマットが決定されます。各スキーマ・フォーマットは、異なる一連の表を生成します。次のスキーマ・フォーマットを使用できます。

- 標準Essbaseスキーマ - このスキーマは、抽出フォーマット・タイプが標準、メタデータのみ、選択したメタデータの場合に使用します。
- SQLおよびEssbaseスキーマ - このスキーマは、抽出フォーマット・タイプがEssbaseの場合に使用します。
- ウェアハウス正規化階層スキーマ - このスキーマは、抽出フォーマット・タイプがデータ・ウェアハウスの場合に使用します。

接頭辞表

各スキーマ・フォーマットでは、ディメンション表と1つのデータ・フィールドのキーを含むPREFIX_FACT表が作成されます。また、メタデータの抽出タイムスタンプを追跡するPREFIX_DIMENSION表、HFM_EA_EXTRACT表、およびライター・ロックおよびリーダー・ロックを追跡するPREFIX_LOCK_ACCESS表も作成されます。ライター・ロックは、作成または置換プロセスを使用する場合など、メタデータの変更時に使用されます。リーダー・ロックは、更新プロセスを使用する場合など、メタデータが変更されず、FACT表のデータが更新される場合に使用されます。これらのルールは、ロック・プロセスに使用されます：

- 同じDBMSインスタンスの同じ接頭辞には、一度に1つの書込みのみ行えます。
- ライターが実行中またはキューに存在する場合は、リーダーは書込みが完了するまで実行できません。
- 視点に関係なく、同時に複数の読取りを行えます。



注：

エンティティ・ディメンションの2つの表(PREFIX_ENTITYおよびPREFIX_PARENT)が作成されます。

SQLおよびEssbase集約スキーマ・フォーマットでは、PREFIX_DIMENSION_BASE表も作成されます。

ウェアハウス正規化階層スキーマ・フォーマットでは、PREFIX_DIMENSION_PARENT表が作成されます。



注：

ビュー、年および親ディメンションには、BASE表とPARENT表は作成されません。

たとえば、リレーショナル表の接頭辞がDEMOで、フォーマットがEssbaseの場合は、次の表が作成されます。

- HFM_EA_EXTRACT1
- HFM_LOCK_ACCESS1
- DEMO_FACT
- DEMO_YEAR
- DEMO_VIEW
- DEMO_PARENT
- DEMO_SCENARIOおよびDEMO_SCENARIO_BASE
- DEMO_PERIODおよびDEMO_PERIOD_BASE
- DEMO_VALUEおよびDEMO_VALUE_BASE
- DEMO_ENTITYおよびDEMO_ENTITY_BASE
- DEMO_ICPおよびDEMO_ICP_BASE
- DEMO_ACCOUNTおよびDEMO_ACCOUNT_BASE
- DEMO_CUSTOMおよびDEMO_CUSTOM_BASE

データの作成とスター・スキーマへのエクスポート

スター・スキーマを作成するには、スキーマの表を識別する接頭辞とエクスポートするセルのディメンションのメンバーを指定します。データをエクスポートする前に、アプリケーション・データが連結されていることを確認します。



注意

Financial Managementデータベースをデータのエクスポートの宛先データベースとして使用しないでください。



注:

Oracleデータベース・ユーザーの場合: 抽出されるテキスト・データは、NVARCHAR(Unicode)フォーマットで保管されます。SELECT文でOracleの「translate」コマンドを使用して、テキストをUnicodeからANSIフォーマットに変換します。たとえば、SELECT translate (LABEL using CHAR_CS) as LABEL FROM DEMO_ACCOUNTのように使用します。



注:

大量のデータを抽出する場合は、抽出用に十分なディスク・スペースを確保してください。

Webのポップアップ・ブロックの例外にFinancial Managementを追加することをお勧めします。データのロードなどのいくつかのタスクを実行すると、ステータス・ウィンドウにタスクのステータスが表示されます。コンピュータでポップアップ・ブロックを使用可能にしている場合、このステータス・ウィンドウは表示されません。

▶ スター・スキーマを作成し、Financial Managementデータを抽出するには:

1. 「連結」、「抽出」、「データ」の順に選択します。
2. 次のいずれかのタスクを行います:
 - POVディメンションをテンプレートに保存した場合、「テンプレート」から、テンプレートを選択します。
 - テンプレートを使用していない場合、エクスポートするPOVディメンション・メンバーを選択します。
3. 「抽出先」セクションの「タイプ」オプションで「データベース」を選択します。



注:

管理者または拡張分析のセキュリティ役割が割当てられている場合のみ、このオプションを使用できます。

4. 「抽出フォーマット」で、次のいずれかのオプションを選択します:
 - 標準
 - メタデータのみ - メタデータのみを抽出します。
 - 選択したメタデータのみ - 選択したディメンションのメンバーのメタデータのみを抽出します。
 - **Essbase**
 - データ・ウェアハウス
5. 「オプション」から、1つ以上のオプションを選択します。
 - 動的勘定科目の抽出
 - 算出データ
 - 派生データ
6. 「ライン・アイテムの詳細」から、オプションを選択します。
 - セルの合計要約
 - なし
7. 「スキーマ・アクション」から、「スター・スキーマの作成」を選択します。
8. 「宛先データベース(DSN)」から、スター・スキーマのエクスポート先のデータベースを選択します。



注:

宛先データベースとして、データのエクスポート元と同じFinancial Managementデータベースを使用しないでください。

9. 「リレーショナル表の接頭辞」で、スター・スキーマの表を識別する接頭辞を入力するか、デフォルトのアプリケーション名の接頭辞を使用します。



注:

接頭辞には、英数字を10文字まで入力でき、英字で始まる必要があります。アンダースコアは使用できません。デフォルトの接頭辞はアプリケーション名です。

10. 「抽出」をクリックします。

スター・スキーマの更新

定義済のスター・スキーマにデータをエクスポートできます。スター・スキーマを更新するとき、別のディメンションのメンバーを指定できます。



注:

スター・スキーマを更新すると、ファクト表が更新されます。

Webのポップアップ・ブロックの例外にFinancial Managementを追加することをお勧めします。データのロードなどのいくつかのタスクを実行すると、ステータス・ウィンドウにタスクのステータスが表示されます。コンピュータでポップアップ・ブロックを使用可能にしている場合、このステータス・ウィンドウは表示されません。

▶ スター・スキーマを更新するには:

1. 「連結」、「抽出」、「データ」の順に選択します。
2. 次のいずれかのタスクを行います:
 - POVディメンションをテンプレートに保存した場合、「テンプレート」から、テンプレートを選択します。
 - テンプレートを使用していない場合、エクスポートするPOVディメンション・メンバーを選択します。
3. 宛先タイプの抽出で、「データベース」を選択します。



注:

管理者または拡張分析のセキュリティ役割が割当てられている場合のみ、このオプションを使用できます。

4. 「抽出フォーマット」で、次のいずれかのオプションを選択します:
 - 標準
 - メタデータのみ - メタデータのみを抽出します。
 - 選択したメタデータのみ - 選択したディメンションのメンバーのメタデータのみを抽出します。
 - Essbase
 - データ・ウェアハウス
5. 「オプション」から、1つ以上のオプションを選択します。
 - 動的勘定科目の抽出
 - 算出データ
 - 派生データ
6. 「ライン・アイテムの詳細」から、オプションを選択します。
 - セルの合計要約
 - なし
7. 「スキーマ・アクション」から、「スター・スキーマの更新」を選択します。
8. 「宛先データベース(DSN)」から、スター・スキーマのエクスポート先のデータベースを選択します。



注:

宛先データベースとして、データのエクスポート元と同じFinancial Managementデータベースを使用しないでください。

9. 「リレーショナル表の接頭辞」で、スター・スキーマの表を識別する接頭辞を入力するか、デフォルトのアプリケーション名の接頭辞を使用します。



注:

接頭辞には、英数字を10文字まで入力でき、英字で始まる必要があります。アンダースコアは使用できません。デフォルトの接頭辞はアプリケーション名です。

10. 「抽出」をクリックします。

スター・スキーマの削除

不要になったスター・スキーマは削除できます。スター・スキーマを削除すると、スター・スキーマのデータ、メタデータおよび表がすべて削除されます。

▶ スター・スキーマを削除するには:

1. 「連結」、「抽出」、「データ」の順に選択します。
2. 宛先タイプの抽出で、「データベース」を選択します。
3. 「宛先データベース(DSN)」から、スター・スキーマを含むデータベースを選択します。
4. 「リレーショナル表の接頭辞」に、スター・スキーマの表を識別する接頭辞を入力します。



注:

接頭辞には、英数字を10文字まで入力でき、英字で始まる必要があります。デフォルトの接頭辞はアプリケーション名です。

5. 「スキーマ・アクション」から、「スター・スキーマの削除」を選択します。

スター・スキーマのテンプレートの作成

スター・スキーマのテンプレートを作成し、POVに名前を付けて保存すると、再使用できます。

▶ スター・スキーマのテンプレートを作成するには:

1. 「連結」、「抽出」、「データ」の順に選択します。
2. POVディメンションのメンバーを選択します。
3. 「宛先データベース(DSN)」から、スター・スキーマのエクスポート先のデータベースを選択します。
4. 「リレーショナル表の接頭辞」に、スター・スキーマの表を識別する接頭辞を入力します。



注:

接頭辞には、英数字を10文字まで入力でき、英字で始まる必要があります。デフォルトの接頭辞はアプリケーション名です。

5. 「**テンプレートの保存**」をクリックします。
6. テンプレート名を入力して、「**OK**」をクリックします。

スター・スキーマのテンプレートの削除

- ▶ スター・スキーマのテンプレートを削除するには:
1. 「**連結**」、「**抽出**」、「**データ**」の順に選択します。
 2. 「**テンプレート**」で、削除するテンプレートを選択します。
 3. 「**テンプレートの削除**」をクリックします。
 4. 削除を確認するメッセージが表示されたら、「**OK**」をクリックします。

9

レポートの定義

この項の内容:

仕訳レポート・スクリプトの定義	209
内部取引照合レポート・スクリプトの定義	210
内部取引レポート・スクリプトの定義	223

Financial Managementでは、次のレポートの種類を定義できます:

- 仕訳レポート: 仕訳モジュールで選択した条件に従って、特定の仕訳または仕訳リストの情報を示します。
- 内部取引パートナ(ICP)照合レポート: 連結するときに消去される内部取引を示します。

レポートごとに新しいレポート定義を作成するか、レポート定義を開いてそれを変更できます。

次の方法のいずれかを使用して、レポートを定義します:

- レポート・ページ・オプションを使用して、値を指定します。『Oracle Hyperion Financial Managementユーザーガイド』を参照してください。
- テキスト・エディタを使用して、スクリプトのレポート・オプションを指定します。

すべてのレポートのヘッダーに、日付と時刻、およびユーザー・フィールドが自動的に表示されます。視点の定義はレポートごとに異なります。レポートの例の詳細は、[209ページの仕訳レポート・スクリプトの定義](#)および[210ページの内部取引照合レポート・スクリプトの定義](#)を参照してください。

サンプルの内部取引パートナ照合レポートは、Financial Managementのサンプル・アプリケーションをインストールするときに取り込まれます。これらのファイルは、Financial Managementをインストールしたディレクトリの「Sample Applications」フォルダに配置されます。

仕訳レポート・スクリプトの定義

仕訳レポートを作成して、仕訳ステータスをチェックしたり、仕訳の調整を見直したりします。仕訳レポートを作成して、特定の仕訳または仕訳のリストの情報を表示できます。たとえば、特定のステータス、残高の種類、エンティティまたは勘定科目を持つ仕訳のみを表示するように選択できます。また、レポートに表示する列を選択したり、その列のソート順を変更することもできます。

レポート定義の仕訳レポートにエンティティや勘定科目のフィルタを設定できます。エンティティ・フィルタの構文は、EntityFilter=ParentName.ChildName(UnitedStates.Marylandなど)です。勘定科目フィルタの構文は、AccountFilter=MemberName(Salesなど)です。

この例では、サンプルの仕訳レポートの定義を示しています。

```
ReportType=Journal
```

```
ReportDescription=Tax Journals
```

```
POV=S#Actual.Y#2014.P#January.V#<Entity Curr Adjs>
```

```
DisplayColumn_0=Account,Ascending,NoRepeat,Yes,Label
```

```
DisplayColumn_1=ICP,,NoRepeat,No,Label
```

```
DisplayColumn_2=Products,,NoRepeat,No,Label
```

```
DisplayColumn_3=Markets,,NoRepeat,No,Label
```

```
StatusFilter=Working,Submitted,Rejected,Approved,Posted
```

```
TypeFilter=Regular
```

```
BalanceTypeFilter=Balanced,Unbalanced
```

```
EntityFilter=UnitedStates.Maryland
```

```
AccountFilter=Sales
```

内部取引照合レポート・スクリプトの定義

内部取引照合レポートを使用すると、内部取引を追跡して、分析や監査に役立てることができます。内部取引照合レポートは、内部取引パートナー(IsICP)属性を使用可能にして設定されたエンティティや内部取引パートナー・ディメンションの一致を示します。

内部取引パートナー(ICP)照合レポートは、連結時に消去される内部取引を表示します。内部取引は、内部取引パートナー・ディメンションで管理されます。ICPディメンションは、勘定科目として存在できるすべての内部取引残高を含んでいます。データ・グリッド、データ・ロード、仕訳またはデータ・フォームを使用して、内部取引を入力できま

す。Financial Managementでは、勘定科目およびカスタム・ディメンションの内部取引詳細を追跡して消去できます。内部取引は、2つの内部取引エンティティの共通の親が最初に見つかった時点で消去されます。内部取引は、値ディメンションの[Elimination]メンバーを使用して消去されます。

ユーザー・インタフェースを使用して、内部取引照合レポートを作成できます。

レポートの勘定科目を選択するか、調整勘定科目オプションを使用できます。調整勘定科目オプションを使用すると、調整勘定科目に基づいて勘定科目や照合勘定科目が生成されます。戻し取引、カスタム・ディメンションまたは内部取引詳細を非表示にするように選択できます。レポートを印刷するとき、レポート設定を上書きして、必要に応じたレポートをカスタマイズできます。

内部取引照合レポートのメンバー・リストの選択

生成されたレポートに視点ディメンションを含むディメンション・メンバーを指定します。レポートの「エンティティ」および「パートナ」のメンバー・リストを選択できます。たとえば、「エンティティ」フィールドの「地域」のメンバー・リストを選択できます。「エンティティ」および「パートナ」のメンバー・リストを選択すると、「エンティティ」リスト内のすべてのエンティティと「パートナ」リスト内のすべてのエンティティを対比してICP取引が処理されます。ICP属性が使用可能なエンティティの取引のみが処理されます。

内部取引照合レポートの勘定科目の選択

勘定科目および取引と照合する照合勘定を定義する必要があります。勘定科目と照合勘定を調整勘定に基づいて生成する場合は、「調整勘定」オプションを指定すると、勘定科目と照合勘定が自動的に生成されます。

内部取引照合レポートの小数点以下の桁数の指定

Decimalキーワードを内部取引照合レポート定義に追加して、レポートに表示する小数点以下の桁数を指定できます。小数点以下の桁数はデフォルトで0-9です。小数点以下の桁数を指定しない場合は、勘定科目に定義されているデフォルトの小数点設定が使用されます。

内部取引照合レポート用スタイル・シートの選択

レポートの作成時にドロップダウン・リストからスタイル・シートを選択できます。レポート定義にStyleSheetキーワードを追加して、レポートに使用するスタイル・シートを指定することもできます。たとえば、StyleSheet=HFM_IntercompanyDefault.xlsと指定します。レポート定義にスタイル・シートを指定しない場合、デフォルトのスタイル・シートが使用されます。

内部取引照合レポートの通貨の指定

指定する通貨で内部取引照合レポートを生成できます。これにより、金額を連結する前に、レポートを実行して、共通の通貨で内部取引を検証できます。たとえば、EUR通貨で値をチェックするには、視点の値をUSDからEURに変更してレポートを生成します。

まだ換算されていない通貨を使用してレポートを定義する場合は、ルール・ファイルのサブ換算セクションに定義されている換算ルールを使用して、換算プロセスが実行されます。また、換算された金額が対応する通貨の値ディメンション・メンバーに保管されます。ただし、レポートの通貨が以前に換算済で、エンティティの換算ステータスがOKの場合、再換算は不要で、内部取引照合レポートの処理には保管されている換算済の金額が使用されます。

たとえば、EUR通貨の内部取引照合レポートを実行すると、まず、換算済データがEUR通貨(V#EUR)用に作成されているかどうかを確認されます。また、換算ステータスがOKであることも確認されます。エンティティの換算ステータスがTR (換算が必要)である場合は、換算済データが有効になるように再換算が行われます。それ以外の場合、処理には保管されている換算済の金額が使用されます。ただし、データがEURに換算されていない場合は、サブ換算に定義されているように換算プロセスが実行され、換算された金額がEURの値メンバーに保管されます。EURの換算済の金額も、内部取引照合レポートに使用されます。

内部取引照合レポートの非表示オプション

内部取引照合レポートを作成するとき、いくつかのタイプの内部取引と詳細をレポートで非表示にできます。レポートの印刷時にもこれらの非表示オプションを選択できます。

一致したものの非表示

内部取引照合レポートを作成するとき、一致するエンティティ/パートナーの取引を表示または非表示にできます。非表示にするように選択すると、エンティティおよびパートナーの値が照合許容差の値またはパーセントの範囲内にある場合、取引は非表示にされます。たとえば、エンティティに299個、パートナーに200個ある場合、差異は99になります。照合許容差の値が100でエンティティとパートナーの差異がこの例のように100未満の場合は、照合許容差の範囲内であるため、取引は非表示にされます。

エンティティ/パートナーの取引を非表示にしないように選択すると、エンティティおよびパートナーの値が照合許容差の値またはパーセントの範囲内にある場合でも非表示にされません。前の例では、値の差異が99で、照合許容差の範囲内にありますが、取引は非表示にされません。指定された照合許容差は単位で表示されます。

戻し取引の非表示

デフォルトで、エンティティの取引が表示されるたびに内部取引パートナーから戻し取引が表示されます。レポートの生成時に、これらの戻し取引を非表示にするように選択できます。

レポートに1つの照合勘定科目しかないときにこのオプションは非常に便利です。たとえば、収益と費用の内部取引の保管に使用する「現金」勘定科目がある場合です。この場合、レポートには1つの照合勘定科目しかないため、戻し取引の非表示オプションを使用して、重複エントリを避ける必要があります。

詳細の非表示

レポートに異なる金額のみを表示する場合は、内部取引の詳細を非表示にできます。このオプションを選択すると、レポートには内部取引は表示されず、エンティティ/パートナーの各セクションの合計差異のみが印刷されます。不一致が存在するために、各内部取引を表示する必要がある場合は、レポートを再生成して、内部取引の詳細を表示できます。

カスタム・ディメンションの非表示

任意のカスタム・ディメンションの列を非表示にするように選択できます。

「メンバー表示」オプション

ディメンション・メンバーのラベル、説明、または両方をレポートに表示できます。

「グループ化の基準」オプション

内部取引パートナーとの取引をカスタム・ディメンションで分類できます。このオプションに従って詳細がソートされ、グループの小計が表示されます。

内部取引照合レポート・スクリプトのキーワード

このセクションのキーワードを使用して、内部取引パートナー照合レポート・スクリプトを定義します。スクリプトを作成したら、ファイル名に拡張子RPTを付けて保存します。



注:

内部取引パートナー照合レポート・スクリプトではキーワードの大文字と小文字が区別されません。

ReportType

このキーワードは、レポート・タイプを指定します。スクリプトでは必須キーワードです。

構文

```
ReportType=Intercompany
```

ReportDescription

このキーワードは、レポートの説明を指定します。レポートの説明には、40文字まで入力できます。スクリプトでは必須キーワードです。

構文

```
ReportDescription=  
ReportDescription
```

*ReportDescription*をレポートの説明に置換します。例:

```
ReportDescription=Intercompany Elimination Report
```

StyleSheet

このキーワードは、レポートに使用するスタイル・シートを指定します。

構文

```
StyleSheet=  
StyleSheetFileName
```

*StyleSheetFileName*をレポートのスタイル・シートに置換します。例:

```
ReportDescription=HFM_IntercompanyDefault.xsl
```

レポート定義にスタイル・シートを指定しない場合、デフォルトのスタイル・シートが使用されます。

POV

このキーワードは、レポートの視点を指定します。スクリプトでは必須キーワードです。

構文

```
POV=S#  
Scenario  
.Y#  
Year  
.P#  
Period  
.V#  
Value  
.W#  
View
```

Scenario、*Year*、*Period*、*Value*および*View*を有効なディメンションのメンバーに置換します。例:

```
POV=S#Actual.Y#2014.P#January.V#USD.W#YTD
```



注:

Adjs値メンバー(たとえばparent curr adjs)はサポートされていません。

エンティティ

このキーワードは、レポートに表示するエンティティまたはエンティティのメンバー・リストを指定します。

構文

```
Entity=E#  
Parent.Entity
```

```
Entity=E{  
EntityList  
}
```

*Entity.Parent*をエンティティと親の組合せに置換します。*EntityList*を有効なメンバー・リストに置換します。例:

```
Entity=E#UnitedStates.Connecticut
```

```
Entity=E{Geographical.[Base]}
```

Partner

このキーワードは、レポートに表示するパートナーまたはパートナーのメンバー・リストを指定します。

構文

```
Partner=E#  
PartnerParent  
.  
PartnerEntity
```

```
Partner=E{  
PartnerList  
}
```

*PartnerParent.PartnerEntity*をパートナーの親とエンティティの組合せに置換します。*PartnerList*を有効なパートナーのメンバー・リストの名前に置換します。例:

```
Partner=E#UnitedStates.Florida
```

```
Partner=E{Geographical.[Base]}
```

AccountEntityとAccountPartner

AccountEntity_xおよびAccountPartner_xキーワードは、照合の勘定科目を指定します。照合する勘定科目のペアごとに、ゼロで始まるAccountEntity_xおよびAccountPartner_xを指定します。1対多または多対多の照合レポートを作成するには、正しいキーワードを使用して追加の勘定科目を指定します。1つのレポート内に重複するキーワードを使用できません。たとえば、1つのレポート内にAccountEntity_0が複数存在することはできません。

構文

```
AccountEntity_0=A#Sales.C1#GolfBalls.C2#Customer2
```

```
AccountPartner_0=A#Purchases.C1#GolfBalls.C2#Customer2
```

この例では、これらのキーワードを使用して、1つの照合レポートですべての勘定科目を示します。

```
AccountEntity_0=A#1004780
```

```
AccountEntity_1=A#1004790
```

```
AccountEntity_2=A#1005850
```

```
AccountEntity_3=A#1005850
```

```
AccountPartner_0=A#2000100
```

```
AccountPartner_1=A#2000140
```


AccountPartner_2=A#2000210

AccountPartner_3=A#2000250

AccountPartner_4=A#2000320

AccountPartner_5=A#2000430

AccountPartner_6=A#2000560

AccountPartner_7=A#2000630

AccountPartner_8=A#2000680

この例では、EntityおよびPartnerキーワードを使用して、異なる勘定科目のペアで異なるレポートを作成します。これは、1つの勘定科目を2つの勘定科目と照合するレポートの一例です(1対多)。

Report 1

AccountEntity_0=A#1004780

AccountPartner_0=A#2000100

AccountPartner_1=A#2000140

この例では、2つの勘定科目を1つの勘定科目と照合するレポートを示しています(多対1)。

Report 2

AccountEntity_0=A#1004790

AccountEntity_1=A#2000210

AccountPartner_1=A#2000250

下の例は、1つの勘定科目を1つの勘定科目と照合しています。

```
Report 3
```

```
AccountEntity_0=A#1005850
```

```
AccountPartner_0=A#2000320
```

下の例は、2つの勘定科目を4つの勘定科目と照合しています。

```
Report 4
```

```
AccountEntity_0=A#1005850
```

```
AccountEntity_1=A#1005860
```

```
AccountPartner_0=A#2000430
```

```
AccountPartner_1=A#2000560
```

```
AccountPartner_2=A#2000630
```

```
AccountPartner_3=A#2000680
```

SuppressIfMatch

このキーワードは、エンティティとパートナーの値が照合許容差内であれば、取引を非表示にします。

構文

```
SuppressIfMatch=  
Boolean
```

BooleanをYesまたはNoに置換します。例:

SuppressIfMatch=Yes

たとえば、エンティティに299個、パートナに200個ある場合、差異は99になります。照合許容差の値が100で、この例のようにエンティティとパートナの差異が100未満の場合は、照合許容差内なので取引が非表示になります。

エンティティ/パートナの取引を非表示にしないように選択すると、エンティティおよびパートナの値が照合許容差の値の範囲内にある場合でも非表示にされません。前の例では、値の差異が99で、照合許容差の範囲内にありますが、取引は非表示にされません。



注:

照合許容差は単位で表示されます。

SuppressReversedTransactions

このキーワードは、エンティティの取引について、そのパートナからの取引の戻しを非表示にします。

構文

SuppressReversedTransactions=
Boolean

*Boolean*をYesまたはNoに置換します。例:

SuppressReversedTransactions=Yes

デフォルトで、エンティティの取引が表示されるたびに内部取引パートナから戻し取引が表示されます。レポートの作成時に、これらの戻し取引を非表示にするように選択できます。

このオプションは、収益と費用の内部取引について記述するwash勘定科目が1つある場合など、レポートに照合勘定が1つある場合に便利です。この場合、レポートには1つの照合勘定しかないなので、戻し取引の非表示オプションを使用して、重複エントリを避ける必要があります。

SuppressDetails

このキーワードは、内部取引の詳細を非表示にして、エンティティ/パートナの各セクションの合計差異のみを印刷します。

構文

```
SuppressDetails=  
Boolean
```

BooleanをYesまたはNoに置換します。例:

```
SuppressDetails=Yes
```

差額のみを表示する内部取引照合レポートを作成する場合、レポートの作成時に内部取引の詳細を非表示にできません。このオプションを選択すると、レポートには内部取引は表示されず、エンティティ/パートナーの各セクションの合計差異のみが印刷されます。不一致が存在するために、各内部取引を表示する必要がある場合は、レポートを再生成して、内部取引の詳細を表示できます。

MatchingTolerance

このキーワードを使用して、貸借の不一致が一定額を超える取引のみを表示するための値を指定するか、デフォルト値の0を使用します。



注:

照合許容差は単位で表示されます。

構文

```
MatchingTolerance=ToleranceValue
```

ToleranceValueを10億未満の数値に置換します。このキーワードの上限は、999999999です。例:

```
MatchingTolerance=100
```

たとえば、エンティティに299個、パートナーに200個ある場合、差異は99になります。照合許容差の値が100で、この例のようにエンティティとパートナーの差異が100未満の場合は照合許容差内なので、SuppressIfMatchキーワードを使って、取引を非表示にします。

SuppressCustoms

このキーワードは、カスタム・ディメンションを非表示にします。

構文

```
SuppressCustomAlias=Boolean
```

*Boolean*をYesまたはNoに置換します。デフォルトはYesです。例:

```
SuppressCustomFlows=Yes
```

ScaleFactor

このキーワードは、小数点の位置を識別して値を表示する単位を指定します。たとえば、位取り3を入力すると、レポートの値は、1000単位で表示されます。位取りが0の場合、レポートの値は1単位で表示されます。

構文

```
ScaleFactor=  
Scale
```

*Scale*を次の番号のいずれかに置換します。

- 0 = 1単位
- 1 = 十
- 2 = 百
- 3 = 千
- 4 = 1万
- 5 = 10万
- 6 = 百万
- 7 = 1千万
- 8 = 1億
- 9 = 十億

例:

```
ScaleFactor=3
```

この例では、数字12345.78はレポートで12.345678と表示されます。

Decimal

このキーワードは、レポートに表示する小数点以下の桁数を指定し、勘定科目で定義した設定を上書きできます。

構文

```
Decimal=  
NumberDecimalPlaces
```

*NumberDecimalPlaces*を0-6の数値に置換します。小数点以下の桁数を指定しない場合は、勘定科目に定義されているデフォルトの小数点設定が使用されます。例:

```
Decimal=3
```

この例では、数字123.4567はレポートで123.457と表示されます。

DisplayLabels

このキーワードは、メンバーのラベルをレポートに表示するかどうかを指定します。

構文

```
DisplayLabels=  
Boolean
```

*Boolean*をYesまたはNoに置換します。デフォルトはYesです。例:

```
DisplayLabels=Yes
```

DisplayDescriptions

このキーワードは、メンバーの説明をレポートに表示するかどうかを指定します。

構文

```
DisplayDescriptions=  
Boolean
```

*Boolean*をYesまたはNoに置換します。デフォルトはNoです。例:

```
DisplayDescriptions=Yes
```

DisplayPlugElimAccts

このキーワードは、内部取引の影響を受けた調整勘定の要約を表示するかどうかを指定します。

構文

```
DisplayPlugElimAccts=  
Boolean
```

*Boolean*をYesまたはNoに置換します。例:

```
DisplayPlugElimAccts=Yes
```

GroupByCustom

このキーワードは、ディメンションごとにカスタム・ディメンション取引をグループ化します。このキーワードはオプションです。

構文

```
GroupByCustom=  
Custom
```

*Custom*をカスタム・ディメンションの別名に置換します。レポートのカスタム・ディメンションは、このディメンションで分類されます。例:

```
GroupByCustom=Flows
```

内部取引レポート・スクリプトの定義

次のタイプの内部取引レポートを定義できます。

- 内部取引 - 取引のリストを作成します。
- 勘定科目による内部取引照合 - 選択した勘定科目に基づく照合レポートを作成します。

- ・ IDによる内部取引照合 - 取引IDに基づく照合レポートを作成します。

表30 内部取引レポート・スクリプトのキーワード

キーワード	説明
ReportType	レポート・タイプを指定します。例: ReportType=ICTransactions
ReportLabel	レポート名を指定します。例: ReportLabel=Transaction Report
ReportDescription	レポートの説明を入力します。例: ReportDescription=Intercompany Transaction Detail Report
ReportSecurityClass	レポートのセキュリティ・クラスを指定します。デフォルトは [Default] です。
POV	レポートの有効な視点を指定します。例: POV=S#ActMon.Y#2014.P#January
ScaleFactor	レポートの位取りを指定します。スケールは、0から9までの値です。
Decimal	レポートに表示する小数点以下の桁数を指定します。小数点以下の桁数は、0から9までの値です。
FromAmtおよびToAmt	取引金額の範囲を指定します。
IncludeMatched	レポートに照合済の取引を含める場合はTRUEを、含めない場合はFALSEを指定します。
IncludeUnMatched	レポートに未照合の取引を含める場合はTRUEを、含めない場合はFALSEを指定します。
IncludeMisMatched	レポートに不一致の取引を含める場合はTRUEを、含めない場合はFALSEを指定します。
IncludePosted	レポートに転記済の取引を含める場合はTRUEを、含めない場合はFALSEを指定します。
IncludeUnPosted	レポートに転記の戻し済の取引を含める場合はTRUEを、含めない場合はFALSEを指定します。
Entity	エンティティは有効なICPの親エンティティではなく、基本エンティティである必要があります。
Partner	パートナ・エンティティは、勘定科目の有効なICPエンティティである必要があります。
Entity Account	エンティティ取引を表示している場合は、エンティティ勘定を指定します。
Partner Account	パートナ取引を表示している場合は、パートナ勘定を指定します。
TransactionID	このIDは必須です。取引のIDを40文字以内で入力する必要があります。サブIDと組み合わせて使用する場合、この

キーワード	説明
	IDは、シナリオ/年/期間において、エンティティ/パートナ/勘定科目/C1/C2の一意的識別子になります。
TransactionSubID	取引サブIDを指定します。
TransactionCurrency	これは、請求書取引に使用する通貨です。アプリケーションで定義された有効な通貨である必要があります。
ReferenceID	<p>これはオプションです。参照IDを入力して取引の参照情報を保管できます。たとえば、エンティティは、請求書を発行したエンティティとは異なる固有の請求書一連番号を持つ場合があります。情報目的のみのために、この参照IDに追加情報を入力できます。</p> <p>エンティティの参照IDを「取引ID」に入力し、パートナ・エンティティから受け取った請求書の番号を入力できます。</p>
MatchCode	<p>これはオプションです。照合コードは、照合プロセスのタイプを区別するための接頭辞です。</p> <ul style="list-style-type: none"> • A - 勘定科目を使用する自動照合 • I - 取引IDを使用する自動照合 • R - リファレンスIDを使用する自動照合 • M - 手動照合
ReasonCode	<p>これはオプションです。理由コードは、管理者が定義した有効な理由コードであることが必要です。理由コードの主な目的は、取引が「不一致」ステータスを持つようになった理由(パートナ・エンティティからの請求書を紛失した、パートナによって正しくない額が入力されたなど)を示すことです。取引が「一致」ステータスを持つ場合は、取引に理由コードを割り当てる必要はありません。取引が「未照合」ステータスを持つ場合は、理由コードを割り当てることはできません。</p>
FromDate	オプションです。これは有効な日付であることが必要です。
ToDate	オプションです。これは有効な日付であることが必要です。
DisplayColumns Section	<p>レポートに表示される列およびそれらの列の表示形式を指定します。構文は、<DisplayedColumns>, <Sort>, <Label>, <Repeat>, <Totals>です。ここで、<Displayed Columns>は、有効な列、<Sort>はSortまたはNo Sort、<Label>は、ラベル、説明、または両方、<Repeat>はRepeatまたはNoRepeat、<Total>はTotalまたはNoTotalです。</p> <p>カスタム・ディメンションの場合、列に対して次のような別名が使用されます:</p> <p>Customers, NoSort, Label, Repeat, NoTotal</p>

キーワード	説明
DisplayEntityTransactions	<p>レポートにエンティティ取引を含める場合はTRUEを、含めない場合はFALSEを指定します。</p> <p>特定のエンティティおよびパートナーの内部取引のみを表示するように選択できます。また、そのエンティティと対応するパートナーの取引を表示するように選択することもできます。たとえば、エンティティとパートナーの選択肢からエンティティAとパートナーBの取引のみを選択すると、エンティティAがパートナーBと持つ取引のみが表示されます。ただし、エンティティBとパートナーAの該当する取引を参照する場合は、エンティティ取引とパートナー取引を含むように選択できます。</p>
DisplayPartnerTransactions	<p>レポートにパートナーの取引を含める場合はTRUEを、含めない場合はFALSEを指定します。</p>
SuppressDetails	<p>取引の詳細を非表示にして、小計行のみを表示する場合はTRUEを指定します。</p>

10

ルールの管理

この項の内容:

ルールのタイプ	227
ルールに関する注意事項	230
連結時のルールの実行	238
デフォルトの換算方法	239
Financial Managementのオブジェクト	240
よく使用するルール	240
Notepad ++でのルールの作成	245
ルール・ファイルの作成	246
ルールのロード	248
ルールの抽出	250

Financial Managementルールを使用して、アプリケーション内のデータの計算を自動化します。ルールは次の目的に使用できます:

- 特定のエンティティ、シナリオおよび期間に関するデータ入力レベルの値を計算する。
- 特定のエンティティ、シナリオ、期間の特定のセルへのデータの入力を防ぐ。
- 親エンティティ・レベルでの入力を可能にする。
- 比率や差異などの、階層間で集約できないデータを計算する。
- 親エンティティから基本エンティティに配賦する。
- 複雑な通貨変換を実行したり、為替レートの差異を計算したり、連結に必要なその他の計算を行ったりする。
- 勘定科目を動的に計算する式を定義する。
- アプリケーションで内部取引をサポートする勘定科目を指定する。

Notepad ++などのテキスト・エディタでルールを記述できます。



注:

ルールをOracle Hyperion Calculation Managerで使用するには、[439ページの第13章「Calculation Managerを使用したルールの作成」](#)を参照してください。

ルールのタイプ

値の設定やクリア、データの計算、通貨の換算、データの連結、別の場所へのデータの配賦を実行するルールや、データの入力を防ぐためのルールを作成できます。

表31 ルールのタイプ

ルールのタイプ	説明
計算	<p>ユーザーが計算を実行するとき、計算ルールが実行されます。計算ルールを使用して、ディメンションの階層の自然な順序では計算できない計算を実行できます。たとえば、比率や期首残高を導出するための計算を作成できます。Calculate()ルーチンは、データの計算時や連結時に実行されます。</p>
換算	<p>ユーザーが換算を実行するとき、換算ルールが実行されます。換算ルールを使用して、標準でない換算に関連する計算を実行できます。Translate()ルーチンは、データの換算時または連結時に実行されます。たとえば、アプリケーションで、資産と負債の勘定科目の換算にはデフォルトのClosingRate、収益と費用の勘定科目の換算にはAverageRateをそれぞれ使用している場合には、純利益に関連した換算差の計算には別の換算レートを使用する必要があります。Financial Managementは、次の場合に換算ルールを実行します。</p> <ul style="list-style-type: none"> • データ・グリッドで右クリックし、表示されたメニューから、「換算」または「換算の強制」を選択して、ユーザーが通貨換算を実行する場合。これは、データの入力時に換算を実行する場合に便利です。 • ユーザーが連結を実行し、親エンティティのデフォルト通貨が子エンティティのそれと異なる場合。
連結	<p>ユーザーが連結を実行するとき、連結ルールが実行されます。連結ルールを使用して、法定アプリケーションで使われることが最も多い、標準ではない連結を実行します。Consolidate()ルーチンは、データの連結時に実行されます。</p>
配賦	<p>配賦ルールは、あるエンティティのデータをエンティティのリストに配賦します。たとえば、配賦ルールを使用して、他のエンティティの総管理支出に基づいたエンティティのリストの管理支出を計算できます。</p>
入力	<p>入力ルールは、親エンティティ・レベルでの入力を可能にします。エンティティ通貨の値ディメンションのみがサポートされます。子から親エンティティのエンティティ通貨値ディメンションへのコントリビューション値はロールアップされません。Parent.Childの組合せのコントリビューション値は保管されます。</p>
NoInput	<p>NoInputルールは、基本エンティティ・レベルでの入力を不可能にします。これにより、計算対象セルが予約されます。NoInputルールでNoInput関数を複数回使用して、連続していない複数のセルへのデータ入力を禁止できます。</p>

ルールのタイプ	説明
	<p>NoInputルールには、次の制約事項とガイドラインが適用されます。</p> <ul style="list-style-type: none"> • 次のFinancial Management関数のみをNoInputルールでサポートします。 <ul style="list-style-type: none"> ○ 349ページのNoInput ○ 344ページのList <p>注:</p> <p>NoInputルールでは、すべてのVBScriptメソッドがサポートされています。</p> <ul style="list-style-type: none"> • List関数の場合、固定リスト、システム・リストおよび動的リストをサポートします。動的リストは、List関数がサポートするメタデータ属性を参照できます。 • If...Then構造は、メタデータ属性をテストできます。ただし、NoInputルールは、現在の視点のメンバーに対するテストをサポートしません。たとえば、If HS.Entity.Member = "CT"をテストできません。 • NoInput関数をループで使用する場合は注意する必要があります。単純な文でもループで使用すると、何千ものセルがメモリーにロードされる可能性があるため、NoInputを含むループではパフォーマンスへの影響を確認する必要があります。
動的計算	<p>動的ルールを使用して式を定義し、勘定科目を動的に計算できます。基本勘定科目のみを動的に計算できます。親の勘定科目には動的ルールを使用できません。動的計算ルールを作成するときは、次のことに注意してください。</p> <ul style="list-style-type: none"> • 等式の右側は、同じシナリオ/年/エンティティの組合せを参照する必要があります。これは、計算で前年度の値を参照できないことを意味します。 • 式の左側では、動的勘定科目のみが有効です。 • 動的勘定科目は、式の右側では使用できません。 • 式の左側で使用できるのは、勘定科目とビューのみです。 • ビューが指定されていない場合、計算はYTDおよび期別に対して実行します。ビューが指定されている場合、計算は、指定されたビューに対してのみ実行します。 • HS.View.PeriodNumberは、HS.Dynamic計算で使用できる唯一のHS文です。 • Sub Dynamicセクションの文はすべて、シーケンシャルに実行されます。

ルールのタイプ	説明
取引	取引ルールは、内部取引をサポートするアプリケーション内の勘定科目を指定します。取引をサポートするセルは、データ・グリッドとデータ・フォームで読取り専用です。
エクイティ・ピックアップ	エクイティ・ピックアップ・ルールは、所有エンティティ、所有者エンティティおよび出資比率を指定します。これは、Sub EquityPickupセクションの実行時のデフォルトの視点です： <ul style="list-style-type: none"> • 現在のシナリオ、年および期間 • エンティティ: 処理されるペアの所有者 • 値: <Entity Currency>
OnDemand	オンデマンド・ルールはデータ・フォームで使用され、計算のサブセットを実行して、結果をデータ・フォームで迅速に確認できます。Sub Calculate (他にはなし)で使用できるすべてのHS関数は、オンデマンド・ルールでも使用できます。

Microsoft VBScript言語に基づいた一意のスクリプトでFinancial Managementのルールを作成します。ルールは、スクリプト構文を生成するための関数、オブジェクトおよびその他の引数の組合せを使用して構築されます。各ルーチン内で、2つのタイプの関数を使用してルールを記述します。

- 計算を定義するFinancial Managementの関数
- 条件付き文を作成するために使用されるVBScript関数

ルールに関する注意事項

次は、アプリケーションのルールを記述する際の注意事項です。

- ルールは、ユーザーが計算、換算、連結および配賦を行うときに実行されます。計算ルールは、計算または連結の適用先であるエンティティ/値ディメンションの交差ごとに一度実行されます。[231ページの計算コマンドを含む計算ルール](#)を参照してください。
- Financial Managementが計算を適用するディメンションのメンバーは、ユーザーのカーソルが置かれたデータ・グリッド・セルと視点バーで指定されたメンバーに依存します。[235ページの現在のディメンション・メンバー](#)を参照してください。
- 関数が通貨に関連する値メンバーにデータを挿入すると、Financial Managementが、値メンバー内の現在の値を削除する場合があります。[237ページの関数によるデータの自動クリア](#)を参照してください。
- ルールは、ルーチン内の連続した順序で実行され、ルーチンの実行順序は設定されています。[238ページの連結時のルールの実行](#)を参照してください。

計算コマンド

計算ルーチンは、シナリオ、年、期間、エンティティおよび値の指定された交差に対して実行されます。計算ルーチンを実行すると、シナリオ、年、期間、エンティティおよび値の現在の交差に対応する現在のデータ表にデータが書き込

まれるかクリアされます。ユーザーが計算ルーチンを実行すると、アプリケーション内の任意の場所からデータが読み取られます。ただし、データは現在のデータ表のみに書き込まれます。



注:

連結時に階層に[None]エンティティがあると、計算ルーチンは失敗します。

次のコマンドのいずれかを選択して、データ・グリッドでこれらのプロセスを実行します:

- 計算
- 計算の強制
- コントリビューションの計算
- コントリビューション計算の強制
- 連結
- データのあるものすべてを連結
- すべて連結
- 換算
- 換算の強制

計算コマンドを選択すると、Financial Managementにより、RLEファイルでSub Calculate()ルーチンが実行されます。Calculate()ルーチンは、指定されたシナリオ、年および期間内の指定されたエンティティおよび値の組合せに対して、勘定科目とカスタム・ディメンションのメンバーを計算します。

計算の強制

計算の強制オプションは、選択された値メンバーおよびその値メンバーが依存するすべての値メンバーのみに強制的にルールを適用します。たとえば、エンティティ通貨に対する計算の強制はエンティティ通貨メンバーのみにルールを適用します。エンティティ通貨の合計に対する計算の強制は、エンティティ通貨、エンティティ通貨の調整およびエンティティ通貨の合計にルールを適用します。計算の強制は、値の3つ組からなる単一メンバーのみに影響を与えることが可能な唯一のコマンドです。

計算コマンドを含む計算ルール

計算ルールが実行される回数は、ユーザーおよびその他の要素によって選択される計算コマンドによって異なります。



注意

ルールを記述する前にこれらのセクションを注意してお読みください。値ディメンションの特定のメンバーにのみ起こる操作が目的である場合は、操作を実行する前にVBScriptのIf構造とMember関数を使用して現在のメンバーに対してテストを行う必要があります。If構造とMember関数の使用に関する手順および例については、[242ページの条件付きのルール](#)を参照してください。

ユーザーがCalculateまたはForce Calculateコマンドを選択すると、Financial Managementでは、エンティティのデフォルト通貨(Entity Currency)の値メンバーと現在のエンティティ・メンバーの交差に対してアプリケーションの計算ルールが実行されます。エンティティのAllowAdjs属性がメタデータで使用可能である場合、Financial Managementでは、2度目のルールが実行され、エンティティのデフォルト通貨の調整(Entity Curr Adjs)を保管する値ディメンションのメンバーとエンティティの交差に対してルールが適用されます。

例

たとえば、Californiaという名前のエンティティのデフォルト通貨がUSDであり、そのAllowAdjs属性がメタデータで使用可能になっている場合、1度目はCaliforniaとUSDの交差に対して、2度目はCaliforniaとUSD Adjsの交差に対してというように、計算ルールは2度実行されます。

連結コマンドを使用した計算ルール

ユーザーが連結コマンドのいずれかを選択すると、Financial Managementでは、まだ連結されていない各子エンティティを交差するいくつかの値ディメンションに対して計算ルールが実行されます。子に対して実行したら、Financial Managementでは、親のデフォルト通貨の値ディメンションのメンバーと親エンティティの交差に対してルールが実行されます。

1. ルールは、子エンティティのデフォルト通貨(Entity Currency)を保管する値メンバーと子エンティティの交差に対して実行されます。
2. 子エンティティのAllowAdjs属性がYに設定されている場合、ルールは、子エンティティのデフォルト通貨の調整(Entity Curr Adjs)を保管する値メンバーと子エンティティの交差に対して実行されます。
3. 子と親で通貨が異なる場合、ルールは、親の母国通貨(Parent Currency)に換算された値を保管する値メンバーと子エンティティの交差に対して実行されます。
4. 子と親で通貨が異なり、ルール・ファイルが換算ルールを含む場合、Financial Managementは、手順3の前に換算ルールを実行します。
5. 子と親で通貨が異なり、子エンティティのAllowAdjs属性がYに設定されている場合、ルールは親の母国通貨に換算された調整(Parent Curr Adjs)を保管する値メンバーと子エンティティの交差に対して実行されます。
6. 親エンティティのAllowAdjFromChildren属性がYに設定されている場合、ロジック・ルールは、子エンティティとParent Adjs値の交差に対して実行されます。
7. ルールは、子エンティティとProportion値の交差に対して実行されます。
8. ルールは、子エンティティとElimination値の交差に対して実行されます。
9. 親エンティティのAllowAdjFromChildren属性がYに設定されている場合、ルールは、子エンティティとContribution Adjs値の交差に対して実行されます。
10. まだ連結されていないデータを含むその他の各子エンティティに対して、手順1-9を繰り返します。
11. ルールは、親エンティティのデフォルト通貨を保管する値メンバーと親エンティティの交差に対して実行されま

例

たとえば、UnitedStatesという名前の親エンティティに、EastUSという名前とWestUSという名前の子がいるとします。子はAllowAdjs属性を使用可能にしています。UnitedStatesエンティティは、AllowAdjs属性と

AllowAdjFromChildren属性を使用可能にしています。3つのすべてのエンティティは、デフォルト通貨USDを共有します。

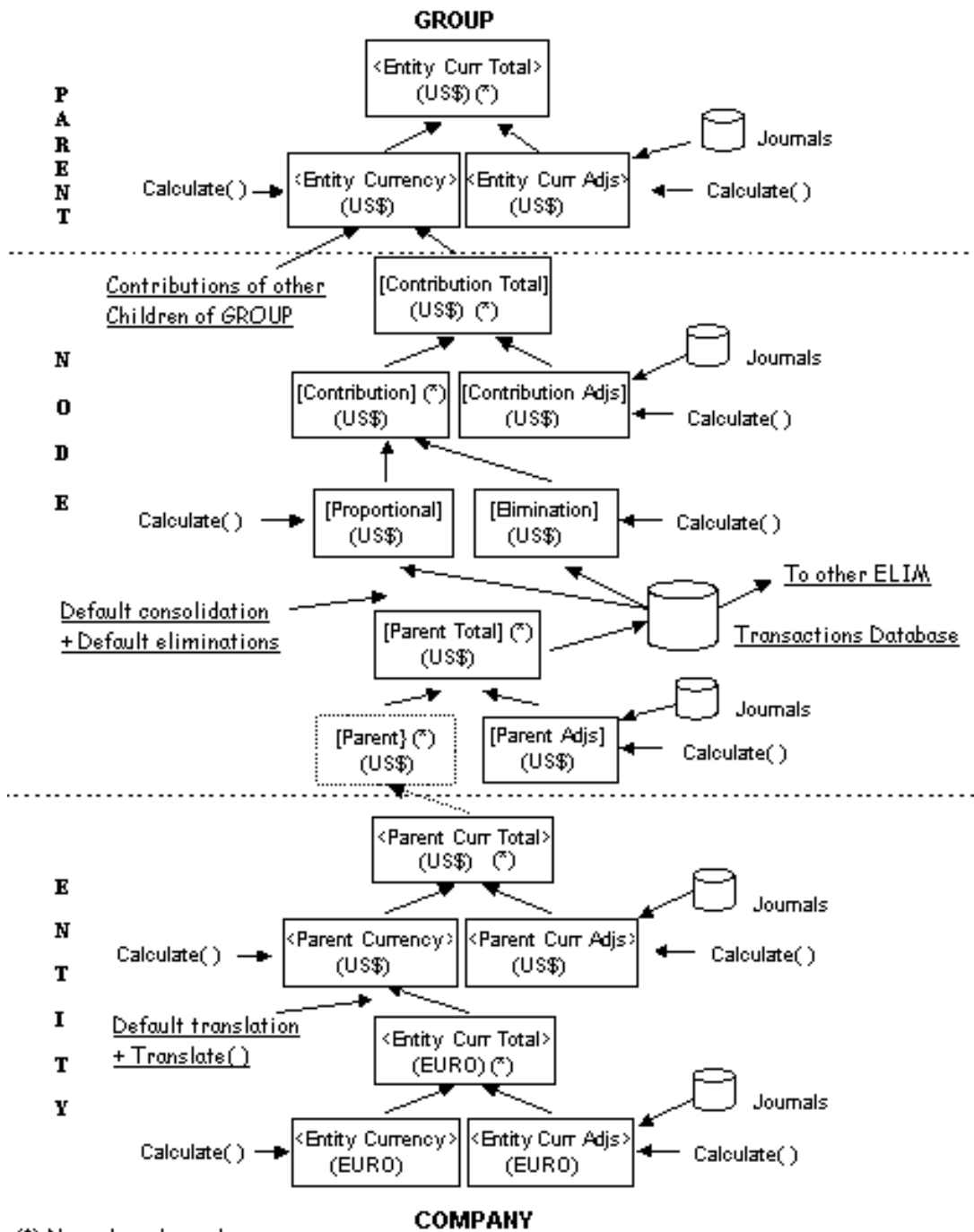
この場合、EastUSとWestUSのデータを変更し、United Statesを連結すると、Financial Managementでは、エンティティ・ディメンションと値ディメンションにおける次の各交差に対してルールが実行されます。

