

Oracle® Hyperion Financial Management

Administrator's Guide

リリース 11.1.2.3

Financial Management Administrator's Guide, 11.1.2.3

Copyright © 2000, 2013, 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。

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 RIGHTS:

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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際、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により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	15
第 1 章 Financial Management のバージョン情報	17
Financial Management の機能	17
アーキテクチャ	18
Performance Management Architect	18
EPM Workspace	18
Financial Management の次元	18
シナリオ次元	19
年次元	19
期間次元	19
エンティティ次元	19
値次元	20
勘定科目次元	20
内部取引次元	20
ビュー次元	20
カスタム次元	21
ユーザー定義の要素	21
第 2 章 アプリケーションの管理	25
アプリケーションの管理	26
アプリケーション・プロファイルの作成	27
言語の入力	28
カレンダーの定義	28
期間単位の編集	29
期間の編集	30
カスタム次元の作成	32
アプリケーション・プロファイルの保存	34
アプリケーションの作成	35
アプリケーションの登録	36
アプリケーション・モジュールの構成	37
カスタム次元を追加するためのアプリケーションの変更	38
アプリケーションを開く	39

アプリケーションを閉じる	39
アプリケーション・サーバー・クラスタの変更	39
アプリケーション・リストの表示	40
アプリケーションの削除	40
アプリケーション要素のロード	41
アプリケーション要素の抽出	42
データの抽出のデータベース情報の指定	42
クラシック管理アプリケーションのコピー	44
クラシック管理アプリケーションの名前変更	46
Performance Management Architect アプリケーションのコピー	47
クラシック・アプリケーションとしてのコピー	48
Performance Management Architect としてのコピー(ターゲットにソース・アプリケーションが存在する場合)	50
サンプル・アプリケーションの使用方法	52
システム・メッセージの使用	53
システム・メッセージの表示と印刷	53
システム・メッセージの削除	54
システム・メッセージの詳細文字列	54
システム・ユーザーの管理	55
ユーザーの表示	55
ユーザーのログアウト	56
サーバーとアプリケーションの管理	57
接続の使用不可と使用可能	57
サーバーの同期化	58
タスクの監査	58
データの監査	61
タスクの実行の監視	62
無効なレコードのスキャンと消去	64
第3章 アプリケーション・セキュリティの管理	65
アプリケーション・セキュリティに関する考慮事項	66
Financial Management からの Shared Services Console の起動	66
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	67
Financial Management のアプリケーションのセキュリティ・クラスの設定	67
セキュリティ・クラスへのユーザー・アクセス権の割当て	70
電子メール・アラートの設定	71
Financial Management アプリケーションに対するセキュリティ・レポートの実行	72
アプリケーション・セキュリティのロード	72

セキュリティ情報の消去とロード	74
セキュリティ情報を消去する前に行っておくべきタスク	74
セキュリティ情報を消去した後で行うタスク	75
アプリケーション・セキュリティの抽出	76
第4章 メタデータの管理	79
勘定科目の定義	80
勘定科目の種別の動作	82
動的勘定科目の定義	84
カスタム・メンバーの定義	84
カスタム次元情報の追加	86
カスタム次元情報のインポート	86
カスタム次元情報の手動での編集	87
エンティティ・メンバーの定義	87
シナリオ・メンバーの定義	89
アプリケーション設定の定義	91
期間別組織	94
連結メソッドの定義	94
連結メソッドの使用方法	96
手動での連結メソッドの割当て	96
POWN または POWNMIN 連結メソッドの使用	97
通貨の定義	98
セル・テキスト・ラベルの定義	100
システム生成の勘定科目	102
連結勘定科目	102
所有権勘定科目	103
システムで生成された勘定科目の編集	103
内部取引パートナーの設定	104
システムで生成された ICP 次元のメンバーの編集	105
システム生成の値メンバーの編集	105
セキュリティに基づくメタデータのフィルタ	106
APP フォーマットのメタデータ・ファイルの作成	106
ファイル・フォーマット	108
バージョン	108
アプリケーション設定	109
通貨	110
メンバー	110
連結メソッド	115
階層	116

メタデータ・ファイルに含まれない次元	119
メタデータ・マネージャのビューの使用方法	120
メタデータ・ファイルのフォーマットの変更	121
ツリー・ビューでのタスク	122
リスト・ビューでのタスク	126
リスト・ビューのメタデータのソート	128
「ファイルのプロパティ」でのメタデータ・レポートの作成	129
メタデータの参照整合性	129
メタデータの参照整合性チェック	130
メタデータ・ログ・ファイルの参照整合性エラー	130
メタデータのロード	130
メタデータ・ロードの変更の表示	133
メタデータの抽出	134
第 5 章 メンバー・リストの管理	137
メンバー・リスト・ファイルの作成	138
EnumMemberLists	138
EnumMembersInList	139
動的メンバー・リスト	142
動的 POV メンバー・リスト	143
メンバー・リストのロード	146
メンバー・リストの抽出	147
システム生成のリストと次元	148
第 6 章 仕訳の管理	149
仕訳ファイルの作成	149
ファイル・フォーマット・セクション	150
バージョン・セクション	150
仕訳グループ・セクション	151
標準セクション	151
繰返しセクション	151
ヘッダー・セクション	152
仕訳のロード	154
仕訳の抽出	155
第 7 章 データ・フォームの管理	157
フォーム・デザイナーでのデータ・フォームの作成	158
視点の設定	159
フォームの詳細の指定	159
データ・フォームのオンデマンド・ルールの指定	160

フォームの行および列オプションの指定	161
フォームのヘッダーの指定	161
データ・フォーム・スクリプトの使用	161
AddMember	164
BackgroundPOV	166
Blank	167
Cn	168
Calc1	169
CalcByRow	169
Cell_Link	170
CellText	171
CustomHeader	171
CustomHeaderStyle	172
DynamicPOV	173
FormInputBoxLength	173
FormNumDecimals	174
FormRowHeight	174
FormScale	175
HeaderOption	175
HideInPov	176
指示	177
LineItemDetailSinglePeriod	177
リンク	178
MaxCells	178
MaxColsForSparseRetrievalMethod	179
NoSuppress	179
NumDecimals	179
OnDemandRules	180
上書き	180
POVOrder	182
PrintNumDataColsPerPage	182
PrintNumRowsPerPage	183
PrintRepeatHeadersonAllPages	183
Rn	184
ReadOnly	185
ReportDescription	186
ReportLabel	186
ReportSecurityClass	187
ReportType	187

RowHeaderPct	187
SCalc	188
Scale	189
SelectablePOVList	190
ShowDescriptions	191
ShowLabels	191
文字列	192
Style	192
SuppressColHeaderRepeats	194
SuppressInvalidColumns	195
SuppressInvalidRows	195
SuppressNoDataColumns	196
SuppressNoDataRows	196
SuppressRowHeaderRepeats	197
SuppressZeroColumns	197
SuppressZeroRows	197
相対期間の使用	198
競合する属性の優先順位	198
データ・フォームの編集	199
データ・フォームのロード	200
データ・フォームの抽出	201
データ・フォームの削除	201
第 8 章 データベースへのデータの抽出	203
データベースにデータを抽出するためのデータ・リンク(UDL)ファイルの作成	204
Oracle 用データ・リンクの構成	204
Microsoft SQL Server 用データ・リンクの構成	205
IBM DB2 用データ・リンクの構成	206
UDL ファイルの暗号化	206
スター・スキーマ	207
スター・スキーマのフォーマット	208
接頭辞テーブル	209
データの作成とスター・スキーマへのエクスポート	210
スター・スキーマの更新	211
スター・スキーマの削除	213
スター・スキーマのテンプレートの作成	213
スター・スキーマのテンプレートの削除	214

第 9 章 レポートの定義	215
仕訳レポート・スクリプトの定義	215
内部取引照合レポート・スクリプトの定義	216
内部取引照合レポートのメンバー・リストの選択	217
内部取引照合レポートの勘定科目の選択	217
内部取引照合レポートの小数点以下の桁数の指定	217
内部取引照合レポート用スタイル・シートの選択	217
内部取引照合レポートの通貨の指定	217
内部取引照合レポートの非表示オプション	218
内部取引照合レポート・スクリプトのキーワード	219
内部取引レポート・スクリプトの定義	228
第 10 章 ルールの管理	231
ルールのタイプ	231
ルールに関する注意事項	233
計算コマンド	234
現在の次元メンバー	238
勘定科目式	239
関数によるデータの自動消去	240
エラー・メッセージ	241
連結時のルールの実行	241
デフォルトの換算方法	242
Financial Management のオブジェクト	243
よく使用するルール	244
データの再使用	244
値の計算による勘定科目の設定	245
条件付きのルール	245
すべての勘定科目の期首残高の設定	247
Notepad ++でのルールの作成	248
Notepad ++の起動	248
Notepad ++の構成	248
ルール・ファイルの作成	249
ルールのロード	251
ルールの抽出	253
第 11 章 ルール関数	255
関数の概要	258
ABSExp	265
AccountType	266
AccountTypeID	267

AddEntityToList	268
AddEntityToListUsingIDs	269
AddMemberToList	269
AddMemberToListUsingIDs	270
Alloc	270
AllowAdjFromChildren	272
AllowAdjs	273
ApplicationName	274
CalculateExchangeRate	274
CalculateRate	275
CalcStatus	276
Clear	278
Con	279
Consol1、Consol2、Consol3	280
ConsolidateYTD	281
ContainsCellText	282
ContainsCellTextWithLabel	283
Currency	284
CustomTop	285
DataUnitItem	286
Decimal	287
DefaultFreq	288
DefaultParent	289
DefaultTranslate	290
DefaultView	291
DefCurrency	292
DOWn	293
Dynamic	294
Exp	295
次元の交差に関する考慮事項	297
期間および年のキーワード	297
算術計算	298
Exp 内での他の関数の配置	299
複数の勘定科目の同時設定	299
Exp および次元の交差に関する考慮事項	300
GetCell	303
GetCellNoData	303
GetCellRealData	305
GetCellText	306

GetCellTextWithLabel	306
GetCellType	307
GetCustomLabelArray	308
GetItem	309
GetItemIDs2	310
GetItemIDs2ExtDim	311
GetNumItems	312
GetNumLID	313
GetRate	314
GetSubmissionGroup	315
GetSubmissionPhase	315
Holding	316
ICPTopMember	317
ICPWeight	318
IDFromMember	319
ImpactStatus	320
Input	321
IsAlmostEqual	322
IsBase	323
IsCalculated	325
IsChild	326
IsConsolidated	328
IsDescendant	329
IsFirst	331
IsICP	332
IsLast	333
IsTransCur	334
IsTransCurAdj	335
IsValidDest	335
IsZero	336
List	337
Member	339
MemberFromID	340
Method	342
NoInput	342
NoRound	343
NumBase	344
Number	346
NumChild	347

NumCustom	349
NumDescendant	349
NumPerInGen	352
NumPeriods	353
OpenCellTextUnit	354
OpenDataUnit	355
OpenDataUnitSorted	356
Owned	356
Owner	357
PCon	357
PEPU	358
PeriodNumber	359
PlugAcct	360
POwn	361
PVAForBalance	362
PVAForFlow	363
RateForBalance	363
RateForFlow	364
ReviewStatus	365
ReviewStatusUsingPhaseID	366
Round	367
Scale	368
SecurityAsPartner	369
SecurityClass	371
SetCellTextWithLabel	373
SetData	373
SetDataWithPOV	374
SubmissionGroup	375
SupportsProcessManagement	376
SupportsTran	377
SwitchSign	378
SwitchType	379
Trans	381
TransPeriodic	382
UD1...3	383
ValidationAccount	384
ValidationAccountEx	385
XBRLTags	385

第 12 章 カスタム関数	387
管理レポート関数	388
Average	388
Cumulative	391
Difference	394
DSO - 売上債権回転日数	396
Opening	399
Rate	402
ビジネス・ルール関数	408
Custom_Alloc	408
Increase_Decrease	412
Pro_Rata_Ratio	414
Spread	416
Units_Rates	418
第 13 章 Calculation Manager を使用したルールの作成	421
Calculation Manager のセキュリティ役割	421
Calculation Manager でのアプリケーションの操作	422
Calculation Manager へのルールの移行	422
関数セレクタでの VB 関数のサポート	423
Financial Management の特別な VB Script 関数	424
第 14 章 内部取引の管理	427
内部取引の設定	427
内部取引の期間を開く	427
照合許容差を設定する	428
「転記前に照合」オプションの設定	429
内部取引の期間を閉じる	430
理由コードの管理	430
理由コードの追加	431
理由コードの編集	431
理由コードの削除	431
内部取引のモニター	432
エンティティのロックとロック解除	433
内部取引の要約の表示	434
第 15 章 プロセス管理の提出フェーズの管理	437
提出フェーズの定義	437
送信グループの設定	438
提出グループおよびフェーズの例	439

提出グループのフェーズへの割当て	442
未割当ての提出グループの表示	443
第 16 章 電子メール・アラートの管理	445
プロセス管理のアラートの設定	445
内部取引のアラートの設定	447
付録 A. レジストリ設定	449
用語集	457
索引	461

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。

1

Financial Managementのバージョン情報

この章の内容

Financial Management の機能	17
アーキテクチャ	18
Performance Management Architect	18
EPM Workspace	18
Financial Management の次元	18
ユーザー定義の要素	21

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。
- パワフルな多次元分析機能。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。各部署への情報の配賦、様々な通貨の換算、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

アーキテクチャ

Financial Management は、多層システムとして機能するように設計されています。

- クライアントを使用して、アプリケーション・プロファイル情報およびメタデータを作成および保守できます。
- Web サーバー層では、Financial Management の Web 対応部分にアクセスします。
- 中間層には、ドメインに関する機能、およびリレーショナル・データベースとの接続が含まれます。
- データ層には、リレーショナル・データベースと、すべての Financial Management データおよびメタデータが含まれています。

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、次元、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 Administrator's Guide 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 User's Guide およびオンライン・ヘルプを参照してください。

Financial Management の次元

次元は、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。次元の例には、勘定科目、エンティティおよび期間があります。Financial Management は、システム定義の次元を 8 つ提供し、勘定科目に適用できるカスタム次元を無制限に追加できます。

次元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 は勘定科目次元のメンバーです。

次元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、データは次元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、システム定義の次元について説明します。次元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 Administrator's Guide 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、[第4章「メタデータの管理」](#)を参照してください。

シナリオ次元

シナリオ次元は、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、シナリオ次元のメンバーの属性を定義できます。

年次元

年次元は、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、年次元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。

期間次元

期間次元は、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、期間次元に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、期間次元に対して年、月および週をサポートします。

エンティティ次元

エンティティ次元は、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。

エンティティ次元は、システムの連結次元です。エンティティ次元の階層は、データのよう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次元で管理されます。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次元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。

値次元

値次元は、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。値次元は、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。

勘定科目次元

勘定科目次元は、通常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、勘定科目次元のメンバーの属性を定義します。

内部取引次元

内部取引次元は、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、勘定科目次元と任意のカスタム次元との組合せに使用される予約次元です。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。

ビュー次元

ビュー次元は、期別、年次累計、四半期累計などの期間単位のカレンダー・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。

カスタム次元

カスタム次元は、勘定科目に関連付けられた次元です。これらの次元を使用すると、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、カスタム次元には、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次元では、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。
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。

要素	最小の長さ	最大の長さ	制限
メタデータ・マネージャ			
メンバーのラベル	1	80	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: <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icolon(;) 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icolon(;) ● スラッシュ(/)

要素	最小の長さ	最大の長さ	制限
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icolon(;)
仕訳の説明	0	255	なし
仕訳グループ	0	30	なし
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	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: <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icolon(;) ● チルダ(~) <p>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。</p>
データ・グリッド			
セルの説明	1	1900	なし

要素	最小の長さ	最大の長さ	制限
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	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: <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(-) ● 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	次の文字は、ドキュメント名には使用できません: <ul style="list-style-type: none"> ● アスタリスク(*) ● アット記号(@) ● 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(/) ● 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: 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

この章の内容

アプリケーションの管理.....	26
アプリケーション・プロファイルの作成.....	27
アプリケーションの作成.....	35
アプリケーションの登録.....	36
アプリケーション・モジュールの構成.....	37
カスタム次元を追加するためのアプリケーションの変更	38
アプリケーションを開く	39
アプリケーションを閉じる.....	39
アプリケーション・サーバー・クラスタの変更	39
アプリケーション・リストの表示	40
アプリケーションの削除.....	40
アプリケーション要素のロード	41
アプリケーション要素の抽出	42
データの抽出のデータベース情報の指定.....	42
クラシック管理アプリケーションのコピー	44
クラシック管理アプリケーションの名前変更	46
Performance Management Architect アプリケーションのコピー	47
サンプル・アプリケーションの使用方法.....	52
システム・メッセージの使用	53
システム・ユーザーの管理.....	55
サーバーとアプリケーションの管理.....	57
タスクの監査.....	58
データの監査.....	61
タスクの実行の監視	62
無効なレコードのスキャンと消去	64

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 Administrator's Guide 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次元で構成されます。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、「次元エディタ」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 User Security Administration Guide 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 Enterprise Performance Management System Installation and Configuration Guide 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 User Security Administration Guide 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。[27 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。[35 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。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、カスタム次元、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。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言語、カレンダー、期間単位、期間およびカスタム次元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。

注： アプリケーション・プロファイルは、Windows の Financial Management Desktop で作成します。

以下のトピックを参照してください。

- [言語の入力](#)
- [カレンダーの定義](#)
- [期間単位の編集](#)
- [期間の編集](#)
- [カスタム次元の作成](#)
- [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 Windows の Financial Management Desktop で、「アプリケーション・プロファイルの定義」を選択します。
- 2 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、 をクリックしてファイルを検索します。

注： 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。

- 3 「次へ」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ラベルには、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。

▶ 言語を指定するには:

1 グリッドに言語を入力し、「[Tab]」キーを押して次の行に移動します。

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 「次へ」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、期間およびビュー次元を使用して設定します。年次元のメンバーは、アプリケーションの開始年と年の合計を定義します。期間次元のメンバー階層は、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。期間次元の最上位レベルの親は、常に[年]メンバーです。ビュー次元は、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。ビュー次元には、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、期間次元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。

▶ カレンダーを設定するには:

1 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 次のいずれかのアクションを行います。

- 「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、カレンダーの最初の月を「開始月」リストから選択します。

- 「カスタム・カレンダー」を選択した場合:
 - 「基本期間数」に、年内の期間数を入力します。
 - 「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。

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。

- 「手動で定義したカレンダー」を選択した場合は、次の手順に進みます。
- 3 「開始年」に、カレンダーの開始年を入力します。
 - 4 「年数」に、含める年数の合計を入力します。
 - 5 「次へ」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。

期間単位を編集しているときは、期間単位 1 の行の最初の列に含まれている年次累計(YTD)の期間単位のラベルを変更できません。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「手動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。期間の次元のレベルごとに、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。

▶ 期間単位を指定するには:

- 1 「言語」リストから言語を選択します。
- 2 「表示」列で値を入力または編集して、ビューの説明を入力します。

期間単位は、6 つまで入力できます。

注： ビューのラベルは、80 文字まで入力できます。ビューの説明は、40 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ビューの説明を含めるようにしてください。

3 「次へ」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。

兄弟期間と子期間を追加する手順については、[31 ページの「兄弟期間と子期間の追加」](#)を参照してください。

期間を選択したら、期間単位を表示して、期間のラベルや説明を変更できます。

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。

▶ 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、「説明」で期間の説明を入力または編集します。

期間の説明には、40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。Year 期間の説明は編集できません。

ヒント： 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。

2 「ラベル」で、期間のラベルを入力または編集します。

期間のラベルには、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。Year 期間のラベルは編集できません。期間のラベルに対して行った変更は、アプリケーション・プロファイルで定義されたすべての言語で反映されます。

ヒント： とを使用して、期間階層のレベルを展開したり、縮小したりできます。

3 「言語」リストから言語を選択します。

言語ごとに一意の期間の説明を作成できます。言語ごとに一意の期間ラベルを作成することはできません。



4 「次へ」をクリックします。

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 期間を右クリックし、「兄弟の挿入」または「子の挿入」を選択します。

ヒント: ツールバーの「子の挿入」ボタン  および「兄弟の挿入」ボタン  を使用することもできます。

- 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 期間を右クリックし、「複数の挿入」を選択します。
- 3 次のいずれかのオプションを選択します:
 - 「複数の兄弟期間」  は、選択した期間に複数の兄弟期間を追加します。
 - 「複数の子期間」  は、選択した期間に複数の子期間を追加します。
- 4 追加する期間の数を入力します。
- 5 新しい期間のラベルの接頭辞を入力します。

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。

期間の移動

マウスを使用して、期間をドラッグ・アンド・ドロップできます。期間は、階層内の兄弟または子の位置に移動できます。

▶ 期間を移動するには:

- 1 移動する期間をハイライトします。
- 2 期間をクリックし、階層内の別の位置にドラッグします。
- 3 次のいずれかのオプションを選択します:
 - ドラッグしている期間を、現在ハイライトされている期間の兄弟にする場合は、「ここ(兄弟)に移動」を選択します。
 - ドラッグしている期間を、現在ハイライトされている期間の子にする場合は、「ここ(子)に移動」を選択します。

注： また、期間を選択した期間の兄弟または子としてコピーすることもできます。マウスのボタンを放したときに、「ここ(兄弟)にコピー」または「ここ(子)にコピー」のオプションを選択します。

期間のコピー

期間の階層に、単一期間や複数期間をコピーできます。選択した期間とそのすべての子期間をコピーできます。コピーした期間を、選択した期間の兄弟または子として貼り付けることができます。

▶ 期間をコピーするには:

- 1 コピーする期間をハイライトします。
- 2 ハイライトした期間を右クリックし、オプションを選択します。
 - ハイライトした期間のみをコピーする場合は、「期間のコピー」を選択します。
 - ハイライトした期間とそのすべての子期間をコピーする場合は、「すべてコピー」を選択します。
- 3 コピーした期間の貼付け先の期間を右クリックし、オプションを選択します。
 - コピーした期間を、選択した期間の兄弟として貼り付ける場合は、「兄弟として貼付け」を選択します。
 - コピーした期間を、選択した期間の子として貼り付ける場合は、「子として貼付け」を選択します。
- 4 「ラベル」で期間ラベルを編集します。


ラベルには、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。
- 5 「説明」に、期間の説明を入力します。

説明には40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。

期間の削除

期間を階層から削除すると、その期間の子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、次のいずれかのアクションを行います。
- 期間を右クリックし、「削除」を選択します。
 - 期間をハイライトし、をクリックします。

カスタム次元の作成

カスタム次元情報は、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。

カスタム次元テーブルを開くと、アプリケーションの最初の2つのカスタム次元が自動的に表示されます。これらのカスタム次元は、通貨レート情報に使用する必要があります。これらの次元のサイズを「大」にする必要があります。1つ目の次元は通貨(元)の情報に、2つ目の次元は通貨(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次元に対して、次元名(短い説明)と次元別名(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次元の他に、追加のカスタム次元を作成できます:

- アプリケーションに、一意のカスタム次元名と次元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 という次元名を使用している場合、次元別名を PROD にはできません。
- 次元名は最大 10 文字です。
- 次元別名は最大 20 文字です。
- 次元名と別名に、スペースや空白を使用することはできません。
- カスタム次元に、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」。

カスタム次元の最大数は、データベース・タイプと各カスタム次元のサイズによって異なります。

- 「大」のカスタム次元は、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。

カスタム次元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次元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つのカスタム次元が必要です。3つ以上のカスタム次元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」の次元数、MD = 「中」の次元数、SD = 「小」の次元数。

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

実用的には、カスタム次元に「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、カスタム次元を「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、次元に 32,536 より多いメンバーを持つ予定である場合のみ、カスタム次元に「大」サイズを使用することをお勧めします。

次の例は、カスタム次元のサンプルを示しています。

カスタム次元名	カスタム次元別名	カスタム次元サイズ	通貨に使用
Prod	Products	大	通貨(元)
Mkt	Markets	大	通貨(先)
Flows	BSFlows	小	N/A
Cust	Customers	大	N/A


▶ カスタム次元を作成するには:

- 1 カスタム次元テーブルから、プラス記号(+)を選択して行を挿入します。
- 2 「カスタム次元名」に、一意の次元名を入力します。
- 3 「カスタム次元別名」に、一意の次元別名を入力します。
- 4 「カスタム次元サイズ」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 追加の次元を作成するには、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

