

Oracle® Hyperion Financial Management 管理者ガイド



リリース 11.2.13

F26633-16

2023年6月

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, centered within a solid red square.

ORACLE®

著者: EPM Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

目次

ドキュメントのアクセシビリティについて

ドキュメントのフィードバック

1 Financial Management について

Financial Management の機能	1-1
EPM Workspace	1-1
Financial Management のディメンション	1-2
シナリオ・ディメンション	1-2
年ディメンション	1-2
期間ディメンション	1-2
エンティティ・ディメンション	1-3
値ディメンション	1-3
勘定科目ディメンション	1-3
内部取引ディメンション	1-3
ビュー・ディメンション	1-3
カスタム・ディメンション	1-4
ユーザー定義の要素	1-4

2 アプリケーションの管理

アプリケーションの管理	2-1
アプリケーションの作成	2-2
新規アプリケーションの作成	2-2
頻度の定義	2-5
期間の編集	2-6
兄弟期間と子期間の追加	2-6
期間の削除	2-7
機能の定義	2-7
アプリケーション設定の指定	2-7

アプリケーション・モジュールの有効化または無効化	2-8
カスタム・ディメンションの作成	2-9
アプリケーション・プロファイルの保存	2-12
プロファイル・ファイルからのアプリケーションの作成	2-12
アプリケーションからのコピー	2-12
アプリケーションの変更	2-13
アプリケーションの登録	2-14
アプリケーションを開く	2-14
アプリケーションを閉じる	2-14
アプリケーション・サーバー・クラスタの変更	2-15
アプリケーション・リストの表示	2-15
アプリケーションの削除	2-15
アプリケーション要素のロード	2-16
アプリケーション要素の抽出	2-17
Financial Management アプリケーションの複製	2-17
アプリケーションのインポート	2-18
Oracle Database のデータベース・リンクの作成	2-20
SQL Server データベース用のリンク・サーバーの作成	2-20
システム・メッセージの使用	2-22
システム・メッセージの表示	2-22
システム・メッセージの削除	2-22
システム・メッセージの詳細文字列	2-23
アプリケーション・アクセス権の管理	2-24
ユーザーのログアウト	2-24
接続の使用不可と使用可能	2-24
アプリケーションの接続ステータスの表示	2-25
システム・ユーザーの管理	2-25
ユーザーの表示	2-26
選択したユーザーまたはすべてのユーザーのログアウト	2-26
サーバーとアプリケーションの管理	2-27
管理モードの有効化および無効化	2-27
サーバーの同期化	2-28
タスクの監査	2-28
データの監査	2-31
実行中のタスクのモニタリング	2-32
実行中のタスクの表示	2-33
実行中のタスクの停止	2-34
実行中のタスクのリフレッシュ	2-34
無効なレコードのスキャンとクリア	2-35

HFM インサイトを使用したシステム・ステータスのモニタリング	2-36
HFM インサイトへのアクセス	2-36
HFM インサイトのメイン・ダッシュボード	2-36
HFM インサイトでのアプリケーションのフィルタ	2-37
HFM インサイトのアプリケーションの詳細	2-37

3 アプリケーション・セキュリティの管理

アプリケーション・セキュリティに関する考慮事項	3-2
Financial Management からの Shared Services Console の起動	3-2
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	3-3
セキュリティ・クラスの設定	3-3
セキュリティ・クラスの作成	3-4
セキュリティ・クラスの削除	3-4
セキュリティ・クラスの選択	3-5
セキュリティ・クラスへのユーザー・アクセス権の割当て	3-5
電子メール・アラートの設定	3-7
セキュリティ・レポートの実行	3-7
アプリケーション・セキュリティのロード	3-8
セキュリティ情報のクリアとロード	3-10
セキュリティ情報をクリアする前に行っておくべきタスク	3-11
セキュリティ情報をクリアした後で行うタスク	3-11
アプリケーション・セキュリティの抽出	3-12

4 メタデータの管理

勘定科目の定義	4-1
勘定科目タイプの動作	4-5
動的勘定科目の定義	4-7
カスタム・メンバーの定義	4-8
エンティティ・メンバーの定義	4-10
シナリオ・メンバーの定義	4-11
アプリケーション設定の定義	4-15
期間別組織	4-18
連結メソッドの定義	4-18
連結メソッドの使用方法	4-20
手動での連結メソッドの割当て	4-21
POWN または POWNMIN 連結メソッドの使用	4-21
通貨の定義	4-23

セル・テキスト・ラベルの定義	4-24
システム生成の勘定科目	4-27
連結勘定科目	4-27
出資比率勘定科目	4-28
システムで生成された勘定科目の編集	4-28
内部取引パートナーの設定	4-29
システムで生成された ICP メンバーの編集	4-30
システム生成の値メンバーの編集	4-30
セキュリティに基づくメタデータのフィルタ	4-31
APP フォーマットのメタデータ・ファイルの作成	4-32
ファイル・フォーマット	4-33
バージョン	4-34
アプリケーション設定	4-34
通貨	4-36
メンバー	4-37
勘定科目	4-37
シナリオ	4-38
エンティティ	4-39
カスタム	4-39
値	4-40
内部取引パートナー	4-41
連結メソッド	4-42
階層	4-42
勘定科目の階層	4-43
シナリオ階層	4-45
エンティティ階層	4-45
カスタム階層	4-45
メタデータ・ファイルに含まれないディメンション	4-46
メタデータの参照整合性	4-47
メタデータの参照整合性チェック	4-47
メタデータ・ログ・ファイルの参照整合性エラー	4-47
メタデータのマージ・ユーティリティの使用	4-48
メタデータのロード	4-49
メタデータ・ロードの変更の表示	4-52
メタデータの抽出	4-53

5 メンバー・リストの管理

メンバー・リスト・ファイルの作成	5-2
------------------	-----

EnumMemberLists	5-2
EnumMembersInList	5-3
動的メンバー・リスト	5-6
動的 POV メンバー・リスト	5-7
メンバー・リストのロード	5-10
メンバー・リストの抽出	5-11
システム生成のリストとディメンション	5-11

6 仕訳の管理

仕訳ファイルの作成	6-1
ファイル・フォーマット・セクション	6-2
バージョン・セクション	6-2
仕訳グループ・セクション	6-3
標準セクション	6-3
繰返しセクション	6-3
ヘッダー・セクション	6-4
仕訳のロード	6-6
仕訳の抽出	6-8

7 データ・フォームの管理

フォーム・デザイナーでのデータ・フォームの作成	7-1
視点の設定	7-2
フォームの詳細の指定	7-3
データ・フォームのオンデマンド・ルールの指定	7-3
フォームの行および列オプションの指定	7-4
フォーム・ヘッダーの指定	7-5
データ・フォーム・スクリプトの使用	7-5
AddMember	7-8
BackgroundPOV	7-10
Blank	7-11
Cn	7-12
Calc1	7-13
CalcByRow	7-13
Cell_Link	7-14
CellText	7-15
CustomHeader	7-15
CustomHeaderStyle	7-16
DynamicPOV	7-17

FormInputBoxLength	7-17
FormNumDecimals	7-18
FormRowHeight	7-18
FormScale	7-19
HeaderOption	7-19
HideInPov	7-20
Instructions	7-20
LineItemDetailSinglePeriod	7-21
Link	7-21
MaxCells	7-22
MaxColsForSparseRetrievalMethod	7-22
NoSuppress	7-23
NumDecimals	7-23
OnDemandRules	7-23
Override	7-24
POVOrder	7-25
PrintNumDataColsPerPage	7-26
PrintNumRowsPerPage	7-26
PrintRepeatHeadersonAllPages	7-27
Rn	7-27
ReadOnly	7-29
ReportDescription	7-29
ReportLabel	7-30
ReportSecurityClass	7-31
ReportType	7-31
RowHeaderPct	7-31
SCalc	7-32
Scale	7-33
SelectablePOVList	7-33
ShowDescriptions	7-34
ShowLabels	7-35
String	7-35
Style	7-36
SuppressColHeaderRepeats	7-38
SuppressInvalidCols	7-38
SuppressInvalidRows	7-39
SuppressNoDataCols	7-39
SuppressNoDataRows	7-40
SuppressRowHeaderRepeats	7-40
SuppressZeroCols	7-40
SuppressZeroRows	7-41

相対期間の使用	7-41
競合する属性の優先順位	7-42
データ・フォームの編集	7-43
データ・フォームのロード	7-43
データ・フォームの抽出	7-44
データ・フォームの削除	7-45

8 データベースへのデータの抽出

データ・ソース名(DSN)の構成	8-1
スター・スキーマ	8-2
スター・スキーマのフォーマット	8-4
接頭辞表	8-4
データの作成とスター・スキーマへのエクスポート	8-5
スター・スキーマの更新	8-7
スター・スキーマの削除	8-8
スター・スキーマのテンプレートの作成	8-9
スター・スキーマのテンプレートの削除	8-9

9 レポートの定義

仕訳レポート・スクリプトの定義	9-1
内部取引照合レポート・スクリプトの定義	9-2
内部取引照合レポートのメンバー・リストの選択	9-3
内部取引照合レポートの勘定科目の選択	9-3
内部取引照合レポートの小数点以下の桁数の指定	9-3
内部取引照合レポート用スタイル・シートの選択	9-3
内部取引照合レポートの通貨の指定	9-3
内部取引照合レポートの非表示オプション	9-4
内部取引照合レポート・スクリプトのキーワード	9-5
ReportType	9-5
ReportDescription	9-5
StyleSheet	9-5
POV	9-6
エンティティ	9-6
パートナ	9-6
AccountEntity と AccountPartner	9-7
SuppressIfMatch	9-10
SuppressReversedTransactions	9-11
SuppressDetails	9-11

MatchingTolerance	9-11
SuppressCustoms	9-12
ScaleFactor	9-12
Decimal	9-13
DisplayLabels	9-13
DisplayDescriptions	9-13
DisplayPlugElimAccts	9-14
GroupByCustom	9-14
内部取引レポート・スクリプトの定義	9-14

10 ルールの管理

ルールのタイプ	10-1
ルールに関する注意事項	10-4
計算コマンド	10-5
計算の強制	10-5
計算コマンドを含む計算ルール	10-5
連結コマンドを使用した計算ルール	10-6
現在のディメンション・メンバー	10-9
勘定科目式	10-10
関数によるデータの自動クリア	10-11
エラー・メッセージ	10-12
連結時のルールの実行	10-12
デフォルトの換算方法	10-13
Financial Management のオブジェクト	10-14
ルールでの VBScript の使用	10-15
VBScript 演算子	10-15
VBScript 文	10-16
VBScript のキーワード	10-16
VBScript 関数	10-17
日付と時間の関数	10-17
フォーマット関数	10-17
変換関数	10-17
算術関数	10-18
配列関数	10-18
文字列関数	10-18
その他の関数	10-19
VBScript オブジェクト	10-19
よく使用するルール	10-20
データの再使用	10-20

値の計算による勘定科目の設定	10-21
条件付きのルール	10-21
ディメンションのメンバーのテスト	10-22
複数のメンバーのテスト	10-23
メンバーごとに異なるアクションを実行	10-23
すべての勘定科目の期首残高の設定	10-24
ルール・ファイルの作成	10-25
ルールのロード	10-27
ルールの抽出	10-29
ルール・プロファイリング	10-29
ルール・ファイルへのプロファイル情報の追加	10-29
ルール・プロファイリングの開始と停止	10-30
ルール・プロファイル結果の表示	10-31
ルール・ビュー	10-31
エンティティ・ビュー	10-32
ルール・プロファイルの削除	10-33

11 ルール関数

関数の概要	11-1
ABSExp	11-9
AccountType	11-10
AccountTypeID	11-11
AddEntityToList	11-12
AddEntityToListUsingIDs	11-13
AddMemberToList	11-13
AddMemberToListUsingIDs	11-13
Alloc	11-14
AllowAdjFromChildren	11-16
AllowAdjs	11-17
ApplicationName	11-18
CalculateExchangeRate	11-18
CalculateRate	11-19
CalcStatus	11-19
CellTextUnitItem	11-22
Clear	11-24
Con	11-25
Consol1、Consol2、Consol3	11-26
ConsolidateYTD	11-27
ContainsCellText	11-27

ContainsCellTextWithLabel	11-29
Currency	11-30
CustomTop	11-30
DataUnitItem	11-31
Decimal	11-32
DefaultFreq	11-33
DefaultParent	11-34
DefaultTranslate	11-35
DefaultView	11-36
DefCurrency	11-37
Down	11-38
Dynamic	11-39
Exp	11-40
ディメンションの交差に関する考慮事項	11-42
期間および年のキーワード	11-42
算術計算	11-43
Exp 内での他の関数の配置	11-44
複数の勘定科目の同時設定	11-44
Exp およびディメンションの交差に関する考慮事項	11-45
すべての交差が有効な場合	11-45
交差が無効な場合	11-46
右側に 1 つの有効なメンバーがある場合	11-47
GetCell	11-48
GetCellNoData	11-48
GetCellRealData	11-50
GetCellText	11-51
GetCellTextWithLabel	11-51
GetCellType	11-52
GetCustomLabelArray	11-53
GetItem	11-54
GetItemIDs2	11-55
GetItemIDs2ExtDim	11-56
GetNumItems	11-57
GetNumLID	11-58
GetRate	11-59
GetSubmissionGroup	11-60
GetSubmissionPhase	11-60
Holding	11-60
ICPTopMember	11-61
ICPWeight	11-62
IDFromMember	11-63

ImpactStatus	11-64
Input	11-65
IsAlmostEqual	11-66
IsBase	11-67
IsCalculated	11-69
IsChild	11-70
IsConsolidated	11-72
IsDescendant	11-73
IsFirst	11-75
IsICP	11-76
IsLast	11-77
IsTransCur	11-78
IsTransCurAdj	11-78
IsValidDest	11-79
IsZero	11-80
List	11-82
Member	11-83
MemberFromID	11-84
Method	11-86
NoInput	11-86
NoRound	11-87
NumBase	11-88
Number	11-90
NumChild	11-91
NumCustom	11-93
NumDescendant	11-93
NumPerInGen	11-96
NumPeriods	11-97
OpenCellTextUnit	11-97
OpenDataUnit	11-99
OpenDataUnitSorted	11-99
Owned	11-100
Owner	11-100
PCon	11-101
PEPU	11-101
PeriodNumber	11-102
PlugAcct	11-103
POwn	11-104
PVAForBalance	11-105
PVAForFlow	11-106
RateForBalance	11-107

RateForFlow	11-107
ReviewStatus	11-108
ReviewStatusUsingPhaseID	11-109
Round	11-110
Scale	11-111
SecurityAsPartner	11-112
SecurityClass	11-113
SetCellTextWithLabel	11-115
SetData	11-115
SetDataWithPOV	11-116
SubmissionGroup	11-117
SupportsProcessManagement	11-118
SupportsTran	11-119
SwitchSign	11-119
SwitchType	11-121
Trans	11-122
TransPeriodic	11-123
UD1...3	11-124
ValidationAccount	11-126
ValidationAccountEx	11-127
XBRLTags	11-127

12 カスタム関数

管理レポート関数	12-2
Average	12-2
Cumulative	12-5
Difference	12-8
DSO - 売上債権回転日数	12-11
Opening	12-14
Rate	12-17
ビジネス・ルール関数	12-24
Custom_Alloc	12-24
Increase_Decrease	12-28
Pro_Rata_Ratio	12-31
Spread	12-33
Units_Rates	12-36

13 Calculation Manager を使用したルールの作成

Calculation Manager のセキュリティ役割	13-1
Calculation Manager でのアプリケーションの操作	13-1
Calculation Manager へのルールの移行	13-2
関数セレクトタでの VB 関数のサポート	13-2
Financial Management の特別な VB Script 関数	13-3

14 内部取引の管理

内部取引の設定	14-1
内部取引の期間を開く	14-1
照合許容差を設定する	14-2
「転記前に照合」オプションの設定	14-3
内部取引の期間を閉じる	14-3
理由コードの管理	14-4
理由コードの追加	14-4
理由コードの編集	14-5
理由コードの削除	14-5
内部取引のモニタリング	14-5
エンティティのロックとロック解除	14-7
内部取引の要約の表示	14-8

15 プロセス管理の提出フェーズの管理

提出フェーズの定義	15-1
送信グループの設定	15-2
送信グループおよびフェーズの例	15-3
送信グループのフェーズへの割当て	15-5
未割当ての送信グループの表示	15-6

16 電子メール・アラートの管理

プロセス管理のアラートの設定	16-1
内部取引のアラートの設定	16-2

A 構成設定

使用可能な構成設定	A-2
構成設定の変更	A-8

値の上書き	A-9
設定表の表示の変更	A-9
設定の検索	A-9
有効な設定の表示	A-10
設定のエクスポート	A-10
設定の削除	A-10

B パフォーマンスの最適化

パフォーマンスの概要	B-1
一般的な EPM インストール・ディレクトリの参照	B-1
Financial Management のレコードおよびサブキューブ	B-2
Financial Management のチューニングに関する推奨事項	B-3
パフォーマンスの問題の診断	B-4
モニタリング・ツールの使用	B-4
リモート診断エージェント(RDA)の使用	B-6
参照アプリケーションの使用	B-7
オペレーティング・システム・パラメータのチューニング	B-7
Windows パラメータのチューニング	B-7
Web サーバーのチューニング	B-8
HFM Web のチューニング	B-8
HFM Web のタイムアウト・パラメータ	B-8
HFM Web のチューニング・パラメータ	B-10
Financial Management アプリケーションのチューニング	B-12
一般的にチューニングされた Financial Management の設定	B-12
MaxNumDataRecordsInRAM	B-12
MinDataCacheSizeInMB	B-13
MaxDataCacheSizeInMB	B-13
MaxNumCubesInRAM	B-13
IcmSystemReportTranslationMode	B-13
NumConsolidationThreads	B-14
MaxNumConcurrentConsolidations	B-15
NumThreadsToUseWhenUpdatingCalcStatusSystemWasChanged	B-16
大規模アプリケーション用の Financial Management のメモリー設定	B-16
アプリケーション固有の設定	B-16
Financial Management アプリケーション・サーバーのチューニング	B-18
Financial Management のチューニング例	B-18
アプリケーション・データベースのメンテナンス	B-19
基本的な設計上の考慮事項	B-19
Financial Management 用の Oracle 11g データベースのチューニング	B-21

はじめに	B-21
一般的なパフォーマンスの問題	B-21
Oracle 11g Database のチューニング・ガイドライン	B-22
Oracle 初期化パラメータ	B-22
CURSOR_SHARING	B-22
MEMORY_TARGET	B-22
MEMORY_MAX_TARGET	B-23
SGA_TARGET	B-23
SGA_MAX_SIZE	B-23
PGA_AGGREGATE_TARGET	B-23
LOG_BUFFER	B-24
OPTIMIZER_MODE	B-24
OPTIMIZER_INDEX_COST_ADJ	B-24
PROCESSES	B-24
SESSIONS	B-25
TRANSACTIONS	B-25
OPEN_CURSORS	B-25
SESSION_CACHED_CURSORS	B-25
TRACE_ENABLED	B-25
STATISTICS_LEVEL	B-26
TIMED_STATISTICS	B-26
TIMED_OS_STATISTICS	B-26
Oracle Database リリース 11g のメモリー設定を決定する方法	B-26
合計メモリーのサイズ設定(MEMORY_TARGET)	B-26
SGA のサイズ設定(SGA_TARGET)	B-28
PGA のサイズ設定(PGA_AGGREGATE_TARGET)	B-28
LOG_BUFFER のサイズ設定	B-28
Oracle Database リリース 11g のプロセス数を計算する方法	B-29
その他の考慮事項	B-30
共有サーバーと専用サーバー	B-30
オンライン REDO ログ・ファイルのサイズ	B-30
表領域とセグメントの断片化	B-31
インデックスの断片化	B-31
機能 DEFERRED_SEGMENT_CREATION の無効化	B-31
機能 DEFERRED_SEGMENT_CREATION の無効化	B-31
定期的なメンテナンスおよびチューニング・プラン	B-32
よくある質問	B-32

ドキュメントのアクセシビリティについて

オラクル社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。

Oracle サポートへのアクセス

サポートをご契約のお客様には、My Oracle Support を通して電子支援サービスを提供しています。詳細は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>、聴覚に障害があるお客様は <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> を参照してください。

ドキュメントのフィードバック

このドキュメントに対するフィードバックを送るには、Oracle Help Center トピックのページの下部にあるフィードバック・ボタンをクリックします。epmdoc_ww@oracle.com に電子メールを送信することもできます。

1

Financial Management について

次も参照:

- [Financial Management の機能](#)
- [EPM Workspace](#)
- [Financial Management のディメンション](#)
- [ユーザー定義の要素](#)

Financial Management の機能

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

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

EPM Workspace

Oracle Hyperion 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、EPM Workspace で実行できるタスクの詳細は、『Oracle Hyperion Enterprise Performance Management Workspace ユーザー・ガイド』およびオンライン・ヘルプを参照してください。

Financial Management のディメンション

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

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

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

一部のデータ・ディメンションでは階層を定義できません。たとえば、年ディメンションでは階層をサポートしていません。

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

次の項では、システム定義のディメンションについて説明します。

シナリオ・ディメンション

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

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

年ディメンション

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

期間ディメンション

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

エンティティ・ディメンション

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

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

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

値ディメンション

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

勘定科目ディメンション

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

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

内部取引ディメンション

内部取引ディメンションは、勘定科目に存在する内部取引のすべての残高を表します。これは、勘定科目ディメンションと任意のカスタム・ディメンションとの組合せに使用される予約ディメンションです。Oracle Hyperion Financial Management では、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して消去できます。また、内部取引照合レポートを実行して内部取引を表示できます。

ビュー・ディメンション

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

カスタム・ディメンション

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

ユーザー定義の要素

Oracle Hyperion 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。

ノート: メタデータには ASCII 文字のみが許可されます。

表 1-1 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			

表 1-1 (続き) 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、ドイツ語の大文字、エスツェットなどの特殊文字は使用できません。 <p>ノート: 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	<ul style="list-style-type: none"> アンパサンド(&)は使用できません。 スペースは使用できません。
メンバーのラベル	1	80	<p>一意にする必要があります。空白も含め、80 文字まで入力できますが、1 文字目は空白にできません。次の文字は使用できません:</p> <ul style="list-style-type: none"> アンパサンド(&) 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(/) マイナス記号(-) シャープ記号(#) ピリオド(.) プラス記号(+) セミコロン(:) <p>ノート: エンティティ名を「ALL」にすることはできません。</p>
メンバーの説明	0	40	<p>アンパサンド(&)は使用できません。</p> <p>ノート: 説明モードで作業を選択する場合は、メンバーの説明でピリオド(.)を使用しないでください。</p>
別名のラベル	0	80	<p>アンパサンド(&)は使用できません。</p>
セキュリティ			

表 1-1 (続き) ユーザー定義の要素

要素	最小の長さ	最大の長さ	制限
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(-) • シャープ記号(#) • ピリオド(.) • プラス記号(+) • セミコロン(;) • スラッシュ(/)
仕訳 仕訳のラベル	1	20	次の文字は使用できません: <ul style="list-style-type: none"> • アスタリスク(*) • アット記号(@) • カンマ(,) • 中カッコ({}) • 二重引用符"" • スラッシュ(/) • マイナス記号(-) • シャープ記号(#) • ピリオド(.) • プラス記号(+) • セミコロン(;)
仕訳の説明	0	255	なし
仕訳グループ	0	30	なし
仕訳のライン・アイテムの説明	0	50	なし
ロード/抽出			

表 1-1 (続き) ユーザー定義の要素

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

表 1-1 (続き) ユーザー定義の要素

要素	最小の長さ	最大の長さ	制限
ドキュメント名 (フォルダおよびレポート名を含む)	1	16	<p>次の文字は、ドキュメント名には使用できません:</p> <ul style="list-style-type: none"> • アスタリスク(*) • アット記号(@) • 円記号(バックスラッシュ) (\) • 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(<) • 線() • シャープ記号(#) • 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>ノート: ドキュメント名の末尾または先頭は空白文字にできません。</p>

2

アプリケーションの管理

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

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

アプリケーションは、アプリケーション・サーバーで実行します。アプリケーション・サーバーのクラスターを設定して、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 インストレーションおよび構成ガイド*を参照してください。

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

システムのユーザー

- ユーザーのリスト
- ユーザーのログアウト

サーバーとアプリケーションの管理

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

システム・メッセージ

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

管理者グループへのユーザーの割当てについては、*Oracle Enterprise Performance Management System ユーザー・セキュリティ管理ガイド*を参照してください。

アプリケーションの管理

アプリケーションの定義は、次のステップを使用して行います。

1. 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・サーバーを指定し、アプリケーション・ラベルと説明、カレンダー、言語および期間単位のプロパティを指定します。[アプリケーションの作成](#)を参照してください。
2. 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。[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。

3. 勘定科目、エンティティ、シナリオ、カスタム・ディメンション、アプリケーション設定、連結メソッド、通貨などのアプリケーションのメタデータを定義します。[メタデータの管理](#)を参照してください。
4. 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。

アプリケーションの作成

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

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

アプリケーションの作成には、いくつかの方法があります。

- 新規アプリケーションが作成されます。このオプションでは、アプリケーションの作成ウィザードが起動されます。アプリケーション・カレンダー、モジュール構成およびカスタム・ディメンションを定義できます。[新規アプリケーションの作成](#)を参照してください。
- アプリケーションからコピーします。新規アプリケーションを作成するには、アプリケーションからアプリケーション・カレンダー、モジュール構成およびカスタム・ディメンションを表示し、アプリケーション作成ウィザードを使用して変更します。[アプリケーションからのコピー](#)を参照してください。
- プロファイル・ファイルからアプリケーションを作成します。カレンダーを表示して変更することもできます。[プロフィール・ファイルからのアプリケーションの作成](#)を参照してください。

新規アプリケーションの作成

新規アプリケーションを作成するには、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。

アプリケーションを作成するには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「アプリケーション」タブで、「アクション」、「新規」の順に選択します。
アプリケーション作成ウィザードが起動されます。
3. **アプリケーションのオプション**から、「**新規アプリケーション**」を選択し、「次」をクリックします。
4. 「**アプリケーション・プロパティ**」セクションで、「名前」に新規アプリケーションの名前を入力します。

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

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

▲ 注意:

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

5. 「説明」に、アプリケーションの説明を入力します。
アプリケーションの説明にはスペースを含む最大 255 文字を使用できます。アンパサンド(&)は使用できません。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. 「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
7. 「クラスタ」リストから、新規アプリケーションが稼働するアプリケーション・サーバー・クラスタを選択します。
サーバー・クラスタが表示されない場合は、サーバー・クラスタの登録が必要な場合があります。Oracle Enterprise Performance Management System インストレーションおよび構成ガイドを参照してください。
8. 「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。

✎ ノート:

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

9. 「言語」では、アプリケーション全体で使用される説明に対して、10 までの言語を指定できます。言語名を区切るにはカンマを使用します。言語名ごとに、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。

▲ 注意:

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

10. 年、期間およびビュー・ディメンションを使用してカレンダーを定義します。年ディメンションのメンバーは、アプリケーションの開始年と年の合計を定義します。期間ディメンションのメンバー階層は、基本レベルと要約レベルの期間を定義します。ビュー・ディメン

ションでは、年(YTD)、半年(HYTD)、四半期(QTD)、月(MTD)などの、期間累計ビューを定義します。

▲ 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開始年、期間の説明またはラベルを変更することはできません。

- 「年」セクションで、「開始年」にカレンダーの開始年を入力します。
- 「年数」に、アプリケーションに含める年数の合計を入力します。

✎ ノート:

アプリケーション内の年数は、既存のアプリケーションで変更可能な唯一のプロファイル定義です。[アプリケーションの変更](#)を参照してください。

11. 「期間」セクションで、含める期間を選択します。

- 月
含める期間(半年、四半期または三半期)を選択します
「開始月」ドロップダウン・リストから、カレンダーの最初の月を選択します。
- 四半期
オプション: 半年を含める場合に選択します。
- カスタムカスタム・カレンダーを定義する場合は、期間のラベルの接頭辞と基本期間数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。
 - 「期間のラベルの接頭辞」に、含める期間の接頭辞を入力します。
ラベルには、スペースも含めて 10 文字まで使用できます。
 - 「基本期間数」に、年内の期間数を入力します。
期間数に 10 より小さい任意の数値、ラベルの接頭辞に **NewPeriod** を入力すると、これらの期間は **NewPeriod1**(以降同様)として階層に追加されます。期間数が 10 以上 100 未満である場合、期間は **NewPeriod01** から始まります。期間数が 100 以上 1000 未満である場合、期間は **001** から始まり、以降同様になります。

✎ ノート:

手動カレンダーを定義する場合は、期間のラベルの接頭辞と基本期間数を空白のままにする必要があります。期間単位と期間は、必要に応じて後で入力できます。

12. 「次」をクリックします。

頻度の定義

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(YTD、HYTD、QTD、MTD)は、カレンダーを定義するときに選択した期間に基づいて作成されます。期間単位は、追加、変更、削除できます。また、各期間単位にわかりやすいラベルを入力したり、あらかじめ定義した各言語で表示したりできます。

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

ノート:

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。

カレンダーを手動で定義するように選択した場合、期間単位のグリッドは空になるため、必要なビューとその説明を入力する必要があります。期間のディメンションのレベルごとに、1つの期間単位を入力する必要があります。

注意:

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

期間単位を定義するには:

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

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

ノート:

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

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

 ヒント:

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

3. 「次」をクリックします。

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。

 ノート:

基本期間数は0よりも大きい必要があります。

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

 ノート:

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

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

期間を入力して編集するには:

1. 期間階層から期間を選択し、期間ラベルまたは説明を入力または編集します。
期間のラベルには、40文字まで入力できます。期間の説明には、80文字まで入力できます。スペースも1文字としてカウントされることに注意してください。
2. 必ず各期間の説明を付けてください。
説明を付けずに後で追加する場合は、アプリケーション・プロファイルを変更して、そのプロファイルを使用するすべてのアプリケーションを再度作成する必要があります。

兄弟期間と子期間の追加

期間の階層に、兄弟期間や子期間を1つ以上追加できます。

子期間または兄弟期間を1つ追加するには:

1. 1つの子期間または兄弟期間を追加する期間をハイライトします。

2. 次のいずれかのオプションを選択します:

- 「子の挿入」ボタン  をクリックするか、「アクション」、「子の挿入」の順に選択します。
- 「兄弟の挿入」ボタン  をクリックするか、「アクション」、「兄弟の挿入」の順に選択します。

3. 新しい兄弟期間または子期間の名前を入力します。

兄弟期間または子期間を複数追加するには:

1. 複数の子期間または兄弟期間を追加する期間をハイライトします。

2. 「多数の挿入」ボタン  をクリックするか、「アクション」、「多数の挿入」の順に選択します。

3. 追加する期間の数を入力します。

4. 新しい期間のラベルの接頭辞を入力し、「OK」をクリックします。

期間数に 10 より小さい任意の数値、ラベルの接頭辞に NewPeriod を入力すると、これらの期間は NewPeriod1(以降同様)として階層に追加されます。期間数が 10 以上 100 未満である場合、期間は NewPeriod01 から始まります。期間数が 100 以上 1000 未満である場合、期間は 001 から始まり、以降同様になります。