1. EastUSおよびEntity Currency
2. EastUSおよびEntity Currency Adjs(EastUSのAllowAdjs属性はYに設定されています)。
3. EastUSおよびParent。
4. EastUSおよびParent Adjs(UnitedStatesのAllowAdjFromChildren属性はYに設定されています)。
5. EastUSおよびProportion。
6. EastUSおよびElimination。
7. EastUSおよびContribution Adjs(UnitedStatesのAllowAdjFromChildren属性はYに設定されています)。
8. WestUSおよびEntity Currency
9. WestUSおよびEntity Currency Adjs(WestUSのAllowAdjs属性はYに設定されています)。
10. WestUSおよびParent。
11. WestUSおよびParent Adjs(UnitedStatesのAllowAdjFromChildren属性はYに設定されています)。
12. WestUSおよびProportion。
13. WestUSおよびElimination。
14. WestUSおよびContribution Adjs(UnitedStatesのAllowAdjFromChildren属性はYに設定されています)。
15. UnitedStatesおよびEntity Currency

次に連結プロセスの例を示します。

1つ目の例は、エンティティの通貨および親通貨が異なる場合のプロセスを示しています。

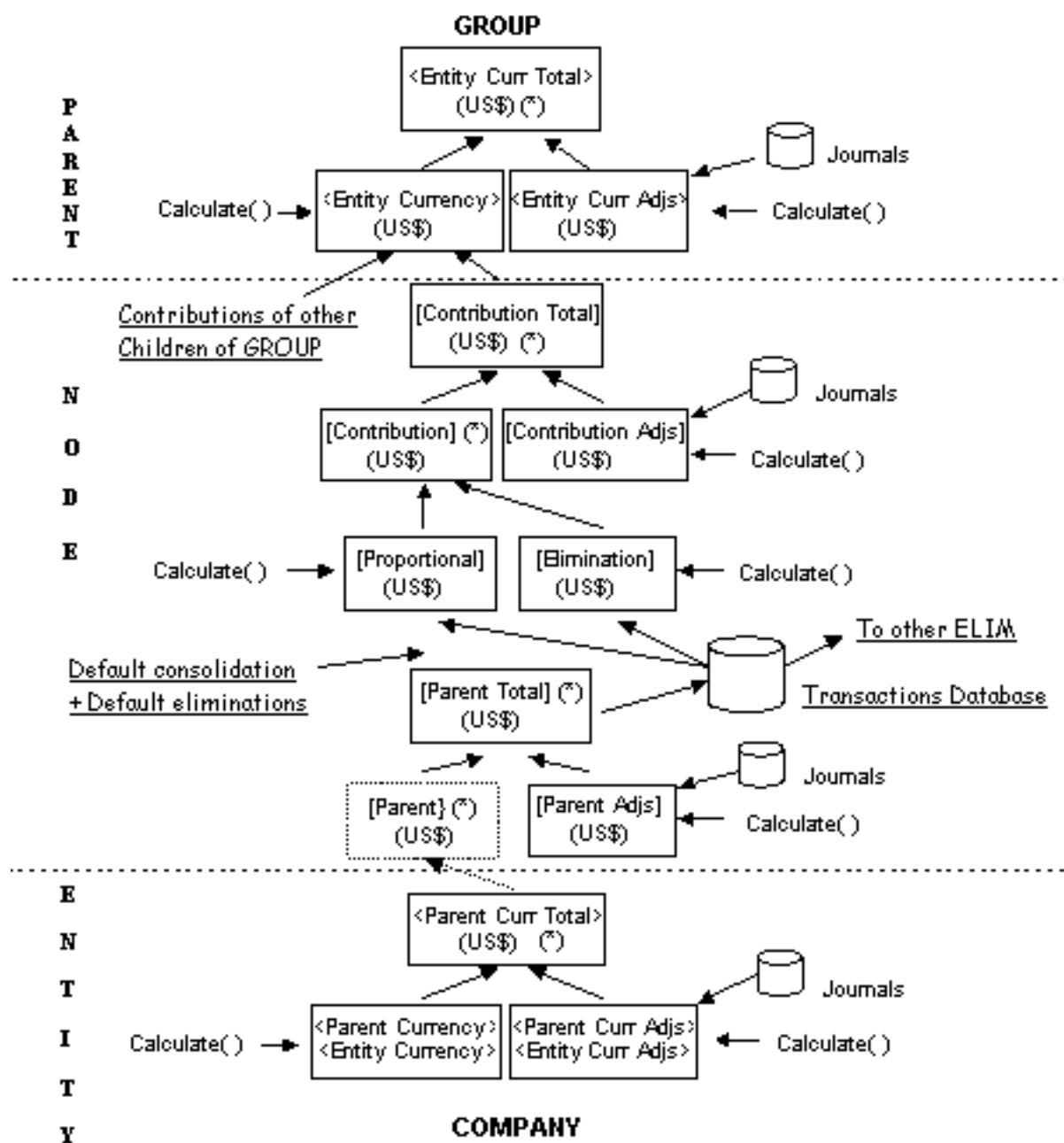
Consolidation Process
(Entity Currency and Parent Currency are Different)



(*) Non-stored members

次の例は、エンティティの通貨と親通貨が同じ場合のプロセスを示しています。

Consolidation Process (Entity Currency and Parent Currency are the Same)



(*) Non-stored members

現在のディメンション・メンバー

デフォルトで、Financial Managementは、ユーザーが計算コマンドを選択するときに、現在のディメンション・メンバーに計算ルールを適用します。231ページの計算コマンドを含む計算ルールを参照してください。現在のディメンション・メンバーは次の要因によって決定されます。

- データ・グリッドで、ユーザーがデータを計算、換算または連結するときにカーソルを置いているセル。
- 視点バーで指定したディメンションのメンバー。
- 連結の場合、エンティティ・ディメンションと値ディメンションの現在のメンバーは、ルールを実行するたびに変更されます。[232ページの連結コマンドを使用した計算ルール](#)を参照してください。
- ルールは、現在選択されているディメンションのメンバーに対して適用されます。ただし、計算ルールは例外で、アプリケーションのすべての勘定科目において適用されます。

カーソルが置かれたセルのディメンション・メンバーが視点バー内の対応するディメンションの対応するメンバーと異なる場合、セルのディメンション・メンバーは、視点バーのディメンション・メンバーを上書きします。たとえば、ユーザーが視点バー内でEuropeという名前のエンティティを指定し、カーソルをGermanyという名前のエンティティのセル内に配置した状態で換算を実行すると、Germanyが現在のエンティティになります。

ディメンション・メンバーを引数に指定すると、現在のディメンションとは関係なく、いくつかの関数を使用して、特定のディメンション・メンバーを操作できます。たとえば、Entityオブジェクトを指定してNumBase関数を使用すると、引数で指定するエンティティの基本エンティティの数を取得できます。

さらに複雑な関数の場合は、勘定科目式を作成して、ディメンションのメンバーを指定できます。

勘定科目式

一部の関数は、勘定科目式を引数として必要です。最も簡単なフォームの勘定科目式は、Financial Managementによって実行される関数の適用先である勘定科目を指定する文字列です。

次の表は、勘定科目式の文字を示したものです。

表32 勘定科目式でディメンションを指定するために使用される文字

文字	ディメンション
A#	勘定科目
I#	内部取引パートナー
C#	カスタム
S#	シナリオ
Y#	年
P#	期間
W#	ビュー
E#	エンティティ
V#	値

勘定科目の式を理解するために、関数の引数に指定されたディメンション・メンバーの値を除去するClear関数について検討します。この例では、引数「A#Sales」は勘定科目式です：

HS.Clear

"A#Sales"

A#文字は、勘定科目ディメンションを表し、単語Salesは、Financial Managementによって実行されるClear関数の適用先である勘定科目ディメンションのメンバー名です。この勘定科目式は、Financial Managementに、Sales勘定科目に保管されたデータをクリアするよう指示します。

勘定科目の式を使用するときは、勘定科目の式で指定した勘定科目と、次に示すディメンションの現在のメンバーの交差に、Financial Managementによって関数が適用されます。

- エンティティ
- 期間
- シナリオ
- 値
- ビュー
- 年
- カスタム - メタデータの勘定科目に設定されたCustomTopMemberを使用します。たとえば、勘定科目の式がカスタム3ディメンションのメンバーを指定していない場合、Financial Managementでは、勘定科目に指定されたCustomTopMemberによって定義されたとおりに、すべての有効なカスタム3ディメンションのメンバーが使用されます。
- 内部取引パートナー - すべての有効なICPメンバー。

内部取引とカスタム・ディメンションのデフォルトを上書きするには、勘定科目の式にメンバーを指定します。各ディメンションは、特定の文字で表されます。2つ以上のディメンションを勘定科目の式に含める場合は、ディメンションをピリオドで区切る必要があります。

勘定科目の式を作成する場合、これらのディメンション・メンバーのすべてを指定する必要はありません。関数を適用するメンバーを指定します。たとえば、この行は、Sales勘定科目とHardware ICPの交差をクリアします。

HS.Clear "A#Sales.I#Hardware"

関数によるデータの自動クリア

関数によって通貨に関連する値メンバーにデータが挿入されると、Financial Managementでは、次のいずれかの条件に該当する場合、データがメンバーから自動的にクリアされます。

- 交差するエンティティ・メンバーが親である場合
- 交差する勘定科目メンバーが計算済勘定科目である場合



ヒント:

通貨関連の値メンバーは、システム生成のEntity Currencyメンバーと、USD、EURO、LIRAなどのユーザー定義通貨メンバーです。

また、関数により調整用データが値メンバーに挿入されると、Financial Managementでは、交差する勘定科目メンバーが計算済勘定科目である場合に、このメンバーからのデータが自動的にクリアされます。



ヒント:

調整に関連する値メンバーは、名前にコード「Adjs」が含まれるメンバーです。たとえば、USD Adjs、EURO Adjs、LIRA Adjsなどです。

エラー・メッセージ

Financial Managementが構文エラーを検出すると、次の情報を含むエラー・メッセージを表示します。

- エラーの原因となるRLEファイル内の行番号
- 該当するFinancial Managementのオブジェクトおよび関数
- エラーの説明

連結時のルールの実行

連結プロセスでは、ルールは定義済シーケンスで実行されます。特定の親の基本の子それぞれに対して、値ディメンション内の様々な要素の計算シーケンスが次の順序で適用されます:

1. メタデータでIsCalculatedとして定義された勘定科目がEntity Currencyでクリアされます。
2. メタデータでIsCalculatedとして定義された勘定科目がEntityCurrAdjsでクリアされます。
3. Sub Calculate()ルーチンがEntity Currencyで実行されます。
4. Sub Calculate()ルーチンがEntityCurrAdjsで実行されます。
5. ParentCurrencyデータがクリアされます。
6. EntityCurrencyおよびEntityCurrAdjsの合計値に対して、デフォルトの換算が、収益、費用、資産、負債として定義されたすべての勘定科目に適用されます。フロー属性または残高属性を持つ勘定科目の場合、デフォルトでは取引が適用されず、EntityCurrencyおよびEntityCurrAdjsの合計値は、親通貨にロール・アップされません。
7. Sub Translate()ルーチンが実行されます。
8. Sub Calculate()ルーチンがParentCurrencyで実行されます。
9. メタデータで「IsCalculated」として定義された勘定科目がParentCurrAdjsでクリアされます。
10. Sub Calculate()ルーチンがParentCurrAdjsで実行されます。
11. メタデータで「IsCalculated」として定義された勘定科目がParentAdjsでクリアされます。
12. Sub Calculate()ルーチンがParentAdjsで実行されます。
13. ProportionおよびEliminationのデータがクリアされます。
14. ParentおよびParentAdjsの合計値に対して、デフォルトの連結と消去が実行されます。
15. Sub Calculate()ルーチンがProportionおよびEliminationで実行されます。

16. メタデータで「IsCalculated」として定義された勘定科目がContributionAdjsでクリアされます。

17. Sub Calculate()ルーチンがContributionAdjsで実行されます。

上記の手順が各基本子に対して繰り返された後、次のシーケンスが親エンティティに適用されます。

1. Entity Currencyデータがクリアされます。
2. 各子のProportion、EliminationおよびContributionAdjsの合計が親エンティティのEntityCurrencyに書き込まれます。
3. Sub Calculate()ルーチンがEntity Currencyで実行されます。
4. メタデータで「IsCalculated」として定義された勘定科目がEntityCurrAdjsでクリアされます。
5. Sub Calculate()ルーチンがEntityCurrAdjsで実行されます。



注:

親がさらに別の親に連結される場合、このシーケンスは子の連結シーケンスの手順5から継続されます。

デフォルトの換算方法

デフォルトの換算は、次の順序で実行されます。

1. システムにより、現在のエンティティで直接換算レートがチェックされ、そのレートが換算に使用されます。
2. 直接換算レートが見つからなかった場合は、現在のエンティティの間接レートから直接レートが計算されます。
3. 現在のエンティティで直接レートと間接レートのどちらも見つからない場合、[None]エンティティが検索され、直接レートが使用されます。
4. [None]エンティティの直接レートが見つからない場合は、[None]エンティティの間接レートから直接レートが計算されます。
5. [None]エンティティの間接レートが存在しない場合、[None]エンティティのアプリケーション通貨を使用した三角換算法によってレートが計算されます。
6. 三角換算法が失敗すると、エンティティは換算されません。



注:

最初に現在のエンティティ内の換算レートが検索されます。現在のエンティティで見つからなければ、[None]エンティティ内の換算レートが検索されます。[None]エンティティで見つからなければ、三角換算法を使用して換算されます。三角換算法は、3番目の共通通貨を使用して、残高をある通貨から別の通貨に変換する方法です。

たとえば、EUROからYENへの変換で、その換算を実行するための直接レートまたは間接レートが見つからない場合や、EUROとYENの両方をUSDに換算できる場合は、三角換算法を使用することでEUROの残高をUSDに変換してから、USDの残高をYENに変換します。

Financial Managementのオブジェクト

このセクションでは、関数の使用時にFinancial Managementのオブジェクトを表すために使用する必要がある構文について説明します。

Financial Managementの最上位レベルのオブジェクトは、HSオブジェクトです。これは、Financial Managementの関数を使用するとき、最初の3つの文字を「HS.」にする必要があることを意味します。

```
HS.
```

この例では、Clear関数を記述する方法を示します。

```
HS.Clear
```

Clearの前に「HS.」の文字を付けないと、エラーが発生します。



注意

標準のVBScript関数の前に「HS.」の文字は使用できません。そのようにした場合、エラーが発生します。「HS.」の文字は、Financial Managementの関数の前のみを使用します。

Financial Managementの一部の関数は、HSオブジェクトの子であるオブジェクトのみに適用されます。これらの関数では、HSの文字と関数名の間に該当するオブジェクトの名前を入れる必要があります。これらのオブジェクトは、HSオブジェクトの子であり、Financial Managementで使用されます。

- Account
- AppSettings
- Custom
- DataUnit
- Entity
- ICP
- Node
- Parent
- Period
- Scenario
- Value
- Year

よく使用するルール

次のセクションでは、よく使用される単純ないくつかのルールを記述する方法について示します。これらのルールの説明には、VBScriptを使い慣れていない読者のために、ステップごとの手順が含まれています。これらの手順の後には例が続きます。VBScriptに習熟している読者は、手順をスキップして例に進んでください。



ヒント:

これらの手順では、作成したルール・ファイルとコードがCalculate()サブルーチンに置かれているものとします。[246ページのルール・ファイルの作成](#)を参照してください。

データの再使用

EXP関数を使用して、ある勘定科目のデータを別の勘定科目に挿入します。EXPの引数は、設定される勘定科目と値の取得元の勘定科目を含みます。引数は等式の形式にし、等号の左側にはターゲットの勘定科目を指定し、右側にはソースの勘定科目を指定します。



注:

EXPは、現在のディメンション・メンバーを含む勘定科目の交差にデータを挿入します ([235ページの現在のディメンション・メンバー](#)を参照)。さらに、勘定科目式の文字を使用して、現在のカスタム・メンバーとICPメンバーを上書きできます。

この例では、計算ルールによって、PrevCash勘定科目がCash勘定科目の値に設定されます。

```
HS.EXP "A#PrevCash = A#Cash"
```

勘定科目式の文字を使用して、EXPの引数で、等号の両側にディメンションのメンバーを指定できます。[297ページのExp](#)を参照してください。この例は、PrevCash勘定科目とカスタム3ディメンションのGolfメンバーの前年度の交差のデータを、PrevCashとGolfの当年度の交差に挿入します。

```
HS.EXP "A#PrevCash.C3#Golf = A#Cash.Y#Prior.C3#Golf"
```



ヒント:

勘定科目式の文字Y#の後に続くPriorキーワードによって、EXPは、前年度のデータを取得できます。勘定科目の式には、年や期間に適用される同様のキーワードが複数あります。[299ページの期間および年のキーワード](#)を参照してください。

値の計算による勘定科目の設定

もう1つの共通タスクは、2つの勘定科目に含まれる値を計算し、結果を別の勘定科目に挿入することです。EXP関数では、その引数で、等号の右側に加算、減算、乗算および除算を置くことができます。

この計算ルールの例では、Sales勘定科目の値をUnitsSold勘定科目の値で割り、その結果をAvgPrice勘定科目に挿入します。

```
HS.EXP "A#AvgPrice = A#Sales / A#UnitsSold"
```

条件付きのルール

特定のディメンションのメンバーが視点の現在のメンバーである場合にのみアクションを実行するルールにします。たとえば、Actual(実績)が現在のシナリオである場合にはある方法で勘定科目の値を計算し、Budget(予算)が現在のシナリオである場合には別の方法で勘定科目の値を計算します。



ヒント:

Financial Managementによって現在のディメンション・メンバーをどのように判別するかについては、[235ページの現在のディメンション・メンバー](#)を参照してください。

これを行うには、VBScriptのIf構造でMember関数を使用します。Member関数は、次のディメンションの現在のメンバーの名前を取得します。

- ・ エンティティ(現在のエンティティを取得するにはEntityオブジェクトを、現在のエンティティの親を取得するにはParentオブジェクトをそれぞれ使用)
- ・ 期間
- ・ シナリオ
- ・ 値
- ・ 年

If構造を使用すると、特定の条件がTRUEである場合にのみ文を実行できます。次のセクションでは、MemberをIf構造で使用してディメンションのメンバーをテストする別の方法をいくつか示します。



ヒント:

これらのセクションでは、If構造の側面のみをいくつか取り上げています。If構造の詳細は、MicrosoftのVBScriptのドキュメントを参照してください。VBScriptのドキュメントは、MicrosoftのWebサイトからダウンロードできます。

ディメンションのメンバーのテスト

特定のディメンションのメンバーが現在のメンバーである場合にのみFinancial Managementでアクションを行うには、Member関数の戻り値をテストするIf構造を使用します。

この例では、現在のシナリオがBudget(予算)である場合に、Financial ManagementによってUnitsSoldとPriceの勘定科目の値が乗算され、結果がSales勘定科目に挿入されます。

```
If HS.Scenario.Member = "Budget" Then
```

```
HS.EXP "A#Sales = A#UnitsSold * A#Price"
```

```
End If
```



ヒント:

すべてのIf構造は、If...Then文で始まり、End If文で終わる必要があります。条件を満たすとき実行されるアクションは、上記のように、If...Then文とEnd If文の間にあります。

複数のメンバーのテスト

If...Then文では、複数のメンバーをテストできます。つまり、1つのディメンションで2つ以上のメンバーのアクションを実行できます。「ディメンションのメンバーのテスト」セクションにある例について考えます。現在のシナリオがBudget(予算)またはForecast(予測)の場合は、Sales勘定科目の値を計算します。

複数のメンバーをテストするには、If...Then文で2つのMember関数とVBScriptのOrキーワードを使用します。Orを最初のMember関数の後に置き、OrとThenの間に2番目のMemberを置きます。



ヒント:

この技法を使用して、複数のメンバーをテストできます。テストするメンバーごとに、Member関数とOrキーワードの組合せを含めます。

この例では、現在のシナリオがBudget(予算)またはForecast(予測)である場合に、Financial ManagementによってUnitsSoldとPriceの勘定科目の値が乗算され、結果がSales勘定科目に挿入されます。

```
If HS.Scenario.Member = "Budget" Or HS.Scenario.Member = "Forecast" Then
```

```
HS.EXP "A#Sales = A#UnitsSold * A#Price"
```

```
End If
```

メンバーごとに異なるアクションを実行

ディメンションのメンバーごとに異なるアクションを実行するルールを持つことができます。たとえば、現在のシナリオがBudget(予算)のときはある計算をし、Actual(実績)のときは別の計算をするルールを持つことができます。

異なるアクションを条件に従って実行するには、If構造に1つ以上のElseIf文を含む必要があります。メンバーごとにそのElseIf文のテストを行います。ElseIf文の直下にそのメンバーのために実行するアクションを配置します。

この例では、現在のシナリオがBudget(予算)またはActual(実績)のどちらであるかに応じて、異なる勘定科目が更新されます。

- 現在のシナリオがBudget(予算)の場合は、Financial ManagementによってUnitsSoldとPriceの勘定科目の値が乗算され、結果がSales勘定科目に挿入されます。
- 現在のシナリオがActual(実績)の場合は、Financial ManagementによってSales勘定科目の値がUnitsSold勘定科目の値で除算され、その結果がPrice勘定科目に挿入されます。

```
If HS.Scenario.Member = "Budget" Then
```

```
HS.EXP "A#Sales = A#UnitsSold * A#Price"
```

```
ElseIf HS.Scenario.Member = "Actual" Then
```

```
HS.EXP "A#Price = A#Sales / A#UnitsSold"
```

```
End If
```



ヒント:

If...ThenとElseIfの文で指定した条件が一切満たされない場合に別のアクションを行うには、VBScriptを使用してElse文をIf構造に追加します。詳細は、MicrosoftのVBScriptのドキュメントを参照してください。

すべての勘定科目の期首残高の設定

勘定科目の期首残高を設定するには、If構造でExpおよびIsFirst関数を使用します。ExpおよびIf構造は前のセクションで紹介されています。IsFirstは、現在の期間が現在のシナリオのデフォルト期間単位で最初の期間であるかどうかを判別します。たとえば、シナリオに月次のデフォルト期間単位がある場合、IsFirstは、現在の期間が年の最初の月であるかどうかを判別します。

期首残高を設定するには、If構造のIf...Then文にIsFirstを挿入し、この文とEnd If文の間にExpを挿入します。特定の勘定科目をExpの引数に含めることができるかぎり、すべての勘定科目の期首残高を設定できます。

この例では、すべての勘定科目の期首残高の設定方法を示します。この例を再入力するか、変更せずに計算ルールにコピーできます。

HS.EXP "A#ALL = A#ALL.P#Prior"



注:

この例では、A#の後にキーワードALLが続きます。これは、ルールがすべての勘定科目に適用されることを示します。P#文字の後にはキーワードPriorが続きます。これは、EXPが、現在の期間より前の期間から勘定科目データを取得することを示します。

Notepad ++でのルールの作成

Notepad ++は、無料の強力なテキスト・エディタで、ユーザー定義の構文を、カスタマイズ可能なカラー・スキームでハイライトして表示します。Notepad ++を使用して、Financial Managementのルールおよびメンバー・リストを編集できます。

Notepad ++の起動

Notepad++は、Financial Management Utilitiesフォルダにインストールされます。

▶ Notepad ++を起動するには:

1. Financial Managementをインストールしたディレクトリの「Utilities」フォルダに移動します。
2. **Notepad ++**でルール/メンバー・リストを編集を選択します。

Notepad ++の構成

ルール・ファイルの様々な部分の色やフォントを構成できます。これは、ルール・ファイルの各部分をすばやく識別するために便利です。たとえば、Financial ManagementのオブジェクトとメソッドをVisual Basicのオブジェクトとは異なる色に設定できます。

ルール・ファイルの次の部分の色とフォントを変更できます。

- テキスト
- テキスト選択
- 数値
- 演算子
- コメント
- 文字列
- 文
- Financial Managementのメソッド
- Visual Basicのオブジェクト
- プロパティ

- イベント
- 関数
- 定数
- Financial Managementのオブジェクト

▶ Notepad ++を構成するには:

1. Notepad ++を起動します。
2. 「**View**」、「**User-Defined Dialogue**」の順に選択します。
3. 「**Import**」をクリックし、Financial Managementをインストールした「Utilities」フォルダから**HFM Rules.xml**を選択します。
4. Notepad ++を終了します。

次回Notepad ++を起動すると、「言語」メニューの下に、ユーザー定義言語としてHFMルールが表示されます。この機能を使用して、ルール・ファイルの作成や、既存のルール・ファイルの編集を行うことができます。

ルール・ファイルの作成

Notepad ++などのテキスト・エディタでルールを作成できます。ルール・ファイルは、マルチバイト文字セット(MBCS)対応のASCIIフォーマットを使用するか、Little Endianバイト・オーダーによってUnicodeフォーマットにエンコードされたファイルを使用して作成できます。ルール・ファイルでは、Visual Basic Scriptの関数とFinancial Managementの関数を使用します。デフォルトでは、ルール・ファイルにはRLEファイル拡張子が使用されます。

Financial Managementのルールのすべてのタイプを任意の順序でルール・ファイルに含めることができます。246ページの表 33は、各ルーチンを定義するための基本的な構文を示します。

表33 ルール・ルーチン

ルール・ルーチン	構文
Sub Calculate	Sub Calculate() 'ここに、計算ルールを入力します。 End Sub
Sub Translate	Sub Translate() 'ここに、換算ルールを入力します。 End Sub
Sub Allocate	Sub Allocate() 'ここに、配賦ルールを入力します。 End Sub
Sub Input	Sub Input() 'ここに、入力ルールを入力します。 End Sub

ルール・ルーチン	構文
Sub NoInput	<pre>Sub NoInput() 'ここに、NoInputルールを入力します。 End Sub</pre>
Sub Consolidate	<pre>Sub Consolidate() 'ここに、連結ルールを入力します。 End Sub</pre>
Sub Dynamic	<pre>Sub Dynamic() 'ここに、動的ルールを入力します。 End Sub</pre>
Sub Transactions	<pre>Sub Transactions() 'ここに、取引ルールを入力します。 End Sub</pre>
Sub EquityPickup	<pre>Sub EquityPickUp() 'ここに、エクイティ・ピックアップ・ルールを入力します。 End Sub</pre>
Sub OnDemand	<pre>Sub OnDemand_<ruleName> 'ここに、オンデマンド・ルールを入力します。 End Sub</pre>

エクイティ・ピックアップ・ルールの例

次のセクションでは、資本ピックアップ・ルールのサンプル・セクションを示します。エクイティ・ピックアップを計算するために、アプリケーション管理者は、Sub EquityPickUpという名前のルール・ファイルに新しいセクションを作成する必要があり、ここで、EPUの計算が定義されます。セクションを実行するときのデフォルトの視点は次のとおりです。

- 現在のシナリオ、年および期間
- エンティティ: 処理されるペアの所有者
- 値: エンティティ通貨

```
Sub EquityPickUp()
```

```
Owner = Hs.Entity.Member
```

```
Owned = Hs.Entity.Owned
```

```
OwnerDefaultCurrency = HS.Entity.DefCurrency("")
```

```
lPown = Hs.GetCell("E#" & Owned & ".I#" & Owner & ".V#[None].A#[Shares  
%Owned].C1#[None].C2#[None].C3#[None].C4#[None]")
```

```
Hs.Clear "A#IncomeFromSubsidiary.I#" & Owned
```

```
Hs.Exp "A#IncomeFromSubsidiary.I#" & Owned & " = E#" & Owned & ".V#" &  
OwnerDefaultCurrency & ".A#NetIncome.I#[ICP Top] *" & lPown
```

```
End Sub
```

オンデマンド・ルールの例

データ・フォームからオンデマンド・ルールを実行するために、管理者は、Sub OnDemandという名前のルール・ファイルに新しいセクションを作成する必要があり、ここで、オンデマンド・ルールが定義されます。

```
Sub OnDemand_Calculation
```

```
HS.Exp "A#CogsTP=15424"
```

```
HS.Exp "A#Admex=32452"
```

```
End Sub
```

Sub Calculate (他にはなし)で使用できるすべてのHS関数はオンデマンド・ルールでも使用できることに注意してください。Sub Calculateとは異なり、以前にIsCalculatedデータ・ポイントに書き込まれたデータは、オンデマンド・ルールの実行時にクリアされないことにも注意してください。

ルールのロード

ルール変更はデータに反映され、メタデータに依存します。そのため、ルールのロード・プロセスにより、Financial Managementサーバー・クラスタにグローバル・ロックが適用されます。ルール・ロードは、事前に開始した次のタイプの操作が終了するまでは処理されません:

- 連結
- データ入力
- データ、仕訳またはセキュリティ・ロード
- データベースへのデータの抽出
- メンバー・リスト・ロード
- メタデータ・ロード

ルール・スキャン・プロセスには、ディメンション・メンバーとその他のパラメータが検証されるよう、ロード・プロセスと同じ制約が適用されます。ルール・スキャンとロードのプロセスはキューに入れられ、ブロック・タスクの終了後に自動的に開始されます。ルールは長時間にわたる連結中などではなく、小規模のアクティビティ実行中にロードすることをお勧めします。「タスクの実行」ページでは、連結ロードまたはデータ・ロードなど、どのプロセスが処理中であるかを確認できます。

アプリケーションにルール・ファイルがロードされた後、そのアプリケーションを使用しているユーザーに対し、システムが変更されたのでアプリケーションからログオフして再びログオンする必要があることが通知されます。

ルール・ファイルに内部取引が含まれている場合、ルール・ファイル内の新規取引に対してアプリケーションの転記済取引を確認できます。ルール・ファイルのSub Transactionsセクションは、内部取引の詳細をサポートする勘定科目を定義します。



注:

Webのポップアップ・ブロックの例外にFinancial Managementを追加することをお勧めします。データのロードなどのいくつかのタスクを実行すると、ステータス・ウィンドウにタスクのステータスが表示されます。コンピュータでポップアップ・ブロックを使用可能にしている場合、このステータス・ウィンドウは表示されません。

▶ ルールをロードするには:

1. 「連結」、「ロード」、アプリケーション要素の順に選択します。
2. 「ルール」セクションにロードするファイル名を入力するか、「参照」をクリックしてファイルを検索します。



注:

デフォルトでは、ルール・ファイルにはRLEファイル拡張子が使用されます。ロード・プロセスでは、TXTやCSVなどのファイル拡張子を使用できますが、RLEファイル拡張子の使用をお勧めします。

3. オプション: 転記済の内部取引が、ロードするルール・ファイルのSub Transactionsのセクションにある文と整合性があるかどうかを確認するには、「整合性チェック」を選択します。
4. オプション: ファイル・フォーマットが正しいかどうかを確認するには、「スキャン」をクリックします。
5. 「ロード」をクリックします。



ヒント:

オプションをデフォルト値にリセットするには、「リセット」をクリックします。

ルールの抽出

ルールを抽出すると、そのルールは、マルチバイト文字セット(MBCS)対応のASCIIファイルに保存されます。デフォルトでは、ルール・ファイルにはRLEファイル拡張子を使用されます。ルールを抽出したら、そのルールをテキスト・エディタで表示および変更できます。

▶ ルールを抽出するには:

1. 「連結」、「抽出」、アプリケーション要素の順に選択します。
2. 「ルール」セクションで、「抽出」をクリックします。
3. ブラウザに表示されるダウンロードの手順に従います。

使用しているWebブラウザによって、手順が異なります。ファイルは、設定したWebディレクトリに保存してください。

11

ルール関数

この項の内容:

関数の概要	254
ABSExp	261
AccountType	262
AccountTypeID	264
AddEntityToList	265
AddEntityToListUsingIDs	266
AddMemberToList	266
AddMemberToListUsingIDs	267
Alloc	267
AllowAdjFromChildren	269
AllowAdjs	271
ApplicationName	272
CalculateExchangeRate	272
CalculateRate	273
CalcStatus	274
CellTextUnitItem	276
Clear	278
Con	279
Consol1, Consol2, Consol3	281
ConsolidateYTD	282
ContainsCellText	283
ContainsCellTextWithLabel	284
Currency	285
CustomTop	286
DataUnitItem	287
Decimal	288
DefaultFreq	289
DefaultParent	290
DefaultTranslate	291
DefaultView	292
DefCurrency	293
Down	295
Dynamic	296
Exp	297
GetCell	304
GetCellNoData	305
GetCellRealData	307

GetCellText	308
GetCellTextWithLabel	308
GetCellType	309
GetCustomLabelArray	311
GetItem	311
GetItemIDs2	313
GetItemIDs2ExtDim	314
GetNumItems	315
GetNumLID	316
GetRate	317
GetSubmissionGroup	318
GetSubmissionPhase	318
Holding	319
ICPTopMember	320
ICPWeight	321
IDFromMember	322
ImpactStatus	324
Input	325
IsAlmostEqual	326
IsBase	327
IsCalculated	329
IsChild	330
IsConsolidated	333
IsDescendant	334
IsFirst	336
IsICP	337
IsLast	339
IsTransCur	340
IsTransCurAdj	340
IsValidDest	341
IsZero	342
List	344
Member	346
MemberFromID	347
Method	348
NoInput	349
NoRound	350
NumBase	351
Number	354
NumChild	354
NumCustom	357
NumDescendant	357
NumPerInGen	360
NumPeriods	361
OpenCellTextUnit	362
OpenDataUnit	363
OpenDataUnitSorted	364
Owned	364

Owner	365
PCon	365
PEPU	367
PeriodNumber	368
PlugAcct	369
POwn	370
PVAForBalance	371
PVAForFlow	372
RateForBalance	373
RateForFlow	373
ReviewStatus	374
ReviewStatusUsingPhaseID	376
Round	376
Scale	378
SecurityAsPartner	379
SecurityClass	380
SetCellTextWithLabel	382
SetData	383
SetDataWithPOV	384
SubmissionGroup	385
SupportsProcessManagement	386
SupportsTran	387
SwitchSign	388
SwitchType	390
Trans	391
TransPeriodic	392
UD1...3	393
ValidationAccount	395
ValidationAccountEx	396
XBRLTags	397

これらの関数を使用して、次のタイプのルールを作成できます。

- 配賦ルール
- 計算ルール
- 連結ルール
- 動的計算ルール
- エクイティ・ピックアップ・ルール
- 入力ルール
- NoInputルール
- オンデマンド・ルール
- 取引ルール
- 換算ルール

関数の中には、特定のルーチンに固有のものと、複数のルーチン内で複数のタイプのルールに使用できるものがあります。

すべての関数の概要については、[254ページの関数の概要](#)を参照してください。

関数の概要

[254ページの表 34](#)では、Financial Managementの関数、その関数で使用できるオブジェクトおよびルールのタイプの概要を示します。関数は、アルファベット順に表示されます。各関数の詳細なセクションが表の後に追加されています。



注:

11.1.2.2より前のFinancial Managementリリースから移行されたレガシー・アプリケーションは、"Custom1...4"オブジェクトを使用できます。11.1.2.2で作成されたすべてのアプリケーションは、新しい"Custom(Alias)"オブジェクト構文を使用して、カスタム別名/短い説明/長い説明を使用して参照するカスタム・ディメンションを指定します。

ルール・タイプはこの表では次のように省略されています。

- Alloc - 配賦
- Calc - 計算
- Con - 連結
- Dyn - 動的計算
- EPU - エクイティ・ピックアップ
- Tran - 換算
- Trans - 取引

表34 Financial Managementの関数

関数	説明	オブジェクト	ルールのタイプ
261ページのABSExp	計算式を実行し、結果を絶対値として保管します。	HS	Calc, Tran, Alloc
262ページのAccountType	メンバーの勘定科目タイプを取得します。	Account	Calc, Tran, Con, Alloc
264ページのAccountTypeID	メンバーの勘定科目の種類別IDを取得します。	Account	Calc, Tran, Con, Alloc
265ページのAddEntityToList	指定されたエンティティと親をメンバー・リストに追加します。	HS	メンバー・リスト
266ページのAddEntityToListUsingIDs	エンティティと親IDを使用して、指定されたエンティティと親を内部リストに追加します。	HS	メンバー・リスト
266ページのAddMemberToList	メンバーをメンバー・リストに追加します。	HS	メンバー・リスト

関数	説明	オブジェクト	ルールのタイプ
267ページのAddMemberToListUsingID	UsingIDを使用して、指定されたメンバーを内部リストに追加します。	HS	メンバー・リスト
267ページのAlloc	データをセルに割り当てます。	HS	Alloc
269ページのAllowAdjFromChild	このメンバーの子からの仕訳の転記が可能かどうかを調べます。	Entity, Parent	Calc, Tran, Con, Alloc
271ページのAllowAdjs	このメンバーの仕訳の転記が可能かどうかを調べます。	Entity, Parent	Calc, Tran, Con, Alloc
272ページのApplicationName	ルールが実行されているアプリケーションの名前を戻します。	AppSettings	Calc, Tran, Con, Alloc
272ページのCalculateExchangeRate	ある通貨から別の通貨への為替レートを計算します。	HS	Calc, Tran, Con, Alloc, Dyn, Trans
273ページのCalculateRate	指定されたエンティティの現在の為替レートを取得します。	HS	Calc, Tran, Con, Alloc, Dyn, Trans
274ページのCalcStatus	セルの算出ステータスを取得します。	HS	Calc, Tran, Con, Alloc
278ページのClear	データをセルから除去します。	HS	Calc, Tran, Con, Alloc
279ページのCon	[Proportion]および[Elimination]勘定科目にデータを挿入します。	HS	Con
281ページのConsol1、Consol2、Consol3	Consol1-3システム勘定科目の値を取得します。	Node	Calc, Tran, Con, Alloc
282ページのConsolidateYTD	シナリオが累計メソッドまたは期別メソッドのいずれを使用して連結されるかを調べます。	Scenario	Calc, Tran, Con, Alloc
283ページのContainsCellText	セルにセル・テキストが入っているかどうかを調べます。	HS	Calc, Tran, Con, Alloc
284ページのContainsCellText	指定されたラベルのセル・テキストがセルに入っているかどうかを調べます。	HS	Calc, Tran, Con, Alloc

関数	説明	オブジェクト	ルールのタイプ
285ページのCurrency	アプリケーションまたは値メンバーの通貨のタイプを取得します。	AppSettings、Value	Calc、Tran、Con、Alloc
286ページのCustomTop	勘定科目のCustomTopMemberを取得します。	Account	Calc、Tran、Con、Alloc
287ページの水DataUnitItem	連結、計算または換算時に処理するデータ単位アイテムを取得します。	HS	Calc、Tran、Con
288ページのDecimal	指定した勘定科目の小数点以下の桁数を取得します。	Account	Calc、Tran、Con、Alloc
289ページのDefaultFreq	デフォルトのシナリオの期間単位を取得します。	Scenario	Calc、Tran、Con、Alloc
290ページのDefaultParent	メンバーのデフォルトの親を取得します。	Account	Calc、Tran、Con、Alloc
291ページのDefaultTranslate	ルールを使用せずに換算を計算します。	HS	Tran
292ページのDefaultView	デフォルトのシナリオのビューを取得します。	Scenario	Calc、Tran、Con、Alloc
293ページのDefCurrency	エンティティまたは親のデフォルト通貨を取得します。	Entity、Parent	Calc、Tran、Con、Alloc
295ページのDOWn	ノードの直接出資(DOWn)比率を取得します。	Node	Calc、Tran、Con、Alloc
296ページのDynamic	計算が必要な動的勘定科目の式を指定します。	HS	Dyn
297ページのExp	計算式を実行し、データを指定された視点に挿入します。	HS	Calc、Tran、Alloc
304ページのGetCell	セルのデータを取得します。	HS	Calc、Tran、Con、Alloc
305ページのGetCellNoData	セルに含まれているデータを取得し、セルにデータがない場合にはその旨を示す。	HS	Calc、Tran、Con、Alloc
307ページのGetCellRealData	セルに含まれているデータを取得し、セルに実際のデータが含まれているかどうかを示します。	HS	Calc、Tran、Con、Alloc

関数	説明	オブジェクト	ルールのタイプ
308ページのGetCellText	指定された視点のデフォルトのセル・テキストを取得します。	HS	Calc, Tran, Con, Alloc
308ページのGetCellTextWithLabel	指定したセル・テキスト・ラベルの視点からセル・テキストを取得します。	HS	Calc, Tran, Con, Alloc
309ページのGetCellType	セル・タイプを取得します。	HS	Calc, Tran, Con, Alloc
311ページのGetCustomLabelNames	カスタム・ディメンション名および別名のリストを戻します。	HS	Calc, Tran, Con, Alloc
311ページのGetItem	連結する個々のレコードを取得します。	DataUnit	Calc, Tran, Con
313ページのGetItemIDs2	ディメンションのID番号を使用して、連結のプロセスを行う個々の記録を取得します。	DataUnit	Calc, Tran, Con
315ページのGetNumItems	連結するレコードの数を取得します。	DataUnit	Calc, Tran, Con
316ページのGetNumLID	指定されたPOVのライン・アイテムの詳細の数を取得します。	HS	Calc, Tran, Con, Alloc
317ページのGetRate	セルの換算レートを取得します。	HS	Calc, Tran, Con, Alloc
318ページのGetSubmissionGroups	セルの提出グループを取得します。	HS	Calc, Tran, Con, Alloc
318ページのGetSubmissionPhases	セルの提出フェーズを取得します。	HS	Calc, Tran, Con, Alloc
319ページのHolding	メンバーの持株会社を取得します。	Entity, Parent	Calc, Tran, Con, Alloc
320ページのICPTopMember	現在の勘定科目ディメンション・メンバー、または指定した勘定科目メンバーのICPTopMemberを取得します。	Account	Calc, Tran, Con, Alloc
321ページのICPWeight	[ICP Top]の値のメンバーに集約されるICPエンティティ・バランスの比率を取得します。	AppSettings	Calc, Tran, Con, Alloc
322ページのIDFromMember	指定したメンバーのID番号を取得します。	Account, Custom1-Custom4, Entity, ICP, Parent, Year, Period, Scenario, Value, V	Calc, Tran, Con, Alloc

関数	説明	オブジェクト	ルールのタイプ
324ページのImpactStatus	指定したデータ単位のステータスを「影響済」に変更します。	HS	Calc
325ページのInput	指定した勘定科目について、親のエンティティレベルで入力できるようにします。	HS	Input
326ページのIsAlmostEqual	2つの値が等しいかどうかをチェックします。	HS	Calc, Tran, Con, Alloc, Dyn, Trans
327ページのIsBase	メンバーが基底メンバーであるかどうかを調べます。	Account, Custom1-Custom4, Entity, Parent, Node	Calc, Tran, Con, Alloc
329ページのIsCalculated	勘定科目が計算されているかどうかを調べます。	Account	Calc, Tran, Con, Alloc
330ページのIsChild	メンバーが別のメンバーの子であるかどうかを調べます。	Account, Custom1-Custom4, Entity, Parent, Node	Calc, Tran, Con, Alloc
333ページのIsConsolidated	勘定科目が連結されているかどうかを調べます。	Account	Calc, Tran, Con, Alloc
334ページのIsDescendant	メンバーが別のメンバーの子孫であるかどうかを調べます。	Account, Custom1-Custom4, Entity, Parent, Node	Calc, Tran, Con, Alloc
336ページのIsFirst	期間または年がアプリケーションの最初の期間または年であるかどうかを調べます。	Period, Year	Calc, Tran, Con, Alloc
337ページのIsICP	メンバーがICPであるかどうかを調べます。	Account, Entity, Parent	Calc, Tran, Con, Alloc
339ページのIsLast	年または期間がアプリケーションの最後の年または期間であるかどうかを調べます。	Period, Year	Calc, Tran, Con, Alloc
340ページのIsTransCur	値メンバーが換算された通貨メンバーであるかどうかを調べます。	Value	Calc
340ページのIsTransCurAdj	値メンバーが換算された通貨調整メンバーであるかどうかを調べます。	Value	Calc
341ページのIsValidDest	指定したPOVが有効な宛先かどうかを調べます。	HS	Calc, Tran, Con, Alloc

関数	説明	オブジェクト	ルールのタイプ
342ページのIsZero	渡された値がゼロかどうかをチェックします。	HS	Calc, Tran, Con, Alloc, Dyn, Trans
344ページのList	リストに含まれている要素を取得します。	Account, Custom1-Custom4, Entity, Parent, ICP, Node, Scenario	Calc, Tran, Con, Alloc
346ページのMember	メンバー名を取得します。	Entity, Parent, Period, Scenario, Value, Year, View	Calc, Tran, Con, Alloc
347ページのMemberFromID	指定したID番号のメンバーを取得します。	Account, Custom1-Custom4, Entity, ICP, Parent, Year, Period, Scenario, Value, View	Calc, Tran, Con, Alloc
348ページのMethod	メンバーの連結メソッドを取得します。	Node	Calc, Tran, Con, Alloc
349ページのNoInput	セルへの入力を禁止します。	HS	NoInput
350ページのNoRound	丸め処理をオフにします。	HS	Calc, Tran, Con, Alloc
351ページのNumBase	基底メンバーの数を取得します。	Account, Custom1-Custom4, Entity, Parent, Node, Period	Calc, Tran, Con, Alloc
354ページのNumber	現在の期間の番号を取得します。	Period	Calc, Tran, Con, Alloc
354ページのNumChild	メンバーの子の数を取得します。	Account, Custom1-Custom4, Entity, Parent, Node	Calc, Tran, Con, Alloc
357ページのNumCustom	アプリケーションに定義されているカスタム・ディメンションの数を取得します。	HS	Calc, Tran, Con, Alloc
357ページのNumDescendant	メンバーの子孫の数を取得します。	Account, Custom1-Custom4, Entity, Parent, Node	Calc, Tran, Con, Alloc
360ページのNumPerInGen	処理中の現在の期間に対する世代の期間の数を取得します。	Period	Dynamic
361ページのNumPeriods	指定したシナリオの期間単位に対して定義されている期間の数を取得します。	Scenario	Dynamic
362ページのOpenCellTextUnit	複数のセルのセル・テキストを戻します。	HS	Calc, Tran, Con
363ページのOpenDataUnit	連結のデータ単位を取得します。	HS	Calc, Tran, Con
364ページのOpenDataUnitSort	計算、換算または連結のデータ単位をソート順に取得します。	HS	Calc, Tran, Con
364ページのOwned	現在処理されているペアの所有エンティティを取得します。	エンティティ	エクイティ・ピックアップ

関数	説明	オブジェクト	ルールのタイプ
365ページのOwner	現在処理されているペアの所有者エンティティを取得します。	エンティティ	エクイティ・ピックアップ
365ページのPCon	連結比率を取得します。	Node	Calc、Tran、Con、Alloc
367ページのPEPU	EPU表から出資比率を取得します。	HS	エクイティ・ピックアップ
368ページのPeriodNumber	取得中のデータのビューの期間番号を取得します。	View	Dynamic
369ページのPlugAcct	調整勘定科目を取得します。	Account	Calc、Tran、Con、Alloc
370ページのPOwn	出資比率を取得します。	Node	Calc、Tran、Con、Alloc
371ページのPVAFForBalance	BALANCE勘定科目のデフォルトの換算メソッドを調べます。	AppSettings	Calc、Tran、Con、Alloc
372ページのPVAFForFlow	FLOW勘定科目のデフォルトの換算メソッドを調べます。	AppSettings	Calc、Tran、Con、Alloc
373ページのRateForBalance	BALANCEのデフォルト・レートを取得します。	AppSettings	Calc、Tran、Con、Alloc
373ページのRateForFlow	FLOWのデフォルト・レートを取得します。	AppSettings	Calc、Tran、Con、Alloc
374ページのReviewStatus	セルのプロセス管理の承認ステータスを取得します。	HS	Calc、Tran、Con、Alloc
376ページのReviewStatusUsingPOV	セルのPOV IDごとにプロセス管理の承認ステータスを取得します。	HS	Calc、Tran、Con、Alloc
376ページのRound	データを丸めます。	HS	Calc、Tran、Con、Alloc
378ページのScale	指定した通貨の位取りを取得します。	Currency	Calc、Tran、Con、Alloc
379ページのSecurityAsPartner	ICPエンティティのセキュリティ・クラスを取得します。	Entity、Parent	Calc、Tran、Con、Alloc
380ページのSecurityClass	ディメンションのメンバーのセキュリティ・クラスを取得します。	Account、Scenario、Entity、Parent、Con、Alloc、Custom4	Calc、Tran、Con、Alloc
382ページのSetCellTextWithLabel	指定された視点(POV)とセル・テキスト・ラベルのセル・テキストにテキスト文字列を書き込みます。	HS	Calc
383ページのSetData	個々のレコードを設定します。	HS	Calc、Tran