Design Application Profile モジュール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。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するには:

- 1 「ファイル名」で、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。
- 2 「終了」をクリックします。

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 Administrator's Guide 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次元および次元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 User Security Administration Guide 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 Installation and Configuration Guide 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。数字で始めたり、スペースや特殊文字を含むことはできません。5 文字を超える国際文字、およびアンパサンド(&)記号を含むことはできません。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 という拡張子が使用されます。

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 User Security Administration Guide 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、モジュールに関連付けられたセキュリティ役割を持っている必要があります、そうしないと、モジュールはそのユーザーに表示されません。

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、アプリケーションの任意のユーザーに対して、そのモジュールは表示されません。

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く必要があります。

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・モジュールを無効化するには:

- 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。

カスタム次元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次元を追加できます。既存のカスタム次元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： カスタム次元を追加し、アプリケーションを作成または配置した後は、カスタム名または別名を変更できません。

カスタム次元を追加するには、「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。

カスタム次元を追加する前に、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。アプリケーションのコピー・プロセスを使用して、メタデータおよびデータを含む、アプリケーションのバックアップ・コピーを作成します。[44 ページの「クラシック管理アプリケーションのコピー」](#)を参照してください。
- 元のアプリケーションで必要な場合、データ監査情報を復元できるように、データ監査テーブル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、データ監査テーブルを消去します。
- 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ます。
- アプリケーションがアプリケーション・サーバー上で実行されていないことを確認します。

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。

▶ カスタム次元を追加するには：

- 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、追加のカスタム次元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次元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、そこで進行状況を監視したり、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。新規カスタム次元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、「ナビゲート」、「アプリケーション」、「連結」の順に選択して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」テーブルには、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。テーブルには、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 Administrator's Guide 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:

- [72 ページの「アプリケーション・セキュリティのロード」](#)
- [130 ページの「メタデータのロード」](#)
- [146 ページの「メンバー・リストのロード」](#)
- [251 ページの「ルールのロード」](#)
- [37 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。

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:

- [76 ページの「アプリケーション・セキュリティの抽出」](#)
- [134 ページの「メタデータの抽出」](#)
- [147 ページの「メンバー・リストの抽出」](#)
- [253 ページの「ルールの抽出」](#)
- [37 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せん。

▶ 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。

データの抽出のデータベース情報の指定

データベースにデータを抽出する場合、Financial Management 構成ユーティリティを使用して、データベース情報を指定します。



データベースへのデータの抽出機能により、Oracle Essbase を使用して、データを分析し、レポートを作成できます。データ・ソース名(DSN)を使用して、リレーショナル・データベースの宛先を指定します。データソース名およびパスを指定する必要があります。データソース名は、必要に応じて追加、削除または変更できます。

注： この機能は、インストール時に、Financial Management サーバー・コンポーネントをインストールした場合にのみ使用可能です。ユーティリティは、Financial Management\Server フォルダにあります。

複数のサーバーを使用してデータベースにデータを抽出する場合は、各アプリケーション・サーバーで複数のサーバーのサポートを使用可能にし、次の要素が同じになるようにすべてのアプリケーション・サーバーを設定しておく必要があります: クロック設定、データソース、および UDL ファイル。

詳細は、[第 8 章「データベースへのデータの抽出」](#)を参照してください。

▶ データソースの名前を追加するには:

- 1 Financial Management 構成ユーティリティを起動します。
- 2 「拡張分析の DSN」ページから、「DSN の追加」をクリックします。
- 3 「DSN 名」に、データソース名を入力します。
- 4 「DSN パス」にデータソースのパスを入力するか、パスを参照します。
- 5 **オプション:** Financial Management データ・テーブルとインデックスが作成されたデータベース・テーブルスペースを指定するには:
 - 1  (「データ・テーブルスペース」フィールドの隣)をクリックし、データ・テーブルの場所を選択します。
 - 2 「OK」をクリックします。
 - 3  (「インデックス・テーブルスペース」フィールドの隣)をクリックし、インデックスの場所を選択します。

注： Oracle および IBM DB2 SMS では、データ表領域のみを指定します; インデックス表領域は使用しません。

- 4 「OK」をクリックします。
- 6 「OK」をクリックします。

▶ データソース名を変更するには:

- 1 「拡張分析の DSN」ページから、「DSN 情報」で、DSN を選択します。
- 2 「DSN の変更」をクリックし、名前を変更します。
- 3 「OK」をクリックします。

▶ データソースの名前を削除するには:

- 1 「拡張分析の DSN」ページから、「DSN 情報」で、DSN を選択します。
- 2 「DSN の削除」をクリックし、「OK」をクリックします。

クラシック管理アプリケーションのコピー

Financial Management は、同じサーバーまたは別のアプリケーション・サーバーにアプリケーションをコピーおよび移動するためのユーティリティを提供します。アプリケーションは、データベース間でコピーできます。たとえば、Microsoft SQL データベースで作成されたアプリケーションを Oracle データベースにコピーできます。

アプリケーションのコピー・ユーティリティ HFMCopyApplication.exe または HFMCopyApplication_x64.exe (64 ビット・サーバー用)は、次のデフォルトの場所にインストールされています:

```
%Oracle_Home%\FinancialManagement\Utilities
```

注: アプリケーションをコピーする前に、コピーするアプリケーションからすべてのユーザーが切断され、すべてのプロセスが停止していることを確認します。そうでない場合は、コピーされるアプリケーションに正しくないデータが含まれる可能性があります。

ソースまたは宛先のデータベースに接続するには、次のいずれかの方法を使用できます:

- 構成済の Financial Management データベース接続を使用します。この場合、Financial Management アプリケーション・サーバー・コンポーネントがインストールおよび構成されたマシン上で、アプリケーションのコピー・ユーティリティを実行する必要があります。
- UDL ファイルを使用してデータベース接続パラメータを定義します
- データベース接続情報を直接入力します

接続情報を入力または選択した後で、データベース接続をテストできます。コピー処理中にメタデータ階層を確認するよう選択することもできます。システムは、シナリオ、エンティティ、勘定科目、値およびカスタムのメタデータ次元テーブルをスキャンして、問題が存在しないことを確認します。問題が検出された場合、メタデータ検証タスクは完了せず、エラーがログ・ファイルに記録されます。この場合、Oracle サポートに連絡し、ログ・ファイルのコピーを提供してください。

アプリケーションをコピーしたら、新規アプリケーションを Shared Services に登録し、新規アプリケーションのアプリケーション管理者の役割でユーザーを設定する必要があります。Oracle Enterprise Performance Management System User Security Administration Guide を参照してください。

▶ アプリケーションをコピーするには:

- 1 Windows のエクスプローラで、Financial Management のインストール先のユーティリティ・ディレクトリに移動し、HFMCopyApplication.exe または HFMCopyApplication_x64.exe をダブルクリックします。
- 2 ようこそ画面で「次へ」をクリックします。

- 3 タスクの選択ページで、「HFM アプリケーションのコピー」を選択し、「次」をクリックします。
- 4 次のデータベース接続オプションを選択します:
 - 「構成済の HFM データベース接続を使用」
 - 「UDL ファイルを使用」(その後、ファイル名およびパスを入力します)
 - 「接続情報の入力」(その後、データベース情報を入力し、「OK」をクリックします。)
- 5 「接続のテスト」をクリックして、選択した接続情報をテストし、「次」をクリックします。
- 6 ソース・アプリケーションのリストで、コピーするアプリケーションを選択し、「次」をクリックします。
- 7 ターゲット・アプリケーションの名前を入力するか、以前にコピーしたアプリケーションから名前を選択し、「次」をクリックします。
- 8 「オプション」パネルで、任意のアクションまたはすべてのアクションを選択し、「次へ」をクリックします。
 - 「アプリケーション・データのコピー」: ソース・アプリケーションから宛先アプリケーションにデータをコピーします。これにより、ソース・アプリケーションと同一のコピーが生成されます。
 - 「アプリケーション・データ・テーブルのコピー」: ソース・アプリケーションから宛先アプリケーションにアプリケーション・データ・テーブルをコピーします。
 - 「監査データのコピー」: ソース・アプリケーションから宛先アプリケーションにタスクおよびデータ監査データをコピーします。
 - 「クラスタ設定のコピー」: ソース・アプリケーションから宛先アプリケーションにクラスタ情報をコピーします。
 - 「メタデータ階層の検証」: メタデータ次元階層テーブルで可能性のあるエラーを検出します。
 - 「既存のアプリケーションの上書き(存在する場合)」: 宛先アプリケーションと同じ名前のアプリケーションを上書きします。
 - コピーする前にすべてのアプリケーション・テーブルを削除します
 - コピーされるテーブルのみを削除します。
- 9 オプション: 「詳細設定オプション」をクリックして追加のオプションを使用します。
 1. 「詳細設定オプション」タブで、必要に応じて次の値を変更します:
 - クライアント側カーソルの使用
 - サーバー側カーソルの使用
 - SQL バインドの使用
 - デフォルトのスレッド・カウントを使用
 - カスタムのスレッド・カウントを使用
 - SQL エラーのログ

- タスク再試行の回数
2. 「データ・オプション」タブで、必要に応じて次の値を変更します:
 - 年オプション
 - シナリオ・オプション
 - データ・オプション
 - 無効なレコード・オプション
 3. 「データベース・オプション」タブで、必要に応じて次の値を変更します:
 - データベース・テーブルスペースの指定
 - データ・テーブルスペース
 - インデックス・テーブルスペース
 4. 「OK」をクリックして、「詳細設定オプション」ボックスを閉じます。
- 10** 設定を確認し、「次へ」をクリックします。
- 11** 処理が完了したら、「OK」をクリックします。
- 12** 完了したタスクまたは失敗したタスク(存在する場合)のいずれかでタスクの詳細を表示するには、タスクをダブルクリックするか、タスクを選択して疑問符(?)アイコンをクリックします。
- 13** 「閉じる」をクリックします。
- 14** 「終了」をクリックして終了するか、「ログ・ファイルの表示」をクリックしてログ・ファイルを表示し、コピー操作時に発生したエラーを確認します。
- 注：** 正しくない大きさの Oracle インスタンスを使用すると、ユーティリティですべてのテーブル行のコピーが失敗する可能性があります。エラー・ログに Oracle のエラーが表示されるか、テーブル行が失われている場合は、REDO ログのサイズを大きくする必要があります。
- 15** アプリケーションをコピーしたら、新規アプリケーションを Oracle Hyperion Shared Services に登録し、ユーザーに新規アプリケーションの管理者権限を設定します。

クラシック管理アプリケーションの名前変更

アプリケーションのコピー・ユーティリティを使用して、クラシック管理アプリケーションの名前を変更できます。このプロセス中に、コピーは発生しません。アプリケーション・データベース・テーブルおよびデータ・エント리는、新しいアプリケーション接頭辞に名前変更されます。

注： アプリケーションの名前を既存のアプリケーション名に変更することはできません。

▶ アプリケーションの名前を変更するには:

- 1 Windows のエクスプローラで、Financial Management のインストール先のユーティリティ・ディレクトリに移動し、HFMCopyApplication.exe または HFMCopyApplication_x64.exe をダブルクリックします。

- 2 ようこそ画面で「次へ」をクリックします。
- 3 タスクの選択ページで、「HFM アプリケーションの名前変更」を選択し、「次」をクリックします。
- 4 次のデータベース接続オプションを選択します:
 - 「構成済の HFM データベース接続を使用」
 - 「UDL ファイルを使用」 (その後、ファイル名およびパスを入力します)
 - 「接続情報の入力」 (その後、データベース情報を入力し、「OK」をクリックします。)
- 5 「接続のテスト」をクリックして、選択した接続情報をテストし、「次」をクリックします。
- 6 ソース・アプリケーションのリストで、コピーするアプリケーションを選択し、「次」をクリックします。
- 7 ターゲット・アプリケーションの名前を入力するか、以前にコピーしたアプリケーションから名前を選択し、「次」をクリックします。
- 8 設定を確認し、「次へ」をクリックします。

Performance Management Architect アプリケーションのコピー

Performance Management Architect アプリケーションは、アプリケーション・サーバー間でコピーできます。コピー処理では、ソースからターゲット環境に Performance Management Architect のメタデータが移行され、Performance Management Architect の次元ライブラリで使用可能になります。

ターゲット・アプリケーションを作成することも、ターゲット・アプリケーションを置換することも可能です。配置済または未配置のアプリケーションをコピーすると、ソース・アプリケーション・ステータスに基づいて、コピーされるターゲット・アプリケーションが同じステータスで作成されます。コピー処理が完了したら、新しい環境の Shared Services にアプリケーションを登録します。

アプリケーションのコピー・ユーティリティである HFMCopyApplication.exe または HFMCopyApplication_x64.exe は、Financial Management インストールのユーティリティ・ディレクトリにデフォルトでインストールされます。

アプリケーションをコピーするには、アプリケーション管理者のセキュリティ役割が必要です。

コピー処理は、アプリケーションのコピー方法や、ソースとターゲットの環境など、可能性のある次のシナリオによって異なります。

- クラシック・アプリケーションとしてコピーする場合。ソース・アプリケーションは、ターゲット宛先に存在する場合と存在しない場合があります。[48 ページの「クラシック・アプリケーションとしてのコピー」](#)を参照してください。
- Performance Management Architect アプリケーションとしてコピーする場合。ソース・アプリケーションは、ターゲット宛先に存在します。[50 ページの](#)

「Performance Management Architect としてのコピー(ターゲットにソース・アプリケーションが存在する場合)」を参照してください。

クラシック・アプリケーションとしてのコピー

▶ アプリケーションをコピーするには:


- 1 ターゲット宛先にソース・アプリケーションが存在する場合は、ターゲット環境にログオンし、Performance Management Architect 内からアプリケーションを削除します。

ターゲット宛先にソース・アプリケーションが存在しない場合は、手順2に進みます。

- 2 ソースおよび宛先の Financial Management サーバーが実行中でないことを確認します。
- 3 アプリケーションのコピー・ユーティリティがインストールされているディレクトリに移動し、HFMCopyApplication.exe または HFMCopyApplication_x64.exe をダブルクリックします。

注: デフォルトの場所は、Financial Management インストールのユーティリティ・ディレクトリです。


- 4 「ようこそ」画面で、「次へ」をクリックします。

- 5  をクリックして、コピーするアプリケーションのソース・データベースの接続ファイルを検索し、「次へ」をクリックします。

ソース・データベース接続の検証が成功すると、ユーティリティにより、ソース・アプリケーションが表示されます。

- 6 アプリケーションのリストから、コピーするアプリケーションを選択し、「次へ」をクリックします。
- 7 「これは、EPM Architect アプリケーションです。クラシック管理アプリケーションに変換しますか?」というプロンプトで、「はい」をクリックします。
- 8 選択したアプリケーションがターゲット宛先に現在存在する場合は、コピーする前に削除する必要があるという警告のメッセージが表示されます。次のいずれかのアクションを選択します:

- 既存のアプリケーションを削除していない場合は、「いいえ」をクリックして削除します。
- ターゲット環境にアプリケーションが存在しないか、すでに削除してある場合は、「はい」をクリックして続行します。

- 9  をクリックして、宛先データベースの接続ファイル(UDL)を検索し、「次へ」をクリックします。

- 10 アプリケーションのリストで、コピー先アプリケーションを選択するか、新規アプリケーションの名前を入力して、「次へ」をクリックします。

- 11 「オプション」でオプションを選択して、「次へ」をクリックします。

- **アプリケーション・データのコピー:** ソース・アプリケーションから宛先アプリケーションにデータをコピーします。これにより、ソース・アプリケーションと同一のコピーが生成されます。
 - **監査データのコピー:** ソース・アプリケーションから宛先アプリケーションにタスクおよびデータ監査データをコピーします。
 - **「クラスタ設定のコピー」:** ソース・アプリケーションから宛先アプリケーションにクラスタ情報をコピーします。
 - **「既存のアプリケーションの上書き」:** 宛先アプリケーションと同じ名前のアプリケーション(存在する場合)を上書きします。
 - コピーする前にすべてのアプリケーション・テーブルを削除します。
 - コピーされるテーブルのみを削除します。
 - **オプション:** 「詳細設定オプション」をクリックして追加のオプションを使用します。
 1. 必要に応じて次の値を変更します。
 - クライアント側のカーソルを使用するか、サーバー側のカーソルを使用します。
 - SQL バインドの使用
 - デフォルトのスレッド・カウントを使用するか、カスタムのスレッド・カウントを使用します。
 - SQL エラーのログ
 - タスク再試行の回数
 2. 「OK」をクリックして、「詳細設定オプション」ボックスを閉じます。
- 12** 設定を確認し、「次へ」をクリックします。
- 13** コピー処理が完了したら、「OK」をクリックします。
- 14** 完了したタスクまたは失敗したタスク(存在する場合)でタスクの詳細を表示するには、タスクをダブルクリックするか、タスクを選択して疑問符(?)アイコンをクリックします。
- 15** 「閉じる」をクリックします。
- 16** 「終了」をクリックして終了するか、「ログ・ファイルの表示」をクリックしてログ・ファイルを表示し、コピー操作時に発生したエラーを確認します。
- 注:** 正しくない大きさの Oracle インスタンスを使用すると、ユーティリティですべてのテーブル行のコピーが失敗する可能性があります。エラー・ログに Oracle のエラーが表示されるか、テーブル行が失われている場合は、REDO ログのサイズを大きくする必要があります。
- 17** ターゲットの Performance Management Architect のアプリケーション・ライブラリにログオンします。
- 18** 新しくコピーされたアプリケーションを選択し、「アプリケーションのアップグレード」を選択してアプリケーションをアップグレードします。

Performance Management Architect としてのコピー (ターゲットにソース・アプリケーションが存在する 場合)

Performance Management Architect アプリケーションをコピーする場合は、ライフサイクル管理(LCM)を使用して、次元、セキュリティ、ルール、ドキュメント、Web フォーム、Web グリッド、仕訳帳およびメンバー・リストなどの Financial Management アーチファクトを移行します。

Performance Management Architect アプリケーション全体にデータを移行する場合は、データを手動で移動する必要があります。

▶ アプリケーションをコピーするには:


- 1 Shared Services Console から、LCM アーチファクトのリストに移動し、「ファイル・システムへのエクスポート」を選択します。
- 2 LCM 移行ウィザードを使用してオプションを選択し、ファイル・システムにファイルをエクスポートします。選択したすべてのアーチファクトとともに、移行定義の XML が指定されたファイル・システムに抽出されます。
- 3 「マージ」または「アーチファクトの置換」オプション(次元にのみ適用可能)を選択して、移行定義ファイルをファイル・システムから該当のアプリケーションにインポートします。

「マージ」オプションを使用すると、ターゲット宛先に存在するソース以外のすべてのアーチファクトを、ターゲット・アプリケーションに含めることができます。


「置換」オプションでは、ターゲット・アプリケーションがソースのコピーとして作成されますが、それらの共有次元を使用するターゲットの Performance Management Architect システムのその他のアプリケーションに影響を与える可能性があります。

- 4 インポート処理が正常に終了すると、ターゲット・アプリケーションがソース・アプリケーションのアーチファクトで更新されます。これで、アプリケーションのコピー・プロセスを開始できます。
- 5 ソースおよび宛先の Financial Management サーバーが実行中でないことを確認します。
- 6 アプリケーションのコピー・ユーティリティがインストールされているディレクトリに移動し、HFMCopyApplication.exe または HFMCopyApplication_x64.exe をダブルクリックします。

注: デフォルトの場所は、Financial Management インストールのユーティリティ・ディレクトリです。

- 7 「よろこそ」画面で、「次へ」をクリックします。
- 8  をクリックして、コピーするアプリケーションのソース・データベースの接続ファイルを検索し、「次へ」をクリックします。

ソース・データベース接続の検証が成功すると、ユーティリティにより、ソース・アプリケーションが表示されます。

- 9 アプリケーションのリストから、コピーするアプリケーションを選択し、「次へ」をクリックします。
- 10 「これは、EPM Architect アプリケーションです。クラシック管理アプリケーションに変換しますか?」というプロンプトで、「いいえ」をクリックします。
- 11 アプリケーションをコピーする前に、アプリケーションに関連するアーチファクトの LCM 移行を実行する必要があるという警告のメッセージが表示されます。次のいずれかのアクションを実行します:
 - LCM 移行を実行していない場合は、「いいえ」をクリックして、その手順を完了します。
 - LCM 移行を実行済の場合は、「はい」をクリックして続行します。
- 12  をクリックして、宛先データベースの接続ファイル(UDL)を検索し、「次へ」をクリックします。
- 13 アプリケーションのリストで、コピー先のアプリケーション名を選択するか、新しいアプリケーションの名前を入力して、「次へ」をクリックします。
- 14 「オプション」パネルでオプションを選択し、「次へ」をクリックします。
 - **アプリケーション・データのコピー:** ソース・アプリケーションから宛先アプリケーションにデータをコピーします。これにより、ソース・アプリケーションと同一のコピーが生成されます。
 - **監査データのコピー:** ソース・アプリケーションから宛先アプリケーションにタスクおよびデータ監査データをコピーします。
 - **「クラスタ設定のコピー」:** ソース・アプリケーションから宛先アプリケーションにクラスタ情報をコピーします。
 - **「既存のアプリケーションの上書き」:** 宛先アプリケーションと同じ名前のアプリケーション(存在する場合)を上書きします。
 - コピーする前にすべてのアプリケーション・テーブルを削除します。
 - コピーされるテーブルのみを削除します。
 - **オプション:** 「詳細設定オプション」をクリックして追加のオプションを使用します。
 1. 必要に応じて次の値を変更します。
 - クライアント側のカーソルを使用するか、サーバー側のカーソルを使用します。
 - SQL バインドの使用
 - デフォルトのスレッド・カウントを使用するか、カスタムのスレッド・カウントを使用します。
 - SQL エラーのログ
 - タスク再試行の回数
 2. 「OK」をクリックして、「詳細設定オプション」ボックスを閉じます。
- 15 設定を確認し、「次へ」をクリックします。

- 16 コピー処理が完了したら、「OK」をクリックします。
- 17 完了したタスクまたは失敗したタスク(存在する場合)のタスクの詳細を表示するには、タスクをダブルクリックするか、タスクを選択して疑問符(?)アイコンをクリックします。
- 18 「閉じる」をクリックします。
- 19 「終了」をクリックして終了するか、「ログ・ファイルの表示」をクリックしてログ・ファイルを表示し、コピー操作時に発生したエラーを確認します。

注： 正しくない大きさの Oracle インスタンスを使用すると、ユーティリティですべてのテーブル行のコピーが失敗する可能性があります。エラー・ログに Oracle のエラーが表示されるか、テーブル行が失われている場合は、REDO ログのサイズを大きくする必要があります。

- 20 ターゲットの Performance Management Architect のアプリケーション・ライブラリにログオンします。
- 21 新しく更新されたアプリケーションのステータスが、配置されたアプリケーションと同期していませんになっています。アプリケーションを再配置します。

コピー処理が成功した場合、ターゲット・アプリケーションは、対応するすべてのアーティファクトとアプリケーション・データを含むソース・アプリケーションのコピーになります。LCM 移行中に「マージ」オプションを選択した場合は、アプリケーションに新しいメタデータが含まれています。「置換」オプションを選択した場合、ターゲット・アプリケーションはソース・アプリケーションと同一です。

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。

インストール・プロセス中にサンプル・アプリケーション・コンポーネントをインストールした場合、ファイルが ZIP ファイルとしてインストールされ、Financial Management\SampleApps, and Starter Kits\SampleApps などの Financial Management をインストールしたディレクトリのサンプル・アプリケーション・フォルダに配置されます。使用するサンプル・アプリケーションのみ解凍できます。

サンプル・ファイルをインストールしていない場合は、Financial Management を再インストールし、「サンプル・アプリケーション」コンポーネントを選択することで取得できます。

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)	次元のメンバー・リスト

サンプル・ファイル	コンテンツ
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
データ(.dat)	該当するシナリオ、年、データ
ルール(.rle)	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립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 User Security Administration Guide を参照してください。

次の手順を参照してください:

- [53 ページの「システム・メッセージの表示と印刷」](#)
- [54 ページの「システム・メッセージの削除」](#)

システム・メッセージの表示と印刷

▶ システム・メッセージを表示して印刷するには:

- 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 オプション: メッセージを印刷するには、「システム・メッセージの詳細」ページで「印刷」をクリックします。
 - 7 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
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。55 ページの「ユーザーの表示」および56 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 User Security Administration Guide 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。[57 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。

57 ページの「接続の使用不可と使用可能」および 56 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 User Security Administration Guide 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。

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
- データの消去
- 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)テーブルに保管されます。テーブル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、テーブルを消去します。ログ・サイズを監視して定期的に消去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。
- 6 **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: ログを消去すると、消去処理の記録がログに残り、これを消去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、ユーザーおよび次元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す。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
- データの消去
- 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(POV)
- 視点(POV)に入力された値

データ監査ログ情報は `APPNAME_DATA_AUDIT` テーブルに保管されます。テーブル内の情報をバックアップまたは抽出できます。ログ・サイズを監視して定期的に消去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(POV)」から、次元メンバーを選択する次元をクリックします。
- 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: ログを消去すると、タスク監査ログに消去処理の記録が残り、これを消去することはできません。

タスクの実行の監視

「タスクを実行します」機能を使用して、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。

注: 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。

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エクスポート
- 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タスク・ステータス・タイプ別にフィルタ処理できます。

- 初期化しています
- 実行しています
- 一時停止しました
- 未初期化しています
- 停止済
- 中止済
- 完了
- 応答なし
- スケジュールにより停止
- スケジュールにより開始

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 レジストリ・キーを変更することによって、デフォルトを変更できます。

▶ 実行中のタスクを監視および中止するには:

1 次の手順のいずれかを行います:

- 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

3 タスクを中止するには、タスクを選択して「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。

ヒント: タスク・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。

無効なレコードのスキャンと消去

無効なレコードの消去機能を使用して、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。無効なレコードを消去するには、管理者のセキュリティ・アクセス権が必要です。

無効なレコードの削除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、「無効なレコードの消去」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。
- 「無効なレコードの消去」をクリックして、レコードを消去します。

3

アプリケーション・セキュリティの管理

この章の内容

アプリケーション・セキュリティに関する考慮事項.....	66
Financial Management からの Shared Services Console の起動.....	66
アプリケーション・セキュリティのロード.....	72
アプリケーション・セキュリティの抽出.....	76

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。72 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。66 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 User Security Administration Guide 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。
次元エディタ	手動で次元を作成および管理するだけでなく、次元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。

セキュリティ役割	説明
Financial Management アプリケーション作成者/Financial Management マネージャ	<p>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次元しか変更できません。</p> <p>「アプリケーション作成者」の役割を持つユーザーが Oracle Hyperion EPM Architect からアプリケーションを配置する場合、その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。72 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76 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、35 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 Administrator's Guide 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 User Security Administration Guide 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 Financial Management から起動するには、「ナビゲート」、「管理」、「Shared Services Console」の順に選択します。

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 User Security Administration Guide 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。テーブル内で、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、テーブル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。

Financial Management のアプリケーションのセキュリティ・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 Administrator's Guide を参照してください。

次の手順を実行できます:

- [68 ページの「セキュリティ・クラスの作成」](#)
- [69 ページの「セキュリティ・クラスの削除」](#)
- [69 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(+)
- セミコロン(;))
- 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・クラス次元から削除できます。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、テーブル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テーブルの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: 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、テーブル内のセルを選択し、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。

Financial Management アプリケーションに対するセキュリティ・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 User Security Administration Guide 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消去するには、「すべて選択解除」を選択します。

注意 「すべて選択解除」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、消去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 User Security Administration Guide を参照してください。消去オプションを選択する前に、74 ページの「セキュリティ情報の消去とロード」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消去とロード

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消去して、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消去して、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。74 ページの「セキュリティ情報を消去する前に行っておくべきタスク」を参照してください。
 - 2 既存のセキュリティ情報を消去して、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。75 ページの「セキュリティ情報を消去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消去する前に行っておくべきタスク

セキュリティ情報を消去して、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消去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消去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消去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消去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消去した後で行うタスク

セキュリティ情報を消去して、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

- ▶ セキュリティ情報を消去してロードした後にメタデータを更新するには、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消去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消去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消去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消去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消去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消去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消去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消去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 Installation and Configuration Guide 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

この章の内容

勘定科目の定義	80
カスタム・メンバーの定義.....	84
カスタム次元情報の追加.....	86
エンティティ・メンバーの定義	87
シナリオ・メンバーの定義.....	89
アプリケーション設定の定義	91
期間別組織.....	94
連結メソッドの定義	94
通貨の定義.....	98
セル・テキスト・ラベルの定義	100
システム生成の勘定科目	102
内部取引パートナーの設定.....	104
セキュリティに基づくメタデータのフィルタ	106
APP フォーマットのメタデータ・ファイルの作成.....	106
メタデータ・マネージャのビューの使用方法	120
メタデータの参照整合性.....	129
メタデータのロード.....	130
メタデータ・ロードの変更の表示.....	133
メタデータの抽出	134

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 Administrator's Guide を参照してください。

メタデータは、次元名、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

勘定科目次元は、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、[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	<p>(必須)次のいずれかの値です。</p> <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。 <p>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。</p> <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 of 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。 <p>82 ページの「勘定科目の種別の動作」を参照してください。</p>
CalcAttribute	<p>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</p> <p>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。</p>
CustomTop	<p>カスタム次元の階層のどの最上位メンバーが、この勘定科目で有効かどうかを指定します。</p> <p>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： 属性の数は、アプリケーションに定義されたカスタム次元の数に基づいています。カスタム次元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次元別名で置換されます。</p>
DefaultParent	<p>勘定科目のデフォルトの親です。</p>

属性	説明
Description	勘定科目の説明です。 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slash)は使用できません。
EnableCustomAggr	カスタム次元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次元で集約できるようにする場合はYを指定し、集約できるようにしない場合は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次元の数に基づいています。カスタム次元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次元別名で置換されます。
EnableDataAudit	勘定科目を監査できるかどうかを指定します。 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。
ICPTopMember	勘定科目の内部取引パートナーの最上位メンバー 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。エンティティ次元のその他のメンバーすべてが勘定科目で無効です。
IsCalculated	勘定科目を計算できるかどうかを指定します。 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(") ● 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(/)

属性	説明
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。
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 Ent	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。
XBRL Tags	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、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、カスタム次元、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

勘定科目の種別	親勘定科目									
	A	L	R	E	F	B	BR	C	G	D
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
GROUPLABEL(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

カスタム次元は勘定科目次元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。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	カスタム次元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	<p>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。</p> <p>注： カスタム次元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。</p> <p>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:</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(/)
SecurityClass	カスタム次元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。
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>

属性	説明
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>次元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>

カスタム次元情報の追加

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、カスタム次元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次元情報をインポート
- メタデータ・マネージャでカスタム次元情報を手動で編集

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次元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。

カスタム次元情報のインポート

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次元情報をメタデータ・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次元を選択できます。メタデータ次元リストには、カスタム次元の次元別名(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次元に対する追加、削除または並替えが行われません。

必要な場合には、カスタム次元情報をインポートする前に、メタデータ・ファイルの既存の次元を変更してください。カスタム次元別名は、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次元を削除することも可能です。グリッドからカスタム次元を削除する際には、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： カスタム次元テーブルの最初の2つのエント리는削除できません。既存のエント리를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、カスタム次元を追加できます。新しい次元は、メタデータ・リストの最後に追加されます。リストの途中で次元を挿入するには、追加してから並べ替える必要があります。

▶ カスタム次元を編集するには:

- 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:** カスタム次元の名前の変更、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次元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。

カスタム次元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次元をインポートするかわりに、手動でカスタム次元を追加できます。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次元情報を変更できます。

これまでにカスタム次元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次元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次元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、テーブル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、カスタム次元別名の情報は任意の順序で入力できます。

▶ カスタム次元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、カスタム次元別名を入力します。
- 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。エンティティ次元に対して連結を実行すると、次元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。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。
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 次元のメンバーとして自動的に表示されます。
Member	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。 <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(/) 注： エンティティ名を「ALL」にすることはできません。
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。

属性	説明
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

シナリオ次元は、予算、実績、予測などの一連の関連データを表します。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(POV)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。

属性	説明
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。
EnableDataAudit	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。 <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	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。
Member	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。 <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	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。
SecurityClass	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。
SupportsProcessManagement	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。 <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。

属性	説明
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： 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： 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： 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?
- どの次元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?

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: 値次元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次元メンバーを表示するか、アクセス権を持っている次元メンバーのみを表示するかを指定します。次の次元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 ● カスタム <p>ユーザーがアクセス権を持っていない次元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>
FDMAAppName	Oracle Hyperion Financial Data Quality Management アプリケーションの名前
ICPEntitiesAggregationWeight	値次元の[ICP Top]メンバーに集約する内部取引パートナー・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。

属性	説明
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次元の数に基づいています。カスタム次元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次元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次元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次元を保護する場合は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次元の数に基づいています。カスタム次元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次元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。

属性	説明
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。 有効な値はYまたはNです。デフォルトはNです。
ValidationAccount	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と、カスタム次元の[ICPTop]との有効な交差を持っている必要があります。 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。

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 次元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、特に指定がない限り、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

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。</p> <p>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({}) ● 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
説明	<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: POWN および POWNMIN の詳細は、97 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: 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。</p>
UsedByCalcRoutine	<p>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。</p>

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)次元の[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 User's Guide](#) 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 次元に保管されます。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・テーブル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・テーブルの 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次元が必要です。通貨次元には、エンティティ次元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。通貨次元に追加された各通貨は、システム生成メンバーとして値次元に表示されます。値次元から通貨を選択し、通貨に換算されたデータ値を表示できます。

通貨次元種別を使用して次元を作成し、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次元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。</p> <p>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。</p> <p>また、為替レートの入力方法も決定します。たとえば、データが千の単位に調整される場合、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 ● 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)次元に使用できます。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(POV)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 User's Guide 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

表 14 に記載されているシステム勘定科目は、エンティティ次元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。
[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づ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。
メソッド	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

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。 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)次元全体で管理されます。ICP 次元は、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、勘定科目次元およびカスタム次元とともに使用する予約済の次元です。

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]値次元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]値次元の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せん。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。[87 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示します。
[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 次元のメンバーの編集

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 つの値次元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、次の値次元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:

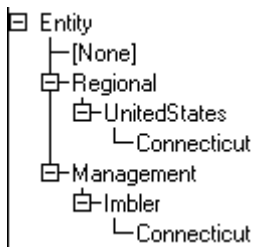
- 1 メタデータ・マネージャで、システム生成の値次元メンバーが含まれているファイルを開きます。
- 2 「メンバー属性」タブで、階層内のメンバーを選択し、説明属性を変更します。
- 3 必要に応じて、手順2を繰り返し、その他の値メンバーの説明を追加します。
- 4 「ファイルの保存」をクリックします。

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次元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
- エンティティ次元
- シナリオ次元
- カスタム次元
- 勘定科目次元

- 値次元
- 内部取引パートナ次元
- アプリケーション設定
- 連結メソッド

各種の次元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： カスタム次元の場合、これらのセクションは、カスタム次元の別名を使用する必要があります。

エンティティ次元、シナリオ次元、勘定科目次元およびカスタム次元のメタデータは、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。カスタム次元に次元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す。

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、その後に、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
- 次元
- 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 = 3.4  
  
!FILE_FORMAT = 3.40  
  
!FILE_FORMAT = 03.40  
  
!FILE_FORMAT = 03.4
```

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 = 3.5.365
```

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、[91 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
```

[98 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、次元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、次元メンバーを入力します。

```
!MEMBERS=Customers
```

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

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

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

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

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

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

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

注： カスタム次元の場合、このセクションは、カスタム次元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、カスタム、値およびICPの各次元のメンバー・セクションのフォーマットについて説明します。

勘定科目

勘定科目次元メンバーの構文:

```
'Label, AccountType, IsCalculated, IsConsolidated, IsICP, PlugAcct, CustomTop,
NumDecimalPlaces, UsesLineItems, EnableCustomAggr, UserDefined1, UserDefined2,
UserDefined3, XBRLTags, SecurityClass, ICPTopMember, EnableDataAudit, DefaultParent,
Descriptions
```

[80 ページの「勘定科目の定義」](#)を参照してください。

`Custom_Order` キーワードは、カスタム次元の列を定義するために、すべての新規メタデータ・ファイルに必要です。このセクションは、アプリケーション・プロファイル(.per)ファイルのカスタム順序と一致している必要があります。

例:

```
!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

シナリオ次元メンバーの構文:

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

[89 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

エンティティ次元メンバーの構文:

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

[87 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;;;;;DefaultParent=Regional
```

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

カスタム

カスタム次元メンバーの構文:

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

[84 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 次元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;;;;;DefaultParent=#root
```

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

```
Customer2;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、値次元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次元のメンバーの場合、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
```

システム定義の値次元の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次元の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、これらの値次元の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

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、値次元の 3 つのメンバーを示します。

表 17 値次元の説明

値メンバー	説明
USD Total	US ドル合計
USD Adjs	US ドル調整
USD	US 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)次元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
```

[94 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： カスタム次元のノード・レコードには、3番目の属性が含まれていません。119 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。<>の中に実際の次元名を入力してください。

```
!HIERARCHIES=<>
```

カスタム次元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 次元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： カスタム次元の場合、このセクションは、カスタム次元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次元の階層セクションのフォーマットを説明します。

勘定科目次元の階層

勘定科目次元の階層の構文:

```
parentmemberlabel  
;  
childmemberlabel
```

この例では、勘定科目次元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
```

シナリオ次元の階層

シナリオ次元の階層の構文:

```
parentmemberlabel  
;  
childmemberlabel
```

この例では、シナリオ次元の階層を指定します。

```
!HIERARCHIES=Scenario
```

```
;Actual
```

```
;Budget
```

エンティティ次元の階層

エンティティ次元の階層の構文:

```
parentmemberlabel  
;  
childmemberlabel
```

この例では、エンティティ次元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

カスタム次元の階層の構文:

```
parentmemberlabel  
;  
childmemberlabel  
;  
AggregationWeight
```

この例は、カスタム次元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  
  
Golf;GolfClubs;1
```

メタデータ・ファイルに含まれない次元

年、期間およびビューの各次元は、メタデータ・ファイルには含まれません。これらの次元は、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次元メンバーは、そのほとんどがシステムで定義されます。ただし、値次元のメンバーの説明、および内部取引パートナー次元のセキュリティ・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次元の次の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 により次の値次元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: この次元は自動生成されます。内部取引パートナ次元のメンバーは、IsICP 属性が TRUE に設定されているエンティティ次元のメンバーごとに生成されます。115 ページの「内部取引パートナ」で説明されているように、いくつかの内部取引パートナ・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、次元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。

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。

タブ	説明
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。[122 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。[126 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:

- 122 ページの「メンバーの追加と変更」
- 123 ページの「カスタム次元のノード属性の変更」
- 123 ページの「メンバー・リストからのメンバーの追加」
- 124 ページの「メンバーの移動」
- 124 ページの「メンバーのレベルを上げる」
- 124 ページの「メンバーの位置の変更」
- 125 ページの「メンバーの削除と除去」
- 125 ページの「孤児メンバーの追加」
- 125 ページの「孤児メンバーの除去」
- 126 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、このメンバーは最上位メンバー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、デフォルトで最上位メンバーが「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。最上位のメンバー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、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	カスタム次元に使用します。このタブには、現在選択されているカスタム次元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・リストから次元を選択します。

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、階層内の最上位のメンバーのみです。

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 メンバーをすべて追加または更新するまで、手順2および3を繰り返します。

カスタム次元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。[84 ページの「カスタム・メンバーの定義」](#)を参照してください。

カスタム次元の集約加重には、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次元を選択します。
- 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・リストから次元を選択します。
- 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・リストから次元を選択します。
- 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・リストから次元を選択します。
- 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・リストから次元を選択します。
- 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す。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・リストから次元を選択します。
- 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。
 - **次元から削除:** メンバーは次元から削除されます。

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・リストから次元を選択します。
- 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、次元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・リストから次元を選択します。
- 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、選択した次元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・リストから次元を選択します。
- 2 階層内のメンバーを選択し、タスクを実行します:
 -  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。
 -  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:

[127 ページの「メンバーの追加と変更」](#)

[127 ページの「メンバーのコピー、切り取りおよび貼付け」](#)


[128 ページの「メンバーの削除」](#)

[128 ページの「メタデータの検証」](#)

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・リストから次元を選択します。
- 2 次のいずれかのアクションを行います:
 - 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。
 - メンバーを追加するには、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 メンバーの変更および追加がすべて完了するまで、手順2を繰り返します。

メンバーのコピー、切取りおよび貼付け




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・リストから次元を選択します。
- 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せん。

▶ 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・リストから次元を選択します。
- 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・リストから次元を選択します。
- 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・リストから次元を選択します。
- 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・リストから次元を選択します。
- 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 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。

注： メタデータをロードするときは、必ず「整合性の確認」オプションを選択してください。

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(POV)についても確認されます。次元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::2012  
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(POV)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 2012、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消去することができ、データの整合性も確認できます。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	次元メンバーがロード・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次元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ます。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次元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ます。ロード方法として置換を選択して、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、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消去	アプリケーション・データベース内のすべての次元メンバーと、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。129 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消去」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。[130 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、勘定科目次元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
- Performance Management 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
- カスタム次元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。

値次元などのシステム定義の次元のメンバーは、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次元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できません。

- 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

この章の内容

メンバー・リスト・ファイルの作成.....	138
メンバー・リストのロード.....	146
メンバー・リストの抽出.....	147
システム生成のリストと次元.....	148

メンバー・リストを使用すると、次元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、エンティティ次元の視点(POV)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。エンティティ次元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、または指定された次元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。

ユーザー指定の次元メンバーを含む、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(POV)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消去するときに、メンバー・リストを選択できます。Financial Management のすべての次元のメンバー・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、ユーザー指定の次元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次元では、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。248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 `EnumMemberLists` (): どの次元にメンバー・リストがあるかを指定し、各次元のメンバー・リストを定義します。各次元のメンバー・リストの数やその次元内にある各メンバー・リストの名前を定義します。
- Sub `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、次元の次元別名を戻します。次元名および次元別名(ロング名)を指定できるカスタム次元で、`HS.Dimension` 関数は、次元別名(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、142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・リストを含める次元、各次元のリスト数、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 ElementLists(n)	<p>各ステートメントのリスト数を指定します</p> <p>Element はメンバー・リストを作成している次元の名前で、n はその次元に定義しているメンバー・リストの合計数です。たとえば、ファイルにエンティティ次元の 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>Dim EntityLists(3)</pre>
HS.Dimension = "Element"	<p>ここで、Element は次元名です。例:</p> <pre>If HS.Dimension = "Entity" Then</pre>
ElementLists(n) = "ListName"	<p>各リストの名前と数値 ID を指定します</p> <p>ここで、Element は次元名、n はメンバー・リストの ID 番号、ListName はリストの名前です。</p> <p>@POV キーワードを使用して、現在 POV に設定されている次元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。</p> <p>注： 動的 POV メンバー・リストは、Oracle Hyperion Financial Reporting および Oracle Hyperion Web Analysis で使用する場合がありますのみサポートされます。</p> <p>例:</p> <pre>EntityLists(1) = "NewEngland"</pre> <pre>EntityLists(2) = "Alloc"</pre> <pre>EntityLists(3) = "AllEntities(@POV)"</pre>
HS. SetMemberLists ElementLists	<p>指定された名前および数値 ID を設定します</p> <p>ここで、Element は次元名です。例:</p> <pre>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、スクリプ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次元では、HS.AddMemberToList ステートメントを使用して、メンバーを指定する必要があります。エンティティ次元の場合は、HS.AddEntityToList ステートメントを使用して、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 = "Element"	Element は次元です。例: <pre>If HS.Dimension = "Entity" Then</pre>
HS.MemberListID=n	数値 ID でメンバー・リストを指定します ここで、n 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 <pre>HS.MemberListID = 1</pre>
HS.AddEntityToList Member HS.AddMemberToList Member	エンティティ以外の次元のリストにメンバーを追加します ここで、Member はメンバー名です <pre>HS.AddEntityToList "UnitedStates", "Maine"</pre> <pre>HS.AddMemberToList "July"</pre>
HS.AddEntityToList Parent, Member	エンティティ次元のリストにメンバーを追加します ここで、Parent は追加しているメンバーの親で、Member はエンティティ次元のメンバーです。 <pre>HS.AddEntityToList "UnitedStates", "California"</pre>
HS.Entity.List	エンティティ次元のリストにメンバーを追加します。 <pre>HS.Entity.List("", "[Base])</pre>
HS.MemberListEntity HS. MemberListScenario HS. MemberListYear HS. MemberListPeriod	動的メンバー・リストの指定に使用します。

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： 各次元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。すべての次元にメンバー・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す。248 ページの「[Notepad ++でのルールの作成](#)」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 つ以上の次元の現在の POV メンバーに基づいて動的に作成されます。これらのリストは、Oracle Hyperion Financial Reporting および Oracle Hyperion Web Analysis でのみ使用できます。

注： この機能はエンティティ・メンバー・リストにのみ適用されます。

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

                End If

            End For

        End If

    End If

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られますが、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します。

▶ 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次元

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次元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次元

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
[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

この章の内容

仕訳ファイルの作成.....	149
仕訳のロード.....	154
仕訳の抽出.....	155

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、その後には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
\	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 = 2012

!PERIOD = January

ヘッダー・セクションの次に、Actual というシナリオの、2012 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、および繰り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>

属性	値
<ステータス>	<ul style="list-style-type: none"> ● W = 作業中 ● S = 提出済 ● A = 承認済 ● P = 転記済 ● R = 却下済
<値次元>	<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: 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: 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>エンティティ次元の有効なメンバー。値次元の 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>勘定科目次元の有効なメンバー。標準仕訳では、入力勘定になります。勘定科目の種類は、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次元の有効なメンバー。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: カスタム次元の有効なメンバー</p>
<amount type>	<ul style="list-style-type: none"> ● D = 借方 ● C = 貸方
<金額>	<p>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</p>
<ライン・アイテムの説明>	<p>オプション: 仕訳詳細の説明(最大 50 文字)。</p>

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 User's Guide の期間の管理についての説明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消去されます。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。

▶ 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 ロードするファイルの名前を入力するか、「参照」をクリックしてファイルを検索します。

注： 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
入れませんが、JLF ファイル拡張子の使用をお勧めします。

4 オプション: 「追加」をクリックして、仕訳をロードする行を追加します。

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： ロード中にエラーが検出されると、そのエラーが表示されます。

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次元を選択でき
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次元を、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次元では、特定のメンバーを選択しない
と、次元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、次元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
- 繰り返しテンプレート
- 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」フィールドでクエリーを実行できます。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。

この章の内容

フォーム・デザイナーでのデータ・フォームの作成.....	158
データ・フォーム・スクリプトの使用.....	161
相対期間の使用.....	198
競合する属性の優先順位.....	198
データ・フォームの編集.....	199
データ・フォームのロード.....	200
データ・フォームの抽出.....	201
データ・フォームの削除.....	201

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次元の視点(POV)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(POV)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。[161 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:

- [159 ページの「視点の設定」](#)
- [159 ページの「フォームの詳細の指定」](#)
- [161 ページの「フォームの行および列オプションの指定」](#)
- [161 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(POV)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、各次元に、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次元で構成されます。

POV バーの次元には、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、選択した次元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、選択した次元の有効な動的 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、行または列で次元が使用されていない場合は、次元の初期値としてユーザー POV のメンバーが使用されます。
- 次元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、シナリオ次元がグリッド列に使用され、期間次元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次元をドラッグ・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 つを超える次元をドラッグ・アンド・ドロップする場合、次元の順序を変更できます。次元をクリックすると、「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、行または列に次元をドラッグ・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。行に使用される次元は、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、[161 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。[249 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次元要素は、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、161 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、184 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、168 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次元ごとのヘッダー・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。次元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せん。

- 構文が `HeaderOption` 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。
- 次元の `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、161 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
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。
BackgroundPOV	フォームのバックグラウンド次元のメンバーを指定します。
Blank	フォームに空白の行、列またはセルを挿入します。
Cn	フォームの各列を定義します。
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。
Cell_Link	別のデータ入力フォームとリンクするため、行定義内でリンクとともに使用します。
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。
CustomHeader	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。 注： 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: <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)

スクリプトの構文	説明
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。
DynamicPOV	廃止されています。使用しないでください。
FormInputBoxLength	フォーム上の入力ボックスの幅を指定します。
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す。 NumDecimals を使用すると、行、列またはセルのこの設定を上書きされます。
FormRowHeight	フォーム内のすべての行の高さを指定します。
FormScale	フォームの位取りを指定します。
HeaderOption	フォームで次元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。
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。
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。
リンク	別のデータ入力フォームとリンクするため、 Cell_Link とともに使用します。
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。
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。
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、フォーム内の次の非表示の設定を上書きします。 SuppressInvalidRows 、 SuppressNoDataRows 、 SuppressZeroRows 、 SuppressInvalidColumns 、 SuppressNoDataColumns および SuppressZeroColumns 。
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 FormNumDecimals を上書きします。
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。
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。
HideInPov	POV で次元を非表示にするかどうかを指定するために使用します。
POVOrder	POV での次元名の順序を指定するために使用します。
PrintNumDataColsPerPage	各ページに印刷する列数を指定します。
PrintNumRowsPerPage	各ページに印刷する行数を指定します。
PrintRepeatHeadersonAllPages	各ページにヘッダーを印刷します。
Rn	フォームの各行を定義します。
ReadOnly	読取り専用の行、列またはセルを指定します。

スクリプトの構文	説明
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。
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(_)
ReportSecurityClass	フォームのセキュリティ・クラスを指定します。
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。
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。
SCalc	行、列またはセルのサーバー側の計算式を指定します。
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。FormScale を参照してください。
SelectablePOVList	フォーム用に選択できる次元のメンバーを指定します。
ShowDescriptions	次元のメンバーの説明を表示します。
ShowLabels	次元のメンバーのラベルを表示します。
文字列	列、行またはセルにテキスト文字列を追加します。
Style	行、列、セルまたは次元ヘッダーのスタイル属性を指定します。
SuppressColHeaderRepeats	繰り返される列ヘッダーを非表示にします。
SuppressInvalidColumns	無効なセルの入った列を非表示にします。
SuppressInvalidRows	無効なセルの入った行を非表示にします。
SuppressNoDataColumns	データのない列を非表示にします。
SuppressNoDataRows	データのない行を非表示にします。
SuppressRowHeaderRepeats	繰り返される行ヘッダーを非表示にします。
SuppressZeroColumns	ゼロの入った列を非表示にします。
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 およびカスタム次元で使用できます。

注： NoData セルは、[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次元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次元の[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次元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次元は、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、その次元の [SelectablePOVList](#) 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	シナリオ次元の有効なメンバーの名前。
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	エンティティ次元の有効なメンバーの名前。
Value	値次元の有効なメンバーの名前。
Account	勘定科目の有効なメンバーの名前。
ICP	ICP 次元の有効なメンバーの名前。
Custom	有効なカスタム・メンバーの名前。

注： Background POV 内の次元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次元の場合、ユーザーの視点(POV)からの次元メンバーが使用されます。

例