5. 期間の追加が終了したら、「次」をクリックします。

期間の削除

期間を階層から削除すると、その期間の子孫もすべて削除されます。

ノート:

[Year]期間は削除できません。

期間を削除するには:

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

機能の定義

「機能」画面には、有効化されたアプリケーション・モジュールに基づいて、アプリケーション設定とデフォルト値が表示されます。任意のデフォルト値を変更できます。

アプリケーション設定の指定

アプリケーション設定の詳細は、[アプリケーション設定の定義](#)を参照してください。

次の設定を指定できます。

- **アプリケーションの通貨** - アプリケーションで使用する通貨。通貨は指定することができ、指定された通貨は通貨ディメンションに追加されます。[通貨の定義](#)を参照してください。
- **残高勘定のレート勘定** - BALANCE 勘定科目のレート勘定。勘定科目は指定することができ、勘定科目ディメンションに追加されます。[勘定科目の定義](#)を参照してください。
- **フロー勘定のレート勘定** - FLOW 勘定科目のレート勘定。勘定科目は指定することができ、勘定科目ディメンションに追加されます。[勘定科目の定義](#)を参照してください。
- **残高勘定の換算メソッド** - BALANCE 勘定科目の換算メソッド。[勘定科目の定義](#)を参照してください。
- **フロー勘定の換算メソッド** - FLOW 勘定科目の換算メソッド。[勘定科目の定義](#)を参照してください。

アプリケーション設定を指定するには:

1. 「機能」画面で、次のアプリケーション設定の値を指定するか、デフォルト値を使用します。
 - **アプリケーションの通貨** - アプリケーションの通貨を入力します。すべての通貨が一覧表示され、選択できます。たとえば、「US」と入力すると、「USD (US ドル)」通貨が表示され、選択できます。
 - **残高勘定のレート勘定** - BALANCE 勘定科目に使用するレート勘定を入力します。
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。[勘定科目の定義](#)を参照してください。
 - **フロー勘定のレート勘定** - FLOW 勘定科目に使用するレート勘定を入力します。
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。[勘定科目の定義](#)を参照してください。
 - **残高勘定の換算メソッド** - ドロップダウン・リストから、BALANCE 勘定科目の換算メソッドを選択します。
 - PVA - 期別値換算メソッドに対応
 - VAL - 為替レート値換算メソッドを使用
 - **フロー勘定の換算メソッド** - ドロップダウン・リストから、FLOW 勘定科目の換算メソッドを選択します。
 - PVA - 期別値換算メソッドに対応
 - VAL - 為替レート値換算メソッドを使用
2. アプリケーション・モジュールを有効化または無効化するには、[アプリケーション・モジュールの有効化または無効化](#)を参照してください。

アプリケーション・モジュールの有効化または無効化

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

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

 **ノート:**

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

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

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

アプリケーション・モジュールを有効化または無効化するには:

1. 「機能」画面で、アプリケーション・モジュールを有効化する場合は「はい」、無効化する場合は「いいえ」を選択します(あるいはデフォルト値を使用します)。

- **プロセス・コントロールの有効化**

 **ノート:**

プロセス・コントロールを有効にしないと、提出フェーズは有効にできません。

- **提出フェーズの有効化**
- **出資比率の管理の有効化**
- **仕訳の有効化**
- **データ管理の有効化**
- **監査タスクの有効化**
- **内部取引の有効化**
- **エクイティ・ピックアップの有効化**

2. 「次」をクリックします。

カスタム・ディメンションの作成

ディメンションとディメンションのセキュリティ・プロパティを構成するには、「ディメンション」画面を使用します。カスタム・ディメンションを追加または削除したり、カスタム・ディメンションの名前および別名を指定できます。

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

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

- アプリケーションに、一意のカスタム・ディメンション名とディメンション別名を入力する必要があります。名前と別名の両方が一意である必要があります。たとえば、**PROD** というディメンション名を使用している場合、ディメンション別名を **PROD** にはできません。カスタム・ディメンション名も通貨の名前と同じすることはできません。
- ディメンション名は最大 10 文字です。
- ディメンション別名は最大 20 文字です。
- ディメンション名と別名に、スペースや空白を使用することはできません。
- カスタム・ディメンションに、次のいずれかのサイズを選択する必要があります: 「小」、「中」、「大」。

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

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

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

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

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

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

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

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

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

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

実用的には、カスタム・ディメンションに「中」を使用することをお勧めします。

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

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

次の例は、カスタム・ディメンションのサンプルを示しています。

カスタム・ディメンション名	カスタム・ディメンション別名	カスタム・ディメンション・サイズ	通貨に使用
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・ディメンションを作成するには:

1. 「ディメンション」表で、「カスタム・ディメンションの作成」(+)ボタンをクリックするか、「アクション」、「カスタム・ディメンションの作成」の順に選択します。
2. 「ショート名」に、一意のディメンション名を入力します。
3. 「別名」に、一意のディメンションの別名を入力します。
4. 「ディメンション・サイズ」に、次のいずれかのオプションを選択します: 「小」、「中」または「大」。
5. 追加のディメンションを作成するには、これらのステップを繰り返します。
6. **オプション:** アプリケーションを作成するためのステップをすべて完了したら、「アプリケーションの作成」をクリックします。

アプリケーションが正常に作成されると、アプリケーションが作成されたこととメタデータが正常にロードされたことを示す確認メッセージが表示されます。確認メッセージで、「OK」をクリックします

ノート:

アプリケーションの作成後にディメンション数を変更することはできません。追加のカスタム・ディメンションが必要な場合は、追加のカスタム・ディメンションを含む新しいアプリケーションを作成する必要があります。

アプリケーション・プロファイルの保存

アプリケーション作成ウィザードのすべてのステップを完了して「アプリケーションの作成」をクリックすると、ウィザード・ナビゲーションで選択したオプションを使用して、バイナリ形式のアプリケーション・プロファイル(.PER)・ファイルが生成されます。その後で、そのプロファイル・ファイルを使用してアプリケーションが作成されます。選択した設定で、メタデータが更新およびロードされます。

アプリケーションのプロファイル・ファイルが作成されると、そのプロファイル・ファイルを使用して新規アプリケーションを作成できます。[プロファイル・ファイルからのアプリケーションの作成](#)を参照してください。

プロファイル・ファイルからのアプリケーションの作成

新規アプリケーションの作成時には、ウィザード・ナビゲーションで選択したオプションを使用して、バイナリ形式のアプリケーション・プロファイル(.PER)・ファイルが生成されます。その後で、そのプロファイル・ファイルを使用してアプリケーションが作成されます。既存のアプリケーションのプロファイル・ファイルを使用して新規アプリケーションを作成できます。

プロファイル・ファイルを使用する場合、アプリケーション設定にはデフォルト値が使用されます。

プロファイル・ファイルからアプリケーションを作成するには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「アプリケーション」タブで、「アクション」、「新規」の順に選択します。
アプリケーション作成ウィザードが起動されます。
3. アプリケーションのオプションから、「プロファイル・ファイルからアプリケーション」を選択し、「次」をクリックします。
4. 「プロファイル・ファイルの参照」画面で「参照」をクリックし、アプリケーション・プロファイル(.PER)ファイルを選択して「次」をクリックします。
5. 必要に応じてアプリケーション・プロパティを変更してから、「アプリケーションの作成」をクリックします。

アプリケーションからのコピー

アプリケーションを作成する場合、既存のアプリケーションからアプリケーション・プロファイルのコピーできます。新規アプリケーションを作成するには、アプリケーションからアプリケーション・カレンダー、モジュール構成およびカスタム・ディメンションを表示し、アプリケーション作成ウィザードを使用して変更できます。

アプリケーションを作成するには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「アプリケーション」タブで、「アクション」、「新規」の順に選択します。
アプリケーション作成ウィザードが起動されます。

3. **アプリケーションのオプション**から、「**アプリケーションからのコピー**」を選択し、「**次**」をクリックします。
4. 「**アプリケーションの選択**」画面で、コピーするアプリケーションを選択します。
選択したアプリケーションの情報(アプリケーション設定やモジュール構成など)がウィザードに移入されます。現在の値を変更できます。
 - 「**アプリケーション・プロパティ**」画面からアプリケーションを作成するには、「**アプリケーションの作成**」をクリックします。
 - アプリケーションの情報を変更するには、「**次**」をクリックしてウィザードの指示に従います。

アプリケーションの変更

アプリケーションを作成した後で、アプリケーション・プロファイルを変更することは通常できません。ただし、管理者は次の設定を変更できます。

- アプリケーション・モジュールの有効化または無効化。[アプリケーション・モジュールの有効化または無効化](#)を参照してください。
- アプリケーション内で使用する年数の増加。

アプリケーションでサポートされる年数は変更できますが、次の制限があります。

- 新しいプロファイル終了年が現在の終了年よりも後であること。年数は増加のみが可能で、数を減らすことはできません。
- 新しい終了年が 2100 年よりも前であること。
- アプリケーション・プロファイルの開始年は変更できません。

例

Current start year: 2005

Current number of years: 10 (End year 2015)

Request for modification: 15 (End year 2020)

システムでアプリケーションが管理モードに設定されると、アプリケーション・プロファイルの終了年を処理できるようになります。現在のアプリケーション・プロセスが停止し、アプリケーションにログインしているすべてのユーザーが強制的にログアウトされます。

アプリケーションを変更するには:

1. 「**ナビゲート**」、「**管理者**」、「**連結管理**」の順に選択します。
2. 「**アプリケーション**」タブで、変更するアプリケーションを選択します。
3. 「**アプリケーションの変更**」をクリックするか、「**アクション**」、「**アプリケーションの変更**」の順に選択するか、アプリケーションを右クリックして「**アプリケーションの変更**」を選択します。
アプリケーション・プロパティのほとんどはグレー表示されています。
4. 「**年数**」を変更するには、アプリケーションの年数を入力します。

5. 「**アプリケーションの変更**」をクリックします。

変更されたアプリケーションが処理される前に、警告メッセージが表示されます。「はい」をクリックすると、アプリケーションが管理モードに設定され、年数が検証されます。検証エラーがない場合は、年ディメンションの更新、管理モードの無効化の順に実行されます。

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**」、「**管理者**」、「**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。

アプリケーションを開く

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

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

アプリケーションを開くには:

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

 **ノート:**

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。管理モジュールをリロードするこのオプションを使用して、接続したサーバーを変更できます。

サーバーのクラスタを変更するには:

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

アプリケーション・リストの表示

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

アプリケーション・リストを表示するには:

1. 「ナビゲート」、「管理者」、「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

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

アプリケーションを削除するには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「管理」、「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せん。「**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、エラーを修正してリロードできます。

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:

- [アプリケーション・セキュリティのロード](#)
- [メタデータのロード](#)
- [メンバー・リストのロード](#)
- [ルールのロード](#)

アプリケーション要素をロードするには:

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

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

 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。

7. **オプション:** ログ・ファイルをダウンロードするには、「ダウンロード・ログ」をクリックします。「開く」をクリックしてログ・ファイルを表示するか、「保存」をクリックしてファイルをローカルに保存する場所を選択します。

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リロードできます。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:

- [アプリケーション・セキュリティの抽出](#)
- [メタデータの抽出](#)
- [メンバー・リストの抽出](#)
- [ルールの抽出](#)

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

アプリケーション要素を抽出するには:

1. アプリケーションを開きます。
2. 「連結」、「抽出」、「アプリケーション要素」の順に選択します。
3. 抽出プロセスのオプションを指定します。
4. 「抽出」をクリックして個々のファイルを抽出するか、「すべて抽出」をクリックします。
5. 「ダウンロード」をクリックして、抽出したファイルをダウンロードします。
6. **オプション:** ログ・ファイルをダウンロードするには、「ダウンロード・ログ」をクリックします。「開く」をクリックしてログ・ファイルを表示するか、「保存」をクリックしてファイルをローカルに保存する場所を選択します。

Financial Management アプリケーションの複製

アプリケーションのコンテンツを複製して新しいアプリケーションを作成できます。アプリケーションを複製するには、アプリケーション管理者役割と、Financial Management アプリケーション作成者役割の Shared Services でプロビジョニングされていることが必要です。

アプリケーションを同じ環境に複製して名前を変更するか、新しいアプリケーションを別のプロジェクトおよびクラスタに登録できます。同じデータベース内にあるアプリケーションのみを複製できます。

デフォルトでは、プロセスによって、アプリケーション・プロファイル情報、メタデータ、およびセキュリティ情報が複製されます。データ表およびタスク監査とデータ監査の表を、

ソース・アプリケーションから宛先アプリケーションに複製するよう選択することもできます。

アプリケーションを複製するには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「管理」、「アプリケーション」の順に選択します。
3. 複製するアプリケーションを選択します。
4. 「アプリケーションの複製」をクリックするか、「アクション」、「アプリケーションの複製」の順に選択するか、アプリケーション名を右クリックして「アプリケーションの複製」を選択します。
5. 「名前」に、作成する新規アプリケーションのアプリケーション名を入力します。
6. 「説明」に、アプリケーションの説明を入力します。
7. 「クラスタ」で、アプリケーションを登録するクラスタを選択します。
8. 「ユーザー管理プロジェクト」で、アプリケーションを登録するプロジェクトを選択します。
9. **オプション: 監査表の複製** オプションを選択して、データ監査表とタスク監査表をソースから宛先アプリケーションに複製します。
10. **オプション: データ表の複製** オプションを選択して、データ表をソースから宛先アプリケーションに複製します。
11. 「複製」をクリックします。

複製プロセスを開始したときに、管理モードが有効化されます。すべてのユーザーはアプリケーションからログアウトされ、プロセスが完了するまで再度ログインできなくなります。

「管理タスク」ページが表示され、「アプリケーションの複製」タスクの進行状況をモニターできます。

ソース・アプリケーションが正常に複製されると、新しく作成されたアプリケーションは指定したクラスタおよびプロジェクトに登録され、新規アプリケーションを開けるようになります。

アプリケーションのインポート

同種データベースを使用して、データベース間または EPM システム間でアプリケーションをインポートできますが、EPM システムは両方とも Oracle Database または SQL Server のいずれかを使用する必要があります。ソース・アプリケーションは前のリリースから 11.1.2.4 までさかのぼることができますが、古いアプリケーションをアップグレードするには、1つの Financial Management サーバーの **EPM System コンフィギュレータ** で Financial Management アップグレード・アプリケーション・タスクを実行する必要があります。アプリケーションをインポートするには、アプリケーション管理者役割と、Financial Management アプリケーション作成者役割の Shared Services でプロビジョニングされていることが必要です。

アプリケーションを同じ環境にインポートして名前を変更するか、新しいアプリケーションを別のプロジェクトおよびクラスタに登録できます。

すべての年およびシナリオのデータをインポートするか、特定の年およびシナリオを選択できます。

デフォルトでは、プロセスによって、アプリケーション・プロファイル情報、メタデータ、およびセキュリティ情報が複製されます。タスクとデータの監査情報を、ソース・アプリケーションから宛先アプリケーションにインポートするよう選択することもできます。

アプリケーションのインポートの前提条件

アプリケーションをインポートする前に、次のステップを実行する必要があります。

- ソース・データベースと宛先データベースの間にデータ通信リンクを作成する必要があります。
- Oracle データベースを使用している場合は、データベース・リンクを作成する必要があります。
- SQL Server データベースを使用している場合は、リンク・サーバーを作成する必要があります。
- HFM データベースに、データベース・リンクまたはリンク・サーバーを作成する権限があることを確認してください。
- 役割を介したユーザーのプロビジョニングではなく、データベース・リンクまたはリンク・サーバーで構成されたデータベース・ユーザーに対して明示的なワнтаイム DB 権限を付与する必要があります。

アプリケーションをインポートするには:

1. 宛先 EPM システムにログオンします。
2. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
3. 「管理」、アプリケーションのインポートの順に選択します。
4. Oracle データベース間でインポートする場合は**データベース・リンク**を選択し、SQL Server データベース間でインポートする場合は**リンク・サーバー**を選択します。
データベース・リンクまたはリンク・サーバーを選択すると、ソース・アプリケーションがリストされます。
5. アプリケーション・リストから、ソース・アプリケーションを選択します。
6. **宛先アプリケーション名**および**宛先アプリケーションの説明**を指定します。
ソース・アプリケーションはこの名前と説明でインポートされます。
7. 「クラスタ」で、アプリケーションを登録するクラスタを選択します。
8. 「ユーザー管理プロジェクト」で、アプリケーションを登録するプロジェクトを選択します。
9. **オプション: 監査表のインポート**・オプションを選択して、データ監査表とタスク監査表をソースから宛先アプリケーションにインポートします。
10. **オプション: データ表のインポート**・オプションを選択して、データ表をソースから宛先アプリケーションにインポートします。
「**データ表のインポート**」を選択した場合は、シナリオ・フィルタと年フィルタも選択する必要があります。
11. **Optional:** ソース・データ表をターゲットのアプリケーションにコピーした後消去するには、「**ソース・データの消去**」オプションを選択します。
12. 「**インポート**」をクリックします。

インポート・プロセスを開始すると、この操作はデータ量が多い操作であるという警告が表示され、続行するかどうかを尋ねられます。

「管理タスク」ページが表示され、アプリケーションのインポート・タスクの進行状況をモニターできます。

ソース・アプリケーションが正常にインポートされると、新しく作成されたアプリケーションは指定したクラスタおよびプロジェクトに登録され、新規アプリケーションを開けるようになります。

Oracle Database のデータベース・リンクの作成

アプリケーションをインポートする前に、ソース・データベースと宛先データベースの間にデータベース・リンクを作成する必要があります。

宛先データベースから、次のステップを実行します。

ソース Oracle DB、HFM スキーマへのデータベース・リンクを作成するには:

1. **sysdba** ユーザーとしてログインし、HFM スキーマへのデータベース・リンクを作成するための権限を付与します。

```
GRANT CREATE DATABASE LINK TO hfm;
```

2. 宛先システムの HFM スキーマにログインし、次のコマンドを実行します。

```
CREATE DATABASE LINK <link name> CONNECT TO<hfm schema name>IDENTIFIED BY HFM1 USING '///host name:<port>/<service name>';
```

たとえば、ホスト **SLCK58001** (Oracle がポート **1521** で実行されている)にある HFM スキーマに接続するには、次のようになります:

```
CREATE DATABASE LINK ToTestSystem CONNECT TO HFM IDENTIFIED BY HFM1 USING '///slck58001.xxxx:1521/service name';
```

3. 検証ステップ: 次のコマンドを実行すると、ソース・システムにあるアプリケーションがリストされます。

```
Select * from HSX_DATASOURCES@ToTestSystem
```

4. データベース・リンクを削除するコマンド:

```
drop database link ToTestSystem;
```

5. すべてのデータベース・リンクをリストするコマンド:

```
select * from all_db_links
```

SQL Server データベース用のリンク・サーバーの作成

アプリケーションをインポートする前に、SQL Server データベースを使用している場合は、宛先データベースとソース・データベースの間にリンク・サーバーを作成する必要があります。

宛先データベースから、次のステップを実行します。

ソース **SQLServer** インスタンスへのリンク・サーバーを作成するには:

1. 宛先 **HFM** データベースで、4つのコマンドを順序どおりに実行(それぞれ **exec** で開始)します。

```
exec sp_addlinkedserver N'<linked server name>',@srvproduct=N'',
@provider=N'SQLNCLI', @datasrc=N'<host name>'
```

```
exec sp_addlinkedsrvlogin N'<linked server
name>',@useself=N'FALSE',@rmtuser=N'<hfm db>',@rmtpassword=N'<hfm db
password>'
```

```
exec sp_serveroption @server=N'<linked server
name>',@optname=N'rpc',@optvalue=N'true'
```

```
exec sp_serveroption @server=N'<linked server name>',@optname=N'rpc
out',@optvalue=N'true'
```

たとえば、**ToTestSystem** という名前のリンク・サーバーを **stm10082 上の sqlserver** に作成し、ソースの **sqlserver HFM** データベース資格証明が **hfm/hyperion** である場合は、次のようにします。

```
exec sp_addlinkedserver
N'ToTestSystem',@srvproduct=N'',@provider=N'SQLNCLI',@datasrc=N'STM10082.x
xx.COM'
```

```
exec sp_addlinkedsrvlogin
N'ToTestSystem',@useself=N'FALSE',@rmtuser=N'hfm', @rmtpassword=N'password'
```

```
exec
sp_serveroption@server=N'ToTestSystem',@optname=N'rpc',@optvalue=N'true'
```

```
exec sp_serveroption @server=N'ToTestSystem',@optname=N'rpc
out',@optvalue=N'true'
```

2. 検証ステップ: 次のコマンドを実行すると、ソース **SQL Server** にあるアプリケーションがリストされます。

```
select * from [ToTestsystem].[hfm].[dbo].HSX_DATASOURCES;
```

3. リンク・サーバーを削除するコマンド:

```
sp_dropserver'<linked server name>', 'droplogins'
```

Example: `sp_dropserver 'ToTestSystem', 'droplogins'`

- すべてのリンク・サーバーをリストするコマンド:

```
sp_linkedservers
```

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、Oracle Hyperion 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。管理者グループへのユーザーの割当てについては、*Oracle Enterprise Performance Management System ユーザー・セキュリティ管理ガイド*を参照してください。

次の手順を参照してください:

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

システム・メッセージの表示

システム・メッセージを表示するには:

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

システム・メッセージの削除

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

システム・メッセージを削除するには:

1. 「ナビゲート」、「管理者」、「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 2-1 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

アプリケーション・アクセス権の管理

管理機能を使用して、アプリケーションに関する次のアクセス関連の設定を制御できます。

- 指定されたアプリケーションのすべてのユーザーをログアウトします。[ユーザーのログアウト](#)を参照してください。
- すべてのユーザーのアプリケーション・アクセスを有効および無効にします。[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、アクセス関連の設定を管理できます。管理者グループへのユーザーの割当てについては、*Oracle Enterprise Performance Management System ユーザー・セキュリティ管理ガイド*を参照してください。

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・サーバーからすべてのユーザーをログオフできます。

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。

ログアウト・アクションでは、選択したアプリケーションのすべてのユーザーが強制的にログアウトされます。

ノート:

選択したユーザーをログアウトさせるには、[システム・ユーザーの管理](#)を参照してください。

アプリケーションまたはサーバーからすべてのユーザーをログオフするには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「管理」、「システム」、「管理」の順に選択します。
3. 「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、確認プロンプトで「はい」をクリックします。
4. 「ログアウト結果」ダイアログ・ボックスで、「OK」をクリックします。

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。

ユーザー・アクセスには次のオプションを使用できます。

- **接続を無効にする** - このオプションでは、管理者を含むすべてのユーザーの接続を無効化します。どのユーザーも、選択したアプリケーションにアクセスできません。
- **管理者に対してのみ接続を有効にする** - 管理者にのみアプリケーション・アクセスが許可されます。
- **すべてのユーザーに対して接続を有効にする** - すべてのユーザーにアプリケーション・アクセスが許可されます。

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」、「管理者」、「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. 次のいずれかのアクションを行います:
 - 「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。
 - 「管理者に対してのみ接続を有効にする」をクリックするか、「アクション」、「管理者に対してのみ接続を有効にする」の順に選択します。
 - 「すべてのユーザーに対して接続を有効にする」をクリックするか、「アクション」、「すべてのユーザーに対して接続を有効にする」の順に選択します。
4. 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。

アプリケーションの接続ステータスの表示

接続ステータスは、サーバー・リストまたはアプリケーション・リストで表示できます。

サーバー別に表示する場合は、サーバー名およびそのサーバーで適用可能なアプリケーションとそのステータス(「使用可能」または「使用不可」)のリストが表示されます。

アプリケーション別に表示する場合は、アプリケーション名、格納場所のサーバー、およびそれぞれのステータス(「使用可能」または「使用不可」)が表示されます。

アプリケーションの接続ステータスを表示するには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「管理」、「システム」、「管理」の順に選択します。
3. メニュー・アイコンを使用して表示タイプを切り替えます。
 - リストがサーバー別に表示されている場合は、「クリックしてアプリケーションごとに表示」を選択します。
 - リストがアプリケーション別に表示されている場合は、「クリックしてサーバーごとに表示」を選択します。

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。[ユーザーの表示](#)および[ユーザーのログアウト](#)を参照してください。

 **ノート:**

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。管理者グループへのユーザーの割当てについては、*Oracle Enterprise Performance Management System ユーザー・セキュリティ管理ガイド*を参照してください。

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。

 **ノート:**

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
- サーバー名
- アプリケーション名
- ステータス

ログオンしたユーザーを表示するには:

1. 「ナビゲート」、「管理者」、「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。

選択したユーザーまたはすべての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。[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」、「管理者」、「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

サーバーとアプリケーションを管理するには、管理者であることが必要です。

関連トピック:

- [管理モードの有効化および無効化](#)
- [サーバーの同期化](#)

管理モードの有効化および無効化

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

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

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

管理モードを有効化および無効化するには:

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

サーバーの同期化

Oracle Hyperion Financial Management アプリケーション・サーバー間の同期は、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。

- 配賦
- 勘定科目による自動照合
- ID による自動照合
- 記帳ロジック(計算)
- 連結
- 取引の作成
- データ監査の除去
- データのクリア
- データのコピー
- データの入力
- データの抽出
- データのロード
- データのロック

- データのロック解除
- すべての取引の削除
- 無効なレコードの削除
- 取引の削除
- ドキュメントの添付
- ドキュメントの添付を解除する
- 取引の編集
- 内部取引期間の管理
- 内部取引理由コードの管理
- 内部取引手動照合
- 勘定科目による内部取引照合レポート
- IDによる内部取引照合レポート
- 内部取引レポート
- アイドル
- 仕訳の入力
- 仕訳期間のクローズ
- 仕訳期間のオープン
- 仕訳の転記
- 仕訳テンプレートの入力
- 仕訳の転記の戻し
- エンティティのロックとロック解除
- ログオフ
- ログオン
- メンバー・リストの抽出
- メンバー・リストのロード
- メンバー・リストのスキャン
- メタデータの抽出
- メタデータのロード
- メタデータのロード差異
- メタデータのスキャン
- アプリケーションの変更
- すべての取引の転記
- 取引の転記
- ルールの抽出
- ルールのロード

- ルールのスキャン
- セキュリティの抽出
- セキュリティのロード
- タスク監査の除去
- 取引の抽出
- 取引のロード
- 取引のスキャン
- 換算
- すべての取引の照合の解除
- 内部取引の照合の解除
- すべての取引の転記の戻し
- 取引の転記の戻し

タスク監査ログには次の情報が含まれます。

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

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

タスク監査を実行するには:

1. 「ナビゲート」、「管理者」、「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、「**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。

 **ノート:**

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

データの監査

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

データ変更を監査する勘定科目およびシナリオの `EnableDataAudit` メタデータ属性を使用できます。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
- データのクリア
- データのコピー
- データのロード
- 仕訳の入力

データ監査ログには次の情報が含まれます。

- ユーザー名
- 実行した操作
- 変更時間
- サーバー名

- 視点
- 視点に入力された値

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

データ監査を実行するには:

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

 ノート:

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

実行中のタスクのモニタリング

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

デフォルトで、実行中のタスクはデータベースに 900 秒(15 分)間残ります。

AutoClearDeadTasks システム設定を変更することで、実行中のタスクを保持するかどうか、またはデフォルトの 15 分後にパージするかどうかを変更できます。[使用可能な構成設定](#)を参照してください。

- [実行中のタスクの表示](#)
- [実行中のタスクの停止](#)
- [実行中のタスクのリフレッシュ](#)

実行中のタスクの表示

実行中のタスクを表示するには:

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

次のタスクを表示および終了できます。

- 連結
- データのロード
- データの抽出
- データベースへのデータの抽出のエクスポート
- 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ステータス・タイプ別にフィルタ処理できます:

- 初期化中
- 実行中
- 一時停止
- 初期化解除中
- 停止
- 中止
- 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

実行中のタスクの停止

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

タスクを停止するには:

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

実行中のタスクのリフレッシュ

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

ノート:

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

「タスクの実行」リストをリフレッシュするには:

1. タスク・リストから、「**リフレッシュ**」をクリックするか、「**アクション**」、「**リフレッシュ**」の順に選択します。

2. ドロップダウン・リストから、リフレッシュ・モードを選択します:
 - デフォルト:
 - 手動
 - 5 秒
 - 10 秒
 - 15 秒
 - 30 秒
 - 60 秒

 ノート:

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

無効なレコードのスキャンとクリア

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

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

無効なレコードをスキャンして削除するには:

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

 ノート:

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

4. 次のいずれかのアクションを選択します:
 - 「スキャン」をクリックし、無効なレコードをスキャンします。
 - 「無効なレコードのクリア」をクリックして、レコードをクリアします。

HFM インサイトを使用したシステム・ステータスのモニタリング

HFM インサイトは、システム管理者が Oracle Hyperion Financial Management システムのステータスをモニタリングするためのダッシュボードを提供します。一定期間の間にシステム・コンポーネントから収集されたデータが表示され、管理者は必要に応じて修正アクションを取ることができます。

HFM インサイトへのアクセス

HFM インサイトにアクセスするには、Oracle Hyperion Financial Management アプリケーション管理者の役割が必要です。

ノート:

モニタリング用にアプリケーション・プロセスを実行しておく必要はありません。

HFM インサイトにアクセスするには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「システム」、「インサイト」の順に選択します。

HFM インサイトのメイン・ダッシュボード

HFM インサイトを開くと、メイン・ページには、各サーバー上で実行されているアプリケーションとそれぞれのステータスを行ごとに示した表が表示されます。複数サーバー環境では、アプリケーションをサーバー別にフィルタ処理できます。

表には、フィルタで選別されたアプリケーションとサーバーの組合せごとに行が表示されます。たとえば、3つのアプリケーションと2つのサーバーを選択した場合、表には6行(サーバー1のアプリケーション1、サーバー2のアプリケーション2というように)が表示されます。各行に次の情報が表示されます。

- アプリケーション・アイコン、アプリケーション名およびサーバー名
- 次のキー・パフォーマンス・インディケータ(KPI)に基づいたチャート。データは過去2週間に限定されます。
 - ユーザー - 過去2週間のユーザー数
 - エラー - 過去2週間のエラー数
 - タスク - 過去2週間のタスク数
 - メモリー - 物理メモリー(MB)
 - CPU - CPU 使用状況
- ステータスの概要:

- 次のいずれかのステータスが表示された「**ステータス**」アイコン:
 - * 管理モード
 - * クラッシュ
 - * 停止中(プロセスがロードされていません)
 - * 応答していません
 - * 稼働中(プロセスが稼働中です)
- **開始時間** - XFMDatasource プロセスが最後に開始された時間
- **アップタイム** - アップタイム(開始時間と最後の ping 時間との差)の計算
- **最終ヘルス・チェック時刻** - XMFDataSource プロセスへの最後に成功した ping のタイムスタンプ(HsxServer ログから)