関数	説明	オブジェクト	ルールのタイプ
384ページのSetDataWithPOV	データをノードまたは通貨キューブに挿入します。	HS	Calc, Tran
385ページのSubmissionGroup	ディメンション・メンバーのプロセス管理の提出グループを取得します。	HS	Calc, Tran, Con, Alloc
386ページのSupportsProcessManagement	メンバーがプロセス管理をサポートしているかどうかを調べます。	Scenario	Calc, Tran, Con, Alloc
387ページのSupportsTran	アプリケーションで、内部取引詳細のサポートが必要な勘定科目を定義します。	Scenario, Year, Entity, Account	Trans, C1...4
388ページのSwitchSign	メンバーの貸方が借方に切り替えられているかどうかを調べます。	Custom1-Custom4	Calc, Tran, Con, Alloc
390ページのSwitchType	メンバーの勘定科目の種類が切り替えられるかどうかを調べます。	Custom1-Custom4	Calc, Tran, Con, Alloc
391ページのTrans	YTDメソッドを使用して換算します。	HS	Tran
392ページのTransPeriodic	期別メソッドを使用して換算します。	HS	Tran
393ページのUD1...3	メンバーのユーザー定義属性を取得します。	Account, Entity, Parent, Scenario, Custom1-Custom4	Calc, Tran, Con, Alloc
395ページのValidationAccount	検証勘定科目を取得します。	AppSettings	Calc, Tran, Con, Alloc
396ページのValidationAccount	プロセス管理の提出フェーズの検証勘定科目を取得します。	AppSettings	Calc, Tran, Con, Alloc
397ページのXBRLTags	勘定科目のXBRLタグを取得します。	Account	Calc, Tran, Con, Alloc

ABSExp

計算式を実行し、結果を絶対値として保管します。この関数は、結果を絶対値として保管する以外はExp関数と同じです。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 配賦

構文

```
HS.ABSExp "  
  DestPOV  
  =  
  Expression  
  "
```

表35 ABSExp関数の構文

パラメータ	説明
<i>DestPOV</i>	データの配置先を識別する宛先視点。勘定科目のメンバーを必ず指定し、オプションでICPおよびカスタム・メンバーを指定します。使用に関する次のルールに注意してください: <ul style="list-style-type: none">• ICPメンバーを指定しないと、デフォルトで[ICP None]が使用されます。• データベースに不要な値を移入することを避けるには、データを配置する場所の定義について、ルールをできるだけ明確にする必要があります。適切な方法は、勘定科目ディメンションに、明確に定義されたカスタム・ディメンションの交差を含めることです。この明確に定義された交差では、Financial Managementの検証チェックを使用して、無効な交差にデータが書き込まれないようにします。299ページのディメンションの交差に関する考慮事項を参照してください。
<i>Expression</i>	計算式。

戻り値

なし。

例

この例では、StateTax勘定科目の金額を設定します。例では、2014のSales勘定科目の値と2014のStateRate勘定科目のレートを乗算して、絶対値を計算します。

```
HS.ABSExp "A#StateTax = A#Sales.Y#2014 * A#StateRate.Y#2014"
```

AccountType

現在の勘定科目のメンバーまたは指定された勘定科目の勘定科目の種別を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Account.AccountType("Account")
```

```
HS.Account.AccountType("")
```

ここで、Accountは、有効な勘定科目のメンバーの名前です。

Sub Consolidateサブルーチンで関数を使用している場合にのみ、空白文字列(" ")を使用して、この関数を現在のメンバーに適用できます。関数を使用していない場合は、この関数の使用時に勘定科目を指定します。

戻り値

指定した勘定科目の勘定科目の種別。



注:

勘定科目の種別は、すべて大文字です。

次のタイプがあります。

- ASSET(資産)
- LIABILITY(負債)
- REVENUE(収益)
- EXPENSE(費用)
- DYNAMIC(動的)
- FLOW(フロー)
- BALANCE(残高)
- BALANCERECURRING(経常残高)
- CURRENCYRATE(通貨レート)
- GROUPLABEL(グループラベル)

例

この例では、Sales勘定の勘定科目の種別がREVENUEのときに、If...ThenとEnd Ifの文の間にあるすべての文が実行されます。

```
If HS.Account.AccountType("Sales") = "REVENUE" Then
```

```
...
```

```
End If
```

AccountTypeID

現在の勘定科目のメンバーまたは指定された勘定科目の勘定科目の種別IDを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦
- 動的計算
- 取引

構文

```
HS.Account.AccountTypeID("Account")
```

```
HS.Account.AccountTypeID("")
```

*Account*は、有効な勘定科目メンバーの名前です。

Sub Consolidateサブルーチンで関数を使用している場合にのみ、空白文字列("")を使用して、この関数を現在のメンバーに適用できます。関数を使用していない場合は、この関数の使用時に勘定科目を指定する必要があります。

戻り値

指定された勘定科目のIDです。264ページの表 36は、対応するIDの有効な勘定科目の種別を示します。

表36 勘定科目の種別ID

勘定科目の種別	ID
REVENUE(以前のリリースではINCOME)	0

勘定科目の種別	ID
EXPENSE(費用)	1
ASSET(資産)	2
LIABILITY(負債)	3
BALANCE(残高)	4
FLOW(フロー)	5
CURRENCYRATE(通貨レート)	7
GROUPLABEL(グループラベル)	10
BALANCERECURRING(経常残高)	11
DYNAMIC(動的)	12

例

```
If HS.Account.AccountTypeID("Investments") = 2 Then
```

```
...
```

```
End If
```

AddEntityToList

指定されたメンバーをメンバー・リストに追加します。この関数は、メンバー・リスト・ファイルでのみ使用できます。

構文

```
HS.AddEntityToList("
  Parent
  ", "
  Child
")
```

表37 AddEntityToList関数の構文

パラメータ	説明
Parent	有効な親エンティティの名前。
Child	親エンティティの子の名前。

戻り値

なし

例

```
HS.AddEntityToList "UnitedStates", "Maine"
```

AddEntityToListUsingIDs

エンティティおよび親のIDを使用して、指定されたメンバーをメンバー・リストに追加します。この関数は、メンバー・リスト・ファイルでのみ使用できます。

構文

```
HS.AddEntityToListUsingIDs(  
  ParentID  
  ,  
  ChildID  
)
```

表38 AddEntityToListUsingIDs関数の構文

パラメータ	説明
ParentID	親エンティティのID。
ChildID	親エンティティの子のID。

戻り値

なし

AddMemberToList

指定されたメンバーをメンバー・リストに追加します。この関数は、メンバー・リスト・ファイルでのみ使用できます。

構文

```
HS.AddMemberToList("  
  Member  
")
```

Memberは、有効なディメンション・メンバーの名前です。

戻り値

なし

例

```
HS.AddMemberToList "July"
```

AddMemberToListUsingIDs

指定されたメンバーをメンバー・リストに追加します。この関数は、メンバー・リスト・ファイルでのみ使用できます。

構文

```
HS.AddMemberToListUsingIDs(  
  MemberID  
)
```

MemberIDは、有効なディメンション・メンバーのIDです。

戻り値

なし

Alloc

別の視点にデータを配賦します。この関数は、配賦ルールで使用できます。

構文

```
HS.Alloc ("   
  SourcePOV  
  " , "  
  DestPOV  
  " , "  
  EntityList
```

```

" , "
AllocExp
" , "
PlugAccount
")

```

表39 Alloc関数の構文

パラメータ	説明
SourcePOV	<p>配賦されるデータのソース視点。勘定科目のメンバーを必ず指定し、オプションでICPおよびカスタム・メンバーを指定します。ICPおよびカスタム・メンバーを指定しない場合は、次のようになります。</p> <ul style="list-style-type: none"> • デフォルトのICPメンバーが、[ICP Top]になります。 • デフォルトのカスタム・メンバーはその勘定科目のTopMemberです。
DestPOV	<p>データの配賦先を識別する宛先視点。勘定科目のメンバーを必ず指定し、オプションでICPおよびカスタム・メンバーを指定します。使用に関する次のルールに注意してください:</p> <ul style="list-style-type: none"> • ICPメンバーを指定しないと、デフォルトで[ICP None]が使用されます。 • カスタム・メンバーを指定しないと、デフォルトで[None]が使用されます。 • エンティティのメンバーを指定し、EntityList引数を[Base]にした場合は、指定したメンバーが親とみなされます。
EntityList	<p>データの配賦先エンティティを識別するメンバー・リスト。システム定義の[Base]エンティティ・リストを使用するか、ユーザー定義のリストを使用できます。</p> <p>[Base]システム定義のリストを使用する場合は、宛先視点に親メンバーとして指定されたエンティティが使用されません。ユーザー定義のリストを使用し、そのリストに親メンバーが含まれる場合、親メンバーは無視されます。</p>
AllocExp	<p>各エンティティに配賦されるデータを識別する式。この式は、次のタイプの値を含むことができます。</p> <ul style="list-style-type: none"> • 数値 • 数値を識別する勘定科目の式。勘定科目のメンバーは必ず指定し、オプションでICPおよびカスタム・ディメンションのメンバーを指定できます。使用に関する次のルールに注意してください: <ul style="list-style-type: none"> - カスタム・メンバーを指定しない場合、デフォルトはTopMemberです。

パラメータ	説明
	<ul style="list-style-type: none"> - ICPメンバーを指定しない場合、デフォルトは[ICP Top]です。 - シナリオ、年、期間、ビューまたは値のメンバーを指定しない場合、デフォルトは現在になります。 - エンティティ・メンバーを指定しない場合、デフォルトは宛先エンティティになります。
PlugAccount	<p>調整勘定科目の名前。この引数はオプションです。ソース視点の勘定科目と調整勘定の属性を比較して、必要に応じて、ソース視点にある勘定の金額の符号を逆にします。勘定科目のメンバーを必ず指定し、オプションでICPおよびカスタム・メンバーを指定します。使用に関する次のルールに注意してください:</p> <ul style="list-style-type: none"> • ソース視点のエンティティ・メンバーが親の場合、またはソース視点の値メンバーがEntity Currencyでない場合、この引数は使用できません。 • ICPのメンバーを指定しないと、デフォルトで[ICP None]が使用されます。 • カスタム・メンバーを指定しないと、デフォルトで[None]が使用されます。

戻り値

なし。

例

この例では、TangibleAssets勘定科目からのデータは現金勘定科目に配賦されます。

```
Call HS.ALLOC
("A#TangibleAssets", "A#Cash", "NewEngland", "A#TangibleAssets", "A#Plug")
```

AllowAdjFromChildren

指定されているエンティティまたは親メンバーで、子からの仕訳の転記が可能であるかどうかを指定します。この関数は、次のタイプのルールで使用できます:

- 計算

- ・ 換算
- ・ 連結
- ・ 配賦



注:

メンバーがツリー階層で別のメンバーのすぐ1つ下のレベルにある場合、そのメンバーは子です。

構文

```
HS.Entity.AllowAdjFromChildren("
Entity
")
```

```
HS.Entity.AllowAdjFromChildren(" ")
```

```
HS.Parent.AllowAdjFromChildren("
Entity
")
```

```
HS.Parent.AllowAdjFromChildren(" ")
```

Entityは、有効なエンティティまたは親メンバーの名前です。

この関数を現在のエンティティまたは親に適用するには、空白の文字列(" ")を使用します。

戻り値

指定されたエンティティの子からの仕訳の転記が可能な場合にTRUEとなり、不可能な場合にFALSEとなるブール式。

例

この例では、Franceというエンティティの子メンバーの仕訳を転記できる場合に、If...ThenとEnd Ifの間にあるすべての文が実行されます。

```
If HS.Entity.AllowAdjFromChildren("France") = TRUE then
```

```
...
```

```
End If
```

AllowAdjs

指定されているエンティティまたは親メンバーで、仕訳の転記が可能かどうかを指定します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Entity.AllowAdjs("
Entity
")
```

```
HS.Entity.AllowAdjs(" ")
```

```
HS.Parent.AllowAdjs("
Entity
")
```

```
HS.Parent.AllowAdjs(" ")
```

Entityは、有効なエンティティまたは親メンバーの名前です。

この関数を現在のエンティティまたは親に適用するには、空白の文字列(" ")を使用します。

戻り値

指定されたエンティティの仕訳の転記が可能の場合にTRUEとなり、不可能な場合にFALSEとなるブール式。

例

この例では、Franceというエンティティの仕訳を転記できる場合は、If..ThenとEnd Ifの間にあるすべての文が実行されます。

```
If HS.Entity.AllowAdjs("France") = TRUE then
```

```
...
```

```
End If
```

ApplicationName

ルールが実行されているアプリケーションの名前を戻します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.ApplicationName
```

戻り値

ルールが実行されているアプリケーションの名前。

例

```
strApp=HS.ApplicationName
```

CalculateExchangeRate

ある通貨から別の通貨への為替レートを計算します。

構文


```

HS.CalculateExchangeRate (
  ScenarioID
  ,
  YearID
  ,
  PeriodID
  ,
  EntityID
  ,
  RateAccountID
  ,
  FromCurrencyID
  ,
  ToCurrencyID
)

```

表40 CalculateExchangeRate関数の構文

パラメータ	説明
ScenarioID	シナリオのID。
YearID	年のID。
PeriodID	期間のID。
EntityID	エンティティのID。
RateAccountID	レート勘定のID。
FromCurrencyID	換算元の通貨のID。
ToCurrencyID	換算先の通貨のID。

戻り値

為替レート。

CalculateRate

指定されたエンティティの現在の為替レートを取得します。

構文

```

HS.CalculateRate (
  ScenarioID
  ,
  YearID
  ,
  PeriodID
  ,
  EntityID
  ,

```

```
ValueID  
,  
RateAccountID  
)
```

表41 CalculateRate関数の構文

パラメータ	説明
ScenarioID	シナリオのID。
YearID	年のID。
PeriodID	期間のID。
EntityID	エンティティのID。
ValueID	値のID。
RateAccountID	レート勘定のID。

戻り値

指定されたエンティティの為替レート。

CalcStatus

指定された視点の算出ステータスを取得します。この関数は、次のタイプのルールで使用できます：

- ・ 計算
- ・ 換算
- ・ 連結
- ・ 配賦

構文

```
HS.CalcStatus("  
Scenario.Year.Period.Entity.Value  
")
```

表42 CalcStatus関数の構文

パラメータ	説明
Scenario	有効なシナリオ・メンバーの名前。
Year	有効な年。
Period	有効な期間。

パラメータ	説明
Entity	有効なエンティティ・メンバーの名前。
Value	有効な値メンバーの名前。

戻り値

指定された視点の算出ステータスを示す文字列。有効なステータス・コードを次に示します。

表43 計算ステータス・コード

ステータスのコード	説明
OK	指定したシナリオ、年、期間、エンティティおよび値ディメンションのデータは、変更されていません。
OK ND	OK - データなし。データは計算済ですが、データのないセルは計算に含まれていません。
OK SC	OK - システム変更済。指定したシナリオ、年、期間、エンティティおよび値ディメンションのデータに影響する変更が実行されています。たとえば、新しいルール・ファイル、メタデータ・ファイルまたはメンバー・リスト・ファイルがロードされたか、通貨レートが変更されました。値10,000など、データ自体は変更されていません。ディメンション・メンバー属性などに変更が行われています。たとえば、勘定科目に関連付けられている小数点以下の桁数が2に設定されている場合、値を10,000.00に変更する必要があります。
CH	計算が必要。指定されたシナリオ、年、期間、エンティティおよび値ディメンションの少なくとも1つが変更されているか、メタデータ・パラメータまたはルールが変更されています。その結果、計算が実行されていないので、このディメンションのその他のデータ・セルが最新ではない可能性があります。基本レベルのエンティティの場合は、データ・セルがデータ入力か、データ・ファイルのロードで入力された可能性があります。エンティティの場合は、データ・セルが仕訳の転記で入力された可能性があります。
CH ND	計算が必要 - データなし。これは、セルで計算が実行される初回を示します。
TR	換算が必要。選択された値ディメンションのメンバーは、エンティティのデフォルト通貨ではなく、換算された値が最新のものではない可能性があります。
TR ND	換算が必要 - データなし。これは、セルで換算が実行される初回を示します。
CN	連結が必要。次の変更が行われたため、指定したシナリオ、年、期間、エンティティおよび値ディメンションのデータが最新ではない可能性があります：

ステータスのコード	説明
	<ul style="list-style-type: none"> 子エンティティのデータ 同じエンティティのデフォルト通貨のデータ メタデータ・パラメータまたはルール
CN ND	連結が必要 - データなし。親にデータはありませんが、子エンティティのデータが変更されています。これは、セルに対して最初の連結が行われることを示します。
ロック済	<p>指定したシナリオ、年、期間、エンティティおよび値ディメンションのデータが、管理者によってロックされました。手動で変更することも、計算で変更することもできなくなりました。</p> <p>注:</p> <p>267ページのAlloc関数を使用して、ロックされた宛先POVでデータを変更できます。</p>
NoData	指定したシナリオ、年、期間、エンティティおよび値ディメンションにデータがありません。
NoAccess	ユーザーが、指定したディメンションのメンバーへのアクセス権を持っていません。

例

この例では、指定された視点のステータスが「OK」のときに、If...ThenとEnd Ifの間にあるすべての文が実行されます。

```
If HS.CalcStatus("S#Actual.Y#2014.P#January.E#Connecticut.V#<EntityCurrency>") = "OK" Then
```

...

```
End If
```

CellTextUnitItem

連結、計算または換算時に処理するセル・テキスト単位アイテムを戻します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

構文

```
HS.OpenCellTextUnit("", "[Default]", "Entity", "Ascending")
```

戻り値

The entry for the specified item.

例

```
Set MyCellTextUnit = HS.OpenCellTextUnit("", "[Default]", "Entity", "Ascending")
```

```
NumItems= MyCellTextUnit.GetNumItems
```

```
For i = 0 to NumItems - 1
```

```
Scenario = MyCellTextUnit.Item(i).Scenario
```

```
Year = MyCellTextUnit.Item(i).Year
```

```
Period = MyCellTextUnit.Item(i).Period
```

```
Entity = MyCellTextUnit.Item(i).Entity
```

```
Value = MyCellTextUnit.Item(i).Value
```

```
Account = MyCellTextUnit.Item(i).Account
```

```
ICP = MyCellTextUnit.Item(i).ICP
```

```
Flow = MyCellTextUnit.Item(i)Custom("Flows")
```

```
Nature = MyCellTextUnit.Item(i).Custom("Nature")
```

```
...
```

```
...
```

```
...
```

```
CellText = MyCellTextUnit.Item(i).CellText
```

```
POV = MyCellTextUnit.Item(i).POV
```

```
Next
```

Clear

勘定科目、ICPおよびカスタム・メンバーの組合せからデータを除去します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Clear "  
Account.ICP.Custom1.Custom2.Custom3.Custom4  
"
```

表44 Clear関数の構文

パラメータ	説明
Account	有効な勘定科目メンバーの名前。
ICP	有効なICPメンバーの名前。
Custom1、Custom2、Custom3、Custom4	カスタム1、カスタム2、カスタム3およびカスタム4の有効なメンバーの名前。

現在の視点(現在のエンティティ、期間、シナリオ、値、ビューおよび年のメンバー)と交差するすべてのセルのデータを除去するには、次の例のように、A#に続けてALLキーワードを入力します。

```
HS.Clear "A#ALL"
```

セルおよびカスタムまたはICPのディメンションのすべての交差をクリアするには、ALLキーワードを使用するか、A#文字を省略します。この例では、A#文字を省略して、カスタム1ディメンションのGolfBallsメンバーと交差するすべての勘定科目からデータをクリアします。

```
HS.Clear "C1#GolfBalls"
```

戻り値

なし。

例

この例では、Sales勘定科目がカスタム・ディメンションのGolfBallsメンバーと交差する部分に保管されているデータをクリアします。

```
HS.Clear "A#Sales.C1#GolfBalls"
```

Con

[Proportion]および[Elimination]の値ディメンションのメンバーにデータを入力します。この関数は、連結ルールで使用できます。

構文

```
HS.Con ("
  DestPOV
",
  Factor
, "
  Nature
")
```

表45 Con関数の構文

パラメータ	説明
DestPOV	これらのディメンションの組合せは次のとおりです。

パラメータ	説明
	<ul style="list-style-type: none"> 勘定科目 カスタム1、カスタム2、カスタム3、カスタム4 内部取引 エンティティ 値
Factor	数値、あるいは算術演算子(+ - * /)やHS.GetCellなどの関数を使用した式。
Nature	<p>監査目的に使用する文字列。この文字列はデータベースに保管されて、取引の会計目的に関する情報を提供します。連結の実行後に、連結ソースと宛先取引をユーザーが表示できるようにするには、このパラメータにテキストを含める必要があります。テキストを含めないと、取引情報が保管されません。取引情報は、データ・グリッドのソースや宛先の取引オプション、またはエンティティの詳細レポートから表示できます。また、Nautre文字列の情報を使用して、連結および消去のエントリの仕訳レポートを生成することも可能です。このデータを仕訳レポートで表示する場合、このパラメータは必須です。比率および消去のエントリの仕訳は、RTS/RTD表からレポートされます。要件によっては、[Proportion] HSではなく、[Elimination] HS.Con エントリの仕訳データを作成する方が適切な場合もあります。RTS/RTD表・エントリのボリュームを減らすためのConエントリ。</p> <p>注:</p> <p>RTS/RTD表・エントリを生成するNatureパラメータを使用すると、データベースのサイズが増大し、連結のパフォーマンスに影響する可能性があります。</p>

戻り値

なし。

例

```
Call HS.Con ("V#[Elimination]", -1*dPCon, "")
```

```
Call HS.Con ("V#[Elimination]", -1*dPCon, "Elimination")
```

```
Call HS.Con ("V#[Elimination]"PCON, "DefaultConsolidation")
```


Consol1、Consol2、Consol3

指定されたparent.entityノードのConsol1、Consol2またはConsol3勘定科目から値を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 配賦

構文

シナリオ、年、期間メンバーと、parent.entity形式で指定したエンティティ・メンバーを組み合わせます。

```
HS.Node.Consol  
n  
("S#  
Scenario  
.Y#  
Year  
.P#  
Period  
.E#  
Parent  
.  
Entity  
")
```

```
HS.Node.Consol  
n  
("")
```



注:

この関数を現在のシナリオ、年、期間およびエンティティに適用するには、空白の文字列("")を使用します。

表46 Consol1、Consol2、Consol3の関数の構文

パラメータ	説明
Scenario	有効なシナリオ・メンバーの名前。
Year	有効な年。
Period	有効な期間。
Parent.Entity	Parent.Entity形式の、有効なノードの名前。

戻り値

Consol1、Consol2、Consol3のシステム勘定科目の値。

例

この例では、Actual(実績)シナリオからGroup1.Ent1ノードのConsol1の値を取得します。

```
dVar1 = HS.Node.Consol1 ("S#Actual.E#Group1.Ent1")
```

ConsolidateYTD

現在のシナリオ・メンバーまたは指定されたシナリオ・メンバーが年次累計で連結されるかどうかを調べます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Scenario.ConsolidateYTD("Scenario")
```

```
HS.Scenario.ConsolidateYTD("")
```

「Scenario」は有効なシナリオ・メンバーの名前です。

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

戻り値

シナリオが累計メソッドを使用して連結される場合はTRUEとなり、期別メソッドを使用して連結される場合はFALSEとなるブール式。

例

この例では、Actual(実績)シナリオが年次累計メソッドを使用して連結されるように設定されている場合、If...ThenとEnd Ifの文の間にあるすべての文が実行されます。

```
If HS.Scenario.ConsolidateYTD("Actual") = TRUE Then
```

```
...
```

```
End If
```

ContainsCellText

指定されたセルにセル・テキストが含まれているかどうかを調べます。この関数は、次のタイプのルールで使用できません:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.ContainsCellText("POVExpression")
```

POVExpressionは、メンバーの組合せです。ディメンションを指定していない場合は、次のデフォルト値が使用されます。

- Account - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- Scenario - 現在のシナリオ・メンバー
- Entity - 現在のエンティティ・メンバー
- Value - 現在の値メンバー
- YearおよびPeriod - 現在のメンバー



注:

この関数を計算ルールで使用すると、デフォルト・エントリが適用されます。配賦ルール、換算ルールまたは連結ルールで使用する場合は、完全なサブキューブのAccount/ICP/Custom POVを指定する必要があります。シナリオ、年、期間、ビュー、エンティティおよび値ディメンションは、デフォルトで現在のメンバーに設定されます。

戻り値

指定したデータ・セルに、任意のセル・テキスト・ラベルのセル・テキストが含まれる場合はTRUEになるブール式; 含まれない場合はFALSE。

例

この例では、指定されたセルに、任意のセル・テキスト・ラベルのセル・テキストが含まれていない場合に、If...ThenとEnd Ifの各行の間にある文が実行されます。

```
If HS.ContainsCellText("A#Sales.C1#Prod1.C2#Region1") = "False" then
```

```
...
```

```
End If
```



注:

セル・テキスト・ラベルを使用している場合、既存のセル・テキスト・ラベル([Default]を含む)のいずれかにセル・テキスト・エントリが含まれると、この関数は値Trueを返します。

ContainsCellTextWithLabel

指定されたラベルのセル・テキストがセルに入っているかどうかを調べ、セル・テキスト情報を更新します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.ContainsCellTextWithLabel("POVExpression", "CellTextLabel")
```

POVExpressionは、メンバーの組合せです。ディメンションを指定しないと、次のデフォルト値が使用されます。

- Account - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- Scenario - 現在のシナリオ・メンバー

- Entity - 現在のエンティティ・メンバー
- Value - 現在の値メンバー
- YearおよびPeriod - 現在のメンバー

CellTextLabelは、デフォルトのセル・テキスト・ラベル([Default])か、ロードされた有効なセル・テキスト・ラベルの1つです。



注:

この関数を計算ルールで使用すると、デフォルト・エントリが適用されます。配賦ルール、換算ルールまたは連結ルールで使用する場合は、完全なサブキューブのAccount/ICP/Custom POVを指定する必要があります。シナリオ、年、期間、ビュー、エンティティおよび値ディメンションは、デフォルトで現在のメンバーに設定されます。

戻り値

指定したデータ・セルに、指定されたセル・テキスト・ラベルのセル・テキストが含まれる場合はTRUEになるブール式; 含まれない場合はFALSEになります。

例

この例では、指定されたセルに、指定されたセル・テキスト・ラベルのセル・テキストが含まれていない場合に、If...ThenとEnd Ifの各行の間にある文が実行されます。

```
If HS.ContainsCellTextWithLabel ("A#Sales.C1#Prod1.C2#Region1", "Rating") = "False" then ... End If
```

Currency

アプリケーションまたは現在の値ディメンションのメンバーの通貨を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.AppSettings.Currency
```

```
HS.Value.Currency
```

戻り値

アプリケーションまたは値メンバーの通貨の名前を示す文字列。

例

この例では、アプリケーションの通貨がEuroの場合に、If...ThenとEnd Ifの間にあるすべての文が実行されます。

```
If HS.AppSettings.Currency = "Euro" Then
```

```
...
```

```
End If
```

CustomTop

現在のメンバーまたは指定された勘定科目メンバーのCustomTopMemberを戻します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Account.CustomTop("Member", "CustomDimName")
```

```
HS.Account.CustomTop("", "CustomDimName")
```

ここで、*CustomDimName*はカスタム・ディメンションで、*Member*は勘定科目ディメンション・メンバーです。指定されていないメンバーは、Sub Consolidateサブルーチンでのみ適切です。



注:

この関数を現在のメンバーに適用するには、空白の文字列(" ")を使用します。

戻り値

指定した勘定科目メンバーのCustomTopMember。

例

```
HS.Account.CustomTop("Prod", "Sales")
```

DataUnitItem

連結、計算または換算時に処理するデータ単位アイテムを戻します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

構文

```
HS.OpenDataUnit("")
```

戻り値

指定したアイテムのメンバー・ラベル。

例

```
Set MyDataUnit = HS.OpenDataUnit("")
```

```
NumItems= MyDataUnit.GetNumItems
```

```
For i = 0 to NumItems - 1
```

```
Account = MyDataUnit.Item(i)Account.
```

```
ICP = MyDataUnit.Item(i).ICP
```

```
Flow = MyDataUnit.Item(i)Custom("Flows")
```

```
Nature = MyDataUnit.Item(i).Custom("Nature")
```

```
...
```

```
...
```

```
...
```

```
Data = MyDataUnit.Item(i).Data
```

```
POV = MyDataUnit.Item(i).POV
```

```
Next
```

Decimal

指定した勘定科目の小数点以下の桁数を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Account.Decimal("AccountName")
```

```
HS.Account.Decimal("")
```



注:

Sub Consolidateサブルーチンを使用する場合にこの関数を現在の勘定科目に適用するには、空白の文字列("")を使用します。


```
HS.Account.Decimal(  
  Var1  
)
```

表47 Decimal関数の構文

パラメータ	説明
AccountName	勘定科目の有効なメンバーの名前
Var1	勘定科目メンバーを示すVBScript変数

戻り値

勘定科目に割り当てられる小数点以下の桁数の値を表す整数。有効な値は0から9です。

例

この例では、Sales勘定の小数点以下の桁数を2に設定している場合に、If...ThenとEnd Ifの間にあるすべての文が実行されます。

```
If HS.Account.Decimal("Sales") = 2 Then
```

```
...
```

```
End If
```

DefaultFreq

現在のシナリオのメンバーまたは指定されたシナリオのメンバーのデフォルトの期間単位を取得します。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Scenario.DefaultFreq("  
  Scenario  
")
```

```
HS.Scenario.DefaultFreq("")
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

Scenarioは、有効なシナリオ・メンバーの名前です。

戻り値

シナリオのデフォルトの期間単位を示す文字列。

例

この例では、Actualシナリオのデフォルトの期間単位がYTDの場合に、If...ThenとEnd Ifの間にあるすべての文が実行されます。

```
If HS.Scenario.DefaultFreq("Actual") = "YTD" Then
```

```
...
```

```
End If
```

DefaultParent

現在のメンバーまたは指定されたメンバーのデフォルトの親を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.<Object>.DefaultParent("Member")
```

```
HS.<Object>.DefaultParent("")
```

```
HS.Custom("Label").DefaultParent("Member")
```

```
HS.Custom("Label").DefaultParent("")
```

ここで、*Member*は、有効なディメンション・メンバーです。



注:

この関数を現在のメンバーに適用するには、空白の文字列(" ")を使用します。

表48 DefaultParent関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none">• Account• Custom1...4• カスタム(カスタム・ディメンション・ラベル)• Entity• Scenario
<i>Member</i>	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none">• Account• Custom1...4• Custom• Entity• Scenario

戻り値

メンバーのデフォルトの親を示す文字列。

DefaultTranslate

ルールを使用せずに換算を計算します。この関数は、アプリケーション設定を上書きし、SubTranslateルールでのみ使用できます。

構文

```
HS.DefaultTranslate(  
  dRateForBalanceAccounts  
  ,  
  dRateForFlowAccounts  
  ,  
  bUsePVAForFlowAccounts  
  ,  
  bUsePVAForBalanceAccounts  
)
```

表49 DefaultTranslate関数の構文

パラメータ	説明
dRateForBalanceAccounts	レートの数字
dRateForFlowAccounts	レートの数字
bUserPVAForFlowAccounts	TRUEまたはFALSE
bUsePVAForBalanceAccounts	TRUEまたはFALSE

戻り値

なし。

例

この例では、親メンバーがUnited Statesの場合に、If...ThenとEnd Ifの間のすべての文が実行されます。

```
If HS.Parent.Member="UnitedStates" Then
```

```
  HS.DefaultTranslate .25, .27, True, False
```

```
End If
```

DefaultView

現在のシナリオのメンバーまたは指定されたシナリオのメンバーのデフォルトのビューを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

- 配賦

構文

```
HS.Scenario.DefaultView("Scenario")
```

```
HS.Scenario.DefaultView("")
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

Scenarioは、有効なシナリオ・メンバーの名前です。

戻り値

指定されたシナリオのデフォルトのビューを示す文字列。有効な値はYTDとPeriodicです。

例

この例では、Actual(実績)シナリオのデフォルトのビューがYTDである場合に、If...ThenとEnd Ifの間にあるすべての文が実行されます。

```
If HS.Scenario.DefaultView("Actual") = "YTD" Then
```

```
...
```

```
End If
```

DefCurrency

現在のエンティティまたは親メンバー、または指定のエンティティまたは親メンバーのデフォルト通貨を取得します。エンティティを指定すると、エンティティの通貨が戻されます。親通貨を取得するには、親エンティティを指定する必要があります。この関数は、次のタイプのルールで使用できます:

- 計算

- ・ 換算
- ・ 連結
- ・ 配賦

構文

```
HS.
<Object>
.DefCurrency("
Entity
")
```

```
HS.
<Object>
.DefCurrency("")
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表50 DefCurrency関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> ・ Entity ・ Parent
Entity	有効なエンティティ・メンバーの名前

戻り値

指定したエンティティまたは親のデフォルト通貨を含む文字列。

例

この例では、ConnecticutというエンティティとEastRegionというエンティティのデフォルト通貨が同じでない場合に、If...ThenとEnd Ifの間にあるすべての文が実行されます。

```
Then      If HS.Entity.DefCurrency("Connecticut")<> HS.Parent.DefCurrency("EastRegion")
```

...

End If

DOWn

指定されたparent.entityノードの直接出資比率を取得します。この関数は、次のタイプのルールで使用できます：

- ・ 計算
- ・ 換算
- ・ 連結
- ・ 配賦

構文

シナリオ、年、期間メンバーと、parent.entity形式で指定したエンティティ・メンバーを組み合わせます。

```
HS.Node.DOWn("S#  
Scenario  
.Y#  
Year  
.P#  
Period  
.E#  
Parent  
.Entity  
")
```

```
HS.Node.DOWn("")
```



注：

この関数を現在のノードに適用するには、空白の文字列("")を使用します。

表51 DOWn関数の構文

パラメータ	説明
Scenario	有効なシナリオ・メンバーの名前
Year	有効な年。
Period	有効な期間。
Parent.Entity	Parent.Entity形式の、有効なノードの名前

戻り値

直接出資比率を示す数値。

例

```
dVar1 = HS.Node.Down("S#Actual.Y#2014.P#Q1.E#Group1.Ent1")
```

Dynamic

計算が必要な動的勘定科目の式を指定します。この関数は、動的ルールで使用できます。

この関数は、同じサブキューブ内のデータのみを参照できます。別のサブキューブのデータを参照する必要がある場合は、parking勘定科目を作成して、他のキューブの情報を保管することが必要です。たとえば、前年度の式内のデータを参照するには、parking勘定科目を使用して、前年度のデータ情報を保管し、同じキューブ内の動的計算で参照できるようにする必要があります。



注:

Dynamic関数にはHS.View.PeriodNumber関数を埋め込むことができます。例:

```
HS.Dynamic "A#AvgUnits.I#ICP None.C1#None.C2#None = A#AccumUnits.I#ICP  
None.C1#None.C2#None / HS.View.PeriodNumber"
```

構文

```
HS.Dynamic "  
DestPOV  
=  
Expression  
"
```

表52 Dynamic関数の構文

パラメータ	説明
DestPOV	「動的」タイプが設定された有効な勘定科目メンバー。計算を行うビューも指定できます。 注: ビューを指定しないと、式は、期別とYTDに対して実行されます。ビューごとに別の式を持つには、式内に期別またはYTDを指定する必要があります。

パラメータ	説明
Expression	計算式。



注:

Dynamic関数では、IF...THEN文はサポートされません。

戻り値

なし。

例

```
Sub Dynamic
```

```
HS.Dynamic "A#GM% = A#GM/(A#Sales * 100)/HS.View.PeriodNumber"
```

```
End Sub
```

GM%勘定科目の予想結果は次のとおりです。

カスタム1	Sales	GM	GM%(動的計算)	計算式
製品	600	140	23.33%	140 / 600 * 100
P1	100	10	10%	10 / 100 * 100
P2	200	40	20%	40 / 200 * 100
P3	300	90	30%	90 / 300 * 100

Exp

勘定科目、ICPおよびカスタムのメンバー(カスタム1から4まで)の組合せにデータを挿入します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 配賦

構文

```
HS.Exp "
DestPOV
=
Expression
"
```

表53 Exp関数の構文

パラメータ	説明
DestPOV	データの挿入先を識別する宛先視点。使用に関する次のルールに注意してください: データベースに不要な値を移入することを避けるには、データを配置する場所の定義について、ルールをできるだけ明確にする必要があります。適切な方法は、勘定科目ディメンションに、明確に定義されたICPおよびカスタム・ディメンションの交差を含めることです。この明確に定義された交差では、Financial Managementの検証チェックを使用して、無効な交差にデータが書き込まれないようにします。
Expression	計算式。

データの宛先は等号の左側にあり、割り当てられるデータは右側にあります。この例は、UnitsSold勘定科目およびカスタム・ディメンションの[None]メンバーと交差するセルを40000に設定します。

```
HS.Exp "A#UnitsSold.C1#[None].C2#[None].C3#[None].C4#[None]" _
```

```
& " = 40000"
```

等号の右側で、勘定科目式の文字を使用してディメンション・メンバーを表すことができます。したがって、セルのグループのデータをセルの別のグループに代入できます。この例は、Taxes勘定科目およびカスタム・ディメンションの[None]メンバーと交差するセルを、Sales勘定科目および指定されたカスタム・ディメンションと交差するセルのデータの6パーセントに設定します。

```
HS.Exp "A#Taxes.C1#[None].C2#[None].C3#[None].C4#[None]" _
```

```
& " = A#Sales.C1#AllProducts.C2#AllCustomers.C3#[None]" _
```

```
& ".C4#[None] * .06"
```



ヒント:

1つのExp文に複数の勘定科目を設定できます。[301ページの複数の勘定科目の同時設定を参照してください。](#)

ディメンションの交差に関する考慮事項

Expがデータを取得し、Expがデータを配置するディメンションの交差を指定しない場合は、ソース交差と宛先交差は次の要素によって決定されます。

- 宛先。内部取引パートナ・ディメンションまたはカスタム・ディメンションのメンバーが等号の左側に指定されていない場合は、Expによって、勘定科目およびディメンションの有効な各交差にデータが挿入されます。宛先勘定科目を指定していない場合は、Financial Managementによって、現在の視点に有効なすべての勘定科目にデータが挿入されます。[301ページの複数の勘定科目の同時設定](#)を参照してください。
- ソース。ディメンションのメンバーが等号の右側に指定されていない場合は、次のようないくつかの可能性がありま
す。
 - ディメンションのメンバーが1つのみの場合、Expはこのメンバーとソース勘定科目の交差からデータを取得しま
す。
 - ディメンションにソース勘定科目との有効な交差が1つのみある場合、Expはこの交差からデータを取得しま
す。
 - ディメンションにソース勘定科目と交差するメンバーが複数ある場合、データのソース交差は次のように等号の
左側によって決定されます。
 - メンバーが左側に指定されている場合は、Expは、このメンバーとソース勘定科目の交差からデータの取得を
試みます。
 - メンバーが左側に指定されていない場合は、Expは、宛先勘定科目およびディメンションのメンバーの有効な
各交差へのデータの挿入を試みます。Expは、メンバーおよびソース勘定科目の対応する交差から宛先交差
のデータを取得します。



注:

ソース交差が無効な場合、Expは対応する宛先交差のデータを変更しません。

これらの考慮事項を説明する例の詳細は、[302ページのExpおよびディメンションの交差に関する考慮事項](#)を参照してください。

期間および年のキーワード

動的なルールを作成するには、メンバー名かわりに [299ページの表 54](#)のキーワードを使用して、宛先の年または期間ディメンションのメンバーを表します。

表54 期間および年のキーワードと説明

キーワード	説明
CUR	現在の期間または年。
FIRST	アプリケーションに有効な最初の期間または年。
LAST	アプリケーションに有効な最後の期間または年。
NEXT	現在の期間または年の後の期間または年。

キーワード	説明
PRIOR	現在の期間または年の前の期間または年。



注:

期間と年のキーワードでは大文字と小文字が区別されます。キーワードはすべて大文字で指定する必要があります。

プラス記号(+)とマイナス記号(-)を年と期間のキーワードに使用できます。この例は、現在の年より前の2年分のMisc勘定科目の値をMiscPast勘定科目に設定します。

```
HS.Exp "A#MiscPast = A#Misc.Y#CUR-2"
```

キーワードPrior, First, Last, CurrentまたはNextの直後にプラス記号(+)およびマイナス記号(-)および数字を使用する場合は、等式の順序で正しい構文を使用するように注意する必要があります。この場合は、次のメソッドのいずれかを使用してルールを記述できます。

変数を正しく区切るために、常に丸カッコを使用してください。例:

```
HS.Exp "A#9001_Group.C4#[None] = A#9001_Group.P#Prior" & "+" & VAR &
"*A#9001_Group.V#[ParentTotal]"
```

または

```
HS.Exp "A#9001_Group.C4#[None] = (A#9001_Group.P#Prior" & "+" & VAR & ")
*A#9001_Group.V#[Parent Total]"
```

算術計算

等号の右側で、加算、減算、乗算および除算ができます。次の標準のVBScript文字を使用する必要があります。

```
+ - * /
```

この例では、前年のTaxes勘定科目とPurchases勘定科目の値を加算して、その合計をCash勘定科目に挿入します。

```
HS.Exp "A#Cash = A#Taxes.Y#PRIOR + A#Purchases.Y#PRIOR"
```



注:

ステータスがNoDataの勘定科目で乗算または除算を行っても、等号の左側の勘定科目のデータは変更されません。ゼロ(0)はデータとみなされます。つまり、データとして0.00が含まれている勘定科目のステータスはNoDataにはなりません。

Exp内での他の関数の配置

関数が単一の数値を戻す場合は、Exp関数内のその関数をネストできます。ただし、String引数が含まれている関数をネストすると、Stringを引用符で囲えません。次の例では、NumBase関数がExp関数内でネストされているため、そのString関数は引用符で囲えません。

```
HS.Exp "A#SalesAlloc = A#Sales/HS.Entity.NumBase(Regional)"
```

複数の勘定科目の同時設定

現在の視点と交差するすべての勘定科目にデータを挿入するには、勘定科目式でAllを使用します。これを使用して、すべての勘定科目の期首残高を設定できます。次の例では、現在の期間が最初の期間であるかどうかをIsFirst関数で検証します。最初の期間であると、Expは、現在の期間の各勘定科目の値を前年の最後の期間の勘定科目の値に設定します。

```
If HS.Period.IsFirst = TRUE Then
```

```
HS.Exp "A#ALL = A#ALL.Y#PRIOR.P#LAST"
```

```
End If
```

勘定科目およびカスタム・ディメンションまたは内部取引パートナー(ICP)ディメンションのすべての交差にデータを挿入するには、Allキーワードを使用するか、またはA#文字を指定しません。A#文字を指定しない次の例では、カスタム1ディメンションのGolfBallsメンバーと交差する各勘定科目にデータが挿入されます。GolfBallsと勘定科目の個々の有効な交差の場合は、前の期間の交差の金額が、現在の期間の交差に配置されます。

```
HS.Exp "C1#GolfBalls = C1#GolfBalls.P#PRIOR"
```

戻り値

なし。

例

この例では、StateTax勘定科目の金額を設定します。2014年のSales勘定科目の金額に、2014年のStateRate勘定科目のレートを乗算して計算します。

```
HS.Exp "A#StateTax = A#Sales.Y#2014 * A#StateRate.Y#2014"
```

Expおよびディメンションの交差に関する考慮事項

次の例は、299ページのディメンションの交差に関する考慮事項に記載されている考慮事項を示しています。次のタイプの交差が取り上げられています。

- ・ ソース勘定科目と宛先勘定科目に対して、すべての交差が有効である。302ページのすべての交差が有効な場合を参照してください。
- ・ ソース勘定科目と宛先勘定科目に対して、一部の交差が有効で、その他は無効である。303ページの交差が無効な場合を参照してください。
- ・ ソース勘定科目に対して1つのメンバーが有効である。304ページの右側に1つの有効なメンバーがある場合を参照してください。

次のすべての例で、Member001、Member002、Member003という名前のカスタム1ディメンションのメンバーとともに、TargAcctおよびSourceAcctという名前の勘定科目が使用されます。これらすべての例のソース交差データは、302ページの表 55に記載されています。

表55 ディメンション交差のデータの例

メンバー	SourceAcct交差のデータ
Member001	10
Member002	NoDataステータス
Member003	15

すべての交差が有効な場合

次の例では、TargAcct勘定科目、SourceAcct勘定科目、およびカスタム1メンバーの交差がすべて有効です。

```
HS.Exp "A#TargAcct = A#SourceAcct"
```

この関数は、TargAcct勘定科目とカスタム1メンバーとの交差に次のデータを挿入します。

カスタム1メンバー	データ	交差
Member001	10	SourceAcctおよびMember001
Member002	---	該当なし。SourceAcctとMember002の交差のステータスがNoDataであるため、TargAcctとMember002の交差は変更されません。
Member003	15	SourceAcctおよびMember003

この例では、等号の左側でMember001メンバーとともにExpが使用されています。

```
HS.Exp "A#TargAcct.C1#Member001 = A#SourceAcct"
```

TargAcctとMember001の交差は10に設定されています。ExpはSourceAcctとMember001の交差からデータを取得します。これは、Member001が左側に指定されているためです。

この例では、等号の右側でMember003メンバーとともにExpが使用されています。

```
HS.Exp "A#TargAcct = A#SourceAcct.C1#Member003"
```

この関数は、カスタム1メンバーとTargAcct勘定科目との交差に次のデータを挿入します。

カスタム1メンバー	データ	交差
Member001	15	SourceAcctおよびMember003
Member002	15	SourceAcctおよびMember003
Member003	15	SourceAcctおよびMember003

交差が無効な場合

次の例では、ソース勘定科目と宛先勘定科目にそれぞれ無効な交差があります。

- SourceAcct.Member002およびSourceAcct.Member003は有効で、Member001は無効です。
- TargAcct.Member002およびMember003は有効で、Member001は無効です。

この例では、等号のどちら側でもカスタム1メンバーを指定せずに、Expが使用されています。

```
HS.Exp "A#TargAcct = A#SourceAcct"
```

ExpによってTargAcct勘定科目のデータが変更されることはありません。Expによって次の演算が試みられるためです。

- TargAcct.Member001 = SourceAcct.Member001。SourceAcctとMember001は無効な交差です。
- TargAcct.Member002 = SourceAcct.Member002。SourceAcctとMember002の交差のステータスはNoDataであるため、TargAcctとMember002の交差は変更されません。
- TargAcct.Member003 = SourceAcct.Member003。TargAcctとMember003は無効な交差です。

この例では、等号の左側でMember001を指定して、Expが使用されています。

```
HS.Exp "A#TargAcct.C1#Member001 = A#SourceAcct"
```

Expは無効な交差(SourceAcctとMember001)からデータの取得を試みるので、TargAcct.Member001は変化しません。

この例では、等号の右側でMember003を指定して、Expが使用されています。

```
HS.Exp "A#TargAcct = A#SourceAcct.C1#Member003"
```

この関数は、カスタム1メンバーとTargAcct勘定科目との交差に次のデータを挿入します。

カスタム1メンバー	データ	交差
Member001	15	SourceAcctおよびMember003

カスタム1メンバー	データ	交差
Member002	15	SourceAcctおよびMember003
Member003	N/A	該当なし。Member003はTargAcct勘定科目には無効な交差です。

右側に1つの有効なメンバーがある場合

次の例では、有効なメンバーがソース勘定科目には1つのみで、宛先勘定科目には2つあります。

- SourceAcct。Member003は唯一の有効な交差です。
- TargAcct。Member002およびMember003は有効で、Member001は無効です。

この例では、等号のどちら側でもカスタム1メンバーを指定せずに、Expが使用されています。

```
HS.Exp "A#TargAcct = A#SourceAcct"
```

この関数は、カスタム1メンバーとTargAcct勘定科目との交差に次のデータを挿入します。

カスタム1メンバー	データ	交差
Member001	15	SourceAcctおよびMember003(SourceAcct勘定科目の唯一の有効な交差)
Member002	15	SourceAcctおよびMember003(SourceAcct勘定科目の唯一の有効な交差)
Member003	N/A	該当なし。Member003はTargAcct勘定科目には無効な交差です。

この例では、等号の左側でMember001を指定して、Expが使用されています。

```
HS.Exp "A#TargAcct.C1#Member001 = A#SourceAcct"
```

TargAcctとMember001の交差は15に設定されています。これはSourceAcctとMember003の交差のデータです。



ヒント:

SourceAcct勘定科目とカスタム1ディメンションに有効な交差が複数あった場合、ExpはSourceAcctとMember001の交差からデータの取得を試みます。これが無効な交差であると、Expは宛先勘定科目を変更しません。

GetCell

セルのデータを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.GetCell("
POVExpression
")
```

POVExpressionは、有効な視点(POV)です。

戻り値

指定したセルにあるデータ。



注:

関数によって複数の値が戻された場合はエラーが発生します。

例

この例では、Sales勘定とカスタム1ディメンションのGolfBallsメンバーの交差にある値をdData変数に割り当てます。

```
Dim dData
```

```
dData = HS.GetCell("A#Sales.I#[ICP
None].C1#Golfballs.C2#Customer2.C3#Increases.C4#[None]")
```

GetCellNoData

セルに含まれているデータを取得し、セルにデータが含まれているかどうかを示します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.GetCellNoData("
POV
",
Var1
)
```

表56 GetCellNoData関数の構文

パラメータ	説明
POV	有効な視点。
Var1	セルにデータがあるかどうかを示す変数。

戻り値

戻される値は、セルの内容によって異なります。

- セルに実際のデータが含まれている場合は、データ値が戻され、Var1に戻されるブール値はFALSEになります。
- セルにデータが含まれていない場合は、データ値として0が戻され、Var1に戻されるブール値はTRUEになります。
- セルに計算結果データが含まれている場合は、その値が戻され、Var1に戻されるブール値はFALSEになります。



注意

引数によってGetCellNoDataに複数の値を戻させると、エラーが発生します。

例

この例では、Sales勘定の金額がdSalesAmt変数に割り当てられます。Sales勘定にデータがない場合は、If Then文とEnd If文間にあるすべての文が実行されます。

```
dSalesAmt = HS.GetCellNoData("A#Sales.I#[ICP
None].C1#[None].C2#[None].C3#[None].C4#[None]", bIsNoData)
```

```
If bIsNoData = TRUE then
```

```
...
```

```
End If
```

GetCellRealData

セルに含まれているデータを取得し、セルに実数データが含まれているかどうかを示します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.GetCellRealData("
POV
",
Var1
)
```

表57 GetCellRealData関数の構文

パラメータ	説明
POV	有効な視点。
Var1	セルに実際のデータがあるかどうかを示す変数。

戻り値

戻される値は、セルの内容によって異なります。

- セルに実数データが含まれている場合は、データ値が戻され、var1に返されるブール値はTRUEになります。
- セルにデータが含まれていない場合は、データ値として0が戻され、var1に返されるブール値はFALSEになります。
- セルに計算結果データが含まれている場合は、その値が戻され、var1に返されるブール値はFALSEになります。

例

```
dData = HS.GetCellRealData("A#Sales.C1#Prod1", bIsRealData)
```

```
If bIsRealData = TRUE then
```

```
...
```

End If

GetCellText

視点からデフォルトのセル・テキストを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.GetCellText("
POVExpression
")
```

POVExpressionは、有効な視点(POV)です。

戻り値

視点(POV)の[Default]ラベル・セル・テキスト。

例

```
HS.GetCellText("S#Actual.Y#2014.P#January.E#Connecticut.V#<Entity
Currency>.A#Sales.I#[ICP None].C1#[None].C2#[None].C3#[None].C4#[None]")
```



注:

セル・テキスト・ラベルを使用している場合、この関数は、セル・テキスト・ラベル[Default]のセル・テキストを取得します。

GetCellTextWithLabel

指定したセル・テキスト・ラベルの視点からセル・テキストを取得します。この関数は、次のタイプのルールで使用できます:

- 計算

- 換算
- 連結
- 配賦

構文

```
HS.GetCellTextWithLabel("POVExpression", "CellTextLabel")
```

POVExpressionは、メンバーの組合せです。ディメンションを指定していない場合は、次のデフォルト値が使用されます。

- Account - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- Scenario - 現在のシナリオ・メンバー
- Entity - 現在のエンティティ・メンバー
- Value - 現在の値メンバー
- YearおよびPeriod - 現在のメンバー

CellTextLabelは、デフォルトのセル・テキスト・ラベル([Default])か、ロードされた有効なセル・テキスト・ラベルの1つです。



注:

この関数を計算ルールで使用すると、デフォルトのディメンション・メンバー値が適用されます。配賦ルール、換算ルールまたは連結ルールで使用する場合は、Account、ICPおよびCustom1から4のメンバーを指定する必要があります。シナリオ、年、期間、ビュー、エンティティおよび値は、デフォルトで現在のメンバーに設定されます。

戻り値

POVのセル・テキストおよび指定されたセル・テキスト・ラベル。

例

```
HS.GetCellTextWithLabel("A#Sales.I#[ICP  
None].C1#[None].C2#[None].C3#[None].C4#[None]", "Rating")
```

GetCellType

セル・タイプを取得します。この関数は、次のタイプのルールで使用できます:

- 計算

- ・ 換算
- ・ 連結
- ・ 配賦



注:

勘定科目の種別は、すべて大文字です。

構文

```
HS.GetCellType("
POVExpression
")
```

POVExpressionは、有効な視点(POV)です。

戻り値

特定のセルのタイプ。

次のタイプがあります。

- ・ ASSET(資産)
- ・ LIABILITY(負債)
- ・ REVENUE(収益)
- ・ EXPENSE(費用)
- ・ FLOW(フロー)
- ・ BALANCE(残高)
- ・ BALANCERECURRING(経常残高)
- ・ CURRENCYRATE(通貨レート)
- ・ GROUPLABEL(グループラベル)

例

この例では、セル・タイプがEXPENSE(費用)かどうかをチェックします。EXPENSE(費用)の場合は、If...Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.GetCellType("S#Actual.C4#[None]") = "EXPENSE" Then
```

...