```
BackgroundPOV=S#Actual.Y#2012.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

```
C4=Blank
```

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。

列に指定した次元要素は、[BackgroundPOV](#) または [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。

- [Blank](#)
- [Calc1](#)
- [CellText](#)
- [CustomHeader](#)
- [NumDecimals](#)
- [上書き](#)
- [ReadOnly](#)
- [SCalc](#)
- [Scale](#)
- [文字列](#)
- [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

パラメータ	説明
CalcExpression	S <code>Calc</code> 、Blank または String を使用します。188 ページの「S <code>Calc</code> 」、167 ページの「Blank」および192 ページの「文字列」を参照してください。
POVExpression	次元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、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#2012.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。[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:1
```

例

```
R1=A#Salaries, CellText:1
```

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。行または列で次元がネストしている場合、カスタム・ヘッダーはその次元のみではなく、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つの次元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#2012.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、最初の次元の元のヘッダーを表示し、2番目の次元のヘッダーを非表示にし、3番目の次元の元のヘッダーを表示します。

```
R2=S#Actual.Y#2012.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次元ではなく、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。[192 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次元を除去します。
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。

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 `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、各次元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 には行次元の名前、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。[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、各次元の 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、次元はデータ・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、次元は POV バーに表示されます。

構文

```
HideinPOV=
Dimension
```

次元名は、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次元が POV バーに表示されるため、次元が選択可能な POV の一部である場合、この次元は無視されます。

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 POVB ーバーで非表示です。

```
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。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 **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。[160 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 の様々な次元メンバー、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、同じ次元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	<p>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。</p> <pre>R2=A#Sales, Override(2,3,A#SalesTP)</pre>
EndCell	<p>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。</p>
Override	<p>POV。たとえば、別の勘定科目で上書きするには、<code>A#newacct</code> 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、<code>A#newacct.S#newsscenario</code> を指定します。</p> <p>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。</p> <ul style="list-style-type: none"> ● Blank ● CellText ● NumDecimals ● ReadOnly ● SCalc ● Scale ● 文字列 ● 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#2012.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 での次元名の順序を指定するために使用します。次元名は、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次元は、次元番号順に、リストの末尾に追加されます。

例

次の例では、POV バーでの次元の順序は、勘定科目、期間、年、シナリオです。

注： 勘定科目次元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す。

```
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
```

PrintRepeatHeadersonAllPages

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

```
PrintRepeatHeadersonAllPages=  
Boolean
```

Boolean には、TRUE または FALSE を指定します。

例

```
PrintRepeatHeadersonAllPages=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。

行に指定した次元要素は、[BackgroundPOV](#) または [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。

- [AddMember](#)
- [Blank](#)
- [Cell_Link](#)
- [CellText](#)
- [CustomHeader](#)
- [NoSuppress](#)
- [NumDecimals](#)
- 上書き
- [ReadOnly](#)
- [SCalc](#)
- [Scale](#)
- 文字列
- [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	S Calc、Blank または String を使用します。188 ページの「S Calc」、167 ページの「Blank」および192 ページの「文字列」を参照してください。
POVExpression	次元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=S Calc (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。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せん。

例

```
C4=S#Actual.Y#2012,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icolon(;))

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、実際に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。

S Calc の計算式では、次のタイプのアイテムを使用できます。

- 次元メンバーの参照。勘定科目次元メンバーの Purchases および OtherCosts を参照します。R6=S Calc ((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=S Calc (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=S Calc (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=S Calc (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。
- S Calc 計算を使用したセルへの前方参照。

注： S Calc 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。

S Calc 行または列は、計算内で別の S Calc 行または列を参照できますが、S Calc 行または列で別の S Calc 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 S Calc の前方参照は許されます。

```
C1=A#Sales
C2=A#COGS
C3=S Calc (Col(1)-Col(2))
C4=S Calc (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次元のメンバーを指定します。

選択可能な次元の初期値は、その次元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 POV 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
Dimension	選択可能な次元を表す、次のいずれかの文字 <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)

パラメータ	説明
Parent	次元の有効な親
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、フォームの次元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、フォームの次元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。

[ShowLabels](#) および [ShowDescriptions](#) 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・ヘッダーまたは次元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 テーブルでは、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: 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 を含みません。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: Style: text-align: center

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されています。 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
```

SuppressInvalidColumns

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

```
SuppressInvalidColumns=  
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

```
SuppressInvalidColumns=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
```

SuppressNoDataColumns

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

```
SuppressNoDataColumns=  
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

```
SuppressNoDataColumns=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
```

SuppressZeroColumns

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

```
SuppressZeroColumns=  
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

```
SuppressZeroColumns=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。現在の視点(POV)が 2011 年 12 月の場合: C1=Y#@Cur.P#@Cur(+1)は 2012 年 1 月を戻します。

現在の視点(POV)が 2011 年 12 月の場合: C2=P#@Cur(+1)は 2012 年 1 月を戻します (1 月が年の最初の期間の場合、12 月は最後になります)。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 関数を他の次元に使用して現在の POV を取得できます。ただし、他の次元の@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、各次元の値は個別に解決されます。たとえば、勘定科目次元は、フォーム・レベルで設定され、シナリオ次元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。

- 5 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。
- 6 **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。
- 7 「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:

- 1 アプリケーションを開きます。
- 2 「連結」、「ドキュメント」の順に選択します。
- 3 「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。
- 4 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。
- 5 「抽出」をクリックします。
- 6 「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:

- 1 アプリケーションを開きます。
- 2 「連結」、「ドキュメント」、「データ・フォーム」の順に選択します。
- 3 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

この章の内容

データベースにデータを抽出するためのデータ・リンク(UDL)ファイルの作成	204
スター・スキーマ	207
データの作成とスター・スキーマへのエクスポート	210
スター・スキーマの更新	211
スター・スキーマの削除	213
スター・スキーマのテンプレートの作成	213
スター・スキーマのテンプレートの削除	214

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 User's Guide 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。

データをデータベースに抽出する前に、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)を設定し、Financial Management の構成ユーティリティを使用して DSN を構成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 Enterprise Performance Management System Installation and Configuration Guide 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次元の基本レベルのデータのみを抽出できます。

- 1つのシナリオ次元および1つの年次元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。

データベースにデータを抽出するためのデータ・リンク(UDL)ファイルの作成

データベースへのデータの抽出機能を使用するには、データベース・サーバーとアプリケーション・サーバーを接続する、汎用データ・リンク(UDL)ファイルを作成する必要があります。UDLファイルは暗号化することもできます。

▶ データ・リンク(UDL)ファイルを作成するには:

- 1 Windowsのエクスプローラからファイルを保存するフォルダを選択します。

注： Financial Management のインストール・ディレクトリ以外のディレクトリにファイルを保存する必要があります。そうすることで Financial Management をアンインストールしてもファイルを保持することができます。

- 2 「ファイル」、「新規」、「テキスト・ドキュメント」の順に選択します。
- 3 すべての空白を削除してファイル名を変更し、ファイルの拡張子を .udl に変更します。たとえば hfm.udl です。
- 4 UDL ファイルを構成します。

以下のトピックを参照してください。

- [204 ページの「Oracle 用データ・リンクの構成」](#)
- [205 ページの「Microsoft SQL Server 用データ・リンクの構成」](#)
- [206 ページの「IBM DB2 用データ・リンクの構成」](#)

Oracle 用データ・リンクの構成

データ・リンクはデータベース・サーバー名と、データベース・サーバーへのフル・コントロール・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。

Oracle の場合、Financial Management には Oracle OLE DB プロバイダが必要です。これは Oracle Web サイトからダウンロードできます。

▶ Oracle 用データ・リンクを構成するには:

- 1 Windowsのエクスプローラから、作成した UDL ファイルをダブルクリックします。
- 2 「データ・リンク・プロパティ」から、「プロバイダ」タブを選択し、「OLE DB プロバイダ」・リストで「OLE DB 用 Oracle プロバイダ」を選択します。
- 3 「次へ」をクリックします。
- 4 「接続」タブで次を実行します。

1. サーバー名<Oracle グローバル・データベース名>を入力します。
2. ユーザー名とパスワードを入力します。

注： ユーザーにテーブルを作成、更新および削除する権限があることを確認します。

- 5 「パスワードの保存の許可」を選択します。
- 6 接続先のデータベースを選択します。
- 7 「接続のテスト」をクリックします。
- 8 「接続のテストに成功しました」ボックスで、「OK」をクリックします。
- 9 「OK」をクリックして接続を保存し、ダイアログ・ボックスを閉じます。

Microsoft SQL Server 用データ・リンクの構成

データ・リンクはデータベース・サーバー名と、データベース・サーバーへのフル・コントロール・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。

▶ Microsoft SQL Server 用データ・リンクを構成するには:

- 1 Windows のエクスプローラから、作成した UDL ファイルをダブルクリックします。
- 2 「データ・リンク・プロパティ」から、「プロバイダ」タブを選択し、「OLE DB プロバイダ」・リストで「SQL Server 用 Microsoft OLE DB プロバイダ」を選択します。
- 3 「次へ」をクリックします。
- 4 「接続」タブで次を実行します。
 1. リレーショナル・データベースをホストするコンピュータであるサーバー名を入力します。
 2. 特定のユーザー名とパスワードを使用するが選択されていることを確認します。
 3. データベースへのフル・コントロール・アクセス権を持つユーザーのユーザー名とパスワードを入力します

注： デフォルトのユーザー名として sa を使用できます。デフォルトのユーザー名を使用する場合、パスワード用のテキスト・ボックスは空白のままにし、空白パスワードを選択します。デフォルトのユーザー名を使用しない場合、デフォルトのユーザー以外でフル・コントロール・アクセス権を持つユーザーのユーザー名を入力する必要があります。そのユーザーにテーブルを作成、更新および削除する権限があることを確認します。

4. パスワードの保存の許可を選択します。
- 5 接続先のデータベースを選択します。
- 6 「接続のテスト」をクリックします。
- 7 「接続のテストに成功しました」ボックスで、「OK」をクリックします。

- 8 「OK」をクリックして接続を保存し、ダイアログ・ボックスを閉じます。
- 9 UDL ファイル名を右クリックし、「開く」を選択して、選択した UDL ファイルをシステムのデータ・リンク・ファイルに設定します。

IBM DB2 用データ・リンクの構成

データ・リンクはデータベース・サーバー名と、データベース・サーバーへのフル・コントロール・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。

▶ IBM DB2 用データ・リンクを構成するには:

- 1 Windows のエクスプローラから、作成した UDL ファイルをダブルクリックします。
- 2 「データ・リンク・プロパティ」から、「プロバイダ」タブを選択し、「OLE DB プロバイダ」・リストで「DB2 サーバー用 IBM OLE DB プロバイダ」を選択します。
- 3 「次へ」をクリックします。
- 4 「接続」タブで次を実行します。
 1. データ・ソース名を使用するオプションを選択します。

このオプションでは IBM DB2 インストール中に確立したデータ・ソース名を選択します。

注： データ・ソース名がリストに表示されていない場合、Client Configuration Assistant のデータベース・プロパティ・ボックスで、システム・データ・ソースとして選択します。
 2. ユーザー名とパスワードを入力します。

注： ユーザーにテーブルを作成、更新および削除する権限があることを確認します。
- 5 「パスワードの保存の許可」を選択します。
- 6 「接続のテスト」をクリックします。
- 7 「接続のテストに成功しました」ボックスで、「OK」をクリックします。
- 8 「OK」をクリックして接続を保存し、ダイアログ・ボックスを閉じます。
- 9 UDL ファイル名を右クリックし、「開く」を選択して、UDL ファイルをシステムのデータ・リンク・ファイルに設定します。

UDL ファイルの暗号化

Financial Management には、アプリケーション・サーバーがデータベースと通信するために使用する UDL ファイルを暗号化するためのユーティリティがあります。EncryptHFMUDDL ユーティリティは、FinancialManagement/Server ディレクトリにあります。

UDL ファイルの拡張子に基づいて、ファイルが暗号化されているかどうかを検出されます。ファイルの拡張子が .UDL の場合、ファイルは暗号化されていないと見なされます。ファイルの拡張子が .hfmUDL の場合、ファイルは暗号化されているとみなされます。

▶ UDL ファイルを暗号化するには:

- 1 MS-DOS コマンド・プロンプトを開きます。
- 2 `encrypthfmudl hfm.udl hfm.hfmudl` と入力します。

ここで、`hfmUDL` は元の UDL ファイルの場所と名前、`hfm.hfmUDL` は暗号化された UDL ファイルの場所です。

注: 暗号化された UDL ファイルの拡張子は、`.hfmudl` にしてください。暗号化されていない元のファイルは、暗号化した後に削除できます。

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・スキーマのテーブル名は、ユーザーが指定する接頭辞で始まります。すべての次元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次元のメンバーに基づいて作成されます。選択する次元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次元ごとに乗算します。

注意 すべての次元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 - 2012
- 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(POV)の値次元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 であるとします。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(POV)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、異なる一連のテーブルを生成します。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み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。

接頭辞テーブル

各スキーマ・フォーマットでは、次元テーブルと 1 つのデータ・フィールドのキーを含む PREFIX_FACT テーブルが作成されます。また、メタデータの抽出タイムスタンプを追跡する PREFIX_DIMENSION テーブル、HFM_EA_EXTRACT テーブル、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 テーブルも作成されます。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 テーブル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: エンティティ次元の 2 つのテーブル(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 テーブルも作成されます。

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 テーブルが作成されます。

注: ビュー、年および親次元には、BASE テーブルと PARENT テーブルは作成されません。

たとえば、リレーショナル・テーブルの接頭辞が DEMO で、フォーマットが Essbase の場合は、次のテーブルが作成されます。

- 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、スキーマのテーブルを識別する接頭辞とエクスポートするセルの次元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 ステートメントで 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 次元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 次元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 - 選択した次元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・テーブルの接頭辞」で、スター・スキーマのテーブル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、別の次元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・テーブルが更新されます。

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 次元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 次元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 - 選択した次元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・テーブルの接頭辞」で、スター・スキーマのテーブル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、メタデータおよびテーブル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・テーブルの接頭辞」に、スター・スキーマのテーブル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 次元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・テーブルの接頭辞」に、スター・スキーマのテーブル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

この章の内容

仕訳レポート・スクリプトの定義	215
内部取引照合レポート・スクリプトの定義	216
内部取引レポート・スクリプトの定義	228

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 User's Guide 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。POV 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、215 ページの「仕訳レポート・スクリプトの定義」および216 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#2012.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(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次元の一致を示します。

内部取引パートナ(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、内部取引パートナ次元で管理されます。ICP次元は、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次元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、値次元の[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、カスタム次元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(POV)次元を含む次元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.xsl 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、POV 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次元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、重複エントリー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。

カスタム次元の非表示

任意のカスタム次元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

次元メンバーのラベル、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

内部取引パートナとの取引をカスタム次元で分類できます。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 POV 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 を有効な次元のメンバーに置換します。
例:

```
POV=S#Actual.Y#2012.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取引が非表示になります。

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、そのパートナー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ー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/パートナー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。

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、`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、カスタム次元を非表示にします。

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、次元ごとにカスタム次元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 をカスタム次元の別名に置換します。レポートのカスタム次元は、この次元で分類されます。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 POV を指定します。例: POV=S#ActMon.Y#2012.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	ICP 次元の有効な基本エンティティ。親エンティティではありません。
Partner	勘定科目の ICP 次元の有効なエンティティ。
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 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 の一意の識別子になります。
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	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。 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。

キーワード	説明
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>カスタム次元の場合、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>
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の選択肢からエンティティ A とパートナ B の取引のみを選択すると、エンティティ A がパートナ 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 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取引を含むように選択できます。</p>
DisplayPartnerTransactions	レポートにパートナ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。
SuppressDetails	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。

この章の内容

ルールのタイプ	231
ルールに関する注意事項	233
連結時のルールの実行	241
デフォルトの換算方法	242
Financial Management のオブジェクト	243
よく使用するルール	244
Notepad ++でのルールの作成	248
ルール・ファイルの作成	249
ルールのロード	251
ルールの抽出	253

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、[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

値の設定や消去、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、次元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</p> <p>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。</p> <p>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。</p> <p>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割り当てます。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次元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次元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> <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"> ○ NoInput ○ List <p>注： 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> ● 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 POV 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。単純なステートメント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。

ルールのタイプ	説明
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。</p> <p>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。 ● 式の左辺では、動的勘定科目のみが有効です。 ● 動的勘定科目は、式の右辺では使用できません。 ● 式の左辺で使用できるのは、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 ステートメントです。 ● Sub Dynamic セクションのステートメントはすべて、シーケンシャルに実行されます。
取引	<p>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。</p>
エクイティ・ピックアップ	<p>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。</p> <p>これは、Sub EquityPickup セクションの実行時のデフォルトの POV です：</p> <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	<p>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。</p>

Microsoft VBScript 言語に基づいた一意のスク립トで Financial Management のルールを作成します。ルールは、スク립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
- 条件付きステートメント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/値次元の交差点ごとに一度実行されます。235 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 が計算を適用する次元のメンバーは、ユーザーのカーソルが置かれたデータ・グリッド・セルと POV バーで指定されたメンバーに依存します。238 ページの「[現在の次元メンバー](#)」を参照してください。

- 関数が通貨に関連する値次元のメンバーにデータを挿入すると、Financial Management が、値次元のメンバー内の現在の値を削除する場合があります。240 ページの「関数によるデータの自動消去」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。241 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・テーブルにデータが書き込まれるか消去されます。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・テーブル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次元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、選択された値次元のメンバーおよびその値次元のメンバーが依存するすべての値次元の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次元の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。値次元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、[245 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)の値次元の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)を保管する値次元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次元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次元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)を保管する値次元の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)を保管する値次元の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次元の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)を保管する値次元の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次元のメンバーと親エンティティの交差に対して実行されます。

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。

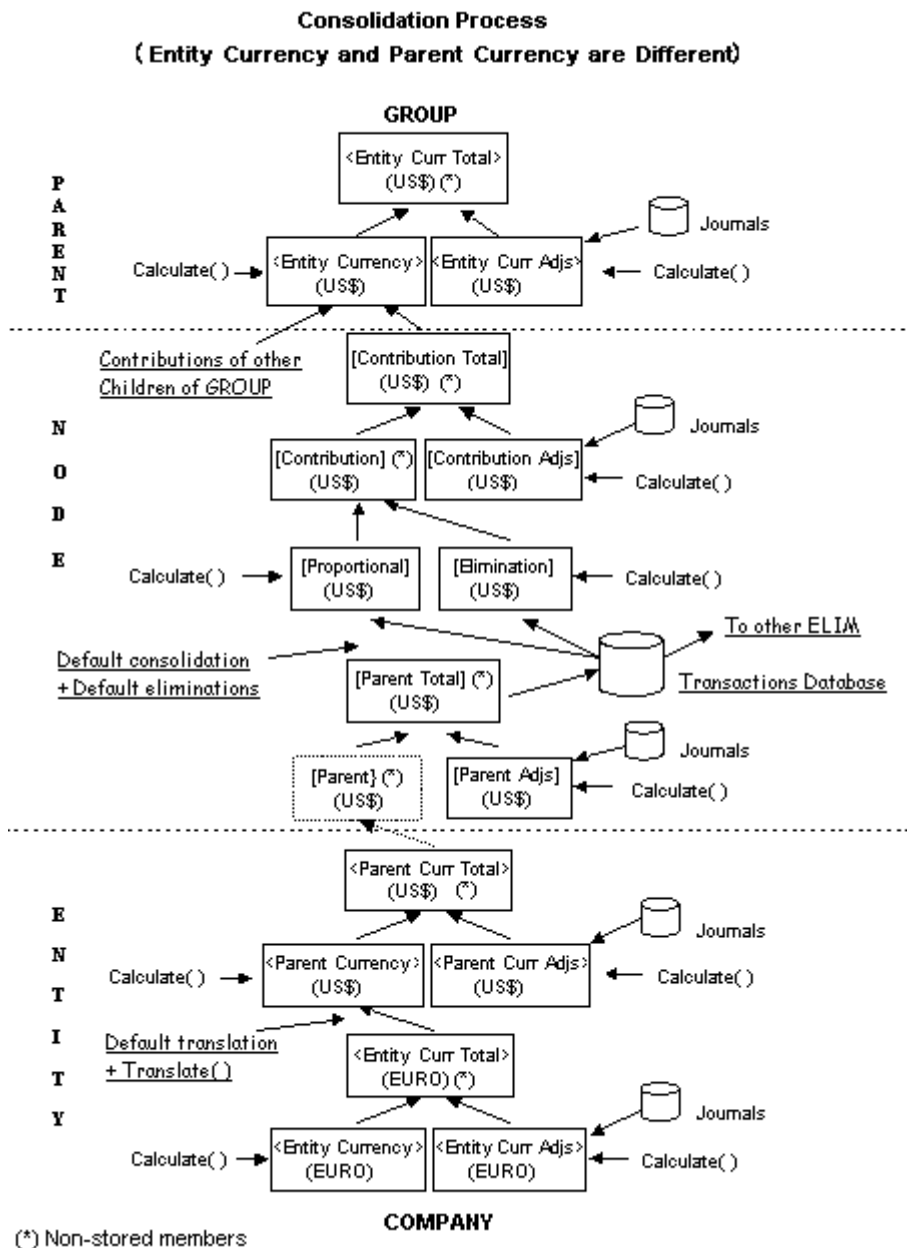
この場合、EastUS と WestUS のデータを変更し、United States を連結すると、Financial Management では、エンティティ次元と値次元における次の各交差に対してルールが実行されます。