デフォルトでは、すべてのメトリックが表示されます。個々のグラフから任意のメトリックを選択解除できます。たとえば、CPUのみを選択して他のメトリックを選択解除することができます。グラフはユーザーの選択内容に応じて調整されます。

「リフレッシュ」オプションを使用すると、いつでも画面をリフレッシュできます。

HFM インサイトでのアプリケーションのフィルタ

アプリケーション・フィルタ - 自分がアプリケーション管理者であるアプリケーションがすべて表示されます。1つ以上のアプリケーションを選択できます。デフォルトでは、リストにある最初の5つのアプリケーションが選択され、そのアプリケーションのステータスが表に表示されます。

サーバー・フィルタ - 使用可能なすべてのサーバーが表示されます。1つ以上のサーバーを選択できます。システムにサーバーが1つしかない場合、サーバー・フィルタは表示されません。サーバー・フィルタが表示される場合、デフォルトでは最初のサーバーが選択され、そのサーバーのアプリケーション・ステータスが表に表示されます。

表示フィルタ - グラフに表示されるメトリックを選択できます。これは複数選択リストです。デフォルトでは、すべてのメトリックが選択され、すべてのメトリック(ユーザー、エラー、タスク、メモリーおよびCPU)がグラフに表示されます。

HFM インサイトのアプリケーションの詳細

アプリケーション・アイコンをクリックすると新しいタブが開き、そのサーバー上で実行されているアプリケーションに関するその他の詳細なメトリックが表示されます。

タブのタイトルは<アプリケーション名>@<サーバー名>です。

- **表示** - 表示する期間のフィルタ。デフォルトでは、過去14日間のデータが表示されます。
- **リフレッシュ** - 「手動」または「自動」リフレッシュを選択します。デフォルト値は「手動」です。

「アプリケーション」ページには次のセクションが表示されます。

アプリケーション要約

- アプリケーション・アイコン、アプリケーション名およびサーバー名
- ステータスの概要(メイン・ページの表示と同様)

- **アプリケーション・ディメンション** - アプリケーション・ディメンションと各ディメンションの合計メンバー
- **再起動** - XFMDatasource プロセスを再起動します

統計

「統計」セクションには、選択したシナリオおよび年ディメンションに基づく詳細データ統計が表示されます。統計を手動で更新する際に選択できます。

統計を更新するには:

1. 「統計」セクションから、「データ」タブをクリックします。
2. 「メンバー・セレクタ」 をクリックし、シナリオおよび年を選択し、「OK」をクリックします。
3. 「実行」ボタン  をクリックしプロセスを実行し、選択したシナリオおよび年の組合せの統計を戻します。

ノート:

「実行」ボタンをクリックし統計を更新すると、プロセスの実行中の一定の時間、ボタンは無効になります。タスクはバックグラウンドで実行され、実行中、アプリケーションの他の部分にナビゲートできます。完了すると、ボタンは再度有効になり、更新された統計で表がリフレッシュされます。「リフレッシュ」をクリックして、更新された値を表示することもできます。

特定の表にレコードがない場合、その表の統計は表示されません。

「データ」タブ

- データ(シナリオ/年の組合せ当たりの DCE および DCN 合計レコード数)
- 無効なレコード(無効なメンバー ID)、DCE および DCN のみ、合計数のみ(エンティティ当たりではない)
- すべての期間のゼロに近い値のレコード
- すべての期間のゼロに近い値の計算済レコード
- PFlow 履歴(シナリオ/年当たり)
- 入力データ(年内の期間が入力またはライン・アイテム詳細の場合)

表カウント・タブ

表カウント・タブには、Oracle Hyperion Financial Management システムの次の表のレコード数および最後に実行された時刻が表示されます。

- タスク監査
- データ監査
- システム・メッセージ(エラー・ログ表)
- ルール・プロファイル・データ

パフォーマンスの概要

「パフォーマンス」セクションには、メイン画面と同じ KPI に基づいたチャートが表示されません。

ユーザー

フリークエント・ユーザー - タスクの監査データに基づいて、ある期間の上位 10 ユーザーとその最大セッション数が表示された棒グラフ。デフォルトでは過去 14 日間のデータが表示されますが、「表示」フィルタを使用して期間を変更することができます。

セッション詳細 - ユーザー・セッションの詳細(ユーザー名、開始時間、終了時間、所要時間など)。任意の列をソートできます。

ユーザー別にフィルタできます。ユーザー検索ボックスに、ユーザー名のすべてまたは一部を入力します。

所要時間の検索ボックスには、特定の期間ログインしていたユーザーのみを表示する場合に、時間値を入力します。たとえば、1 時間ログインしていたユーザーを表示する場合は、「1h」と入力します。ワイルドカードとしてパーセント記号(%)を使用できます。たとえば、所要時間が 1 時間未満のユーザーを表示するには、「検索」ボックスに「%h」と入力します。

メモリー

「メモリー」セクションには、システム・メッセージのデータに基づいて、システム・メモリー・パラメータのグラフが表示されます。次のものが含まれます:

- 合計メモリー
- 使用済メモリー
- NumCubesInRAM
- NumDataRecordsInRAM
- NumRecordsInLargestCube
- MinDataCacheSizeInMB
- MaxDataCacheSizeInMB
- MaxNumCubesInRAM

メモリー・パラメータの詳細は、[使用可能な構成設定](#)を参照してください。

タスク

このセクションには、タスク情報を表示する 4 つのタブが含まれます。

タスク・ビュー

「タスク」ビューには、タスクの監査データに基づいて、選択された期間スパンにおいて頻繁に実行された上位 10 タスクが表示されます。たとえば、過去 2 週間に発生したログオンまたは連結の回数を簡単に表示できます。

- **メトリック** - 期間単位の時間メトリックのフィルタ:
 - 期間単位 - 選択した時間スパン内に発生したタスク数
 - 合計時間 - 選択した時間スパン内に発生したタスクの総所要時間

- 平均時間 - 選択した時間スパン内に発生したタスクの平均所要時間
- **ユーザー** - タスクを実行したユーザー別のフィルタ。「ユーザー」の「検索」ボックスでは、ユーザー名の全部または一部を入力するか、ワイルドカードとしてパーセント記号(%)を使用します。

ユーザー・ビュー

「ユーザー・ビュー」タブには、特定のタスクを実行した上位 10 ユーザーを示す棒グラフが表示されます。

- **メトリック** - タスク時間のフィルタ:
 - 期間単位 - 選択した時間スパン内にユーザーが実行したタスク数
 - 合計時間 - 選択した時間スパン内に実行されたタスクの総所要時間
 - 平均時間 - 選択した時間スパン内に実行されたタスクの平均所要時間
- **タスク** - タスク別のフィルタ。デフォルトで、「連結」が選択されます。

月ビュー

「月ビュー」には、特定のタスクを月ベースで示した棒グラフが表示されます。タスク・アクティビティを月別に比較することはできません。

- **メトリック** - タスク時間のフィルタ:
 - 期間単位 - 選択した月間に実行されたタスク数
 - 合計時間 - 選択した月間のタスクの総所要時間
 - 平均時間 - 選択した月間のタスクの平均所要時間
- **タスク** - タスク別のフィルタ。デフォルトで、「連結」が選択されます。

タスクの詳細

このタブには、タスクの監査データに基づいて、各タスクの詳細情報が表示されます。これにはユーザー名、アクティビティ名、開始時間、終了時間、所要時間、説明が含まれます。任意の列をソートできます。

ユーザー、アクティビティ、所要時間および説明でソートできます。

エラー

「エラー」セクションには、システム・メッセージのデータに基づいて、上位 10 件の頻繁なエラーが表示されます。

- **エラー頻度** - エラー・コードが表示されます。エラー・コードにマウス・カーソルを合わせると、その説明が表示されます。
- **エラーの詳細** - 選択したサーバー上の選択したアプリケーションについて、特定の時間スパンにおけるエラーのエラー・コード、システム・メッセージの要約、日時が表示されます。エラー・コードと日時でソートできます。エラー・コードとシステム・メッセージの要約でフィルタできます。

3

アプリケーション・セキュリティの管理

セキュリティおよびアクセス権限を使用すると、Oracle Hyperion 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。[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。[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。役割の割当ての詳細は、[Oracle Enterprise Performance Management System ユーザー・セキュリティ管理ガイド](#)を参照してください。

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

アプリケーション・セキュリティに関する考慮事項

Oracle Hyperion 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保護において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。[アプリケーション・セキュリティのロード](#)および[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

Oracle Hyperion Financial Management 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。
2. ユーザーをプロビジョニングするには、ユーザーおよびグループをアプリケーションに割り当て、ユーザーに役割を割り当てます。[Oracle Enterprise Performance Management System ユーザー・セキュリティ管理ガイド](#)を参照してください。

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

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

Shared Services Console を起動するには:

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

セキュリティ・クラスを割り当てるユーザーおよびグループの選択

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

ノート:

Hyperion Financial Management では、「ユーザー」と「グループ」は同じ名前にできません。

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

アプリケーションのユーザーとグループを選択するには:

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

セキュリティ・クラスの設定

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

次の手順を実行できます:

- [セキュリティ・クラスの作成](#)
- [セキュリティ・クラスの削除](#)
- [セキュリティ・クラスの選択](#)

ノート:

「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. 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・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(-)
- シャープ記号(#)
- 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・ディメンションから削除できます。セキュリティ・クラスをアプリケーションから削除する前に、そのセキュリティ・クラスとアプリケーション要素の関連付けを解除する必要があります。

エンティティ、勘定科目またはシナリオの関連付けをセキュリティ・クラスから解除するには、メタデータ・ファイルのセキュリティ・クラスを変更します。仕訳の関連

付けをセキュリティ・クラスから解除するには、仕訳ファイルを変更するか、仕訳の処理モジュールで仕訳のセキュリティ・クラスを更新します。

セキュリティ・クラスを削除するには:

1. 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・クラスのリストから、表のビューに基づくセキュリティ・クラスの行/列ヘッダーを選択します。

ヒント:

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

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

セキュリティ・クラスの選択

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

アプリケーションのセキュリティ・クラスを選択するには:

1. 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・クラス」を選択するか、または「アクション」、「セキュリティ・クラス」の順に選択します。
3. 「使用可能なセキュリティ・クラス」で、アプリケーションに割り当てるセキュリティ・クラスを選択し、矢印キーを使用して、それらを「選択されたセキュリティ・クラス」列に移動します。
4. 「OK」をクリックします。

セキュリティ・クラスへのユーザー・アクセス権の割当て

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

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

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

ザーにセキュリティ・クラスに対するすべてのアクセス権またはレベルを上げるアクセス権が必要です。

 **ノート:**

「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 3-1 ユーザー・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. 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. 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。

 **ノート:**

表 1 を参照してください。

- なし

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

 **注意:**

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

 **ノート:**

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。

 **ノート:**

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

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

セキュリティ・レポートの実行

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

セキュリティ・レポートを作成するには:

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

 **ノート:**

HFM フォーマットは、セキュリティ・レポートではサポートされていません。

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 Oracle Hyperion Financial Management アプリケーションに対してのみです。

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

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

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

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

 **ノート:**

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。

 **ノート:**

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

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

 **注意:**

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

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:

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

 **ノート:**

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。
8. **オプション:** ログ・ファイルをダウンロードするには、「ダウンロード・ログ」をクリックします。「開く」をクリックしてログ・ファイルを表示するか、「保存」をクリックしてファイルをローカルに保存する場所を選択します。

セキュリティ情報のクリアとロード

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

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

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

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

 **ノート:**

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

セキュリティ情報をクリアする前に行っておくべきタスク

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

メタデータ

セキュリティ情報をクリアしてロードする前にメタデータを更新するには:

1. アプリケーションのすべてのメタデータ要素を抽出します。
2. 必要に応じて、メタデータ要素のセキュリティ・クラス情報を変更します。

仕訳

セキュリティ情報をクリアしてロードする前に仕訳を更新するには:

1. アプリケーション内の転記済仕訳の転記をすべて戻します。
2. 承認済の仕訳をすべて却下し、仕訳ステータスを「作業中」に戻します。
3. すべての仕訳を抽出します。
4. 必要に応じて、仕訳のセキュリティ・クラス情報を変更します。

グリッド

セキュリティ情報をクリアしてロードする前にグリッドを更新するには:

1. セキュリティ・クラスが割り当てられているすべてのグリッドを抽出します。
2. 必要に応じて、グリッドのセキュリティ・クラス情報を変更します。

データ・フォーム

セキュリティ情報をクリアしてロードする前にデータ・フォームを更新するには:

1. セキュリティ・クラスが割り当てられているデータ・フォームをすべて抽出します。
2. 必要に応じて、データ・フォームのセキュリティ・クラス情報を変更します。

セキュリティ情報をクリアした後で行うタスク

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

メタデータ

メタデータを更新するには:

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

仕訳

セキュリティ情報をクリアしてロードした後に仕訳を更新するには:

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

グリッド

セキュリティ情報をクリアしてロードした後にグリッドを更新するには:

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

データ・フォーム

セキュリティ情報をクリアしてロードした後にデータ・フォームを更新するには:

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

フォルダ

セキュリティ情報をクリアしてロードした後にフォルダを更新するには:

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

レポート

セキュリティ情報をクリアしてロードした後にレポートを更新するには:

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

タスク・リスト

セキュリティ情報をクリアしてロードした後にタスク・リストを更新するには:

1. セキュリティ・クラスが割り当てられているすべてのタスク・リストをリロードします。
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

- 役割のアクセス権
- セキュリティ・クラスのアクセス権

 **ノート:**

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の詳細は、*Oracle Enterprise Performance Management System* インストールおよび構成ガイドを参照してください。

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:

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

 **ノート:**

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って、抽出したファイルをダウンロードします。
使用している Web ブラウザによって、手順が異なります。ファイルは、設定した Web ディレクトリに保存してください。
7. **オプション:** ログ・ファイルをダウンロードするには、「**ダウンロード・ログ**」をクリックします。「**開く**」をクリックしてログ・ファイルを表示するか、「**保存**」をクリックしてファイルをローカルに保存する場所を選択します。

4

メタデータの管理

メタデータは、ディメンション名、メンバー名、プロパティ、為替レート、セキュリティなど、データを説明して格納するアプリケーションの構造要素として定義されます。

メタデータを定義するには、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。

Oracle Hyperion Financial Management では、メタデータに ASCII 文字のみが許可されます。

ノート:

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

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

勘定科目の定義

勘定科目ディメンションは、アプリケーションの勘定科目のチャートを定義します。勘定科目メンバーの属性表に記載されている属性を使用して、勘定科目を定義します。

表 4-1 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>ノート: リリース 4.1 以前の Oracle Hyperion Financial Management では、この勘定科目タイプは 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>勘定科目タイプ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・ディメンションの階層のどの TopMember を、この勘定科目で有効にするかを指定します</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>属性の数は、アプリケーションに定義されたカスタム・ディメンションの数に基づいています。カスタム・ディメンション別名を反映するため、属性名が変更されます。たとえば、属性名のカスタムがカスタム・ディメンション別名で置換されます。</p>
DefaultParent	<p>勘定科目のデフォルトの親です。</p>
説明	<p>勘定科目の説明です</p> <p>説明には、空白を含み、80 文字まで入力できますが、アンパサンド(&)または円記号(バックスラッシュ)()は使用できません。</p>

表 4-1 (続き) 勘定科目メンバーの属性

属性	説明
EnableCustomAggr	<p>カスタム・ディメンション・データが現在の勘定科目に集約されるかどうかを指定します</p> <p>この属性は、連結ではなく特別合計に使用されます。勘定科目がカスタム・ディメンションで集約できるようにする場合は Y を指定し、集約できるようにしない場合は N を指定します。</p> <p>属性の数は、アプリケーションに定義されたカスタム・ディメンションの数に基づいています。カスタム・ディメンション別名を反映するため、属性名が変更されます。たとえば、属性名のカスタムがカスタム・ディメンション別名で置換されます。</p>
EnableDataAudit	<p>勘定科目を監査できるかどうかを指定します。勘定科目の監査を使用可能にするには Y を指定し、使用不可にするには N を指定します。デフォルトは N です。この属性が勘定科目またはシナリオに割り当てられている場合は、この属性によって、何を監査できるかが決まります。</p>
ICPTopMember	<p>勘定科目の内部取引 PartnerTopMember 指定したメンバーおよびその子孫のすべてが勘定科目で有効です。エンティティ・ディメンションのその他のメンバーすべてが勘定科目で無効です。</p>
IsCalculated	<p>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。</p>
IsConsolidated	<p>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。</p>
IsICP	<p>勘定科目が内部取引勘定科目かどうかを指定します。</p> <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

表 4-1 (続き) 勘定科目メンバーの属性

属性	説明
Member	<p>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。</p> <p>勘定科目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"" • 大なり記号(>) • 小なり記号(<) • マイナス記号(-) • シャープ記号(#) • ピリオド(.) • プラス記号(+) • セミコロン(;) • スラッシュ(/)
NumDecimalPlaces	<p>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。</p>
PlugAcct	<p>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。</p>
SecurityClass	<p>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。</p>
Submission Group	<p>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。</p> <p>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。</p>
UserDefined1、UserDefined2、UserDefined3	<p>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。</p>

表 4-1 (続き) 勘定科目メンバーの属性

属性	説明
UsesLineItems	<p>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。</p> <p>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:</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せん。
XBRL タグ	<p>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。</p>

勘定科目タイプの動作

各勘定科目には勘定科目タイプがあります。勘定科目タイプによって、子勘定科目の親勘定科目への集約方法と、時間経過に従って勘定科目残高を累計する方法が決まります。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。

勘定科目タイプは、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。

ノート:

データのロード時に、Oracle Hyperion Financial Management ではデータのナチュラル記号をプラスと見なします。

たとえば、ASSET という勘定科目タイプでは、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。この勘定科目タイプ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 という勘定科目タイプは、カスタム・ディメンション、期間および期間累計のビューの親の値を正しく計算するために必要です。合計に集約する必要のない関連勘定科目をグループ化するには、GROUPLABEL という勘定科目タイプを使用できます。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タイプには、データが格納されます。

 ノート:

位取りは、FLOW、BALANCE または DYNAMIC の勘定科目タイプに適用されません。

表 4-2 勘定科目タイプの動作

タイプ	YTD 合計	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

次の表は、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タイプ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タイプを表します。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。

 ノート:

略語は、勘定科目タイプの最初の 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、勘定科目タイプが親勘定科目に集約されないことを示します。

表 4-3 親勘定科目への集約時の勘定科目タイプの動作

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

この例では、勘定科目タイプ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. 動的勘定科目タイプ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。

 **ノート:**

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

2. ルール・ファイルに、`Sub Dynamic ()`セクションを作成します。
3. ルール・ファイルで計算を定義します。
計算の作成の詳細は、ルールの作成のガイドラインを参照してください。

カスタム・メンバーの定義

カスタム・ディメンションは勘定科目ディメンションと関連づけられており、勘定科目に関する追加の詳細を提供します。カスタム・メンバーの属性表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 4-4 カスタム・メンバーの属性

属性	説明
DefaultParent	カスタム・ディメンションのメンバーのデフォルトの親を指定します。
説明	カスタム・メンバーの説明を指定します。説明には、空白を含み、80文字まで入力できます。
IsCalculated	基本レベルのカスタム勘定科目を計算するかどうかを指定します。基本レベルのカスタム勘定科目を計算する場合は、値を手動入力できません。カスタム勘定科目を計算する場合は Y を指定し、計算しない場合は N を指定します。

表 4-4 (続き) カスタム・メンバーの属性

属性	説明
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>カスタム・ディメンション・メンバーの名前を連結メソッドの名前と同じにすることはできません。</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(-) • シャープ記号(#) • ピリオド(.) • プラス記号(+) • セミコロン(;) • スラッシュ(/)
SecurityClass	<p>カスタム・ディメンション・データにアクセスできるユーザーを定義するセキュリティ・クラス名を指定します。セキュリティ・クラス名には、80文字まで入力できます。セキュリティ・アクセス権は、データにのみ適用されます。</p>
Submission Group	<p>送信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>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。</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 から LIABILITY へ • LIABILITY から ASSET へ • EXPENSE から REVENUE へ • REVENUE から EXPENSE へ • BALANCE から FLOW へ • FLOW から BALANCE 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>

表 4-4 (続き) カスタム・メンバーの属性

属性	説明
SwitchTypeForFlow	次のルールを使用する FLOW 勘定の勘定科目タイプの変更を指定します。 <ul style="list-style-type: none"> • ASSET から EXPENSE へ • EXPENSE から ASSET へ • LIABILITY から REVENUE へ • REVENUE から LIABILITY へ • BALANCE から FLOW へ • FLOW から BALANCE へ この勘定科目タイプ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。
UserDefined1、UserDefined2、UserDefined3	ディメンション・メンバーのカスタム情報を保管します。最大 256 文字まで入力できます。関数 UserDefined1、UserDefined2、UserDefined3 は、この属性に保管されているテキストを取得します。

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表します。エンティティにより、データの連結パスが定義されます。エンティティ・ディメンションに対して連結を実行すると、ディメンション階層の定義に従って、子から親へデータがロール・アップされます。複数の親の下位にエンティティ子メンバーを配置することで、複数の連結パスを作成できます。エンティティ・メンバーの属性表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 4-5 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。
説明	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。

表 4-5 (続き) エンティティ・メンバーの属性

属性	説明
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どうかを指定します。エンティティが内部取引エンティティの場合は Y を指定し、内部取引エンティティではない場合は N を指定します。 ICP を選択するメンバーは、 ICP デイメンション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(-) • シャープ記号(#) • ピリオド(.) • プラス記号(+) • セミコロン(;) • スラッシュ(/) 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。
SecurityClass	エンティティのデータにアクセスできるユーザーの有効なセキュリティ・クラスの名前を指定します。セキュリティ・クラス名には、80 文字まで入力できます。
UserDefined1、UserDefined2、UserDefined3	エンティティのカスタム情報を保管します。最大 256 文字まで入力できます。関数 UserDefined1 、 UserDefined2 、 UserDefined3 は、この属性に保管されているテキストを取得します。

シナリオ・メンバーの定義

シナリオ・デイメンションは、予算、実績、予測などの一連の関連データを表します。シナリオ・メンバーの属性表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。**Oracle Hyperion 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 4-6 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。 ConsolidateYTD を N に設定した場合、ZeroViewForAdj および ZeroViewforNonAdj オプションの両方とも期別に設定する必要があります。
DefaultFreq	シナリオに対してデータ入力がある有効な期間タイプを指定します。この属性は必須です。 たとえば、Monthly という値は、四半期期間や年次期間ではなく、月次期間でのみ入力データを抽出できることを示します。期間単位は、アプリケーション・プロファイルで定義する必要があります。
DefaultParent	シナリオのデフォルトの親を指定します。
DefaultView	視点バーでシナリオ・ビューが選択された場合に使用するデータ・ビュー(年次累計または期別)を指定します。この属性は必須です。「YTD」または「期別」を指定します。 シナリオのデフォルト・ビューを変更し、ライン・アイテムの詳細が入力された場合は、まずライン・アイテムの詳細を抽出して保存してください。次に、シナリオからライン・アイテムの詳細を削除してから、ビューを変更します。リロードする前に、抽出されたライン・アイテムの詳細を、新しいデフォルト・ビューと一致するように変更する必要があります。
DefFreqForICTrans	内部取引データのデフォルト期間単位を指定します。この属性は有効な期間単位である必要があります、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。
説明	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。

表 4-6 (続き) シナリオ・メンバーの属性

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

表 4-6 (続き) シナリオ・メンバーの属性

属性	説明
UserDefined1、UserDefined2、UserDefined3	シナリオのカスタム情報を保管します。最大 256 文字まで入力できます。関数 UserDefined1、UserDefined2、UserDefined3 は、この属性に保管されているテキストを取得します。
UsesLineItems	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。 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: <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せん。 シナリオがライン・アイテム詳細を最初に受け入れず、そのシナリオにデータが存在する場合、シナリオまたはシナリオ内の勘定科目のライン・アイテム詳細を有効にできません。そのシナリオのライン・アイテムを有効にするには、既存のデータを削除し、メタデータのライン・アイテムを有効にし、データをアプリケーションにリロードする必要があります。
ZeroViewForAdj	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。 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。
ZeroViewForNonadj	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。 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。

アプリケーション設定の定義

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

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

アプリケーション設定の属性表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 4-7 アプリケーション設定の属性

属性	説明
ConsolidationRules	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。 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。 R: 値ディメンションの比例値を導出します。比例データは保管されないことに注意してください。 N: デフォルトの連結と消去を使用します。
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。

表 4-7 (続き) アプリケーション設定の属性

属性	説明
EnableMetadataSecurityFiltering	<p>ユーザーにすべてのディメンション・メンバーを表示するか、アクセス権を持っているディメンション・メンバーのみを表示するかを指定します。次のディメンション・メンバーがフィルタ処理されます。</p> <ul style="list-style-type: none"> シナリオ エンティティ 内部取引パートナー(ICP) 勘定科目 カスタム <p>ユーザーがアクセス権を持っていないディメンション・メンバーを除外するには Y を指定します。この属性のデフォルトは N です。</p>
FDMApName	<p>Oracle Hyperion Financial Data Quality Management アプリケーションの名前</p>
ICPEntitiesAggregationWeight	<p>値ディメンションの [ICP Top] メンバーに集約する内部取引パートナー・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。</p>
MaxCellTextSize	<p>セル・テキストに使用できる最大文字数を指定します。1900 から 2,147,483,646 までを指定します。0 から 1899 までの値は無効です。デフォルト値は 8,000 です。</p>
MaxNumDocAttachments	<p>ユーザーごとのドキュメントの添付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。</p>
MaxDocAttachmentSize	<p>ドキュメントの添付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。</p>
NodeSecurity	<p>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。</p>
OrgByPeriodApplication	<p>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。</p>
SupportSubmissionPhaseforAccounts	<p>プロセス管理のフェーズ送信を、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。</p> <p>有効な値は Y または N です。デフォルトは N です。</p>

表 4-7 (続き) アプリケーション設定の属性

属性	説明
SupportSubmissionPhaseforCustom	<p>プロセス管理のフェーズ送信を、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。</p> <p>有効な値は Y または N です。デフォルトは N です。</p> <p>属性の数は、アプリケーションに定義されたカスタム・ディメンションの数に基づいています。カスタム・ディメンション別名を反映するため、属性名が変更されます。たとえば、属性名のカスタムがカスタム・ディメンション別名で置換されません。</p>
SupportSubmissionPhaseforICP	<p>プロセス管理のフェーズ送信を、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。</p> <p>有効な値は Y または N です。デフォルトは N です。</p>
UsePVAForBalanceAccounts	<p>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。</p>
UsePVAForFlowAccounts	<p>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。</p>
UseSecurityForAccounts	<p>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。</p>
UseSecurityForCustom	<p>アプリケーションのカスタム・ディメンションをセキュリティで保護するかどうかを指定します。セキュリティでカスタム・ディメンションを保護する場合は Y を指定し、セキュリティで保護しない場合は N を指定します。</p> <p>属性の数は、アプリケーションに定義されたカスタム・ディメンションの数に基づいています。カスタム・ディメンション別名を反映するため、属性名が変更されます。たとえば、属性名のカスタムがカスタム・ディメンション別名で置換されません。</p>
UseSecurityForEntities	<p>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。</p>
UseSecurityForICP	<p>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。</p>

表 4-7 (続き) アプリケーション設定の属性

属性	説明
UseSecurityForScenarios	シナリオをセキュリティで保護するかどうかを指定します。セキュリティでシナリオを保護する場合は Y を指定し、セキュリティで保護しない場合は N を指定します。
UseSubmissionPhase	プロセス管理のフェーズ送信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が必要です。 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、Oracle Hyperion Financial Management では、エンティティ階層のフィルタとしてアクティブなシステム勘定科目が使用されます。アクティブ勘定科目は、親レベルでデータを保管し、ICP ディメンションを使用して子に関する情報を保管する内部取引勘定科目です。親への子エンティティの連結ステータスがアクティブであるか非アクティブであるかを指定します。

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

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

連結メソッドの属性表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 4-8 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(-) • シャープ記号(#) • 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>POWN および POWNMIN については、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>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>

表 4-8 (続き) 連結メソッドの属性

属性	説明
UsedByCalcRoutine	このメソッドを自動出資比率計算に使用するかどうかを指定します。出資比率計算にこのメソッドを使用する場合は Y を指定し、使用しない場合は N を指定します。

連結メソッドの使用方法

連結メソッドは、連結および出資比率の計算プロセスで使用されます。

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

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

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

手動での連結メソッドの割当て

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

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

Scenario: Applicable scenario

Year: Applicable year

Period: Applicable period

Entity: A parent entity

Value: [None]

Account: [Method] system-generated account

Custom: ToCurrency

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

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

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

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

POWN または POWNMIN 連結メソッドの使用

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

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

POWNMIN 計算

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

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** であるとしています。

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:

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

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

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

POWN 計算

ただし、連結がフラット階層で行われる場合は通常、その処理では最終出資比率 (**POWN**) を資本企業の連結比率として使用します。この場合、D のグループへの連結比

率は30%になります。エンティティ D からの資本では、少数株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。
- POWNMN を使用すると、エンティティ D に割り当てられる連結比率は40% (ステージングされた連結プロセスを使用)になります。

通貨の定義

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

通貨ディメンション・タイプを使用してディメンションを作成し、通貨の属性表に記載されている属性を使用して、アプリケーションの通貨を定義します。アプリケーションに必要な各通貨の通貨ディメンションにメンバーを作成します。

表 4-9 通貨の属性

属性	説明
Currency	通貨の名前を指定します。この属性は必須です。通貨名は一意にする必要があります。空白を含み、80 文字まで入力できます。 通貨名には、次の文字は使用しないでください。 <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(-) • シャープ記号(#) • ピリオド(.) • プラス記号(+) • セミコロン(;) • スラッシュ(/)
説明	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。
DisplayInICT	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。

表 4-9 (続き) 通貨の属性

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

セル・テキスト・ラベルの定義

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

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

セル・テキスト・ラベルを定義するガイドラインは、次のとおりです:

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

ノート:

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

次の連結システム勘定科目は、エンティティ・ディメンションの各親に必要で、連結プロセスで使用されます。

ノート:

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 4-10 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。 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。
[CTRL]	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。

表 4-10 (続き) 連結に使用されるシステム勘定科目

勘定科目	説明
Consol1、Consol2、Consol3	連結メソッド。0-255 の数値。

出資比率勘定科目

出資比率の計算には、次の出資比率システム勘定科目が使用されます。

ノート:

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 4-11 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 です。 所有株式の総数は、未処理株式の総数以下である必要があります。
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。 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必要があります。
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。 未処理株式の数または割合を入力します。割合の場合は 100 を入力します。
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。 未処理議決権株式の数または割合を入力します。割合の場合は 100 を入力します。
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。