End If

GetCustomLabelArray

配列のカスタム・ディメンション・ラベルのリストを戻します。この関数は、次のタイプのルールで使用できます：

- ・ 計算
- ・ 換算
- ・ 連結
- ・ 配賦

構文

```
HS.GetCustomLabelArray  
(Names, Aliases)
```

戻り値

カスタムのショート名およびカスタムのロング名の2つの配列を戻します。

例

```
HS.GetCustomLabelArray(  
Custom1  
,  
Products  
)
```

GetItem

連結する個々のレコードを取得します。この関数は、連結ルールで使用できます。



注：

この関数は、4つ以下のカスタム・ディメンションにのみ、使用できます。4つより多いカスタム・ディメンションには、[287ページのDataUnitItem](#)を使用してください。

構文

```
DataUnit.GetItem (  
  lItem  
  ,  
  strAccount  
  ,  
  strICP  
  ,  
  strCustom1  
  ,  
  strCustom2  
  ,  
  strCustom3  
  ,  
  strCustom4  
  ,  
  dData)
```

表58 GetItem関数の構文

パラメータ	説明
lItem	レコードの番号。
strAccount	勘定科目ディメンションの有効なメンバーの名前。
strICP	ICPディメンションの有効なメンバーの名前。
strCustom1	カスタム1ディメンションの有効なメンバーの名前。
strCustom2	カスタム2ディメンションの有効なメンバーの名前。
strCustom3	カスタム3ディメンションの有効なメンバーの名前。
strCustom4	カスタム4ディメンションの有効なメンバーの名前。
dData	指定したセルにあるデータ。

戻り値

勘定科目メンバー、ICPメンバー、カスタム1...4メンバー、およびデータを含む配列。

例

```
Call DataUnit.GetItem(lItem, strAccount, strICP, strCustom1, strCustom2,  
strCustom3, strCustom4, dData)
```

```
If dData = 0 Then
```

```
...
```


End If

GetItemIDs2

ディメンションIDを使用して、連結する個々のレコードを取得します。この関数は、連結ルールで使用できます。



注:

この関数は、4つ以下のカスタム・ディメンションにのみ、使用できます。4つより多いカスタム・ディメンションには、[314ページ](#)のGetItemIDs2ExtDimを使用してください。

構文

```
DataUnit.GetItemIDs2(  
  lItem  
  ,  
  lAccountID  
  ,  
  lICPID  
  ,  
  lCustom1ID  
  ,  
  lCustom2ID  
  ,  
  lCustom3ID  
  ,  
  lCustom4ID  
  ,  
  dData  
)
```

表59 GetItemIDs2関数の構文

パラメータ	説明
lItem	レコードの番号。
lAccountID	勘定科目ディメンションの有効なメンバーのID番号。
lICPID	ICPディメンションの有効なメンバーのID番号。
lCustom1ID	カスタム1ディメンションの有効なメンバーのID番号。
lCustom2ID	カスタム2ディメンションの有効なメンバーのID番号。
lCustom3ID	カスタム3ディメンションの有効なメンバーのID番号。
lCustom4ID	カスタム4ディメンションの有効なメンバーのID番号。
dData	指定したセルにあるデータ。

戻り値

勘定科目、ICP、カスタム1...4のデータを含む変数。

例

```
Call DataUnit.GetItemIDs2(lItem, lAccount, lICP, lCustom1, lCustom2,  
lCustom3, lCustom4, dData)
```

```
If dData = 0 Then
```

```
...
```

```
End If
```

GetItemIDs2ExtDim

ディメンションIDを使用して、連結する個々のレコードを取得します。この関数は、連結ルールで使用できます。



注:

この関数は、4つより多いカスタム・ディメンションがある場合に使用されます。

構文

```
DataUnit.GetItemIDs2ExtDim(  
lItem  
,  
lAccountID  
,  
lICPID  
,  
lCustomID  
,  
dData  
)
```

表60 GetItemIDs2関数の構文

パラメータ	説明
lItem	レコードの番号。

パラメータ	説明
lAccountID	勘定科目ディメンションの有効なメンバーのID番号。
lICPID	ICPディメンションの有効なメンバーのID番号。
lCustomID	カスタム・ディメンションの有効なメンバーのID番号。
dData	指定したセルにあるデータ。

戻り値

ディメンションとメンバーの両方のIDを含む2列の配列。

例

```
Call DataUnit.GetItemIDs2(lItem, lAccount, lICP, lCustom5, dData)
```

```
If dData = 0 Then
```

```
...
```

```
End If
```

GetNumItems

連結するレコードの数を取得します。この関数は、連結ルールで使用できます。

構文

```
NumItems = DataUnit.GetNumItems
```

戻り値

データ単位内のレコードの数。

例

```
Set dataUnit = HS.OpenDataUnit("")
```

```
lNumItems = dataUnit.GetNumItems
```

```
for lItem = 0 to lNumItems - 1
```

```
' Get the next item from the Data Unit
```

```
Call dataUnit.GetItem(lItem, strAccount, strICP, strCustom1, strCustom2,  
strCustom3, strCustom4, dData)
```

GetNumLID

指定した視点のライン・アイテムの詳細の数を取得します。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.GetNumLID(  
POVExpression  
)
```

POVExpressionは有効なPOVの組合せです。ディメンションを指定しないと、次の値が使用されます。

- Account - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- Scenario - 現在のシナリオ・メンバー
- Entity - 現在のエンティティ・メンバー
- Value - <エンティティ通貨>
- YearおよびPeriod - 現在のメンバー



注：

無効な交差を指定すると、0が戻されます。

戻り値

指定したセルのライン・アイテムの詳細の数。

例

この例では、指定したセルにライン・アイテムの詳細が入力されていないと、If...Then行とEnd If行の間にあるすべての文が実行されます。

```
If HS.GetNumLID("A#Sales.C1#Prod1.C2#Region1") = 0 then
```

```
...
```

```
End If
```

GetRate

現在の視点または指定した視点の換算レートを取得します。この関数は、次のタイプのルールで使用できます。

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.GetRate("POVExpression")
```

POVExpressionは有効なデータの視点(POV)です。



注:

レート勘定を指定する必要があります。

戻り値

指定した視点の為替レート。

例

```
dVar1 = HS.GetRate("S#Actual.Y#2014.P#March.V#Euro.E#Connecticut.A#AvgRate")
```

GetSubmissionGroup

セルのプロセス管理の提出グループを取得します。

構文

```
HS.GetSubmissionGroup("A#  
Account  
.C1#  
Custom1  
.C2#  
Custom2  
.C3#  
Custom3  
.C4#  
Custom4  
.I#  
ICP  
")
```

戻り値

プロセス管理の提出グループを表す整数です。有効な値は1-99です。

例

```
dVar1=HS.GetSubmissionGroup("A#Sales.C1#Golfballs.C2#Tennisballs.C3#Soccerballs.C4#Basketballs.I#Ea
```

GetSubmissionPhase

セルのプロセス管理の提出フェーズを取得します。

構文

```
HS.GetSubmissionPhase("S#  
Scenario  
.P#
```

```
Period
.A#
Account
.C1#
Custom1
.C2#
Custom2
.C3#
Custom3
.C4#
Custom4
.I#
ICP
")
```

戻り値

プロセス管理の提出フェーズを表す整数です。有効な値は1-9です。

例

```
dVar1=HS.GetSubmissionPhase("S#Actual.P#January.A#Sales.C1#Golfballs.C2#Tennisballs.C3#Soccerballs")
```

Holding

現在、指定されたエンティティ、または親ディメンション・メンバーの持株会社を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Entity.Holding("
Entity
")
```

```
HS.Entity.Holding("")
```

```
HS.Parent.Holding("
```

```
Entity  
")
```

```
HS.Parent.Holding("")
```

Entityは、有効なエンティティ・ディメンションまたは親メンバーの名前です。空白の文字列(" ")を使用して、この関数を現在のメンバーに割り当てることができます。

戻り値

指定されたエンティティ・メンバーまたは親の持株会社の名前を示す文字列。

例

この例では、EuropeがエンティティFranceの持株会社である場合、If...Then行とEnd If行の間にあるすべての文が実行されます。

```
If HS.Entity.Holding("France") = "Europe" Then
```

```
...
```

```
End If
```

ICPTopMember

現在の勘定科目ディメンション・メンバー、または指定した勘定科目メンバーのICPTopMemberを取得します。この関数は、次のタイプのルールで使用できます。

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Account.ICPTopMember ("  
AccountName  
")
```



```
HS.Account.ICPTopMember ("")
```

```
HS.Account.ICPTopMember (  
Var1  
)
```



注:

この関数をSub Consolidateサブルーチンで使用する場合のみ、空白の文字列(" ")を使用して、現在の勘定科目に適用できます。

表61 ICPTopMember関数の構文

パラメータ	説明
AccountName	勘定科目ディメンションの有効なメンバーの名前。
Var1	VisualBasic変数。

戻り値

勘定科目のICPTopMemberの名前を示す文字列。

例

この例では、Sales勘定のICPTopMemberがTotalProdである場合、If...Then文とEnd If文の間にあるすべての文が実行されます。

```
If HS.Account.ICPTopMember("Sales") = "TotalProd" Then
```

```
...
```

```
End If
```

ICPWeight

アプリケーションのICP加重を取得します。値ディメンションの[ICP Top]メンバーに集約される、内部取引パートナー・エンティティ[ICP Entities]の額の割合です。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

- ・ 配賦

構文

```
HS.AppSettings.ICPWeight
```

戻り値

ICPTopMemberに集約されるICPエンティティの割合。この値は、小数で表した百分率の値(1.0が100パーセント)です。

例

この例では、現在のアプリケーションのICP Weightが1である場合に、If...Then行とEnd If行の間にあるすべての文が実行されます。

```
If HS.AppSettings.ICPWeight = 1 Then
```

```
...
```

```
End If
```

IDFromMember

指定したメンバーのID番号を取得します。この関数は、次のタイプのルールで使用できます:

- ・ 計算
- ・ 換算
- ・ 連結
- ・ 配賦

構文

```
HS.  
<Object>  
.IDFromMember ("  
Element  
")
```

表62 IDFromMember関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> • Account • Custom1...4 • Custom (カスタム・ディメンション・ラベル) • Entity • ICP • Parent • Period • Scenario • Value • Year • View
Element	次のいずれかのディメンションの有効なメンバーの名前。 選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> • Account • Custom1...4 • Custom • Entity • ICP • Parent • Period • Scenario • Value • Year

戻り値

指定したメンバーのID番号。メンバー・エントリが有効なメンバーではない場合、戻り値は-1になります。

例

この例では、ConnecticutというメンバーのID番号を取得します。

```
1EntityID = HS.Entity.IDFromMember("Connecticut")
```

```
1CustomID=HS.Custom("Prod").IDFromMember("P3000-Phones")
```

ImpactStatus

指定したデータ単位のステータスを「影響済」に変更します。この関数は計算ルールで使用できます。

構文

シナリオ、年、期間、エンティティ、および値メンバーの組合せ。シナリオが同じ場合は、年と期間の組合せを将来の期間にする必要があります。値メンバーを指定しないと、現在のメンバーが使用されます。

```
HS.ImpactStatus"S#  
Scenario  
.Y#  
Year  
.P#  
Period  
.E#  
Entity  
.V#  
Value  
"
```

表63 ImpactStatus関数の構文

パラメータ	説明
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。
Entity	エンティティ・ディメンションの有効なメンバーの名前。
Value	値ディメンションの有効なメンバーの名前。



注:

指定されたPOVが、処理されるデータ単位と同じシナリオ、年、期間およびエンティティである場合、エラーが発生し、データ単位には影響を与えません。ルールの実行時にターゲット期間がロックされる場合、VBScriptエラーが戻され、データ単位には影響を与えません。

戻り値

なし。

例

```
HS.ImpactStatus "S#Actual.Y#2014.P#January"
```

Input

基本エンティティ・レベルで入力できるデータ・ポイントに対して親エンティティの<エンティティ通貨>にデータ入力できます。これには、IsCalculatedとしてフラグ設定されているデータ・ポイントおよびNoInputとして設定されているデータ・ポイントは含まれません。エンティティ通貨の値ディメンションのみがサポートされます。この関数を使用すると、親エンティティ・レベルの値は、子のコントリビューションの合計に等しくなりません。

この関数は親エンティティ・レベルでの入力を可能にするため、子のコントリビューションは親エンティティのエンティティ通貨の値ディメンションにロール・アップしません。ただし、親子コントリビューション値が保管され、コントリビューション調整の値ディメンションへの仕訳の転記はできます。この関数は入力ルールで使用できます。



注意

親エンティティ・レベルで入力できる勘定科目では、親エンティティのエンティティ通貨の値メンバーの値が、すべての子のコントリビューションの合計に等しくならないことに注意してください。

構文

```
HS.Input"  
POVExpression  
"
```

POVExpressionは視点(POV)です。

戻り値

なし。

例

この例では、売上勘定と予算シナリオが交差するセルにデータを入力できるようにします。

```
Sub Input
```

```
HS.Input"A#Sales.S#Budget"
```

```
End Sub
```

IsAlmostEqual

渡された値が等しいかどうかを、事前に定義されているFinancial Managementイプシロンに基づいてチェックします。この関数は、すべてのタイプのルールで使用できます。

-0.00000000000001から0.00000000000001の差は、ゼロ差分と見なされます。

構文

```
BooleanValue = HS.IsAlmostEqual(Value1, Value2)
```

戻り値

渡された値が等しい場合はTRUEとなるブール式、等しくない場合はFALSEとなるブール式。

例

```
Dim BoolVal
```

```
Dim Value1
```

```
Dim Value2
```

```
Value1 = 10.1299999999
```

```
Value2 = 10.13
```

```
BoolVal = HS.IsAlmostEqual(Value1, Value2)
```

```
If BoolVal = true Then
```

```
    // do processing
```

```
Else
```

```
// do Processing
```

```
End If
```

IsBase

現在のメンバーまたは指定したメンバーが、アプリケーションまたは指定した親の基本メンバーであるかどうかを調べます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦



注:

基本メンバーとは、子を持たない(ツリー階層の一番下にある)メンバーのことです。

構文

```
HS.  
<Object>  
.IsBase("  
Parent  
", "  
Element  
")
```

```
HS.Node.IsBase("  
Parent  
"  
:  
"  
Entity  
"."S#  
Scenario  
.  
Y#Year  
.P#  
Period  
")
```

```
HS.  
<Object>  
.IsBase("", "")
```

```

HS.Custom(
  Dimension
).IsBase(
  Member
)

```



注:

この関数を現在のメンバーに適用するには、空白の文字列(" ")を使用します。

表64 IsBase関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> • Account • Custom1...4 • Custom (カスタム・ディメンション・ラベル) • Entity • Parent
Parent	有効な親メンバー。 注: Parentは、Nodeと共に使用する場合にのみ必須です。
Element	次のいずれかのディメンションの有効なメンバーの名前。 選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> • Account • Custom1...4 • Custom • Entity • Parent
Entity	エンティティ・ディメンションの有効なメンバーの名前。
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。



注:

Nodeオブジェクトを使用した場合は、要素が指定した親のアクティブな基本メンバーかどうか調べられます。

戻り値

要素が指定した親の下位にある基本メンバーである場合にTRUEとなり、親が指定されていない状態ではアプリケーションの基本メンバーとなるブール式。要素が基本メンバーではない場合はFALSEになります。

ノードの場合は、要素が指定した視点の親の下位にあるアクティブな基本エンティティである場合はTRUEになります。要素がアクティブな基本エンティティでない場合はFALSEになります。

例

この例では、ConnecticutがEastRegionの下位にある基本エンティティであると、If...Then行とEnd If行の間にあるすべての文が実行されます。

```
If HS.Entity.IsBase("EastRegion","Connecticut") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Prod").IsBase("P3000-Phones") = TRUE Then
```

```
...
```

```
End If
```

IsCalculated

現在の勘定科目ディメンション・メンバーまたは指定した勘定科目メンバーが計算済勘定科目であるかどうかを調べます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Account.IsCalculated("Account")
```

)

Accountは、有効な勘定科目メンバーの名前です。

```
HS.Account.IsConsolidated("")
```



注:

Sub Consolidateサブルーチンで関数を使用している場合にのみ、空白の文字列("")を使用して、この関数を現在のメンバーに適用できます。

戻り値

勘定科目が計算された勘定科目の場合はTRUEとなるブール式、そうでない場合はFALSEとなるブール式。

例

この例では、売上勘定が計算されると、If...Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.Account.IsCalculated("Sales") = TRUE Then
```

```
...
```

```
End If
```

IsChild

現在のメンバーまたは指定したメンバーが、指定した親の子であるかどうかを調べます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦



注:

メンバーがツリー階層で別のメンバーのすぐ1つ下のレベルにある場合、そのメンバーは子です。

構文

```
HS.  
<Object>  
.IsChild("  
Parent  
", "  
, "  
Element  
")
```

```
HS.Node.IsChild("  
Parent  
". "  
Entity  
". "S#  
Scenario  
.Y#  
Year  
.P#  
Period  
")
```

```
HS.  
<Object>  
.IsChild("  
Parent  
", "")
```

```
HS.Custom(  
Dimension  
) .IsChild("P3000-Phones")
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表65 IsChild関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none">• Account• Custom1...4• Custom (カスタム・ディメンション・ラベル)• Entity• Parent
Parent	親メンバーの名前(親は必須です。)

パラメータ	説明
Element	次のいずれかのディメンションの有効なメンバーの名前。 選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> • Account • Custom1...4 • Custom • Entity • Parent
Entity	エンティティ・ディメンションの有効なメンバーの名前。
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。



注:

Nodeオブジェクトを使用した場合は、メンバーが、指定した親のアクティブな子かどうか調べられます。

戻り値

要素が指定した親の子である場合にTRUEとなり、そうでない場合にFALSEとなるブール式。

ノードの場合は、要素が指定した親のアクティブな子である場合にTRUEとなり、そうでない場合にFALSEとなります。

例

この例では、ConnecticutがEastRegionの子であると、If...Then行とEnd If行の間にあるすべての文が実行されます。

```
If HS.Entity.IsChild("EastRegion","Connecticut") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Prod").IsChild("P3000-Phones") = TRUE Then
```

```
...
```

```
End If
```

IsConsolidated

現在の勘定科目ディメンション・メンバーまたは指定した勘定科目メンバーが連結勘定科目であるかどうかを調べます。この関数は、次のタイプのルールで使用できます：

- ・ 計算
- ・ 換算
- ・ 連結
- ・ 配賦

構文

```
HS.Account.IsConsolidated("Account")
```

Accountは、有効な勘定科目メンバーの名前です。

```
HS.Account.IsConsolidated("")
```



注：

Sub Consolidateサブルーチンで関数を使用している場合にのみ、空白の文字列("")を使用して、この関数を現在のメンバーに適用できます。

戻り値

勘定科目が親勘定科目に連結される場合にTRUEとなり、そうでない場合にFALSEとなるブール式。

例

この例では、売上勘定科目が連結されると、If...Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.Account.IsConsolidated("Sales") = TRUE Then
```

...

End If

IsDescendant

現在のメンバーまたは指定したメンバーが、指定した親の子孫であるかどうかを調べます。この関数は、次のタイプのルールで使用できます:

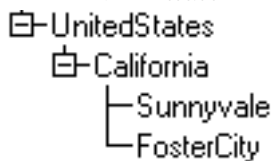
- 計算
- 換算
- 連結
- 配賦



注:

メンバーがツリー階層で親の下位レベルにある場合、そのメンバーは子孫です。子孫は、ツリーの同じ分岐内にあります。

たとえば、次の階層では、FosterCityとSunnyvaleはCaliforniaおよびUnitedStatesの子孫です。



構文

```
HS.  
<Object>  
.IsDescendant  
(  
"  
Parent  
", "  
Element  
"  
)
```

```
HS.Node.IsDescendant("  
Parent  
", "  
Entity  
", "S#
```

```
Scenario.
Y#
Year.
P#
Period
")
```

```
HS.
<Object>
.IsDescendant("
Parent
", "")
```

```
HS.Custom(
Dimension
).IsDescendant(
Member
)
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表66 IsDescendant関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> • Account • Custom1...4 • Custom (カスタム・ディメンション・ラベル) • Entity • Parent
Parent	有効な親メンバーの名前。親は必須です。
Element	次のいずれかのディメンションの有効なメンバーの名前。 選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> • Account • Custom1...4 • Custom • Entity • Parent
Entity	エンティティ・ディメンションの有効なメンバーの名前。
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。

パラメータ	説明
Period	有効な期間。



注:

ノードをオブジェクトとして使用する場合は、メンバーが指定した親のアクティブな子孫であるかどうか判断されます。

戻り値

要素が指定した親の子孫である場合にTRUEとなり、そうでない場合にFALSEとなるブール式。

ノードの場合は、要素が指定した親のアクティブな子孫である場合にTRUEとなり、そうでない場合にFALSEとなります。

例

この例では、ConnecticutがRegionalの子孫であると、If...Then行とEnd If行の間にあるすべての文が実行されます。

```
If HS.Entity.IsDescendant("Regional", "Connecticut") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Prod").IsDescendant("All_Phones", P3000_Phones)
```

```
...
```

```
End If
```

IsFirst

現在の期間または年が、アプリケーションの最初の期間または年であるかどうかを調べます。現在の期間または年がアプリケーションの最初の期間または年であるかどうかの判別に、現在のシナリオのデフォルト期間単位が使用されます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.  
<Object>  
.IsFirst
```

<Object>は、次のキーワードのいずれかです。

- Period
- Year

戻り値

現在の期間または年がアプリケーションの最初の期間または年である場合にTRUEとなり、そうでない場合にFALSEとなるブール式。

例

この例では、現在の期間が最初の期間であると、If...Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.Period.IsFirst = TRUE Then
```

```
...
```

```
End If
```

IsICP

現在の勘定科目ディメンション・メンバーまたはエンティティ・ディメンション・メンバー、あるいは指定した勘定科目メンバーまたはエンティティ・メンバーが内部取引パートナー(ICP)であるかどうかを調べます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

- ・ 配賦

構文

```
HS.
<Object>
.IsICP("
Element
")
```

```
HS.
<Object>
.IsICP("")
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表67 IsICP関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> • Account • Entity
Element	勘定科目ディメンションまたはエンティティ・ディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> • 勘定科目 • エンティティ

戻り値

勘定科目またはエンティティのメンバーが内部取引パートナーである場合にTRUEとなり、そうでない場合にFALSEとなるブール式。

例

この例では、売上勘定科目が内部取引パートナーであると、If...Then行とEnd If行の間にあるすべての文が実行されます。

```
If HS.Account.IsICP("Sales") = TRUE Then
```

```
...
```

```
End If
```

IsLast

現在の期間または年がアプリケーションの最後の期間または年であるかどうかを調べます。この判別には、現在のシナリオのデフォルト期間が使用されます。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.  
<Object>  
.IsLast
```

<Object>は、次のキーワードのいずれかです。

- Period
- Year

戻り値

現在の期間または年がアプリケーションの最後の期間または年である場合にTRUEとなり、そうでない場合にFALSEとなるブール式。

例

この例では、現在の期間が最後の期間であると、If...ThenとEnd Ifの文の間にあるすべての文が実行されます。

```
If HS.Period.IsLast = TRUE Then
```

```
...
```

```
End If
```

IsTransCur

現在の値ディメンション・メンバーが、換算済の通貨メンバーであるかどうかを調べます。この関数は計算ルールで使用できます。

構文

```
HS.Value.IsTransCur
```

戻り値

現在の値メンバーが換算された通貨メンバーである場合にTRUEとなり、そうでない場合にFALSEとなるブール式。

例

この例では、値メンバーが換算済の通貨メンバーであると、If...Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.Value.IsTransCur = TRUE Then
```

```
...
```

```
End If
```

IsTransCurAdj

現在の値ディメンション・メンバーが、換算済の通貨調整メンバーであるかどうかを調べます。この関数は計算ルールで使用できます。

構文

```
HS.Value.IsTransCurAdj
```

戻り値

現在の値メンバーが換算された通貨調整メンバーである場合にTRUEとなり、そうでない場合にFALSEとなるブール式。

例

この例では、値メンバーが換算済の通貨調整メンバーであると、If...Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.Value.IsTransCurAdj = TRUE Then
```

```
...
```

```
End If
```

IsValidDest

指定した視点が有効な宛先かどうかを判別します。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結
- 配賦



注：

この関数では、セルが計算されたセルであるかどうかはチェックされません。

構文

```
HS.IsValidDest("
POVExpression
")
```

POVExpressionは視点(POV)です。ディメンションを指定しないと、次のデフォルト値が使用されます。

- Account - Sub Consolidateサブルーチンで使用される場合は現在の勘定科目メンバー。それ以外の場合は、勘定科目は必須です。
- CustomおよびICP - Sub Consolidateサブルーチンで現在のメンバーが使用されます。それ以外の場合は、勘定科目のTopMemberが使用されます。
- Scenario - 現在のシナリオ・メンバー
- Entity - 現在のエンティティ・メンバー
- Value - 現在の値メンバー

- YearおよびPeriod - 現在のメンバー

戻り値

指定した視点が有効な宛先である場合はTRUE、そうでない場合はFALSEとなるブール値。

例

この例では、指定した宛先が有効であると、If...Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.IsValidDest("A#Sales.I#CT.C1#P1.C2#R1.C3#[None].C4#[None]") = TRUE Then
```

```
...
```

```
End If
```

IsZero

渡された値がゼロに近いかどうかを、事前に定義されているFinancial Managementイプシロンに基づいてチェックします。この関数は、すべてのタイプのルールで使用できます。

この関数は、ゼロとの正確な比較のかわりに推奨されます。正確な比較では、浮動小数点算術によって、無視できる 1×10^{-10} より小さいエラーを引き起こします。

使用しない場合:

```
Difference = Value1 - Value2
```

```
If Difference = 0 Then
```

```
    'process where Difference = 0
```

```
Else
```

```
    'process where Difference <> 0
```

```
End If
```

使用する場合:

```
Difference = Value1 - Value2
```

```
If HS.IsZero(Difference) Then
```

```
    'process where Difference = 0
```

```
Else
```

```
    'process where Difference <> 0
```

```
End If
```

構文

```
BooleanValue = HS.IsZero(Value)
```

戻り値

渡された値がゼロに近い場合にTRUEになるブール値。それ以外の場合は、FALSEになります。

例

```
Dim BoolVal
```

```
Dim Value
```

```
Value = 0.000000001
```

```
BoolVal = HS.IsZero(Value)
```

```
If BoolVal = true Then
```

```
// do processing
```

```
Else
```

```
// do Processing
```

```
End If
```

List

指定したリストにある要素を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.  
<Object>  
.List("  
Parent  
"."  
Listname  
")
```

```
HS.Node.List("  
Parent  
"."  
Listname  
"."S#  
Scenario  
.Y#  
Year.P#Period  
")
```

```
HS.Custom("  
Dimension"  
).List("  
Parent
```



```
"
"/
>Listname
")
```

表68 List関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> • Account • Custom1...4 • Custom (カスタム・ディメンション・ラベル) • Entity • Parent • ICP • Scenario
Parent	有効な親メンバーの名前。
Listname	有効なシステム・リストまたはユーザー定義リストの名前。 注: ノードの場合は、有効なエンティティ・システム・リストが必須。
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。

戻り値

指定したリストのすべての要素が含まれている配列。ノードの場合は、リスト内のアクティブな要素のみ。

例

この例では、現在の勘定科目のユーザー定義リストMyBaseListの要素を取得します。

```
HS.Account.List("", "MyBaseList")
```

この例では、TotalAssets勘定科目のシステム・リスト[Base]の要素を取得します。

```
HS.Account.List("TotalAssets", "[Base]")
```

```
Hs.Custom("Prod").List("Products", "[Base]")
```

Member

現在のメンバーの名前を取得します。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.  
<Object>  
.Member
```

<Object>は、次のキーワードのいずれかです。

- Entity
- Parent
- 期間
- シナリオ
- 値
- Year
- View

戻り値

現在のメンバー名が含まれている文字列。

値オブジェクトの場合、メンバーは、値に関連付けられている通貨ではなく現在の値メンバーの名前を戻します。たとえば、現在の値メンバーがエンティティ通貨で、値がUSD通貨に関連付けられている場合、HS.Parent.MemberはUSDではなく、エンティティ通貨を戻します。



ヒント：

現在の値メンバーに設定されている通貨を取得するには、DefCurrency関数を使用します。

例

この例では、現在のエンティティがCaliforniaであると、If...ThenとEnd Ifの文の間にあるすべての文が実行されます。

```
If HS.Entity.Member = "California" Then
```

```
...
```

```
End If
```

MemberFromID

指定したID番号のディメンション・メンバーを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.  
<Object>  
.MemberFromID(  
  ElementID  
)
```

```
HS.Custom(  
  Dimension  
) .MemberFromID(  
  ElementID  
)
```

表69 MemberFromID関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none">• Account• Custom1...4• Custom (カスタム・ディメンション・ラベル)• Entity• ICP• Parent• Period

パラメータ	説明
	<ul style="list-style-type: none"> • Scenario • Value • Year • View
Parent	有効な親メンバー。
ElementID	<p>次のいずれかのディメンションの有効なメンバーのID番号。選択したオブジェクトによって異なります。</p> <ul style="list-style-type: none"> • Account • Custom1...4 • Custom • Entity • ICP • Parent • Period • Scenario • Value • Year

戻り値

ディメンション・メンバー名。

例

この例では、ID番号001のメンバーを取得します。

```
strEntity = HS.Entity.MemberFromID(001)
```

```
strCustom = HS.Custom("Prod").MemberFromID(001)
```

Method

指定したメンバーの連結メソッドを取得します。ゼロ以外の値が複数あると、その中の最初に見つかった値が戻されます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算

- 連結
- 配賦

構文

```
HS.Node.Method("
POVExpression
")
```

POVExpressionは、シナリオの組合せです。年、期間およびエンティティのメンバー。

戻り値

指定した視点の連結メソッドを示す文字列。

例

この例では、視点のメソッドがGLOBALであると、If...Then文とEnd If文間にあるすべての文が実行されます。

```
    If HS.Node.Method("S#Actual.Y#2014.P#January.E#Regional.Connecticut") =
"GLOBAL" Then
```

```
    ...
```

```
End If
```

NoInput

特定のセルまたはセルのスライスにユーザーがデータを入力できないようにします。この関数は、入力または計算済ではない勘定科目がある場合に便利です。

計算済勘定科目を設定すると、実際にユーザーは計算済勘定科目に入力できなくなります。ただし、データ入力の一部のディメンション交差には可能で、その他のディメンション交差には不可である勘定科目を持っている場合は、NoInputを使用できます。この関数は、NoInputルールで使用できます。

構文

```
HS.NoInput "
POVExpression
"
```

POVExpressionは視点(POV)です。

戻り値

なし。

例

この例では、Sales勘定科目と2014年のBudgetシナリオが交差するセルへの入力禁止されます:

```
Sub NoInput
```

```
HS.NoInput "S#Budget.Y#2014.A#Sales"
```

```
End Sub
```

NoRound

後に続くすべてのExp文の丸め処理をオフにします。この関数は、次のタイプのルールで使用できます:

- ・ 計算
- ・ 換算
- ・ 配賦



ヒント:

Round関数の引数として0を入力しても、丸め処理をオフにできます。たとえば、HS.Round(0)で丸め処理がオフになります。

構文

```
HS.NoRound
```

戻り値

なし。

例

この例では、SalesRound勘定のセルに挿入された金額を小数第2位で丸め、次にNoRound関数を使用して、SalesNoRound勘定のセルに挿入された金額の丸め処理をオフにします。

```
HS.Round 0.1
```

```
HS.Exp "A#SalesRound" = "A#Sales"
```

```
HS.NoRound
```

```
HS.Exp "A#SalesNoRound" = "A#Sales"
```

NumBase

現在のメンバーまたは指定したメンバーの基本メンバーの数を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦



注:

基本メンバーとは、子を持たない(ツリー階層の一番下にある)メンバーのことです。

構文

```
HS.  
<Object>  
.NumBase("Element")
```

```
HS.Node.NumBase("S#Scenario.Y#Year.P#Period")
```

```
.E#  
Entity  
")
```

```
HS.  
<Object>  
.NumBase("")
```

```
HS.Custom(  
Dimension  
) .NumBase(  
Member  
)
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表70 NumBase関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none">• Account• Custom1...4• Custom (カスタム・ディメンション・レベル)• Entity• Parent• Node
Element	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none">• Account• Custom1...4• Custom• Entity• Parent• Node AccountオブジェクトとCustomオブジェクトを使用する場合は、メンバーを指定する必要があります。空白の文字列は使用できません。ディメンション全体の基本メンバーの数を取得するには、次の例のように、ALLを引用符で囲んで指定します。iAcctBase = HS.Account.NumBase("ALL")

パラメータ	説明
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。
Entity	エンティティ・ディメンションの有効なメンバーの名前。

Exp関数にNumBase関数を組み込むこともできます。NumBase関数を組み込む場合は、次の例のように、NumBaseの引数を引用符で囲まないでください。

```
HS.Exp "A#AverageSales = A#Sales/HS.Entity.NumBase(Regional)"
```

戻り値

基本メンバーの数を識別するLONG。NODEの場合は、指定したメンバーのアクティブな基本要素の数を取得します。



注:

1つの分岐に基本エンティティが2回表示される場合は、2個あるとみなされます。

例

この例では、Regionalエンティティ下に基本エンティティの平均売上高を保管するSAlesAllocという名前の勘定科目がアプリケーションに含まれています。SalesAllocの金額を計算するために、Sales勘定の金額を、Regional下の基本エンティティの数で割っています。

```
If HS.Exp"A#SalesAlloc = A#Sales/HS.Entity.NumBase(Regional)" then
```

```
...
```

```
End If
```

```
If HS.Exp"ASalesAlloc = A#Sales/HS.Custom("Prod").NumBase("TotalProducts")
```

```
...
```

```
End If
```

Number

現在の期間の番号を取得します。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.View.PeriodNumber
```

戻り値

現在の期間の番号。

例

この例では、現在の期間が最初の期間であると、If...ThenとEnd Ifの文の間にあるすべての文が実行されます。

```
If HS.View.PeriodNumber = 1 Then
```

```
...
```

```
End If
```

NumChild

現在のディメンション・メンバーまたは指定したメンバーの子メンバーの数を取得します。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結

- 配賦



注:

メンバーがツリー階層で別のメンバーのすぐ1つ下のレベルにある場合、そのメンバーは子です。指定したオブジェクトの1つ下のメンバーのみがカウントされます。

構文

```
HS.
<Object>.
NumChild("
Element
")
```

```
HS.Node.NumChild("S#
Scenario
.Y#
Year
.P#
Period
.E#
Entity
")
```

```
HS.
<Object>
.NumChild("")
```

```
HS.Custom(
Dimension
).NumChild(
Member
)
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表71 NumChild関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> • Account

パラメータ	説明
	<ul style="list-style-type: none"> • Custom1-4 • Custom (カスタム・ディメンション・ラベル) • Entity • Node • Parent
Element	<p>次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。</p> <ul style="list-style-type: none"> • Account • Custom1...4 • Custom • Entity • Node • Parent <p>AccountオブジェクトとCustomオブジェクトを使用する場合は、メンバーを指定する必要があります。空白の文字列は使用できません。ディメンション全体の子メンバーの数を取得するには、次の例のように、ALLを引用符で囲って指定します。 <code>iAcctBase = HS.Account.NumChild("ALL")</code></p>
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。
Entity	エンティティ・ディメンションの有効なメンバーの名前。

NumChild関数をExp関数に組み込むことができます。NumChild関数を組み込む場合は、NumChildの引数を引用符で囲わないでください。

戻り値

子メンバーの数を識別する文字列。NODEの場合は、指定したメンバーのアクティブな子の数を取得します。

例

この例では、Regionalエンティティ直下にエンティティの平均売上高を保管するSalesChildという名前の勘定科目がアプリケーションに含まれています。SalesChildの金額を計算するために、Sales勘定の金額を、Regional直下の子の数で割っています。

```
HS.Exp "A#SalesChild = A#Sales/HS.Entity.NumChild(Regional)"
```

NumCustom

アプリケーションに定義されているカスタム・ディメンションの合計数を戻します。たとえば、5つのカスタム・ディメンションを作成して、1つのカスタム・ディメンションを削除し、2つのカスタム・ディメンションを追加する場合、この関数で戻される値は、アプリケーションに定義されているカスタム・ディメンションの合計数である6です。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.NumCustom
```

戻り値

アプリケーションのカスタム・ディメンションの数。

例

```
nCustoms = HS.NumCustom
```

NumDescendant

現在のディメンション・メンバーまたは指定したメンバーの子孫の数を取得します。この関数は、次のタイプのルールで使用できます:

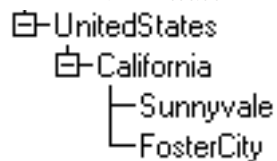
- 計算
- 換算
- 連結
- 配賦



注:

メンバーがツリー階層で親の下位レベルにある場合、そのメンバーは子孫です。子孫は、ツリーの同じ分岐内にあります。

たとえば、次の階層では、FosterCityとSunnyvaleはCaliforniaおよびUnitedStatesの子孫です。



構文

```

HS.
<Object>.
NumDescendant("
Element
")
  
```

```

HS.Node.NumDescendant("S#
Scenario
.Y#
Year
.P#
Period
.E#
Entity
")
  
```

```

HS.
<Object>
.NumDescendant("")
  
```

```

HS.Custom(
Dimension
).NumDescendant(
Member
)
  
```



注:

関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表72 NumDescendant関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> • Account • Custom1...4 • Custom (カスタム・ディメンション・ラベル)

パラメータ	説明
	<ul style="list-style-type: none"> Entity Node Parent
Element	<p>次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。</p> <ul style="list-style-type: none"> Account Custom1...4 Custom Entity Node Parent <p>AccountオブジェクトとCustomオブジェクトを使用する場合は、メンバーを指定する必要があります。空白の文字列は使用できません。ディメンション全体の子孫の数を取得するには、次の例のように、ALLを引用符で囲って指定します。<code>iAcctBase = HS.Account.NumDescendant("ALL")</code></p>
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。
Entity	エンティティ・ディメンションの有効なメンバーの名前。

戻り値

指定したメンバーの子孫の数。NODEの場合は、指定したメンバーの下のアクティブな子孫の数を取得します。



注:

1つの分岐に子孫エンティティが2回表示される場合は、2個あるとみなされます。

例

この例では、エンティティFranceに子孫がないと、If...Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.Entity.NumDescendant("France") = 0 Then
```

```
...
```

```
End If
```

```
If HS.Custom("Product").NumDescendant("P3000-Phones") = 0 Then
```

```
...
```

```
End If
```

NumPerInGen

処理中の現在の期間に対する世代の期間の数を取得します。この関数は、Dynamic SUB関数で使用できます。

構文

```
HS.Period.NumPerInGen
```

戻り値

ビューの期間の数を表す値。

例

```
HS.Dynamic "A#MarginPct=A#GrossMargin/HS.Period.NumPerInGen"
```

たとえば、現在の期間が4月で、4月がカレンダー・ファイルの4番目の世代にある場合(月ごとの世代)は、月ごとの世代の期間の数は12です。現在の期間が、カレンダー・ファイルの3番目の世代にあるQ2である場合(四半期世代)は、期間の数は4です。

月ごとの世代(4番目の世代):

1月、2月、3月、4月、5月、6月、7月、8月、9月、10月、11月、12月

この世代の期間の数として12が戻されます。

四半期世代(3番目の世代):

Q1、Q2、Q3、Q4

この世代の期間の数として4が戻されます。

半期世代(2番目の世代):

HY1、HY2

この世代の期間の数として2が戻されます。

年次世代(1番目の世代):

年

この世代の期間の数として1が戻されます。

NumPeriods

指定したシナリオの期間単位に対して定義されている期間の数を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Scenario.NumPeriods("
ScenarioName
")
```

または

```
HS.Scenario.NumPeriods("")
```

または

```
HS.Scenario.NumPeriods(
Var1
)
```

戻り値

期間単位に対する期間の数を表す数値。たとえば、シナリオが月次の場合は、期間の数として12が戻されます。シナリオが四半期ごとの場合は、期間の数として4が戻されます。

例

この例では、実績シナリオの期間単位に対して定義されている期間の数が戻されます。

```
HS.Scenario.NumPeriods("Actual")
```

OpenCellTextUnit

複数のセルのセル・テキストを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

セル・テキストを取得するPOVの一部を指定できます(例: POV全体ではなく、シナリオと年)。ページ・ディメンション・メンバー(シナリオ、年、期間、値、エンティティ)を指定しない場合、現在のメンバーのセル・テキストのみが戻されます。サブキューブ・ディメンション(勘定科目、ICP、カスタム)を指定しない場合、すべての基本および親メンバーのセル・テキストが戻されます。シナリオおよび年のメンバーを指定して、現在のサブキューブではないサブキューブのセル・テキストを取得できます。期間、値および/またはエンティティのメンバー・リストを指定して、複数のサブキューブのセル・テキストを取得することもできます。セル・テキスト・ラベルを取得するよう指定し、ディメンション・メンバーまたはセル・テキスト・ラベルの昇順または降順で、セル・テキストをソートするかどうかを指定することもできます。ソートの値を指定しない場合、セル・テキスト単位でソートが実行されず、データベース内の順序に基づいて結果が戻されます。

構文

```
Set CTU = HS.OpenCellTextUnit(POVExpression, Label(s), Sort_Dimension, Sort_Order)
```

POVExpressionはPOV、Labelはセル・テキスト・ラベル(なし、1つまたは複数)、Sort_Orderは昇順または降順です。

戻り値

指定したPOVのすべてのセル・テキストおよびラベルの情報が戻されます。

例

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014", "", "", "")
```

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014.P{.[Base]}.A{.[Base]}", "", "Period", "Ascending")
```

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014.E{.[Base]}.P{.[Base]}", "CouponRate", "LABEL", "Ascending")
```

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014.E{.[Base]}.P{.[Base]}.A#Purchases", "", "Brands", "Descending")
```

複数のラベルのセル・テキストを取得する場合は、カンマ区切り値としてラベルを入力できます。ラベルの値を指定しない場合、すべてのラベルが取得されます。

例

```
Set CTU = HS.OpenCellTextUnit("A#Sales", "Label-1, Label-2", "Label", "Descending")
```

OpenDataUnit

連結、計算または換算時に処理するデータ単位を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

構文

```
HS.OpenDataUnit(  
    POVExpression  
)
```

ここで、POVExpressionはPOVです。POV式の一部として、関数ではAccount、ICP、C1、C2、C3およびC4のユーザー定義リストとシステム・リストがサポートされています。

戻り値

この関数をSub Consolidateサブルーチンで使用すると、データのあるすべてのレコードが戻されますが、勘定科目は連結勘定として指定されている勘定科目のみが戻されます。

Sub CalculateまたはSub Translateサブルーチンで使用すると、連結勘定として設定されている勘定科目を含め、データのあるすべてのレコードが戻されます。



注:

勘定科目は、IsConsolidated属性の値がTRUEである場合に連結されます。

例

```
Set DataUnit=HS.OpenDataUnit{"A{TotalRev.[Base]}.C1{C1Top.[Base]}.C2{MyC2List}.C3#[None]"}
```

OpenDataUnitSorted

計算、換算または連結時に処理するデータ単位を、指定した順序にデータをソートして取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

構文

```
HS.OpenDataUnitSorted(  
  POVExpression, <dimension to be sorted>, Ascending  
  or  
  Descending  
)
```

ここで、POVExpressionはPOV文字列、<dimension to be sorted>はディメンション名の文字列であり、次の6つのディメンションのいずれかのみを指定できます: AccountまたはA、ICPまたはI、Custom1またはC1、Custom2またはC2、Custom3またはC3、Custom4またはC4。次のいずれかを指定する必要があります: Account、ICP、Custom1、Custom2、Custom3、Custom4。3番目のパラメータは文字列値("Ascending"または"A"、または"Descending"または"D")です。

例

```
Set DataUnit= HS.OpenDataUnitSorted("S#Actual.E#Group1","C1","Ascending")
```

Owned

現在処理中のエンティティの組合せの所有エンティティを取得します。この関数は、エクイティ・ピックアップ・ルールで使用されます。

構文

```
HS.Entity.Owned
```

戻り値

所有エンティティ。

例

```
Owned=HS.Entity.Owned
```

Owner

現在処理中のエンティティの組合せの所有者を取得します。この関数は、エクイティ・ピックアップ・ルールで使用されます。

構文

```
HS.Entity.Owner
```

戻り値

所有者エンティティ。

例

```
Owner=HS.Entity.Owner
```

PCon

現在のメンバーまたは指定したメンバーの連結比率を取得します。この関数は、次のタイプのルールで使用できます:

- ・ 計算
- ・ 換算
- ・ 連結
- ・ 配賦

構文

```
HS.Node.PCon("S#
Scenario
.Y#
Year
.P#
Period
.E#
Entity
")
```

```
HS.Node.PCon("")
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表73 PCon関数の構文

パラメータ	説明
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。
Entity	エンティティ・ディメンションの有効なメンバーの名前。

戻り値

メンバーの連結比率。

例

この例では、指定した視点の連結比率が取得されます。

```
Pcon = HS.Node.PCon("S#Actual.Y#2014.P#January.E#Regional.Connecticut")
```