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

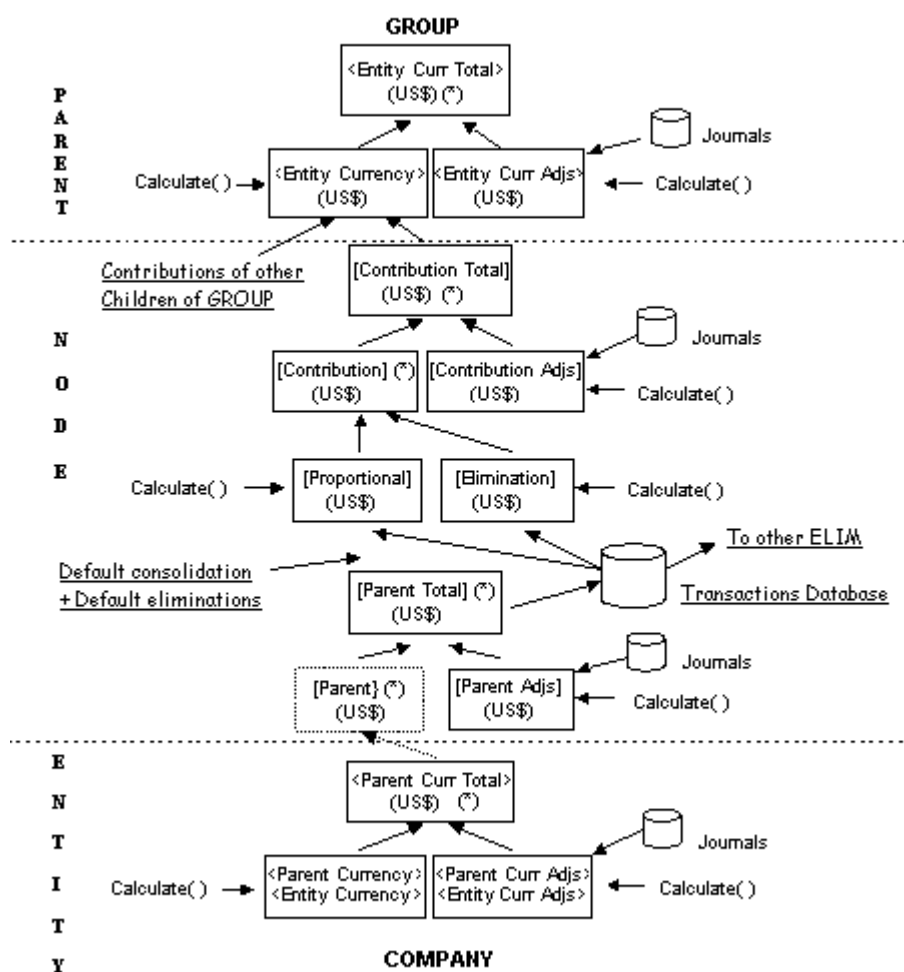
次に連結プロセスの例を示します。

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。



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

現在の次元メンバー

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、現在の次元メンバーに計算ルールを適用します。235 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。現在の次元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。
- POV バーで指定した次元のメンバー。
- 連結の場合、エンティティ次元と値次元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。235 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次元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次元メンバーが POV バー内の対応する次元の対応するメンバーと異なる場合、セルの次元メンバーは、POV バーの次元メンバーを上書きします。たとえば、ユーザーが POV 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。

次元メンバーを引数に指定すると、現在の次元とは関係なく、いくつかの関数を使用して、特定の次元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、次元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 勘定科目式で次元を指定するために使用される文字

文字	次元
A#	勘定科目
I#	内部取引パートナ
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次元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、勘定科目次元を表し、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次元のメンバー名です。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消去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、次に示す次元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せん。

- 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 CustomTop メンバーを使用します。たとえば、勘定科目の式がカスタム 3 次元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 CustomTop メンバー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 次元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次元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。各次元は、特定の文字で表されます。2 つ以上の次元を勘定科目の式に含める場合は、次元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、これらの次元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 の交差を消去します。

```
HS.Clear "A#Sales.I#Hardware"
```

関数によるデータの自動消去

関数によって通貨に関連する値次元の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消去されます。

- 交差するエンティティ・メンバーが親である場合
- 交差する勘定科目次元のメンバーが計算済勘定科目である場合

ヒント： 通貨関連の値次元のメンバーは、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次元のメンバーに挿入されると、Financial Management では、交差する勘定科目次元の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消去されます。

ヒント： 調整に関連する値次元のメンバーは、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、値次元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 で消去されます。
2. メタデータで IsCalculated として定義された勘定科目が EntityCurrAdjs で消去されます。
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 データが消去されます。
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・アップされます。
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 で消去されます。
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 で消去されます。
12. Sub Calculate()ルーチンが ParentAdjs で実行されます。
13. Proportion および Elimination のデータが消去されます。
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 で消去されます。
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 データが消去されます。
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 で消去されます。
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、残高の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 の関数の前だけに使用します。

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。249 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、現在の次元メンバーを含む勘定科目の交差にデータを挿入します (238 ページの「[現在の次元メンバー](#)」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 次元の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、等号の両側に次元のメンバーを指定できます。295 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 次元の 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。297 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

特定の次元のメンバーが POV 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 によって現在の次元メンバーをどのように判別するかについては、[238 ページの「現在の次元メンバー」](#)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、次の次元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 である場合にのみステートメントを実行できます。次のセクションでは、Member を If 構造で使用して次元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。

次元のメンバーのテスト

特定の次元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 ステートメントで始まり、End If ステートメント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 ステートメントと End If ステートメントの間にあります。

複数のメンバーのテスト

If...Then ステートメントでは、複数のメンバーをテストできます。つまり、1つの次元で2つ以上のメンバーのアクションを実行できます。「次元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 ステートメントで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

次元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 ステートメントを含む必要があります。メンバーごとにその ElseIf ステートメントのテストを行います。Elseif ステートメント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 のステートメント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 ステートメントを 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 ステートメントに IsFirst を挿入し、このステートメントと End If ステートメントの間に 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
- ステートメント
- 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。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

ルール・ルーチン	構文
Sub Input	<pre>Sub Input() 'ここに、入力ルールを入力します。 End Sub</pre>
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ます。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 POV 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・ルールの実行時に消去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、次元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ステートメント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。

この章の内容

関数の概要.....	258
ABSEp.....	265
AccountType.....	266
AccountTypeID.....	267
AddEntityToList.....	268
AddEntityToListUsingIDs.....	269
AddMemberToList.....	269
AddMemberToListUsingIDs.....	270
Alloc.....	270
AllowAdjFromChildren.....	272
AllowAdjs.....	273
ApplicationName.....	274
CalculateExchangeRate.....	274
CalculateRate.....	275
CalcStatus.....	276
Clear.....	278
Con.....	279
Consol1、Consol2、Consol3.....	280
ConsolidateYTD.....	281
ContainsCellText.....	282
ContainsCellTextWithLabel.....	283
Currency.....	284
CustomTop.....	285
DataUnitItem.....	286
Decimal.....	287
DefaultFreq.....	288
DefaultParent.....	289
DefaultTranslate.....	290
DefaultView.....	291
DefCurrency.....	292
Down.....	293
Dynamic.....	294
Exp.....	295
GetCell.....	303

GetCellNoData	303
GetCellRealData.....	305
GetCellText.....	306
GetCellTextWithLabel	306
GetCellType.....	307
GetCustomLabelArray.....	308
GetItem	309
GetItemIDs2	310
GetItemIDs2ExtDim	311
GetNumItems.....	312
GetNumLID	313
GetRate	314
GetSubmissionGroup	315
GetSubmissionPhase	315
Holding.....	316
ICPTopMember	317
ICPWeight.....	318
IDFromMember.....	319
ImpactStatus	320
Input.....	321
IsAlmostEqual	322
IsBase.....	323
IsCalculated	325
IsChild.....	326
IsConsolidated	328
IsDescendant.....	329
IsFirst.....	331
IsICP.....	332
IsLast.....	333
IsTransCur.....	334
IsTransCurAdj.....	335
IsValidDest.....	335
IsZero.....	336
List	337
Member	339
MemberFromID.....	340
Method.....	342
NoInput	342
NoRound.....	343
NumBase	344
Number	346
NumChild	347
NumCustom	349

NumDescendant	349
NumPerInGen	352
NumPeriods	353
OpenCellTextUnit	354
OpenDataUnit	355
OpenDataUnitSorted	356
Owned	356
Owner	357
PCon	357
PEPU	358
PeriodNumber	359
PlugAcct	360
POwn	361
PVAFforBalance	362
PVAFforFlow	363
RateForBalance	363
RateForFlow	364
ReviewStatus	365
ReviewStatusUsingPhaseID	366
Round	367
Scale	368
SecurityAsPartner	369
SecurityClass	371
SetCellTextWithLabel	373
SetData	373
SetDataWithPOV	374
SubmissionGroup	375
SupportsProcessManagement	376
SupportsTran	377
SwitchSign	378
SwitchType	379
Trans	381
TransPeriodic	382
UD1...3	383
ValidationAccount	384
ValidationAccountEx	385
XBRLTags	385

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、[258 ページの「関数の概要」](#)を参照してください。

関数の概要

表 34 では、Financial Management の関数、その関数で利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テーブル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次元を指定します。

ルール・タイプはこのテーブル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
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
AccountType	メンバーの勘定科目タイプを取得します。	Account	Calc、Tran、Con、Alloc

関数	説明	オブジェクト	ルールのタイプ
AccountTypeID	メンバーの勘定科目の種別 ID を取得します。	Account	Calc、Tran、Con、Alloc
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
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
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
AddMemberToListUsingIDs	メンバー ID 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
Alloc	データをセルに配賦します。	HS	Alloc
AllowAdjFromChildren	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
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
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
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
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
CalcStatus	セルの算出ステータスを取得します。	HS	Calc、Tran、Con、Alloc
Clear	データをセルから除去します。	HS	Calc、Tran、Con、Alloc
Con	[Proportion]および[Elimination]勘定科目にデータを挿入します。	HS	Con
Consol1、Consol2、Consol3	ノードの Consol1-3 システム勘定科目の値を取得します。	Node	Calc、Tran、Con、Alloc
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
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
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ます。	HS	Calc、Tran、Con、Alloc
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

関数	説明	オブジェクト	ルールのタイプ
CustomTop	勘定科目の CustomTop メンバーを取得します。	Account	Calc、Tran、Con、Alloc
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
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
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
DefaultParent	メンバーのデフォルトの親を取得します。	Account	Calc、Tran、Con、Alloc
DefaultTranslate	ルールを使用せずに換算を計算します。	HS	Tran
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
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
DOWn	ノードの直接出資(DOWn)比率を取得します。	Node	Calc、Tran、Con、Alloc
Dynamic	計算が必要な動的勘定科目の式を指定します。	HS	Dyn
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
GetCell	セルのデータを取得します。	HS	Calc、Tran、Con、Alloc
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
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
GetCellText	指定された視点(POV)のデフォルトのセル・テキストを取得します。	HS	Calc、Tran、Con、Alloc
GetCellTextWithLabel	指定したセル・テキスト・ラベルの視点(POV)からセル・テキストを取得します。	HS	Calc、Tran、Con、Alloc
GetCellType	セル・タイプを取得します。	HS	Calc、Tran、Con、Alloc
GetCustomLabelArray	カスタム次元名および別名のリストを戻します。	HS	Calc、Tran、Con、Alloc
GetItem	連結する個々のレコードを取得します。	DataUnit	Calc、Tran、Con
GetItemIDs2	次元の 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

関数	説明	オブジェクト	ルールのタイプ
GetNumItems	連結するレコードの数を取得します。	DataUnit	Calc、Tran、Con
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
GetRate	セルの換算レートを取得します。	HS	Calc、Tran、Con、Alloc
GetSubmissionGroup	セルの提出グループを取得します。	HS	Calc、Tran、Con、Alloc
GetSubmissionPhase	セルの提出フェーズを取得します。	HS	Calc、Tran、Con、Alloc
Holding	メンバーの持株会社を取得します。	Entity、Parent	Calc、Tran、Con、Alloc
ICPTopMember	現在の勘定科目次元メンバー、または指定した勘定科目メンバーの ICP 最上位メンバーを取得します。	Account	Calc、Tran、Con、Alloc
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
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、View	Calc、Tran、Con、Alloc
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
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
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
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
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
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
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

関数	説明	オブジェクト	ルールのタイプ
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
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
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
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
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
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
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
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
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
Member	メンバー名を取得します。	Entity、Parent、Period、Scenario、Value、Year、View	Calc、Tran、Con、Alloc
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
Method	メンバーの連結メソッドを取得します。	Node	Calc、Tran、Con、Alloc
NoInput	セルへの入力を禁止します。	HS	NoInput
NoRound	四捨五入をオフにします。	HS	Calc、Tran、Con、Alloc

関数	説明	オブジェクト	ルールのタイプ
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
Number	現在の期間の番号を取得します。	Period	Calc、Tran、Con、 Alloc
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
NumCustom	アプリケーションに定義されているカスタム次元の数を取得します。	HS	Calc、Tran、Con、 Alloc
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
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
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
OpenCellTextUnit	複数のセルのセル・テキストを戻します。	HS	Calc、Tran、Con
OpenDataUnit	連結のデータ単位を取得します。	HS	Calc、Tran、Con
OpenDataUnitSorted	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
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
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
PCon	連結比率を取得します。	Node	Calc、Tran、Con、 Alloc
PEPU	EPU テーブルから出資比率を取得します。	HS	エクイティ・ピックアップ
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
PlugAcct	調整勘定科目を取得します。	Account	Calc、Tran、Con、 Alloc
POwn	出資比率を取得します。	Node	Calc、Tran、Con、 Alloc
PVAForBalance	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

関数	説明	オブジェクト	ルールのタイプ
PVAForFlow	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
RateForBalance	BALANCE のデフォルト・レートを取得します。	AppSettings	Calc、Tran、Con、Alloc
RateForFlow	FLOW のデフォルト・レートを取得します。	AppSettings	Calc、Tran、Con、Alloc
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
ReviewStatusUsingPhaseID	セルのフェーズ 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
Round	データを四捨五入します。	HS	Calc、Tran、Con、Alloc
Scale	指定した通貨の位取りを取得します。	Currency	Calc、Tran、Con、Alloc
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
SecurityClass	次元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、Custom1-Custom4	Calc、Tran、Con、Alloc
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
SetData	個々のレコードを設定します。	HS	Calc、Tran
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
SubmissionGroup	次元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
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。	Scenario	Calc、Tran、Con、Alloc
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、C1...4	Tran
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
SwitchType	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。	Custom1-Custom4	Calc、Tran、Con、Alloc
Trans	YTD メソッドを使用して換算します。	HS	Tran
TransPeriodic	期別メソッドを使用して換算します。	HS	Tran

関数	説明	オブジェクト	ルールのタイプ
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
ValidationAccount	検証勘定科目を取得します。	AppSettings	Calc、Tran、Con、Alloc
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。	AppSettings	Calc、Tran、Con、Alloc
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
DestPOV	データの配賦先の宛先 POV。 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 次元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。 ● 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、勘定科目次元に、明確に定義されたカスタム次元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。297 ページの「次元の交差に関する考慮事項」を参照してください。
Expression	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、2012 の Sales 勘定科目の値と 2012 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#2012 * A#StateRate.Y#2012"
```

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 のステートメントの間にあるすべてのステートメントが実行されます。

```
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。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
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、有効な次元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、有効な次元メンバーの ID です。

戻り値

なし

Alloc

別の POV 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>割り当てられるデータのソース POV。</p> <p>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最上位メンバーです。
DestPOV	<p>データの配賦先の宛先 POV。</p> <p>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 次元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。 ● 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 POV 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す。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 およびカスタム次元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、デフォルトは最上位メンバーです。 - 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 POV 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 POV にある勘定の金額の符号を逆にします。</p> <p>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 POV のエンティティ・メンバーが親の場合、またはソース POV 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割り当てられます。

```
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ステートメントが実行されます。

```
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

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ステートメントが実行されます。

```
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返します。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 POV 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	シナリオ次元の有効なメンバーの名前。
Year	年次元の有効な年。
Period	期間次元の有効な期間。
Entity	エンティティ次元の有効なメンバーの名前。
Value	値次元の有効なメンバーの名前。

戻り値

指定された視点(POV)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 算出ステータス・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次元のデータは、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次元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。次元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次元の少なくとも 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、この次元のその他のデータ・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。選択された値次元のメンバーは、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次元のデータが最新ではない可能性があります： <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	指定したシナリオ、年、期間、エンティティおよび値次元のデータが、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。 注： 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。
NoData	指定したシナリオ、年、期間、エンティティおよび値次元にデータがありません。
NoAccess	ユーザーが、指定した次元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 POV 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ステートメントが実行されます。

```
If HS.CalcStatus("S#Actual.Y#2012.P#January.E#Connecticut.V#<EntityCurrency>") = "OK" Then
```

...

End If

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	勘定科目の有効なメンバーの名前。
ICP	ICP 次元の有効なメンバーの名前。
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 次元の有効なメンバーの名前。

現在の POV(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 の次元のすべての交差を消去するには、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 次元の GolfBalls メンバーと交差するすべての勘定科目からデータを消去します。

```
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次元の GolfBalls メンバーと交差する部分に保管されているデータを消去します。

```
HS.Clear "A#Sales.C1#GolfBalls"
```

Con

[Proportion]および[Elimination]の値次元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	これらの次元の組合せは次のとおりです。 <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。</p> <p>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。</p> <p>また、Nature 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 テーブル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があります。RTS/RTD テーブル・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： RTS/RTD テーブル・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、期間次元のメンバーと、`parent.entity` 形式で指定したエンティティ次元の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	シナリオ次元の有効なメンバーの名前。
Year	年次元の有効な年。
Period	期間次元の有効な期間。
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、シナリオ次元の有効な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 のステートメントの間にあるすべてのステートメントが実行されます。

```
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す：

- 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。次元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 - シナリオ次元の現在のメンバー
- 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せん。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次元は、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 の各行の間にあるステートメントが実行されます。

```
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 を戻します。

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。次元を指定しないと、次のデフォルト値が使用されます。

- Account - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- Scenario - シナリオ次元の現在のメンバー
- 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せん。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次元は、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 の各行の間にあるステートメントが実行されます。

```
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次元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ステートメントが実行されます。

```
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 CustomTop メンバーを戻します。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 はカスタム次元で、Member は勘定科目次元メンバーです。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 CustomTop メンバー。

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
```

```
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	勘定科目次元のメンバーを示す 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ステートメントが実行されます。

```
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ステートメントが実行されます。

```
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、有効な次元メンバーです。

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(カスタム次元ラベル) ● Entity ● Scenario
Member	次のいずれかの次元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** の間のすべてのステートメントが実行されます。

```
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ステートメントが実行されます。

```
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	エンティティ次元の有効なメンバーの名前。

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ステートメントが実行されます。

```

    If HS.Entity.DefCurrency("Connecticut") <> HS.Parent.DefCurrency("EastRegion")
Then
    ...

End If

```

D0wn

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、期間次元のメンバーと、parent.entity 形式で指定したエンティティ次元のメンバーを組み合わせます。

```

HS.Node.D0wn("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 D0wn 関数の構文

パラメータ	説明
Scenario	シナリオ次元の有効なメンバーの名前。
Year	年次元の有効な年。
Period	期間次元の有効な期間。
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#2012.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。
Expression	計算式。

注： Dynamic 関数では、IF...THEN ステートメント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 POV。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"> ● 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、勘定科目次元に、明確に定義された ICP およびカスタム次元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次元の[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次元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次元の[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次元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 ステートメントに複数の勘定科目を設定できます。299 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。

次元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次元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。内部取引パートナー次元またはカスタム次元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、勘定科目および次元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 POV に有効なすべての勘定科目にデータが挿入されます。299 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。次元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ります。
 - 次元のメンバーが 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。
 - 次元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。
 - 次元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次元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、300 ページの「Exp および次元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、メンバー名のかわりに表 54 のキーワードを使用して、宛先の年または期間次元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。

キーワード	説明
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。
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が、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次元または内部取引パートナー(ICP)次元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 次元の 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。この金額は、2012 年の Sales 勘定の金額に、2012 年の StateRate 勘定のレートを乗算して計算されます。

```
HS.Exp "A#StateTax = A#Sales.Y#2012 * A#StateRate.Y#2012"
```

Exp および次元の交差に関する考慮事項

次の例は、297 ページの「次元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。300 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。301 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。302 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 次元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、表 55 に記載されています。

表 55 次元交差のデータの例

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
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 次元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 次元の 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 POV。
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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
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 POV。
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 に戻されるブール値は 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 に戻されるブール値は 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 に戻されるブール値は FALSE になります。

例

```
dData = HS.GetCellRealData("A#Sales.C1#Prod1",bIsRealData)
```

```
If bIsRealData = TRUE then
```

```
...
```

```
End If
```

GetCellText

視点(POV)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#2012.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(POV)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。次元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 - シナリオ次元の現在のメンバー
- 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、デフォルトの次元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を使用する場合は、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 ステートメントと End If ステートメント間にあ
るすべてのステートメントが実行されます。

```
If HS.GetCellType("S#Actual.C4#[None]") = "EXPENSE" Then  
  
    ...  
  
End If
```

GetCustomLabelArray

配列のカスタム次元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つの配列を返します。

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つ以下のカスタム次元にのみ、使用できます。4つより多いカスタム次元には、[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	勘定科目次元の有効なメンバーの名前。
strICP	ICP 次元の有効なメンバーの名前。
strCustom1	カスタム 1 次元の有効なメンバーの名前。
strCustom2	カスタム 2 次元の有効なメンバーの名前。

パラメータ	説明
strCustom3	カスタム 3 次元の有効なメンバーの名前。
strCustom4	カスタム 4 次元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

勘定科目次元のメンバー、ICP 次元のメンバー、カスタム 1..4 次元のメンバー、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

次元 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 つ以下のカスタム次元にのみ、使用できます。4 つより多いカスタム次元には、[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	勘定科目次元の有効なメンバーの ID 番号。
lICPID	ICP 次元の有効なメンバーの ID 番号。
lCustom1ID	カスタム 1 次元の有効なメンバーの ID 番号。
lCustom2ID	カスタム 2 次元の有効なメンバーの ID 番号。
lCustom3ID	カスタム 3 次元の有効なメンバーの ID 番号。
lCustom4ID	カスタム 4 次元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

次元 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次元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。
lAccountID	勘定科目次元の有効なメンバーの ID 番号。
lICPID	ICP 次元の有効なメンバーの ID 番号。
lCustomID	カスタム次元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

次元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(POV)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。次元を指定しないと、次の値が使用されます。

- Account - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- Scenario - シナリオ次元の現在のメンバー
- 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ステートメントが実行されます。

```
If HS.GetNumLID("A#Sales.C1#Prod1.C2#Region1") = 0 then
```

```
...
```

```
End If
```

GetRate

現在の POV または指定した POV 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 POV の為替レート。

例

```
dVar1 = HS.GetRate("S#Actual.Y#2012.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.C  
4#Basketballs.I#EastSales" )
```

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.C4#Basketballs.I#EastSales" )
```

Holding

現在、指定されたエンティティ、または親次元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次元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ステートメントが実行されます。

```
If HS.Entity.Holding("France") = "Europe" Then  
  
    ...  
  
End If
```

ICPTopMember

現在の勘定科目次元メンバー、または指定した勘定科目メンバーの ICP 最上位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.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	勘定科目次元の有効なメンバーの名前。
Var1	VisualBasic 変数。

戻り値

勘定科目の ICP の最上位メンバーの名前を示す文字列。

例

この例では、Sales 勘定の ICP 最上位メンバーが TotalProd である場合、If...Then ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
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。値次元の[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

ICP トップ・メンバーに集約される 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ステートメントが実行されます。

```
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 (カスタム次元ラベル)● Entity● ICP● Parent● Period● Scenario● Value● Year● View
Element	次のいずれかの次元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	シナリオ次元の有効なメンバーの名前。
Year	年次元の有効な年。
Period	期間次元の有効な期間。
Entity	エンティティ次元の有効なメンバーの名前。
Value	値次元の有効なメンバーの名前。

注： 指定したPOV が、処理されてい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とエラーが発

戻り値

なし。

例

```
HS.ImpactStatus "S#Actual.Y#2012.P#January"
```

Input

基本の計算済勘定科目の親エンティティへのデータ入力を可能にします。エンティティ通貨の値次元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次元にロール・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次元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せん。

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
ンプリメンテーショ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。

-0.000000000000001 から 0.000000000000001 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 (カスタム次元ラベル)● 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	次のいずれかの次元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	エンティティ次元の有効なメンバーの名前。
Scenario	シナリオ次元の有効なメンバーの名前。
Year	年次元の有効な年。
Period	期間次元の有効な期間。

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(POV)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ステートメントが実行されます。

```
If HS.Entity.IsBase("EastRegion","Connecticut") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Prod").IsBase("P3000-Phones") = TRUE Then
```

```
...
```

```
End If
```

IsCalculated

現在の勘定科目次元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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
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 (カスタム次元ラベル) ● Entity ● Parent
Parent	親メンバーの名前(親は必須です。)
Element	次のいずれかの次元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	エンティティ次元の有効なメンバーの名前。
Scenario	シナリオ次元の有効なメンバーの名前。
Year	有効な年。
Period	期間次元の有効な期間。

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ステートメントが実行されます。

```
If HS.Entity.IsChild("EastRegion","Connecticut") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Prod").IsChild("P3000-Phones") = TRUE Then
```

```
...
```

```
End If
```

IsConsolidated

現在の勘定科目次元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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
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 (カスタム次元ラベル)● Entity● Parent
Parent	有効な親メンバーの名前。親は必須です。
Element	次のいずれかの次元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

パラメータ	説明
Entity	エンティティ次元の有効なメンバーの名前。
Scenario	シナリオ次元の有効なメンバーの名前。
Year	年次元の有効な年。
Period	期間次元の有効な期間。

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ステートメントが実行されます。

```
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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
If HS.Period.IsFirst = TRUE Then  
  
    ...  
  