 **ノート:**

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. 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メタデータ・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、必要に応じてステップ 3 を繰り返します。
5. 「ファイルの保存」をクリックします。

 **ノート:**

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

内部取引パートナーの設定

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

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

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

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

調整勘定科目は、消去の完了時に **[Elimination]** 値ディメンションの 2 つの内部取引勘定科目間の差異を保管する勘定科目です。調整勘定科目は、ICP 勘定科目として設定できます。調整勘定科目を ICP 勘定科目として設定するには、**IsICP** メタデータ属性を **Y** または **R** に設定し、消去が該当の ICP メンバーに書き込まれるようにします。調整勘定科目を ICP 勘定科目とし

て設定しない場合は、IsICP 属性を N に設定します。その結果、消去は[ICP None]に書き込まれます。

連結時に、有効な内部取引エンティティ間の取引が消去されます。[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 4-12 システム生成の ICP 要素

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

システムで生成された ICP メンバーの編集

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

ノート:

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. 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メタデータ・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、必要に応じてステップ 2 を繰り返します。
4. 「**ファイルの保存**」をクリックします。

ノート:

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。

 **ノート:**

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

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

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

CurrencyName は通貨のラベルです。

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

 **ノート:**

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. システム生成の値メンバーが含まれているメタデータ・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、ステップ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。

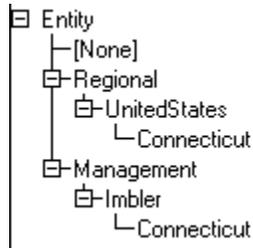
 **ノート:**

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

セキュリティに基づくメタデータのフィルタ

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

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



APP フォーマットのメタデータ・ファイルの作成

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

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

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

- メンバー
- 階層

ノート:

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

エンティティ・ディメンション、シナリオ・ディメンション、勘定科目ディメンションおよびカスタム・ディメンションのメタデータは、メンバー・セクションおよび階層セクションに配置されます。カスタム・ディメンションにディメンション・セクシ

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。

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

ノート:

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(「通貨」、「メンバー」、「階層」など)を続ける必要があります。TRUE と FALSE は、TRUE の場合は Y、FALSE の場合は N です。アポストロフィ(')で始まる行は、コメント行とみなされ無視されます。

次のセクションはメタデータ・ファイルで使用できます。

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

ファイル・フォーマット

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

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

majorNumber と *minorNumber* は、1つまたは2つの桁から構成されます。*majorNumber* には先行ゼロを含めることができ、*minorNumber* には後続ゼロを含めることができます。

majorNumber と *minorNumber* を含める必要があり、小数点にはピリオド(.)以外は使用しないでください。次の例は、有効なファイル・フォーマットを示しています:

```
!FILE_FORMAT = 11.12

!FILE_FORMAT = 11.120

!FILE_FORMAT = 011.120

!FILE_FORMAT = 011.12
```

バージョン

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

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

この例では、有効なバージョン値を示しています。

```
!VERSION = 11.1.4749
```

アプリケーション設定

メタデータ・ファイルのこのセクションは、**Oracle Hyperion 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、[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
```

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
```

[通貨の定義](#)を参照してください。

この例では、通貨の属性を指定します。

```
!CURRENCIES
```

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

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

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

メンバー

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

```
!MEMBERS=Customers
```

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

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

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

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

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

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

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

ノート:

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

次の各トピックでは、勘定科目、シナリオ、エンティティ、カスタム、値および ICP の各ディメンションのメンバー・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

勘定科目ディメンション・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
```

[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・ディメンション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・ディメンション・メンバーの構文:

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

[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 ;Default  
Parent=#root
```

```
Budget;MTD;Periodic;Periodic;Periodic;N; ; ; ;Y; ;10;Y;N;N;MTD ;Default  
Parent=#root
```

エンティティ

エンティティ・ディメンション・メンバーの構文:

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

[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・ディメンション・メンバーの構文:

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

[カスタム・メンバーの定義](#)を参照してください。

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

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

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

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

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

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

表 4-13 値ディメンションの説明

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

ノート:

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

内部取引パートナー

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

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

内部取引パートナー・メンバーの構文:

```
Label;SecurityClass;Descriptions
```

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

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

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

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

連結メソッド

メタデータ・ファイルのこのセクションは、連結メソッドを定義します。

連結メソッドの構文:

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

[連結メソッドの定義](#)を参照してください。

この例では、連結メソッドの属性を指定します。

```
!CONSOLIDATION_METHODS
```

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

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

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

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

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

階層

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

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

 **ヒント:**

カスタム・ディメンションのノード・レコードには、3番目の属性が含まれていません。カスタム階層を参照してください。

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

```
!HIERARCHIES=<>
```

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

```
!HIERARCHIES=Customers
```

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

```
;AllCustomers;0
```

```
AllCustomers;Customer2;1
```

```
AllCustomers;Customer3;1
```

```
AllCustomers;Customer4;1
```

```
AllCustomers;Customer5;1
```

 **ノート:**

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

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

勘定科目の階層

勘定科目ディメンションの階層の構文:

```
parentmemberlabel;childmemberlabel
```

この例では、勘定科目ディメンションの階層を指定します。

```
!HIERARCHIES=Account

; [None]

; ExchangeRates

ExchangeRates; Rate1

ExchangeRates; Rate2

; Plug

; NetProfit

NetProfit; NetIncome

NetIncome; GrossMargin

GrossMargin; Sales

GrossMargin; TotalCosts

TotalCosts; Purchases

TotalCosts; Salaries

TotalCosts; OtherCosts

NetIncome; AdminExpenses

NetIncome; InterestCharges

NetProfit; Taxes
```

シナリオ階層

シナリオ・ディメンションの階層の構文:

```
parentmemberlabel;childmemberlabel
```

この例では、シナリオ・ディメンションの階層を指定します。

```
!HIERARCHIES=Scenario
```

```
;Actual
```

```
;Budget
```

エンティティ階層

エンティティ・ディメンションの階層の構文:

```
parentmemberlabel;childmemberlabel
```

この例では、エンティティ・ディメンションの階層を指定します。

```
!HIERARCHIES=Entity
```

```
;[None]
```

```
;Regional
```

```
Regional;UnitedStates
```

```
UnitedStates;California
```

```
California;Sunnyvale
```

```
California;FosterCity
```

カスタム階層

カスタム・ディメンションの階層の構文:

```
parentmemberlabel;childmemberlabel;AggregationWeight
```

この例は、カスタム・ディメンションの階層を指定しています:

```
!HIERARCHIES=Products
```

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

```
;AllProducts;0
```

```
AllProducts;Golf;1
```

```
Golf;GolfBalls;1
```

```
Golf;GolfShoes;1
```

```
Golf;GolfTees;1
```

```
Golf;GolfClubs;1
```

メタデータ・ファイルに含まれないディメンション

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

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

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

メタデータの参照整合性

Oracle Hyperion 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で有効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。

ノート:

データのロックにより、格納済データは変更できなくなります。階層が変更された場合は、集約データの変更が可能になります。古い集約データに影響を与えずに階層の合計を変更するには、既存の階層を変更するのではなく、新しい階層を追加して古い階層を維持する必要があります。

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のあるメタデータ属性の、特定の変更に関する情報が含まれています。

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

メタデータ・ログ・ファイルの参照整合性エラー

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

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

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

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

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

メタデータのマージ・ユーティリティの使用

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

ノート:

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

使用方法

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

ここで:

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

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

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

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

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

例

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

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

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

```
-i true
```

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

メタデータのマージ・ユーティリティを使用するには:

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

メタデータのロード

メタデータ・ファイルをロードすると、**Oracle Hyperion 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。

ノート:

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

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

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

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

注意:

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

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

表 4-14 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	<p>ディメンション・メンバーがロード・ファイルとアプリケーションのデータベースに存在する場合、データベースのメンバーはロード・ファイルのメンバーに置換されます。ロード・ファイルに参照されない他のディメンション・メンバーがデータベースにある場合、データベースのメンバーは変更されません。</p> <p>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。</p>
置換	<p>アプリケーション・データベース内のすべてのディメンション・メンバーが削除され、ロード・ファイルのメンバーがデータベースに入られます。</p> <p>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。</p>
ロードする前にメタデータをすべてクリア	<p>アプリケーション・データベース内のすべてのディメンション・メンバーと、対応するデータ、仕訳および内部取引が削除されます。</p> <p>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。</p>
整合性チェック	<p>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。メタデータの参照整合性を参照してください。</p>

 **ノート:**

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。

 **ノート:**

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 ファイル拡張子も受け入れますが、XML または APP ファイル拡張子の使用をお勧めします。

4. **オプション: 「メタデータとデータのクリア**」を選択します。

 **ノート:**

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

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

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

 **ノート:**

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。[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:

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

 **ノート:**

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。
11. **オプション:** ログ・ファイルをダウンロードするには、「**ダウンロード・ログ**」をクリックします。「**開く**」をクリックしてログ・ファイルを表示するか、「**保存**」をクリックしてファイルをローカルに保存する場所を選択します。

メタデータ・ロードの変更の表示

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

メタデータ・ロードの変更レポートには、次の詳細が含まれます:

- 追加されたメンバー
- 削除されたメンバー
- メンバー属性の変更(説明の変更は含まない)
- 新規の親/子関係
- 削除された親/子関係
- カスタム・ディメンションでの親子関係の集約加重の変更

メタデータ・ロードのレポートのサンプル出力

** Entity **

Added member WestRegion.

Changed IsICP for China from Y to N.

Added Parent/Child Regional/Asia.

Removed Parent/Child Europe/Bulgaria.

** Account **

Renamed member SalesIC to SalesInterco.

Deleted member EastSales.

** Scenario **

Added member Forecast.

メタデータ・ロードの変更を表示するには:

1. 「ナビゲート」、「管理者」、「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リロードする必要があります。

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

メタデータを抽出するには:

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

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

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

 **ノート:**

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

4. ファイル・フォーマットを選択します:
 - **Financial Management (.app)**
 - **Financial Management (.xml)**
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って、抽出したファイルをダウンロードします。

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。
8. **オプション:** ログ・ファイルをダウンロードするには、「**ダウンロード・ログ**」をクリックします。「**開く**」をクリックしてログ・ファイルを表示するか、「**保存**」をクリックしてファイルをローカルに保存する場所を選択します。

5

メンバー・リストの管理

メンバー・リストを使用すると、ディメンション内のメンバーのサブセットを指定できるようになり、メンバーの階層を参照する時間を短縮できます。たとえば、エンティティ・ディメンションの視点で **Italy**、**France** および **UK** が頻繁に使用される場合、頻繁に使用されるメンバーを含む **European Region** という名前のメンバー・リストを作成できます。エンティティ・ディメンションの階層全体を参照するのではなく、このリストから **European** 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。

Oracle Hyperion Financial Management では、2 タイプのメンバー・リストが使用されます。

- システム定義メンバー・リスト
- ユーザー定義メンバー・リスト

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

ユーザー定義メンバー・リストには、ユーザーが指定したメンバーが含まれます。**Financial Management** のすべてのディメンションのメンバー・リストを作成できます。

静的および動的メンバー・リストを作成できます。

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

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

ノート:

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可能です。

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。

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。

ノート:

メンバー・リスト名に引用符(" ")またはアンパサンド(&)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 ()**: どのディメンションにメンバー・リストがあるかを指定し、各ディメンションのメンバー・リストを定義します。各ディメンションのメンバー・リストの数やそのディメンション内にある各メンバー・リストの名前を定義します。
- **Sub EnumMembersInList ()**: 各メンバー・リスト内のメンバーを定義します。

ノート:

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

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

EnumMemberLists

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

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

表 5-1 EnumMembersInList の構文

構文	説明
<code>HS.Dimension = "Element"</code>	<code>Element</code> はディメンションです。例: <code>If HS.Dimension = "Entity" Then</code>
<code>HS.MemberListID=n</code>	数値 ID でメンバー・リストを指定します ここで、 <i>n</i> は、 <code>EnumMemberLists</code> 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 <code>HS.MemberListID = 1</code>
<code>HS.AddEntityToList Member</code> <code>HS.AddMemberToList Member</code>	エンティティ以外のディメンションのリストにメンバーを追加します ここで、 <i>Member</i> はメンバー名です <code>HS.AddEntityToList</code> <code>"UnitedStates, ""Maine"</code> <code>HS.AddMemberToList "July"</code>
<code>HS.AddEntityToList Parent, Member</code>	エンティティ・ディメンションのリストにメンバーを追加します ここで、 <i>Parent</i> は追加しているメンバーの親で、 <i>Member</i> はエンティティ・ディメンションのメンバーです。 <code>HS.AddEntityToList</code> <code>"UnitedStates", "California"</code>
<code>HS.Entity.List</code>	エンティティ・ディメンションのリストにメンバーを追加します <code>HS.Entity.List("""[Base]")</code>
<code>HS.MemberListEntity</code> <code>HS.MemberListScenario</code> <code>HS.MemberListYear</code> <code>HS.MemberListPeriod</code>	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 "United States", "Massachusetts"

        HS.AddEntityToList "United States", "Rhode Island"

        HS.AddEntityToList "United States", "Maine"

    ElseIf HS.MemberListID = 2 Then

        HS.AddEntityToList ,"United States" "Connecticut"

    ElseIf HS.MemberListID = 3 Then

        HS.AddEntityToList "United States", "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"

```

 ノート:

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

動的メンバー・リスト

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

メンバー・リストを構築するには、Oracle Hyperion Financial Management の関数および引数を使用できます。

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

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

    If HS.MemberListID=1 Then

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

        'Entities are read into an array.

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

            'Loops through all entities.

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

                HS.AddEntityToList "",ELi(i)

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

            End If

        Next

    End If

End If
```

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

動的 POV メンバー・リスト

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

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

```

Sub EnumMemberLists ()

Dim EntityLists (5)

If HS.Dimension = "Entity" Then

    EntityLists (1) = "AllEntities"

    EntityLists (2) = "AppCur"

    EntityLists (3) = "NoappCur"

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

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

    HS.SetMemberLists EntityLists

End If

End Sub

Sub EnumMembersInList ()

If HS.Dimension = "Entity" Then

    If HS.MemberListID = 1 Then

        HS.AddEntityToList "", "Corp_Ops"

        HS.AddEntityToList "", "China"

        HS.AddEntityToList , "", "Germany"

        HS.AddEntityToList "", "UK"

```

YearPOV=HS.MemberListYear

PerPOV=HS.MemberListPeriod

EntPOV=HS.MemberListEntity

```

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

    If ( EntPOV <> "" ) Then

        EntList=HS.Node.List("E#" & EntPOV,"[Descendants]", "S#" & ScenPOV &
        "..Y#" & YearPOV & ".P#" & PerPOV)

        If IsArray(EntList) Then

            For each Ent in EntList

                If Ent <> "[None]" Then

                    If HS.Node.Method("S#" & ScenPOV & ".Y#" & YearPOV & ".P#" &
                    PerPOV & ".E#" & EntPOV & "." & Ent)="GLOBAL" Then

                        If HS.MemberListID = 4 Then HS.AddEntityToList "",Ent

                    End If

                    If HS.Node.POwn("S#" & ScenPOV & ".Y#" & YearPOV & ".P#" &
                    PerPOV & ".E#" & EntPOV & "." & Ent) > 0.5 Then

                        If HS.MemberListID = 5 Then HS.AddEntityToList "",Ent

                    End If

                End If

            End If

        End If

    End If

Next

```

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リ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。

ノート:

Web のポップアップ・ブロックの例外に Oracle Hyperion Financial Management 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。

ノート:

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 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。

 ノート:

無効なメンバー・リストがロードされた場合、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:

```
EPMHFM-102: Error executing VBScript,
An unexpected error has occurred. Check server log or contact your system
administrator for more information.
```

6. **オプション:** ログ・ファイルをダウンロードするには、「**ダウンロード・ログ**」をクリックします。「**開く**」をクリックしてログ・ファイルを表示するか、「**保存**」をクリックしてファイルをローカルに保存する場所を選択します。

メンバー・リストの抽出

アプリケーションからメンバー・リストを抽出できます。抽出されたメンバー・リストは、ASCII ファイルとして保存されます。デフォルトでは、メンバー・リスト・ファイルに LST ファイル拡張子が使用されます。メンバー・リストを抽出した後は、テキスト・エディタでメンバー・リストの情報を表示および変更できるようになります。

メンバー・リストを抽出するには:

1. アプリケーションを開きます。
2. 「**連結**」、「**抽出**」、「**アプリケーション要素**」の順に選択します。
3. 「**メンバー・リスト**」セクションで、「**抽出**」をクリックします。
4. ブラウザに表示されるダウンロードの手順に従って、抽出したファイルをダウンロードします。
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. **オプション:** ログ・ファイルをダウンロードするには、「**ダウンロード・ログ**」をクリックします。「**開く**」をクリックしてログ・ファイルを表示するか、「**保存**」をクリックしてファイルをローカルに保存する場所を選択します。

システム生成のリストとディメンション

次の表に、システム生成のリストの名前と、各リストがそれぞれのディメンションで使用できるかどうかを示します。

表 5-2 システム生成のリストとディメンション

システム・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]		<input type="checkbox"/>							

表 5-2 (続き) システム生成のリストとディメンション

システム・シナリオ リスト	エンティティ	勘定科目	ICP	カスタム	値	年	期間		
[Descendants]	<input checked="" type="checkbox"/>								
[Children]		<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>				
[Base]		<input checked="" type="checkbox"/>							
[Parents]		<input checked="" type="checkbox"/>							
[Ancestors]		<input checked="" type="checkbox"/>							
[System]			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					
[Currencies]						<input checked="" type="checkbox"/>			
[Consolidation Method]						<input checked="" type="checkbox"/>			
[Inputs]							<input checked="" type="checkbox"/>		
[Adjustments]							<input checked="" type="checkbox"/>		
[Totals]							<input checked="" type="checkbox"/>		
[Default Currencies]							<input checked="" type="checkbox"/>		
[First Generation]									<input checked="" type="checkbox"/>
[Second Generation]									<input checked="" type="checkbox"/>

表 5-2 (続き) システム生成のリストとディメンション

システム・ リスト	シナリオ	エンティテ イ	勘定科目	ICP	カスタム	値	年	期間	
[Third Generatio n]									<input checked="" type="checkbox"/>
[Fourth Generatio n]									<input checked="" type="checkbox"/>
[Fifth Generatio n]									<input checked="" type="checkbox"/>
[Sixth Generatio n]									<input checked="" type="checkbox"/>

6

仕訳の管理

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が、この ASCII テキスト・ファイルは、Oracle Hyperion Financial Management アプリケーションにロードできます。必要な場合、ASCII ファイルはロードする前にテキスト・エディタを使用して編集できます。

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

仕訳ファイルの作成

仕訳ファイルは、マルチバイト文字セット(MBCS)対応の ASCII フォーマットを使用するか、Little Endian バイト・オーダーによって Unicode フォーマットにエンコードされたファイルを使用して作成できます。デフォルトでは、仕訳ファイルには JLF ファイル拡張子が使用されます。

セキュリティ情報ファイルには、次のセクションを含めることができます。

- ファイル・フォーマット
- バージョン
- 仕訳グループ
- 標準
- 繰返し
- ヘッダー - シナリオ、年、期間

感嘆符(!)で始まる行は、仕訳ファイルの新しいセクションの開始を示しており、その後に有効なセクション名(「年」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
#	シャープ記号

文字	説明
%	パーセント記号
	パイプ記号
?	疑問符
;	セミコロン
~	チルダ

 **ノート:**

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。

 **ノート:**

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、仕訳の抽出に使用した Oracle Hyperion Financial Management のバージョンが含まれます。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。

 **ノート:**

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。

ノート:

自動逆仕訳テンプレート用に繰返しテンプレートを作成することはできません。タイプ属性の値は、通常の **R** にする必要があります。

次に、繰返しテンプレートの構文を示します。

```
!RECURRING = <label>, <balancing attribute>, <type>, <value>, <journal
group>, <securityclass>, <SingleParent.SingleEntity>
```

```
!DESC=<journal description>
```

```
<parent.entity>, <account>, <ICP>, <CustomDimensionName>, <amount
type>, <amount>, <line item desc>
```

ヘッダー・セクション

このファイル・セクションには、シナリオ、年および期間に関する情報が含まれます。仕訳ファイルのヘッダー情報とこれに対応する明細行がヘッダー・セクションの後に続きます。ここでは、特定のシナリオ、年および期間に基づいて仕訳情報が編成されます。仕訳ファイルには、複数のヘッダー・セクションを指定できます。

この構文では、シナリオ、年および期間を指定します：

```
!SCENARIO= Actual
```

```
!YEAR = 2014
```

```
!PERIOD = January
```

ヘッダー・セクションの次に、**Actual** というシナリオの、**2014** 年の **July(1月)** の仕訳の詳しい情報を入れます。

ヘッダー・セクションの仕訳サブセクションでは、指定したシナリオ、期間および年の仕訳詳細情報を定義します。

[表 1](#) に、仕訳サブセクションで使用される属性を示します。これらの属性は、通常の仕訳、および繰返しテンプレートと標準テンプレートで使用されます。

表 6-1 仕訳の属性の説明

属性	値
<label>	ユーザー定義の仕訳のラベル(最大 20 文字まで)
<balancing attribute>	<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
<type>	<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>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>

表 6-1 (続き) 仕訳の属性の説明

属性	値
<status>	<ul style="list-style-type: none"> W = 作業中 S = 送信済 A = 承認済 P = 転記済 R = 却下済
<value dimension>	<ul style="list-style-type: none"> [Contribution Adjs] [Parent Adjs] <Entity Curr Adjs> <Parent Curr Adjs>
<journal group>	<p>オプション: ユーザー定義パラメータ(最大 30 文字)</p> <p>あらかじめグループがプレロードされている必要があります。</p>
<security class>	<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>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>
<journal description>	<p>仕訳の説明(最大 255 文字)。各行が「!DESC=」構文で始まる場合は、複数行の説明をロードできます。</p>
<parent.entity>	<p>エンティティ・ディメンションの有効なメンバー。値ディメンションの Contribution Adjs、Parent Adjs および ParentCurrency Adjs の各メンバーにのみ、親が必要です。この属性は、Single Parent.Single Entity 属性が使用されていない場合にのみ使用されます。</p>
<account>	<p>勘定科目ディメンションの有効なメンバー。標準仕訳では、入力勘定になります。勘定科目タイプは、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・ディメンションの有効なメンバー。この属性はオプションですが、少なくとも「ICP None」と指定する必要があります。</p>
<custom>	<p>オプション: カスタム・ディメンションの有効なメンバー</p>
<amount type>	<ul style="list-style-type: none"> D = 借方 C = 貸方
<amount>	<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>

表 6-1 (続き) 仕訳の属性の説明

属性	値
<line item description>	オプション: 仕訳詳細の説明(最大 50 文字)。

次の構文によって、通常仕訳を指定します。

```
!JOURNAL = <label>, <balancing attribute>, <type>, <status>, <value
dimension>, <journal group>, <SecurityClass>,
<SingleParent.SingleEntity>
```

```
!DESC=<journal description>
```

```
<parent.entity>, <account>, <ICP>, <CustomDimensionName>, <amount
type>, <amount>, <line item desc>
```

仕訳のロード

作業中、却下済、送信済、承認済および転記済の仕訳と、標準および繰返しの仕訳テンプレートをロードできます。連結プロセスによって作成されるため、自動連結仕訳はロードできません。

仕訳をロードする前に、最初に仕訳をロードする期間を開いておく必要があります。*Oracle Hyperion Financial Management ユーザーズ・ガイド*の期間の管理を参照してください。

仕訳は、置換モードでロードされます。これにより、新しい仕訳データがロードされる前に、特定仕訳ラベルのデータすべてがクリアされます。開いている期間には、転記済の仕訳をロードする必要があります。転記済の自動逆仕訳をロードする場合、次の期間に承認済の逆仕訳が自動的に生成され、生成された逆仕訳を手動で転記する必要があります。

デフォルトでは、仕訳ファイルには **JLF** ファイル拡張子が使用されます。ロード・プロセスでは、**TXT** や **CSV** などのファイル拡張子を受け入れますが、**JLF** ファイル拡張子の使用をお勧めします。

デフォルトのロード・オプションを変更すると、オプションがすべての行に対して更新されます。特定の行を有効にしてその行のオプションを更新する上書きオプションを使用できます。

ロード・プロセスが完了すると、ログのリンクが表示されるため、エラーを表示できません。

 **ノート:**

Web のポップアップ・ブロックの例外に Oracle Hyperion Financial Management 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:

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

 **ノート:**

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

3. **オプション:** Firefox をブラウザとして使用している場合は、複数のファイルをロードできます。まだ選択していない場合は「**複数の選択**」を選択し、「**参照**」をクリックしてファイルを検索します。

一度に最大 10 個のファイルを選択できます。次に、システムによって選択したファイルが詳細行に移入されます。

また、ファイルをファイル選択コンテンツ領域にドラッグ・アンド・ドロップできます。

4. **オプション:** 「**追加**」をクリックして、仕訳をロードするために行をさらに追加します。

 **ヒント:**

行を削除するには、行を選択して「**削除**」をクリックします。

5. **オプション:** 行のデフォルトのファイル設定を上書きするには、「**上書き**」をクリックします。

 **ヒント:**

ロード・オプションをデフォルト値にリセットするには、「**リセット**」をクリックします。

6. 「**ロード**」をクリックします。

 ノート:

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

仕訳の抽出

アプリケーションから、仕訳、繰返し仕訳および仕訳テンプレートを抽出できます。仕訳を抽出するシナリオ、年、期間、エンティティおよび値ディメンションを選択できます。また、抽出する仕訳ステータス、仕訳タイプおよび仕訳の残高の種類を選択できます。

仕訳を抽出すると、マルチバイト文字セット(MBCS)をサポートする ASCII ファイルに保存されます。デフォルトでは、仕訳ファイルには JLF ファイル拡張子が使用されます。メンバー・リストを抽出した後は、テキスト・エディタでメンバー・リストの情報を表示および変更できるようになります。

自動連結仕訳を外部システムに抽出できますが、その仕訳を Oracle Hyperion Financial Management に再インポートできません。これらの仕訳を抽出する場合、自動連結仕訳の残高の種類は空白です。

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

仕訳を抽出するには:

1. 「連結」、「抽出」、「仕訳」の順に選択します。
2. 仕訳を抽出するシナリオおよび年を指定します。
3. **オプション:** 仕訳を抽出する期間を選択します。
4. **オプション:** 抽出するエンティティおよび値ディメンションを、1 つ以上選択します。

自動連結仕訳を抽出するには、値メンバーとして[Proportion]または[Elimination]、あるいはその両方を選択する必要があります。

 ノート:

期間、エンティティおよび値ディメンションでは、特定のメンバーを選択しないと、ディメンションのすべてのメンバーを抽出するとみなされます。ただし、メンバーを明示的に選択すると、ディメンションの横に複数の選択があることを示すプラス記号(+)が表示されます。

5. 「区切り文字」に、ファイルの仕訳データを区切るために使用する区切り文字を入力します。次の文字が有効です:

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

 ノート:

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

6. 抽出する**仕訳の種類**を選択します:

- テンプレート
- 繰返しテンプレート
- 仕訳

7. **オプション**: 抽出する仕分けのラベルおよびグループを入力します。

ワイルドカードとしてパーセント記号(%)を使用できます。

自動連結仕訳の場合、連結ルールで指定した監査取引の種類値に対して「ラベル」フィールドで問合せを実行できます。ラベルには番号を使用できません。

8. 抽出する仕訳のステータス、種類および残高の種類を選択するか、「**すべて**」を選択します。

9. 「**抽出**」をクリックします。

10. ブラウザに表示されるダウンロードの手順に従います。

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

7

データ・フォームの管理

データ・フォームは、特定のデータを表示および入力するために設定された、事前定義済の行および列が含まれるテンプレートです。ユーザーは、データ・フォームへの行および列の追加や削除を行うことはできません。変更できるのは、フォームの設定時に指定されているディメンションの視点の選択のみです。あるフォームから別のフォームへのリンクを定義し、リンクされたフォーム内のデータをユーザーがドリル・スルーして表示できるようにすることが可能です。

データ・フォームを作成するには、管理者またはデータ入力フォームの管理のセキュリティ役割に割り当てる必要があります。データ・フォームのロード、抽出および削除には、「データ入力フォームの管理」役割が割り当てられている必要があります。

データ・フォームは2つの方法で作成できます：

- フォーム・デザイナーのデータ・フォーム・オプションの使用
- スクリプト・ビューのスクリプトの書込み

デフォルトでは、データ・フォームのファイル拡張子は **WDF** です。

デザイナーおよびスクリプト・ビューを容易に切り替えることができます。スクリプトにエラーが含まれる場合、スクリプト・ビューで表示されます。スクリプト・ビューから、データ・フォームを印刷することもできます。

データ・フォームのデザイナー・ビューから「フォームを開く」に切り替えることができます。フォームの保存を求めるプロンプトが表示され、データ・フォームが表示されます。

フォーム・デザイナーでのデータ・フォームの作成

フォーム・デザイナーを使用して、視点を設定し、フォームの詳細、行、列およびヘッダー・オプションを指定します。

データ・フォームの作成後、スキャンして妥当性を確認できます。フォーム・スクリプトが生成および検証されます。エラーがある場合、スクリプト・ビューに表示されます。[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：

- [視点の設定](#)
- [フォームの詳細の指定](#)
- [フォームの行および列オプションの指定](#)
- [フォーム・ヘッダーの指定](#)

データ・フォームを作成するには：

1. 「**連結**」、「**ドキュメント**」の順に選択します。
2. 「**新規**」、「**データ・フォーム**」の順にクリックするか、「**アクション**」、「**新規**」、「**データ・フォーム**」の順に選択します。

デフォルトでは、フォーム・デザイナーが表示されます。

 ヒント:

「デザイナー」ページが自動的に表示されない場合、「**デザイナー**」をクリックするか、「**アクション**」、「**デザイナー**」の順に選択します。

3. データ・フォームの視点を設定します。
4. フォームの詳細を指定します。
5. 行および列オプションを指定します。
6. **オプション**: 行または列を追加するには、「**新規行の追加**」、「**新規列の追加**」の順にクリックするか、「**アクション**」、「**新規行の追加**」または「**新規列の追加**」の順に選択します。

 ヒント:

行または列を削除するには、「**列/行の削除**」をクリックするか、「**アクション**」、「**列/行の削除**」の順に選択します。

7. ヘッダー・オプションを指定します。

 ヒント:

フォームをデフォルト値または最後に保存した定義にリセットするには、「**リセット**」をクリックします。

8. **オプション**: 「**スキャン**」をクリックして、フォームの妥当性を確認します。
9. 「**保存**」をクリックしてフォームを保存し、フォームの情報を入力して、「**保存**」をクリックします。

視点の設定

POV バーを使用して、バックグラウンド POV および選択可能 POV を定義できます。フォームのバックグラウンド POV は、各ディメンションに、データ・フォームに表示される初期値を指定します。フォームの Selectable POV は、ユーザーが選択できるメンバーのディメンションで構成されます。