PEPU

エクイティ・ピックアップ(EPU)表から出資比率を取得します。この関数は、エクイティ・ピックアップ・ルールで使用されます。

構文

```
HS.PEPU("S#.Y#.P#", Owner, Owned)
```

戻り値

EPU表の出資比率。

例

```
HS.PEPU(S#Actual.Y#2014.P#Jan, Group, CT)
```

または

```
HS.PEPU("", "", "")
```

デフォルト・パラメータ: 値が空白の場合、この関数は現在のシナリオ、年、期間のエンティティの組合せの出資比率を戻します。

例

```
Sub EquityPickUp()
```

```
Owned=Hs.Entity.Owned
```

```
OwnerCurrencyTotl=Hs.Entity.DefCurrency & "Total"
```

```
Hs.Clear "A#Inv.C4#EPU.I#" & Owned
```

```
Hs.Exp "A#Inv.C4#EPU.I#" & Owned & "=A#EQ.C4#C3Tot.I#[ICPTot].E#" &
```

Owned & “.V#” & OwnerCurrencyTotl & “*” & Hs.PEPU (,,,))

End Sub

PeriodNumber

取得中のデータのビューの期間番号を取得します。この関数は、Dynamic SUB関数で使用できます。

構文

HS.View.PeriodNumber

戻り値

ビューの期間の数を表す値。

例

Hs.Dynamic "A#MarginPct=A#GrossMargin/HS.View.PeriodNumber"

	期間	YTD	QTD	HYTD
1月	1	1	1	1
2月	1	2	2	2
3月	1	3	3	3
第1四半期	1	1	1	1
4月	1	4	1	4
5月	1	5	2	5
6月	1	6	3	6
第2四半期	1	2	1	2
HY1	1	1	1	1
7月	1	7	1	1
8月	1	8	2	2
9月	1	9	3	3
第3四半期	1	3	1	1
10月	1	10	1	4
11月	1	11	2	5

	期間	YTD	QTD	HYTD
12月	1	12	3	6
第4四半期	1	4	1	2
HY2	1	2	2	1
年	1	1	1	1

PlugAcct

現在の勘定科目メンバーまたは指定した勘定科目の調整勘定科目を取得します。この関数は、次のタイプのルールで使用できます:

- ・ 計算
- ・ 換算
- ・ 連結
- ・ 配賦

構文

```
HS.Account.PlugAcct("
Account
")
```

```
HS.Account.PlugAcct("")
```

Accountは、有効な勘定科目ディメンション・メンバーの名前です。



注:

Sub Consolidateサブルーチンで関数を使用している場合にのみ、空白の文字列("")を使用して、この関数を現在のメンバーに適用できます。

戻り値

メンバーの調整勘定科目の名前を表す文字列。

例

この例では、Sales勘定の調整勘定科目がPlug1であると、If...Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.Account.PlugAcct("Sales") = "Plug1" Then
```

```
...
```

```
End If
```

POwn

現在のメンバーまたは指定したメンバーの最大出資比率を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Node.POwn("S#  
Scenario  
.Y#  
Year  
.P#  
Period  
.E#  
Entity  
")
```

```
HS.Node.POwn("")
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表74 POwn関数の構文

パラメータ	説明
<i>Scenario</i>	シナリオ・ディメンションの有効なメンバーの名前。
<i>Year</i>	有効な年。
<i>Period</i>	有効な期間。

パラメータ	説明
<i>Entity</i>	エンティティ・ディメンションの有効なメンバーの名前。

戻り値

メンバーの出資比率。

例

この例では、指定した視点の出資比率が取得されます。

```
POwn = HS.Node.POwn("S#Actual.Y#2014.P#January. E#Regional.Connecticut")
```

PVAForBalance

BALANCE勘定科目(ASSETおよびLIABILITY勘定科目)のデフォルト換算メソッドを指定します。この関数は、換算中に通貨換算のアプリケーション・デフォルトを上書きします。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.AppSettings.PVAForBalance
```

戻り値

BALANCE勘定が期別値(PVA)換算メソッドを使用する場合にTRUEとなり、為替レート値(VAL)換算メソッドを使用する場合にFALSEとなるブール式。

例

この例では、アプリケーションのBALANCE勘定で期別値換算メソッドが使用されると、If...Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.AppSettings.PVAForBalance = TRUE Then
```

```
...
```

```
End If
```

PVAForFlow

FLOW勘定(REVENUEおよびEXPENSE勘定)のデフォルト換算メソッドを指定します。この関数は、換算中に通貨換算のアプリケーション・デフォルトを上書きします。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.AppSettings.PVAForFlow
```

戻り値

FLOW勘定が期別値(PVA)換算メソッドを使用する場合にTRUEとなり、為替レート値(VAL)換算メソッドを使用する場合にFALSEとなるブール式。

例

この例では、アプリケーションのFLOW勘定科目で為替レート換算メソッドの値が使用されると、If...ThenとEnd Ifの間にあるすべての文が実行されます。

```
If HS.AppSettings.PVAForFlow = FALSE Then
```

```
...
```

```
End If
```

RateForBalance

BALANCE勘定(ASSETおよびLIABILITY勘定科目)で使用するデフォルト換算レート勘定科目を取得します。この関数は、次のタイプのルールで使用できます:

- ・ 計算
- ・ 換算
- ・ 連結
- ・ 配賦

構文

```
HS.AppSettings.RateForBalance
```

戻り値

BALANCE勘定で使用するデフォルトの換算レートが含まれているレート勘定科目を表す文字列。

例

この例では、BALANCE勘定のデフォルト換算レート勘定科目がRate1であると、If..Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.AppSettings.RateForBalance = "Rate1" Then
```

```
...
```

```
End If
```

RateForFlow

FLOW勘定(REVENUEおよびEXPENSE勘定)で使用するデフォルト換算レート勘定を取得します。この関数は、次のタイプのルールで使用できます:

- ・ 計算
- ・ 換算
- ・ 連結
- ・ 配賦

構文

```
HS.AppSettings.RateForFlow
```

戻り値

FLOW勘定で使用するデフォルトの換算レートが含まれているレート勘定科目を表す文字列。

例

この例では、FLOW勘定のデフォルト換算レート勘定がレート2であると、If...Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.AppSettings.RateForFlow = "Rate2" Then
```

```
...
```

```
End If
```

ReviewStatus

指定した視点の確認ステータスを取得します。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.ReviewStatus("S#  
Scenario  
.Y#  
Year  
.P#  
Period  
.E#  
Entity  
.V#  
Value  
")
```

```
HS.ReviewStatus("")
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表75 ReviewStatus関数の構文

パラメータ	説明
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。
エンティティ	エンティティ・ディメンションの有効なメンバーの名前。
Value	値ディメンションのメンバーの名前。

戻り値

メンバーの確認ステータスを示す文字列。有効な確認ステータスは次のとおりです。

- ・ 開始していません
- ・ 第1パス
- ・ 確認レベル1-10
- ・ 提出済
- ・ 承認済
- ・ 発行済
- ・ サポートされていません

例

この例では、指定した視点の確認ステータスが「提出済」であると、If...Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.ReviewStatus("") = "Submitted" Then
```

```
...
```

```
End If
```

ReviewStatusUsingPhaseID

プロセス管理の提出フェーズIDを使用して、指定した視点の確認ステータスを取得します。

構文

```
HS.ReviewStatusUsingPhaseID("S#  
Scenario  
.Y#  
Year  
.P#  
Period  
.E#  
Entity  
",  
n  
)
```

*n*は、プロセス管理の提出フェーズを表す整数です。有効な値は1-9です。

戻り値

提出フェーズIDを使用してメンバーの確認ステータスを示す文字列。有効な確認ステータスは次のとおりです。

- ・ 開始していません
- ・ 第1パス
- ・ 確認レベル1-10
- ・ 提出済
- ・ 承認済
- ・ 発行済
- ・ サポートされていません

例

```
HS.ReviewStatusUsingPhaseID("S#Actual.Y#2014.P#January.E#Connecticut",3)
```

Round

Exp関数から戻されたデータを丸めます。丸め処理の位置を引数で指定します。この関数は、次のタイプのルールで使用できます:

- ・ 計算
- ・ 換算

- 連結



ヒント:

計算ルールで、状況に応じて丸め処理の位置を変える必要がある場合は、Roundを含む複数の文を使用できます。

構文

```
HS.  
Round  
(  
Unit  
)
```

Unitは丸め処理の係数です。値1は、最も近い整数に四捨五入します。値0.1は、最も近い10分の1の値に丸めます。値0は丸め処理をオフにします。

この引数に0を指定すると、計算ルール内の後続のすべてのExp関数に対して丸め処理がオフになります。この構文は、HS.NoRound: HS.Round(0)と同じ効果を持ちます。



注意

勘定科目のNumDecimalPlaces属性によって、小数点の右側の最大桁数が決まります。Round関数は、この属性を上書きしません。

戻り値

なし。

例

この例では、SalesRound勘定のセルに挿入された金額を小数第2位で丸め、次にNoRoundを使用して、SalesNoRound勘定のセルに挿入された金額の丸め処理をオフにします。

```
HS.Round(0.1)
```

```
HS.Exp"A#SalesRound = A#Sales"
```

```
HS.NoRound
```

```
HS.Exp"A#SalesNoRound = A#Sales"
```

Scale

指定した通貨の位取りを取得します。この関数は、次のタイプのルールで使用できます：

- ・ 計算
- ・ 換算
- ・ 連結
- ・ 配賦

構文

```
HS.Currency.Scale("
Currency
")
```

```
HS.Currency.Scale(
Var1
)
```

表76 Scale関数の構文

パラメータ	説明
<i>Currency</i>	有効な通貨の名前。
<i>Var1</i>	VisualBasic変数。

戻り値

指定した通貨の位取りを示す数字(0-9)。小数点の位置を指定して、通貨の金額を表示および保管するときの単位を指定します。戻り値は次のとおりです。

- ・ 0 = 1単位
- ・ 1 = 十
- ・ 2 = 百
- ・ 3 = 千
- ・ 4 = 1万
- ・ 5 = 10万
- ・ 6 = 百万
- ・ 7 = 1千万

- 8 = 1億
- 9 = 十億

例

この例では、フランス・フラン(FF)の位取りが3であると、If...Then文とEnd If文間にあるすべての文が実行されます。

```
If HS.Currency.Scale("FF") = 3 Then
```

```
...
```

```
End If
```

SecurityAsPartner

エンティティまたは親が内部取引パートナーとして使用される場合に、指定されたエンティティまたは親に割り当てられるセキュリティ・クラスを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Entity.SecurityAsPartner(  
Entity  
)
```

```
HS.Entity.SecurityAsPartner("")
```

```
HS.Entity.SecurityAsPartner(  
Var1  
)
```

```
HS.Parent.SecurityAsPartner(  
Entity  
)
```



注:

この関数を現在のエンティティ・メンバーに適用するには、空白の文字列("")を使用します。

表77 SecurityAsPartner関数の構文

パラメータ	説明
<i>Entity</i>	エンティティ・ディメンションの有効なメンバーの名前。
<i>Var1</i>	VisualBasic変数。

戻り値

ICPとして使用する場合に、エンティティまたは親に割り当てられているセキュリティ・クラスを示す文字列。

例

この例では、内部取引パートナーとして使用されるときClass1がFranceのセキュリティ・クラスである場合に、If...ThenとEnd Ifの文間にある文が実行されます。

```
If HS.Entity.SecurityAsPartner("France") = "Class1" Then
```

```
...
```

```
End If
```

```
If HS.Parent.SecurityAsPartner("France") = "Class1" Then
```

```
...
```

```
End If
```

SecurityClass

指定されたメンバーのセキュリティ・クラスを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

- 配賦

構文

```
HS.<
Object
>.SecurityClass("
Element
")
```

```
HS.<
Object
>.SecurityClass("")
```

```
HS.<
Object
>.SecurityClass(
Var1
)
```

```
HS.Custom(
Dimension
).SecurityClass("
Element
")
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表78 SecurityClass関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> • Account • Scenario • Entity • Parent • Custom1...4 • Custom (カスタム・ディメンション・ラベル)
Element	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> • 勘定科目

パラメータ	説明
	<ul style="list-style-type: none"> シナリオ エンティティ Custom1...4 カスタム
<i>Var1</i>	VisualBasic変数。

戻り値

指定したメンバーに割り当てられているセキュリティ・クラスの名前。

例

この例では、Class1がCash勘定科目に割り当てられたセキュリティ・クラスである場合に、If...ThenとEnd Ifの文間にある文が実行されます。

```
If HS.Account.SecurityClass("Cash") = "Class1" Then
```

```
...
```

```
End If
```

```
If HS.Custom(  
Dimension  
) .SecurityClass(  
Member  
) Then
```

```
...
```

```
End If
```

SetCellTextWithLabel

指定されたPOVおよびセル・テキスト・ラベルのセル・テキスト情報を更新します。この関数は、次のタイプのルールで使用できます：

計算

構文

```
HS.SetCellTextWithLabel("POVExpression", "CellTextLabel", "CellTextString")
```

POVExpressionはメンバーの組合せで、CellTextLabelはデフォルトのセル・テキスト・ラベル([Default])か、ロードされた有効なセル・テキスト・ラベルの1つです。CellTextStringは、書き込まれるテキスト文字列です。Account、ICPおよびCustom POVメンバーを指定する必要があります。シナリオ、年、期間、ビュー、エンティティおよび値は、デフォルトで現在のメンバーに設定されます。

戻り値

なし

例

```
HS.SetCellTextWithLabel "A#Asset.I#[ICP  
None].C1#[None].C2#[None].C3#[None].C4#[None]", "Rating", "AAA"
```

SetData

個々のレコードを設定します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算

構文

```
HS.SetData  
  lView  
  , 1  
  Account  
  , 1  
  ICP  
  , 1  
  Custom1  
  , 1  
  Custom2  
  , 1  
  Custom3  
  ,  
  lCustom4  
  ,  
  dData  
  ,  
  bAddToExistingData
```

表79 SetData関数の構文

パラメータ	説明
<i>lView</i>	0 = シナリオ・ビュー

パラメータ	説明
	1 = 期別ビュー 2 = 年次累計ビュー
<i>lAccount</i>	データを設定する勘定科目のID番号。
<i>lICP</i>	データを設定するICPのID番号。
<i>lCustom1...4</i>	データを設定するカスタム・ディメンションのID番号。
<i>dData</i>	設定するデータ値。
<i>bAddToExistingData</i>	TRUE = データを累積する FALSE = データを置換する

戻り値

なし。

例

```
HS.SetData 2, 002, , , , , 25000, TRUE
```

SetDataWithPOV

データをノードまたは通貨キューブに挿入します。この関数は、次のタイプのルールで使用できます。

- ・ 計算
- ・ 換算

構文

```
HS.SetDataWithPOV
POV
,
dData
,
bAddToExistingDataInCache
```

表80 SetData関数の構文

パラメータ	説明
<i>POV</i>	有効なPOV。
<i>dData</i>	設定するデータ値。

パラメータ	説明
<i>bAddToExistingData</i>	TRUE = データを累積する FALSE = データを置換する

戻り値

なし。

例

```
HS.SetDataWithPOV "V#YTD.A#Asset.I#[ICP
None].C1#None.C2#None.C3#None.C4#None", 25000, TRUE
```

SubmissionGroup

ディメンション・メンバーのプロセス管理の提出グループを取得します。

構文

```
HS.<
Dimension
>.SubmissionGroup(
Dimension member
)
```

ここで、*Dimension*は、次のいずれかのディメンションです: 勘定科目、カスタム1...4またはカスタム

```
HS.Account.SubmissionGroup(
Account
)
```

```
HS.Custom1.SubmissionGroup(
Custom1
)
```

```
HS.Custom2.SubmissionGroup(
Custom2
)
```

```
HS.Custom3.SubmissionGroup(
```

```
Custom3  
)
```

```
HS.Custom4.SubmissionGroup(  
Custom4  
)
```

```
HS.Custom(  
Custom Dimension Label  
) .SubmissionGroup(  
(CustomDimensionLabel)  
)
```

戻り値

ディメンション・メンバーの提出グループを表す整数。有効な値は1-99です。

例

```
HS.Account.SubmissionGroup("Sales")
```

```
HS.Custom("Product").SubmissionGroup("P3000-Phones")
```

SupportsProcessManagement

シナリオでプロセス管理をサポートしているかどうかを調べます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Scenario.SupportsProcessManagement ("Scenario")
```

```
HS.Scenario.SupportsProcessManagement ("")
```



注:

この関数を現在のシナリオに適用するには、空白の文字列("")を使用します。

```
HS.Scenario.SupportsProcessManagement(  
  Var1  
)
```

表81 SupportsProcessManagement関数の構文

パラメータ	説明
<i>Scenario</i>	有効なシナリオ。
<i>Var1</i>	シナリオメンバーを表すVBScript変数。

戻り値

シナリオでプロセス管理が使用可能な場合はTRUE、そうでない場合はFALSEとなるブール値。

例

この例では、実績シナリオでプロセス管理が使用可能である場合に、IfとEnd Ifの間にあるすべての文が実行されます。

```
If HS.Scenario.SupportsProcessManagement("Actual") = "TRUE" then
```

```
...
```

```
End IF
```

SupportsTran

内部取引をサポートするアプリケーションの勘定科目を指定します。この関数は、取引ルールのみで使用できます。

構文

```
HS.SupportsTran "  
  POVExpression  
"
```

POVExpressionは、勘定科目、カスタム1-4、シナリオ、エンティティおよび年の各メンバーの組合せです。

戻り値

なし。

例

```
HS.SupportsTran "S#ActMon.A#RecltIC.C1#Closing"
```

SwitchSign

現在のカスタム・メンバーまたは指定されたカスタム・メンバーの貸方が借方に切り替えられているかどうかを調べます。この関数は、次のルールを使用して、貸方/借方の符号を逆にします。

- ASSET勘定からLIABILITY勘定へ
- LIABILITY勘定からASSET勘定へ
- EXPENSEからREVENUEへ
- REVENUEからEXPENSEへ
- BALANCE勘定からFLOW勘定へ
- FLOW勘定からBALANCE勘定へ

この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.  
<Object>  
.SwitchSign("  
Member  
")
```

```
HS.  
<Object>  
.SwitchSign("")
```

```
HS.Custom(  
  Dimension  
) .SwitchSign(  
  Member  
)
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表82 SwitchSign関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none">• Custom1• Custom2• Custom3• Custom4• Custom
Member	カスタム・ディメンションの有効なメンバーの名前。

戻り値

カスタム・メンバーの貸方が借方に切り替わる場合にTRUE、そうでない場合にFALSEとなるブール式。

例

この例では、貸方と借方が切り替わった場合に、If...ThenとEnd Ifの文の間にあるすべての文が実行されます。

```
If HS.Custom1.SwitchSign("") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Product").SwitchSign("P3000-Phones") = TRUE Then
```

```
...
```

SwitchType

現在のカスタム・メンバーまたは指定されたカスタム・メンバーの勘定科目の種別が切り替えられているかどうかを調べます。この関数は、次のルールを使用して、カスタム・ディメンション・メンバーの勘定科目の種別を変更します。

- ASSET勘定からEXPENSE勘定へ
- EXPENSE勘定からASSET勘定へ
- LIABILITYからREVENUEへ
- REVENUEからLIABILITYへ
- BALANCE勘定からFLOW勘定へ
- FLOW勘定からBALANCE勘定へ

この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.
<Object>
.SwitchType("
Member
")
```

```
HS.
<Object>
.SwitchType("")
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表83 SwitchType関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか:

パラメータ	説明
	<ul style="list-style-type: none"> • Custom1 • Custom2 • Custom3 • Custom4 • Custom
<i>Member</i>	カスタム・ディメンションの有効なメンバーの名前。

戻り値

カスタム・メンバーの勘定科目の種別が切り替わる場合にTRUE、そうでない場合にFALSEとなるブール式。

例

この例では、現在のカスタム1メンバーの勘定科目の種別が切り替わる場合に、If...ThenとEnd Ifの文の間にあるすべての文が実行されます。

```
If HS.Custom1.SwitchType("") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Product").SwitchType("P3000-Phones") = TRUE Then
```

```
...
```

```
End If
```

Trans

年次累計メソッドを使用して通貨を換算します。この関数は換算ルールで使用できます。

構文

```
HS.Trans("
  DestPOV
```

```

", "
SourcePOV
", "
Rate1
", "
Rate2
")

```

表84 Trans関数の構文

パラメータ	説明
DestPOV	宛先視点。宛先は、勘定科目、カスタム1...4、ICPのメンバーの組合せを指定できます。指定されていない各ディメンションに対して、ディメンションの有効なすべてのメンバーに書き込まれます。指定された各ディメンションに対して、指定されたメンバーのみに書き込まれます。
SourcePOV	ソース視点。ソースはディメンションの組合せを指定できます。勘定科目、カスタム1...4およびICPのディメンションが指定されていない場合は、Destinationメンバーと同じメンバーが使用されます。シナリオ、年、期間およびエンティティのディメンションが指定されていない場合は、現在のメンバーが使用されます。値ディメンションが指定されていない場合は、<EntityCurrTotal>メンバーが使用されます。ソース視点空白の場合は、ソース視点として宛先視点を使用されます。
Rate1-2	為替レート。為替レートは、定数、為替レート勘定科目または特定のセルを指定できます。Rate勘定科目はエンティティおよび[None]エンティティに入力されます。デフォルトの換算プロセスについては、 239ページのデフォルトの換算方法 を参照してください。

戻り値

なし。

例

この例では、Rate1勘定科目のレートを使用して、年次累計メソッドによりSales勘定科目が換算されます。

```
HS.Trans("A#Sales", "A#LastYearSales", "A#Rate1", "")
```

TransPeriodic

期別メソッドを使用して通貨を換算します。この関数は換算ルールで使用できます。

構文

```
HS.TransPeriodic("DestPOV", "SourcePOV", "Rate1", "Rate2")
```

表85 TransPeriodic関数の構文

パラメータ	説明
DestPOV	宛先視点。宛先は、勘定科目、カスタム1-4、ICPのメンバーの組合せを指定できます。指定されていない各ディメンションに対して、ディメンションの有効なすべてのメンバーに書き込まれます。指定された各ディメンションに対して、指定されたメンバーのみに書き込まれます。
SourcePOV	ソース視点。ソースはディメンションの組合せを指定できます。勘定科目、カスタム1、カスタム2、カスタム3、カスタム4およびICPのディメンションが指定されていない場合は、Destinationメンバーと同じメンバーが使用されます。シナリオ、年、期間およびエンティティのディメンションが指定されていない場合は、現在のメンバーが使用されます。値が指定されていない場合は、EntityCurrTotalメンバーが使用されます。ソースが空白の場合は、ソースとして宛先が使用されます。
Rate1-2	為替レート。為替レートは、定数、為替レート勘定科目または特定のセルを指定できます。Rate勘定科目はエンティティおよびNoneエンティティに入力されます。デフォルトの換算プロセスについては、 239ページのデフォルトの換算方法 を参照してください。

戻り値

なし

例

この例では、Rate1勘定科目の為替レートを使用して、期別メソッドによりSales勘定科目が換算されます。

```
HS.TransPeriodic("A#Sales", "A#LastYearSales", "A#Rate1", "")
```

UD1...3

現在のメンバーまたは指定されたメンバーのUserDefined1...3属性に保管されたテキストを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.
<Object>
.UD1...3(str
Element
)
```

```
HS.
<Object>
.UD1...3("")
```

```
HS.Custom(
Dimension
).UD1(
Member
)
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表86 UD1-3関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> • Account • Entity • Parent • Scenario • Custom1...4 • Custom
要素	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> • Account • Entity • Parent

パラメータ	説明
	<ul style="list-style-type: none"> • Scenario • Custom1...4 • Custom

戻り値

メンバー用に保存されたユーザー定義のテキストを含む文字列。

例

この例では、UD1勘定科目のユーザー定義テキストがHistoryである場合に、If...ThenとEnd Ifの文間にある文が実行されます。

```
If HS.Account.UD1(strAccount) = "History" Then
```

```
...
```

```
End If
```

```
If HS.Custom("Product").UD1("P3000-Phones") = "History" Then
```

```
...
```

```
End If
```

ValidationAccount

アプリケーションの検証勘定を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.AppSettings.ValidationAccount
```

戻り値

アプリケーションの検証勘定の名前を示す文字列。

例

アプリケーションの検証勘定科目がMyAccountである場合に、If...ThenとEnd Ifの文間にある文が実行されます。

```
If HS.AppSettings.ValidationAccount = "MyAccount" Then
```

```
...
```

```
End If
```

ValidationAccountEx

プロセス管理の提出フェーズの検証勘定科目を取得します。

構文

```
HS.AppSettings.ValidationAccountEx(  
  n  
)
```

*n*は、プロセス管理の提出フェーズを表す整数です。有効な値は1から9です。

戻り値

プロセス管理の提出フェーズの検証勘定科目名を示す文字列。

例

この例では、提出フェーズ5に定義された検証勘定を戻します。

```
HS.AppSettings.ValidationAccountEx(5)
```

XBRLTags

指定された勘定科目メンバーに割り当てられたXBRLを取得します。この関数は、次のタイプのルールで使用できません:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Account.XBRLTags("Account")
```

```
HS.Account.XBRLTags("")
```



注:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

表87 XBRLTags関数の構文

パラメータ	説明
Account	有効な勘定科目。

戻り値

指定した勘定科目のXBRLタグを表す文字列。

12

カスタム関数

この項の内容:

管理レポート関数	400
ビジネス・ルール関数	423

このセクションでは、管理レポート関数およびプランニング関数用にFinancial Managementで使用できる内部HSカスタム関数をリストします。関数には、説明、関数のタイプ、構文、例およびサンプル・スクリプトが含まれます。

表88 管理レポート関数

カスタム関数	説明	構文	関数のタイプ	Hyperion Enterpriseでの同等
Average	財務の平均を計算します	Average (POV, Periods)	関数	AVE A12
Cumulative	前の期間からの金額を累計します	Cumulative (POV, View, NumPeriod)	関数	CUM CTD YTD
Difference	現在と開始時の差を計算します	Difference (POV, View)	関数	DIF DFB
DSO	売上債権回転日数を計算します	DSO (DSO, Debtor, Sales, DIP)	プロシージャ	Procedure
Opening	期首残高を繰り越します	Opening (POV, View)	関数	OPE BASE BASEFLOW
Rate	関連する為替レートを取得します	Rate (ExchangeRate, Triangulation Currency)	関数	CrossRate

表89 プランニング関数

カスタム関数	説明	パラメータ	関数のタイプ
Units_Rates	単位 * レート(C=A*B)	Unit_Rates (Description, Units, Rates)	プロシージャ
Custom_Alloc	カスタム・ディメンションに配賦します	Custom_Alloc (Destination, Source, Factor, FactorN, FactorD, Elimination)	プロシージャ

カスタム関数	説明	パラメータ	関数のタイプ
Increase_Decrease	勘定科目をある比率で増減します	Increase_Decrease (Destination, Source, Factor, Scale, Inverse)	プロシージャ
Pro_Rata_Ratio	2つの勘定科目の比率	Pro_Rata_Ratio (Destination, SourceN, SourceD)	プロシージャ
Spread	合計金額を年のすべての期間に分散します	Spread (Destination, Source, Factor, FactorN, FactorD, Temp, Per)	プロシージャ

管理レポート関数

このセクションでは、使用可能な管理レポートのカスタム関数をリストします。

Average

多数の期間に渡る勘定科目の平均値を計算します。

戻り値

HS.EXP関数の一部として使用される正しい式を表す文字列が戻されます。

構文

```
Average (
  PointOfView
  ,
  Periods
)
```

表90 Average関数の構文

パラメータ	有効な値
PointOfView	A#CASH.C1#[None].I#[ICP Top]など、勘定科目、カスタム1から4、ICPメンバーの有効な組合せ フロー・タイプの勘定科目の場合、この関数で計算されるのは、期間の値の平均のみです。
Periods	次のいずれかの値にする必要があります: YTD - 現在の年におけるある期間の累積データを平均するための年次累計オプションを指定します。

パラメータ	有効な値
	<p>Periodic - 現在の年のみにおける現在の期間と直前の期間を平均するための期間オプションを指定します。最初の期間では、この値はソースの値と同じです。</p> <p>[任意の整数] - 平均を計算する期間数を指定します。月のカテゴリで年間の平均を計算する場合は、12を指定します。</p>

詳細説明

この関数は、過去の指定された期間数に渡り、勘定科目の平均値を計算します。ソースが残高タイプの勘定科目の場合、平均は入力されたデータに基づきます。ソースがフロータイプの勘定科目の場合、平均は期間データのみに基づきます。

平均値の導出方法は、*Periods*パラメータによって異なります。

- *Periods*パラメータがYTDの場合、平均値は現在の年における現在までの全期間の合計を、現在の期間数で割った値です。
- *Periods*パラメータが期別の場合、平均値は現在と直前の期間の合計を2で割った値です。現在の期間が年の最初の期間である場合、平均値はソースの値と同じです。
- *Periods*パラメータが数値の場合、平均値は、現在の期間と過去の各期間の指定された期間数の合計を、指定された数で割った値です。

例

SALES勘定科目では、Averageカスタム関数で使用される*Periods*パラメータに基づいて、2014年1月、2月、3月に次の値が戻されます。処理中のシナリオに設定されたデフォルト・ビューはYTDです。

表91 Average関数の例

勘定科目	Oct2013	Nov2013	Dec2013	Jan2014	Feb2014	Mar2014
A#Sales	9,000	10,500	11,700	800	1,900	3,200
Average("A#Sales", "YTD")	N/A	N/A	N/A	800	950	1,067
Average("A#Sales", "Periodic")	N/A	N/A	N/A	800	950	1,200
Average("A#Sales", "3")	N/A	N/A	N/A	1,167	1,033	1,067

サンプル・スクリプト

```
' sample statement written in the calling routine
```

```
Sub Calculate()
```

```
  Hs.Exp "A#AVG_SALES = " & Average("A#Sales", "12")
```

```
End Sub
```

```
' programming of the AVERAGE function
```

```
FUNCTION Average(strPOV, strPERIOD)
```

```
  DIM nPERIOD  
  DIM strCUM  
  DIM i
```

```
  strPOV = UCASE(strPOV)  
  strPERIOD = UCASE(strPERIOD)
```

```
  IF strPERIOD = "PERIODIC" THEN
```

```
    IF HS.PERIOD.ISFIRST = TRUE THEN
```

```
      nPERIOD = 1
```

```
    ELSE
```

```
      nPERIOD = 2
```

```
    END IF
```

```
  ELSEIF strPERIOD = "YTD" THEN
```

```
    nPERIOD = HS.PERIOD.NUMBER()
```

```
  ELSEIF CINT(strPERIOD) > 0 THEN
```

```
nPERIOD = CINT(strPERIOD)
```

```
ELSE
```

```
EXIT FUNCTION
```

```
END IF
```

```
FOR i = 0 TO nPERIOD-1
```

```
IF i = 0 THEN
```

```
strCUM = strPOV & ".W#PERIODIC"
```

```
ELSE
```

```
strCUM = strCUM & "+" & strPOV & ".W#PERIODIC.P#CUR-" & i
```

```
END IF
```

```
NEXT
```

```
Average = "(" & strCUM & ") / (" & nPERIOD & ")"
```

```
END FUNCTION
```

Cumulative

指定された勘定科目の前の期間の合計値を計算します。

戻り値

HS.EXP関数の一部として使用される正しい式を表す文字列が戻されます。

構文

```
Cumulative (  
  PointOfView  
  ,  
  View, NumPeriod  
)
```

表92 Cumulative関数の構文

パラメータ	有効な値
PointOfView	A#CASH.C1#[None].I#[ICP Top]など、勘定科目、カスタム1から4、ICPメンバーの有効な組合せ
View	次のいずれかの値にする必要があります: " " (二重引用符) - 処理中のシナリオに定義されているデフォルト・ビューに基づきます(YTDまたはPeriodic)。 YTD - ユーザーにより年次累積オプションが指定され、シナリオに設定されたデフォルト・ビューが上書きされます。 Periodic - 期別オプションが指定され、シナリオに設定されたデフォルト・ビューが上書きされます。
NumPeriod	累計する現在のシナリオ内の期間数を表す整数で、現在の期間から開始されます。 NumPeriodは0または負数で、関数は現在の年の初めから集計します。

詳細説明

この関数は、指定された期間の合計、または指定された勘定科目の年次累計の合計を計算します。デフォルトでは、累積されたデータのビューはシナリオのデフォルトです; ただし、フロー・タイプの勘定科目の場合は上書きできます。

- ViewパラメータがYTDの場合、この関数では年次累計値が累計されます。
- ViewパラメータがPeriodicの場合は、期間の値が累計されます。
- Viewパラメータが空白(" ")の場合は、シナリオのデフォルト・ビューを使用してデータが累計されます。

例

CASH勘定科目では、Cumulative関数で使用するNumberパラメータに基づいて、2014年1月、2月、3月に次の値が戻されます。

SALES勘定科目では、Cumulative関数で使用するViewおよびNumberパラメータの両方に基づいて、2014年1月、2月、3月に次の値が戻されます。処理中のシナリオに設定されたデフォルト・ビューはYTDです。

表93 Cumulative関数の例

勘定科目	Oct2013	Nov2013	Dec2013	Jan2014	Feb2014	Mar2014
A#Cash	1,000	1,500	1,200	800	1,100	1,300
Cumulative("A#Cash", "", 0)	N/A	N/A	N/A	800	1,900	3,200
Cumulative("A#Cash", "", 3)	N/A	N/A	N/A	3,500	3,100	3,200
A#Sales	9,000	10,500	11,700	800	1,900	3,200
Cumulative("A#Sales", "", 0)	N/A	N/A	N/A	800	2,700	5,900
Cumulative("A#Sales", "Periodic", 0)	N/A	N/A	N/A	800	1,900	3,200
Cumulative("A#Sales", "Periodic", 3)	N/A	N/A	N/A	3,500	3,100	3,200

サンプル・スクリプト

```
' sample statement written in the calling routine
```

```
Sub Calculate()
```

```
HS.EXP "A#TOT_Cash ="&Cumulative("A#Cash", " ", 0)
```

```
End Sub
```

```
' programming of the Cumulative function
```

```
Function Cumulative(StrPov, StrVIEW, nPERIOD)
```

```
DIM strCUM
DIM i
```

```
IF nPERIOD <= 0 THEN
```

```
nPERIOD = HS.PERIOD.NUMBER() - 1
```

```
ELSE
```

```
nPERIOD = nPERIOD - 1
```

```
END IF
```

```
IF strVIEW = "" THEN
```

```
strVIEW = HS.SCENARIO.DEFAULTVIEW("")
```

```
END IF
```

```
strPOV = UCASE(strPOV)  
strVIEW = UCASE(strVIEW)
```

```
IF strVIEW = "PERIODIC" THEN
```

```
strVIEW = ".W#PERIODIC"
```

```
ELSEIF strVIEW = "YTD" THEN
```

```
strVIEW = ".W#YTD"
```

```
ELSE
```

```
EXIT FUNCTION
```

```
END IF
```

```
FOR i = 0 TO nPERIOD
```

```
IF i = 0 THEN
```

```
strCUM = strPOV & strVIEW
```

```
ELSE
```

```
strCUM = strCUM &" "& strPOV & strVIEW &".P#CUR-"&i
```

```
END IF
```

```
NEXT
```

```
Cumulative = "(" & strCUM & ")"
```

```
END FUNCTION
```

Difference

現在の期間の値と開始値の差を計算します。

戻り値

HS.EXP関数の一部として使用される正しい式を表す文字列が戻されます。

構文

```
Difference (  
  PointOfView  
,  
  View  
)
```

表94 Difference関数の構文

パラメータ	有効な値
PointOfView	A#CASH.C1#[None].I#[ICP Top]など、勘定科目、カスタム1から4、ICPメンバーの有効な組合せ

パラメータ	有効な値
View	次のいずれかの値にする必要があります: " " (二重引用符) - 処理中のシナリオに定義されているデフォルト・ビューに基づきます(YTDまたはPeriodic)。 YTD - 年次累積オプションが指定され、シナリオに設定されたデフォルト・ビューが上書きされます。 Periodic - 期別オプションが指定され、シナリオに設定されたデフォルト・ビューが上書きされます。

詳細説明

この関数は、現在の期間の値と開始値の差を計算します。(現在 - 開始時)

開始値の導出方法は、関数に渡されるViewパラメータによって異なります。

- ViewパラメータがYTDの場合、開始値は、前年の最後の期間から取得されます。
- ViewパラメータがPeriodicの場合、開始値は、現在の年の前の期間から取得されます。現在の期間が年の最初の期間である場合、開始値は、前年の最後の期間から取得されます。
- Viewパラメータが空白(" ")の場合、開始値は、シナリオのデフォルトのデータ・ビューに基づきます。

例

CASH勘定科目では、Difference関数で使用されるViewパラメータに基づいて、2014年1月、2月、3月に次の値が戻されます。処理中のシナリオに設定されたデフォルト・ビューはYTDです。Difference関数は、現在の期間の値から開始値を減算します。

表95 Difference関数の例

勘定科目	Dec2013	Jan2014	Feb2014	Mar2014
A#Cash	900	1,200	1,100	1,500
Difference("A#Cash"N/A "")		300	200	600
Difference("A#Cash"N/A "YTD")		300	200	600
Difference("A#Cash"N/A "Periodic")		300	-100	400

サンプル・スクリプト

```
' sample statement written in the calling routine
```



```
Sub Calculate()
```

```
  Hs.Exp "A#DiffCash = " & Difference("A#Cash", "YTD")
```

```
End Sub
```

```
' programming of the DIFFERENCE function
```

```
FUNCTION DIFFERENCE(strPOV, strVIEW)
```

```
  IF strVIEW = "" THEN
```

```
    strVIEW = HS.SCENARIO.DEFAULTVIEW ("")
```

```
  END IF
```

```
    strPOV = UCASE(strPOV)  
    strVIEW = UCASE(strVIEW)
```

```
  IF strVIEW = "PERIODIC" THEN
```

```
    DIFFERENCE = "(" & strPOV & "-" & strPOV & ".P#PRIOR" & ")"
```

```
  ELSEIF strVIEW = "YTD" THEN
```

```
    DIFFERENCE = "(" & strPOV & "-" & strPOV & ".Y#PRIOR.P#LAST" & ")"
```

```
  ELSE
```

```
    EXIT FUNCTION
```

```
  END IF
```

END FUNCTION

DSO - 売上債権回転日数

全数法を使用して、当期借方の売上回収日数が計算されます。

戻り値

このルーチンは、当期取引の借方内における売上回収日数を表す単一の値を計算します。含まれるDSOサブルーチンでは、次のことが仮定されています：

- ・ 借方も売上も正数です。
- ・ ルーチンではHS.GETCELL関数を使用されるため、指定されるパラメータは完全に定義された視点(勘定科目/C1/C2/C3/C4/ICPなど)である。
- ・ ルーチンでは、可能なかぎりさかのぼって日数が計算されます。ただし、任意の期間の期別売上が負数であるかゼロの場合には停止します。

構文

```
CALL DSO (  
  strDSO, strDEBTOR, strSALES, strDIP  
)
```

表96 DSO関数の構文

パラメータ	有効な値
strDSO	カスタム・ディメンションと内部取引ディメンションを含む完全に定義された勘定科目。この勘定科目は計算の宛先です。
strDEBTOR	カスタム・ディメンションと内部取引ディメンションを含む完全に定義された勘定科目。この勘定科目は当期取引の借方のソースです。
strSALES	カスタム・ディメンションと内部取引ディメンションを含む完全に定義された勘定科目。この勘定科目は売上のソースです。 特に、期間単位への参照は除外されます。
strDIP	カスタム・ディメンションと内部取引ディメンションを含む完全に定義された勘定科目。この勘定科目はその期間の日数のソースです。

パラメータ	有効な値
	これは、[なし]エンティティであるとみなされます。

詳細説明

このルーチンは、現在の期間のDebtors勘定科目(パラメータ2)およびSales勘定科目(パラメータ3)の値を取得し、それらを比較します。どちらかがゼロまたは負数の場合は計算が止まります。借方の値が累積売上の値を超える連続した各期間(現在の期間からさかのぼる)では、ルーチンにより、Period勘定科目(パラメータ4)の日数に指定されているとおりに、その期間の日数が累計に追加されます。

このようにして、すべての借方の値が全数法で処理されると、最後の期間の日数は、期別売上値に対する失効前借方の一部として計算されます。

こうして、累計が宛先の勘定科目(パラメータ1)に転記されます。

例

この例では、表示されている月の合計未処理日数を計算しています。

表97 DSO関数の例

月	借方	期別売上	月の日数	DSOの式	合計DSO
9月	12,000	2,500	30	100%	30
8月	N/A	1,750	31	100%	31
7月	N/A	2,250	31	100%	31
6月	N/A	2,500	30	100%	30
5月	N/A	2,000	31	100%	31
4月	N/A	2,250	30	2000/2250	26.7
合計	N/A	N/A	N/A	N/A	179.7

サンプル・スクリプト

```
' Use within the calculation section:
' 1. Standard use
```

```
CALL DSO("A#DSO", "A#TradeDebtors.C1#AllAges.C2#[None].I#[ICP
Top]", "A#TotalSales.C1#[None].C2#AllProducts.I#[ICP Top]", "A#DIP")
```

```
' 2. Use with a common custom dimension
```

```
set vPRODUCT = ARRAY("C2#PRODUCT1", "C2#PRODUCT2", ... , "C2#PRODUCTn")
```

```
FOR EACH iITEM IN vPRODUCT
```

```
CALL DSO("A#DSO."&iITEM, "A#TradeDebtors.C1#AllAges.I#[ICP  
Top]."&iITEM, "A#TotalSales.C1#[None].I#[ICP Top]."&iITEM, "A#DIP")
```

```
NEXT
```

```
' Actual script of Sub-routine
```

```
SUB DSO(strDSO, strDEBTOR, strSALES, strDIP)
```

```
DIM vTEST
```

```
DIM vDSO
```

```
DIM vCOUNT
```

```
DIM vXS_1
```

```
DIM vXS
```

```
HS.CLEAR(strDSO)
```

```
vTEST = HS.GETCELL(strDEBTOR) * HS.GETCELL(strSALES&".W#Periodic") *  
HS.GETCELL(strDIP&".E#[None]")
```

```
' checks if any of the parameters are zero (uses principle of  $X * 0 = 0$ )
```

```
IF vTEST = 0 THEN
```

```
EXIT SUB
```

```
ELSE
```

```
vDSO = 0
```

```
vCOUNT = 0
```

```
vXS_1 = HS.GETCELL(strDEBTOR)
```

```
vXS = vXS_1 - HS.GETCELL(strSALES&".W#Periodic")
```

```

' ensures that periodic sales are not negative or zero

WHILE vXS > 0 AND vXS_1 > vXS

    vDSO = vDSO + HS.GETCELL(strDIP&".E#[None].P#CUR-" &vCOUNT)
vCOUNT = vCOUNT + 1
vXS_1 = vXS
vXS = vXS - HS.GETCELL(strSALES&".W#Periodic.P#CUR-" &vCOUNT)

WEND

IF vXS = vXS_1 THEN

    vCOUNT = vCOUNT - 1

END IF

    vDSO = vDSO + (vXS_1 / HS.GETCELL(strSALES&".W#Periodic.P#CUR-"
&vCOUNT)*HS.GETCELL(strDIP&".E#[None].P#CUR-" &vCOUNT))

IF vDSO < 0 THEN

    vDSO = 0

END IF

END IF

HS.EXP strDSO &"="& vDSO

END SUB

```

Opening

完全に定義された指定の勘定科目(勘定科目/C1/C2/C3/C4/ICP)の開始値を取得します。

戻り値

この関数では、HS.EXP関数の一部として使用される正しい式を表す文字列が戻されます。

構文

```
Opening (
  PointOfView, View
)
```

表98 Opening関数の構文

パラメータ	有効な値
PointOfView	A#CLOSE.C1#[None].I#[ICP Top]など、勘定科目、カスタム1から4、ICPメンバーの有効な組合せ
View	次のいずれかの値にする必要があります: " " (二重引用符) - 処理中のシナリオに定義されているデフォルト・ビューに基づきます(YTDまたはPeriodic)。 YTD - 年次累積オプションが指定され、シナリオに設定されたデフォルト・ビューが上書きされます。 Periodic - 期別オプションが指定され、シナリオに設定されたデフォルト・ビューが上書きされます。

詳細説明

この関数は、指定された勘定科目の開始値を計算します。開始値の導出方法は、Viewパラメータによって異なります。

- ViewパラメータがYTDの場合、開始値は、前年の最後の期間から取得されます。
- ViewパラメータがPeriodicの場合、開始値は、現在の年の前の期間から取得されます。現在の期間が年の最初の期間である場合、開始値は、前年の最後の期間から取得されます。
- Viewパラメータが空白(" ")の場合、開始値は、シナリオのデフォルトのデータ・ビューに基づきます。

例

FA_COST勘定科目では、Opening関数で使用されるViewパラメータに基づいて、2014年1月、2月、3月に次の値が戻されます。処理中のシナリオに設定されたデフォルト・ビューはYTDです。

表99 Opening関数の例

勘定科目	Dec2013	Jan2014	Feb2014	Mar2014
A#FA_COST	900	1,200	1,100	1,500

勘定科目	Dec2013	Jan2014	Feb2014	Mar2014
Opening("A#FA_COST/A ")	900	900	900	900
Opening("A#FA_COST/A " YTD")	900	900	900	900
Opening("A#FA_COST/A "Periodic ")	900	1,200	1,100	

サンプル・スクリプト

```
' sample statement written in the calling routine
```

```
Sub Calculate()
```

```
Hs.Exp "A#Open_FA_Cost = " & Opening("A#FA_Cost", "YTD")
```

```
End Sub
```

```
' programming of the OPENING function
```

```
FUNCTION OPENING(strPOV, strVIEW)
```

```
IF strVIEW = "" THEN
```

```
strVIEW = HS.SCENARIO.DEFAULTVIEW ("")
```

```
END IF
```

```
strPOV = UCASE(strPOV)  
strVIEW = UCASE(strVIEW)
```

```
IF strVIEW = "PERIODIC" THEN
```

```
OPENING = strPOV & ".P#PRIOR"
```

```
ELSEIF strVIEW = "YTD" THEN
```

```
OPENING = strPOV & ".Y#PRIOR.P#LAST"
```

```
ELSE
```

```
EXIT FUNCTION
```

```
END IF
```

```
END FUNCTION
```

Rate

親と子の間で関連する為替レートを計算し、値を乗数として戻します。

戻り値

この関数は、通常、換算セクションでHS.EXP関数の一部として使用される値を戻します。

構文

```
Rate (  
ExchangeRate, TriangulationCurrency  
)
```

表100 Rate関数の構文

パラメータ	有効な値
ExchangeRate	勘定科目文字列として指定されたCurrencyRateタイプのメイン勘定科目で、A#EOP_RATEなどのカスタム・ディメンションや内部取引ディメンションへの参照は含まれません
TriangulationCurrency	文字列または二重引用符(" ")のいずれかの有効な通貨ラベル。通貨を指定する際は、カスタム・ディメンションを参照する必要はありません。

詳細説明

- この関数は、親と子の間で関連する為替レートを計算し、値を乗数として戻します。この値は、*TriangulationCurrency*パラメータに基づいて計算されます。
- TriangulationCurrency*パラメータが有効な通貨ラベルの場合、クロス・レートはこの通貨に基づきます。
- TriangulationCurrency*パラメータが空白(" ")の場合は有効な直接レートが検索され、検出されないときには、アプリケーションの通貨に対してトライアングレーションが使用されます。
- レート値が見つからない場合は、関数により1が戻されます。

後続の表には、データの検索方法と、その検索が行われる順序が示されています。順序は、(1)のように、カッコで囲んだ数値で表されています。いずれの場合も、検索はまず子エンティティで行われ、データが見つからない場合に[None]エンティティから行われます。

次の表では、子の通貨または親の通貨がトライアングレーション通貨と同一であり、トライアングレーションが空白の場合には、アプリケーションの通貨と同一です。

表101 Rateの例 - トライアングレーション通貨と同一の場合

		カスタム1ディメンションのレート	
		子	親
カスタム2ディメンションのレート	子		(2)
親	(1)		

次の表では、トライアングレーションが指定されており、子と親のいずれの通貨とも同一ではありません。

表102 Rateの例 - トライアングレーション通貨と異なる場合

		カスタム1ディメンションのレート		
		子	親	トライアングレーション
カスタム2ディメンションのレート	子			(2)
親				
トライアングレーション		(1)		

次の表では、トライアングレーションが指定されておらず、アプリケーションの通貨が子と親のいずれの通貨とも同一ではありません。

表103 Rateの例 - トライアングレーションが指定されていない場合

		カスタム1ディメンションのレート		
		子	親	アプリケーション
カスタム2ディメンションのレート	子		(2)	(4)

		カスタム1ディメンションのレート		
親	(1)			
アプリケーション		(3)		

例

アプリケーションの通貨がユーロで、C2#EUROに対して[None]エンティティに入力された次のレートを使用して、French子エンティティをUS親エンティティに換算する必要があります:

表104 Rate関数の例

	期首レート	期末レート
C1#FFR	0.16000	0.16500
C1#USD	1.15862	1.15785

次の関数は、期首残高勘定科目に、関連する期末レートと期首レートの差を乗算します。これは、ローカル通貨とアプリケーションの通貨間で換算が一貫していない場合に、運動解析を計算する際に便利です。

```
HS.EXP "A#FX0 = A#OPEN * (" & RATE("A#EOP_RATE", " ") & "-" & RATE("A#OPE_RATE", " ") &")"
```

前の例では、子のOPEN勘定科目の値がFFR 10,000,000の場合、USの親FX0勘定科目の値は、USD 44,102 $[10,000,000 * (0.165 / 1.15785 - 0.16 / 1.15862)]$ になります。