End If
```

IsICP

現在の勘定科目次元メンバーまたはエンティティ次元メンバー、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(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	勘定科目次元またはエンティティ次元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ステートメントが実行されます。

```
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 のステートメントの間にあるすべてのステートメントが実行されます。

```
If HS.Period.IsLast = TRUE Then  
  
    ...  
  
End If
```

IsTransCur

現在の値次元メンバーが、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

現在の値次元の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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
If HS.Value.IsTransCur = TRUE Then
```

```
...
```

```
End If
```

IsTransCurAdj

現在の値次元メンバーが、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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
If HS.Value.IsTransCurAdj = TRUE Then
```

```
...
```

```
End If
```

IsValidDest

指定した POV 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。次元を指定しないと、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。その以外の場合は、勘定科目のトップ・メンバーが使用されます。
- Scenario - シナリオ次元の現在のメンバー
- 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 POV 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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
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。

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 (カスタム次元ラベル) ● 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	シナリオ次元の有効なメンバーの名前。
Year	年次元の有効な年。
Period	期間次元の有効な期間。

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： 値次元の現在の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 のステートメントの間にあるすべてのステートメントが実行されます。

```
If HS.Entity.Member = "California" Then  
  
    ...  
  
End If
```

MemberFromID

指定した ID 番号の次元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 (カスタム次元ラベル) ● Entity ● ICP ● Parent ● Period ● Scenario ● Value ● Year ● View
Parent	有効な親メンバー。
ElementID	次のいずれかの次元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。 <ul style="list-style-type: none"> ● Account ● Custom1...4 ● Custom ● Entity ● ICP ● Parent ● Period ● Scenario ● Value ● Year

戻り値

次元メンバー名。

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(POV)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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
If HS.Node.Method("S#Actual.Y#2012.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、データ入力が一部の次元交差には可能で、その他の次元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 2012 年の Budget シナリオが交差するセルへの入力
が禁止されます:

```
Sub NoInput
```

```
HS.NoInput "S#Budget.Y#2012.A#Sales"
```

```
End Sub
```

NoRound

後続くすべての Exp ステートメントの四捨五入をオフにします。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、四捨五入をオフにできま
す。たとえば、HS.Round(0)で四捨五入がオフになります。

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る金額を小数第 2 位で四捨五入し、次に NoRound 関数を使用して、SalesNoRound 勘定のセルに挿入される金額を四捨五入しません。

```
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 (カスタム次元ラベル) ● Entity ● Parent ● Node
Element	次のいずれかの次元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 <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。</p> <p>次元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。</p> <pre>iAcctBase = HS.Account.NumBase("ALL")</pre>
Scenario	シナリオ次元の有効なメンバーの名前。
Year	年次元の有効な年。
Period	期間次元の有効な期間。
Entity	エンティティ次元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せん:

- 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 のステートメントの間にあるすべてのステートメントが実行されます。

```
If HS.View.PeriodNumber = 1 Then  
  
    ...  
  
End If
```

NumChild

現在の次元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 ● Custom1-4 ● Custom (カスタム次元ラベル) ● Entity ● Node ● Parent
Element	次のいずれかの次元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 ● 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。</p> <p>次元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。</p> <pre>iAcctBase = HS.Account.NumChild("ALL")</pre>
Scenario	シナリオ次元の有効なメンバーの名前。
Year	年次元の有効な年。
Period	期間次元の有効な期間。
Entity	エンティティ次元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次元の合計数を戻します。たとえば、5つのカスタム次元を作成して、1つのカスタム次元を削除し、2つのカスタム次元を追加する場合、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次元の合計数である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次元の数。

例

```
nCustoms = HS.NumCustom
```

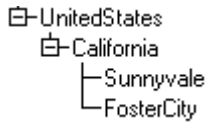
NumDescendant

現在の次元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 (カスタム次元ラベル) ● Entity ● Node ● Parent
Element	次のいずれかの次元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 ● 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。</p> <p>次元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。</p> <pre>iAcctBase = HS.Account.NumDescendant("ALL")</pre>
Scenario	シナリオ次元の有効なメンバーの名前。
Year	年次元の有効な年。
Period	期間次元の有効な期間。
Entity	エンティティ次元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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
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.NumPeriod("
ScenarioName
")
```

または

```
HS.Scenario.NumPeriod(" ")
```

または

```
HS.Scenario.NumPeriod(
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.NumPeriod("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)。ページ次元メンバー(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。サブキューブ次元(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、次元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#2012", "", "", "")
```

```
Set CTU = HS.OpenCellTextUnit ("S#Actual.Y#2012.P(. [Base] .A(. [Base])", "", "Period", "Ascending")
```

```
Set CTU = HS.OpenCellTextUnit ("S#Actual.Y#2012.E(. [Base]) .P(. [Base])", "CouponRate", "LABEL", "Ascending")
```

```
Set CTU = HS.OpenCellTextUnit ("S#Actual.Y#2012.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-1", "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>は次元名の文字列であり、次の6つの次元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	シナリオ次元の有効なメンバーの名前。
Year	年次元の有効な年。
Period	期間次元の有効な期間。
Entity	エンティティ次元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#2012.P#January.E#Regional.Connecticut")
```

PEPU

エクイティ・ピックアップ(EPU)テーブル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 テーブルの出資比率。

例

```
HS.PEPU(S#Actual.Y#2012.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

	期間	YTD	QTD	HYTD
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
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、有効な勘定科目次元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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
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
Scenario	シナリオ次元の有効なメンバーの名前。
Year	年次元の有効な年。
Period	期間次元の有効な期間。
Entity	エンティティ次元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#2012.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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
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ステートメントが実行されます。

```
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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
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	シナリオ次元の有効なメンバーの名前。
Year	有効な年。
Period	期間次元の有効な期間。
エンティティ	エンティティ次元の有効なメンバーの名前。
Value	値次元のメンバーの名前。

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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
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#2012.P#January.E#Connecticut",3)
```

Round

Exp 関数から戻されたデータを四捨五入します。四捨五入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、状況に応じて四捨五入の位置を変える必要がある場合は、Round を含む複数のステートメントを使用できます。

構文

```
HS.
Round
(
Unit
```

)

Unit は四捨五入の係数です。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 の値に四捨五入します。値 0 は四捨五入をオフにします。

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 関数に対して四捨五入がオフになります。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る金額を小数第 2 位で四捨五入し、次に NoRound を使用して、SalesNoRound 勘定のセルに挿入される金額の四捨五入をオフにします。

```
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(  
  Val1  
)
```

表 76 Scale 関数の構文

パラメータ	説明
Currency	有効な通貨の名前。
Var1	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 ステートメントと End If ステートメント間にあるすべてのステートメントが実行されます。

```
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
Entity	エンティティ次元の有効なメンバーの名前。
Var1	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 のステートメント間にあるステートメントが実行されます。

```
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 (カスタム次元ラベル)
Element	次のいずれかの次元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 ● シナリオ ● エンティティ ● Custom1...4 ● カスタム
Var1	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 のステートメント間にあるステートメントが実行されます。

```

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
IView	0 = シナリオ・ビュー 1 = 期別ビュー 2 = 年次累計ビュー
IAccount	データを設定する勘定科目の ID 番号。
IICP	データを設定する ICP の ID 番号。
ICustom1...4	データを設定するカスタム次元の ID 番号。
dData	設定するデータ値。
bAddToExistingData	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
POV	有効な POV。
dData	設定するデータ値。
bAddToExistingData	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

次元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、次のいずれかの次元です: 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

次元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
Scenario	有効なシナリオ。
Var1	シナリオ次元のメンバーを表す 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ステートメントが実行されます。

```
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	カスタム次元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 のステートメントの間にあるすべてのステートメントが実行されます。

```

If HS.Custom1.SwitchSign("") = TRUE Then

    ...

End If

If HS.Custom("Product").SwitchSign("P3000-Phones") = TRUE Then

    ...

End If

```

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、カスタム次元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: <ul style="list-style-type: none"> ● Custom1 ● Custom2 ● Custom3 ● Custom4 ● Custom
Member	カスタム次元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 のステートメントの間にあるすべてのステートメントが実行されます。

```

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 POV。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次元に対して、次元の有効なすべてのメンバーに書き込まれます。指定された各次元に対して、指定されたメンバーのみに書き込まれます。
SourcePOV	ソース POV。ソースは次元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 の次元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次元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。値次元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 POV が空白の場合は、ソース POV として宛先 POV 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、 242 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せん。

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 POV。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次元に対して、次元の有効なすべてのメンバーに書き込まれます。指定された各次元に対して、指定されたメンバーのみに書き込まれます。
SourcePOV	ソース POV。ソースは次元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 の次元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次元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、 242 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

パラメータ	説明
要素	次のいずれかの次元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 ● 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** のステートメント間にあるステートメントが実行されます。

```
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 のステートメント間にあるステートメントが実行されます。

```
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す:

- 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

この章の内容

管理レポート関数	388
ビジネス・ルール関数	408

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	カスタム次元に割り当てます	Custom_Alloc (Destination, Source, Factor, FactorN, FactorD, Elimination)	プロシージャ
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

カスタム関数	説明	パラメータ	関数のタイプ
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。 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。 [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。

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する **Periods** パラメータに基づいて、2012 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	Oct2011	Nov2011	Dec2011	Jan2012	Feb2012	Mar2012
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、2012 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、2012 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	Oct2011	Nov2011	Dec2011	Jan2012	Feb2012	Mar2012
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
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、2012 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	Dec2011	Jan2012	Feb2012	Mar2012
A#Cash	900	1,200	1,100	1,500
Difference("A#Cash", "")	N/A	300	200	600
Difference("A#Cash", "YTD")	N/A	300	200	600
Difference("A#Cash", "Periodic")	N/A	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値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	カスタム次元と内部取引次元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	カスタム次元と内部取引次元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	カスタム次元と内部取引次元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	カスタム次元と内部取引次元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。 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、それらと比較します。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せん。
- 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、2012 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	Dec2011	Jan2012	Feb2012	Mar2012
A#FA_COST	900	1,200	1,100	1,500

勘定科目	Dec2011	Jan2012	Feb2012	Mar2012
Opening("A#FA_COST", " ")	N/A	900	900	900
Opening("A#FA_COST", "YTD")	N/A	900	900	900
Opening("A#FA_COST", "Periodic ")	N/A	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 などのカスタム次元や内部取引次元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、カスタム次元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ギュレーション通貨と同一であり、トライアンギュ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 - トライアンギュレーション通貨と同一の場合

		カスタム 1 次元のレート	
		子	親
カスタム 2 次元のレート	子		(2)
親	(1)		

次の表では、トライアンギュ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 - トライアンギュレーション通貨と異なる場合

		カスタム 1 次元のレート		
		子	親	トライアンギュレーション
カスタム 2 次元のレート	子			(2)
親				
トライアンギュレーション		(1)		

次の表では、トライアンギュ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 - トライアンギュレーションが指定されていない場合

		カスタム 1 次元のレート		
		子	親	アプリケーション
カスタム 2 次元のレート	子		(2)	(4)
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#FXO = 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 FXO 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#FXO","A#FXO","A#EOP_RATE", ""
HS.EXP "A#FXO = 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、割当ての基本として、消去 POV に対する合計割当数量の転記を逆に
するオプション付きで係数 POV を使用し、ソースの視点(POV)を宛先 POV に割
り当てます。この関数は、カスタム次元の割当て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	次元メンバーの有効な組合せである有効なソース POV。Source は割り当てられる金額です。
Factor	有効なソース POV。Factor は、割当て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、割当て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、割当て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、割当ての基準として、消去 POV に対する合計割当金額の転記を逆にするオプション付きで係数 POV を使用し、ソース POV を宛先 POV に割り当てます。この関数は、カスタム次元の割当て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 に割り当てられます。割当金額の逆数が Allocations 勘定科目に転記されます。

表 106 Custom_Alloc 関数の例

勘定科目	Jan2012	Feb2012	Mar2012
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	50	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・ステートメント。
- 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 Clitem

    C1list = HS.Custom1.List("Alloc")

    For Each Clitem in C1list

        Source = "A#Telephone.C1#[None]"
        Destination = "A#Telephone.C1#" & Clitem
        Factor = "A#Factor.C1#" & Clitem
        FactorN = "A#Sales.C1#" & Clitem
        FactorD = "A#Sales.C1#TotalProducts"
        Elimination = "A#ProductAllocations.C1#" & Clitem

        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	次元メンバーの有効な組合せである有効なソース POV。Source は割り当てられる金額です。
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	Jan2012	Feb2012	Mar2012
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・ステートメント。
- 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	次元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	Jan2012	Feb2012	Mar2012
A#GrossMargin	1000	100	750
A#TotalRevenues	2000	400	1000
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・ステートメント。
- 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、プロファイル勘定科目(収益プロファイル、4-4-5 など)に定義されているプロファイルに基づいて、宛先勘定科目のすべての期間に割り当てます。

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	次元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。Factor は、割当て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、分散割当て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、分散割当ての基準として使用される分母係数です。
Temp	有効な宛先勘定科目。Temp 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。Temp 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)を割り当てます。

時間ベースの割り当ては、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割り当てられる、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割り当てられます。

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・ステートメント。
- 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	次元メンバーの有効な組合せである有効なソース POV。
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	Jan2012	Feb2012	Mar2012
A#UnitsSold	1000	2000	5000
A#Price	1.25	1.00	0.50
Units_Rates("A#UnitsSold", A#Price)	1250	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・ステートメント。
- 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
```

この章の内容

Calculation Manager のセキュリティ役割	421
Calculation Manager でのアプリケーションの操作	422
Calculation Manager へのルールの移行	422
関数セレクタでの VB 関数のサポート	423
Financial Management の特別な VB Script 関数	424

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配置など、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。

ルールを配置するには、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 User Security Administration Guide 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配置することもできます。

注： ルールを Calculation Manager に配置した後で VB Script ルールをロードしようとする、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。



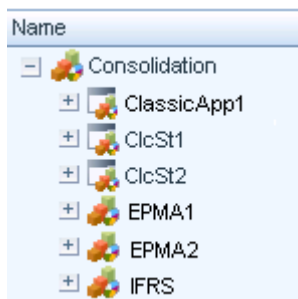
Calculation Manager の連結フォルダを開くときは、使用しているアプリケーションのリストがアルファベット順に表示されます。クラシック・アプリケーションはアイコンで示され、Oracle Hyperion EPM Architect アプリケーションでは、次のアイコンが使用されます:.

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 の使用方法の詳細は、Oracle Hyperion Calculation Manager Designer's Guide 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(.rle)ファイルがある場合、それらのファイルを手動で変換するかわりに、ルール・マイグレータ・ユーティリティを使用して Calculation Manager にファイルを移行できます。ユーティリティは、VB Script ファイルを XML フォーマットのグラフィカル・ルール・オブジェクトに変換し、Calculation Manager にロードできます。ルール・マイグレータ・ユーティ

リティ FMRulesMigrator.exe は、デフォルトで Financial Management \Utilities フォルダにインストールされます。

関数セレクトタでの VB 関数のサポート

Calculation Manager の関数セレクトタは 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 を戻します。
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戻します。
Join	配列内の多くの部分文字列で構成される文字列を戻します。
LBound	配列の指示された次元の最も小さいサブスクリプトを戻します。
Split	指定された数の部分文字列を含む、ゼロから始まる 1 次元配列を戻します。
UBound	配列の指示された次元の最も大きいサブスクリプトを戻します。

日付/時刻の関数

関数	説明
Date	現在のシステム日付を戻します。
DateAdd	指定された時間間隔が追加された日付を戻します。
DateDiff	2 つの日付間の間隔数を戻します。
DatePart	指定された日付の指定された部分を戻します
DateSerial	指定された年、月および日の日付を戻します。
Day	月の日(1 から 31 まで)を表す数字を戻します。
Month	年の月(1 から 12 まで)を表す数字を戻します。
MonthName	指定された月の名前を戻します。

数学

関数	説明
Abs	指定された数の絶対値を戻します。
Fix	指定された数の整数部を戻します。

関数	説明
Int	指定された数の整数部を戻します。

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戻します。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戻します。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戻します。
Len	文字列内の文字数を戻します。
Mid	文字列から指定された数の文字を戻します。
Right	文字列の右側から指定された数の文字を戻します。
StrComp	2つの文字列を比較し、比較の結果を表す値を戻します。
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

値	ループ変数	VBScript 生成
@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 の次のステートメントが生成されます。

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 の次のステートメントが生成されます。

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・スペースを再配賦します。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 ステートメントが生成されます。

```
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 ステートメントが生成されます。

```
RedimPreserve VarArrayX(5)
```

または

```
RedimPreserve VarArrayXY(5,9)
```

または

```
RedimPreserve VarArrayXY(5,i)
```

この章の内容

内部取引の設定	427
理由コードの管理	430
内部取引のモニター	432
エンティティのロックとロック解除	433
内部取引の要約の表示	434

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次元にまたがって、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 User's Guide 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。427 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次元を指定します。249 ページの「ルール・ファイルの作成」および 251 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。428 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。431 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。98 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。429 ページの「[「転記前に照合」オプションの設定](#)」および428 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、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： 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がチェックされます。</p>

「転記前に照合」オプションの値	説明
制限	<p>「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：</p> <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： 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:

- 431 ページの「理由コードの追加」
- 431 ページの「理由コードの編集」
- 431 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 User's Guide 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。

内部取引のモニター

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監視します。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入力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監視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。

内部取引を監視するには、「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(POV)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(POV)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 User's Guide を参照してください。

▶ 内部取引を監視するには:

- 1 「連結」、「内部取引」、「モニター」の順に選択します。
- 2 POV 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、ステータスを監視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 User's Guide 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・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。

この章の内容

提出フェーズの定義	437
送信グループの設定	438
提出グループおよびフェーズの例	439
提出グループのフェーズへの割当て	442
未割当ての提出グループの表示	443

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次元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。

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次元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。[91 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次元(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次元による提出が必要で、カスタム次元または内部取引(ICP)次元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 つの次元を有効にする必要があります。
- 検証勘定を割り当てます。[91 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、提出グループを次元メンバーに割り当てます。[438 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。[442 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次元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。

複数の次元がフェーズ送信に使用される場合、次元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

これらの次元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

これらの次元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次元メンバーの属性を設定し、次元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当てを示します。

会計	提出グループ
HistData	0
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

管理者は、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次元メンバーに適用されません。

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はできないからです。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、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消去されます。

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。これらの次元メンバーは、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消去できます。

2 シナリオを変更するには、POVでシナリオ次元をクリックし、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次元メンバーに割り当てられたグループと、どの次元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。グループが次元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。

この章の内容

プロセス管理のアラートの設定	445
内部取引のアラートの設定	447

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 Installation and Configuration Guide 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ます。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 User Security Administration Guide 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。次のアクション(「第1パス」、「確認レベル1」から「確認レベル10」、「提出済」、「承認済」、「発行済」)のアラートを設定できます。

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 User Security Administration Guide 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 User's Guide を参照してください。



レジストリ設定

この付録には、Financial Management で使用可能なレジストリ設定がリストされています。

設定は、「Server」フォルダの下のレジストリ・エディタで使用できます：

```
HKEY_LOCAL_MACHINE\SOFTWARE\Hyperion Solutions\Hyperion Financial Management  
\Server
```

レジストリ・キーが存在しない場合は、regedit を使用して、新しい DWORD としてキーを追加し、値を設定します。

▶ サーバーのレジストリ設定を変更するには：

- 1 「スタート」、「ファイル名を指定して実行」の順に選択します。
- 2 「開く」ボックスに「regedt32」と入力し、「OK」をクリックします。
- 3 HKEY_LOCAL_MACHINE\SOFTWARE\Hyperion Solutions\Hyperion Financial Management\Server を選択します。
- 4 変更するレジストリ設定をダブルクリックして値を変更し、「OK」をクリックします。

表 118 レジストリ設定

設定	説明	値
AllowedMissedHeartBeats	実行中のタスクを取り消すまでにシステムで失敗可能なハートビート数	範囲: 5-20 デフォルト値: 10
AutoClearDeadTasks	このキーは、\RunningTasks の下にあります。 完了したタスクをシステムから自動的に消去するかどうか	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消去します デフォルト値: 1
AutoClearDeadTasksAfterSeconds	このキーは、\RunningTasks の下にあります。 完了したタスクを自動的に消去するまでシステムが待機する秒数。デフォルトで、実行中のタスクはデータベースに 1800 秒(30 分)間残ります。最小値は AllowedMissedHeartBeats 回、HeartBeatRateInSeconds で、デフォルトは 600 秒です。	範囲: 600 秒(10 分) - 864,000 秒(10 日) デフォルト値: 1800 秒(30 分)

設定	説明	値
AutoClearEAFlatfileTasksAfterSeconds	このキーは、\RunningTasks の下にあります。 「タスクの実行」からフラット・ファイルへのデータの抽出タスクを自動的に消去し、アプリケーション・サーバーからフラット・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
ConsolidationMultiThreadingScheme	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。これは、すべてのアプリケーション・サーバーで実行する必要があります。	0 - 無効化します 1 - 有効化します デフォルト値: 0
COTFSize	サブキューブ内の集約データのキャッシュ・サイズ。	範囲: 範囲にハード・リミットはありません。任意の正数を受け入れます。ただし、小さすぎる値を設定した場合、システムは集約データの大部分のキャッシュを無効化します。大きい値を設定すると、メモリーの使用量が増加します。 デフォルト値: 20,480
CSUseSQLBinding	算出ステータスに SQL バインドを使用するかどうか	0 - 算出ステータスの SQL バインドを無効化します 1 - 算出ステータスの SQL バインドを使用します デフォルト値: 1
EACommandTimeout	データベースへのデータの抽出のみに対して。SQL クエリーがデッドロックであると決定されるまで経過する必要がある秒数。	0 - タイムアウトなし 1-65,535 - タイムアウト (秒単位) デフォルト値: 120
EALogEntirePOV	「データベースへのデータの抽出」での POV をログ・ファイルに記録するかどうか。ログ・ファイルは「タスクの実行」から使用可能です。	0 - ロギングを無効化します 1 - ロギングを有効化します デフォルト値: 1
EATraceADOErrors	「データベースへのデータの抽出」の ActiveX データ・オブジェクト(ADO)エラーをログにトレースするかどうか。	0 - トレースを無効化します 1 - トレースを有効化します デフォルト値: 0
EAUseSQLBinding	「データベースへのデータの抽出」に SQL バインドを使用するかどうか。 注意 デフォルト値の 1 を使用することをお勧めします。SQL バインドを使用不可にすると、パフォーマンスが低下することがあります。	0 - SQL バインドを無効化します 1 - SQL バインドを使用します デフォルト値: 1