POV バーのディメンションには、メンバーを選択してバックグラウンド POV メンバーを指定します。メンバー・リストを選択すると、データ・フォーム・ユーザーがメンバーを選択できる選択可能 POV として使用されます。2つを超えるアイテムおよび1つを超えるメンバーまたはメンバー・リストは選択できません。

動的 POV メンバー・リストを定義した場合、それを使用して、選択したディメンションに対して有効なメンバーを迅速に選択できます。「メンバー選択」ダイアログ・ボックスで、選択したディメンションの有効な動的 POV メンバー・リストが、ダイアログ・ボックスの下部に接尾辞@POV を付けて表示されます。

POV を使用する場合は、次のルールが適用されます。

- メンバーまたはメンバー・リストが選択されず、行または列でディメンションが使用されていない場合は、ディメンションの初期値としてユーザー POV のメンバーが使用されません。
- ディメンションのバックグラウンドおよび選択可能 POV に値を指定し、バックグラウンド POV のメンバーが選択可能 POV のメンバー・リストに含まれていない場合は、選択可能 POV のメンバー・リストにある最初のメンバーがバックグラウンド POV メンバーとして使用されます。
- 階層内のすべてのメンバーを操作できるように、ユーザーには、メンバー階層へのフル・アクセス権限が必要です。たとえば、ユーザーがカスタム 4 のすべてのメンバーにアクセスする場合、このケースでは、親エンティティ CustomTop にアクセスできるようにする必要があります。

フォーム・デザイナー・グリッドは、行および列を含みます。新しいフォームの場合、デフォルトでは、シナリオ・ディメンションがグリッド列に使用され、期間ディメンションがグリッド行に使用されます。列または行をフォームに追加し、POV バーからグリッドの行および列にディメンションをドラッグ・アンド・ドロップしてデータ・フォーム・レイアウトを定義できます。同じ行または列に 1 つを超えるディメンションをドラッグ・アンド・ドロップする場合、ディメンションの順序を変更できます。ディメンションをクリックすると、「メンバー選択」ページからメンバーを選択してグリッドまたは POV に含めることができます。

グリッドの行または列を選択すると、対応する行および列のオプションが「フォームの詳細」プロパティ・ペインに表示されます。行または列に計算式を含めることができます。この場合、行または列にディメンションをドラッグ・アンド・ドロップできないので、かわりにプロパティ・ペインを使用して、行または列に計算式を入力する必要があります。行に使用されるディメンションは、列に使用できません。この逆の場合も同様です。

フォームの詳細の指定

「フォームの詳細」セクションでは、データ・フォームのグリッド、印刷、表示および非表示オプションなどのフォーム・プロパティを指定します。データ・フォームのユーザーに指示を含め、データ・フォームで使用可能なオンデマンド・ルールを指定できます。

ノート:

「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、[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。[ルール・ファイルの作成](#)を参照してください。

データ・フォームのオンデマンド・ルールを指定するには:

1. 「**連結**」、「**ドキュメント**」の順に選択します。
2. 「**新規**」、「**データ・フォーム**」の順にクリックするか、「**アクション**」、「**新規**」、「**データ・フォーム**」の順に選択します。

デフォルトでは、フォーム・デザイナーが表示されます。

ヒント:

「デザイナー」ページが自動的に表示されない場合、「**デザイナー**」をクリックするか、「**アクション**」、「**デザイナー**」の順に選択します。

3. 「**フォームの詳細**」パネルの「**オン・デマンド・ルール**」で、次のいずれかの方法を使用します。
 - ルール名がわかっている場合、カンマ区切りリストのテキスト・ボックスに、名前を入力します。
 - 使用可能なルールを検索するには、テキスト・ボックスの横にある「**編集**」ボタンをクリックし、「**使用可能なルール**」ポップアップ・ダイアログ・ボックスから、フォームで使用する1つ以上のルールを選択し、「**OK**」をクリックします。
4. 「**OK**」をクリックします。
5. データ・フォームを保存します。

フォームの行および列オプションの指定

行および列に指定するディメンション要素は、バックグラウンド **POV** または選択可能 **POV** に設定される要素を上書きします。行または列ごとに1つのメンバー・リストのみを指定できます。

各オプションの有効な値については、[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、[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、[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。

 ノート:

「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・ヘッダーの指定

「ヘッダー」セクションでは、ラベルの表示、説明または両方などのディメンションごとのヘッダー・プロパティ、スタイル、メンバーおよびメンバー・リストのラベルまたは説明の最大長、長さが固定かどうかを指定できます。ディメンションごとに別の長さを指定できます。

「ヘッダー」セクションにも「その他」ボックスがあり、次のように使用できます。

- 構文が **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。
- ディメンションの **HeaderOption** キーワードに無効な構文がフォームに含まれている場合は、「その他」ボックスにその構文が表示されます。

各オプションの有効な値については、[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。

 ノート:

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、Oracle Hyperion 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。

 ノート:

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 7-1 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・ディメンション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 Link 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。 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)
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	非推奨です。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が上書きされます。
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	フォームでディメンション・ヘッダーをどのように表示するかを指定します。ラベルまたは説明、あるいはその両方を表示し、スタイル属性、最大幅または固定幅を設定します。

表 7-1 (続き) 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
Instructions	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。
Link	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、フォーム内の次の他の非表示の設定を上書きします： SuppressInvalidRows 、 SuppressNoDataRows 、 SuppressZeroRows 、 SuppressInvalidCols 、 SuppressNoDataCols 、 SuppressZeroCols 。
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 FormNumDecimals の小数点設定を上書きします。
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。
Override	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 でディメンションを非表示にするかどうかを指定するために使用します。
POVOrder	POV でのディメンション名の順序を指定するために使用します。
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。
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。

表 7-1 (続き) 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
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ディメンションのメンバーを指定します。
ShowDescriptions	ディメンションのメンバーの説明を表示します。
ShowLabels	ディメンションのメンバーのラベルを表示します。
String	列、行またはセルにテキスト文字列を追加します。
Style	行、列、セルまたはディメンション・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。
SuppressInvalidCols	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。
SuppressNoDataCols	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。
SuppressZeroCols	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・ディメンションで使用できません。

ノート:

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

ノート:

要約行は非表示にできません。

- 要約行に指定された **POV** と同じものを含むリスト行。ただし、リスト行は、メンバーを追加中のディメンションのリストを含みます。リストは少なくとも、要約行リストのメンバーを含む必要があります。

ノート:

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・ディメンションの[Base]メンバー・リストを指定します。**Row 2** には、メンバー・リストの合計と **AddMember** のアイコンで使用するメンバー・リストを指定します。

 **ノート:**

フォーム内の複数の行で **AddMember** オプションを使用できますが、要約行ごとに対応するリスト行を入力する必要があります。

BackgroundPOV

このキーワードは、フォームのバックグラウンド・ディメンションのメンバーを指定します。バックグラウンド **POV** に指定されていないディメンションは、動的とみなされ、フォームを開いたときにユーザーの **POV** から取得されます。

階層内のすべてのメンバーを操作できるように、ユーザーには、メンバー階層への完全なセキュリティ・アクセス権限が必要です。たとえば、ユーザーがカスタム 4 のすべてのメンバーにアクセスする場合、このケースでは、親エンティティ **CustomTop** にアクセスできるようにする必要があります。

 **ノート:**

バックグラウンド **POV** のメンバーが、そのディメンションの **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 7-2 BackgroundPOV キーワードの構文

パラメータ	説明
Scenario	有効なシナリオ・メンバーの名前。
View	有効なビュー。
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。

表 7-2 (続き) BackgroundPOV キーワードの構文

パラメータ	説明
Entity	有効なエンティティ・メンバーの名前。
Value	有効な値メンバーの名前。
Account	有効な勘定科目メンバーの名前。
ICP	有効な ICP メンバーの名前。
Custom	有効なカスタム・メンバーの名前。

 ノート:

Background POV 内のディメンションのすべてのメンバーを指定する必要はありません。メンバーを指定していないディメンションの場合、ユーザーの視点からのディメンション・メンバーが使用されます。

例

```
BackgroundPOV=S#Actual.Y#2014.P#January.W#Periodic.V#<Entity
Currency>.A#Sales.I#[ICPNone].C4#[None]
```

Blank

フォーム内の空白の行、列またはセルを指定します。このオプションは、行または列の定義内、またはセルの上書き定義内で使用します。空白の行、列またはセルは空でかつ読取り専用であり、データまたは POV が含まれていないため、セルのコンテキスト・メニューは使用不可になっています。Blank オプションはサーバー側の計算式とみなされるため、SCalc を使用する任意の箇所で使用できます。

例

```
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。

列に指定したディメンション要素は、[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。

ノート:

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](#)
- [Override](#)
- [ReadOnly](#)
- [SCalc](#)
- [Scale](#)
- [String](#)
- [Style](#)

構文

`Cn=CalcExpression`

`Cn=POVExpression`

表 7-3 列のキーワードの構文

パラメータ	説明
n	列番号。
CalcExpression	SCalc 、 Blank または String を使用します。 SCalc 、 Blank および String を参照してください。

表 7-3 (続き) 列のキーワードの構文

パラメータ	説明
POVExpression	ディメンションの有効な交差またはメンバー・リスト

例

```
C1=S#Actual.P#July,CustomHeader:Actual_July
```

```
C2=S#Actual.P#August,(Override 2,3, P#July)
```

```
C3=S#Budget.P#September
```

```
C4=S#Calc(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#2014.P#January.w#<Scenario
View>.E#UnitedStates.Connecticut.V#USD.A#Sales.I#[ICP
NONE].C1#Golfballs.C2#Customer2.C3#[None].C4#[None]

C1=S#Actual

C2=S#Budget

C3=SCalc(Col(2)-Col(1)),CustomHeader:Variance

C4=SCalc(Col(3)/Col(1)),CustomHeader:Variance %

R1=A#Sales

R2=A#Purchases

R3=A#OtherCosts

R4=SCalc(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。[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。

 ノート:

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_Linkx
```

```
Linkx=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。

 ノート:

セル・テキストが含まれているデータ・フォームを抽出すると、セル・テキストのエントリのタイトル(最初の 69 文字)のみが抽出されます。

構文

```
CellText:<cell text label>
```

例

```
R1=A#Salaries, CellText:[Default]
```

```
R2=A#Salaries, CellText:Validation
```

CustomHeader

列または行のカスタム・ヘッダーを指定します。このオプションは、計算された列または行があり、特定の数式を非表示にして、**Variance** などの説明で置き換える場合に便利です。行

または列でディメンションがネストしている場合、カスタム・ヘッダーはそのディメンションのみではなく、その行または列のヘッダー全体に適用されます。

セミコロンを区切り文字に使用して、後続するセルのカスタム・ヘッダーを指定できます。たとえば、この構文を使用すると、行内の3つのディメンション・ヘッダーにシナリオ、年および月のカスタム・ヘッダーを指定できます。

```
R2=S#Actual.Y#2014.P#January,CustomHeader:Scenario;Year;Month
```

一部のヘッダー・セルのみを置き換えるには、ピリオド(.)を使用して、元のヘッダーを表示する必要があることを示します。また、ピリオドを省略することによってヘッダーを非表示にすることもできます。たとえば、次の構文は、最初のディメンションの元のヘッダーを表示し、2番目のディメンションのヘッダーを非表示にし、3番目のディメンションの元のヘッダーを表示します。

```
R2=S#Actual.Y#2014.P#January,CustomHeader:.;;.
```

▲ 注意:

計算される行または列に **CustomHeader** オプションを使用する場合は、ヘッダーの最初のセルのみにテキストを設定できます。

構文

```
CustomHeader:HeaderName
```

HeaderName には、使用するヘッダーを指定します。

✎ ノート:

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(col(1)-col(2)),customheader:Variance
```

CustomHeaderStyle

このオプションは、ある行または列のヘッダーにカスタム属性を割り当てます。このオプションは、すべてのヘッダーにわたる単一ディメンションではなく、ある行または列の全体に対応するヘッダーに適用される点でキーワード **HeaderOption** の **Style** オプションとは異なります。競合が存在する場合、**CustomHeaderStyle** オプションは、キーワード **HeaderOption** の **Style** オプションより優先的に使用されます。

`CustomHeaderStyle` を使用するかわりに 2 つのスタイルを混在させるには、以下の例に示すように、`CustomHeaderStyle` オプションの前にセミコロンを挿入します。 [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、次のステップに従ってこのキーワードを削除してください:

1. **BackgroundPOV** キーワードから、DynamicPOV キーワードで指定されたすべてのディメンションを除去します。
2. DynamicPOV キーワードを削除します。

FormInputBoxLength

入力ボックスの幅を指定します。デフォルトは 20 文字の幅です。



ノート:

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。

ノート:

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 桁まで表示されません。

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、各ディメンションのヘッダーのデフォルトを指定するために使用します。現在サポートしているオプションは次のとおりです：

- **Length** - 行ヘッダーの最大長。最大長と最小長を同一にする場合は、数値を使用しても、単語「Fixed」を使用してもかまいません。

ノート:

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

```
HeaderOptionDimension=Length:n
```

```
HeaderOptionDimension>ShowDescription
```

```
HeaderOptionDimension>ShowLabel
```

```
HeaderOptionDimension=Style:Property:Value;Property:Value...
```

Dimension には行ディメンションの名前、*n* にはヘッダーの長さまたは「Fixed」、*Property* にはスタイルのプロパティ、*Value* にはプロパティの値をそれぞれ指定します。Style を参照してください。

例

```
HeaderOptionPeriod=Length:4
```

```
HeaderOptionScenario=ShowDescription
```

```
HeaderOptionsAccount=ShowLabel,Style:font-style:italic
```

HideInPov

このキーワードは、各ディメンションの **HideInPOV** オプションを指定するために使用します。この値が存在する場合、ディメンションはデータ・フォームの **POV** バーに表示されません。この値が見つからない場合、ディメンションは **POV** バーに表示されます。

構文

```
HideinPOV=Dimension
```

ディメンション名は、ショート名またはロング名で表示されており、カンマ区切りのリストです。

ノート:

選択可能なすべてのディメンションが **POV** バーに表示されるため、ディメンションが選択可能な **POV** の一部である場合、このディメンションは無視されます。

例

次の例では、データ・フォームが開いている場合、シナリオ、年、期間が **POV** バーで非表示です。

```
HideinPov=S,Y,P
```

Instructions

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
```

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。[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。

ノート:

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_Linkx`

`Linkx=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。

ノート:

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

ノート:

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、または [FormNumDecimals](#) によって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。

**ノート:**

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。[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
```

Override

このオプションは、POVの様々なディメンション・メンバー、式の計算式または1つ以上連続する列または行のテキストを指定します。また、スタイルの変更にも使用します。このオプションは、行または列の定義内で使用します。

ノート:

連続しないセルを上書きするには、フォーム・オプションの「その他」フィールドに上書きを入力します。複数の上書きを入力するには、各上書きをカンマで区切ります。次の例は、個々の3つのセルを上書きします。

```
Override(1,1,string("455")),Override(3,3,string("23")),
Override(5,5,string("2234"))
```

Override オプションでは、メンバー・リストを使用できません。競合が存在する場合は最後の値を使用することで、複数の上書きを1つのセルに適用できます。ほとんどの場合、1つのセルに複数の上書きを混在させることもできます。たとえば、ある行の上書きが **NumDecimals** の値を指定し、ある列の上書きが **Scale** の値を指定する場合、同じディメンションが使用されている場合を除き、競合は存在しません。先行するセミコロンを使用すると、**Style** の値を混在させることができます。

上書きが交差する場合は、**Override** キーワードの後にセミコロンを含むことで上書きを混在させることもできます。セミコロンがない場合、フォーム内で上書きが競合すると、行の値は列の値よりも優先するため、行に対して定義されたスタイルが使用されます。行と列の定義の上書きを混在させるには、行のキーワードにセミコロンを含む必要があります。

構文

```
Override(StartCell,EndCell,Override)
```

パラメータ	説明
StartCell	上書きの開始位置を表す整数値。行の上書きが定義されている場合、このパラメータは、上書きが適用される開始列を示します。この例では、上書きは列2から開始します。 R2=A#Sales, Override(2,3,A#SalesTP)
EndCell	上書きの終了位置を表す整数値。前述の例では、上書きは列3で終了します。

パラメータ	説明
Override	<p>POV。たとえば、別の勘定科目で上書きするには、A#newacct を指定します。別のシナリオと勘定科目で上書きするには、A#newacct.S#newscenario 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 • String • Style

例

次の例では、行 2 の列 2 および列 3 が、期間に **January**、月の売上金額のかわりに **PriorSales** の金額を使用して上書きされます:

```
R2=A#Sales, Override(2,3,Y#2014.P#January.A#PriorSales)
```

次の例では、行 2 の列 2 および列 3 が、**Sales1**、**Sales2**、**Sales3** の平均を計算する式で上書きされます。

```
R2=A#Sales, Override(2,3,SCalc((A#Sales1+A#Sales2+A#Sales3)/3))
```

次の例では、行 2 の列 2 および列 3 が、読み取り専用オプションで上書きされます。

```
R2=A#Sales, Override(2,3,readonly)
```

POVOrder

このキーワードは、**POV** でのディメンション名の順序を指定するために使用します。ディメンション名は、ショート名またはロング名のいずれかを指定することができ、表示する順序で、カンマ区切りリストに示されます。リストに指定されないディメンションは、ディメンション番号順に、リストの末尾に追加されます。

例

次の例では、**POV** バーでのディメンションの順序は、勘定科目、期間、年、シナリオです。

 ノート:

勘定科目ディメンションが非表示として指定された場合、期間が **POV** バーに最初に表示されます。

```
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。

 ノート:

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。

 ノート:

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。

ノート:

1 つの行で複数のリストを使用できます。

行に指定したディメンション要素は、[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
- Override
- ReadOnly
- SCalc
- Scale
- String
- Style

構文

Rn=CalcExpression

Rn=POVExpression

表 7-4 行のキーワードの構文

パラメータ	説明
n	行番号。
CalcExpression	SCalc、Blank または String を使用します。 SCalc、Blank および String を参照してください。
POVExpression	ディメンションの有効な交差またはメンバー・リスト

例

```
R1=A#Sales.I#[ICP Entities],AddMember:I{[Base]}
```

```
R2=A#Sales.I{[Base]}
```

```
R3=A#HeadCount.I#[ICP None],NoSuppress
```

```
R4=A#Purchases.I#[ICP None],CustomHeader:ABC
```

```
R5=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。

**ノート:**

Excel にエクスポートする場合は、読取り専用のフォーマットが維持されます。

例

```
C4=S#Actual.Y#2014,ReadOnly
```

ReportDescription

このキーワードは、フォームの説明を指定します。

構文

```
ReportDescription=Description
```

Description には、フォームの説明を指定します。説明には、255 文字まで入力できます。

例

```
ReportDescription=Intercompany Detail
```

ReportLabel

このキーワードは、フォームの名前を指定します。このキーワードは必須です。

構文

```
ReportLabel=Label
```

Label には、フォームの名前を指定します。名前には、**40** 文字まで入力できます。次の文字は名前には使用できません:

- アスタリスク(*)
- アット記号(@)
- 円記号(バックスラッシュ) (\)
- コロン(:)
- カンマ(,)
- 中カッコ({})
- スラッシュ(/)
- 小なり記号と大なり記号(<)
- シャープ記号(#)
- カッコ (
- ピリオド(.)
- パイプ(|)
- プラス記号(+)
- 疑問符(?)
- 引用符""
- セミコロン(;

ノート:

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せん。

たとえば、実際の行ヘッダーの幅がフォームの合計幅の **25%** で、指定された値が **40%** の場合、コンテンツが **25%** で表示されるため、行ヘッダーは **40%** にはなりません。ただし、指定された値が **10%** の場合、行ヘッダーの幅は実際の **25%** の幅から、指定された **10%** の幅にスクロール・バー付きで収まるように小さくなります。行ヘッダーに複数の列がある場合は、すべての列の幅が加算され、フォームの合計幅と比較されます。

構文

```
RowHeaderPct:n
```

n には、**1-100** の値を指定します。

例

```
RowHeaderPct=30
```

SCalc

SCalc 関数を使用して、列および行に、標準の算術演算子を使用するカスタム式を作成できます。この値は、行または列の定義内、またはセルの上書き定義内で使用します。指定された計算式がアプリケーション・サーバーで実行されます。次の例では、列 1 から列 2 を減算します:

```
SCalc(col(1)-col(2))
```

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>])
```

ノート:

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。

SCalc の計算式では、次のタイプのアイテムを使用できます。

- ディメンション・メンバーの参照。勘定科目ディメンション・メンバーの **Purchases** および **OtherCosts** を参照します。R6=SCalc(((A#Purchases)-(A#OtherCosts))*100)
- 構文 Cell (*RowIndex*, *ColumnIndex*) を使用したセル参照。次の例は、フォーム内の 4 行 2 列目のセルを参照します。R1=SCalc(Cell(4,2))
- 構文 Row (*RowIndex*) を使用した行の参照。次の例は、4 行目を 2 行目で除算します。R3=SCalc(Row(4)/Row(2))

ノート:

行または列にメンバー・リストが含まれる場合は、リストのメンバーの合計を使用して計算が行われます。

- 構文 Col (*ColumnIndex*) を使用した列の参照。次の例は、1 列目と 3 列目を加算します。C4=SCalc(Col(1)+Col(3))

- ネストされた計算式。括弧を使用してネストします。
- SCalc 計算を使用したセルへの前方参照。

 **ノート:**

SCalc 計算は、データが保存されるまで行われません。計算結果は、フォームをリフレッシュするまでフォームには表示されません。

SCalc 行または列は、計算内で別の SCalc 行または列を参照できますが、SCalc 行または列で別の SCalc 行または列を前方参照することはできません。たとえば、次の SCalc の前方参照は許されません。

```
C1=A#Sales
C2=A#COGS
C3=SCalc(Col(1)-Col(2))
C4=SCalc(Col(3)/Col(1)*100)
```

Scale

このオプションは、計算されない列、行またはセルの位取りを指定します。

構文

Scale:*n*

n には、-12 から 12 の値を指定します。

例

C4=A#Inventory,Scale:2

SelectablePOVList

このキーワードは、ユーザーが選択できるディメンションのメンバーを指定します。

選択可能なディメンションの初期値は、そのディメンションに初期値が指定されていれば、バックグラウンド POV から取得します。初期値が指定されていなければ、初期値はユーザーの POV から取得します。リストの初期値が無効な場合は、リストの最初のメンバーがデフォルトになります。

ユーザーが新しいメンバーを選択すると、選択されたメンバーはユーザーの視点に組み込まれます。

構文

```
SelectablePOVList=Dimension{MemberList}
```

```
SelectablePOVList=Dimension{[SystemList]}
```

```
SelectablePOVList=Dimension{Parent.[SystemList]}
```

表 7-5 SelectablePOVList キーワードの構文

パラメータ	説明
Dimension	<p>選択可能なディメンションを表す、次のいずれかの文字</p> <ul style="list-style-type: none"> • S (シナリオの場合) • W (ビューの場合) • E (エンティティの場合) • V (値の場合) • P (期間の場合) • A (勘定科目の場合) • I (ICP の場合) • C (カスタムの場合)
Parent	ディメンションの有効な親
MemberList	有効なメンバー・リストの名前
SystemList	有効なシステム・リストの名前

例

```
SelectablePOVList=S{ActualBudget}.Y{Years}.P{Months}.E{[Hierarchy]}.  
C1{ProductsAllocate}.C2{Customers}.C3{AllChannels}
```

ShowDescriptions

このキーワードは、フォームのディメンション要素の説明を表示するかどうかを指定します。デフォルトは **FALSE** です。

 ノート:

説明が存在しない場合は、ラベルが表示されます。エンティティの説明が存在しない場合は、エンティティ・ラベルのみが表示されます; 親ラベルは含まれません。

構文

```
ShowDescriptions=Boolean
```

Boolean には、TRUE または FALSE を指定します。

例

```
ShowDescriptions=True
```

ShowLabels

このキーワードは、フォームのディメンション要素のラベルを表示するかどうかを指定します。デフォルトは TRUE です。

[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せん。

構文

```
ShowLabels=Boolean
```

Boolean には、TRUE または FALSE を指定します。

例

```
ShowLabels=False
```

String

このオプションは、列、行またはセルにテキスト文字列を指定します。このオプションは、行または列の定義内、またはセルの上書き定義内で使用します。String オプションは、サーバー側の計算式とみなされるため、SCalc を使用する任意の箇所で使用できます。

例

```
C4=String ("Show this read-only string")
```

Style

このオプションは、行、列、セル、カスタム・ヘッダーまたはディメンション・ヘッダーのスタイル属性を指定します。このオプションは、行、列、セルの上書き、カスタム・ヘッダーまたはヘッダー・オプションの定義内で使用します。

▲ 注意:

フォームにスタイル属性を適用する場合は注意が必要です。Style オプションは強力な機能であるため、間違っても使用した場合、フォームの外観が大幅に変更される可能性があります。

ノート: 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、Oracle Hyperion 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。

Style オプション・キーワードを 1 行で 1 回使用して、複数のプロパティ:値ペアを追加できます。例:

```
C3=S#Actual.Y#@CUR(-1),Style:font-color:blue;background-color:red;font-weight:bold
```

Style Example 表では、使用できるプロパティと値を示します。完全なリストは、W3C の「Property Index」を参照してください。

表 7-6 Style の例

プロパティ	値
Color (前景)、Background-color	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

表 7-6 (続き) Style の例

プロパティ	値
Font	フォント・スタイル。例: Style: font-style: italic 1つの Style 値でフォントのプロパティを 6つ まで組み合わせることができます。例: Style: font-style:italic;font- family:Arial;font-size:12px
Font-weight	フォントの太さ。値は、 demi-bold 、 demi-light 、 extra-bold 、 light 、 extra-light 、 demi-light を含みます。例: Style: font-weight: extra-bold
Font-size	フォントのポイント・サイズ。例: Style: font-size: 12px
配置	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わされています。**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。

	<i>Actual</i>	
	<i>July</i>	<i>August</i>
Sales	500,000.00	600,000.00
Purchases	250,000.00	300,000.00

構文

`SuppressColHeaderRepeats=Boolean`

Boolean には、**TRUE** または **FALSE** を指定します。

例

`SuppressColHeaderRepeats=False`

SuppressInvalidCols

このキーワードは、無効なセルが入った列をフォームで非表示にするかどうかを指定します。デフォルトは **FALSE** です。

 ノート:

この設定に関係なく、管理者以外のユーザーには無効な列は表示されません。

構文

```
SuppressInvalidCols=Boolean
```

Boolean には、TRUE(無効な列を非表示にする場合)または FALSE(無効な列を表示する場合)を指定します。

例

```
SuppressInvalidCols=False
```

SuppressInvalidRows

このキーワードは、無効なセルが入った行をフォームで非表示にするかどうかを指定します。デフォルトは FALSE です。

 ノート:

この設定に関係なく、管理者以外のユーザーには無効な行は表示されません。

構文

```
SuppressInvalidRows=Boolean
```

Boolean には、TRUE(無効な行を非表示にする場合)または FALSE(無効な行を表示する場合)を指定します。

例

```
SuppressInvalidRows=False
```

SuppressNoDataCols

このキーワードは、データのないセルが入った列をフォームで非表示にするかどうかを指定します。デフォルトは FALSE です。

構文

```
SuppressNoDataCols=Boolean
```

Boolean には、**TRUE**(データの無い列を非表示にする場合)または **FALSE**(データの無い列を表示する場合)を指定します。

例

```
SuppressNoDataCols=False
```

SuppressNoDataRows

このキーワードは、データの無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**(データの無い行を非表示にする場合)または **FALSE**(データの無い行を表示する場合)を指定します。

例

```
SuppressNoDataRows=False
```

SuppressRowHeaderRepeats

このキーワードは、繰り返される行ヘッダーを非表示にするかどうかを指定します。デフォルトは **TRUE** です。

構文

```
SuppressRowHeaderRepeats=Boolean
```

Boolean には、**TRUE** または **FALSE** を指定します。

例

```
SuppressRowHeaderRepeats=False
```

SuppressZeroCols

このキーワードは、ゼロが入った列をフォームで非表示にするかどうかを指定します。デフォルトは **FALSE** です。

構文

```
SuppressZeroCols=Boolean
```

Boolean には、**TRUE**(ゼロの列を非表示にする場合)または **FALSE**(ゼロの列を表示する場合)を指定します。

例

```
SuppressZeroCols=False
```

SuppressZeroRows

このキーワードは、ゼロが入った行をフォームで非表示にするかどうかを指定します。デフォルトは **FALSE** です。

構文

```
SuppressZeroRows=Boolean
```

Boolean には、**TRUE**(ゼロの行を非表示にする場合)または **FALSE**(ゼロの行を表示する場合)を指定します。

例

```
SuppressZeroRows=False
```

相対期間の使用

行と列に次のキーワードを使用して、現在の **POV** に関連する期間を選択できます:

- @Cur
- @First
- @Last
- @Next
- @Prev

 ノート:

年と期間の値では、大文字と小文字は区別されません。

たとえば、**POV** で選択されている期間が 1 月の場合、この式を使用して作成された列または行には、1 月のデータが表示されます:

```
P#CUR+1
```

期間が現在の年以降に延長されたとき、または前の年を参照する必要があるときにシステムで相対的な年を適用するには、**Y#@Cur** を行または列の定義の一部として指定する必要があります。

たとえば、現在の期間が最後の期間である場合は、**C1=Y#@Cur.P#@Cur(+1)**を指定して、正しい年を戻す必要があります。現在の **POV** が **2013 December** の場合、**C1=Y#@Cur.P#@Cur(+1)**は **2014 January** を戻します。

現在の POV が 2013 December の場合、`C2=P#@Cur(+1)`は 2014 January を戻します (January が年の最初の期間の場合、December は最後です)。したがって、年の情報を正しく反映するには、列の定義に年を指定します。例: `C2=Y#@Cur.P#@Cur(+1)`

`@CUR` 関数を他のディメンションに使用して現在の POV を取得できます。ただし、他のディメンションの `@CUR` は、順序どおりではないため、`+2` などのオフセットを使用できません。たとえば、現在の POV が実績シナリオを持つ場合、列の有効な例は、`C1=S#@CUR` を使用して実績を戻すことになります。POV シナリオを予算に変更すると、この同じ列の定義が予算を戻します。

競合する属性の優先順位

データ・フォームの小数点以下の桁数、スケールおよび POV を設定する方法は複数あります。たとえば、小数点以下の桁数の設定は次のように行います。

- フォームの場合 - `FormNumDecimals=3`
- 列の場合 - `C4=A#Inventory,NumDecimals:1`
- 行の場合 - `R4=A#Sales,NumDecimals:2`
- セルの上書きの場合 - `R2=S#Actual.P#August, Override(2,7,P#July,NumDecimals:4)`

これらの属性のいずれかの設定が、同じ属性の競合する設定と交差する場合があります。たとえば、ある列がスケール設定 1 を持ち、交差する行がスケール設定 2 を持つ場合があります。小数点以下の桁数およびスケール属性に適用される値はこの優先順位に基づいていることを理解することは重要です。