サンプル・スクリプト

```
' sample statement written in the calling routine
```

```
SUB TRANSLATE()
```

```
HS.TRANS "A#FX0", "A#FX0", "A#EOP_RATE", ""
HS.EXP "A#FX0 = A#OPEN * (" & RATE("A#EOP_RATE", " ") & "-" & RATE("A#OPE_RATE", " ") &")"
```

```
END SUB
```

```
' programming of the RATE function
```

```
FUNCTION RATE(sRATE, sTRI)
```

```
DIM sCCUR, sPCUR, sACUR, bRET, retValue, s3rdCUR
DIM i
```

```
    sRATE = UCASE(sRATE)
sTRI = UCASE(sTRI)
sCCUR = UCASE(HS.ENTITY.DEFCURRENCY(""))
sPCUR = UCASE(HS.VALUE.CURRENCY)
sACUR = UCASE(HS.APPSETTINGS.CURRENCY)
retValue = 0
```

```
    ' check whether there is a triangulation specified, or if triangulation or
application currencies are the same as either parent or child and set up the select
case
```

```
    IF sTRI = sCCUR OR sTRI = sPCUR OR (sTRI = " " AND (sACUR = sCCUR OR sACUR =
sPCUR)) THEN
```

```
        i = 1
```

```
    ELSEIF sTRI <> " " THEN
```

```
        i = 2
```

```
    ELSE
```

```
        i = 3
```

```
    END IF
```

```
    SELECT CASE i
```

```
        CASE 1
```

```
            ' bRET is a boolean that returns true if data is found. First search the
child...
' ...then search the [None] entity
```

```
bRET = GETVALUECP(".V#<Entity Currency>",retValue,sRATE,sCCUR,sPCUR)
```

```
IF NOT bRET THEN
```

```
bRET = GETVALUECP(".E#[None]",retValue,sRATE,sCCUR,sPCUR)
```

```
END IF
```

```
CASE 2
```

```
' use a dynamic parameter name for ease of writing the triangulation checks
```

```
s3rdCUR = sTRI
```

```
bRET = GETVALUE3(".V#<Entity Currency>",retValue,sRATE,sCCUR,sPCUR,s3rdCUR)
```

```
IF NOT bRET THEN
```

```
bRET = GETVALUE3(".E#[None]",retValue,sRATE,  
sCCUR,sPCUR,s3rdCUR)
```

```
END IF
```

```
CASE 3
```

```
' this case is used when the 2nd parameter is blank and is the most complex.
```

```
' first check direct rates in the child...
```

```
' ... then check triangulation against application currency in the child
```

```
' then check direct rates in [None].
```

```
'... finally check triangulation in [None]
```

```
s3rdCUR = sACUR
```

```
bRET = GETVALUECP(".V#<Entity Currency>", retValue, sRATE, sCCUR, sPCUR)
```

```
IF NOT bRET THEN
```

```
bRET = GETVALUE3(".V#<Entity Currency>", retValue, sRATE, sCCUR, sPCUR, s3rdCUR)
```

```
IF NOT bRET THEN
```

```
bRET = GETVALUECP(".E#[None]", retValue, sRATE, sCCUR, sPCUR)
```

```
IF NOT bRET THEN
```

```
bRET = GETVALUE3(".E#[None]", retValue,  
sRATE, sCCUR, sPCUR, s3rdCUR)
```

```
END IF
```

```
END IF
```

```
END IF
```

```
END SELECT
```

```
IF bRET THEN
```

```
RATE = retValue
```

```
ELSE
```

```
RATE = 1
```

```
END IF
```

```
END FUNCTION
```

```
FUNCTION GETVALUECP(sENTITY, sVALUE, sRATE, sCCUR, sPCUR)
```

```
' this sub-function is used when comparing direct rates between child and  
parent
```

```
GETVALUECP = FALSE
```

```
' check if data exists for direct rate child to parent. If it does return it.  
' if no direct child to parent rate check for indirect parent to child rate...  
' return the inverse of the indirect rate.
```

```
IF HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & sPCUR & sENTITY) <> 0 THEN
```

```
sVALUE = CDBL(HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & sPCUR & sENTITY))
```

```
GETVALUECP = TRUE
```

```
ELSEIF HS.GETCELL(sRATE & ".C1#" & sPCUR & ".C2#" & sCCUR & sENTITY) <> 0  
THEN
```

```
sVALUE = CDBL(1 / HS.GETCELL(sRATE & ".C1#" & sPCUR & ".C2#" & sCCUR &  
sENTITY))
```

```
GETVALUECP = TRUE
```

```
END IF
```

```
END FUNCTION
```

```
FUNCTION GETVALUE3(sENTITY, sVALUE, sRATE, sCCUR, sPCUR, s3rdCUR)
```

```
    ' this sub-function is used when triangulating  
' check if data exists for direct rate child to triangulation...  
' ... if it does return the direct relative rate child to parent...  
' if no direct child to triangulation rate check for indirect triangulation to child  
rate...  
' ... return the inverse of the indirect relative rates.
```

```
GETVALUE3 = FALSE
```

```
IF HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & s3rdCUR & sENTITY) <> 0 THEN
```

```
    sVALUE = CDBL(HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & s3rdCUR &  
sENTITY) / HS.GETCELL(sRATE & ".C1#" & sPCUR & ".C2#" & s3rdCUR & sENTITY))
```

```
GETVALUE3 = TRUE
```

```
ELSEIF HS.GETCELL(sRATE & ".C1#" & s3rdCUR & ".C2#" & sCCUR & sENTITY) <> 0  
THEN
```

```
    sVALUE = CDBL(HS.GETCELL(sRATE & ".C1#" & s3rdCUR & ".C2#" & sPCUR &  
sENTITY) / HS.GETCELL(sRATE & ".C1#" & s3rdCUR & ".C2#" & sCCUR & sENTITY))
```

```
GETVALUE3 = TRUE
```

```
END IF
```

```
END FUNCTION
```

ビジネス・ルール関数

このセクションでは、ビジネス・ルール・カスタム関数をリストします。

Custom_Alloc

この関数は、配賦の基本として係数POVを使用し、ソースの視点(POV)を宛先POVに割り当てます。また、消去POVに対する合計配賦額の転記を逆にするオプションもあります。この関数は、カスタム・ディメンションの配賦用に設計されています。

戻り値

戻り値はありません。

構文

```
Custom_Alloc(  
    Destination, Source, Factor, FactorN, FactorD,  
    Elimination  
)
```

表105 Custom_Alloc関数の構文

パラメータ	有効な値
Destination	勘定科目、ICP、カスタム1から4のメンバーの有効な組合せである有効な宛先POV。
Source	ディメンション・メンバーの有効な組合せである有効なソースPOV。Sourceは配賦される金額です。
Factor	有効なソースPOV。Factorは、配賦係数の格納に使用される勘定科目です。
FactorN	有効なソースPOV。FactorNは、配賦の基準として使用される分子係数です。
FactorD	有効なソースPOV。FactorDは、配賦の基準として使用される分母係数です。
Elimination	有効なソースPOV。Eliminationは空の文字列("")にすることが可能で、その場合、このパラメータは無視されます。Eliminationパラメータが設定されている場合、Destination POVに転記される金額には-1が乗算され、消去POVに転記されます。

詳細説明

この関数は、配賦の基本として係数POVを使用し、ソースのPOVを宛先POVに割り当てます。また、消去POVに対する合計配賦額の転記を逆にするオプションもあります。この関数は、カスタム・ディメンションの配賦用に設計されています。

Factorパラメータには、FactorDで除算されたFactorNの結果が格納されます。これは、係数が、現在のエンティティ以外のエンティティを参照できるようにするために必要です。

ソースPOVのエンティティが親である場合、子レベルで計算を実行する前に、その親を連結する必要があります。親の通貨が子の通貨と異なる場合、関連するすべての通貨の換算も、子レベルで計算を開始する前に実行する必要があります。

呼出しルーチンに変数を設定し、Destination、Source、Factor、FactorN、FactorDおよびElimination POVを定義するCustom_Allocに渡すことをお勧めします。また、呼出しルーチンの変数名をCustom_Alloc関数と同一になるように設定することもお勧めします。

Eliminationパラメータは空の文字列("")にすることが可能で、その場合、このパラメータは無視されます。Eliminationパラメータが設定されている場合、宛先POVに転記される金額には-1が乗算され、消去POVに転記されます。

例

Total Salesに対するProducts Salesの比率に基づいて、Telephone勘定科目がProductsに配賦されます。配賦額の逆数がAllocations勘定科目に転記されます。

表106 Custom_Alloc関数の例

勘定科目	Jan2014	Feb2014	Mar2014
A#Telephone.C1#[None]	100	300	400
A#Sales".C1#Product1	1000	1000	1000
A#Sales.C1#Product2	1000	2000	3000
A#Sales.C1#TotalProducts	2000	3000	4000
Custom_Alloc("A#Telephone", "A#Telephone.C1#[None]", "A#Factor", "A#Sales", "A#Sales.C1#TotalProducts", "A#ProductAllocations.C1#[None]")	N/A	N/A	N/A
A#Factor.C1#Product1	0.50	0.33	0.25
A#Factor.C1#Product2	0.50	0.66	0.75
A#Telephone.C1#Product1	50	100	100
A#Telephone.C1#Product2	20	200	300
A#ProductAllocations.C1#[None]	-100	-300	-400

CUSTOM_ALLOC関数から戻される結果は次のとおりです:

```
HS.EXP "A#Factor = A#Sales / A#Sales.C1#TotalProducts"
```

```
HS.EXP "A#Telephone = A#Telephone.C1#[None] * A#Factor"
```

```
HS.EXP "A#Allocations.C1#[None] = (A#Telephone.C1#[None] * -1)"
```

サンプル・スクリプト

このスクリプトには、次の情報が含まれます:

- 呼出しルーチンで記述されたサンプル文。
- 呼出しルーチンに設定され、Custom_Alloc関数に渡される変数。
- Custom_Alloc関数と同一になるように設定された呼出しルーチンの変数名。

```
Sub Calculate()
```

```
    Dim Destination
```

```
    Dim Source
```

```
    Dim Elimination
```

```
    Dim Factor
```

```
    Dim FactorN
```

```
    Dim FactorD
```

```
    Dim C1list
```

```
    Dim C1item
```

```
C1list = HS.Custom1.List("Alloc")
```

```
For Each C1item in C1list
```

```
    Source = "A#Telephone.C1#[None]"
```

```
    Destination = "A#Telephone.C1#" & C1item
```

```
    Factor = "A#Factor.C1#" & C1item
```

```
    FactorN = "A#Sales.C1#" & C1item
```

```
    FactorD = "A#Sales.C1#TotalProducts"
```

```
    Elimination = "A#ProductAllocations.C1#" & C1item
```

```
    Call Custom_Alloc(Destination,Source,Factor,FactorN,  
FactorD,Elimination)
```

```
Next
```

```
End Sub
```

```
' Beginning of the Custom_Alloc function
```

```
Sub Custom_Alloc(Destination,Source,FactorN,FactorD,  
Elimination)
```

```
HS.Clear Factor
```

```
HS.Exp Factor & " = " & FactorN & "/" & FactorD
```

```
HS.EXP Destination & " = " & Source & " * " & Factor
```

```
If Elimination <> "" Then
```

```
HS.EXP Elimination & " = " & Source & " * -1 * " & Factor
```

```
End If
```

```
End Sub
```

Increase_Decrease

この関数は、宛先POVを比率係数で増加または減少させます。比率係数は、ソースPOV、VBScript定数、またはVBScript変数から取得されます。

戻り値

戻り値はありません。

構文

```
Increase_Decrease(  
Destination, Source, Factor, Scale, Inverse  
)
```

表107 Increase_Decrease関数の構文

パラメータ	有効な値
Destination	勘定科目、ICP、カスタム1から4のメンバーの有効な組合せである有効な宛先POV。
Source	ディメンション・メンバーの有効な組合せである有効なソースPOV。Sourceは配賦される金額です。
Factor	有効なソースPOV、定数または変数。
Scale	1または100の整数値。FactorはScaleで除算されます。
Inverse	TRUEまたはFALSE。TRUEの場合は、Factorの符号が逆になります。これは、Factorが正数として格納される場合に減少させるときに使用できます(逆も同様)。FALSEの場合は、格納されたFactorの符号を使用して、増加させるか減少させるかが決定されます。

詳細説明

この関数は、宛先POVを比率係数で増加または減少させます。比率係数は、ソースPOV、VBScript定数、またはVBScript変数から取得されます。

一般的に、ソースPOVは宛先POVと同一ですが、別にすることも可能です。

Scaleパラメータは、必要な場合に、係数を小さくするために使用されます。これは、係数がソースPOVが取得され、調整されていないフォーム(50%が0.50ではなく50として格納される)に格納される場合に適用されます。

Inverseパラメータは、係数の符号を逆にするために使用されます。これは、係数がソースPOVから取得され、絶対値として格納される場合に適用されます。InverseパラメータがTRUEに設定されている場合、係数には-1が乗算されます。InverseパラメータがFALSEに設定されている場合、係数は-1で乗算されません。

例

この例では、Telephone勘定科目が10%増加されます。

表108 Increase_Decrease関数の例

勘定科目	Jan2014	Feb2014	Mar2014
A#Telephone	100	300	400
A#Factor/C1[None]	10	10	10
Increase_Decrease("A#Telephone", "A#Telephone", "A#Factor.C1#[None]",100,False)	N/A	N/A	N/A
A#Telephone	110	330	440

INCREASE_DECREASE関数から戻される結果は次のとおりです:

```
HS.EXP "A#Telephone = A#Telephone * (1+ (A#Factor.C1#[None]/100))"
```

サンプル・スクリプト

- 呼出しルーチンで記述されたサンプル文。
- 呼出しルーチンに設定され、Increase_Decrease関数に渡される変数。
- Increase_Decrease関数と同一になるように設定された呼出しルーチンの変数名。

```
Sub Calculate()
```

```
Dim Destination
```

```
Dim Source
```

```
Dim Factor
```

```
Dim Scale
```

```
Dim Inverse
```

```
Destination = "A#Telephone"
```

```
Source = "A#Telephone"
```

```
Factor = "A#Factor.C1#[None]"
```

```
Scale = "100"
```

```
Inverse = False
```

```
Call Increase_Decrease(Destination,Source,Factor,Scale,  
Inverse)
```

```
End Sub
```

```
' Beginning of the Increase_Decrease function
```

```
Sub Increase_Decrease(Destination,Source,Factor,Scale,Inverse)
```

```
If Inverse = False Then
```

```
HS.EXP Destination & " = " & Source & " *
```

```
(1 + (" & Factor & " / " & Scale & "))"
```

```
Else
```

```
HS.EXP Destination & " = " & Source & " *
```

```
(1 + ((" & Factor & " * -1) / " & Scale & ))"
```

```
End If
```

```
End Sub
```

Pro_Rata_Ratio

この関数は、2つのソースPOV ($C = A / B$)の比率を計算します。

戻り値

戻り値はありません。

構文

```
Pro_Rata_Ratio(  
Destination, SourceN, SourceD  
)
```

表109 Pro_Rata_Ratio関数の構文

パラメータ	有効な値
Destination	勘定科目、ICP、カスタム1から4のメンバーの有効な組合せである有効な宛先POV。
SourceN	ディメンション・メンバーの有効な組合せである有効なソースPOV。SourceNは比率計算の分子です。
SourceD	有効なソースPOV。SourceDは比率計算の分母です。

詳細説明

この関数は、2つのソースPOV ($C = A / B$)の比率を計算します。

ベスト・プラクティスとして、呼出しルーチンに変数を設定し、Destination、SourceNおよびSourceD POVを定義するPro_Rata_Ratio関数に渡されるようにすることをお勧めします。また、呼出しルーチンの変数名をPro_Rata_Ratio関数と同一になるように設定することもお勧めします。

親メンバーの加重平均比率は自動的に計算されません。親メンバーの値は、子の値の集約として表示されます。この結果、親メンバーの値は数学的には正しくなくなります。そのため、Ratio勘定科目では、集約を無効にしておくことをお勧めします。

例

MarginPct勘定科目は、GrossMargin/TotalRevenuesの値を戻します。

表110 Pro_Rata_Ratio関数の例

勘定科目	Jan2014	Feb2014	Mar2014
A#GrossMargin	1000	100	750
A#TotalRevenues	2000	400	1000

勘定科目	Jan2014	Feb2014	Mar2014
Pro_Rata_Ratio("A#GrossMargin", "#TotalRevenues")	0.50	0.25	0.75

PRO_RATA_RATIO関数から戻される結果は次のとおりです:

```
HS.EXP "A#MarginPct = A#GrossMargin / A# TotalRevenues"
```

サンプル・スクリプト

このスクリプトには、次の情報が含まれます:

- 呼出しルーチンで記述されたサンプル文。
- 呼出しルーチンに設定され、Pro_Rata_Ratio関数に渡される変数。
- Pro_Rata_Ratio関数と同一になるように設定された呼出しルーチンの変数名。

```
Sub Calculate()
```

```
    Dim Destination 'Destination POV
    Dim SourceN     'Source Numerator POV
    Dim SourceD     'Source Denominator POV
```

```
    Destination = "A#MarginPct"
    SourceN = "A#GrossMargin"
    SourceD = "A#TotalRevenues "
```

```
Call Pro_Rata_Ratio(Destination, SourceN, SourceD)
```

```
End Sub
```

```
' Beginning of the Pro_Rata_Ratio function
```



```
Sub Pro_Rata_Ratio(Destination,SourceN,SourceD)
```

```
HS.EXP Destination & " = " & SourceN & " / " & SourceD
```

```
End Sub
```

Spread

この関数は、ソース勘定科目の単一の期間値(たとえば、P#[Year])を、プロフィール勘定科目(収益プロフィール、4-4-5など)に定義されているプロフィールに基づいて、宛先勘定科目のすべての期間に配賦します。

戻り値

戻り値はありません。

構文

```
Spread(  
Destination, Source, Factor, FactorN, FactorD, Temp, Per  
)
```

表111 Spread関数の構文

パラメータ	有効な値
Destination	勘定科目、ICP、カスタム1から4のメンバーの有効な組合せである有効な宛先POV。
Source	ディメンション・メンバーの有効な組合せである有効なソースPOV。ソースPOVには、P#[Year]などの単一の期間を含める必要があります。単一の期間数が分散される数です。
Factor	有効なソースPOV。 <i>Factor</i> は、配賦係数の格納に使用される勘定科目です。
FactorN	有効なソースPOV。 <i>FactorN</i> は、分散配賦の基準として使用される分子係数です。
FactorD	有効なソースPOV。 <i>FactorD</i> は、分散配賦の基準として使用される分母係数です。

パラメータ	有効な値
Temp	有効な宛先勘定科目。 <i>Temp</i> は、ソース値を一時的に格納する勘定科目です。
Per	Januaryなど、タイムフレームの最初の期間の名前を定義する期間文字列。 <i>Temp</i> 値は最初の期間に格納され、パラメータは計算でこの値を参照する必要があります。

詳細説明

この関数は、Profile POV (Revenueプロファイル、4-4-5など)に定義されているプロファイルに基づいて、宛先POVのすべての期間にソースPOVの単一の期間値(P#[Year]など)を配賦します。

時間ベースの配賦は、まず1年間の金額が入力されてから、適切なプロファイルに基づいて期間全体に金額が配賦される、予算策定アプリケーションに特に適しています。

ソースPOVには、単一の期間を含める必要があります。期間は通常P#[Year]ですが、P#Januaryなど、任意の単一の期間にすることが可能です。

ソースPOVの値は、計算により一時勘定科目に格納されます。ソース勘定科目と宛先勘定科目が通常同一であるため、このようにする必要があります。これが当てはまる場合、P#[Year]の値は、計算が1つの期間から次の期間へ進むにつれて変わります。そのため、まず値を格納し、すべての期間で参照できるようにする必要があります。

呼出しルーチンに変数を設定し、Destination、Source、Profile、TempおよびPeriod1パラメータを定義するSpread関数に渡すことをお勧めします。また、呼出しルーチンの変数名をSpread関数と同一になるように設定することもお勧めします。

例

Telephone勘定科目の年の値が、4-4-5という四半期ごとの比率を使用して期間全体に配賦されます。

SPREAD関数から戻される結果は次のとおりです:

```
HS.EXP "A#TempTelephone.C1#[None] = A#Telephone.C1#[None].P#[Year]" (where
Period.Number = 1)
```

```
HS.EXP "A#Telephone.C1#[None] = A#TempTelephone P#January *
E.Globals.A#Profile445.C1#[None].P#Cur /
E.Globals.A#Profile445.C1#[None].P#[Year]"
```

サンプル・スクリプト

このスクリプトには、次の情報が含まれます:

- 呼出しルーチンで記述されたサンプル文。
- 呼出しルーチンに設定され、Spread関数に渡される変数。
- Spread関数と同一になるように設定された呼出しルーチンの変数名。

```
Sub Calculate()
```

```
Dim Destination
```

```
Dim Source
```

```
Dim Factor
```

```
Dim FactorN
```

```
Dim FactorD
```

```
Dim Temp
```

```
Dim Per
```

```
Source = "A#Telephone.C1#[None].P#[Year]"
```

```
Destination = "A#Telephone.C1#[None]"
```

```
Factor = "A#Factor.C1#[None]"
```

```
FactorN = "E#Globals.A#Profile445.C1#[None].P#CUR"
```

```
FactorD = "E#Globals.A#Profile445.C1#[None].P#[Year]"
```

```
Temp = "A#TempTelephone.C1#[None]"
```

```
Per = "January"
```

```
Call Spread(Destination,Source,Factor,  
FactorN,FactorD,Temp,Per)
```

```
End Sub
```

```
' Beginning of the Spread function
```

```
Sub Spread(Destination,Source,Factor,FactorN,FactorD,Temp,Per)
```

```
If HS.Period.Number = 1 Then
```

```
HS.Exp Temp & " = " & Source
```

```
End If
```

```
HS.Clear Factor  
HS.EXP Factor & " = " & FactorN & " / " & FactorD
```

```
HS.Clear Destination  
HS.EXP Destination & " = " & Temp & ".P#" & Per & " * " & Factor
```

```
End Sub
```

Units_Rates

この関数は、2つのソースPOV ($C = A * B$)の積を計算します。

戻り値

戻り値はありません。

構文

```
Units_Rates(  
Destination, Units, Rates  
)
```

表112 Units_Rates関数の構文

パラメータ	有効な値
Destination	勘定科目、ICP、カスタム1から4のメンバーの有効な組合せである有効な宛先POV。
Units	ディメンション・メンバーの有効な組合せである有効なソースPOV。

パラメータ	有効な値
Rates	有効なソースPOV。

詳細説明

この関数は、2つのソースPOV ($C = A * B$)の積を計算します。ベスト・プラクティスとして、呼出しルーチンに変数を設定し、Destination、UnitsおよびRates POVを定義するUnits_Rates関数に渡されるようにすることをお勧めします。また、呼出しルーチンの変数名をUnits_Rates関数と同一になるように設定することもお勧めします。

例

Sales勘定科目は、UnitsSold * Priceの値を戻します。

表113 Pro_Rata_Ratio関数の例

勘定科目	Jan2014	Feb2014	Mar2014
A#UnitsSold	1000	2000	5000
A#Price	1.25	1.00	0.50
Units_Rates("A#UnitsSold", A#Price)	1,250	2000	2500

UNITS_RATES関数から戻される結果は次のとおりです:

```
HS.EXP "A#Sales = A#UnitsSold * A#Price"
```

サンプル・スクリプト

このスクリプトには、次の情報が含まれます:

- 呼出しルーチンで記述されたサンプル文。
- 呼出しルーチンに設定され、Units_Rates関数に渡される変数。
- Units_Rates関数と同一になるように設定された呼出しルーチンの変数名。

```
Sub Calculate()
```

```
Dim Destination
```

```
Dim Units
```

```
Dim Rates
```

```
Destination = "A#Sales"  
Units = "A#UnitsSold"  
Rates = "A#Price"
```

```
Call Units_Rates(Destination,Units,Rates)
```

```
End Sub
```

```
' Beginning of the Units_Rates function
```

```
Sub Units_Rates(Destination,Units,Rates)
```

```
HS.EXP Destination & " = " & Units & " * " & Rates
```

```
End Sub
```

13

Calculation Manager を使用したルールの作成

この項の内容:

Calculation Managerのセキュリティ役割	439
Calculation Managerでのアプリケーションの操作	439
Calculation Managerへのルールの移行	440
関数セレクタでのVB関数のサポート	440
Financial Managementの特別なVB Script関数	442

計算マネージャ・モジュールは、Financial Managementの計算ルールを作成するための共通のユーザー・インタフェースを提供します。グラフィカルなフローによって、計算プロセスの理解を深めることができます。このフローでは、グラフィカル・ビューとVB Scriptビューを切り替えられます。計算マネージャは、すべての計算ルールを管理するための中央リポジトリを提供して、アプリケーション間でルールを共有します。計算ルールをインポート、エクスポートおよび印刷できます。また、ナビゲーションを簡単に行うためにカスタム・フォルダを作成できます。

Calculation Managerのセキュリティ役割

これらの役割は、Financial ManagementにおけるCalculation Managerアクセスで使用できます。

- ルール管理者 - ルール・オブジェクト、テンプレートおよび変数の作成、変更および削除や、ルール・セットの検証およびデプロイなど、指定されたアプリケーションに対してCalculation Managerのあらゆるタスクを実行できます
- ルール・デザイナー - ルール・オブジェクトを作成し、このオブジェクトを変更または削除できます
- ルール・ビューア - ルール・オブジェクトを表示および検証できます

Oracle Hyperion Enterprise Performance Management WorkspaceからCalculation Managerにアクセスするには、ルール管理者、ルール・デザイナーまたはルール・ビューアのセキュリティ役割が必要です。

ルールをデプロイするには、ルール管理者のセキュリティ役割が必要です。

セキュリティ役割の詳細は、*Oracle Enterprise Performance Management Systemユーザー・セキュリティ管理ガイド*を参照してください。

Calculation Managerでのアプリケーションの操作

Calculation Managerは、Performance Management Architectアプリケーションまたはクラシック・アプリケーションで使用できます。Calculation Managerは、Performance Management Architectとともにインストール

することも、別にインストールしてOracle Hyperion Enterprise Performance Management Workspaceからアクセスすることもできます。

アプリケーションを作成する場合、VB Scriptルールをロードすることも、Calculation Managerを使用してルールを設計し、Financial Managementアプリケーションにデプロイすることもできます。



注:

ルールをCalculation Managerにデプロイした後でVB Scriptルールをロードしようとする
と、Calculation Managerルールが上書きされるというプロンプトが表示されます。「OK」をクリックして続行するか、「取消し」をクリックします。



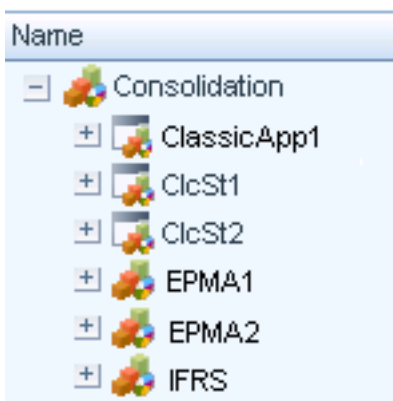
Calculation Managerの連結フォルダを開くときは、使用しているアプリケーションのリストがアルファベット順に表示されます。クラシック・アプリケーションはアイコンで示され、Oracle Hyperion EPM Architectアプリケーションでは、次のアイコンが使用されます:。

図1 サンプルCalculation Managerアプリケーション・リスト



アプリケーション・フォルダを展開して、ルール・セット、ルール、式、スクリプトおよびテンプレートを表示できます。Calculation Managerの使用方法的詳細は、『Oracle Hyperion Calculation Manager設計者ガイド』またはオンライン・ヘルプを参照してください。

Calculation Managerへのルールの移行

以前のリリースの既存のVB Scriptルール(.r1e)ファイルがある場合、それらのファイルをインポートしてCalculation Managerに移行できます。『Oracle Hyperion Calculation Manager設計者ガイド』のルールの移行に関する項を参照してください。

関数セクタでのVB関数のサポート

Calculation Managerの関数セクタはVB関数をサポートします。



注:

VB関数が選択のUIで使用できない場合でも、スクリプトのコンポーネントでは、他のVB関数も使用できます。

配列関数

関数	説明
Array	配列を含むvariantを返します。
Filter	フィルタ条件に基づいた文字列配列のサブセットを含むゼロから始まる配列を返します。
Join	配列内の多くの部分文字列で構成される文字列を返します。
LBound	指定された配列の次元の最も小さいサブスクリプトを返します。
Split	指定された数の部分文字列を含む、ゼロから始まる一次元配列を返します。
UBound	指定された配列の次元の最も大きいサブスクリプトを返します。

日付/時刻の関数

関数	説明
Date	現在のシステム日付を返します。
DateAdd	指定された時間間隔が追加された日付を返します。
DateDiff	2つの日付間の間隔数を返します。
DatePart	指定された日付の指定された部分を返します
DateSerial	指定された年、月および日の日付を返します。
Day	月の日(1から31まで)を表す数字を返します。
Month	年の月(1から12まで)を表す数字を返します。
MonthName	指定された月の名前を返します。

数学

関数	説明
Abs	指定された数の絶対値を返します。
Fix	指定された数の整数部を返します。
Int	指定された数の整数部を返します。

文字列

関数	説明
InStr	別の文字列内で、ある文字列が最初に出現する位置を返します。検索は文字列の最初の文字から始めます。
InStrRev	別の文字列内で、ある文字列が最初に出現する位置を返します。検索は文字列の最後の文字から始めます。
LCase	指定された文字列を小文字に変換します。
Left	文字列の左側から指定された数の文字を返します。

関数	説明
Len	文字列内の文字数を返します。
Mid	文字列から指定された数の文字を返します。
Right	文字列の右側から指定された数の文字を返します。
StrComp	2つの文字列を比較し、比較の結果を表す値を返します。
Trim	文字列の左と右の両側にあるスペースを除去します。
UCase	指定された文字列を大文字に変換します。

Financial Managementの特別なVB Script関数

これらの特別な関数は、Oracle Hyperion Calculation ManagerのUIで配列やループを処理するために、Financial Management用に実装されました。

Range

値	ループ変数	VBScript生成
@Range(1-50)	i	Dim i(50) I(1)=1 I(2)=2 I(3)=3 .. I(50)=50
@Range(5-10)	i	Dim i(6) I(1)=5 I(2)=6 I(3)=7 I(4)=8 I(5)=9 I(6)=10
@Range(1,3-5,7-9)	i	Dim i(8) I(1)=1 I(2)=3 I(3)=4 I(4)=5 I(5)=7 I(6)=8 I(7)=9

For/ForStep

値	ループ変数	VBScript生成	コメント
@For(2,10)	アイテム	For Item=2 to 10	New @ForLoop @For(from, to)
@ForStep(2,10,2)	アイテム	For Item =2 to 10 step 2	New @ForStep loop function @ForStep(from, to, step) 注: 逆のステップが必要な場合は、ステップの前に負数の符号(-)を追加します。例: @ForStep(2,10,-2)

ExitFor

@ExitFor - ループの終了

VB Scriptの次の文が生成されます。

For each *element* in group

[*statements*]

Exit For

[*statements*]

Next [*element*]

または

For *counter*=start To end [Step *step*]

[*statements*]

Exit For

[*statements*]

Next [*counter*]

ExitSub

@ExitSub - ルールの終了

VB Scriptの次の文が生成されます。

Sub *name* [(*x,y*)]

[*statements*]

ExitSub

[statements]

EndSub

ReDim

1つ以上の動的配列変数を再次元化し、それらのストレージ・スペースを再割り当てします。オプションのPreserveキーワードを使用すると、配列を再次元化するときに配列の内容を現状のままに維持できます。

```
{VarArrayX(5)} = @Redim
```

```
{VarArrayXY(5,9)} = @Redim
```

次のVB文が生成されます。

```
Redim VarArrayX(5)
```

```
Redim VarArrayXY(5,9))
```

RedimPreserve

```
{VarArrayXY(5)} = @RedimPreserve
```

または

```
{VarArrayXY(5,9)} = @RedimPreserve
```

または

```
{VarArrayXY(5,{i})} = @RedimPreserve
```

次のVB文が生成されます。

```
RedimPreserve VarArrayX(5)
```

または

```
RedimPreserve VarArrayXY(5,9)
```

または

```
RedimPreserve VarArrayXY(5,i)
```

この項の内容:

内部取引の設定	445
理由コードの管理	448
内部取引のモニタリング	449
エンティティのロックとロック解除	451
内部取引の要約の表示	452

内部取引の設定

内部取引は、組織内の2つのエンティティの間での取引です。Oracle Hyperion Financial Managementでは、勘定科目とカスタム・ディメンションにまたがって、内部取引の詳細を追跡して調整できます。内部取引モジュールでは、内部取引勘定科目の差異の特定、レポート作成、調整を効率的に行えます。

内部取引の管理ページを使用して、内部取引を処理します。取引の処理については、『Oracle Hyperion Financial Management ユーザー・ガイド』を参照してください。

内部取引を入力する前に、次の設定手順を実行する必要があります:

- 内部取引の期間を開きます。445ページの[内部取引の期間を開く](#)を参照してください。
- 内部取引ルールを定義して、アプリケーションにロードします。HS.SupportsTran関数は、内部取引をサポートするアプリケーションの勘定科目、シナリオおよびカスタム・ディメンションを指定します。246ページの[ルール・ファイルの作成](#)および248ページの[ルールのロード](#)を参照してください。
- 照合許容差を定義します。446ページの[照合許容差を設定する](#)を参照してください。
- 取引のステータスが不一致である理由を示す理由コードを定義します。448ページの[理由コードの追加](#)を参照してください。
- 通貨換算レートを入力します。85ページの[通貨の定義](#)を参照してください。

内部取引の期間を開く

内部取引を入力、ロードまたは処理する前に、まず取引の期間を開く必要があります。期間には、「開いていない」、「開いている」または「閉じています」のステータスがあります。期間のデフォルトのステータスは「開いていない」です。期間が開いて、取引が入力された後は、「締切り済」にのみ変更できます。「開いていない」には戻せません。

期間ごとに、「転記前に照合」オプションを設定し、自動照合や手動照合のプロセスに適用される照合許容差を入力できます。447ページの[「転記前に照合」オプションの設定](#)および446ページの[照合許容差を設定する](#)を参照してください。

内部取引期間を開くには、「アプリケーション管理者」のセキュリティ役割が必要です。

▶ 期間を開くには:

1. 「連結」、「内部取引」、「期間」の順に選択します。
2. 「シナリオ」で、期間のシナリオを選択します。
3. 「年」から、期間の年を選択します。
4. 開くそれぞれの期間を選択します。
5. オプション: 期間の取引ID許容差値またはパーセント、勘定科目の許容差値、または手動の許容差値を入力します。
6. 「転記前に照合/検証」列で、オプションを選択します。

- 取引を転記する前に照合ステータスを確認する必要がある場合は、「はい」または「制限」を選択します。
- 照合ステータスを確認する必要がない場合は、「いいえ」を選択します。

7. 「期間を開く」をクリックするか、または「アクション」、「期間を開く」の順に選択します。

期間の現在のステータスが「開いている」に変わります。

8. オプション: 期間の設定を保存するには、「期間の設定の保存」をクリックするか、または「アクション」、「期間の設定の保存」の順に選択します。



ヒント:

変更を保存せずに期間を元のステータスに復元するには、「復元」をクリックするか、または「アクション」、「復元」の順に選択します。

照合許容差を設定する

自動照合と手動照合のプロセスに期間別の照合許容差を設定できます。勘定科目と手動照合の許容差に値を設定できます。取引ID (TID)許容差の場合は、値、パーセントまたはその両方を指定できます。

取引IDにパーセントを入力すると、エンティティの取引の合計とパートナーの取引の合計で小さい方の値が使用され、その値にパーセントが適用されて、許容値が生成されます。

たとえば、TID123のエンティティAで3つの取引を持ち、これらの取引の合計が1000であるとして、TID123のパートナーBが合計1020の5つの取引を持ちます。エンティティの合計とパートナーの合計の差は20です。ただし、3%の許容差を指定すると、小さい方の合計の3%(1000x3%)が計算され、結果が30になります。その値を差と比較すると、許容差内にあるため、取引は一致とみなされます。

パーセント以外に、たとえば値15を入力すると、パーセント値が入力された値と比較され、最も小さい方の値が比較に使用されます。この例では、エンティティの合計とパートナーの合計の差は20、パーセントの許容差は30、値の許容差は15です。これは、許容差内とみなされないため、取引は一致しません。

TIDの値とパーセントを空白のままにすることもできます。いずれかがゼロ値の場合、または両方が空白の場合、取引の差がゼロの取引のみが照合されます。

勘定科目の許容差と手動照合の許容差の場合、照合許容差は、アプリケーションの通貨で表され、入力された値はアプリケーションの通貨の位取りで表されます。照合プロセス時に、各取引がアプリケーションの通貨に変換され、合計値の差が期間に設定された照合許容差と比較されます。比較は1の単位で実行されます。

取引をTID許容値と比較する場合は、TID/RIDの照合プロセス時に、取引はアプリケーションの通貨に換算されません。

「転記前に照合」オプションの設定

期間ごとに、「転記前に照合」オプションを設定できます。照合/検証オプションでは、取引の転記前に取引の照合ステータスを確認する必要があるかどうかを定義し、エンティティをロックするか期間を閉じる前に実行する必要がある検証タイプを定義します。

「転記前に照合」オプションを選択する場合は、転記する前に、取引を一致させるか理由コードを割り当てる必要があります。期間を閉じる前、またはエンティティをロックする前には、一致した取引または理由コードのある不一致の取引すべてを転記する必要があります。

「制限」オプションを選択する場合は、転記する前に取引を一致させる必要がありますが、転記が戻されている取引を含む期間を閉じることや、エンティティをロックすることは可能です。

表114 「転記前に照合」オプションの値の説明

「転記前に照合」オプションの値	説明
いいえ	すべての取引を転記できます。
はい	次の条件のいずれかを満たす場合は、取引を転記できます。 <ul style="list-style-type: none"> 取引のステータスが「一致」である。 ステータスが「不一致」の取引に有効な理由コードが含まれている。 <p>注:</p> <p>未照合の取引または理由コードのない不一致の取引は転記できません。</p> <p>また、期間を閉じるかエンティティをロックする前に、一致した取引または理由コードのある不一致取引のすべてが転記されているかどうかチェックされます。</p>
制限	「転記前に照合」オプションを「制限」に設定すると、次のいずれかの条件が満たされた場合に取引を転記できます: <ul style="list-style-type: none"> 取引のステータスが「一致」である。 ステータスが「不一致」の取引に有効な理由コードが含まれている。 <p>注:</p> <p>未照合の取引または理由コードのない不一致の取引は転記できません。</p> <p>転記が戻されている取引を含む期間を閉じることや、エンティティをロックすることは可能です。</p>

内部取引の期間を閉じる

内部取引を処理したら、期間を閉じて、取引の変更を防ぐことができます。「転記前に照合」オプションを選択した場合は、一致した取引および理由コードのある不一致の取引すべてを転記する必要があります。後続の取引用に期間はロックされますが、取引の参照や、期間に対するレポートの実行は可能です。

内部取引期間を閉じるには、「アプリケーション管理者」のセキュリティ役割が必要です。

▶ 期間を閉じるには:

1. 「連結」、「内部取引」、「期間」の順に選択します。
2. 「シナリオ」リストから、期間を閉じるシナリオを選択します。
3. 「年」リストから、期間を閉じる年を選択します。
4. 閉じる期間を選択します。
5. 「期間を閉じる」をクリックするか、または「アクション」、「期間を閉じる」の順に選択します。

理由コードの管理

内部取引がアプリケーションで作成されると、デフォルトの照合ステータスは「未照合」になります。自動照合プロセス時に、照合ステータスは「一致」または「不一致」に更新されます。

取引が「不一致」ステータスを持つ理由を示す理由コードを定義できます。たとえば、パートナ・エンティティの請求書がないや、パートナが正しくない値を入力したなどの理由があります。アプリケーションの有効な理由コードのリストを定義したら、ユーザーは、内部取引を入力するときに、そのリストから理由コードを選択して割り当てることができます。

期間に転記前に照合/検証オプションを選択した場合は、「一致」ステータスの取引または、有効な理由コードを含む「不一致」ステータスの取引を転記できます。

アプリケーションの理由コードは追加、編集、削除できます。

次の手順を参照してください:

- [448ページの理由コードの追加](#)
- [449ページの理由コードの編集](#)
- [449ページの理由コードの削除](#)

理由コードの追加

アプリケーションの不一致取引の理由コードのリストを作成できます。作成したら、ユーザーは取引の理由コードの1つを割り当てることができます。

理由コードを手動で追加したり、取引のロード・プロセス時にロードできます。取引のロードについては、『*Oracle Hyperion Financial Management ユーザーガイド*』を参照してください。

▶ 理由コードを追加するには:

1. 「連結」、「内部取引」、「理由コード」の順に選択します。

2. 「**行の追加**」をクリックするか、または「**アクション**」、「**行の追加**」の順に選択します。
3. 「**名前**」に、理由コードのラベルを入力します。



注:

ラベルには最大20文字まで入力できます。スペースも1文字としてカウントされることに注意してください。次の文字は無効です: アンパサンド(&)、アスタリスク(*)、円記号(バックスラッシュ) (¥)、カンマ(,)、中カッコ({}), スラッシュ(/)、ハイフン(-)、数字記号(#)、ピリオド(.)、プラス記号(+)およびセミコロン(;)。

4. オプション: 「**説明**」に、理由コードの説明を入力します。



注:

説明には40文字まで入力できます。

5. 「**保存**」をクリックするか、「**アクション**」、「**保存**」の順に選択します。

理由コードの編集

理由コードを作成したら、そのコードの説明を編集できます。理由コードのラベルは編集できません。

▶ 理由コードを編集するには:

1. 「**連結**」、「**内部取引**」、「**理由コード**」の順に選択します。
2. 理由コードのリストから、編集する理由コードを選択します。
3. 「**説明**」列で説明を編集し、「**OK**」をクリックします。

理由コードの削除

アプリケーションの理由コード・リストから不要となった理由コードを削除できます。

▶ 理由コードを削除するには:

1. 「**連結**」、「**内部取引**」、「**理由コード**」の順に選択します。
2. 理由コードのリストから、削除する理由コードを選択します。
3. 「**行の削除**」か「**すべて削除**」をクリックするか、または「**アクション**」、「**行の削除**」か「**すべて削除**」の順に選択します。
4. 「**はい**」をクリックして、理由コードを削除します。

内部取引のモニタリング

内部取引のモニター機能を使用して、内部取引照合プロセスをモニターします。多数の内部取引が期間内に入力されると、すべての取引が正常に輸入されて一致することを確認するための照合プロセスに時間がかかります。同時にすべての取引が入力されるとは限らないため、管理者は照合プロセスをモニターする必要があります。内部取引のモニター機能を使用すると、内部取引プロセスを開始した内部取引パートナーを簡単に見つけることができます。

内部取引をモニターするには、「内部取引管理者」のセキュリティ役割が割り当てられている必要があります。この役割は、プロセス・ステータス、ロック・ステータス、エンティティの詳細、内部取引の要約を表示し、電子メール・アラートを実行できます。

内部取引のモニター・ページには、プロセス・ステータスとロック・ステータスが付いた内部取引エンティティのリストが表示されます。エンティティ、プロセス・ステータスまたはロック・ステータスで表示をフィルタできます。エンティティは、内部取引のモニター詳細情報にリンクされます。エンティティをクリックすると、「一致」、「不一致」、「未照合」などのステータス別に転記済と未転記の取引数が表示されます。

表115 内部取引のエンティティ・ステータス

ステータス	説明
未開始	エンティティには、視点のシナリオ、年および期間に内部取引がありません。
開始済	エンティティには、視点のシナリオ、年および期間に少なくとも1つの内部取引があります。
ロック不可	その期間に「転記前に照合」を選択したが、一致した取引も理由コードのある不一致の取引も転記しなかった場合、エンティティのステータスは「ロック不可」になります。

リスト内のエンティティをクリックして、照合ステータスで分類されたエンティティの転記済および転記を戻し済の合計取引数を表示できます。

たとえば、エンティティAの横にあるプラス(+)記号をクリックして展開すると、ページには、エンティティAの転記済の取引と転記の戻し済の取引の数がステータス別に表示されます。「未照合」列で転記を戻し済の取引の値をクリックすると、フィルタが適用された内部取引の管理ページにリンクされ、ステータスが「未照合」のエンティティAの転記を戻した取引が表示されます。同時に複数のエンティティを展開して、その詳細を表示できます。

内部取引のモニター・ページで任意のエンティティに電子メール・アラートを送信できます。『Oracle Hyperion Financial Management ユーザー・ガイド』を参照してください。

▶ 内部取引をモニターするには:

1. 「連結」、「内部取引」、「モニター」の順に選択します。
2. 視点バーで、シナリオ、年、期間を選択します。
3. 「表示オプション」リストから、次のオプションを選択します:
 - ラベルを使用してエンティティ情報を表示するには、「ラベル」を選択します。
 - 説明を使用してエンティティ情報を表示するには、「説明」を選択します。
 - ラベルと説明を使用してエンティティ情報を表示するには、「両方」を選択します。
4. 「フィルタ」リストから、「エンティティ」に、ステータスをモニターするエンティティを入力または選択します。



注:

「エンティティ」を空白のままにした場合は、すべてのエンティティがリストに表示されます。

5. オプション: アクティブなエンティティのみを表示するには、「アクティブなもののみ表示」を選択します。
6. オプション: 「プロセスのステータス」によって取引リストをフィルタするには、次の取引タイプのうち1つ以上を選択します:

- ・ 未開始
 - ・ 開始済
7. オプション:「ロックのステータス」によって取引リストをフィルタするには、次の取引タイプのうち1つ以上を選択します:
- ・ ロック可
 - ・ ロック不可
 - ・ ロック済
8. オプション:「プロセス」または「ロック」ステータスを基準にして取引リストをソートするには、列見出しをクリックし、「昇順」または「降順」を選択します。
9. エンティティの横にあるプラス記号(+)をクリックするか、コンテキスト・メニューから「詳細の表示」を選択して、「内部取引のモニター詳細」を表示します。
10. 「内部取引のモニター詳細」ボックスで、いずれかのステータス列の値をクリックします。
- 選択したフィルタ条件とともに、内部取引の管理ページが表示されます。
11. オプション: エンティティの電子メール・アラートを送信するには、内部取引のモニターページから、エンティティを選択して「電子メール・アラート」をクリックするか、コンテキスト・メニューから「電子メール・アラート」を選択するか、「アクション」、「電子メール・アラート」の順に選択します。

エンティティのロックとロック解除

シナリオ、年および期間のエンティティにロックを適用して、エンティティの内部取引の以後の変更を防止できます。その期間に対して「転記前に照合」オプションが選択されている場合、エンティティをロックする前に、一致した取引および理由コードのある不一致の取引すべてを転記する必要があります。



注:

取引ロック・ステータスはデータ・ロック・ステータスとは異なります。データ・ロック・ステータスについては、『Oracle Hyperion Financial Management ユーザー・ガイド』を参照してください。

「内部取引管理者」のセキュリティ役割が割り当てられている場合、プロセス・ステータス、ロック・ステータス、エンティティの詳細、内部取引の要約を表示し、電子メール・アラートを実行できます。

エンティティがロックされると、新規の内部取引を入力できません。また、既存の取引を削除または変更することもできません。ロックされたエンティティに取引を転記すること、転記を戻すこと、またはロックされたエンティティを含む取引の照合ステータスを更新することはできません。したがって、パートナー・エンティティがロックされていない場合でも、エンティティの照合ステータスは更新できないため、パートナーは取引を照合できません。

たとえば、エンティティAがロックされているとします。エンティティAの以後の取引を入力できません。エンティティには転記や照合を実行できません。エンティティBがロックされていない場合は、パートナーのエンティティAとのエンティティBの内部取引を持つことができます。ただし、エンティティBをエンティティAと照合しようとする、エンティティAの照合ステータスを更新できないため、プロセスは失敗します。

提出フェーズを使用している場合は、すべてのフェーズが「発行済」ステータスを持つまでエンティティをロックできません。

▶ エンティティをロックするには:

1. 「連結」、「内部取引」、「モニター」の順に選択します。
2. 「シナリオ」で、エンティティのシナリオを選択します。
3. 「年」から、エンティティの年を選択します。
4. 「期間」から、エンティティの期間を選択します。
5. ロックするエンティティを選択します。
6. 「ロック」をクリックするか、またはコンテキスト・メニューから「ロック」を選択するか、または「アクション」、「ロック」の順に選択します。



ヒント:

エンティティのロックを解除するには、ロックを解除するエンティティを選択し、「ロック解除」をクリックするか、またはコンテキスト・メニューから「ロック解除」を選択するか、または「アクション」、「ロック解除」の順に選択します。

内部取引の要約の表示

すべての内部取引のステータスの要約を表示し、必要な場合、特定のエンティティを表示するためにエンティティをフィルタできます。

▶ 内部取引の要約を表示するには:

1. 「連結」、「内部取引」、「モニター」の順に選択します。
2. エンティティのシナリオ、年および期間を選択します。
3. エンティティ・フィルタを使用して、取引の要約を表示する1つ以上のエンティティを選択します。
4. 「要約」をクリックするか、またはコンテキスト・メニューから「要約」を選択するか、または「アクション」、「要約」の順に選択します。
5. 要約を確認したら、「OK」をクリックします。

15

プロセス管理の提出フェーズの管理

この項の内容:

提出フェーズの定義	453
送信グループの設定	454
提出グループおよびフェーズの例	455
提出グループのフェーズへの割当て	457
未割当ての提出グループの表示	458

プロセス管理は、財務データの確認と承認を管理します。データは確認目的でプロセス単位に編成されており、特定のシナリオ、年、期間、エンティティおよび値ディメンションのデータが組み合されています。確認サイクル中に、レベルの移行、提出、承認、却下および公開などのアクションをプロセス単位に実行できます。

プロセス管理では、データ・プロセス単位を異なる提出フェーズに分割し、データのサブセットを処理できます。確認プロセス時に、プロセス単位全体ではなく、フェーズのプロセス単位ごとにレベルを上げることができます。これにより、シナリオを追加して確認プロセスを適用する必要がなくなりました。

プロセス管理の提出フェーズを管理するには、UseSubmissionPhaseアプリケーション属性が有効になっている必要があり、アプリケーション管理者または確認監督者である必要があります。