設定	説明	値
EnableEntireCubeReloading	いくつかのレコードがアップロードされたときにデータのサブキューブ全体の再ロードを有効化するかどうか。 注： デフォルト設定の 0 を使用することをお勧めします。	0 - 無効化します 1 - 有効化します デフォルト値: 0
EnableRulesCaching	ルールのキャッシュを有効化するかどうか。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 - 無効化します 1 - 有効化します デフォルト値: 1
FilterInvalidRecords	ロード・プロセス中に、システムはどのレコードをロードするか	0 - 無効なレコードをロードします 1 - 無効なメンバーを含むレコードをフィルタします 2 - 無効なメンバーおよび交差を含むレコードをフィルタします デフォルト値: 2
HeartBeatRateInSeconds	このキーは、RunningTasks の下にあります。 実行中のタスクの「進行中」メッセージをシステムが保持する秒数	範囲: 30-300 デフォルト値: 60
HsvActionsLogLevel	注： このキーは、Hyperion Financial Management\Web の下にあります。 システムは自動タスクフローのどのイベントを記録するか。メッセージは、EPM_Oracle Instance\diagnostics\logs\hfm\hfmtaskflows-dialog に記録されます。	0 - なし。ロギングは発生しません 1 - エラー。例外で発生することを記録します 2 - 警告。予期しない入力パラメータなど、警告メッセージを記録します 3 - デバッグ。タスクの自動化および主要なメソッドの入力パラメータを記録します 4 - トレース。すべてのメソッドおよびクラスでの入力および終了メソッドを記録します デフォルト値: 0

設定	説明	値
IcmSystemReportTranslationMode	このキーは、\SystemReports の下にあります。システム・レポートに使用する通貨換算ロジック	<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。レジストリ・エントリが指定されていない場合、これはデフォルト設定です。</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せん。</p> <p>2 - オンザフライ。デフォルト換算を使用 - 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>
LogDebugInfo	算出ステータスに関する情報を記録するかどうか。問題に影響を与える算出ステータスのデバッグの目的で使用します。	<p>0 - システムは算出ステータスに関する情報を記録します</p> <p>1 - システムは算出ステータスに関する情報を記録しません</p> <p>デフォルト値: 0</p>
LogSysInfoHeartBeatRate	このキーは、\RunningTasks の下にあります。システム情報メッセージを記録するハートビート・スレッドの間隔。	<p>範囲: 0-60 分</p> <p>デフォルト値: 15</p>
MaxCubeUpdateThreadWaitInSeconds	プロセスの停止前にデータ・キューブを更新する際にシステムが待機する秒数	<p>範囲: 480 秒。最大ハード・リミットはありません。</p> <p>デフォルト値: 900 秒(15 分)</p>
MaxDataCacheSizeinMB	Oracle Hyperion Financial Management アプリケーション・サーバーがセル値およびセル・ステータスを格納するために割り当てるメモリの最大容量(MB)	<p>範囲: 範囲にハード・リミットはありません。</p> <p>デフォルト値: 260</p>
MaxNumCommitThreads	複数のサブキューブを更新する際に許可されるマルチスレッド接続の最大数	<p>範囲: 範囲にハード・リミットはありません</p> <p>デフォルト値: 8</p>

設定	説明	値
MaxNumConcurrentConsolidations	アプリケーション・サーバーごとに許可される同時連結の最大数。設定値を超えて実行される連結は、スケジュールされた連結としてキューに入ります。	範囲: 1-8 デフォルト値: 8
MaxNumCubesInRAM	特定の時点でメモリーに保持されているキューブ数。この設定を小さくすると、疎アプリケーション(たとえば、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率を軽減できます。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。 注: この設定は、リリース 11.1.2 および 9.3.3 にのみ適用されます。	範囲: 100-500,000 デフォルト値: 30,000
MaxNumCubesInTemporaryCache	データ取得時に、メモリーに保持されるサブキューブの数(たとえば、データ・グリッド、データ・フォームおよびレポート)。密アプリケーションの場合、この数値を小さくすると、データ取得時の高いメモリー使用率を軽減できます。連結は、このパラメータによって影響を受けません。 注: この設定は、リリース 11.1.1.3、9.3.1.4 および 9.2.1.2 にのみ適用されます。	範囲: 0-10,000 デフォルト値: 100
MaxNumDataRecordsInRAM	システムが RAM に格納するデータ・レコードの最大数	範囲: 範囲にハード・リミットはありません デフォルト値: 1,000,000
MaxNumRecordsInCalcStatusCache	システムが算出ステータス・キャッシュに格納するレコードの最大数	範囲: 0-1,000,000 デフォルト値: 1000
MaxNumRetriesOfBaseLevelCalculation	連結時に許可される基本レベル・エンティティの追加計算の最大数。代替の連結ロジックを有効化するために使用されます。数値を大きくするとステータスが「OK」になる可能性が高まりますが、パフォーマンス低下の原因になる場合があります。すべてのアプリケーション・サーバーでレジストリを変更する必要があります。	正の整数(たとえば、1、2 または 3) ゼロまたは負数 - 既存のデフォルト連結プロセスを保持します デフォルト値: 0
MinDataCacheSizeinMB	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: 130
MultiThreadCalcStatus	影響を与える算出ステータスに複数のスレッドを許可するかどうか	0 - 無効化します 1 - 有効化します デフォルト値: 1
MultiThreadDataLoad	データ・ロードの複数のスレッドを許可するかどうか。 注意 この設定は、常に有効化する必要があります。テスト目的でのみ使用されます。	0 - 無効化します 1 - 有効化します デフォルト値: 1

設定	説明	値
NumConsolidationThreads	連結ごとに許可される連結スレッドの数。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。詳細は、チューニング・ガイドを参照してください	範囲: 1-8 デフォルト値: 4
NumCubesLoadedBeforeCheckingLRU	最も長い間使用されていないもの(LRU)をチェックするまでにシステムがロードするデータ・サブキューブの数	範囲: 範囲にハード・リミットはありません デフォルト値: 100
NumDataLoadsAllowed	このキーは、\RunningTasks の下にあります。すべてのアプリケーション・サーバーについてアプリケーションごとに許可されるデータ・ロードの数	範囲: 1-32 デフォルト値: 8
NumEAExportsAllowed	このキーは、\RunningTasks の下にあります。すべてのアプリケーション・サーバーについてアプリケーションごとに許可される「データベースへのデータの抽出」のエクスポートの数	範囲: 1-32 デフォルト値: 8
NumEAThreads	データ処理中に使用される「データベースへのデータの抽出」のスレッドの数	範囲: 1-32 デフォルト値: 8
NumMaxDBConnections	データベース接続の最大数	範囲: 範囲にハード・リミットはありません デフォルト値: 40
NumMinutesBeforeCheckingLRU	最も長い間使用されていない(LRU)プロセスをシステムがチェックするまでの分数	範囲: 範囲にハード・リミットはありません デフォルト値: 15
NumVBScriptEngines	ルール・エンジン・プールのサイズ。	範囲: 1-64 デフォルト値: 32
PageInBatchSize	データ・レコードでのページング時のバッチ・サイズ	範囲: 範囲にハード・リミットはありません デフォルト値: 500
SQLCommandTimeout	SQL クエリーがデッドロックであると決定されるまで経過する必要がある秒数	範囲: 範囲にハード・リミットはありません デフォルト値: 60
SubCubeProgressUnits	単一のサブキューブ・プロセスが終了するまでの進行状況バー・ステップの数	範囲: 1-100 デフォルト値: 10
UseHeartBeats	このキーは、\RunningTasks の下にあります。システムがまだ応答していることを確認するために実行中のタスクにハートビート・プロセスを有効化するかどうか	0 - 無効化します 1 - 有効化します デフォルト値: 1

設定	説明	値
UseOldAlgorithmForFreeLRU	最も長い間使用されていない(LRU)使用可能なプロセスに既存のアルゴリズムを使用するかどうか 注意 この設定は、常に有効化する必要があります。	0 - アルゴリズムを無効化します 1 - アルゴリズムを有効化します デフォルト値: 1
UseOldSearchingForNoInput	アプリケーションが開いている間に NoInput キャッシュを使用するかどうか。 デフォルト設定の 0 を使用することをお勧めします。	0 - 無効化します 1 - 有効化します デフォルト値: 0
UseRecursiveUnlock	親エンティティの下の特定の子エンティティをロック解除します。 デフォルトでは、エンティティ次元および値次元全体が再帰的にロック解除されるため、データ・グリッドのデータをロック解除する際に選択する必要があるのは、最上位の親エンティティのみです。	0 - 再帰的なロック解除を無効化し、特定の子エンティティをロック解除します 1 - 再帰的なロック解除を有効化します デフォルト値: 1
UseSQLBinding	SQL バインドを使用するかどうか。 注意 この設定は、有効化する必要があります。	0 - SQL バインドを無効化します 1 - SQL バインドを使用します デフォルト値: 1
UseStickyServer	これは、次の下にあるクライアント上のクラスター・レベルのキーです: HKEY_LOCAL_MACHINE\SOFTWARE\Hyperion Solutions\Hyperion Financial Management\Client\Clusters\machinename ユーザーが後続のログオンでも同じアプリケーション・サーバーを使用する、固定サーバー・サポートを無効化するかどうか。サポートを無効化するには、Web サーバーおよび Win32 クライアントごとにこのレジストリ設定を作成する必要があります。	0 - 固定サーバー・サポートを無効化します 1 - 固定サーバー・サポートを有効化します デフォルト値: 1

用語集

Extensible Markup Language (XML) データに属性を割り当てるタグのセットで構成される言語です。スキーマに基づいて、複数のアプリケーションの間で解釈可能です。

ODBC Open Database Connectivity の略です。データベース管理システム(DBMS)の情報処理方法に関係なく、あらゆるアプリケーションにより使用されるデータベース・アクセスの方法です。

「OK」ステータス 集計ステータスの1つです。エンティティが集計済であり、階層の下にあるデータが変更されていないことを示します。

Open Database Connectivity (ODBC) 標準のアプリケーション・プログラミング・インタフェース(API)です。これにより、アプリケーションから複数のサードパーティ・データベースにアクセスできます。

PVA 「期別価額メソッド」を参照してください。

Shared Services レジストリ Shared Services リポジトリの一部です。ほとんどの EPM System 製品の EPM System 配置情報(インストール・ディレクトリ、データベース設定、コンピュータ名、ポート、サーバー、URL、依存サービス・データなど)を管理します。

XML 「Extensible Markup Language」を参照してください。

アクティビティ・レベルの承認 操作対象のデータに依存せず、アプリケーションへのユーザー・アクセス、およびアプリケーションで実行できるアクティビティのタイプを定義します。

値次元 入力値、換算値、および連結の詳細を定義する次元。

アプリケーション 1)特定のタスクまたはタスクのグループを実行するために設計されたソフトウェア・プログラムです(スプレッドシート・プログラム、データベース管理システムなど)。2)必要とされる特定の分析またはレポート、あるいはその両方に対応するために使用される、次元および次元メンバーの関連するセットです。

アプリケーションの通貨 アプリケーションのデフォルトのレポート用通貨です。

依存エンティティ 組織内の他のエンティティに属するエンティティです。

「影響」ステータス 親エンティティに連結する子エンティティの変更を示すステータスです。

エンティティ 部門、子会社、工場、地域、製品など、組織での財務報告単位となる組織単位を示す次元です。

親 直接レポートする依存エンティティを1つ以上含むエンティティです。親はエンティティであると同時に少なくとも1つのノードに関連しているため、エンティティ、ノード、および親の情報が関連付けられています。

親の調整 親に関連して子に転記される仕訳エントリーです。

会社間消去 「消去」を参照してください。

会社間照合 アプリケーション内の会社間勘定科目の対の残高を比較するプロセスです。会社間の受取勘定科目は、対応する会社間の支払い勘定科目と比較されます。一致する勘定科目は、組織の連結合計から会社間取引を消去するために使用されます。

会社間照合レポート 会社間勘定科目の残高を比較して、勘定の収支が合っているかどうかを示すレポートです。

確認レベル プロセス管理の確認ステータス・インディケータの1つです。「開始していません」、「第1パス」、「提出済」、「承認済」、「発行済」など、プロセス単位のレベルを示します。

カスタム次元 ユーザーが作成および定義する次元です。チャンネル、製品、部署、プロジェクト、または地域がカスタム次元になることがあります。

カレンダー ユーザー定義の期間、およびその関係です。暦年または会計年度は、Q1、Q2、Q3、およびQ4から構成されます。

為替レート・タイプ 為替レートの識別子です。異なるレートのタイプが使用されるのは、一定期間および年間について複数のレートが存在することがあるためです。従来より、期末時点でのレートを当期の平均レートおよび期末レートとして定義します。その他、履歴レート、予算レート、予測レートなどのレート・タイプがあります。レート・タイプは特定の時点に適用されます。

換算 「通貨換算」を参照してください。

換算レート 「為替レート」を参照してください。

勘定科目の種別 時間の経過に伴う勘定科目の値のフロー、およびその符号の振る舞いを示すプロパティです。勘定科目の種別のオプションには、費用、収益、資産、負債、および資本が含まれます。

外部認証 アプリケーションの外部に保管されたユーザー情報を使用して、Oracle EPM System 製品にログオンすることを指します。ユーザーアカウントはEPM System で管理されますが、パスワード管理およびユーザー認証は、Oracle Internet Directory(OID) や Microsoft Active Directory(MSAD)などの企業ディレクトリを使用して外部サービスで実行されます。

期別価額メソッド(PVA) 通貨換算方法の1つです。一定期間における期別の為替レート値を適用して通貨を算出します。

基本エンティティ 組織の構造の一番下に位置し、他のエンティティを持たないエンティティです。

基本通貨 日常の業務取引が行われる通貨です。

子 データベース・アウトライン内で親を持つメンバーです。

コントリビューション 子エンティティから親に追加される値です。それぞれの子は親に対するコントリビューションを持ちます。

コントロール比率 所属するグループ内でエンティティが受ける支配の程度です。

三角換算法 第3の共通通貨を介して、ある通貨から別の通貨に残高を変換する手段です。たとえば、デンマーク・クローネから英国ポンドに換算する場合、残高はクローネからユーロに換算され、その後ユーロからポンドに換算されます。

残高勘定科目 特定の時点の符号なしの値を保管する勘定科目の種別です。

資産勘定科目 勘定科目タイプの1つです。会社の資産の値を保管します。

支出勘定科目 期別の値と年次累計値を保管する勘定科目です。値が正の場合は、純利益が減ります。

システム抽出 アプリケーションのメタデータをASCII ファイルに変換する機能。

子孫 データベース・アウトラインで親の下に位置するメンバーです。たとえば、年、四半期および月を含む次元では、メンバー Qtr2 およびメンバー April がメンバー Year の子孫となります。

シナリオ データを分類するための次元です(実績、予算、予測1、予測2など)。

集計ルール 階層のノードを集計する際に実行されるルールです。親の残高が正しく連結されるように、顧客固有の適切な式を含めることができます。消去プロセスは、このルール内でコントロールできます。

出資比率 エンティティが親によって所有される程度です。

消去 組織内のエンティティ間での取引をゼロに設定(消去)するプロセスです。

仕訳(JE) シナリオと期間の勘定科目残高の借方または貸方に計上する調整の集合です。

次元 ビジネス・データを整理して値の取得や保持のために使用されるデータ・カテゴリです。通常、次元には関連するメンバーをグループ化した階層が含まれます。たとえば、年次元は多くの場合四半期、月などの期間の各単位ごとのメンバーが含まれます。

自動逆仕訳 次期に逆仕訳する調整を入力するための仕訳です。

世代 データベースでのメンバー関係を定義する階層ツリー内のレイヤーです。世代は、次元の最上位のメンバー(世代1)から子メンバーへと1世代ずつ下に配置されていきます。一意の世代名を使用すると、階層ツリー内のレイヤーを特定できます。

セル 1)多次元データベースの次元の交差を表すデータ値です;ワークシート上の行および列の交差を指します;2)管理ドメインに属するノードの論理グループです。

組織 各エンティティ、およびその関係を定義するエンティティの階層です。

祖先 その下にメンバーを含む分岐メンバーです。たとえば、メンバー Qtr2 とメンバー 2006 はメンバー April の祖先です。

属性 次元メンバーの特性です。たとえば、従業員次元メンバーには、名前、年齢、または住所の各属性がある場合があります。Product 次元のメンバーはサイズ、味などの複数の属性を持つ可能性があります。

貸借一致の仕訳 借方の合計と貸方の合計が等しい仕訳です。

置換 データ・ロードのオプションの1つです。データ・ロード・ファイルに指定された期間のすべての勘定科目からの既存の値を消去し、データ・ロード・ファイルからの値をロードします。ロード・ファイルに勘定科目が指定されていない場合、指定された期間に該当する値が消去されます。

調整勘定科目 会社間勘定科目の消去プロセスで均衡しない差額が格納される勘定科目です。

直接レート 為替レート・テーブルに入力する通貨レートです。直接レートは通貨換算に使用されます。たとえば、残高を日本円から米ドルに変換する場合、変換元通貨を日本円、変換先通貨を米ドルとして、為替レート・テーブルに期間またはシナリオのレートを入力します。

通貨換算 データベースの通貨の値を別の通貨に変換するプロセスです。たとえば、1米ドルをユーロに変換するには、ドルに為替レート(たとえば、0.923702)を乗じます(1×0.923702)。変換後のユーロの額は0.92になります。

通常仕訳 特定の期間に一度かぎりの調整を入力するための機能です。通常仕訳は貸借一致、エンティティごとに貸借一致、貸借不一致のいずれかになります。

「データなし」ステータス 集計ステータスの1つです。特定の期間と勘定科目のデータがこのエンティティに含まれていないことを示します。

データ・フォーム Web ブラウザなどのインタフェースからデータベースにデータを入力でき、データまたは関連テキストを表示して分析できるグリッド表示です。一部の次元メンバー値は固定され、データが特定の視点から表示されます。

データ・ロック 指定された基準(期間、シナリオなど)に従ってデータの変更を防ぐ機能です。

動的ビュー勘定科目 勘定科目の1つです。勘定科目の値は、表示されているデータから動的に計算されます。

動的メンバー・リスト システムにより作成される名前付きメンバー・セットです。ユーザーが定義した基準が使用されます。アプリケーションでリストが参照されるとき、リストは自動的にリフレッシュされます。次元メンバーの増減に応じて基準が自動的にリストに適用され、変更内容が反映されます。

ドリルダウン 次元の階層を使用してクエリー結果セット内をナビゲートすることです。ドリルダウンにより、ユーザーのパーспекティブが集約データから詳細に移ります。たとえば、ドリルダウンにより年と四半期の階層関係、または四半期と月の階層関係が明らかになります。

認証 安全対策としてのIDの確認です。一般に、認証はユーザー名およびパスワードに基づきます。パスワードおよびデジタル・シグネチャは認証のフォームです。

反復テンプレート 各期間に対して同一の調整を行うための仕訳テンプレートです。

「非アクティブ」ステータス エンティティの当期の連結が非アクティブになっていることを示すステータスです。

標準仕訳テンプレート 各期間に共通する調整を転記するために使用する仕訳の機能です。たとえば、共通する勘定科目ID、エンティティID、または金額を含む標準テンプレートを作成すると、これを多数の通常仕訳の基準として使用できます。

ビジネス・ルール 期待される一連の結果値を生成するためにアプリケーション内に作成される論理式または式です。

ビュー 年次累計または期別のデータ表示です。

ファイルの区切り文字 データ・ソース内のフィールドを区切る文字です(カンマ、タブなど)。

負債勘定科目 一定時点における会社の負債残高を保管する勘定科目の種別です。未払費用、買掛金勘定、長期借入金などが含まれます。

フロー勘定科目 期別と年次累計の符号なしの値を保管する勘定科目です。

変換先通貨 残高の変換後の通貨です。為替レートを入力して、変換元通貨から変換先通貨に変換します。たとえばユーロから米ドルに変換する場合、変換先の通貨は米ドルです。

変換元通貨 為替レートを使用して変換先通貨に変換される前の、値の元の通貨です。

「変更済」ステータス 集計ステータスの1つです。エンティティのデータが変更されたことを示します。

マージ データ・ロード・オプションの1つです。データ・ロード・ファイルで指定された勘定科目の値のみを消去し、データ・ロード・ファイルの値で置換します。

メタデータ データベースに保管された、またはアプリケーションにより使用されるデータのプロパティと属性を定義および説明するデータ・セットです。メタデータには、次元名、メンバー名、プロパティ、期間、およびセキュリティなどが含まれます。

メンバー・リスト 次元のメンバー、関数、他のメンバー・リストを示す名前付きのグループです。システムまたはユーザーにより定義されます。

持株会社 法的エンティティ・グループの一部であるエンティティです。グループ内のすべてのエンティティに対して直接的または間接的に投資しています。

ユーザー定義メンバー・リスト ユーザー定義による、特定の次元に含まれるメンバーの静的なセットです。

リレーショナル・データベース 関連する2次元テーブルにデータを保管するデータベースです。「多次元データベース」と対比してください。

連結 従属するエンティティからのデータを親エンティティに集約するプロセスです。たとえば、次元YearにQtr1、Qtr2、Qtr3、およびQtr4というメンバーが含まれている場合、この連結はYearになります。

連結比率 親に連結された子の値の割合です。

ロック済 ユーザーやプロセスがデータを変更するのを防ぐために、ユーザーが呼び出すプロセスです。

「ロック済」ステータス 集計ステータスの1つです。変更できないデータがエンティティに含まれていることを示します。

索引

A - Z

- ABSExp 関数, 265
 AccountTypeID 関数, 267
 AccountType 関数, 266
 AccountType 属性, 80
 AddEntityToListUsingIDs 関数, 269
 AddEntityToList 関数, 268
 AddMemberToList 関数, 269, 270
 AddMember キーワード、データ・フォーム、
164
 Alloc 関数, 270
 AllowAdjFromChildren 関数, 272
 AllowAdjFromChildren 属性, 88
 AllowAdjs 関数, 273
 AllowAdjs 属性, 88
 ApplicationName 関数, 274
 APP から XML ファイル・フォーマット, 121
 APP(メタデータ)ファイル、作成, 106
 BackgroundPOV キーワード、データ・フォー
ム, 166
 Blank キーワード、データ・フォーム, 167
 Calc1 キーワード、データ・フォーム, 169
 CalcByRow キーワード、データ・フォーム、
169
 CalcStatus 関数, 276
 CalculateExchangeRate 関数, 274
 CalculateRate 関数, 275
 Cell_Link キーワード、データ・フォーム, 170
 CellText キーワード、データ・フォーム, 171
 Clear 関数, 278
 Cn キーワード、データ・フォーム, 168
 Consol1...3 勘定科目, 102
 Consol1、Consol2、Consol3 関数, 280
 ConsolidateYTD 関数, 281
 ConsolidateYTD 属性, 89
 ConsolidationRules 属性, 92
 ConsolMethod 属性, 95
 ContainsCellText 関数, 283
 ContainsCellText 関数, 282
 Control 属性, 95
 Con 関数, 279
 Currency
 関数, 284
 属性, 98
 CustomHeaderStyle キーワード、データ・
フォーム, 172
 CustomHeader キーワード、データ・フォーム、
171
 CustomTopMember 属性, 80
 CustomTop 関数, 285
 DefaultRateForBalanceAccounts 属性, 92
 DefaultRateForFlowAccounts 属性, 92
 DataUnitItem, 286
 Decimal 関数, 287
 DefaultCurrency 属性, 92
 DefaultFreq 関数, 288
 DefaultFreq 属性, 89
 DefaultParent 関数, 289
 DefaultParent 属性
 エンティティ, 88
 カスタム, 85
 勘定科目, 80
 シナリオ, 89
 DefaultTranslate 関数, 290
 DefaultValueForActive 属性, 92
 DefaultView 関数, 291
 DefaultView 属性, 89
 DefCurrency 関数, 292
 DefCurrency 属性, 88
 DefFreqForICTrans 属性, 90
 Description 属性
 エンティティ, 88
 カスタム, 85
 勘定科目, 81
 シナリオ, 90
 通貨, 99