- セルの上書き
- 行
- 列
- フォーム
- デフォルト - 小数点以下の桁数の場合、デフォルトは勘定科目から取得します。スケールの場合、デフォルトはエンティティの通貨から取得します。

ノート:

POV では、各ディメンションの値は個別に解決されます。たとえば、勘定科目ディメンションは、フォーム・レベルで設定され、シナリオ・ディメンションは行レベルで設定される場合があります。

競合する属性は、優先順位に従って解決されます。たとえば、小数点以下の桁数属性がセルの上書きで定義されている場合、この値が、フォーム、行、列またはデフォルトのかわりに使用されます。

同様に、セルに競合するデータと計算式が指定されている場合、次の優先順位が適用されます。

- SCalc
- データ

 ノート:

フォーム内の SCalc 式は左から右、上から下の順番に評価されます。たとえば、行 1 内のすべての列が順番に評価されると、次に行 2 のすべての列が評価されます。SCalc 式は、評価順序が優先する別の SCalc セルを参照できます。

データ・フォームの編集

データ・フォームは、フォーム・デザイナーを使用して編集できます。一度に編集できるデータ・フォームは 1 つのみです。

 ノート:

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リロードし、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。

 ノート:

Web のポップアップ・ブロックの例外に Oracle Hyperion Financial Management 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。

 ノート:

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られますが、WDF ファイル拡張子の使用をお勧めします。

4. **オプション:** 次のいずれかのステップを実行します:
 - 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、これらのステップを繰り返します。
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。

 ノート:

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リロードできます。新規スクリプトのモデルを使用することもできます。

 ノート:

スクリプトをアプリケーションにリロードする場合は、「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

この章の手順を使用して、データベースにデータを抽出します。

ノート:

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 ユーザーズ・ガイド*のデータの抽出に関する項を参照してください。

データを抽出し、**Oracle Essbase** データベースを使用して、データを分析し、レポートを作成できます。スター・スキーマを使用して、データを **Essbase** データベースに送信します。また、スター・スキーマはサードパーティ製のツールでも使用できます。

ノート:

データをスター・スキーマにエクスポートしたら、データベースを作成します。データベースを作成するには、使用しているリリースのデータベースのドキュメントを参照してください。

データベースにデータを抽出する前に、スター・スキーマを保管するデータベースのデータ・ソース名(DSN)を設定する必要があります。スター・スキーマを複数のデータベースに保管するには、データベースごとに **DSN** を作成します。手順については、*Oracle Enterprise Performance Management System インストレーションおよび構成ガイド*を参照してください。

データベースにデータを抽出するには、管理者または拡張分析のセキュリティ役割が割り当てられている必要があります。

拡張分析のセキュリティ役割が割り当てられていない場合、データの抽出に次の制限事項が適用されます:

- ICP およびすべてのカスタム・ディメンションの基本レベルのデータのみを抽出できます。
- 1つのシナリオ・ディメンションおよび1つの年ディメンションのみを選択できます。
- シナリオのデフォルトの期間単位の期間のみを選択できます。

データ・ソース名(DSN)の構成

「データベースへのデータの抽出」機能を使用するには、データ・ソース名(DSN)を構成して、データベース・サーバーとアプリケーション・サーバー間の接続を指定する必要があります。DSN はデータベース・サーバー名と、データベース・サーバーへのフル・アクセス権を持つユーザーのユーザー名とパスワードなどの、その他のデータベース関連の情報を指定します。

DSN を作成するには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「管理タスク」から「DSNの構成」をダブルクリックします。
3. 「データ・ソースの作成」をクリックします。
4. 宛先データベースのデータ・ソース名を入力します。
5. 「データベース・タイプ」ドロップダウン・リストから、データベース・タイプ「Oracle」、「SQL Server」または「DB2」を選択します。
6. ユーザー名とパスワードを入力します。

 **ノート:**

ユーザーに表を作成、更新および削除する権限があることを確認します。

7. サーバーの**ホスト**名を入力します。
8. **ポート**番号を入力します。
9. **オプション:** デフォルトの表領域またはファイルグループ名を入力します。
10. **オプション:** デフォルトのインデックス表領域またはファイルグループ名を入力します。
11. 「**接続のテスト**」をクリックします。
12. 「成功」の確認メッセージが表示されたら、「**保存**」をクリックして DSN を保存します。
13. 「**リフレッシュ**」をクリックして更新された DSN 情報を表示します。

接続を作成して保存すると、「DSNの構成」ページに表示されます。後で編集する必要がある場合は「DSNの編集」をクリックして編集できます。また、不要になった場合は削除できます。

スター・スキーマ

アプリケーションごとに複数のスター・スキーマを作成できます。各スター・スキーマの表名は、ユーザーが指定する接頭辞で始まります。すべてのディメンションから複数のメンバーを選択して、エクスポートする情報を最適に表すスター・スキーマを作成できます。

 **ノート:**

セル・テキストとライン・アイテムの詳細はスター・スキーマにエクスポートされません。

スター・スキーマ内のデータの組合せは、エクスポートするために選択するディメンションのメンバーに基づいて作成されます。選択するディメンションのメンバーが多くなると、スター・スキーマの作成に必要なデータの組合せの数が多くなるため、エ

クサポート・プロセスの完了に必要な時間が長くなります。データの組合せの数を計算するには、選択したメンバーの数をディメンションごとに乗算します。

▲ 注意:

すべてのディメンションのすべてのメンバーをエクスポートする 選択はしないでください。エクスポートするデータのセグメントを選択します。アプリケーションの大きさや、データの組合せ数によって、エクスポート時間が非常に長くなる場合があります。

たとえば、次のデータをエクスポートできます。

- シナリオ - 実績
- 年 - 2014
- 期間 - 1 月
- ビュー - 期別
- エンティティ - 地域、米国、フロリダ、コネチカット
- 値 - USD
- 勘定科目 - 粗利益、売上、総費用
- ICP - [なし]
- C1 - ボール、テニス・ボール、ゴルフ・ボール
- C2 - 全顧客、顧客 2、顧客 3、顧客 4、顧客 5
- C3 - [なし]
- C4 - [なし]

次に、作成されるスター・スキーマを **Oracle Essbase Integration Services** で使用して、情報を参照および使用する必要がある対象ユーザーを表すために、1 つ以上のデータ・キューブを作成できます。スター・スキーマには、これらのメンバーのデータの組合せが 180 含まれます ($1 * 1 * 1 * 1 * 4 * 1 * 3 * 1 * 3 * 5 * 1 * 1 = 180$)。

✎ ノート:

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、**Oracle Hyperion Financial Management** アプリケーションで表示されているものとは異なる順序で、メタデータが使用されます。

必要な場合は、現地通貨のデータのみを抽出できます。現地通貨のデータのみを抽出するには、視点の値ディメンションからエンティティの通貨メンバーを使用します。エンティティの通貨は、通貨のプレースホルダとして機能し、選択されている各エンティティのデフォルトの通貨を抽出します。

選択したシナリオが YTD の場合、データ抽出により期別の派生データが抽出されます。たとえば、シナリオが ZeroView=YTD であるとします。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。この視点に対して、期間ベースでデータベースへのデータの抽出が実行されている場合、最初の 3 つの期間の値は、それぞれ 100、-100 および 0 になります。

スター・スキーマのフォーマット

スター・スキーマを作成する場合は、抽出フォーマットの次のオプションのいずれかを選択します。

- 標準
- メタデータのみ
- 選択したメタデータのみ
- Essbase
- データ・ウェアハウス

選択する抽出フォーマットのオプションによって、使用するスキーマ・フォーマットが決定されます。各スキーマ・フォーマットは、異なる一連の表を生成します。次のスキーマ・フォーマットを使用できます。

- 標準 Essbase スキーマ - このスキーマは、抽出フォーマット・タイプが標準、メタデータのみ、選択したメタデータの場合に使用します。
- SQL および Essbase スキーマ - このスキーマは、抽出フォーマット・タイプが Essbase の場合に使用します。
- ウェアハウス正規化階層スキーマ - このスキーマは、抽出フォーマット・タイプがデータ・ウェアハウスの場合に使用します。

接頭辞表

各スキーマ・フォーマットでは、ディメンション表と 1 つのデータ・フィールドのキーを含む *PREFIX_FACT* 表が作成されます。また、*PREFIX_DIMENSION* 表および *HFM_EA_EXTRACT* も作成され、メタデータの抽出タイムスタンプが追跡されます。エンティティ・ディメンションの 2 つの表(*PREFIX_ENTITY* および *PREFIX_PARENT*)が作成されます。

SQL および Essbase 集約スキーマ・フォーマットでは、*PREFIX_DIMENSION_BASE* 表も作成されます。

ウェアハウス正規化階層スキーマ・フォーマットでは、*PREFIX_DIMENSION_PARENT* 表が作成されます。

ノート:

ビュー、年および親ディメンションには、*BASE* 表と *PARENT* 表は作成されません。

たとえば、リレーショナル表の接頭辞が *DEMO* で、フォーマットが Essbase の場合は、次の表が作成されます。

- HFM_EA_EXTRACT1
- DEMO_FACT
- DEMO_YEAR
- DEMO_VIEW
- DEMO_PARENT
- DEMO_SCENARIO および DEMO_SCENARIO_BASE
- DEMO_PERIOD および DEMO_PERIOD_BASE
- DEMO_VALUE および DEMO_VALUE_BASE
- DEMO_ENTITY および DEMO_ENTITY_BASE
- DEMO_ICP および DEMO_ICP_BASE
- DEMO_ACCOUNT および DEMO_ACCOUNT_BASE
- DEMO_CUSTOM および DEMO_CUSTOM_BASE

データの作成とスター・スキーマへのエクスポート

スター・スキーマを作成するには、スキーマの表を識別する接頭辞とエクスポートするセルのディメンションのメンバーを指定します。データをエクスポートする前に、アプリケーション・データが連結されていることを確認します。

▲ 注意:

Oracle Hyperion 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。

ノート: Oracle データベース・ユーザーの場合: 抽出されるテキスト・データは、NVARCHAR(Unicode)フォーマットで保管されます。SELECT 文で Oracle の translate コマンドを使用して、テキストを Unicode から ANSI フォーマットに変換します。たとえば、SELECT translate (LABEL using CHAR_CS) as LABEL FROM DEMO_ACCOUNT のように使用します。

大量のデータを抽出する場合は、抽出用に十分なディスク・スペースを確保してください。

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

スター・スキーマを作成し、Financial Management データを抽出するには:

1. 「連結」、「抽出」、「データ」の順に選択します。
2. 次のいずれかのタスクを行います:
 - POV ディメンションをテンプレートに保存した場合、「テンプレート」から、テンプレートを選択します。

- テンプレートを使用していない場合、エクスポートする POV ディメンション・メンバーを選択します。
3. 「抽出先」セクションの「タイプ」オプションで「データベース」を選択します。

 **ノート:**

管理者または拡張分析のセキュリティ役割が割当てられている場合のみ、このオプションを使用できます。

4. 「抽出フォーマット」で、次のいずれかのオプションを選択します:
 - 標準
 - **メタデータのみ** - メタデータのみを抽出します。
 - **選択したメタデータのみ** - 選択したディメンションのメンバーのメタデータのみを抽出します。
 - Essbase
 - データ・ウェアハウス
5. 「オプション」から、1つ以上のオプションを選択します。
 - 動的勘定科目の抽出
 - 算出データ
 - 派生データ
6. 「ライン・アイテムの詳細」から、オプションを選択します。
 - セルの合計要約
 - なし
7. 「スキーマ・アクション」から、「スター・スキーマの作成」を選択します。
8. 「宛先データベース(DSN)」から、スター・スキーマのエクスポート先のデータベースを選択します。

 **ノート:**

宛先データベースとして、データのエクスポート元と同じ Financial Management データベースを使用しないでください。

9. 「リレーショナル表の接頭辞」で、スター・スキーマの表を識別する接頭辞を入力するか、デフォルトのアプリケーション名の接頭辞を使用します。

 **ノート:**

接頭辞には、英数字を 10 文字まで入力でき、英字で始まる必要があります。アンダースコアは使用できません。デフォルトの接頭辞はアプリケーション名です。

10. 「抽出」をクリックします。

スター・スキーマの更新

定義済のスター・スキーマにデータをエクスポートできます。スター・スキーマを更新するとき、別のディメンションのメンバーを指定できます。

ノート:

スター・スキーマを更新すると、ファクト表が更新されます。

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

スター・スキーマを更新するには:

1. 「連結」、「抽出」、「データ」の順に選択します。
2. 次のいずれかのタスクを行います:
 - POV ディメンションをテンプレートに保存した場合、「**テンプレート**」から、テンプレートを選択します。
 - テンプレートを使用していない場合、エクスポートする POV ディメンション・メンバーを選択します。
3. 「抽出先タイプ」から、「**データベース**」を選択します。

ノート:

管理者または拡張分析のセキュリティ役割が割当てられている場合のみ、このオプションを使用できます。

4. 「抽出フォーマット」で、次のいずれかのオプションを選択します:
 - **標準**
 - **メタデータのみ** - メタデータのみを抽出します。
 - **選択したメタデータのみ** - 選択したディメンションのメンバーのメタデータのみを抽出します。
 - **Essbase**
 - **データ・ウェアハウス**
5. 「オプション」から、1つ以上のオプションを選択します。
 - **動的勘定科目の抽出**
 - **算出データ**
 - **派生データ**

6. 「**ライン・アイテムの詳細**」から、オプションを選択します。
 - セルの合計要約
 - なし
7. 「**スキーマ・アクション**」から、「**スター・スキーマの更新**」を選択します。
8. 「**宛先データベース(DSN)**」から、スター・スキーマのエクスポート先のデータベースを選択します。

 **ノート:**

宛先データベースとして、データのエクスポート元と同じ **Financial Management** データベースを使用しないでください。

9. 「**リレーショナル表の接頭辞**」で、スター・スキーマの表を識別する接頭辞を入力するか、デフォルトのアプリケーション名の接頭辞を使用します。

 **ノート:**

接頭辞には、英数字を 10 文字まで入力でき、英字で始まる必要があります。アンダースコアは使用できません。デフォルトの接頭辞はアプリケーション名です。

10. 「**抽出**」をクリックします。

スター・スキーマの削除

不要になったスター・スキーマは削除できます。スター・スキーマを削除すると、スター・スキーマのデータ、メタデータおよび表がすべて削除されます。

スター・スキーマを削除するには:

1. 「**連結**」、「**抽出**」、「**データ**」の順に選択します。
2. 「**抽出先タイプ**」から、「**データベース**」を選択します。
3. 「**宛先データベース(DSN)**」から、スター・スキーマを含むデータベースを選択します。
4. 「**リレーショナル表の接頭辞**」に、スター・スキーマの表を識別する接頭辞を入力します。

 **ノート:**

接頭辞には、英数字を 10 文字まで入力でき、英字で始まる必要があります。デフォルトの接頭辞はアプリケーション名です。

5. 「**スキーマ・アクション**」から、「**スター・スキーマの削除**」を選択します。

スター・スキーマのテンプレートの作成

スター・スキーマのテンプレートを作成し、POV に名前を付けて保存すると、再使用できます。

スター・スキーマのテンプレートを作成するには:

1. 「**連結**」、「**抽出**」、「**データ**」の順に選択します。
2. POV ディメンションのメンバーを選択します。
3. 「**宛先データベース(DSN)**」から、スター・スキーマのエクスポート先のデータベースを選択します。
4. 「**リレーショナル表の接頭辞**」に、スター・スキーマの表を識別する接頭辞を入力します。

ノート:

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

Oracle Hyperion 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。
- 内部取引パートナー(ICP)照合レポート: 連結するときに消去される内部取引を示します。

レポートごとに新しいレポート定義を作成するか、レポート定義を開いてそれを変更できます。

次の方法のいずれかを使用して、レポートを定義します:

- レポート・ページ・オプションを使用して、値を指定します。*Oracle Hyperion Financial Management ユーザーズ・ガイド*を参照してください。
- テキスト・エディタを使用して、スクリプトのレポート・オプションを指定します。

すべてのレポートのヘッダーに、日付と時刻、およびユーザー・フィールドが自動的に表示されます。視点の定義はレポートごとに異なります。レポートの詳細な例については、[仕訳レポート・スクリプトの定義](#)および[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ー照合レポートは、Financial Management のサンプル・アプリケーションをインストールするときに取り込まれます。これらのファイルは、Financial Management をインストールしたディレクトリの「Sample Applications」フォルダに配置されます。

仕訳レポート・スクリプトの定義

仕訳レポートを作成して、仕訳ステータスをチェックしたり、仕訳の調整を見直したりします。仕訳レポートを作成して、特定の仕訳または仕訳のリストの情報を表示できます。たとえば、特定のステータス、残高の種類、エンティティまたは勘定科目を持つ仕訳のみを表示するように選択できます。また、レポートに表示する列を選択したり、その列のソート順を変更することもできます。

レポート定義の仕訳レポートにエンティティや勘定科目のフィルタを設定できます。エンティティ・フィルタの構文は、`EntityFilter=ParentName.ChildName(UnitedStates.Maryland` など)です。勘定科目フィルタの構文は、`AccountFilter=MemberName(Sales` など)です。

この例では、サンプルの仕訳レポートの定義を示しています。

```
ReportType=Journal

ReportDescription=Tax Journals

POV=S#Actual.Y#2014.P#January.V#<Entity Curr Adjs>

DisplayColumn_0=Account,Ascending,NoRepeat,Yes,Label

DisplayColumn_1=ICP,,NoRepeat,No,Label

DisplayColumn_2=Products,,NoRepeat,No,Label

DisplayColumn_3=Markets,,NoRepeat,No,Label

StatusFilter=Working,Submitted,Rejected,Approved,Posted

TypeFilter=Regular

BalanceTypeFilter=Balanced,Unbalanced

EntityFilter=UnitedStates.Maryland

AccountFilter=Sales
```

内部取引照合レポート・スクリプトの定義

内部取引照合レポートを使用すると、内部取引を追跡して、分析や監査に役立てることができます。内部取引照合レポートは、内部取引パートナー(IsICP)属性を使用可能にして設定されたエンティティや内部取引パートナー・ディメンションの一致を示します。

内部取引パートナー(ICP)照合レポートは、連結時に消去される内部取引を表示します。内部取引は、内部取引パートナー・ディメンションで管理されます。ICP ディメンションは、勘定科目として存在できるすべての内部取引残高を含んでいます。データ・グリッド、データ・ロード、仕訳またはデータ・フォームを使用して、内部取引を入力できます。Oracle Hyperion Financial Management では、すべての勘定科目とカスタム・ディメンションについて、内部取引の詳細を追跡して消去できます。内部取引は、2つの内部取引エンティティの共通の親が最初に見つかった時点で消去されます。内部取引は、値ディメンションの[Elimination]メンバーを使用して消去されます。

ユーザー・インタフェースを使用して、内部取引照合レポートを作成できます。

レポートの勘定科目を選択するか、調整勘定科目オプションを使用できます。調整勘定科目オプションを使用すると、調整勘定科目に基づいて勘定科目や照合勘定科目が生成されます。戻し取引、カスタム・ディメンションまたは内部取引詳細を非表示にするように選択できます。レポートを印刷するとき、レポート設定を上書きして、必要に応じたレポートをカスタマイズできます。

内部取引照合レポートのメンバー・リストの選択

生成されたレポートに視点ディメンションを含むディメンション・メンバーを指定します。レポートの「エンティティ」および「パートナ」のメンバー・リストを選択できます。たとえば、「エンティティ」フィールドの「地域」のメンバー・リストを選択できます。「エンティティ」および「パートナ」のメンバー・リストを選択すると、「エンティティ」リスト内のすべてのエンティティと「パートナ」リスト内のすべてのエンティティを対比して ICP 取引が処理されます。ICP 属性が使用可能なエンティティの取引のみが処理されます。

内部取引照合レポートの勘定科目の選択

勘定科目および取引と照合する照合勘定を定義する必要があります。勘定科目と照合勘定を調整勘定に基づいて生成する場合は、「調整勘定」オプションを指定すると、勘定科目と照合勘定が自動的に生成されます。

内部取引照合レポートの小数点以下の桁数の指定

Decimal キーワードを内部取引照合レポート定義に追加して、レポートに表示する小数点以下の桁数を指定できます。小数点以下の桁数はデフォルトで 0-9 です。小数点以下の桁数を指定しない場合は、勘定科目に定義されているデフォルトの小数点設定が使用されます。

内部取引照合レポート用スタイル・シートの選択

レポートの作成時にドロップダウン・リストからスタイル・シートを選択できます。レポート定義に **StyleSheet** キーワードを追加して、レポートに使用するスタイル・シートを指定することもできます。たとえば、**StyleSheet=HFM_IntercompanyDefault.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、視点の値を USD から EUR に変更してレポートを生成します。

まだ換算されていない通貨を使用してレポートを定義する場合は、ルール・ファイルのサブ換算セクションに定義されている換算ルールを使用して、換算プロセスが実行されます。また、換算された金額が対応する通貨の値ディメンション・メンバーに保管されます。ただし、レポートの通貨が以前に換算済で、エンティティの換算ステータスが OK の場合、再換算は不要で、内部取引照合レポートの処理には保管されている換算済の金額が使用されます。

たとえば、EUR 通貨の内部取引照合レポートを実行すると、まず、換算済データが EUR 通貨(V#EUR)用に作成されているかどうかを確認されます。また、換算ステータスが OK であることも確認されます。エンティティの換算ステータスが TR (換算が必要)である場合は、換算済データが有効になるように再換算が行われます。それ以外の場合、処理には保管されている換算済の金額が使用されます。ただし、データが EUR に換算されていない場合は、サブ

換算に定義されているように換算プロセスが実行され、換算された金額が EUR の値メンバーに保管されます。EUR の換算済の金額も、内部取引照合レポートに使用されません。

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。

戻しランザクションの非表示

デフォルトで、エンティティの取引が表示されるたびに内部取引パートナーから戻し取引が表示されます。レポートの生成時に、これらの戻し取引を非表示にするように選択できます。

レポートに 1 つの照合勘定科目しかないときにこのオプションは非常に便利です。たとえば、収益と費用の内部取引の保管に使用する「現金」勘定科目がある場合です。この場合、レポートには 1 つの照合勘定科目しかないため、戻し取引の非表示オプションを使用して、重複エントリを避ける必要があります。

詳細の非表示

レポートに異なる金額のみを表示する場合は、内部取引の詳細を非表示にできます。このオプションを選択すると、レポートには内部取引は表示されず、エンティティ/パートナーの各セクションの合計差異のみが印刷されます。不一致が存在するために、各内部取引を表示する必要がある場合は、レポートを再生成して、内部取引の詳細を表示できます。

カスタム・ディメンションの非表示

任意のカスタム・ディメンションの列を非表示にするように選択できます。

「メンバー表示」オプション

ディメンション・メンバーのラベル、説明、または両方をレポートに表示できます。

「グループ化の基準」オプション

内部取引パートナーとの取引をカスタム・ディメンションで分類できます。このオプションに従って詳細がソートされ、グループの小計が表示されます。

内部取引照合レポート・スクリプトのキーワード

このセクションのキーワードを使用して、内部取引パートナ照合レポート・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。

ノート:

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.xml
```

レポート定義にスタイル・シートを指定しない場合、デフォルトのスタイル・シートが使用されます。

POV

このキーワードは、レポートの視点を指定します。スクリプトでは必須キーワードです。

構文

```
POV=S#Scenario.Y#Year.P#Period.V#Value.W#View
```

Scenario、*Year*、*Period*、*Value* および *View* を有効なディメンションのメンバーに置換します。例:

```
POV=S#Actual.Y#2014.P#January.V#USD.W#YTD
```



ノート:

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]}
```

パートナ

このキーワードは、レポートに表示するパートナまたはパートナのメンバー・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

レポート3は、1つの勘定科目を1つの勘定科目と照合しているレポートの例です。

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。

ノート:

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。

ノート:

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・ディメンションを非表示にします。

構文

```
SuppressCustomAlias=Boolean
```

Boolean を **Yes** または **No** に置換します。デフォルトは **Yes** です。例:

```
SuppressCustomFlows=Yes
```

ScaleFactor

このキーワードは、小数点の位置を識別して値を表示する単位を指定します。たとえば、位取り 3 を入力すると、レポートの値は、1000 単位で表示されます。位取りが 0 の場合、レポートの値は 1 単位で表示されます。

構文

```
ScaleFactor=Scale
```

Scale を次の番号のいずれかに置換します。

- 0 = ー
- 1 = 十
- 2 = 百
- 3 = 千
- 4 = 1 万
- 5 = 10 万
- 6 = 百万
- 7 = 1 千万

- 8 = 1 億
- 9 = 十億

例:

```
ScaleFactor=3
```

この例では、数字 12345.78 はレポートで 12.345678 と表示されます。

Decimal

このキーワードは、レポートに表示する小数点以下の桁数を指定し、勘定科目で定義した設定を上書きできます。

構文

```
Decimal=NumberDecimalPlaces
```

NumberDecimalPlaces を 0-6 の数値に置換します。小数点以下の桁数を指定しない場合は、勘定科目に定義されているデフォルトの小数点設定が使用されます。例:

```
Decimal=3
```

この例では、数字 123.4567 はレポートで 123.457 と表示されます。

DisplayLabels

このキーワードは、メンバーのラベルをレポートに表示するかどうかを指定します。

構文

```
DisplayLabels=Boolean
```

Boolean を Yes または No に置換します。デフォルトは Yes です。例:

```
DisplayLabels=Yes
```

DisplayDescriptions

このキーワードは、メンバーの説明をレポートに表示するかどうかを指定します。

構文

```
DisplayDescriptions=Boolean
```

Boolean を Yes または No に置換します。デフォルトは No です。例:

```
DisplayDescriptions=Yes
```

DisplayPlugElimAccts

このキーワードは、内部取引の影響を受けた調整勘定の要約を表示するかどうかを指定します。

構文

```
DisplayPlugElimAccts=Boolean
```

Boolean を Yes または No に置換します。例:

```
DisplayPlugElimAccts=Yes
```

GroupByCustom

このキーワードは、ディメンションごとにカスタム・ディメンション取引をグループ化します。このキーワードはオプションです。

構文

```
GroupByCustom=Custom
```

Custom をカスタム・ディメンションの別名に置換します。レポートのカスタム・ディメンションは、このディメンションで分類されます。例:

```
GroupByCustom=Flows
```

内部取引レポート・スクリプトの定義

次のタイプの内部取引レポートを定義できます。

- 内部取引 - 取引のリストを作成します。
- 勘定科目による内部取引照合 - 選択した勘定科目に基づく照合レポートを作成します。
- ID による内部取引照合 - 取引 ID に基づく照合レポートを作成します。

表 9-1 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

表 9-1 (続き) 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
ReportSecurityClass	レポートのセキュリティ・クラスを指定します。デフォルトは[Default]です。
POV	レポートの有効な視点を指定します。例: POV=S#ActMon.Y#2014.P#January
ScaleFactor	レポートの位取りを指定します。スケールは、0 から 9 までの値です。
Decimal	レポートに表示する小数点以下の桁数を指定します。小数点以下の桁数は、0 から 9 までの値です。
FromAmt および ToAmt	取引金額の範囲を指定します。
IncludeMatched	レポートに照合済の取引を含める場合は TRUE を、含めない場合は FALSE を指定します。
IncludeUnMatched	レポートに未照合の取引を含める場合は TRUE を、含めない場合は FALSE を指定します。
IncludeMisMatched	レポートに不一致の取引を含める場合は TRUE を、含めない場合は FALSE を指定します。
IncludePosted	レポートに転記済の取引を含める場合は TRUE を、含めない場合は FALSE を指定します。
IncludeUnPosted	レポートに転記の戻し済の取引を含める場合は TRUE を、含めない場合は FALSE を指定します。
エンティティ	エンティティは有効な ICP の親エンティティではなく、基本エンティティである必要があります。
パートナ	パートナ・エンティティは、勘定科目の有効な ICP エンティティである必要があります。
エンティティ 勘定	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。
パートナ勘定	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。

表 9-1 (続き) 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
ReferenceID	<p>これはオプションです。参照 ID を入力して取引の参照情報を保管できます。たとえば、エンティティは、請求書を発行したエンティティとは異なる固有の請求書一連番号を持つ場合があります。情報目的のみのために、この参照 ID に追加情報を入力できます。</p> <p>エンティティの参照 ID を「取引 ID」に入力し、パートナー・エンティティから受け取った請求書の番号を入力できます。</p>
MatchCode	<p>これはオプションです。照合コードは、照合プロセスのタイプを区別するための接頭辞です。</p> <ul style="list-style-type: none"> • A - 勘定科目を使用する自動照合 • I - 取引 ID を使用する自動照合 • R - リファレンス ID を使用する自動照合 • M - 手動照合
ReasonCode	<p>これはオプションです。理由コードは、管理者が定義した有効な理由コードであることが必要です。理由コードの主な目的は、取引が「不一致」ステータスを持つようになった理由(パートナー・エンティティからの請求書を紛失した、パートナー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	<p>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。</p>
ToDate	<p>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。</p>
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、<code><DisplayedColumns>,<Sort>,<Label>,<Repeat>,<Totals></code>です。ここで、<code><Displayed Columns></code>は、有効な列、<code><Sort></code>は <code>Sort</code> または <code>No Sort</code>、<code><Label></code>は、ラベル、説明、または両方、<code><Repeat></code>は <code>Repeat</code> または <code>NoRepeat</code>、<code><Total></code>は <code>Total</code> または <code>NoTotal</code> です。</p> <p>カスタム・ディメンションの場合、列に対して次のような別名が使用されます： <code>Customers</code>、<code>NoSort</code>、<code>Label</code>、<code>Repeat</code>、<code>NoTotal</code></p>

表 9-1 (続き) 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
DisplayEntityTransactions	<p>レポートにエンティティ取引を含める場合は TRUE を、含めない場合は FALSE を指定します。</p> <p>特定のエンティティおよびパートナーの内部取引のみを表示するように選択できます。また、そのエンティティと対応するパートナーの取引を表示するように選択することもできます。たとえば、エンティティとパートナーの選択肢からエンティティ A とパートナー B の取引のみを選択すると、エンティティ A がパートナー B と持つ取引のみが表示されます。ただし、エンティティ B とパートナー A の該当する取引を参照する場合は、エンティティ取引とパートナー取引を含むように選択できます。</p>
DisplayPartnerTransactions	<p>レポートにパートナーの取引を含める場合は TRUE を、含めない場合は FALSE を指定します。</p>
SuppressDetails	<p>取引の詳細を非表示にして、小計行のみを表示する場合は TRUE を指定します。</p>

10

ルール管理

Oracle Hyperion 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。

ノート:

Oracle Hyperion Calculation Manager でルールを操作するには、[Calculation Manager](#) を使用したルールの作成を参照してください。