提出フェーズの定義

確認プロセス要件は、期間ごとに異なる場合があります。たとえば、月締めサイクルは、1月と2月の貸借対照表と損益計算書のデータで単一フェーズの確認プロセスを必要とする場合があります。3月などの四半期月の場合、四半期締めの確認プロセスは、貸借対照表、損益計算書および補足のデータで複数フェーズの送信サイクルを必要とします。

確認プロセス要件は、シナリオごとに異なる場合もあります。たとえば、実績シナリオは、確認のために提出される貸借対照表と損益計算書の勘定科目のみを必要とします。予算シナリオの場合は、すべての勘定科目を必要とし、予測シナリオの場合は、損益計算書の勘定科目と補足データのみを必要とします。

確認プロセスでは、9つまでの送信フェーズを使用できます。たとえば、実績シナリオでは、フェーズ1でレビューの貸借対照表と損益計算書の勘定科目を提出し、フェーズ2で補足データを提出します。予算シナリオでは、フェーズ1で内部取引データを提出し、フェーズ2で貸借対照表と損益計算書の勘定科目を提出し、フェーズ3で補足データを提出します。

提出フェーズを設定するには、次のタスクを実行します:

- アプリケーションとディメンション・メタデータの属性を設定して、提出フェーズを使用します。[78ページのアプリケーション設定の定義](#)を参照してください。
 - UseSubmissionPhaseアプリケーション属性をYに設定します。デフォルトでは、提出フェーズの設定が無効なため、この属性を設定してアプリケーションのフェーズ送信を有効にする必要があります。
 - 必要に応じて、SupportSubmissionPhaseForAccounts、SupportSubmissionPhaseForCustomまたはSupportSubmissionPhaseForICP属性を設定します。プロセス管理に有効にするディメンション(勘定科目、カスタムおよびICPメンバー)を定義する必要があります。たとえば、アプリケーションで勘定科目ディメンションによる提出が必要で、カスタム・ディメンションまたは内部取引(ICP)ディメンションからは必要としない場合は、SupportSubmissionPhaseForAccounts属性を選択します。少なくとも1つのディメンションを有効にする必要があります。
- 検証勘定を割り当てます。[78ページのアプリケーション設定の定義](#)を参照してください。

検証勘定は、データのロックのために、プロセス・コントロールで使用されます。データのレベルを上げたり、承認やロックを行う前には、検証勘定の金額はゼロである必要があります。

プロセス・コントロールで、またはデータのロックのために、検証を使用しない場合、検証勘定を空白のままにします。

フェーズ提出を使用している場合、各提出フェーズに個別の検証勘定を指定できます。アプリケーションには、最大9個の提出フェーズを含めることができます。フェーズ提出を使用するアプリケーションの場合、検証勘定設定は、フェーズ1の検証勘定を識別します。検証勘定2から9までは、残りのフェーズを識別します。
- 提出グループを定義して、提出グループをディメンション・メンバーに割り当てます。[454ページの送信グループの設定](#)を参照してください。
- 提出グループを提出フェーズに割り当てます。[457ページの提出グループのフェーズへの割当て](#)を参照してください。

送信グループの設定

アプリケーションごとに、送信グループに含めるディメンション・メンバーを決定する必要があります。たとえば、送信グループ1に現金勘定科目および投資勘定科目を定義できます。

メタデータ・ファイルで、送信グループに次の属性を設定します: SubmissionGroup=0から99または<blank>。

デフォルトは空白です。<blank>値にすると、デフォルトの値1になります。

送信グループを0に設定すると、勘定科目が確認プロセスに含まれなくなります。

複数のディメンションがフェーズ送信に使用される場合、ディメンション・メンバーの最大のグループ割当てによってセル送信グループ割当てが決定されます。送信グループを割り当てる前に、必要なすべてのグループ割当てを検討する必要があります。

例1:

Account=2

C1=1

C2=2

C3=1

C4=1

ICP=1

これらのディメンションの最大送信グループ数が2であるため、セルの送信グループ値は2となります。

例2:

Account=1

C1=3

C2=2

C3=5

C4=1

ICP=3

これらのディメンションの最大送信グループ数が5であるため、セルの送信グループ値は5となります。

提出グループおよびフェーズの例

フェーズのアプリケーションとディメンション・メンバーの属性を設定し、ディメンション・メンバーの提出グループを定義したら、勘定科目の提出グループを各提出フェーズに割り当てることができます。割り当ては、プロセス管理をサポートするシナリオにのみ適用されます。ただし、割り当ては、シナリオと期間によって実行される必要があります。

これらの例は、サンプルの提出グループと提出フェーズへのそれらの割り当てを示します。

会計	提出グループ
HistData	0

会計	提出グループ
Cash	1
Invest	1
ICRec	2
ICPay	2
Liability	3
Equity	3
Revenue	4
Expense	4
SuppData1	5
SuppData2	5
Headcount	6
MiscData	6

C1(製品)	提出グループ
[None]	1
Golf Balls	7
Tennis Balls	8
Soccer Balls	9

基本勘定科目は、親勘定科目から提出グループを継承しません。親勘定科目は、子の提出グループを前提にしません。提出グループを各勘定科目に割り当てる必要があります。この例では、HistData勘定科目は提出グループ0の割当てを持ちます。これは、勘定科目が確認プロセスを必要としないことを示します。

この例では、実績シナリオの期間による提出グループの割当てを示します。

表116 例: 提出フェーズへの提出グループの割当て

期間	提出フェーズ1	提出フェーズ2	提出フェーズ3
1月(単一フェーズ)	1, 2, 3, 4	該当なし(補足データまたは製品データは必要なし)	該当なし(補足データまたは製品データは必要なし)
2月	1, 2, 3, 4	N/A	N/A
3月(複数フェーズ)	2	1, 3, 4, 7, 8, 9	5, 6
4月	1, 2, 3, 4	N/A	N/A
5月	1, 2, 3, 4	N/A	N/A
6月(複数フェーズ)	2	1, 3, 4, 7, 8, 9	5, 6
7月	1, 2, 3, 4	N/A	N/A
8月	1, 2, 3, 4	N/A	N/A
9月(複数フェーズ)	2	1, 3, 4, 7, 8, 9	5, 6
10月	1, 2, 3, 4	N/A	N/A
11月	1, 2, 3, 4	N/A	N/A
12月(複数フェーズ)	2	1, 3, 4, 7, 8, 9	5, 6

1月 - 単一フェーズの割当て

この例の1月の月締めでは、提出フェーズ1のプロセス管理が必要ですが、提出フェーズ2と3は必要ありません。これは、短い月締めサイクルであるため、内部取引の貸借対照表、損益計算書のデータ(グループ1、2、3、4)が、同じ提出フェーズで提出されます。補足データは必要ありません。

3月 - 複数フェーズの割当て

3月の四半期締めでは、プロセス管理が複数フェーズにわたるデータ提出を使用します。

提出フェーズ1には内部取引データが必要であるため、ICRecおよびICPay勘定科目のデータが含まれます。この例では、提出グループ割当て2の全勘定科目となります。

3月の提出フェーズ2には、貸借対照表および損益計算書勘定科目(Cash, Invest, Liability, Equity, Revenue, Expense)が含まれます。この例では、提出グループ割当て1、3、4の全勘定科目となります。

3月の提出フェーズ3には補足データが含まれます。また、SuppData1、SuppData2、Headcount、MiscDataなどの補足データも含まれます。この例では、提出グループの割当て5と6の全勘定科目となります。

提出グループのフェーズへの割当て

各提出フェーズに提出グループを割り当てることができます。割当ては、プロセス管理をサポートするシナリオにのみ適用されます。グループが指定されていない場合、プロセス管理はグループ内のディメンショナル・メンバーに適用されません。

プロセス管理の提出グループを割り当てするには、アプリケーション管理者または確認監督者である必要があります。

提出グループをシナリオおよび期間別のフェーズに割り当てます。提出フェーズに1つ以上のグループを入力し、複数のグループ割当てにカンマを区切り文字として使用する(たとえば、1, 5, 6, 8, 9)ことができます。グループの範囲を指定できます。たとえば、グループ1、2、3、4、5、7および8を提出フェーズに割り当てするには、1-5, 7-8と指定します。ある範囲内の1つ以上のグループをカンマを使用して入力すると、データを提出してリフレッシュした際に、範囲のグループが表示されます(たとえば、1, 2, 3, 4が1-4として表示されます)。

有効なグループは1から99です。提出フェーズ1のデフォルトは、すべてのグループを表すキーワードALLです。すべてのグループは、割当てを変更するまで提出フェーズ1に属します。

同じグループを同じ期間内の複数のフェーズに割り当てることはできません。たとえば、グループ2から5をフェーズ1に指定し、グループ3と8をフェーズ2に指定できません。グループ3をフェーズ1とフェーズ2の両方に割り当てることができないからです。送信グループは、同じ期間内の1つのフェーズにのみ割り当てることができます。送信グループをすでに送信グループが割り当てられたフェーズに割り当てようとすると、エラー・メッセージが表示されます。すべてのグループを別の送信フェーズに移動すると、指定されたシナリオと期間の元のフェーズが完全にクリアされます。

提出フェーズの割当てをスキップできます。たとえば、グループを提出フェーズ2に割り当てないで、提出フェーズ1および3に割り当てることができます。提出フェーズに割り当てられていないグループは、確認プロセスの一部とみなされません。これらのディメンショナル・メンバーは、確認レベルのセキュリティの確認をせずに、該当するセキュリティクラスのアクセス権を持つすべてのユーザーが使用できます。データを入力する前に、未割当てのセルをプロセス管理で開始する必要はありません。

▶ 提出グループを提出フェーズに割り当てるには:

1. 「連結」、「データ」、「提出フェーズ」の順に選択します。

デフォルトでは、すべてのフェーズが表示されます。右側の「オプション」ペインで、非表示にするフェーズをクリックできます。

2. シナリオを変更するには、POVでシナリオ・ディメンションをクリックし、シナリオを選択して「OK」をクリックします。
3. 各フェーズ列で、その提出フェーズのグループを入力し、**[Enter]**を押します。
 - 複数のグループを入力するには、区切りとしてカンマを使用します。
 - グループの範囲を指定するには、区切りとしてマイナス記号を使用します。
 - すべてのグループを示すには、「すべて」を指定します。
4. 「提出」をクリックするか、または「アクション」、「提出」の順に選択して、データを保存します。
5. 「リフレッシュ」をクリックするか、または「アクション」、「リフレッシュ」の順に選択して、データベースをリフレッシュします。

未割当ての提出グループの表示

「プロセス・コントロール」で、どのフェーズにも割り当てられていない提出グループを表示できます。提出フェーズに割り当てられていない提出グループは、確認プロセスの一部ではありません。未割当てグループを表示すると、グループが間違っ確認プロセスから見落とされていないかを確認できます。

どの提出フェーズにも割り当てられていないディメンション・メンバーに割り当てられたグループと、どのディメンション・メンバーにも割り当てられていない提出フェーズに割り当てられたグループが表示されます。グループがディメンション・メンバーとフェーズのいずれにも割り当てられていない場合、そのグループは表示されません。

たとえば、アプリケーションで1-10の提出グループが割り当てられているとします。

期間	提出フェーズ1	提出フェーズ2	提出フェーズ3
1月	1,2,3,4	5,7	8,9
2月	1,2,3,4	5-8	N/A
3月	2	1,3,4	5,6
4月	1,2,3,4	5,6	8
5月	1,2,3,4	5-8	N/A
6月	2	1,3,4	5,6
7月	1,2,3,4	N/A	N/A
8月	1,2,3,4	N/A	N/A
9月	2	1,3,4	5,6
10月	1,2,3,4	N/A	N/A
11月	1,2,3,4	NA	N/A
12月	2	1,3,4	5,6

上記の例の1月について未割当てのグループを表示するオプションを選択すると、次のグループが未割当てとして表示されます。

期間	未割当てのグループ
1月	6,10

▶ 未割当て提出グループを表示するには:

1. 「連結」、「データ」、「提出フェーズ」の順に選択します。

未割当てグループ情報は、右側のペインに表示されます。

2. 期間の行を選択し、「未割当てグループ」ペインで「リフレッシュ」をクリックします。

16

電子メール・アラートの管理

この項の内容:

プロセス管理のアラートの設定	461
内部取引のアラートの設定	462

内部取引とプロセス管理確認プロセスで電子メール・アラートを使用できます。電子メール・アラートを使用すると、主要なイベントまたはシステム内でのデータの変更をハイライトできます。たとえば、内部取引が不一致であるため一致させる必要があること、またはプロセス単位を次のレベルに上げられることを電子メール・アラートで送信できます。

電子メール・アラートは、標準のSMTP(Simple Mail Transfer Protocol)を使用して送信されるため、インターネット電子メールで動作するすべての電子メール・システムでアラートを使用できます。アラートを使用するには、Financial Management構成ユーティリティを実行する場合、電子メール設定を構成し、SMTPサーバー名を指定する必要があります。『Oracle Enterprise Performance Management Systemインストールおよび構成ガイド』を参照してください。

アラート・プロセスでは、LDAP、MSAD、Native Directoryなどの認証ファイルに保管された電子メール・アドレスを使用します。

電子メール・アラートを送受信する前に、アプリケーションでユーザーとデータのセキュリティ権限を設定する必要があります。アラートのシナリオおよびエンティティに割り当てられたセキュリティ・クラスでは、電子メール・アラートをサポートする必要があり、ユーザーには電子メール・アラートを受信するセキュリティ役割が割り当てられている必要があります。セキュリティ役割の設定については、Oracle Enterprise Performance Management Systemユーザー・セキュリティ管理ガイドを参照してください。

プロセス管理のアラートの設定

プロセス・コントロール・モジュールを設定して、プロセス・コントロールでのステータスの変更に基づいて電子メール・アラートを生成できます。アラートは、「第1パス」、「確認レベル1」から「確認レベル10」、「提出済」、「承認済」、または「発行済」のアクションに対して設定できます。

プロセス単位が「開始していません」レベルの場合、またはサインオフ・アクションに対しては、電子メール・アラートは生成されません。

▶ プロセス管理の電子メール・アラートを設定するには:

1. メタデータ・マネージャから、プロセス単位のシナリオで、SupportsProcessManagementメタデータ属性をAに設定してアラートを許可します。



注:

この属性を使用可能にすると、確認時に、電子メール・アラートを受信するセキュリティ権限を持っているユーザーに向けて電子メール・アラートが生成されます。

2. プロセス・コントロールの電子メール・アラートを受信セキュリティ役割にユーザーを割り当てます。
3. プロセス単位のシナリオとエンティティに割り当てられているセキュリティ・クラスに対する「すべて」または「レベルを上げる」アクセス権をユーザーに割り当てて、各セキュリティ・クラスのアラートを追加します。
4. セキュリティ・クラスのセキュリティ・アクセス・セクションで、アラートを有効にするには、アラートのサポート・オプションをYesのYに設定します。例: [Default];User1@NativeDirectory;All;Y。

すべての条件を満たすユーザーが電子メール・アラートを受信します。

表117 プロセス管理のユーザーの役割とアラート通知

アクションの前後のプロセス単位レベル	通知を受けるプロセス管理のユーザーの役割
第1パス	エンティティに対する「すべて」または「レベルを上げる」アクセス権を持つユーザーに通知されます。
確認レベル1-10	該当する確認レベルの「確認者」および「提出者」の役割のユーザーに通知されます。 たとえば、確認レベル1の場合、「確認者1」と「提出者」の役割のユーザーに通知されます。
提出済	「確認監督者」の役割のユーザーに通知されます。この役割のユーザーのみが提出されたプロセス単位を承認できます。
承認済	「確認者1」から「確認者10」および「提出者」の役割のユーザーに通知されます。
発行済	エンティティに対する「すべて」、「読取り」または「レベルを上げる」アクセス権を持つユーザーに通知されます。

プロセスのレビュー・アクションが実行されると、設定されたセキュリティ権限に従って、該当するユーザーの電子メール・アラートが自動的に生成されます。アクションを実行したユーザーには確認の電子メールも通知されます。



注:

「アプリケーション管理者」の役割を持つユーザーは、電子メール・アラートを受信しません。アプリケーション管理者の役割を持つユーザーが電子メール・アラートを受信するには、管理者を個別ユーザーとして設定し、アラートを受信するように役割を割り当てます。

内部取引のアラートの設定

電子メール・アラートを受信するセキュリティ権限を割り当てられたユーザーに内部取引の電子メール・アラートを生成できます。

セキュリティ役割の設定については、*Oracle Enterprise Performance Management Systemユーザー・セキュリティ管理ガイド*を参照してください。

▶ 内部取引の電子メール・アラートを設定するには:

1. 内部取引の電子メール・アラートを受信セキュリティ役割にユーザーを割り当てます。
2. セキュリティ・クラスのセキュリティ・アクセス・セクションで、アラートを有効にするには、アラートのサポート・オプションをYesのYに設定します。例: [Default];User1@NativeDirectory;All;Y。

すべての条件を満たすユーザーが内部取引モジュールまたは内部取引パートナー照合レポート・モジュールから電子メール・アラートを受信します。

内部取引の電子メール・アラートの生成については、『*Oracle Hyperion Financial Managementユーザー・ガイド*』を参照してください。



構成設定

この項の内容:

使用可能な構成設定	466
構成設定の変更	470
値の上書き	471
設定表の表示の変更	471
設定の検索	471
有効な設定の表示	472
設定のエクスポート	472
設定の削除	472

設定モジュールでは、Financial Managementの構成設定を表示および変更できます。デフォルトで、インストール時に、設定表には推奨される設定が移入されます。

必要に応じて、設定値を変更できます。たとえば、メモリーの使用状況に基づいて値を変更したり、パフォーマンスを改善する必要があります。このような値を更新する上での考慮事項は、[473ページの付録B「パフォーマンスの最適化」](#)を参照してください。

特定のアプリケーションの設定を上書きできます。システム・レベルの値を変更するには、Financial Management管理者のセキュリティ役割が必要です。アプリケーション・レベルの値を変更するには、Financial Management管理者とアプリケーション管理者の両方のセキュリティ役割が必要です。

設定リストをソートして設定を検索し、Excelワークシートにエクスポートできます。設定の上書きを削除することはできませんが、システムによって作成された設定は削除できません。

特定アプリケーションの設定のランタイム値を表示することもできます。[472ページの有効な設定の表示](#)を参照してください。

以下のトピックを参照してください。

- [466ページの使用可能な構成設定](#)
- [470ページの構成設定の変更](#)
- [471ページの値の上書き](#)
- [471ページの設定表の表示の変更](#)
- [471ページの設定の検索](#)
- [472ページの有効な設定の表示](#)
- [472ページの設定のエクスポート](#)
- [472ページの設定の削除](#)

使用可能な構成設定

表118 構成設定

設定	説明	値
AllowOverlappingConsolidationOverwrite	Overwriteが重複する連結プロセスを実行できるかどうかを指定します。	0 - ユーザーは重複する連結を実行できません。現在実行中で重複するものがある場合、新しい連結プロセスを開始できません。 1 - ユーザーは重複する連結を実行できます。別の連結プロセスが現在実行中で重複する場合、新しい連結を開始するかどうかを尋ねる警告メッセージが表示されます。 デフォルト値: 1
AutoClearDeadTasks	完了したタスクをシステムから自動的にクリアするかどうか	0 - 完了したタスクをシステムに残します 1 - 完了したタスクをシステムからクリアします デフォルト値: 0
AutoClearEAFlatfileTasksAfterSeconds	タスクの実行からデータの抽出タスクを自動的にクリアし、アプリケーション・サーバーからファイルを削除するまでシステムが待機する秒数	範囲: 600秒(10分) - 864,000秒(10日) デフォルト値: 86,400秒
ConsolidationMultiThreadingScheme	連結プロセスの実行方法に関連する複数の設定に使用します。各設定は、他の設定に影響を与えることなく有効化または無効化できます。 複数のオプションを選択し、選択したオプションの合計にパラメータを設定できます。たとえば、オプション2および4を使用するには、値を6に設定します。値0は、3つのオプションのいずれも有効になっていない制限されたマルチスレッドでの従来の連結動作を表します。この設定は、サーバーまたはアプリケーション・レベル(あるいはその両方)で上書きできます。	有効値: 1 - 数値が親エンティティに連結されるまでは、連結プロセスですべての子エンティティを[Parent]の値に達するまで計算および換算することが可能です。 2 - マルチスレッドの親レベル・エンティティへの連結プロセスが可能です。これにより、大規模な連結においてパフォーマンスが大幅に向上します。 4 - 全ベースレベルのエンティティの初期計算をスキップします。これらのエンティティは、親エンティティを処理するときに必要な応じて計算されます。 デフォルト値: 2
DataSize	表コンポーネントのフェッチ・サイズ	範囲: 25 - 2,000

設定	説明	値
		デフォルト値: 500
DefaultAdminPage	連結管理のデフォルトの開始ページ	有効値: 次のページのいずれか: システム・メッセージ、アプリケーション、システム設定、システム・ユーザー、システム管理、タスクの監査、データの監査、タスク・フローの管理、タスクフロー・ステータス、DSNの構成 デフォルト値: システム・メッセージ
DefaultColFetchSize	データ・フォームおよびデータ・グリッドのデフォルトの列フェッチ・サイズ	範囲: 25 - 2,000 デフォルト値: 50
DefaultRowFetchSize	データ・フォームおよびデータ・グリッドのデフォルトの行フェッチ・サイズ	範囲: 25 - 2,000 デフォルト値: 250
EnableRulesLogging	ルールのロギングを有効にするかどうか。ルールの問題を特定するためのデバッグの目的で使用します。	0 - ロギングを無効にします 1 - ロギングを有効にします デフォルト値: 0
EnableRunningTasksMaskUserName	管理者以外のユーザーが表示する場合に、「タスクの実行」でユーザー名をマスクするかどうかを指定します。	0 - ユーザー名のマスキングを無効にします 1 - ユーザー名のマスキングを有効にします デフォルト値: 0
HsxDSStartupOption	JHsxServerの起動時にアプリケーションをデフォルトで起動するかどうかはAutoStartによって決定されます。 AutoStartが無効になっている場合、最初のアプリケーションがオープンされたときにアプリケーションは自動的に起動されません。これらは、アクティブなユーザー・セッションが存在しない場合、定期的に再利用されます。 AutoStartが有効になっている場合、JHsxServerの起動時にアプリケーションは起動されます。アプリケーションは定期的にはリサイクルされません。 この設定は、グローバルに設定するか、アプリケーション・サーバー・レベルで上書きできます。	0 - AutoStartは無効です(自動シャットダウンは有効です)。 1 - AutoStartは有効です(自動シャットダウンは無効です)。 デフォルト値: 0

設定	説明	値
IcmSystemReportTranslationMode	<p>通貨換算を必要とする内部取引照合システム・レポートの実行時、デフォルトでは換算された値がデータベースに書き込まれます。これにより、レポートの実行時間が長くなり、データベース・サイズが大幅に増加する可能性があります。この設定を使用すると、換算された値の処理方法を制御できます。</p> <p>換算モード = 0 (サブ換算を実行し、データを保管します)</p> <p>換算モード = 1 (サブ換算を実行し、データを保管しません)</p> <p>換算モード = 2 (デフォルト換算を使用してオンザフライ換算を実行し、サブ換算を実行せず、データを保管しません)</p>	<p>0 - 既存の換算ロジックを使用します。ルールでサブ換算を使用し、データベースへの変更をコミットします。</p> <p>1 - オンザフライ、ルールを使用。ルールでサブ換算を使用しますが、換算された値はデータベースに記録されません。注意: CNステータスの親エンティティも換算されます。</p> <p>2 - オンザフライ、デフォルト換算を使用します。デフォルト換算のみを実行します(サブ換算は実行しません)。変更はデータベースに書き込まれません。注意: CNステータスの親エンティティは換算されません。</p> <p>デフォルト値: 0</p>
MaxDataCacheSizeInMB	セルを格納するためにアプリケーション・サーバーによって割り当てられる最大メモリー(MB)	<p>範囲: 500 - 15,000</p> <p>デフォルト値: 4,500</p>
MaxFileSelectionOnLoad	複数ファイル選択コントロールで選択できる最大ファイル数	<p>値は1以上にする必要があります。</p> <p>デフォルト値: 10</p>
MaxNumConcurrentConsolidation	各サーバーのアプリケーションごとに可能な同時連結の最大数。設定値を超えて実行される連結は、スケジューラされた連結としてキューに入ります。この設定は、グローバルに設定するか、アプリケーションのサーバー・レベルで上書きできます。	<p>範囲: 範囲にハード・リミットはありません。値は正数である必要があります。</p> <p>デフォルト値: 8</p>
MaxNumCubesInRAM	FreeLRUアルゴリズムのサブキューブの最大数	<p>範囲: 100以上の値にする必要があります。</p> <p>デフォルト値: 30,000</p>
MaxNumDataRecordsInRAM	システムがRAMに格納するデータ・レコードの最大数	<p>範囲: 範囲にハード・リミットはありません。値は正数である必要があります。</p> <p>デフォルト値: 30,000,000</p>
MaxNumRetriesOfBaseLevelCalculation	連結プロセスがベースレベル・エンティティに対する計算を再試行する最大回数。この設定では、計算の各繰返し後に計算ステータスが「OK」ではない場合、指定した繰返し数になるまで1	<p>範囲: 1 - 4</p> <p>デフォルト値: 0</p>

設定	説明	値
	<p>つの連結プロセス内でベース・エンティティを複数回再計算することが可能です。これは、あるエンティティで実行されているステータスへの影響ルールが、すでに計算済の別のエンティティに影響する状況を解決します。</p> <p>0以上の任意の値が有効です。</p>	
MinDataCacheSizeInMB	<p>最小データ・キャッシュ・サイズ(MB)。この値を高い数値に設定することによって、データ・キャッシュ増加の試行数を減らすことができるため、メモリの断片化を低減できます。通常、データ・キャッシュは必要に応じて増加され、一度に最大25MB増加します。</p>	<p>範囲: 100 - 5,000</p> <p>デフォルト値: 2,250</p>
NumConsolidationThreads	<p>連結ごとに可能な最大連結スレッド数。値を減らすと、システムによるリソースの使用が制限されるため、連結のパフォーマンスが低下します。</p>	<p>範囲: 1 - 24</p> <p>デフォルト値: 8</p>
NumCubesLoadedBeforeCheckingFreeLRU	<p>FreeLRUアルゴリズムをトリガーするために、RAMにロードする必要があるキューブの数。2つの条件のいずれかを満たした (NumMinutesBeforeCheckingLRUを満たしたか、NumCubesLoadedBeforeCheckingLRUに達した)場合に、FreeLRUがトリガーされます。</p>	<p>範囲: 範囲にハード・リミットはありません。値は正数である必要があります。</p> <p>デフォルト値: 100</p>
NumDataLoadsAllowed	<p>各サーバーのアプリケーションごとに可能な最大同時データ・ロード・タスク数。この設定は、グローバルに設定するか、アプリケーションのサーバー・レベルで上書きできます。</p>	<p>範囲: 範囲にハード・リミットはありません。値は正数である必要があります。</p> <p>デフォルト値: 8</p>
NumEAExportsAllowed	<p>各サーバーのアプリケーションごとに可能な「データの抽出」タスクの最大同時数。この設定は、グローバルに設定するか、アプリケーションのサーバー・レベルで上書きできます。</p>	<p>範囲: 範囲にハード・リミットはありません。値は正数である必要があります。</p> <p>デフォルト値: 8</p>
NumEAThreads	<p>データ抽出ごとのスレッド数</p>	<p>範囲: 範囲にハード・リミットはありません。値は正数である必要があります。</p> <p>デフォルト値: 8</p>
NumMinutesBeforeCheckingLRU	<p>FreeLRUアルゴリズムの制限のチェック間隔(分単位)</p>	<p>1以上の値にする必要があります。</p>

設定	説明	値
		デフォルト値: 15
NumThreadsToUseWhenUpdatingGlobalSystemRanges	グローバルシステムレンジを更新する際に使用するスレッドの数	範囲: 1 - 8 デフォルト値: 8
OverrideUserFetchSizeWhenOpeningDataFormorDataGrid	データ・フォームおよびデータ・グリッドのすべてのユーザーについてフェッチ・サイズを上書きするかどうかを指定します。	0 - すべてのユーザーのフェッチ・サイズを上書きしません 1 - すべてのユーザーのフェッチ・サイズを上書きします デフォルト値: 1
WebformDoCalculateOnSubmit	ユーザーがデータ・フォームで「データの送信」をクリックする際、データの送信後に自動的に計算するかどうか。	0 - 自動的に計算しません 1 - 自動的に計算します デフォルト値: 0

構成設定の変更

設定を変更するには、管理者のセキュリティ権限を持っている必要があります。システム所有の設定の場合、値以外の列は変更できません。

「ノート」列を使用してコメントを入力できます(たとえば、値の変更理由など)。行の「ノート」アイコンの上にマウスを置くと、設定の既存のノートを表示できます。

設定が変更されると、「設定」表が、変更したユーザー名および変更日時を表示するように更新されます。

青で表示される設定は、ユーザーが変更可能な設定がデフォルト値と一致しておらず、値が更新されたことを示します。

▶ 構成設定を変更するには:

1. 「ナビゲート」、「管理」、「連結管理」の順に選択します。
2. 「管理タスク」から、「システム」を展開し、「設定」をクリックします。
3. 「設定」表から、設定の値を変更します。



ヒント:

元の値にリセットするには、「リセット」をクリックして、確認プロンプトから「はい」をクリックします。

4. オプション: コメントを入力するには、行の「ノート」アイコンをクリックしてコメントを入力し、「OK」をクリックします。
5. 「保存」をクリックします。
6. 変更を表示するには、「リフレッシュ」をクリックします。

値の上書き

サーバーおよびアプリケーションの設定のデフォルト値を上書きできます。設定がグローバルの場合、上書きすることはできません。

▶ 設定をコピーするには:

1. 「設定」表から、コピーする設定を選択します。
2. 「上書き」をクリックするか、「アクション」、「上書き」の順に選択するか、設定を右クリックして「上書き」を選択します。
3. 「上書き」ダイアログ・ボックスで、設定値を変更し、設定を適用するサーバーおよびアプリケーションを選択します。
4. オプション: コメントを入力するには、行の「ノート」アイコンをクリックしてコメントを入力し、「OK」をクリックします。
5. 「適用して閉じる」をクリックして、新しい値を適用します。

作成した新しい設定は、表内に太字で表示され、選択された状態になっています。

オプション: 別の設定を上書きするには、「適用して別に上書き」をクリックします。

6. 「保存」をクリックします。

設定表の表示の変更

任意の設定列をソートでき、また、列を並べ替えることができます。

▶ 表示する列を選択するには:

1. 「設定」リストから、「表示」、「列」の順にクリックし、表示する列を選択するか、「すべて表示」を選択します。
2. オプション: 列を表示または非表示にするには、「表示」、「列」、「列の管理」の順に選択し、矢印キーを使用して列を「非表示」または「表示可能」列リストに移動します。
3. オプション: 列のソート順を変更するには、ヘッダー・アイコンをクリックして、「昇順にソート」または「降順にソート」を選択します。

設定の検索

選択した基準に基づいて設定を検索できます。「設定の管理」タブで、設定名、設定を適用するサーバーおよびアプリケーション、設定の更新日、設定を作成または更新したユーザーを検索できます。

「有効な設定」タブでは、アプリケーションに対して、サーバーが実行時に使用した値を参照できます。設定は、名前、サーバー、アプリケーション別に検索できます。

▶ 設定を検索するには:

1. 「検索」フィールドから、検索する基準を選択するか入力します。
2. すべての基準との一致の場合は「すべて」を、選択した基準のいずれかとの一致の場合は「任意」を選択します。
3. 「検索」をクリックします。



ヒント:

「検索」設定をリセットするには、「検索」の隣の「リセット」ボタンをクリックします。

有効な設定の表示

「有効な設定」タブから、アプリケーション設定のランタイム値を表示できます。

▶ 有効な設定を表示するには:

1. 「設定の管理」から、サーバー名およびアプリケーション名を選択します。
2. 「有効な設定」をクリックするか、「アクション」、「有効な設定」の順に選択します。

有効な設定は、選択したアプリケーションおよびサーバーに基づいています。上書きされたデフォルト値は、青で表示されます。

3. メインの設定リストに戻るには、「設定の管理」をクリックするか、「アクション」、「設定の管理」の順に選択します。

設定のエクスポート

設定のリストをエクスポートして、Excelワークシートに保存できます。

▶ 設定をエクスポートするには:

1. 「設定の管理」で「エクスポート」をクリックするか、「アクション」、「エクスポート」の順に選択します。

現在の検索に基づいて行がエクスポートされます。

2. ダウンロードの指示に従い、ファイルをExcelワークシートとしてダウンロードします。

設定の削除

上書きされた設定は削除できます。システムで作成された設定は削除できません。

▶ 設定を削除するには:

1. 「設定」表から、削除する設定行を選択します。
2. 「選択した項目の削除」をクリックするか、「アクション」、「選択した項目の削除」の順に選択するか、設定を右クリックして「選択した項目の削除」を選択します。
3. 確認のプロンプトで「はい」をクリックします。

B

パフォーマンスの最適化

この項の内容:

パフォーマンスの概要	474
一般的なEPMインストール・ディレクトリの参照	474
Financial Managementのレコードおよびサブキューブ	474
Financial Managementのチューニングに関する推奨事項	475
パフォーマンスの問題の診断	475
オペレーティング・システム・パラメータのチューニング	478
Webサーバーのチューニング	478
Financial Managementアプリケーションのチューニング	482
Financial Management用のOracle 11gデータベースのチューニング	490
よくある質問	502

この章では、Financial Managementアプリケーション、データベース管理および一般的なオペレーティング・システムの概念をよく理解していることを前提としています。実際の実装および環境は、ビジネス要件、Financial Managementのデータセット、ネットワーク・トポロジおよびハードウェアの使用状況に基づいて大きく異なります。したがって、このようなガイドラインを自身の実装にどのように適合させるかを検討する必要があります。テスト結果とパフォーマンスの数値はすべて、チューニングの概念を示すための例としてのみ使用されます。



注意

不適切な設定と構成は、Financial Managementの動作を妨げる可能性があります。

チューニング設定を実装する前に、エンドツーエンド・パフォーマンス・テストを実行してデフォルトの構成に対するベースライン・パフォーマンス・データを取得し、チューニング設定を増分変更してからパフォーマンス・データを収集することをお勧めします。構成の変更によってシステム・パフォーマンスが向上し、低下しないことを確認する必要があります。

サポートされているプラットフォームおよびコンポーネントは、Oracle Enterprise Performance Management Systemサポート対象プラットフォームのマトリックス(<http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html>)を参照してください。

パフォーマンスの概要

Oracle Hyperion EPM System Performanceの概要

Oracle Hyperion EPM Systemのパフォーマンスを最大化するには、すべてのコンポーネントをモニター、分析およびチューニングする必要があります。この項では、パフォーマンスをモニターするために使用できるツールと、Financial Managementのパフォーマンスを最適化するための技法について説明します。

一般的なEPMインストール・ディレクトリの参照

この章では次のディレクトリ参照を使用します。

<terms>

<term>

<term_name>ミドルウェア・ホーム</term_name>

<definition>

<brief> ミドルウェア・ホームは、Oracle WebLogic Serverホーム、および必要に応じて1つ以上のOracleホーム (EPM Oracleホームを含む)で構成されています。デフォルトのインストール・ディレクトリはOracle¥Middlewareです。この章全体を通して、ミドルウェア・ホームの場所をMIDDLEWARE_HOMEと呼びます。</brief>

</definition>

</term>

<term>

<term_name>EPM Oracleホーム</term_name>

<definition>

<brief> Oracleホームには、特定の製品をホストするのに必要なインストール・ファイルが含まれます。Oracleホームは、ミドルウェア・ホームのディレクトリ構造内にあります。デフォルトのEPM Oracleホームの場所はMIDDLEWARE_HOME¥EPMSystem11R1です。この章全体を通して、EPM Oracleホームの場所をEPM_ORACLE_HOMEと呼びます。</brief>

</definition>

</term>

<term>

<term_name>EPM Oracleインスタンス</term_name>

<definition>

<brief> さらに、構成時に、一部の製品は構成中に定義されたEPM Oracleインスタンスにコンポーネントをデプロイします。EPM Oracleインスタンスのデフォルトの場所はMIDDLEWARE_HOME¥user_projects¥epmsystem1です。この章全体を通して、EPM Oracleインスタンスの場所をEPM_ORACLE_INSTANCEと呼びます。</brief>

</definition>

</term>

</terms>

Financial Managementのレコードおよびサブキューブ

Financial Managementでは、レコードがディメンション・メンバーの特定の交差についてすべての基本期間のデータを保持します。サブキューブがすべて同じエンティティ、シナリオ、年および値(通貨)に属するレコードの集合です。サブキューブ内には少なくとも4つのディメンション、勘定科目、ICPおよびカスタム(複数)があります。FMアプリケーションには、2つ以上のカスタム・ディメンションが必要です。カスタム・ディメンション数に上限はありません。サブ

キューブは連結を目的とした自然単位のデータであるため、Financial Managementの様々な場所で、データの移動および処理がサブキューブ・ベースで実行されます。

Financial Managementのチューニングに関する推奨事項

Financial Managementのパフォーマンス・チューニングは反復プロセスです。



注:

チューニングは特定の本番ワークロードに対して実行する必要があります。ワークロードがOracle Application Testing Suite (OATS)などのロード生成ツールによって生成された場合にチューニングを実行できます。

この項では、パフォーマンス・チューニングをすぐに開始できるように次のような領域をいくつか説明します。

- オペレーティング・システム・パラメータのチューニング
- HTTP Serverパラメータのチューニング
- Oracle Databaseパラメータのチューニング



注:

このリストはパフォーマンス・チューニングを開始するのに便利な手段ですが、チューニング対象領域の包括的なリストであるという意味ではありません。チューニングによってパフォーマンスが向上する可能性がある場所を把握するには、実装内で特定のパフォーマンスの問題をモニターおよび追跡する必要があります。

パフォーマンスの問題の診断

パフォーマンスの問題が発生した場合、修正処理を実行する前にまず原因を特定することが重要です。問題について詳しい分析が実行されるまでパフォーマンス関連のパラメータ設定を変更したり、他のアクションを実行することはお勧めしません。

モニタリング・ツールの使用

診断プロセスの一環としてパフォーマンス・データを収集するには、モニタリング・ツールを使用することをお勧めします。Financial Managementアプリケーション・サーバー、Webサーバー、データベース・サーバーおよびネットワーク・レイヤーのモニタリングにより、有用なパフォーマンス・データが得られます。

Linux環境では、OS Watcher Black Boxなどのツールを使用してオペレーティング・システムおよびFinancial Managementプロセスをモニターできます。

Windows環境では、Microsoft Performance Monitoringなどのツールを使用してFinancial Managementアプリケーション・プロセスのパフォーマンスをモニタリングできます。Financial Managementアプリケーション固有のカウンタを収集するようにパフォーマンス・モニターを構成する手順は、次のURLを参照してください:

<https://mosemp.us.oracle.com/epmos/faces/DocumentDisplay?id=953372.1>.

モニターするパフォーマンス・カウンタは次のとおりです:

サブシステム	カウンタ	ガイダンス
メモリー	Memory: Free System Page Table Entries	"警告": 空きシステム・ページ表エントリ数が8,000未満の場合。 "クリティカル": 空きシステム・ページ表エントリ数が5,000未満の場合。
メモリー	Memory: Available Mbytes	インストールされている物理メモリーの20%から25%を下回る必要があります。このような場合、ページング・アクティビティを注意深くモニターします。
メモリー	Memory: Page Reads/sec	このカウンタは常に1,000未満である必要があります。
プロセッサ	Processor: % Processor Time: _Total	合計プロセッサ稼働率は約70%から80%を下回る必要があります。
プロセッサ	Processor: % Processor Time: (N)	各プロセッサ・インスタンスは約70%から80%の稼働率を下回る必要があります。
ディスク	PhysicalDisk: Avg. Disk sec/Transfer	約25ミリ秒を下回る必要があります。 一般ルール: Avg. Disk Seconds/Transfer(ディスク待機時間カウンタ)が25ミリ秒を大幅に上回る場合、ディスク・サブシステムは異常かつボトルネックです。このカウンタは問題の解決方法を示しません。問題があることのみを示します。
ディスク	PhysicalDisk: Average Disk Queue Length	平均値はディスクのスピンドル数を下回る必要があります。SANを使用している場合、このカウンタを無視して待機時間カウンタのPhysicalDisk¥Average Disk sec/ReadおよびPhysicalDisk¥Average Disk sec/Writeに的を絞ります。
ディスク	PhysicalDisk: Average Disk sec/Read	平均値は20ミリ秒を下回る必要があります。 スパイク(最大値)は50ミリ秒以下にしてください。
ディスク	PhysicalDisk: Average Disk sec/Write	平均値は20ミリ秒を下回る必要があります。 スパイク(最大値)は50ミリ秒以下にしてください。
ネットワーク	Network Interface: Bytes Total/sec	100Mbps/秒のネットワーク・インタフェース・カード(NIC)の場合、6-7MB/秒を下回る必要があります。 1000Mbps/秒のNICの場合、60-70MB/秒を下回る必要があります。
ネットワーク	Network Interface: Packets Outbound Errors	常にゼロ(0)である必要があります。

プロセス	カウンタ/ガイダンス
Oracle Hyperion Financial Management: インスタンス: 1. JHsxServer 2. XFMDDataSource	<p>Process: % Processor Time - プロセスのプロセッサ稼働率は90%を下回る必要があります。</p> <p>Process: Private Bytes - 特定のプロセス専用に割り当てられたバイト数をレポートします。値はリークしているプロセスに対して上昇する傾向があります。</p> <p>Process: Working Set - プロセスに割り当てられた共有バイト数およびプライベート・バイト数をレポートします。値はリークしているプロセスに対して上昇する傾向があります。</p> <p>Process: Page Faults/sec - プロセスに起因するフォルト(ハードおよびソフトのフォルト)の合計数をレポートします。値はリークしているプロセスに対して上昇する傾向があります。</p> <p>Process: Page File Bytes - ページング・ファイルのサイズをレポートします。値はメモリー・リーク時に上昇する傾向があります。</p> <p>Process: Handle Count - オブジェクト用にオープンされたアプリケーションで作成されるハンドルの数をレポートします。ハンドルは、アクセスする必要があるリソースの識別にプログラムで使用されます。このカウンタの値はメモリー・リーク時に上昇する傾向があります。</p> <p>Process: Virtual Bytes - プロセスで使用されている仮想アドレス空間の現在のサイズ(バイト単位)。perfmonプロセスの仮想バイト数が急速に増えてメモリーを決して解放しないと、アプリケーションのメモリー・リークを示します。</p> <p>Process: Virtual Bytes Peak - プロセスが常時使用している仮想アドレス空間の最大サイズ(バイト単位)。perfmonプロセスの仮想バイト数が急速に増えてメモリーを決して解放しないと、アプリケーションのメモリー・リークを示します。</p> <p>Process: Pool Nonpaged Bytes - ページ化プールのサイズ(バイト単位)。システム・メモリー(物理メモリー)の領域。</p>

リモート診断エージェント(RDA)の使用

リモート診断エージェント(RDA)とは、Perlプログラミング言語で記述されたエンジンによって実行される一連のコマンドライン診断スクリプトです。収集されたデータからは、問題の診断に役立つその環境の全体像がわかります。

RDAを実行すると、特にFinancial Managementアプリケーションでサブキューブのサイズを決定するのに有用です。RDAはMy Oracle SupportのWebサイトから入手できます。スタート・ガイドは、次のナレッジ・ベースの記事を参照してください:

<https://mosemp.us.oracle.com/epmos/faces/DocumentDisplay?id=1100612.1>

参照アプリケーションの使用

参照アプリケーションとは、パフォーマンスの問題の診断に使用するアプリケーションです。参照アプリケーションでは、環境が適切にチューニングされたかどうかを判定するのに有用な内部タイミングとの比較ができる一連

のタスクが実行されます。一部のタイプの参照アプリケーションを実行すると、問題の特定に役立ちます。しかし、Financial Managementのすべてのパフォーマンス特性を明らかにできるアプリケーションはありません。あるアプリケーションでは良好なパフォーマンスを示しても、別のアプリケーションでは低いパフォーマンスを示すことがとてもよくあります。組み込まれたパラメータには、データ量、サブキューブごとのレコード分布、エンティティの構造、通貨の数などがあります。

オペレーティング・システム・パラメータのチューニング

Windowsパラメータのチューニング

Windowsプラットフォームでは通常、デフォルトのTCP/IP設定で十分です。ただし、高負荷の下では、*MaxUserPort*および*TcpTimedWaitDelay*の調整が必要になる場合があります。これらのパラメータは、アプリケーションが要求するユーザー・ポートの可用性を決定します。

パラメータ	デフォルト値	推奨値
<p>TcpTimedWaitDelay</p> <p>このパラメータは、アプリケーションがTCP接続をクローズした後にポートを再利用するためにOSが待機する時間を制御します。デフォルト値は4分です。ユーザー負荷が高いと、このような制限を超えて、アドレス使用中の接続例外が発生することがあります。</p> <p>このパラメータはWindowsレジストリの次の場所で設定します: HKLM\System\CurrentControlSet\Services\Tcpip\Parameters</p> <p>値の名前: TcpTimedWaitDelay</p> <p>値の種類: DWORD</p> <p>データ: 30 (10進数)</p>	240	30
<p>値の名前: MaxUserPort</p> <p>値の種類: DWORD</p> <p>データ: 65534 (10進数)</p>	5000	65534

Webサーバーのチューニング

Financial Management Webサーバーと他のEPMコンポーネントの主なチューニング手順は、最初のEPM System製品とともにコンピュータにインストールされ、コンピュータにインストールされるすべての製品の構成に使用されるEPM Systemコンフィグレータに組み込まれています。詳細は、『Oracle Enterprise Performance Management Systemインストールおよび構成ガイド』を参照してください。

HFM Webのチューニング

HFM Webのタイムアウト・パラメータ

EPM構成ファイル	デフォルト値	推奨値
Webサーバー・プラグインとWebLogic Oracle HTTPサーバーまたはIISを使用してリクエストをWebLogicにプロキシする場合、次の設定が適用されます。使用されるWebサーバー・タイプの選択肢はEPM Systemコンフィグレータから設定されます。パラメータは大文字と小文字が区別され、手動で追加する必要があります。		
WLIOTimeoutSecs プロキシがIISの応答を待機する時間を設定します。	3,600	3,600
WLSocketTimeoutSecs 接続時のソケットのタイムアウトを秒単位で設定します。	2	2
(OHS Webサーバー・プラグイン) EPM_ORACLE_INSTANCE¥httpConfig¥ohs¥config¥OHS¥ohs_component¥mod_wl_ohs.conf hfmadfインスタンスに適用されます: 例: <LocationMatch ^/hfmadf> SetHandler weblogic-handler WeblogicCluster<WLCluster>:<port> WLIOTimeoutSecs 3600 Idempotent OFF WLSocketTimeoutSecs 2 </LocationMatch>		
(IIS Webサーバー・プラグイン) EPM_ORACLE_INSTANCE¥httpConfig¥VirtualHosts¥hfmadf¥iisproxy.ini WIForwardPath=/hfmadf PathTrim=/ WebLogicHost=<host> WebLogicPort=<port> KeepAliveEnabled=true		

EPM構成ファイル	デフォルト値	推奨値
KeepAliveSecs=20 WLIOTimeoutSecs=3600 Idempotent=OFF WLSocketTimeoutSecs=750		
WebLogic スタック・スレッド(WebLogic管理コンソール) HFMアプリケーション・プロセスがロード・ファイルの処理でビジー状態の場合、メタデータのロードなどの長時間実行タスクがスタック状態のように見えることがあります。このような問題が発生した場合は、スタック・スレッド時間の値を増やします。 WebLogicコンソールから次の設定を変更します。 「ロックして編集」を選択します。 「サーバー」を選択して「HFMWeb(N)」をクリックします。 「チューニング」タブを選択します。 「スタック・スレッド最大時間」の値を1200に変更します。 「スタック・スレッド・タイマー間隔」の値を1200に変更します。 「変更のアクティブ化」を選択します。 HFMWeb0サーバーを再起動する必要があります。 HFMWeb(N)はHFMWeb0、HFMWeb1などで、HFM Webサーバーがデプロイされている数と更新しているサーバーによって決まります。		
スタック・スレッド最大時間	600	1200
スタック・スレッド・タイマー間隔	60	1200

HFM Webのチューニング・パラメータ

この項では、その他のHFM Web UIのチューニング・パラメータについて説明します。

チューニング・パラメータと場所

パラメータ	デフォルト値	推奨値
WebLogic		

パラメータ	デフォルト値	推奨値
<p>HFM ADF WebアプリケーションのJavaヒープ・サイズ(XmsおよびXmx)</p> <p>HFM ADF Webアプリケーションの各管理対象インスタンスで使用可能なヒープ量はJVMOptionXXオプションを編集することでチューニングできます。</p> <p>HFMWeb(N)はHFMWeb0、HFMWeb1などで、HFM Webサーバーがデプロイされている数と更新しているサーバー・インスタンスによって決まります。</p>		
<p>Xmsは初期ヒープ・サイズを設定し、Xmxと同じサイズに設定する必要があります。</p>	128m	2048m
<p>Xmxはヒープの最大サイズを設定します。最大ヒープ・サイズを使用可能な物理メモリーの75%より高い値に設定しないことをお勧めします。</p> <p>コンパクト・デプロイメントでは、すべてのEPM製品に対してEPMSystem(N) Webアプリケーションが1つしかないため、HFM ADF Webを個別にチューニングすることはできません。</p>	512m	2048m