- 連結メソッド, 95
- DOWN 勘定科目, 102
- Down 関数, 293
- DynamicPOV キーワード、データ・フォーム、173
- Dynamic 関数, 294
- EnableMetadataSecurityFiltering 属性, 92
- EnableCustomAggr 属性, 81
- EnableDataAudit 属性
 - 勘定科目, 81
 - シナリオ, 90
- EnableProcessManagement 関数, 376
- Exp 関数, 295
- Financial Management
 - アーキテクチャ, 18
 - 構成ユーティリティ, 42
 - 次元, 18
- FormInputBoxLength キーワード、データ・フォーム, 173
- FormNumDecimals キーワード、データ・フォーム, 174
- FormRowHeight キーワード、データ・フォーム, 174
- FormScale キーワード、データ・フォーム, 175
- GetCellNoData 関数, 303
- GetCellRealData 関数, 305
- GetCellTextWithLabel 関数, 306
- GetCellText 関数, 306
- GetCellType 関数, 307
- GetCell 関数, 303
- GetItemIDs2ExtDim 関数, 311
- GetItemIDs2 関数, 310
- GetItem 関数, 309
- GetNumItems 関数, 312
- GetNumLID 関数, 313
- GetRate 関数, 314
- GetSubmissionGroup 関数, 315
- GetSubmissionPhase 関数, 315
- HideInPov キーワード、データ・フォーム, 176
- HoldingCompany 属性, 88
- Holding 関数, 316
- HS オブジェクト, 243
- ICPEntitiesAggregationWeight 属性, 92
- ICPTopMember 関数, 317
- ICPTopMember 属性, 81
- ICPWeight 関数, 318
- ICP メンバー
 - システム生成, 104
 - 編集, 105
- IDFromMember 関数, 319
- ImpactStatus 関数, 320
- Input 関数, 321
- Instructions キーワード、データ・フォーム, 177
- IsAlmostEqual 関数, 322
- IsBase 関数, 323
- IsCalculated 関数, 325
- IsCalculated 属性
 - カスタム, 85
 - 勘定科目, 81
- IsChild 関数, 326
- IsConsolidated 関数, 328
- IsConsolidated 属性, 81
- IsDescendant 関数, 329
- IsFirst 関数, 331
- IsHoldingMethod 属性, 95
- IsICP 関数, 332
- IsICP 属性
 - エンティティ, 88
 - 勘定科目, 81
- IsLast 関数, 333
- IsTransCurAdj 関数, 335
- IsTransCur 関数, 334
- IsValidDest 関数, 335
- IsZero 関数, 336
- LineItemDetailSinglePeriod キーワード、データ・フォーム, 177
- Link キーワード、データ・フォーム, 178
- List 関数, 337
- LST (メンバー・リスト)ファイル
 - 作成, 138
 - 抽出, 147
 - ロード, 146
- MaxColsForSparseRetrievalMethod キーワード、データ・フォーム, 179
- MaxCells キーワード、データ・フォーム, 178
- MaximumReviewLevel 属性, 90
- MemberFromID 関数, 340
- Member 関数, 339
- Member 属性
 - エンティティ, 88
 - カスタム, 85
 - 勘定科目, 81

- シナリオ, 90
- Method 勘定科目, 102
- Method 関数, 342
- NodeSecurity 属性, 93
- NoInput 関数, 342
- NoRound 関数, 343
- NoSuppress キーワード、データ・フォーム、179
- NumBase 関数, 344
- Number 関数, 346
- NumChild 関数, 347
- NumDecimalPlaces 属性, 82
- NumDecimals キーワード、データ・フォーム、179
- NumDescendant 関数, 349
- NumPerInGen 関数, 352
- NumPeriods 関数, 353
- OnDemandRules キーワード、データ・フォーム、180
- OpenCellTextUnit 関数, 354
- OpenDataUnitSorted 関数, 356
- OpenDataUnit 関数, 355
- OrgByPeriodApplication 属性, 93
- Override キーワード、データ・フォーム、180
- Owned 関数, 356
- Owner 関数, 357
- PrintRepeatHeadersonAllPages キーワード、データ・フォーム、183
- PCON 勘定科目, 102
- PCon 関数, 357
- PCTRL 勘定科目, 102
- PEPU 関数, 358
- PercentConsol 属性, 95
- PeriodNumber 関数, 359
- PhasedSubmissionStartYear 属性, 90
- PlugAcct 関数, 360
- PlugAcct 属性, 82
- POVOrder キーワード、データ・フォーム、182
- POWN 勘定科目, 102
- POwn 関数, 361
- PrintNumColsPerPage キーワード、データ・フォーム、182
- PrintNumRowsPerPage キーワード、データ・フォーム、183
- PVAFForBalance 関数, 362
- PVAFForFlow 関数, 363
- RateForBalance 関数, 363
- RateForFlow 関数, 364
- ReadOnly キーワード、データ・フォーム, 185
- ReportDescription キーワード、データ・フォーム、186
- ReportLabel キーワード、データ・フォーム、186
- ReportSecurityClass キーワード、データ・フォーム、187
- ReportType キーワード、データ・フォーム、187
- ReviewStatusUsingPhaseID 関数, 366
- ReviewStatus 関数, 365
- Rn キーワード、データ・フォーム, 184
- Round 関数, 367
- RowHeaderPct キーワード、データ・フォーム、187
- SCalc キーワード、データ・フォーム, 188
- Scale 関数, 368
- Scale キーワード、データ・フォーム, 189
- Scale 属性, 99
- SecurityAsPartner 関数, 369
- SecurityAsPartner 属性, 88
- SecurityClass 関数, 371
- SecurityClass 属性
 - エンティティ, 89
 - カスタム, 85
 - 勘定科目, 82
 - シナリオ, 90
- SelectablePOVList キーワード、データ・フォーム, 190
- SetCellTextWithLabel 関数, 373
- SetDataWithPOV 関数, 374
- SetData 関数, 373
- Shared Services Console, 66
- SharesOutstanding 勘定科目, 103
- SharesOwned 勘定科目, 103
- ShowDescription キーワード、データ・フォーム, 191
- ShowLabels キーワード、データ・フォーム, 191
- String キーワード、データ・フォーム, 192
- Style キーワード、データ・フォーム, 192
- SubmissionGroup 関数, 375
- SupportsProcessManagement 属性, 90
- SupportsTran 関数, 377

SuppressColHeaderRepeats キーワード、データ・フォーム, 194

SuppressInvalidColumns キーワード、データ・フォーム, 195

SuppressInvalidRows キーワード、データ・フォーム, 195

SuppressNoDataColumns キーワード、データ・フォーム, 196

SuppressNoDataRows キーワード、データ・フォーム, 196

SuppressRowHeaderRepeats キーワード、データ・フォーム, 197

SuppressZeroColumns キーワード、データ・フォーム, 197

SuppressZeroRows キーワード、データ・フォーム, 197

SwitchSignForFlow 属性, 85

SwitchSign 関数, 378

SwitchTypeForFlow 属性, 86

SwitchType 関数, 379

ToPercentControlComp 属性, 95

ToPercentControl 属性, 95

TranslationOperator 属性, 99

TransPeriodic 関数, 382

Trans 関数, 381

UD1...3 関数, 383

UDL ファイル

- IBM DB2 用の構成, 206
- Oracle 用の構成, 204
- SQL Server 用の構成, 205
- 暗号化, 206

UsedByCalcRoutine 属性, 95

UsePVAForBalanceAccounts 属性, 93

UsePVAForFlowAccounts 属性, 93

UserDefined1...3 属性

- エンティティ, 89
- カスタム, 86
- シナリオ, 90

UseSecurityForAccounts 属性, 93

UseSecurityForCustom 属性, 93

UseSecurityForEntities 属性, 93

UseSecurityForICP 属性, 93

UseSecurityForScenarios 属性, 93, 94

UsesLineItems 属性, 91

ValidationAccountEx 関数, 385

ValidationAccount 関数, 384

ValidationAccount 属性, 94

VotingOutstanding 勘定科目, 103

VotingOwned 勘定科目, 103

Web データ・フォーム(WDF)ファイル

- 作成, 161
- 抽出, 201
- ロード, 200

Workspace, 18

XBRLTags 関数, 385

XBRLTags 属性, 82

XML から APP ファイル・フォーマット, 121

ZeroViewForAdj 属性, 91

ZeroViewForNonadj 属性, 91

あ行

アクティブ勘定科目, 102

値次元, 20

値メンバー

- システム生成, 105
- 編集, 105

アプリケーション

- Web での作成, 35
- 削除, 40
- 作成, 35
- サンプル・ファイル, 52
- 閉じる, 39
- 開く, 39
- 変更, 38
- モジュールの有効化または無効化, 37

アプリケーション・セキュリティ・ファイル (SEC)

- 抽出, 76
- ロード, 72

アプリケーション設定

- 属性, 91
- 例, 109

アプリケーション・プロファイル(PER)

- カレンダーの定義, 28
- 期間単位の指定, 29
- 期間の指定, 30
- 言語の指定, 28
- 作成, 27
- 保存, 34

アプリケーション要素

- 抽出, 42
- ロード, 41

エンティティ次元, 19

エンティティの属性, 87

オブジェクト、HS, 243

か行

カスタム次元

概要, 21

作成, 32

属性, 84

メタデータの例, 113

監査

タスク, 58

データ, 61

換算のデフォルト, 242

勘定科目式、ルール, 239

勘定科目次元, 20

勘定科目属性, 80

関数

ABSExp, 265

AccountType, 266

AccountTypeID, 267

AddEntityToList, 268

AddEntityToListUsingIDs, 269

AddMemberToList, 269, 270

Alloc, 270

AllowAdjFromChildren, 272

AllowAdjs, 273

ApplicationName, 274

CalcStatus, 276

CalculateExchangeRate, 274

CalculateRate, 275

Clear, 278

Con, 279

Consol1、Consol2、Consol3, 280

ConsolidateYTD, 281

ContainsCellText, 282

ContainsCellTextWithLabel, 283

Currency, 284

CustomTop, 285

DataUnitItem, 286

Decimal, 287

DefaultFreq, 288

DefaultParent, 289

DefaultTranslate, 290

DefaultView, 291

DefCurrency, 292

Down, 293

Dynamic, 294

Exp, 295

GetCell, 303

GetCellNoData, 303

GetCellRealData, 305

GetCellText, 306

GetCellTextWithLabel, 306

GetCellType, 307

GetItem, 309

GetItemItemIDs2, 310

GetItemItemIDs2ExtDim, 311

GetNumItems, 312

GetNumLID, 313

GetRate, 314

GetSubmissionGroup, 315

GetSubmissionPhase, 315

Holding, 316

ICPTopMember, 317

ICPWeight, 318

IDFromMember, 319

ImpactStatus, 320

Input, 321

IsAlmostEqual, 322

IsBase, 323

IsCalculated, 325

IsChild, 326

IsConsolidated, 328

IsDescendant, 329

IsFirst, 331

IsICP, 332

IsLast, 333

IsTransCur, 334

IsTransCurAdj, 335

IsValidDest, 335

IsZero, 336

List, 337

Member, 339

MemberFromID, 340

Method, 342

NoInput, 342

NoRound, 343

NumBase, 344

Number, 346

NumChild, 347

NumDescendant, 349

NumPerInGen, 352

NumPeriods, 353

OpenCellTextUnit, 354

OpenDataUnit, 355

- OpenDataUnitSorted, 356
- Owned, 356
- Owner, 357
- PCon, 357
- PEPU, 358
- PeriodNumber, 359
- PlugAcct, 360
- POwn, 361
- PVAForBalance, 362
- PVAForFlow, 363
- RateForBalance, 363
- RateForFlow, 364
- ReviewStatus, 365
- ReviewStatusUsingPhaseID, 366
- Round, 367
- Scale, 368
- SecurityAsPartner, 369
- SecurityClass, 371
- SetCellTextWithLabel, 373
- SetData, 373
- SetDataWithPOV, 374
- SubmissionGroup, 375
- SupportsProcessManagement, 376
- SupportsTran, 377
- SwitchSign, 378
- SwitchType, 379
- Trans, 381
- TransPeriodic, 382
- UD1...3, 383
- ValidationAccount, 384
- ValidationAccountEx, 385
- XBRLTags, 385
- 概要, 258
- 期間
 - 移動, 31
 - コピー, 32
 - 削除, 32
- 期間次元, 19
- 期間単位, 29
- 期間別組織, 94
- 区切り文字, 21
- 経常残高勘定科目の動作, 83
- コントロール比率, 102
- 残高勘定科目の動作, 83
- 資産勘定科目の動作, 83
- システム勘定科目
 - Consol1...3 勘定科目, 102
 - DOWN 勘定科目, 102
 - Method 勘定科目, 102
 - PCON 勘定科目, 102
 - PCTRL 勘定科目, 102
 - POWN 勘定科目, 102
 - SharesOutstanding 勘定科目, 103
 - SharesOwned 勘定科目, 103
 - VotingOutstanding 勘定科目, 103
 - VotingOwned 勘定科目, 103
 - アクティブ勘定科目, 102
 - 編集, 103
- システムのユーザー
 - 接続の使用可能, 57
 - 接続の使用不可, 57
 - ユーザーの表示, 55
 - ユーザーのログアウト, 56
- システムのユーザーの接続の使用可能, 57
- システムのユーザーの接続の使用不可, 57
- シナリオ次元, 19
- シナリオ属性, 89
- シナリオ・ビューの期間単位, 29
- 収益勘定科目の動作, 83
- 出資比率, 102
- 出資比率勘定科目
 - SharesOutstanding 勘定科目, 103
 - SharesOwned 勘定科目, 103
 - VotingOutstanding 勘定科目, 103
 - VotingOwned 勘定科目, 103
- 仕訳(JLF)ファイル
 - 作成, 149
 - 抽出, 155
 - ラベルの要件, 21
 - ロード, 154
- 仕訳レポート
 - スクリプトの定義, 215
 - 例, 215
- 次元の概要, 18
- スター・スキーマ, 207
 - 更新, 211
 - 削除, 213
 - テンプレートの削除, 214
 - テンプレートの作成, 213
 - データの作成およびエクスポート, 210

さ行

参照整合性, 129

- スター・スキーマへのデータのエクスポート,
210
- セキュリティ(SEC)ファイル
 - 抽出, 76
 - ロード, 72
- セキュリティ・クラスのアクセス権, 70
- セキュリティ・レポート, 72
- 説明の要件, 21
- セル・テキスト・ラベル, 100
- 属性
 - AccountType, 80
 - AllowAdjFromChildren, 88
 - AllowAdjs, 88
 - ConsolidateYTD, 89
 - ConsolidationRules, 92
 - ConsolMethod, 95
 - Control, 95
 - Currency, 99
 - CustomTopMember, 80
 - DefaultRateForBalanceAccounts, 92
 - DefaultRateForFlowAccounts, 92
 - DefaultCurrency, 92
 - DefaultFreq, 89
 - DefaultParent
 - エンティティ, 88
 - カスタム, 85
 - 勘定科目, 80
 - シナリオ, 89
 - DefaultValueForActive, 92
 - DefaultView, 89
 - DefCurrency, 88
 - DefFreqForICTrans, 90
 - Description
 - エンティティ, 88
 - カスタム, 85
 - 勘定科目, 81
 - シナリオ, 90
 - 連結メソッド, 95
 - EnableMetadataSecurityFiltering, 92
 - EnableCustomAggr, 81
 - EnableDataAudit
 - 勘定科目, 81
 - シナリオ, 90
 - HoldingCompany, 88
 - ICPEntitiesAggregationWeight, 92
 - ICPTopMember, 81
 - IsCalculated
 - カスタム, 85
 - 勘定科目, 81
 - IsConsolidated, 81
 - IsHoldingMethod, 95
 - IsICP
 - エンティティ, 88
 - 勘定科目, 81
 - MaximumReviewLevel, 90
 - Member
 - エンティティ, 88
 - カスタム, 85
 - 勘定科目, 81
 - シナリオ, 90
 - NodeSecurity, 93
 - NumDecimalPlaces, 82
 - OrgByPeriodApplication, 93
 - PercentConsol, 95
 - PlugAcct, 82
 - Scale, 99
 - SecurityAsPartner, 88
 - SecurityClass
 - エンティティ, 89
 - カスタム, 85
 - 勘定科目, 82
 - シナリオ, 90
 - Submission Group, 82, 85
 - SupportsProcessManagement, 90
 - SwitchSignForFlow, 85
 - SwitchTypeForFlow, 86
 - ToPercentControl, 95
 - ToPercentControlComp, 95
 - TranslationOperator, 99
 - UsedByCalcRoutine, 95
 - UsePVAForBalanceAccounts, 93
 - UsePVAForFlowAccounts, 93
 - UserDefined1...3
 - エンティティ, 89
 - カスタム, 86
 - 勘定科目, 82
 - シナリオ, 90
 - UseSecurityForAccounts, 93
 - UseSecurityForCustom, 93
 - UseSecurityForEntities, 93
 - UseSecurityForICP, 93
 - UseSecurityForScenarios, 93
 - UsesLineItems
 - 勘定科目, 82

シナリオ, 91

UseSubmissionPhase, 94

ValidationAccount, 94

XBRLTags, 82

ZeroViewForAdj, 91

ZeroViewForNonadj, 91

説明

通貨, 99

た行

タスク 監査, 58

抽出

アプリケーション・セキュリティ・ファイル
(SEC), 76

仕訳ファイル(JLF), 155

データ・フォームのスク립ト(WDF), 201

メタデータ・ファイル(APP または XML),
134

メンバー・リスト(LST)ファイル, 147

ルール・ファイル(RLE), 253

直接出資比率, 102

通貨

メタデータの例, 110

ツリー・ビュー

階層の展開と縮小, 126

孤児メンバーの除去, 125

孤児メンバーの追加, 125

ノード属性の変更, 123

メンバーの位置の変更, 124

メンバーの移動, 124

メンバーの削除と除去, 125

メンバーの追加, 123

メンバーの追加と変更, 122

メンバーのレベルを上げる, 124

「転記前に照合」オプション, 429

デフォルトの換算, 242

電子メール・アラート

内部取引, 447

プロセス管理, 445

データ 監査, 61

データの抽出

スター・スキーマの更新, 211

スター・スキーマの作成, 210

スター・スキーマのテンプレート, 213

スター・スキーマのフォーマット, 208

接頭辞テーブル, 209

データベース情報の指定, 42

データ・リンク(UDL)ファイルの作成, 204

データ・フォーム

オンデマンド・ルール, 160

削除, 201

スク립ト(WDF)の作成, 161

スク립ト(WDF)の抽出, 201

スク립ト(WDF)のロード, 200

スク립トの構文, 161

データ・フォーム・デザイン

POV, 159

指示の指定, 159

フォームの詳細, 159

ヘッダー・オプション, 161

列オプション, 161

データ・フォームでのオンデマンド・ルール,
160

データ・フォームのキーワード

AddMember, 164

BackgroundPOV, 166

Blank キーワード, 167

Calc1, 169

CalcByRow, 169

Cell_Link, 170

CellText, 171

Cn, 168

CUR, FIRST, LAST, NEXT, 198

CustomHeader, 171

CustomHeaderStyle, 172

DynamicPOV, 173

FormInputBoxLength, 173

FormNumDecimals, 174

FormRowHeight, 174

FormScale, 175

HeaderOption, 175

HideInPov, 176

LineItemDetailSinglePeriod, 177

MaxColsForSparseRetrievalMethod, 179

MaxCells, 178

NoSuppress, 179

NumDecimals, 179

OnDemandRules, 180

PrintRepeatHeadersonAllPages, 183

POVOrder, 182

PrintNumColsPerPage, 182

PrintNumRowsPerPage, 183

ReadOnly, 185

ReportDescription, 186

ReportLabel, 186
 ReportSecurityClass, 187
 ReportType, 187
 Rn キーワード, 184
 RowHeaderPct, 187
 SCalc, 188
 Scale, 189
 SelectablePOVList, 190
 ShowDescription, 191
 ShowLabels, 191
 Style, 192
 SuppressColHeaderRepeats, 194
 SuppressInvalidColumns, 195
 SuppressInvalidRows, 195
 SuppressNoDataColumns, 196
 SuppressNoDataRows, 196
 SuppressRowHeaderRepeats, 197
 SuppressZeroColumns, 197
 SuppressZeroRows, 197
 上書き, 180
 指示, 177
 文字列, 192
 リンク, 178
 データ・リンク(UDL)
 IBM DB2 用の構成, 206
 Oracle 用の構成, 204
 SQL Server 用の構成, 205
 暗号化, 206
 年次元, 19
 動的メンバー・リスト, 142

な行

内部取引
 エンティティのロック, 433
 エンティティのロック解除, 433
 監視, 432
 期間を閉じる, 430
 期間を開く, 427
 照合許容差を設定する, 428
 理由コードの削除, 431
 理由コードの追加, 431
 理由コードの編集, 431
 レポート・スクリプト, 228
 内部取引照合レポート
 小数点以下の桁数の指定, 217
 スクリプトの定義, 216

スタイル・シートの選択, 217
 通貨の指定, 217
 非表示オプション, 218
 メンバー・リストの選択, 217
 内部取引次元, 20
 内部取引の理由コード, 430
 内部取引パートナ、設定, 104

は行

費用勘定科目の動作, 83
 ビュー次元, 20
 負債勘定科目の動作, 83
 フロー勘定科目の動作, 83
 プロセス管理
 提出フェーズの説明, 437
 提出フェーズの定義, 437
 提出フェーズの例, 439
 未割当ての提出グループ, 443
 プロセス管理の提出のフェーズ化, 437
 プロセス管理の提出フェーズ, 437

ま行

命名規則, 21
 メタデータでのエンティティの例, 112
 メタデータでの勘定科目の例, 111
 メタデータでのシナリオの例, 112
 メタデータの整合性, 129
 メタデータ・ファイル
 APP または XML の抽出, 134
 APP または XML のロード, 130
 アプリケーション設定の例, 109
 エンティティの例, 112
 カスタム次元の例, 113
 勘定科目の例, 111
 作成, 106
 シナリオの例, 112
 通貨の例, 110
 フォーマット, 121
 含まれない次元, 119
 メンバー, 110
 ロードの変更の表示, 133
 メタデータ・ファイル・セクション
 エンティティ次元の階層, 118
 階層, 116
 カスタム次元の階層, 119
 勘定科目次元の階層, 117

シナリオ次元の階層, 118
 連結メソッド, 115
 メタデータ・マネージャのツリー・ビュー
 階層の展開と縮小, 126
 孤児メンバーの除去, 125
 孤児メンバーの追加, 125
 ノード属性の変更, 123
 メンバーの位置の変更, 124
 メンバーの移動, 124
 メンバーの削除と除去, 125
 メンバーの追加, 123
 メンバーの追加と変更, 122
 メンバーのレベルを上げる, 124
 メタデータ・マネージャのファイル・プロパ
 ティ, 129
 メタデータ・マネージャのリスト・ビュー
 メタデータの検証, 128
 メタデータのソート, 128
 メンバーのコピー、切り取りおよび貼付け,
 127
 メンバーの削除, 128
 メンバーの追加と変更, 127
 列の表示と非表示, 128
 メンバー・リスト(LST)ファイル
 作成, 138
 抽出, 147
 動的リスト, 142
 ロード, 146
 文字制限, 21
 モジュール構成, 37

Notepad ++の使用, 248
 勘定科目式, 239
 作成, 249
 タイプ, 231
 抽出, 253
 ロード, 251
 レコード、無効なレコードの消去, 64
 レベルと説明での無効な文字, 21
 レポート・スクリプト, 215
 連結勘定科目
 Consol1...3 勘定科目, 102
 DOWN 勘定科目, 102
 Method 勘定科目, 102
 PCON 勘定科目, 102
 PCTRL 勘定科目, 102
 POWN 勘定科目, 102
 アクティブ勘定科目, 102
 連結比率, 102
 連結メソッド, 94
 連結ルール、実行順序, 241
 ロード
 アプリケーション・セキュリティ・ファイル
 (SEC), 72
 仕訳ファイル(JLF), 154
 データ・フォームのスクリプト(WDF), 200
 メタデータ・ファイル(APP または XML),
 130
 メンバー・リスト(LST)ファイル, 146
 ルール・ファイル(RLE), 251

や行

ユーザーおよびグループの割当て, 67
 ユーザー定義要素の要件, 21

ら行

ラベルの要件, 21
 リスト・ビュー
 メタデータの検証, 128
 メタデータのソート, 128
 メンバーのコピー、切り取りおよび貼付け,
 127
 メンバーの削除, 128
 メンバーの追加と変更, 127
 列の表示と非表示, 128
 ルール(RLE)ファイル