Webブラウザの最適化

Internet Explorer

チューニング・パラメータと場所

パラメータ	デフォルト値	推奨値
<p>アドオンを無効にしてInternet Explorerブラウザを実行することをお勧めします。アドオンにより、サーバーとの通信で待機時間が発生する可能性があり、レスポンス時間の低下やユーザー・インタフェースでのエラーの原因となります。</p>		
<p>IEの「イントラネット サイトを互換表示で表示する」</p> <p>すべてのエンド・ユーザーのマシンでIEを構成します。</p> <p>「ツール」メニューから、「互換表示設定」を選択します。</p>		オフ(選択解除)

Firefox

チューニング・パラメータと場所のリスト

パラメータ	デフォルト値	推奨値
使用しているOracle Enterprise Performance Management Systemのバージョンでサポートされている最新バージョンのFirefoxを使用することをお勧めします。サポート対象プラットフォームのマトリックス(Oracle EPM Supported Platform Matrix)を参照してください。		
Firefox用のRemote XULアドオンをインストールします。		

Financial Managementアプリケーションのチューニング

一般に、単一のアプリケーション・サーバー上で一度に複数のFinancial Managementアプリケーションを実行すると、すべてのアプリケーションのパフォーマンスに影響を及ぼします。アイドル状態のアプリケーションの稼働にはデータベース接続およびCPU時間が必要であるため、他のアプリケーションがアイドル状態であっても、一度に3つまたは4つを超えるアプリケーションを実行しようとししないでください。

一般的にチューニングされたFinancial Managementの設定

構成設定の詳細なリストは、このガイドの付録Aを参照してください。465ページの付録A「構成設定」を参照してください。

MaxNumDataRecordsinRam

一般に、RAMで保持するレコード数を決定するため、MaxNumDataRecordsInRAMが最も重要な設定です。2つの条件NumMinutesBeforeCheckingLRUまたはNumCubesLoadedBeforeCheckingLRUのいずれかに達した場合に、メモリー内のレコード数がチェックされます。466ページの使用可能な構成設定を参照してください。

範囲: 範囲にハード・リミットはありません。

デフォルト値: 30,000,000



注:

RAM内の合計レコード数がこの値を上回るとFreeLRUがコールされ、Financial Managementサーバーのメモリーを一部解放するために、メモリーからレコードが解放されます。情報メッセージ"FreeLRUCachesIfMoreRAMIsNeededによりデータ・キューブが解放されました"がログに記録されます。

MinDataCacheSizeInMB

この値を高い数値に設定することによって、データ・キャッシュ増加の試行回数を減らすことができるため、メモリーの断片化が削減されます。通常、データ・キャッシュは必要に応じて増加され、一度に最大25MB増加します。

範囲: 範囲にハード・リミットはありません。

デフォルト値: 2,000

MaxDataCacheSizeInMB

この設定は、セル値およびセル・ステータスを格納するためにFinancial Managementアプリケーション・サーバーによって割り当てられるメモリーの最大量を制御します。システムでさらにメモリーが必要な場合、セル値およびセル・ステータスはLRUロジックに基づいてディスクにページ・アウトされます。回避策として、キャッシュ・サイズを増します。

範囲: 範囲にハード・リミットはありません。

デフォルト値: 4,500



注:

ページングが原因で連結操作のパフォーマンスが低下する場合は、MaxDataCacheSizeInMBを増やしてページングを最小限に抑えることができます。この値は"MaxNumDataRecordsInRAM"によって許可される合計メモリー使用量よりも大きい値にする必要があるため、セルが不必要にディスクにページ・アウトされません。MaxDataCacheSizeInMB設定が小さすぎると、データ・レコードを格納するためのメモリーが不足してページングが開始されるため、システム・パフォーマンスが低下します。

MaxNumCubesInRAM

この設定は任意の時点でメモリーに保持されるキューブの数を制御します。2つの条件NumMinutesBeforeCheckingLRUまたはNumCubesLoadedBeforeCheckingLRUのいずれかに達した場合に、メモリー内のキューブ数がチェックされます。この設定を小さくすると、疎アプリケーション(多くのエンティティを含むが、エンティティごとのレコードが相対的に少ないアプリケーション)の高いメモリー使用量を軽減できます。この設定は、連結やデータ取得など、すべてのデータ操作に影響を及ぼします。

範囲: 100-500,000

デフォルト値: 60,000

IcmSystemReportTranslationMode

通貨換算を必要とする内部取引照合システム・レポートの実行時、デフォルトでは換算された値がデータベースに書き込まれます。これにより、レポートの実行時間が長くなり、データベース・サイズが大幅に増加する可能性があります。この設定を使用すると、換算された値の処理方法を制御できます。

範囲: 0、1、2

デフォルト値: 0

有効なデータ・オプション:

- 0 - デフォルトの動作です。換算された値がデータベースに書き込まれます。
- 1 - オンザフライ計算ではルールでサブ換算を使用しますが、換算された値はデータベースに書き込まれません。注意: CNステータスの親エンティティも換算されます。
- 2 - オンザフライ計算ではデフォルト換算のみを実行します(サブ換算ルールなし)。変更はデータベースに書き込まれません。注意: CNステータスの親エンティティは換算されません。

NumConsolidationThreads

この設定は、連結のマルチスレッドを、アプリケーション・サーバーごとに制御します。

範囲: 1 - 24

デフォルト値: 8



注:

値を下げると、システム・リソースの使用量が制限されて、連結のパフォーマンスが下がります。この値を上げると、CPU使用率が高くなり、他のコンポーネントのパフォーマンスに影響を及ぼす可能性があります。



ヒント:

NumConsolidationThreadsを4から8に増加した影響を評価するため、テストを複数回実行しました。次の結果は、1か月の連結時間が短くなったことを示しています。

取引	92ユーザー	92ユーザー
	NumConsolidationThreads = 4	NumConsolidationThreads = 8
	平均応答時間(秒)	平均応答時間(秒)
01_Run_Consolidation_A_1105	10.11	3.22
02_Run_Consolidation_A_0005	16.15	9.47
03_Run_Consolidation_A_2205	7.75	3.19
04_Run_Consolidation_A_3305	18.67	9.17
05_Run_Consolidation_B_1105	8.21	3.14
06_Run_Consolidation_B_0005	13.26	9.27
07_Run_Consolidation_B_2205	7.69	6.20
08_Run_Consolidation_B_3305	18.29	9.41

取引	92ユーザー NumConsolidationThreads = 4 平均応答時間(秒)	92ユーザー NumConsolidationThreads = 8 平均応答時間(秒)
09_Run_Consolidation_C_0005	30.59	22.08



注:

テスト実行時のFinancial Managementアプリケーション・サーバーのCPU使用率は、さらに厳しいワークロードへの対応に使用できる容量があつて許容範囲内でした。



注:

この値を増やす前に、すべてのアプリケーション・サーバーで必ずすべての設定を同じにします。実行中の同時連結とキューで待機中の連結を比較する場合、合計連結時間が実際に悪化する前に特定のサーバーで現在の連結がいくつ実行されるかを確認するためにテストする必要があります。

MaxNumConcurrentConsolidations

この設定は、可能な同時連結の数を、アプリケーション・サーバーごとに制御します。値を超えて実行される連結は、スケジュールされた連結としてキューに入ります。

範囲: 1 - 制限なし

デフォルト値: 8

シナリオ: 3つのFinancial Managementアプリケーション・サーバーがある場合、各サーバーでは最大8つの同時連結を実行できますが、NumConsolidationsAllowedのデフォルト値により、1つのアプリケーションに対して3つのサーバーでは実行中の同時連結が合計8つのみに制限されます。例: ユーザーがServerAで6つの連結を送信した後、ServerBでさらに2つの連結を送信します。この8つの連結はすべて実行されます。それと同時に、ユーザーがServerCで次の連結を送信した場合、これは前の8つのいずれかが終了するまで実行されません(「タスクの実行」ページで、ステータスが「開始(スケジュールによる)」になります)。



注:

この値を増やす前に、すべてのアプリケーション・サーバーで必ずすべてのレジストリ設定を同じにします。実行中の同時連結とキューで待機中の連結を比較する場合、合計連結時間が実際に悪化する前に特定のサーバーで現在の連結がいくつ実行されるかを確認するためにテストする必要があります。

NumThreadsToUseWhenUpdatingCalcStatusSystemWasChanged

メタデータのロード時に、エンティティ階層が変更(メンバーが移動、追加または削除)されると、使用中のメンバーはその子または親と整合性がなくなる可能性があります。Financial Managementでは、シナリオおよび年の各組合せ

の算出ステータスを確認する必要があります。この設定を使用すると、単一のスレッドではなく複数のスレッドで算出ステータスを確認および更新できるようになります。

範囲: 1 - 8

デフォルト値: 8

大規模アプリケーション用のFinancial Managementのメモリー設定

次の表に、使用可能な物理メモリーに応じたパラメータの推奨値を示します。これは、Financial Managementがマシンで実行されている唯一のメモリー集約型プロセスであり、Financial Managementアプリケーションを1つしか実行していないということを前提としています。



注:

複数のアプリケーションがアクティブの場合、サーバーに搭載されている物理メモリーの合計をアプリケーションの数で除算して、各アプリケーションの使用可能な物理メモリーを求めます。

使用可能な物理メモリー	MaxNumDataRecordsInRAM	MaxDataCacheSizeinMB
8 GB	10,000,000	1,500
16 GB	30,000,000	4,500
32 GB	60,000,000	9,000
64 GB	100,000,000	15,000
128 GB	200,000,000	30,000

アプリケーション固有の設定

以前は環境レベルのみの設定であった一部の設定が、アプリケーション・レベルまで拡大されています。通常、アプリケーション・レベルでのFinancial Managementのチューニングは、複数のアプリケーションを同じFinancial Managementアプリケーション・サーバー上で実行する必要があるのに、頻繁に使用されるアプリケーションが1つのみの場合に役立ちます。その場合、頻繁に使用されるアプリケーションをチューニングしてサーバーのメモリーの大部分を使用できるようにし、残りのアプリケーションではデフォルト値を使用してアプリケーションが利用できるメモリー量を制限します。Financial Managementアプリケーション固有またはサーバー固有の設定を使用するには、「構成設定」の章での説明に従って上書き手順を使用します。[471ページの値の上書き](#)を参照してください。



注:

上書きの優先順位は次のようになります:

1. アプリケーション固有設定が存在せず、インストール・レジストリ設定が存在する場合、インストール・レジストリ設定が使用されます。
2. アプリケーション固有設定が存在しない場合、サーバー・キーに定義された設定が使用されます。

3. アプリケーション固有設定もサーバー設定も定義されていない場合、デフォルト値が使用されます。設定およびデフォルト値は、[465ページの付録A「構成設定」](#)を参照してください。

次の設定はアプリケーション固有設定でオーバーライドできます:

- AllowOverlappingConsolidationOverride
- AutoClearDeadTasks
- AutoClearEAFlatfileTasksAfterSeconds
- EnableRulesLogging
- EnableRunningTasksMaskUserNames
- HsxDsStartupOption
- MaxDataCacheSizeInMB
- MaxNumConcurrentConsolidations
- MaxNumCubesInRAM
- MaxNumDataRecordsInRAM
- MaxNumRetriesOfBaseLevelCalculation
- MinDataCacheSizeInMB
- NumConsolidationThreads
- NumCubesLoadedBeforeCheckingLRU
- NumDataLoadsAllowed
- NumEAExportsAllowed
- NumEAThreads
- NumMinutesBeforeCheckingLRU
- NumThreadsToUseWhenUpdatingCalcStatusSystemWasChanged

Financial Managementアプリケーション・サーバーのチューニング

Financial Managementアプリケーション・サーバーをチューニングする場合、代表ユーザーの同時実行性で主要なユーザー・アクティビティを測定するベースライン・テストから始める必要があります。レポート作成およびOracle Smart View for Officeのユーザー・アクティビティを連結アクティビティを区別するために、よくFinancial Managementクラスタを複数使用しますが、その場合、様々なチューニングによる変更が、測定されるユーザー・タスクに基づいてサーバーごとに異なる影響を及ぼすことも考えられます。たとえば、主にレポート作成に使用されるFinancial Managementアプリケーション・サーバーでは、NumConsolidationThreadsを増やしても何の利点も見られませんが、連結を数多く実行しているサーバーでは連結時間に改善が見られます。同様に、主にレポート作成に使用されるアプリケーション・サーバーでは、すべてのレコードをメモリーに保持するのに十分高い値にMaxNumDataRecordsInRAMが設定されている場合、レポート作成を繰り返すとレスポンス時間が短くなる可能性があります。連結を数多く実行するサーバーでは、連結時間の改善は見込めません。アプリケーション・サーバーに付与する役割を決定するときに考慮する別のポイントは、CPU速度、CPUコア数、RAM容量およびRAM速度です。集約型ルールが設定されたFinancial Managementアプリケーションを1つ実行し、主に連結の実行を専門とするアプリケーション・サーバーは通常、クロック速度が遅いCPU/コアの数を増やすのではなく、CPUクロック速度を速くして少なくとも8個のコアを使用すると、最高のパフォーマンスを示します。

FMチューニングの例

CPUが8個、RAMが64GBで、3つのHFMアプリケーションをホストし、その3つのうちの1つのみ(このアプリケーションをCORPと呼びます)が頻繁に使用されるHFMアプリケーション・サーバーの単純なチューニング事例を考えてみます。Financial Managementアプリケーション・サーバーは連結とレポート作成の両方のアクティビティに使用され、データベースはOracleです。

「大規模アプリケーション用のFinancial Managementアプリケーションのメモリー設定」で示した表を参照して、MaxNumDataRecordsinRAMとMaxDataCacheSizeinMBの合計をそれぞれ60,000,000と9,000にとどめる必要があります。3つのアプリケーションのうち2つは頻繁に使用されないため、その2つのアプリケーションではデフォルト値を使用できるようにし、サーバー・リソースを最大限に利用するようにCORPアプリケーションをチューニングします。

2つのアプリケーションでデフォルト値を使用するため、MaxNumDataRecordsinRAMは30,000,000です。つまり、CORPではMaxNumDataRecordsinRAMを40,000,000に設定できます。

2つのアプリケーションでデフォルト値を使用するため、MaxDataCacheSizeinMBのデフォルト値は4,500です。つまり、CORPではMaxDataCacheSizeinMBを6,000に設定できます。

この例でCORP用のチューニングを特別に検討するその他の値は次のようになります:

MinDataCacheSizeInMB - MaxDataCacheSizeinMBの値の半分(3,000)に設定します。

CORPについてアプリケーション・レベルでチューニングし、他の2つのアプリケーションでデフォルト値を使用できるようにしているため、CORPでデフォルト値とは異なる値を使用するために上書きを作成します。設定に対する更新は次のようになります:

MaxNumDataRecordsinRAM (40,000,000)

MaxDataCacheSizeinMB (6,000)

MinDataCacheSizeinMB (4,000)

アプリケーション・データベースのメンテナンス

Financial ManagementとともにOracle/SQL Serverデータベースを使用する場合は、次のベスト・プラクティスをお勧めします:

Financial Managementの表、<appname>_DATA_AUDIT、<appname>_TASK_AUDITおよびHFM_ERRORLOGの場合、次のベスト・プラクティスを実施することをお勧めします:

四半期ごと - 監査ログの確認、アーカイブおよび削除する業務。

半年ごと - システム・メッセージのアーカイブおよび表の切捨て。



ヒント:

これらの表が推奨レコード数(500,000)を超えて大きくなった場合にアクションを実行できるようにアラートを設定します。大規模な監査表によってFinancial Managementのパフォーマンスが重大な影響を受ける可能性があります。

基本的な設計上の考慮事項

- データ監査機能がビジネス要件に含まれていない場合、データ監査を無効にすることをお勧めします。データ監査表が10GBを超えるアプリケーションでは、パフォーマンスの低下が認められています。



ヒント:

メンバーの監査を無効にするには、メタデータ・ファイルですべてのメンバーについてEnableDataAudit属性をNに変更します。

- 設計の悪いルール危険(ルールによるデータ爆発など)を回避するため、本番環境にロードする前にルールをテストする必要があります。システム・パフォーマンスを許容範囲に収めるには効率的なルールが重要です。
- Financial Managementでは、サブキューブがRAMに格納されている間にそれを介してすべての処理を実行するため、サブキューブが大きいくほどパフォーマンスへの影響が大きくなります。可能な限りサブキューブのサイズを最小化してください。たとえば、最適なシステム・パフォーマンスを確保するために、サブキューブが基本レコードの制限200,000件を超えないようにします。
- Financial Managementアプリケーションでゼロをロードしたり、ゼロを計算したりすることはお勧めしません。ゼロはデータとして保管されるため、データベース・サイズが増えてパフォーマンスに影響する場合があります。Financial Managementには、1000などの数値情報のみ格納してください。『Oracle Hyperion Financial Managementユーザーガイド』のデータ・ファイルに関する項を参照してください。
- Financial Managementが正常にシャット・ダウンされない場合、再起動時に一時ファイルが残っていることがあります。最適なパフォーマンスを確保するには、Financial Managementを起動する前に、Financial Management Serverの作業用フォルダから*.db.*のファイル名をすべて削除することをお勧めします。
- データ・グリッドまたはプロセス単位に複数のドキュメントを添付する場合、1つのセルに添付するドキュメントは3つ以下にすることをお勧めします。データベースに与えるパフォーマンスの影響を制限するために、各ドキュメントのサイズは100K未満にする必要があります。



ヒント:

アプリケーションを作成するときに、ユーザーごとにドキュメントの添付ファイルのサイズ制限や最大数を設定できます。この制限は、アプリケーション・メタデータのAppSettings属性で設定できます。

- 「すべてを連結」オプションは、適切な状況下でのみ使用します。このオプションを使用すると、NODATAのエンティティがスキップされないため、連結パフォーマンスに重大な影響を及ぼす可能性があります。



ヒント:

ロジックまたは連結が必要なエンティティのみが更新されるため、「連結(変更したデータのみ連結)」が最も効率的なオプションです。「データのあるものすべてを連結」オプションは、メタデータの変更後にシステム・ステータスを「OK SC」から「OK」に更新するのに便利です。詳細は、『Oracle Hyperion Financial Management ユーザー・ガイド』の連結オプションに関する項を参照してください。

- Financial Managementソフトウェアのほとんどのバージョンに対して、データベース表のインデックスを検査するように設計されたインデックス・ユーティリティが用意されています。このユーティリティは、データベース表のインデックスを調べて必要なインデックスと比較し、変更する必要があるかどうかを確認するように設計されています。



注:

このユーティリティは、データベースの移行やデータベースのリストア、インデックスの妥当性に問題の可能性のある他のタスクの後に使用できます。また、このユーティリティは、DBAが別々の表領域にインデックスを移行してFinancial Managementデータベースのパフォーマンスを向上させるために使用するインデックス作成スクリプトの生成に使用できます。

インデックス更新ユーティリティは3つのモードで動作します。1つ目は、アプリケーションの既存のデータベース表のインデックスを調べて存在する可能性のある問題のレポートを生成します。2つ目のオプションは、レポートと、欠落または正しくない可能性のあるインデックスを削除して再作成するSQLコマンドのファイルを生成します。3つ目のモードは、レポートを生成し、インデックスを削除して再作成するコマンドを即時実行します。

Financial Management用のOracle 11gデータベースのチューニング

はじめに

Financial Managementは、アプリケーション・データを格納するためにリレーショナル・データベースを必要とします。各Financial Managementアプリケーションには、表、インデックス、ストアド・プロシージャなどの一連のオブジェクトが含まれます。このようなオブジェクトの数とサイズは、ユーザーのデータセット、アプリケーションの設計およびレポート要件に応じて異なるため、データベースを設定するために具体的なルール・セットを指定することは困難です。この項では、Oracleデータベースへのデプロイ時に発生する2つの最も一般的な問題について説明します:

- 必要な数のデータベース接続をサポートするためのメモリーが不足しているOracle DB
- レポート作成および連結時の低パフォーマンス

これらの問題はいずれも不適切なOracleパラメータの設定および構成に端を発します。大きすぎるシステム・グローバル領域(SGA)を作成すると、ユーザーの接続およびアクティビティをサポートするための空き物理メモリーの量が

制限されます。反対に、小さすぎるSGAを作成すると、余計なディスク・アクセスが発生してパフォーマンスが低下します。

この項では、一般的なデータベースをモニタリングしてパフォーマンスを最大化するために適切な初期化設定を決定するプロセスを見ていきます。データセット、ワークロードおよびアプリケーションの設計に対する変更に対応するには、このプロセスを定期的に繰り返す必要があります。

Oracle 11gでは、データベースをモニタリングおよびチューニングするプロセスが以前のバージョンよりもずっと簡単になっています。Oracle Enterprise Manager (Data ControlとGrid Controlの両方)を使用して、データベース・パフォーマンスをモニター、診断およびチューニングすることをお勧めします。正確なインスタンスの統計を取得するには、Oracleデータベースの自動メンテナンス・タスクを有効にすることをお勧めします。

データベース起動直後にチューニングを実行しないことが非常に重要です。その時点で、バッファ・キャッシュは空であり、統計は収集されていません。常に、Financial Managementアプリケーションのアクティビティ期間後にデータベースをテストしてチューニングします。

一般的なパフォーマンスの問題

低パフォーマンスの最も一般的な原因は、失効しているか欠落している表の統計です。Financial Managementアプリケーションでは、新しい表を作成するか、既存の表をクリアしてデータセット全体をオンザフライで再ロードまたは置換できます。Oracle 10gおよび11gのデフォルトでは、統計を更新する必要がある表をチェックする夜間ジョブが実行されます。これは、場合によっては十分でないことがあります。次のアクティビティは、スキーマ統計を更新する必要があることをDBAに通知する必要がある一般的なエンド・ユーザーのアクティビティです。

- 新しいHFMアプリケーションを作成およびロード
- 既存のシナリオのデータをクリアおよび再ロード
- 新しいシナリオ/年をオープンしてデータをロードこれにはFinancial Managementのルールによって移入されたシナリオが含まれます。
- Financial Managementアプリケーションは、Financial Managementアプリケーションのコピー・ユーティリティを使用してコピーします。コピー・ユーティリティの詳細は次のURLを参照してください: <https://mosemp.us.oracle.com/epmos/faces/DocumentDisplay?id=968956.1>

シナリオ/年の最初の月のデータがロードされた後では、後続の期間にデータがロードされても表の統計に影響はありません。

Oracle 11g Databaseのチューニング・ガイドライン

Oracle初期化パラメータ

データベース・パフォーマンスを向上させるために、様々な初期化パラメータを細かくチューニングできます。この項では、OracleデータベースでFinancial Managementのパフォーマンスに大きく影響することがわかっているパラメータについて説明します。

CURSOR_SHARING

CURSOR_SHARINGは、同じカーソルを共有できるSQL文のタイプを決定します。このパラメータがEXACTに設定されている場合、同一のテキストを含む文のみが同じカーソルを共有できます。このパラメータがFORCEに設定されている場合、一部のリテラルが異なり、それ以外は同一である文がSQLカーソルを共有および再使用できます。ただし、リテラルが文の意味に影響する場合は除きます。テストにより、このパラメータを値FORCEに設定すると、連結およびレポート作成のパフォーマンスが大幅に向上する可能性があることがわかっています。これは、このパラメータをFORCEに設定すると、OracleデータベースではSQL文の解析に費やす時間が短くなり、必要なメモリーが少なくなるためです。

11.1.2.2.300より前のすべてのリリースに対する推奨設定: FORCE

11.1.2.2.300以降のリリースに対する推奨設定: EXACT

MEMORY_TARGET

MEMORY_TARGETおよびMEMORY_MAX_TARGETの2つは、Oracle Databaseリリース11gの新しいパラメータです。この2つのパラメータは、Oracleデータベースの自動メモリー管理の使用方法を決定します。システム上のメモリーを管理するために自動メモリー管理の使用をお勧めします。この2つのパラメータをゼロ以外の値に設定することで、自動メモリー管理が有効になり、システム・グローバル領域(SGA)とインスタンス・プログラム・グローバル領域(PGA)の間で必要に応じてメモリーを再分散してターゲット・メモリー・サイズにチューニングされます。結果として、次のパラメータのサイズが自動的に設定されます:

SGA_TARGET

SGA_MAX_SIZE

DB_CACHE_SIZE

SHARED_POOL_SIZE

LARGE_POOL_SIZE

JAVA_POOL_SIZE

STREAMS_POOL_SIZE

PGA_AGGREGATE_TARGET

ただし、LOG_BUFFERは自動メモリー管理の影響を受けないため、引き続き手動でサイズを設定する必要があります。LOG_BUFFERのチューニングはこの項で後述します。

MEMORY_TARGETはSGAおよびPGAの合計メモリー・サイズを指定するため、高いパフォーマンスを得るには、比較的高い値に設定する必要があります。Financial Managementの連結およびレポート作成はメモリー集約型のタスクであるため、十分なメモリーが必要です。32ビットのオペレーティング・システムで使用可能なアドレス空間(通常2GBまたは3GB)に制限を設定することを考慮して、このパラメータを少なくとも1.2GBに設定することをお勧めします。一般に、値が高いほどOracleデータベースのパフォーマンスは良好になるため、仮想アドレス空間が不足することなく可能なかぎり高い値にこのパラメータを設定してください。

推奨設定: 最小1.2GB、通常(環境に応じて)1.2GB超の値。

Oracle 11gインスタンスに割り当てるメモリー量を決定する場合は、次の項を確認してください:

[499ページのOracle Databaseリリース11gのプロセス数を計算する方法](#)

MEMORY_MAX_TARGET

MEMORY_MAX_TARGETは、DBAがMEMORY_TARGETパラメータに設定できる最大値を指定します。上限として機能するため、MEMORY_TARGETを誤って高すぎる値に設定することはできません。また、再起動せず実行時にMEMORY_TARGETを増す場合に備えて、Oracleデータベース・インスタンスのメモリーを予約します。したがって、MEMORY_MAX_TARGETをMEMORY_TARGETの値以上にする必要があります。

推奨設定: MEMORY_TARGETの値以上

SGA_TARGET

SGA_TARGETは、すべてのSGAコンポーネントの合計サイズを指定します。自動メモリー管理が有効になっており、SGA_TARGETがゼロ以外の値に設定されている場合、この値はSGAの最小サイズとして機能します。

推奨設定: 自動メモリー管理が有効になっている場合は0 (ゼロ)。それ以外の場合は、[496ページのOracle Databaseリリース11gのメモリー設定を決定する方法](#)を参照してください。

SGA_MAX_SIZE

SGA_MAX_SIZEは、インスタンスの存続期間を通したSGAの最大サイズを指定します。このパラメータは、SGA_TARGETの上限を設定します。自動メモリー管理が有効になっている場合、OracleデータベースではSGA_MAX_SIZEを超えてSGAコンポーネントの合計サイズを増やすことはできません。

推奨設定: 自動メモリー管理が有効になっている場合はデフォルト設定。それ以外の場合は、[496ページのOracle Databaseリリース11gのメモリー設定を決定する方法](#)を参照してください。

PGA_AGGREGATE_TARGET

PGA_AGGREGATE_TARGETは、インスタンスに接続されたすべてのサーバー・プロセスで使用できる合計PGAメモリーを指定します。自動メモリー管理が有効になっており、PGA_AGGREGATE_TARGETがゼロ以外の値に設定されている場合、この値はPGAの最小サイズとして機能します。

推奨設定: 自動メモリー管理が有効になっている場合は0 (ゼロ)。それ以外の場合は、[496ページのOracle Databaseリリース11gのメモリー設定を決定する方法](#)を参照してください。

LOG_BUFFER

LOG_BUFFERは、REDOログ・ファイルへのREDOエントリのバッファリングに使用されるメモリー量(バイト単位)を指定します。REDOログ・エントリには、データベース・ブロック・バッファに対して行われた変更の記録が含まれて

います。Financial Managementは更新頻度の高いトランザクション・システムであり、データベースは常にログ・バッファを使用します。ログ・バッファのサイズを適切に設定すると、データベース・パフォーマンスを向上させることができます。一般に、LOG_BUFFERの値を大きくすると、特にトランザクションが長い場合や多数の場合に、REDOログ・ファイルのI/Oが減ります。バッファが小さすぎると、バッファがクリアされるまで待機してから新しい更新が追加されるため、データベースのパフォーマンスを向上させるには、このバッファを正しく設定することが重要です。

推奨設定: 8MBから始めます。 [496ページのOracle Databaseリリース11gのメモリー設定を決定する方法](#)を参照してください。

OPTIMIZER_MODE

OPTIMIZER_MODEは、インスタンスの最適化方法を選択するためのデフォルト動作を設定します。オプティマイザ・モードは、オンライン・アプリケーションのレスポンスを最適にするためにFIRST_ROWSに設定するか、バッチ操作の合計実行時間を最小限に抑えるためにALL_ROWSに設定できます。Financial Managementでは戻されたデータセット全体のみを処理するため、合計実行時間を最小限に抑える方が適切です。

推奨設定: ALL_ROWS

OPTIMIZER_INDEX_COST_ADJ

OPTIMIZER_INDEX_COST_ADJを指定すると、アクセス・パスの選択するときに索引を使いやすくするか否か、オプティマイザの動作をチューニングできます。つまり、オプティマイザが全表スキャンよりも索引アクセス・パスを選択する傾向を強めたり弱めたりできます。このパラメータのデフォルト値は100パーセントで、その場合、オプティマイザは索引アクセス・パスを通常コストで評価します。しかし、Financial Managementのトランザクションは通常、全表スキャン・パスよりも索引アクセス・パスの方を選ぶため、このパラメータの値を低くすることをお勧めします。

推奨設定: 50

PROCESSES

PROCESSESは、Oracleデータベースに同時に接続できるオペレーティング・システムのユーザー・プロセスの最大数を指定します。Financial Managementが専用サーバーとのみ連携するため、各接続にはプロセスが必要です。各Financial Managementアプリケーションでは、Financial Managementのデータベース接続プール設定で指定された数に加えて、最低20のデータベース接続を必要とします。たとえば、Financial Managementのデータベース接続プール設定が40の場合、各アプリケーション・サーバーのFinancial Managementアプリケーションごとに、少なくとも60のOracleデータベース接続が必要です。

推奨設定: [499ページのOracle Databaseリリース11gのプロセス数を計算する方法](#)を参照してください。

SESSIONS

このパラメータは、データベース・システムで作成できるセッションの最大数を指定します。すべてのログインにはセッションが必要であるため、このパラメータは実質的にOracleデータベースの同時ユーザーの最大数を決定します。デ

フォルト値は、 $1.1 * PROCESSES + 5$ です。デフォルト値を下回る値にこのパラメータを設定することはお薦めしません。

TRANSACTIONS

TRANSACTIONSは、同時トランザクションの最大数を指定します。トランザクションによっては再帰的であるため、このパラメータはSESSIONSより大きい値にし、SESSIONSはPROCESSESより大きい値にして再帰的なトランザクションを考慮する必要があります。デフォルト値は、 $1.1 * SESSIONS$ です。デフォルト値を下回る値にこのパラメータを設定することはお薦めしません。

OPEN_CURSORS

OPEN_CURSORSは、1つのセッションで一度に保持できるオープン・カーソル(プライベートSQL領域へのハンドルの)の最大数を指定します。アプリケーションでオープン・カーソルが不足しない程度に十分高いOPEN_CURSORSの値に設定することが重要です。OPEN_CURSORSで指定したカーソル数をセッションでオープンしなければ、この値を実際に必要な数より高く設定しても、オーバーヘッドは追加されません。

推奨設定: 5000

SESSION_CACHED_CURSORS

SESSION_CACHED_CURSORSは、キャッシュするセッション・カーソル数を指定します。同じSQL文に対して解析コールを繰り返すと、その文のセッション・カーソルがセッション・カーソル・キャッシュに移動されます。その後の解析コールではキャッシュ内にカーソルが見つかるため、カーソルを再オープンする必要がありません。Financial Management接続もキャッシュされるため、Financial Managementアプリケーションのパフォーマンスはこのキャッシュからの恩恵を受けます。

推奨設定: 50

TRACE_ENABLED

TRACE_ENABLEDは、Oracleデータベースの実行履歴またはコード・パスのトレースを制御します。パラメータをTRUEに設定してこのオプションを有効にすると、データベースにオーバーヘッドがさらに追加されるため、通常のFinancial Managementアプリケーション環境ではお薦めしません。

推奨設定: FALSE

STATISTICS_LEVEL

STATISTICS_LEVELは、データベースおよびオペレーティング・システムの統計収集のレベルを指定します。Oracleデータベースでは、自己管理に関する意思決定などの様々な目的のために、これらの統計が収集されま

す。デフォルト設定のTYPICALでは、データベースの自己管理機能に必要なすべての主要統計が収集され、全体的なパフォーマンスが最適化されます。

推奨設定: TYPICAL

TIMED_STATISTICS

TIMED_STATISTICSは、時間に関連する統計情報を収集するかどうかを指定します。Oracle Database リリース11.1.0.7.0以降では、STATISTICS_LEVELの値がTYPICALまたはALLに設定されていると、TIMED_STATISTICSパラメータの値をFALSEに設定できません。

推奨設定: TRUE

TIMED_OS_STATISTICS

TIMED_OS_STATISTICSは、クライアントからサーバーへの要求が発生したとき、または要求が完了したときに、Oracleでオペレーティング・システムの統計情報を収集する間隔(秒単位)を指定します。パラメータを0 (ゼロ)より大きい数値に設定してこのオプションを有効にすると、アプリケーションのパフォーマンスが大幅に低下します。

推奨設定: 0

Oracle Databaseリリース11gのメモリー設定を決定する方法

この項では、Oracleシステム関連の統計をモニターおよび表示する方法と、Oracleデータベースのメモリー・パラメータをチューニングする方法を説明します。最適なメモリー設定を決定する方法はいくつもありますが、望ましい方法は、メモリー・アドバイザ、SGAアドバイザ、共有プール・アドバイザ、バッファ・キャッシュ・アドバイザ、PGAアドバイザなどのメモリー・アドバイザを使用することです。このようなアドバイザを使用して次のタスクを実行するには、DBA権限によるOracleログインが必要です。次の問合せの大部分には、Oracle Enterprise Managerによる同等のグラフィカル・インタフェースがあります。

合計メモリーのサイズ設定(MEMORY_TARGET)

MEMORY_TARGETは、SGAとPGAの両方を含め、Oracleシステム全体の使用可能なメモリーを指定します。Oracle Databaseリリース11gより前では、SGAおよびPGAを別個にチューニングする必要がありました。

データベースをOracle 10gから11gにアップグレードする場合、Oracle 10gデータベースのSGA_TARGETとPGA_AGGREGATE_TARGETを加算してMEMORY_TARGETを求めることができます。

データベースをOracle 9iから11gにアップグレードする場合、

DB_CACHE_SIZE、SHARED_POOL_SIZE、LARGE_POOL_SIZE、JAVA_POOL_SIZEなどのすべてのSGAコンポーネントとPGA_AGGREGATE_TARGETを加算してMEMORY_TARGETを求めることができます。

データベースが以前のバージョンからアップグレードされておらず、履歴参照がない場合、システム・リソースおよびシステム制限に応じて、このパラメータを1から3GBに初期設定することをお勧めします。データベースをしばらくの間使

用した後、このパラメータを次のように調整できます。(このチューニングは前述の2つのアップグレード・シナリオにも適用されます)。

```
SQL> select * from v$memory_target_advice order by memory_size;
```

```
MEMORY_SIZE MEMORY_SIZE_FACTOR ESTD_DB_TIME ESTD_DB_TIME_FACTOR VERSION
```

```
180 .5 458 1.344 0
```

```
270 .75 367 1.0761 0
```

```
360 1 341 1 0
```

```
450 1.25 335 .9817 0
```

```
540 1.5 335 .9817 0
```

```
630 1.75 335 .9817 0
```

```
720 2 335 .9817 0
```

MEMORY_SIZE_FACTORが1の行には、MEMORY_TARGET初期化パラメータで設定した現在のメモリのサイズと、現在のワークロードを完了するのに必要なDB時間が表示されます。前後の行の結果には、代替MEMORY_TARGETサイズの数値が表示されます。代替サイズごとに、データベースのサイズ係数(現在のサイズの倍数)と、MEMORY_TARGETパラメータが代替サイズに変更された場合に現在のワークロードを完了するための推定DB時間が表示されます。現在のMEMORY_TARGETサイズ(この例では360)より合計メモリ・サイズが小さいほど、推定DB時間(ESTD_DB_TIME)が増えます。また、この例では、ESTD_DB_TIME値が減らないため、合計メモリ・サイズを450MBを超えて増やしても何も得られません。したがって、この例では、推奨MEMORY_TARGETサイズは450MBです。

SGAのサイズ設定(SGA_TARGET)

SGAは通常、自動メモリ管理が有効になっている場合、Oracleデータベースによって自動的にチューニングされます。しかし、DBAがSGAのサイズを引き続きモニターし、最適なサイズになっているかどうかを調べることができます。

```
SQL> select * from v$sga_target_advice order by sga_size;
```

```
SGA_SIZE SGA_SIZE_FACTOR ESTD_DB_TIME ESTD_DB_TIME_FACTOR ESTD_PHYSICAL_READS
```

```
290 .5 448176 1.6578 1636103
```

```
435 .75 339336 1.2552 1636103
```

```
580 1 201866 1 513881
```

```
725 1.25 201866 1 513881
```

```
870 1.5 201866 1 513881
```

```
1015 1.75 201866 1 513881
```

```
1160 2 201866 1 513881
```

「合計メモリーのサイズ設定」の項と同様の分析に基づいた場合、SGA_TARGETの現在の設定はすでに最適です。

PGAのサイズ設定(PGA_AGGREGATE_TARGET)

SGAと同様にPGAも、自動メモリー管理が有効になっている場合、Oracleデータベースによって自動的にチューニングされます。次の問合せは、PGAサイズが適切に設定されているかどうかをモニターするために使用できます。結果は、v\$memory_target_sizeおよびv\$sqla_target_sizeの問合せ結果に似ています。

```
SQL> select * from v$sqla_target_advice order by pga_target_for_estimate;
```

LOG_BUFFERのサイズ設定

システム・ビューv\$sysstatのREDOログ・バッファ割当て再試行回数の値には、ユーザー・プロセスがREDOログ・バッファ内のスペースを待機した回数が反映されます。この値は適切にサイズ設定されたデータベースに対してゼロに近い必要があります。例:

```
select name, value
```

```
from v$sysstat
```

```
where name = 'redo buffer allocation retries'
```

```
NAME VALUE
```

```
redo buffer allocation retries 1021967
```

ログ・バッファに更新用のスペースがない場合、データベースは待機してから再試行する必要があります。この例では、データベースは合計1,021,967回再試行しています。パフォーマンスを向上させるには、LOG_BUFFERパラメータ値を増やします。この値はバイト単位で表され、「ログ・ブロック・サイズ」値(オペレーティング・システムのブロック・サイズ)の倍数である必要があります。Financial Managementアプリケーションの場合、LOG_BUFFERを最小値8MBに設定した後、前述の問合せを使用してパフォーマンスをモニターし、必要に応じて増やします。LOG_BUFFERを増やす必要がある場合は、一度に50%ずつ増やすことをお勧めします。

Oracle Databaseリリース11gのプロセス数を計算する方法

Oracleデータベースに同時に接続できるユーザー・プロセスの数は、Oracleデータベース・インスタンスのPROCESSES初期化パラメータによって制限されます。アプリケーションの場合、このパラメータの値はそのアプリケーションの接続プール・サイズ・パラメータによって異なります。各Financial Managementアプリケーションでは、Financial Managementのデータベース接続プール設定値に加えて、最低20のデータベース接続を必要とします。たとえば、Financial Managementのデータベース接続プール設定が40の場合、Financial Managementアプリケーションごとに、少なくとも60のOracleデータベース接続が必要です。その結果、PROCESSESの値は60より大きい値に設定する必要があります。



注:

クラスタ内のサーバーの合計数とアプリケーションの合計数は、必要なデータベース接続の数に影響します。

次の例では、Oracleデータベースに接続するプロセスの数を計算する方法を示します。2つのFinancial Managementアプリケーション・サーバーで構成されるクラスタで、各サーバーに2つのアプリケーションがあるとして、Financial Managementのデータベース接続プール設定は40です。Oracleデータベース接続の最小数は、接続プール設定の合計に20を加算し、アプリケーションの数を乗算し、アプリケーション・サーバーの数を乗算して算出します: $(40 + 20) * 2 * 2 = 240$ 。安全性を高めるためには、この数値に係数1.1を乗算して、補助的な接続およびデータベースの一般的な使用量を考慮します。Oracleデータベースにはバックグラウンド・プロセスもあることを考慮して、この数値に20を加算してPROCESSESの値を求めます。したがって、この場合、PROCESSESを260に設定する必要があります。

通常、

PROCESSES = (Financial Managementの接続プール設定 + 20) * (Financial Managementアプリケーションの数) * (Financial Managementアプリケーション・サーバーの数) * 1.1 + 20

その他の考慮事項

共有サーバーと専用サーバー

Financial Managementでは、Oracleデータベースへの接続がすべて専用サーバー・プロセスによって処理される必要があります。つまり、Financial Managementは共有サーバー・プロセスとは連携しません。専用サーバー・プロセスでは、CPUおよびメモリー・リソースの消費が多くなりますが、高いパフォーマンスが実現します。専用サーバーを使用するには、ネット・サービス名の値の接続記述子にSERVER=DEDICATED句を含める必要があります。次に、専用サーバー・プロセス用に構成されたネット・サービスの例を示します。

```
HFMDB = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = hfm.oracle.com)(PORT = 1521))(CONNECT_DATA =(SERVER = DEDICATED)(SERVICE_NAME = HFMDB1) ) )
```

オンラインREDOログ・ファイルのサイズ

データベースのライター・プロセスおよびアーカイバ・プロセスの動作はREDOログのサイズによって決まるため、REDOログ・ファイルのサイズがパフォーマンスに影響する可能性があります。通常、REDOログ・ファイルの規模が大きくなると、パフォーマンスは向上します。規模が小さいログ・ファイルでは、チェックポイント・アクティビティが増加し、パフォーマンスが低下します。ただし、小規模のログ・ファイルと頻繁なチェックポイントによってリカバリ時間は短縮されます。そのため、毎日の業務効率がリカバリ時間を最小限に抑えることよりも重要である場合は、オンラインREDOログ・ファイルを比較的大きい値に設定します。Financial Managementデータベースの場合、何百MBが通常サイズです。ただし、REDOログ・ファイルのサイズを決定する望ましい方法は、FAST_START_MTTR_TARGETを有効にして、しばらくの間一般的なデータベース・ワークロードを実行することです。その後、次の問合せを実行してREDOログの最適なサイズを取得します。

```
SQL> select optimal_logfile_size from v$instance_recovery;
```

MTTRターゲットとオンラインREDOファイルのチューニング方法の詳細は、『Oracle Databaseパフォーマンス・チューニング・ガイド』を参照してください。

表領域とセグメントの断片化

時間の経過とともに、表領域内でのオブジェクトに対する更新および削除により、新しいデータに再利用されるほど個々には大きくない空のスペースのポケットが作成される傾向があります。このような空のスペースを断片化された空きスペースと呼びます。断片化された空きスペースがあるオブジェクトは無駄なスペースが多くなり、データベースのパフォーマンスに影響を及ぼすことがあります。Financial Managementの連結では、広い範囲にわたって更新、挿入および削除を実行するため、表領域の断片化をモニターして定期的に最適化することは非常に重要です。

このスペースを最適化して再利用する望ましい方法は、オンライン・セグメント縮小を実行することです。オンライン・セグメントの使用の詳細は、『Oracle Database管理者ガイド』を参照するか、Oracleデータベース・サポート・サービスに問い合わせてください。

インデックスの断片化

Financial Managementアプリケーションは通常、インデックスを数百または数千すら作成します。時間の経過とともにアプリケーション・データが変化すると、インデックスが断片化される場合があります。このようなインデックスを定期的にモニタリングして最適化すると、パフォーマンスを向上させることができます。ただし、インデックスの再構築は時間がかかるリソース集約型の操作です。アプリケーションの稼働中にインデックスを再構築することはお勧めしません。Enterprise Managerには、インデックスの統計をモニターするための使いやすいインタフェースが用意されています。Oracle Enterprise Managerを使用してインデックスをモニターおよび最適化する方法の詳細は、Oracleデータベースのドキュメントを参照してください。

機能DEFERRED_SEGMENT_CREATIONの無効化

リリース11.2には、DEFERRED_SEGMENT_CREATION機能が導入されています。すべてのインストールで、デフォルト設定はONです。この機能により、TABLE create文で表が実際には作成されなくなります。表は、データの行が挿入された後でのみ作成されます。

機能DEFERRED_SEGMENT_CREATIONの無効化

リリース11.2には、DEFERRED_SEGMENT_CREATION機能が導入されています。すべてのインストールで、デフォルト設定はONです。この機能により、TABLE create文で表が実際には作成されなくなります。表は、データの行が挿入された後でのみ作成されます。表によってはインポート中に作成されないことがあるため、この機能が原因で、Financial Managementスキーマのエクスポートおよびインポート時に問題が発生する可能性があります。この機能を無効にすることをお勧めします。その場合、表を自動的に作成する必要があります。この機能を無効にするには、SYSTEMまたはSYSを使用してインスタンスにログインし、次のコマンドを発行します：

```
alter system set deferred_segment_creation=false;
```

この文の発行後に作成される表は自動的に作成されます。空の表が含まれるインスタンスがすでに存在する場合にアプリケーションをエクスポートするには、各表を強制的に作成するように個別に変更し、表がEXP (エクスポート) コマンドで使用されることを許可できます。

スキーマが空の表かどうかを判別するには、次の2つのコマンドのいずれかを実行します：

```
select segment_name, segment_type, extents from dba_segments where extents < 1  
and segment_type='TABLE' and owner='<hfm db schema>'
```

```
select table_name from all_tables where owner='<hfm db schema>' and table_name  
not in (select segment_name from dba_segments where owner='<hfm db schema>' and  
segment_type='TABLE' and extents>0)
```

空の表ごとに次のコマンドを発行します:

```
alter table <table_name> allocate extent
```

定期的なメンテナンスおよびチューニング・プラン

これまでの各項では、Oracleのメモリー・パラメータのサイズを正しく設定する一般的なプロセスについて概説しました。その性質上、パフォーマンス・チューニングは反復します。パフォーマンス・ブロックを1つ排除すると、別のブロックが現れることもあるため、パフォーマンスがすぐに向上しない可能性があります。したがって、このプロセスはパフォーマンスが許容範囲内になるまで繰り返す必要があります。Financial Managementアプリケーション・データは全期間を通して絶えず変化するため、先を見越してOracleデータベースのパフォーマンスをモニターしてチューニングするのに定期的なデータベースのメンテナンスおよびチューニング・プランが役立ち、将来発生する可能性があるパフォーマンスの問題を回避します。詳細およびその他のチューニング・オプションは、Oracleデータベース・サポートに問い合せてください。

よくある質問

どのオペレーティング・システムがサポートされていますか?

Oracle Enterprise Performance Management Systemサポート対象プラットフォームのマトリックス([Oracle EPM Supported Platform Matrix](#))を参照してください。

リレーショナル・データベースは64ビットである必要がありますか?

データベースは、サポートされているDBMSのタイプとバージョンであれば、32ビットと64ビットのいずれでもかまいません。

サードパーティ・ソフトウェアおよび拡張ソフトウェアを考慮する必要があります。デフォルトでは、Financial Managementのインストールで64ビットのオペレーティング・システムにインストールされるのは64ビットのソフトウェアのみです。つまり、64ビットのクライアント・コンポーネントのみがFinancial Managementアプリケーション・サーバーにインストールされます。32ビットの接続が必要な場合、32ビットのクライアント・ソフトウェアが64ビットのアプリケーション・サーバーにインストールされないかぎり、コンポーネントが機能しないことがあります。詳細は、『Oracle Enterprise Performance Management Systemインストールおよび構成ガイド』を参照してください。

64ビットのFinancial Managementにはどのようなメモリー制限がありますか?

実際の問題として、64ビットのFinancial Managementは仮想メモリーではなく物理メモリーによって制限されます。メモリー・パラメータを適切に調整した後であれば、使用可能な物理メモリーをすべて利用できます。

Financial Management用にチューニングが必要なメモリー設定はありますか?

Financial Managementのデフォルトのメモリー設定は、32ビット環境での小規模から中規模サイズのアプリケーションに適しています。使用可能なメモリーを最大限に利用するために、月次のアプリケーションには次の設定をお勧めします。関連する設定はMaxNumDataRecordsInRAMおよびMaxDataCacheSizeinMBで、変更する必要があります。次の表に、使用可能なメモリーに応じたこれらのパラメータの推奨値を示します。この表では、Financial Managementがマシンで実行されている唯一のメモリー集約型プロセスであり、Financial Managementアプリケーションを1つしか実行していないということを前提としています。複数のアプリケーションがアクティブの場合、サーバーに搭載されている物理メモリーの合計をFinancial Managementアプリケーションの数で除算して、各アプリケーションの使用可能な物理メモリーを求めます。

使用可能な物理メモリー	MaxNumDataRecordsinRAM	MaxDataCacheSizeinMB
4	4,000,000	500
8	10,000,000	1500
16	30,000,000	4500
32	60,000,000	9000

例: RAMが24 GBのサーバーで月次のOracle Hyperion Financial Managementアプリケーションが2つアクティブであり、MaxNumDataRecordsInRAMの値は22,500,000、MaxDataCacheSizeinMBの値は3375です。

平均で、月次のアプリケーションでは1データ・レコードに使用されるメモリー量は112バイトであり、週次のアプリケーションでは1データ・レコード当たり472バイトを使用し、日次のアプリケーションでは1データ・レコード当たり3,296バイトを使用します。

週次のアプリケーションの場合、MaxNumDataRecordsInRAMを4で除算します。MaxDataCacheSizeinMBについては、最後の列の値を変更する必要はありません。

日次のアプリケーションの場合、MaxNumDataRecordsInRAMを30で除算します。MaxDataCacheSizeinMBについては、最後の列の値を変更する必要はありません。