ルールのタイプ

値の設定やクリア、データの計算、通貨の換算、データの連結、別の場所へのデータの配賦を実行するルールや、データの入力を防ぐためのルールを作成できます。

表 10-1 ルールのタイプ

ルールのタイプ	説明
Calculation	ユーザーが計算を実行するとき、計算ルールが実行されます。計算ルールを使用して、ディメンションの階層の自然な順序では計算できない計算を実行できます。たとえば、比率や期首残高を導出するための計算を作成できます。Calculate()ルールは、データの計算時や連結時に実行されます。

表 10-1 (続き) ルールのタイプ

ルールのタイプ	説明
Translation	<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>Oracle Hyperion 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。
Consolidation	<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>
Allocation	<p>配賦ルールは、あるエンティティのデータをエンティティのリストに配賦します。たとえば、配賦ルールを使用して、他のエンティティの総管理支出に基づいたエンティティのリストの管理支出を計算できます。</p>
Input	<p>入力ルールは、親エンティティ・レベルでの入力を可能にします。エンティティ通貨の値ディメンションのみがサポートされます。子から親エンティティのエンティティ通貨値ディメンションへのコントリビューション値はロールアップされません。Parent.Child の組合せのコントリビューション値は保管されます。</p>

表 10-1 (続き) ルールのタイプ

ルールのタイプ	説明
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>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、現在の視点のメンバーに対するテストをサポートしません。たとえば、If HS.Entity.Member = "CT" をテストできません。 • NoInput 関数をループで使用する場合は注意する必要があります。単純な文でもループで使用すると、何千ものセルがメモリーにロードされる可能性があるため、NoInput を含むループではパフォーマンスへの影響を確認する必要があります。
Dynamic Calculation	<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 文です。 • Sub Dynamic セクションの文はすべて、シーケンシャルに実行されます。

表 10-1 (続き) ルールのタイプ

ルールのタイプ	説明
取引	取引ルールは、内部取引をサポートするアプリケーション内の勘定科目を指定します。取引をサポートするセルは、データ・グリッドとデータ・フォームで読取り専用です。
エクイティ・ピックアップ	エクイティ・ピックアップ・ルールは、所有エンティティ、所有者エンティティおよび出資比率を指定します。 これは、 Sub EquityPickup セクションの実行時のデフォルトの視点です： <ul style="list-style-type: none"> 現在のシナリオ、年および期間 エンティティ: 処理されるペアの所有者 値: <Entity Currency>
OnDemand	オンデマンド・ルールはデータ・フォームで使用され、計算のサブセットを実行して、結果をデータ・フォームで迅速に確認できます。 Sub Calculate (他にはなし)で使用できるすべての HS 関数は、オンデマンド・ルールでも使用できます。

Microsoft VBScript 言語に基づいた一意のスクリプトで **Financial Management** のルールを作成します。ルールは、スクリプト構文を生成するための関数、オブジェクトおよびその他の引数の組合せを使用して構築されます。各ルーチン内で、2 つのタイプの関数を使用してルールを記述します。

- 計算を定義する **Financial Management** の関数
- 条件付き文を作成するために使用される **VBScript** 関数

ルールに関する注意事項

次は、アプリケーションのルールを記述する際の注意事項です。

- ルールは、ユーザーが計算、換算、連結および配賦を行うときに実行されます。計算ルールは、計算または連結の適用先であるエンティティ/値ディメンションの交差ごとに一度実行されます。[計算コマンドを含む計算ルール](#)を参照してください。
- **Oracle Hyperion Financial Management** が計算を適用するディメンション・メンバーは、ユーザーのカーソルが置かれたデータ・グリッド・セルと視点バーで指定されたメンバーによって決まります。[現在のディメンション・メンバー](#)を参照してください。
- 関数が通貨に関連する値メンバーにデータを挿入すると、**Financial Management** が、値メンバー内の現在の値を削除する場合があります。[関数によるデータの自動クリア](#)を参照してください。
- ルールは、ルーチン内の連続した順序で実行され、ルーチンの実行順序は設定されています。[連結時のルールの実行](#)を参照してください。

計算コマンド

計算ルーチンは、シナリオ、年、期間、エンティティおよび値の指定された交差に対して実行されます。計算ルーチンを実行すると、シナリオ、年、期間、エンティティおよび値の現在の交差に対応する現在のデータ表にデータが書き込まれるかクリアされます。ユーザーが計算ルーチンを実行すると、アプリケーション内の任意の場所からデータが読み取られます。ただし、データは現在のデータ表のみに書き込まれます。

ノート:

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、Oracle Hyperion Financial Management により、RLE ファイルで **Sub Calculate()** ルーチンが実行されます。**Calculate()** ルーチンは、指定されたシナリオ、年および期間内の指定されたエンティティおよび値の組合せに対して、勘定科目とカスタム・ディメンションのメンバーを計算します。

計算の強制

計算の強制オプションは、選択された値メンバーおよびその値メンバーが依存するすべての値メンバーのみに強制的にルールを適用します。たとえば、エンティティ通貨に対する計算の強制はエンティティ通貨メンバーのみにルールを適用します。エンティティ通貨の合計に対する計算の強制は、エンティティ通貨、エンティティ通貨の調整およびエンティティ通貨の合計にルールを適用します。計算の強制は、値の 3 つ組からなる単一メンバーのみに影響を与えることが可能な唯一のコマンドです。

計算コマンドを含む計算ルール

計算ルールが実行される回数は、ユーザーおよびその他の要素によって選択される計算コマンドによって異なります。

▲ **注意:**

ルールを記述する前にこれらのセクションを注意してお読みください。値ディメンションの特定のメンバーにのみ起こる操作が目的である場合は、操作を実行する前に **VBScript** の **If** 構造と **Member** 関数を使用して現在のメンバーに対してテストを行う必要があります。**If** 構造と **Member** 関数の使用に関するステップおよび例については、[条件付きのルール](#)を参照してください。

ユーザーが「計算」または「計算の強制」コマンドを選択すると、**Oracle Hyperion Financial Management** では、エンティティのデフォルト通貨(**Entity Currency**)の値メンバーと現在のエンティティ・メンバーの交差に対してアプリケーションの計算ルールが実行されます。エンティティの **AllowAdjs** 属性がメタデータで使用可能である場合、**Financial Management** では、2 度目のルールが実行され、エンティティのデフォルト通貨の調整(**Entity Curr Adjs**)を保管する値ディメンションのメンバーとエンティティの交差に対してルールが適用されます。

例

たとえば、**California** という名前のエンティティのデフォルト通貨が **USD** であり、その **AllowAdjs** 属性がメタデータで使用可能になっている場合、1 度目は **California** と **USD** の交差に対して、2 度目は **California** と **USD Adjs** の交差に対してというように、計算ルールは 2 度実行されます。

連結コマンドを使用した計算ルール

ユーザーが連結コマンドのいずれかを選択すると、**Oracle Hyperion Financial Management** では、まだ連結されていない各子エンティティを交差するいくつかの値ディメンションに対して計算ルールが実行されます。子に対して実行したら、**Financial Management** では、親のデフォルト通貨の値ディメンションのメンバーと親エンティティの交差に対してルールが実行されます。

1. ルールは、子エンティティのデフォルト通貨(**Entity Currency**)を保管する値メンバーと子エンティティの交差に対して実行されます。
2. 子エンティティの **AllowAdjs** 属性が **Y** に設定されている場合、ルールは、子エンティティのデフォルト通貨の調整(**Entity Curr Adjs**)を保管する値メンバーと子エンティティの交差に対して実行されます。
3. 子と親で通貨が異なる場合、ルールは、親の母国通貨(**Parent Currency**)に換算された値を保管する値メンバーと子エンティティの交差に対して実行されます。
4. 子と親で通貨が異なり、ルール・ファイルが換算ルールを含む場合、**Financial Management** は、ステップ 3 の前に換算ルールを実行します。
5. 子と親で通貨が異なり、子エンティティの **AllowAdjs** 属性が **Y** に設定されている場合、ルールは親の母国通貨に換算された調整(**Parent Curr Adjs**)を保管する値メンバーと子エンティティの交差に対して実行されます。
6. 親エンティティの **AllowAdjFromChildren** 属性が **Y** に設定されている場合、ロジック・ルールは、子エンティティと **Parent Adjs** 値の交差に対して実行されます。
7. ルールは、子エンティティと **Proportion** 値の交差に対して実行されます。
8. ルールは、子エンティティと **Elimination** 値の交差に対して実行されます。

9. 親エンティティの **AllowAdjFromChildren** 属性が Y に設定されている場合、ルールは、子エンティティと **Contribution Adjs** 値の交差に対して実行されます。
10. まだ連結されていないデータを含むその他の各子エンティティに対して、ステップ 1-9 を繰り返します。
11. ルールは、親エンティティのデフォルト通貨を保管する値メンバーと親エンティティの交差に対して実行されます。

例

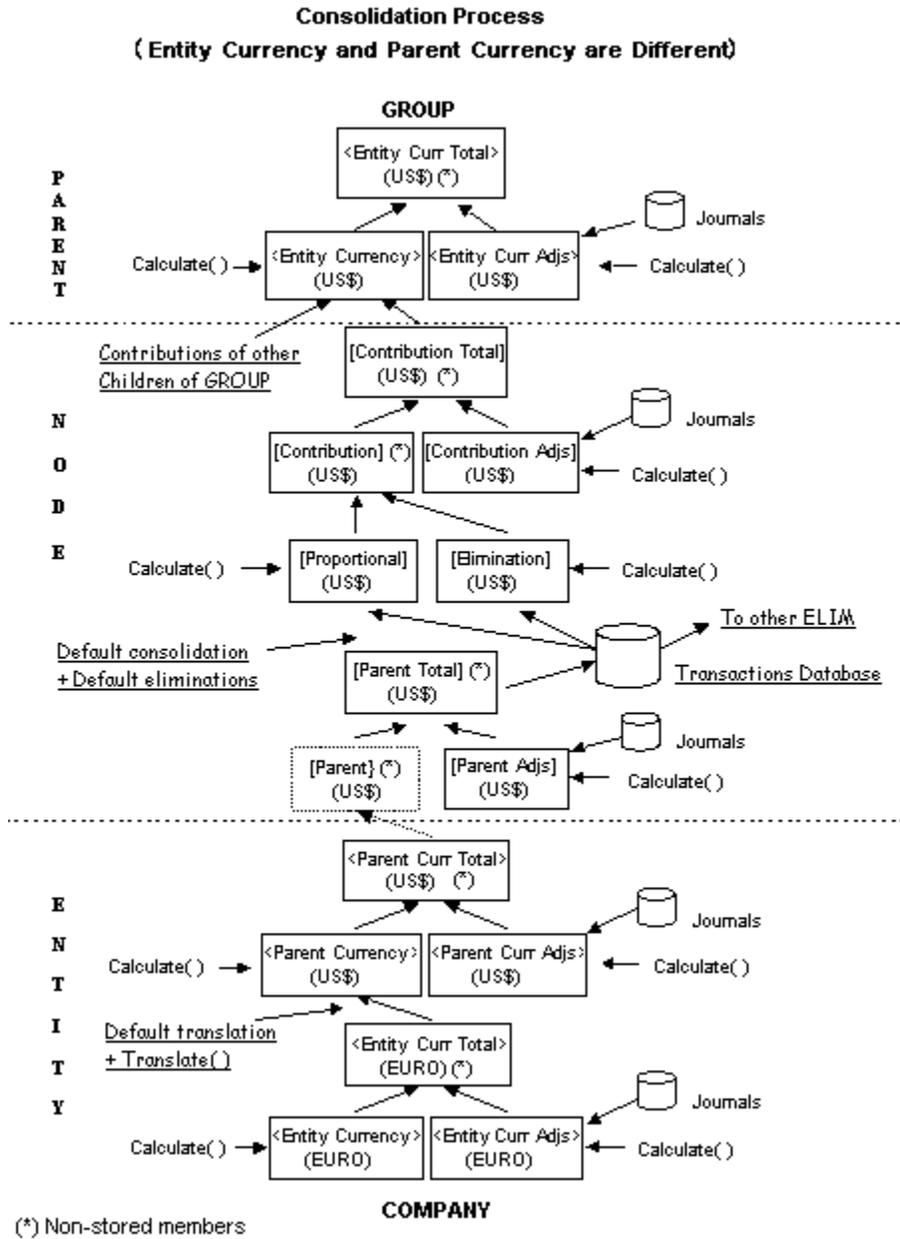
たとえば、**UnitedStates** という名前の親エンティティに、**EastUS** という名前と **WestUS** という名前の子がいるとします。子は **AllowAdjs** 属性を使用可能にしています。**UnitedStates** エンティティは、**AllowAdjs** 属性と **AllowAdjFromChildren** 属性を使用可能にしています。3 つのすべてのエンティティは、デフォルト通貨 **USD** を共有します。

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・ディメンションと値ディメンションにおける次の各交差に対してルールが実行されます。

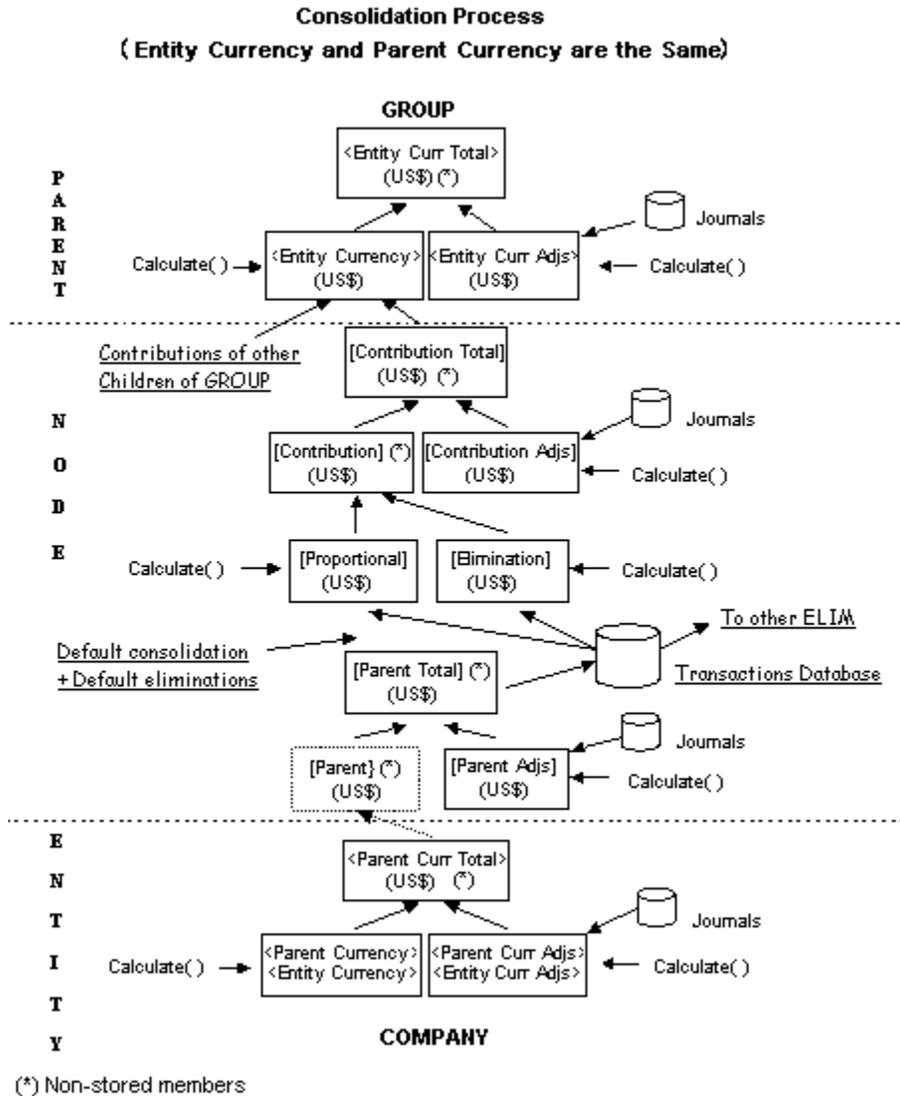
1. **EastUS** および **Entity Currency**
2. **EastUS** および **Entity Currency Adjs**(**EastUS** の **AllowAdjs** 属性は Y に設定されています)。
3. **EastUS** および **Parent**。
4. **EastUS** および **Parent Adjs**(**UnitedStates** の **AllowAdjFromChildren** 属性は Y に設定されています)。
5. **EastUS** および **Proportion**。
6. **EastUS** および **Elimination**。
7. **EastUS** および **Contribution Adjs**(**UnitedStates** の **AllowAdjFromChildren** 属性は Y に設定されています)。
8. **WestUS** および **Entity Currency**
9. **WestUS** および **Entity Currency Adjs**(**WestUS** の **AllowAdjs** 属性は Y に設定されています)。
10. **WestUS** および **Parent**。
11. **WestUS** および **Parent Adjs**(**UnitedStates** の **AllowAdjFromChildren** 属性は Y に設定されています)。
12. **WestUS** および **Proportion**。
13. **WestUS** および **Elimination**。
14. **WestUS** および **Contribution Adjs**(**UnitedStates** の **AllowAdjFromChildren** 属性は Y に設定されています)。
15. **UnitedStates** および **Entity Currency**

次に連結プロセスの例を示します。

1 つ目の例は、エンティティの通貨および親通貨が異なる場合のプロセスを示しています。



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。



現在のディメンション・メンバー

デフォルトで、Oracle Hyperion Financial Management は、ユーザーが計算コマンドを選択するときに、現在のディメンション・メンバーに計算ルールを適用します。[計算コマンドを含む計算ルール](#)を参照してください。現在のディメンション・メンバーは次の要因によって決定されます。

- データ・グリッドで、ユーザーがデータを計算、換算または連結するときにカーソルを置いているセル。
- 視点バーで指定したディメンションのメンバー。
- 連結の場合、エンティティ・ディメンションと値ディメンションの現在のメンバーは、ルールを実行するたびに変更されます。[連結コマンドを使用した計算ルール](#)を参照してください。
- ルールは、現在選択されているディメンションのメンバーに対して適用されます。ただし、計算ルールは例外で、アプリケーションのすべての勘定科目において適用されます。

カーソルが置かれたセルのディメンション・メンバーが視点バー内の対応するディメンションの対応するメンバーと異なる場合、セルのディメンション・メンバーは、視点バーのディメンション・メンバーを上書きします。たとえば、ユーザーが視点バー内で **Europe** という名前のエンティティを指定し、カーソルを **Germany** という名前のエンティティのセル内に配置した状態で換算を実行すると、**Germany** が現在のエンティティになります。

ディメンション・メンバーを引数に指定すると、現在のディメンションとは関係なく、いくつかの関数を使用して、特定のディメンション・メンバーを操作できます。たとえば、**Entity** オブジェクトを指定して **NumBase** 関数を使用すると、引数で指定するエンティティの基本エンティティの数を取得できます。

さらに複雑な関数の場合は、勘定科目式を作成して、ディメンションのメンバーを指定できます。

勘定科目式

一部の関数は、勘定科目式を引数として必要です。最も簡単なフォームの勘定科目式は、**Oracle Hyperion 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 10-2 勘定科目式でディメンションを指定するために使用される文字

文字	ディメンション
A#	勘定科目
I#	内部取引パートナ
C#	カスタム
S#	シナリオ
Y#	年
P#	期間
W#	ビュー
E#	エンティティ
V#	値

勘定科目の式を理解するために、関数の引数に指定されたディメンション・メンバーの値を除去する **Clear** 関数について検討します。この例では、引数「A#Sales」は勘定科目式です:

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

A#文字は、勘定科目ディメンションを表し、単語 **Sales** は、**Financial Management** によって実行される **Clear** 関数の適用先である勘定科目ディメンションのメンバー名です。この勘定科目式は、**Financial Management** に、**Sales** 勘定科目に保管されたデータをクリアするよう指示します。

勘定科目の式を使用するときは、勘定科目の式で指定した勘定科目と、次に示すディメンションの現在のメンバーの交差に、**Financial Management** によって関数が適用されます。

- エンティティ
- 期間
- シナリオ
- 値
- ビュー
- 年
- カスタム - メタデータの勘定科目に設定された **CustomTopMember** を使用します。たとえば、勘定科目の式がカスタム 3 ディメンションのメンバーを指定していない場合、**Financial Management** では、勘定科目に指定された **CustomTopMember** によって定義されたとおりに、すべての有効なカスタム 3 ディメンションのメンバーが使用されます。
- 内部取引パートナ - すべての有効な **ICP** メンバー。

内部取引とカスタム・ディメンションのデフォルトを上書きするには、勘定科目の式にメンバーを指定します。各ディメンションは、特定の文字で表されます。2 つ以上のディメンションを勘定科目の式に含める場合は、ディメンションをピリオドで区切る必要があります。

勘定科目の式を作成する場合、これらのディメンション・メンバーのすべてを指定する必要はありません。関数を適用するメンバーを指定します。たとえば、この行は、**Sales** 勘定科目と **Hardware ICP** の交差をクリアします。

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

関数によるデータの自動クリア

関数によって通貨に関連する値メンバーにデータが挿入されると、**Oracle Hyperion Financial Management** では、次のいずれかの条件に該当する場合、データがメンバーから自動的にクリアされます。

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

ヒント:

通貨関連の値メンバーは、システム生成の **Entity Currency** メンバーと、**USD**、**EURO**、**LIRA** などのユーザー定義通貨メンバーです。

また、関数により調整用データが値メンバーに挿入されると、**Financial Management** では、交差する勘定科目メンバーが計算済勘定科目である場合に、このメンバーからのデータが自動的にクリアされます。

ヒント:

調整に関連する値メンバーは、名前にコード「**Adjs**」が含まれるメンバーです。たとえば、**USD Adjs**、**EURO Adjs**、**LIRA Adjs** などです。

エラー・メッセージ

Oracle Hyperion Financial Management が構文エラーを検出すると、次の情報を含むエラー・メッセージを表示します。

- エラーの原因となる RLE ファイル内の行番号
- 該当する Financial Management のオブジェクトおよび関数
- エラーの説明。

連結時のルールの実行

連結プロセスでは、ルールは定義済シーケンスで実行されます。特定の親の基本の子それぞれに対して、値ディメンション内の様々な要素の計算シーケンスが次の順序で適用されます：

1. メタデータで **IsCalculated** として定義された勘定科目が **Entity Currency** でクリアされます。
2. メタデータで **IsCalculated** として定義された勘定科目が **EntityCurrAdjs** でクリアされます。
3. **Sub Calculate()**ルーチンが **Entity Currency** で実行されます。
4. **Sub Calculate()**ルーチンが **EntityCurrAdjs** で実行されます。
5. **ParentCurrency** データがクリアされます。
6. **EntityCurrency** および **EntityCurrAdjs** の合計値に対して、デフォルトの換算が、収益、費用、資産、負債として定義されたすべての勘定科目に適用されます。フロー属性または残高属性を持つ勘定科目の場合、デフォルトでは取引が適用されず、**EntityCurrency** および **EntityCurrAdjs** の合計値は、親通貨にロール・アップされません。
7. **Sub Translate()**ルーチンが実行されます。
8. **Sub Calculate()**ルーチンが **ParentCurrency** で実行されます。
9. メタデータで **IsCalculated** として定義された勘定科目が **ParentCurrAdjs** でクリアされます。
10. **Sub Calculate()**ルーチンが **ParentCurrAdjs** で実行されます。
11. メタデータで **IsCalculated** として定義された勘定科目が **ParentAdjs** でクリアされます。
12. **Sub Calculate()**ルーチンが **ParentAdjs** で実行されます。
13. **Proportion** および **Elimination** のデータがクリアされます。
14. **Parent** および **ParentAdjs** の合計値に対して、デフォルトの連結と消去が実行されます。
15. **Sub Calculate()**ルーチンが **Proportion** および **Elimination** で実行されます。
16. メタデータで **IsCalculated** として定義された勘定科目が **ContributionAdjs** でクリアされます。
17. **Sub Calculate()**ルーチンが **ContributionAdjs** で実行されます。

上記のステップが各基本子に対して繰り返された後、次のシーケンスが親エンティティに適用されます。

1. Entity Currency データがクリアされます。
2. 各子の Proportion、Elimination および ContributionAdjs の合計が親エンティティの EntityCurrency に書き込まれます。
3. Sub Calculate()ルーチンが Entity Currency で実行されます。
4. メタデータで IsCalculated として定義された勘定科目が EntityCurrAdjs でクリアされます。
5. Sub Calculate()ルーチンが EntityCurrAdjs で実行されます。

ノート:

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ステップ 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。

ノート:

最初に現在のエンティティ内の換算レートが検索されます。現在のエンティティで見つからなければ、[None]エンティティ内の換算レートが検索されます。[None]エンティティで見つからなければ、三角換算法を使用して換算されます。三角換算法は、3 番目の共通通貨を使用して、残高をある通貨から別の通貨に変換する方法です。

たとえば、Euro から Yen への変換で、その換算を実行するための直接レートまたは間接レートが見つからない場合や、Euro と Yen の両方を USD に換算できる場合は、三角換算法を使用することで Euro の残高を USD に変換してから、USD の残高を Yen に変換します。

Financial Management のオブジェクト

この項では、関数の使用時に **Oracle Hyperion 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

ルールでの VBScript の使用

VBScript 関数は、ルールの条件文を記述するために使用されます。

ノート:

Oracle Hyperion Financial Management では、COM スクリプト対応インターフェースとして公開されている .NET オブジェクトをサポートしていません。VBScript などの COM 対応アプリケーションでは、ルールまたはメンバー・リストを介してそれらにアクセスすることができません。

VBScript 演算子

次の VBScript 演算子が、Oracle Hyperion Financial Management ルールでサポートされています。

- And
- Eqv
- Imp
- Is
- Mod
- Not
- Or
- Xor
- =
- <
- >
- <=
- >=
- <>
- +
- &
- -
- /
- \
- *
- ^

VBScript 文

次の VBScript 文が、Oracle Hyperion Financial Management ルールでサポートされています。

- Call
- Const
- Dim
- Do...Loop
- Erase
- Exit
- For...Next
- ForEach...Next
- Function...End Function
- If...Then...Else If...Else...End If
- On Error Go to 0
- On Error Resume Next
- Option Explicit
- Private
- Public
- ReDim
- ReDim Preserve
- SelectCase...Case...End Select
- Set
- While...WEnd
- With...End With
- ' (comment)
- Sub...EndSub

VBScript のキーワード

次の VBScript キーワードが、Oracle Hyperion Financial Management ルールでサポートされています。

- Empty
- Nothing
- Null
- TRUE
- FALSE

VBScript 関数

次の VBScript 関数が、Oracle Hyperion Financial Management ルールでサポートされています。

日付と時間の関数

関数	説明
Cdate	有効な日付と時間の式を、サブタイプ Date の変数に変換します
Date	現在のシステム日付を戻します。
DateDiff	2 つの日付間の間隔数を戻します。
DatePart	DatePart(interval,date[,firstdayofweek[,firstweekofyear]])
DateSerial	DateSerial(year,month,day)
Day	月の日(1 から 31 まで)を表す数字を戻します。
Hour	日の時間(0 から 23 まで)を表す数字を戻します
IsDate	評価済の式を日付に変換できるかどうかを示すブール値を戻します
Minute	時間の分(0 から 59 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。
Month/Name	指定された月の名前を戻します。
Now	現在のシステムの日付と時間を戻します
Second	分の秒(0 から 59 まで)を表す数字を戻します
Time	現在のシステム時刻を戻します
Timer	12:00 AM 以降の秒数を戻します
年	年を表す数値を戻します

フォーマット関数

関数	説明
FormatNumber	数値としてフォーマットされた式を戻します
FormatFunctions	日付または時間としてフォーマットされた式を戻します

変換関数

関数	説明
Asc	文字列の最初の文字を ANSI コードに戻します
CBool	式をサブタイプ Boolean の変数に変換します
CByte	式をサブタイプ Byte の変数に変換します
CDbl	式をサブタイプ Double の変数に変換します
Chr	指定した ANSI コードを文字に変換します
Cint	式をサブタイプ Integer の変数に変換します
CLng	式をサブタイプ Long の変数に変換します

関数	説明
CStr	式をサブタイプ String の変数に変換します

算術関数

関数	説明
Abs	指定された数の絶対値を返します。
Fix	指定された数の整数部を返します。
Int	指定された数の整数部を返します。
Rnd	1 未満で 0 以上の乱数を返します
Sgn	指定された数の記号を示す整数を返します
Sqr	指定された数値の平方根を返します

配列関数

関数	説明
Array	配列を含む variant を返します。
Filter	フィルタ条件に基づいた文字列配列のサブセットを含むゼロから始まる配列を返します。
IsArray	指定された変数が配列であるかどうかを示すブール値を返します
Join	配列内の多くの部分文字列で構成される文字列を返します。
Lbound	指定された配列の次元の最も小さいサブスクリプトを返します。
Split	指定された数の部分文字列を含む、ゼロから始まる一次元配列を返します。
Ubound	指定された配列の次元の最も大きいサブスクリプトを返します。

文字列関数

関数	説明
InStr	別の文字列内で、ある文字列が最初に出現する位置を返します。検索は文字列の最初の文字から始めます。
InStrRev	別の文字列内で、ある文字列が最初に出現する位置を返します。検索は文字列の最後の文字から始めます。
Left	文字列の左側から指定された数の文字を返します。
Len	文字列内の文字数を返します。
LTrim	文字列の左側にあるスペースを除去します
RTrim	文字列の右側にあるスペースを除去します
Trim	文字列の左と右の両側にあるスペースを除去します

関数	説明
Mid 置換	文字列から指定された数の文字を戻します。 指定された文字列の一部を別の文字列で指定の回数置き換えます
Right	文字列の右側から指定された数の文字を戻します。
Space	指定された数のスペースで構成される文字列を返します
StrComp	2 つの文字列を比較し、比較の結果を表す値を戻します。
StrReverse	文字列を逆にします
LCase	指定された文字列を小文字に変換します。
UCase	指定された文字列を大文字に変換します。

その他の関数

関数	説明
CreateObject	指定されたタイプのオブジェクトを作成します
Eval	式を評価し、結果を戻します
IsEmpty	指定された変数が初期化されているかどうかを示すブール値を戻します
IsNull	指定された式に有効なデータが含まれていない (Null) かどうかを示すブール値を戻します
IsNumeric	指定された式を数値として評価できるかどうかを示すブール値を戻します
Round	数値を丸めます

VBScript オブジェクト

次の VBScript オブジェクトが、Oracle Hyperion Financial Management ルールでサポートされています。

ファイル・システム・オブジェクト

- CreateTextFile
- OpenTextFile
- DeleteFile
- FileExist
- FolderExist
- GetBaseName
- GetParentFolderName
- GetFile
- CopyFile

ファイル・オブジェクト

- OpenAsTextStream
- Size

テキスト・ストリーム・オブジェクト

- AtEndOfStream
- Close
- WriteLine
- ReadLine

Err オブジェクト

- Description
- HelpContext
- HelpFile
- Number
- Source
- Raise
- Clear

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。[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。

 ノート:

EXP は、現在のディメンション・メンバーを含む勘定科目の交差にデータを挿入します(現在のディメンション・メンバーを参照)。さらに、勘定科目式の文字を使用して、現在のカスタム・メンバーと ICP メンバーを上書きできます。

この例では、計算ルールによって、PrevCash 勘定科目が Cash 勘定科目の値に設定されます。

```
HS.EXP "A#PrevCash = A#Cash"
```

勘定科目式の文字を使用して、EXP の引数で、等号の両側にディメンションのメンバーを指定できます。Exp を参照してください。この例は、PrevCash 勘定科目とカスタム 3 ディメンションの Golf メンバーの前年度の交差のデータを、PrevCash と Golf の当年度の交差に挿入します。

```
HS.EXP "A#PrevCash.C3#Golf = A#Cash.Y#Prior.C3#Golf"
```

 ヒント:

勘定科目式の文字 Y#の後に続く Prior キーワードによって、EXP は、前年度のデータを取得できます。勘定科目の式には、年や期間に適用される同様のキーワードが複数あります。期間および年のキーワードを参照してください。

値の計算による勘定科目の設定

もう 1 つの共通タスクは、2 つの勘定科目に含まれる値を計算し、結果を別の勘定科目に挿入することです。EXP 関数では、その引数で、等号の右側に加算、減算、乗算および除算を置くことができます。

この計算ルールの例では、Sales 勘定科目の値を UnitsSold 勘定科目の値で割り、その結果を AvgPrice 勘定科目に挿入します。

```
HS.EXP "A#AvgPrice = A#Sales / A#UnitsSold"
```

条件付きのルール

特定のディメンションのメンバーが視点の現在のメンバーである場合にのみアクションを実行するルールにします。たとえば、Actual(実績)が現在のシナリオである場合にはある方法で勘定科目の値を計算し、Budget(予算)が現在のシナリオである場合には別の方法で勘定科目の値を計算します。

 ヒント:

Oracle Hyperion Financial Management によって現在のディメンション・メンバーをどのように判別するかについては、[現在のディメンション・メンバー](#)を参照してください。

これを行うには、VBScript の If 構造で Member 関数を使用します。Member 関数は、次のディメンションの現在のメンバーの名前を取得します。

- エンティティ (現在のエンティティを取得するには Entity オブジェクトを、現在のエンティティの親を取得するには Parent オブジェクトをそれぞれ使用)
- 期間
- シナリオ
- 値
- 年

If 構造を使用すると、特定の条件が TRUE である場合にのみ文を実行できます。次のセクションでは、Member を If 構造で使用してディメンションのメンバーをテストする別の方法をいくつか示します。

 ヒント:

これらのセクションでは、If 構造の側面のみをいくつか取り上げています。If 構造の詳細は、Microsoft の VBScript のドキュメントを参照してください。VBScript のドキュメントは、Microsoft の Web サイトからダウンロードできます。

ディメンションのメンバーのテスト

特定のディメンションのメンバーが現在のメンバーである場合にのみ Oracle Hyperion Financial Management でアクションを行うには、Member 関数の戻り値をテストする If 構造を使用します。

この例では、現在のシナリオが Budget(予算)である場合に、Financial Management によって UnitsSold と Price の勘定科目の値が乗算され、結果が Sales 勘定科目に挿入されます。

```
If HS.Scenario.Member = "Budget" Then

    HS.EXP "A#Sales = A#UnitsSold * A#Price"

End If
```

 ヒント:

すべての If 構造は、If...Then 文で始まり、End If 文で終わる必要があります。条件を満たすときに実行されるアクションは、前述のように、If...Then 文と End If 文の間にあります。

複数のメンバーのテスト

If...Then 文では、複数のメンバーをテストできます。つまり、1つのディメンションで2つ以上のメンバーのアクションを実行できます。「ディメンションのメンバーのテスト」セクションにある例について考えます。現在のシナリオが **Budget(予算)** または **Forecast(予測)** の場合は、**Sales** 勘定科目の値を計算します。

複数のメンバーをテストするには、If...Then 文で2つの Member 関数と VBScript の Or キーワードを使用します。Or を最初の Member 関数の後に置き、Or と Then の間に2番目の Member を置きます。

 ヒント:

この技法を使用して、複数のメンバーをテストできます。テストするメンバーごとに、Member 関数と Or キーワードの組合せを含めます。

この例では、現在のシナリオが **Budget (予算)** または **Forecast (予測)** である場合、**Oracle Hyperion Financial Management** によって **UnitsSold** と **Price** の勘定科目の値が乗算され、結果が **Sales** 勘定科目に挿入されます。

```
If HS.Scenario.Member = "Budget" Or HS.Scenario.Member = "Forecast" Then  
  
    HS.EXP "A#Sales = A#UnitsSold * A#Price"  
  
End If
```

メンバーごとに異なるアクションを実行

ディメンションのメンバーごとに異なるアクションを実行するルールを持つことができます。たとえば、現在のシナリオが **Budget(予算)** のときはある計算をし、**Actual(実績)** のときは別の計算をするルールを持つことができます。

異なるアクションを条件に従って実行するには、If 構造に1つ以上の **Elseif** 文を含む必要があります。メンバーごとにその **Elseif** 文のテストを行います。**Elseif** 文の直下にそのメンバーのために実行するアクションを配置します。

この例では、現在のシナリオが **Budget(予算)** または **Actual(実績)** のどちらであるかに応じて、異なる勘定科目が更新されます。

- 現在のシナリオが **Budget (予算)** である場合、**Oracle Hyperion Financial Management** によって **UnitsSold** と **Price** の勘定科目の値が乗算され、結果が **Sales** 勘定科目に挿入されます。
- 現在のシナリオが **Actual(実績)** の場合は、**Financial Management** によって **Sales** 勘定科目の値が **UnitsSold** 勘定科目の値で除算され、その結果が **Price** 勘定科目に挿入されます。

```
If HS.Scenario.Member = "Budget" Then

    HS.EXP "A#Sales = A#UnitsSold * A#Price"

ElseIf HS.Scenario.Member = "Actual" Then

    HS.EXP "A#Price = A#Sales / A#UnitsSold"

End If
```

ヒント:

If...Then と ElseIf の文で指定した条件が一切満たされない場合に別のアクションを行うには、**VBScript** を使用して **Else** 文を **If** 構造に追加します。詳細は、**Microsoft** の **VBScript** のドキュメントを参照してください。

すべての勘定科目の期首残高の設定

勘定科目の期首残高を設定するには、**If** 構造で **Exp** および **IsFirst** 関数を使用します。**Exp** および **If** 構造は前のセクションで紹介されています。**IsFirst** は、現在の期間が現在のシナリオのデフォルト期間単位で最初の期間であるかどうかを判別します。たとえば、シナリオに月次のデフォルト期間単位がある場合、**IsFirst** は、現在の期間が年の最初の月であるかどうかを判別します。

期首残高を設定するには、**If** 構造の **If...Then** 文に **IsFirst** を挿入し、この文と **End If** 文の間に **Exp** を挿入します。特定の勘定科目を **Exp** の引数に含めることができるかぎり、すべての勘定科目の期首残高を設定できます。

この例では、すべての勘定科目の期首残高の設定方法を示します。この例を再入力するか、変更せずに計算ルールにコピーできます。

```
HS.EXP "A#ALL = A#ALL.P#Prior"
```

 ノート:

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。

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 Oracle Hyperion 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。

ルール・ファイルには、Financial Management のすべてのタイプのルールを任意の順序で含めることができます。表 1 に、各ルーチンを定義する基本構文がリストされています。

表 10-3 ルール・ルーチン

ルール・ルーチン	構文
Sub Calculate	Sub Calculate() 'ここに、計算ルールを入力します。 End Sub
Sub Translate	Sub Translate() 'ここに、換算ルールを入力します。 End Sub
Sub Allocate	Sub Allocate() 'ここに、配賦ルールを入力します。 End Sub
Sub Input	Sub Input() 'ここに、入力ルールを入力します。 End Sub
Sub NoInput	Sub NoInput() 'ここに、NoInput ルールを入力します。 End Sub
Sub Consolidate	Sub Consolidate() 'ここに、連結ルールを入力します。 End Sub
Sub Dynamic	Sub Dynamic() 'ここに、動的ルールを入力します。 End Sub

表 10-3 (続き) ルール・ルーチン

ルール・ルーチン	構文
Sub Transactions	<pre>Sub Transactions() 'ここに、取引ルールを入力します。 End Sub</pre>
Sub EquityPickup	<pre>Sub EquityPickUp() 'ここに、エクイティ・ピックアップ・ルールを入力します。 End Sub</pre>
Sub OnDemand	<pre>Sub OnDemand_<ruleName> 'ここに、オンデマンド・ルールを入力します。 End Sub</pre>

エクイティ・ピックアップ・ルールの例

次のセクションでは、資本ピックアップ・ルールのサンプル・セクションを示します。エクイティ・ピックアップを計算するために、アプリケーション管理者は、**Sub EquityPickUp** という名前のルール・ファイルに新しいセクションを作成する必要があり、ここで、EPU の計算が定義されます。セクションを実行するときのデフォルトの視点は次のとおりです。

- 現在のシナリオ、年および期間
- エンティティ: 処理されるペアの所有者

- 値: エンティティ 通貨

```
Sub EquityPickUp()

Owner = Hs.Entity.Member

Owned = Hs.Entity.Owned

OwnerDefaultCurrency = HS.Entity.DefCurrency("")

lPown = Hs.GetCell("E#" & Owned & ".I#" & Owner &
".V#[None].A#[Shares%Owned].C1#[None].C2#[None].C3#[None].C4#[None]")

Hs.Clear "A#IncomeFromSubsidiary.I#" & Owned

Hs.Exp "A#IncomeFromSubsidiary.I#" & Owned & " = E#" & Owned & ".V#" &
OwnerDefaultCurrency & ".A#NetIncome.I#[ICP Top] *" & lPown

End Sub
```

オンデマンド・ルールの例

データ・フォームからオンデマンド・ルールを実行するために、管理者は、**Sub OnDemand** という名前のルール・ファイルに新しいセクションを作成する必要があり、ここで、オンデマンド・ルールが定義されます。

```
Sub OnDemand_Calculation

HS.Exp "A#CogsTP=15424"

HS.Exp "A#Admex=32452"

End Sub
```

Sub Calculate (他にはなし)で使用できるすべての **HS** 関数はオンデマンド・ルールでも使用できることに注意してください。**Sub Calculate** とは異なり、以前に **IsCalculated** データ・ポイントに書き込まれたデータは、オンデマンド・ルールの実行時にクリアされないことにも注意してください。

ルールのロード

ルール変更はデータに反映され、メタデータに依存します。そのため、ルールのロード・プロセスにより、Oracle Hyperion Financial Management サーバー・クラスタにグローバル・ロ

ックが適用されます。ルール・ロードは、事前に開始した次のタイプの操作が終了するまでは処理されません:

- 連結
- データ入力
- データ、仕訳またはセキュリティ・ロード
- データベースへのデータの抽出
- メンバー・リスト・ロード
- メタデータ・ロード

ルール・スキャン・プロセスには、ディメンション・メンバーとその他のパラメータが検証されるよう、ロード・プロセスと同じ制約が適用されます。ルール・スキャンとロードのプロセスはキューに入れられ、ブロック・タスクの終了後に自動的に開始されます。ルールは長時間にわたる連結中などではなく、小規模のアクティビティ実行中にロードすることをお勧めします。「タスクの実行」ページでは、連結ロードまたはデータ・ロードなど、どのプロセスが処理中であるかを確認できます。

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

ルール・ファイルに内部取引が含まれている場合、ルール・ファイル内の新規取引に対してアプリケーションの転記済取引を確認できます。ルール・ファイルの **Sub Transactions** セクションは、内部取引の詳細をサポートする勘定科目を定義します。

ノート:

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。

ノート:

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** ファイル拡張子も受け入れられますが、**RLE** ファイル拡張子の使用をお勧めします。

3. **オプション:** 転記済の内部取引が、ロードするルール・ファイルの **Sub Transactions** のセクションにある文と整合性があるかどうかを確認するには、「**整合性チェック**」を選択します。

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

 **ヒント:**

オプションをデフォルト値にリセットするには、「**リセット**」をクリックします。

6. **オプション:** ログ・ファイルをダウンロードするには、「**ダウンロード・ログ**」をクリックします。「**開く**」をクリックしてログ・ファイルを表示するか、「**保存**」をクリックしてファイルをローカルに保存する場所を選択します。

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 ファイル拡張子が使用されます。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って、抽出したファイルをダウンロードします。
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. **オプション:** ログ・ファイルをダウンロードするには、「**ダウンロード・ログ**」をクリックします。「**開く**」をクリックしてログ・ファイルを表示するか、「**保存**」をクリックしてファイルをローカルに保存する場所を選択します。

ルール・プロファイリング

ルール・プロファイリングを使用して、主要なサブルーチン・レベルでルールの実行に関する詳細な情報を取得します。ルール・プロファイルには、連結タスクで各エンティティの計算、換算および連結にかかった所要時間に関する統計情報が含まれます。ルール・プロファイルを表示して、ルールが効率的に実行されたことを確認し、ルールの実行中や連結でのエンティティの処理中に発生した問題に対処できます。

ルール・プロファイルを実行および表示するには、アプリケーション管理者セキュリティ・アクセス権が必要です。

ルール・ファイルへのプロファイル情報の追加

特定のプロファイリング・データを取得するには、ルール・ファイル内にブロックを定義します。

ルール・ファイルにプロファイル情報を追加するには:

1. ルール・ファイルを開き、次のようにして主要なサブルーチンをブロック・レベル情報で更新します。
 - HS.StartTimer を使用してブロックの開始を定義します。
 - HS.EndTimer を使用してブロックの終了を定義します。
 - ルール・ファイル内に一意のタイマー ID を定義します。任意の正の整数を使用します。
 - 「Calculate」などのユーザー・アクションを定義します。ブロック名には `_TimeID` が付加されます。

例:

```
HS.StartTimer 1, "Calculate", "", "", "" '1: Unique Timer ID,
Calculate: User Defined Block Name
```

```
                If Hs.Scenario.Member ="BudV1"  or
Hs.Scenario.Member="BudV2" Then
```

```
                If HS.Entity.IsBase "Geographical", Then
```

...

```
                HS.EndTimer 1 '1: Corresponding Unique Timer ID of
earlier User defined block to represent End of the Block
```

2. 更新されたルール・ファイルをロードします。
3. プロファイルするアクションを実行します。

ルール・プロファイリングの開始と停止

デフォルトでは、ルール・プロファイリングは有効化されていません。アプリケーションに対して有効化する必要があります。

プロファイリングは、すべてのサーバーのアプリケーションごとに有効化できますが、個々のサーバーに対してはできません。

ルール・プロファイリングを開始するには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「管理」、「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. アプリケーションを選択して右クリックし、「プロファイル要約」を選択します。

アプリケーションでプロファイリングが実行されていない場合、「プロファイル要約」ページには結果が表示されません。プロファイリングが以前実行されている場合は、前の実行結果を表示できます。

4. 画面上部で、「開始」を選択します。
5. 次の確認が表示されます: 「アクティブなユーザーがいる本番システムではプロファイリングを有効にしないようお勧めします。プロファイリングを有効にすると、システム・パフォーマンスに悪影響を及ぼす可能性があります。プロファイリングを有効にしますか?」確認のプロンプトで「はい」をクリックします。

ノート:

すでに開始しているプロファイリングを停止する場合は、「停止」をクリックします。

ルール・プロファイル結果の表示

ルールが呼び出されるアクションを実行すると、ルールの実行に関する情報がプロファイルによって取得されます。これらのアクションには、データ・グリッド、データ・フォームおよび「エクイティ・ピックアップ」モジュールから使用可能なプロセスが含まれます。アクションには、計算、連結、配賦、換算、入力、入力なし、動的、取引、エクイティ・ピックアップなどがあります。

「プロファイル要約」から、ルール・ビューまたはエンティティ・ビューで結果を表示できます。ルール・ビューには、実行された主要なルールが表示されます。

エンティティ・ビューには、エンティティ構造に基づいてエンティティが表示されます。これは連結ルールが実行された場合にのみ使用可能であり、連結、計算(子ノードを含む)および換算の各プロセスの結果が表示されます。

ルール・プロファイル結果を表示するには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「管理」、「アプリケーション」の順に選択します。
3. アプリケーションを選択して右クリックし、「プロファイル要約」を選択します。

「プロファイル要約」ページには、ルールが実行されたタスク、ユーザーおよび日時などを含む、ルール・プロファイルがリストされます。複数のプロファイルが存在する場合は、ドロップダウンから表示するプロファイルを選択します。

4. ルール・ビューまたは「エンティティ・ビュー」を選択します。
5. 結果ページをリフレッシュするには、「リフレッシュ」をクリックします。

ルール・ビュー

ユーザー・アクション当たり

- このユーザー・アクションの合計所要時間
- ルール処理の合計所要時間(合計システム時間に対する割合)

主要なサブルーチン当たり

- サブブロック - 実行されたサブルーチン
- 合計呼出し - ルール内のユーザー定義の主要サブルーチン/ブロックがユーザー・アクションに対して実行された回数
- 平均時間 - 各サブルーチンの実行の平均所要時間。個々の経過時間の合計をサブブロックの呼出し数で除算した結果です。
- 合計時間 - 個々のサブルーチンの経過時間の合計。たとえば、10 個のエンティティに対する主要なサブルーチンの計算が並列で実行され、それぞれに 1 秒かかった場合、合計時間は 10 秒になります。
- POV 詳細 - サブルーチンの実行における所要時間の上位エンティティ (最上位から最下位の順にソート)

Sub/Block	Total Invocations	Average Time	Total Time	Entity	Period	Value	Elapsed Time
Calculate	33	166	5506	UKSales	February	GSP Total	494.0
Consolidate	8	934	7474	UKSales	February	[Proportion]	454.0
CustomLogic	1	1	1				
Translate	1	250	250				

エンティティ・ビュー

エンティティ・ビューでは、「すべてを連結」、「データのあるものすべてを連結」、「変更済を連結」、「コントリビューションの計算」などの、ユーザーが開始した連結タスクに関する情報が示されます。エンティティ・バー・グラフから、行にカーソルをあわせてブロックの時間の割合を表示できます。エンティティ行は縮小および展開できます。

ユーザーの連結タスク当たり

連結タスクの合計所要時間

エンティティ 当たり

Block	Timings %	Cumulative Time
Base Level Calculate		2652
UnitedStates		5335
WestRegion		1622
EastRegion		5038
EastAdmin		187
EastSales		218
EastProduc		218
A		280
B		452
UKSales		658

Operations	Entity	Period	Value	Thread Time	Timestamp
Calculate Child Nodes	WestRegion	January		624	Wed Sep 02 15:21:36 EDT
Calculate Entity	WestRegion	January	USD	46	Wed Sep 02 15:21:36 EDT
Calculate Entity	WestRegion	January	USD Total	15	Wed Sep 02 15:21:37 EDT
Consolidate Node	WestRegion	January		218	Wed Sep 02 15:21:37 EDT

エンティティ階層内で各分岐を連結した所要時間。階層内のエンティティを基本レベルのエンティティまで展開できます。複数の親を持つエンティティは、重複エントリとして表示されます。

- 累積時間 - 左パネルの時間は、エンティティ 階層で分岐を連結した合計所要時間を示します。各エンティティに表示される値は、エンティティとその子孫の処理にかかった実際のクロック時間です。マルチスレッドによる連結の性質上、個々の子エンティティの合計時間を合計すると、親エンティティの合計時間を超えることがあります。
- 操作 - 右パネルの各行は、左パネルで強調表示されているエンティティの処理中にサブタスクを実行する個別のスレッドを示します。
- スレッド時間 - 各スレッドが各サブタスクを完了する時間

ルール・プロファイルの削除

連結や換算などの特定のアクション、またはアプリケーションのすべてのアクションのルール・プロファイル結果を削除できます。

ルール・プロファイル結果を削除するには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「管理」、「アプリケーション」の順に選択します。
3. アプリケーションを選択して右クリックし、「プロファイル要約」を選択します。
4. 次のいずれかのタスクを行います。
 - 特定のアクションの結果を削除するには、ドロップダウン・リストからアクションを選択し、隣にある「削除」アイコンをクリックします。
 - すべての結果を削除するには、トップ・メニューから「すべて削除」をクリックします。
5. 確認のプロンプトで「はい」をクリックします。

11

ルール関数

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、[関数の概要](#)を参照してください。

関数の概要

表 1 に、Oracle Hyperion Financial Management の関数と、その関数で使用できるオブジェクトおよびルールのタイプの概要を示します。関数は、アルファベット順に表示されます。各関数の詳細なセクションが表の後に追加されています。

ノート:

11.1.2.2 より前の Financial Management リリースから移行されたレガシー・アプリケーションは、"Custom1...4"オブジェクトを使用できます。11.1.2.2 で作成されたすべてのアプリケーションは、新しい Custom(Alias) オブジェクト構文を使用して、カスタム別名/短い説明/長い説明を使用して参照するカスタム・ディメンションを指定します。

ルール・タイプはこの表では次のように省略されています。

- Alloc - 配賦
- Calc - 計算
- Con - 連結
- Dyn - 動的計算

- EPU - エクイティ・ピックアップ
- Tran - 換算
- Trans - 取引

表 11-1 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。	勘定科目	Calc、Tran、Con、Alloc
AccountTypeID	メンバーの勘定科目タイプ ID を取得します。	勘定科目	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	データをセルに割り当てます。	HS	Alloc
AllowAdjFromChildren	このメンバーの子からの仕訳の転記が可能である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である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

表 11-1 (続き) Financial Management の関数

関数	説明	オブジェクト	ルールのタイプ
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。	シナリオ	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
CustomTop	勘定科目の CustomTopMember を取得します。	勘定科目	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。	勘定科目	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。	シナリオ	Calc、Tran、Con、Alloc
DefaultParent	メンバーのデフォルトの親を取得します。	勘定科目	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。	シナリオ	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

表 11-1 (続き) Financial Management の関数

関数	説明	オブジェクト	ルールのタイプ
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・ディメンション名および別名のリストを戻します。	HS	Calc、Tran、Con、Alloc
GetItem	連結する個々のレコードを取得します。	DataUnit	Calc、Tran、Con
GetItemIDs2	ディメンションの ID 番号を使用して、連結のプロセスを行う個々の記録を取得します。	DataUnit	Calc、Tran、Con
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	セルの送信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

表 11-1 (続き) Financial Management の関数

関数	説明	オブジェクト	ルールのタイプ
ICPTopMember	現在の勘定科目ディメンション・メンバー、または指定した勘定科目メンバーの ICPTopMember を取得します。	勘定科目	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「変更済」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。	勘定科目	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。	勘定科目	Calc、Tran、Con、Alloc
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

表 11-1 (続き) Financial Management の関数

関数	説明	オブジェクト	ルールのタイプ
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。	値	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。	値	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	丸め処理をオフにします。	HS	Calc、Tran、Con、Alloc
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。	期間	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・ディメンションの数を取得します。	HS	Calc、Tran、Con、Alloc

表 11-1 (続き) Financial Management の関数

関数	説明	オブジェクト	ルールのタイプ
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。	期間	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。	シナリオ	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 表から出資比率を取得します。	HS	エクイティ・ピックアップ
PeriodNumber	取得中のデータのビューの期間番号を取得します。	ビュー	Dynamic
PlugAcct	調整勘定科目を取得します。	勘定科目	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
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

表 11-1 (続き) Financial Management の関数

関数	説明	オブジェクト	ルールのタイプ
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	データを丸めます。	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	ディメンションのメンバーのセキュリティ・クラスを取得します。	Account、Scenario、Entity、Parent、Custom1-Custom4	Calc、Tran、Con、Alloc
SetCellTextWithLabel	指定された視点(POV)とセル・テキスト・レベルのセル・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	ディメンション・メンバーのプロセス管理の送信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。	シナリオ	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	Trans
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タイプが切り替えられ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

表 11-1 (続き) Financial Management の関数

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。	勘定科目	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 11-2 ABSExp 関数の構文

パラメータ	説明
<i>DestPOV</i>	データの配置先を識別する宛先視点。 勘定科目のメンバーを必ず指定し、オプションで ICP およびカスタム・メンバーを指定します。使用に関する次のルールに注意してください: <ul style="list-style-type: none"> • ICP メンバーを指定しないと、デフォルトで [ICP None] が使用されます。 • データベースに不要な値を移入することを避けるには、データを配置する場所の定義について、ルールをできるだけ明確にする必要があります。適切な方法は、勘定科目ディメンションに、明確に定義されたカスタム・ディメンションの交差を含めることです。この明確に定義された交差では、Oracle Hyperion 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。ディメンションの交差に関する考慮事項を参照してください。
<i>Expression</i>	計算式

戻り値

なし。

例

この例では、**StateTax** 勘定科目の金額を設定します。例では、**2014** の **Sales** 勘定科目の値と **2014** の **StateRate** 勘定科目のレートを乗算して、絶対値を計算します。

```
HS.ABSExp "A#StateTax = A#Sales.Y#2014 * A#StateRate.Y#2014"
```

AccountType

現在の勘定科目のメンバーまたは指定された勘定科目の勘定科目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

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タイプ。

ノート:

勘定科目タイプは、すべて大文字です。

有効な勘定科目タイプは以下のとおりです：

- 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 勘定の勘定科目タイプが REVENUE のときに、If...Then と End If の文の間にあるすべての文が実行されます。

```
If HS.Account.AccountType("Sales") = "REVENUE" Then
```

```
...
```

```
End If
```

AccountTypeID

現在の勘定科目のメンバーまたは指定された勘定科目の勘定科目タイプ 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

指定した勘定科目の ID。表 1 は、対応する ID の有効な勘定科目タイプを示します。

表 11-3 勘定科目タイプ ID

勘定科目タイプ	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 11-4 AddEntityToList 関数の構文

パラメータ	説明
<i>Parent</i>	有効な親エンティティの名前。
<i>Child</i>	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 11-5 AddEntityToListUsingIDs 関数の構文

パラメータ	説明
<i>ParentID</i>	親エンティティの ID。
<i>ChildID</i>	親エンティティの子の ID。

戻り値

なし

AddMemberToList

指定されたメンバーをメンバー・リストに追加します。この関数は、メンバー・リスト・ファイルでのみ使用できます。

構文

```
HS.AddMemberToList ("Member")
```

Member は、有効なディメンション・メンバーの名前です。

戻り値

なし

例

```
HS.AddMemberToList "July"
```

AddMemberToListUsingIDs

指定されたメンバーをメンバー・リストに追加します。この関数は、メンバー・リスト・ファイルでのみ使用できます。

構文

```
HS.AddMemberToListUsingIDs (MemberID)
```

MemberID は、有効なディメンション・メンバーの ID です。

戻り値

なし

Alloc

別の視点にデータを配賦します。この関数は、配賦ルールで使用できます。

構文

```
HS.Alloc ("SourcePOV", "DestPOV", "EntityList", "AllocExp", "PlugAccount")
```

表 11-6 Alloc 関数の構文

パラメータ	説明
<i>SourcePOV</i>	<p>配賦されるデータのソース視点。 勘定科目のメンバーを必ず指定し、オプションで ICP およびカスタム・メンバーを指定します。ICP およびカスタム・メンバーを指定しない場合は、次のようになります。</p> <ul style="list-style-type: none"> デフォルトの ICP メンバーが、[ICP Top] になります。 デフォルトのカスタム・メンバーはその勘定科目の TopMember です。
<i>DestPOV</i>	<p>データの配賦先を識別する宛先視点。 勘定科目のメンバーを必ず指定し、オプションで ICP およびカスタム・メンバーを指定します。使用に関する次のルールに注意してください:</p> <ul style="list-style-type: none"> ICP メンバーを指定しないと、デフォルトで [ICP None] が使用されます。 カスタム・メンバーを指定しないと、デフォルトで [None] が使用されます。 エンティティのメンバーを指定し、<i>EntityList</i> 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。
<i>EntityList</i>	<p>データの配賦先エンティティを識別するメンバー・リスト。システム定義の [Base] エンティティ・リストを使用するか、ユーザー定義のリストを使用できます。</p> <p>[Base] システム定義のリストを使用する場合は、宛先視点に親メンバーとして指定されたエンティティが使用されます。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>

表 11-6 (続き) Alloc 関数の構文

パラメータ	説明
<i>AllocExp</i>	<p>各エンティティに配賦されるデータを識別する式。この式は、次のタイプの値を含むことができます。</p> <ul style="list-style-type: none"> • 数値 • 数値を識別する勘定科目の式。勘定科目のメンバーは必ず指定し、オプションで ICP およびカスタム・ディメンションのメンバーを指定できます。使用に関する次のルールに注意してください: カスタム・メンバーを指定しない場合、デフォルトは TopMember です。 ICP メンバーを指定しない場合、デフォルトは [ICP Top] です。 <p>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。</p> <p>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。</p>
<i>PlugAccount</i>	<p>調整勘定科目の名前。この引数はオプションです。ソース視点の勘定科目と調整勘定の属性を比較して、必要に応じて、ソース視点にある勘定の金額の符号を逆にします。</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"> • ソース視点のエンティティ・メンバーが親の場合、またはソース視点の値メンバーが Entity Currency でない場合、この引数は使用できません。 • ICP のメンバーを指定しないと、デフォルトで [ICP None] が使用されます。 • カスタム・メンバーを指定しないと、デフォルトで [None] が使用されます。

戻り値

なし。

例

この例では、TangibleAssets 勘定科目からのデータは現金勘定科目に配賦されます。

```
Call HS.ALLOC
("A#TangibleAssets", "A#Cash", "NewEngland", "A#TangibleAssets", "A#Plug")
```

AllowAdjFromChildren

指定されているエンティティまたは親メンバーで、子からの仕訳の転記が可能であるかどうかを指定します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

ノート:

メンバーがツリー階層で別のメンバーのすぐ 1 つ下のレベルにある場合、そのメンバーは子です。

構文

```
HS.Entity.AllowAdjFromChildren("Entity")
```

```
HS.Entity.AllowAdjFromChildren(" ")
```

```
HS.Parent.AllowAdjFromChildren("Entity")
```

```
HS.Parent.AllowAdjFromChildren(" ")
```

Entity は、有効なエンティティまたは親メンバーの名前です。

この関数を現在のエンティティまたは親に適用するには、空白の文字列(" ")を使用します。

戻り値

指定されたエンティティの子からの仕訳の転記が可能な場合に **TRUE** となり、不可能な場合に **FALSE** となるブール式。

例

この例では、**France** というエンティティの子メンバーの仕訳を転記できる場合に、**If...Then** と **End If** の間にあるすべての文が実行されます。

```
If HS.Entity.AllowAdjFromChildren("France") = TRUE then
```

```
    ...
```

```
End If
```

AllowAdjs

指定されているエンティティまたは親メンバーで、仕訳の転記が可能かどうかを指定します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Entity.AllowAdjs("Entity")
```

```
HS.Entity.AllowAdjs(" ")
```

```
HS.Parent.AllowAdjs("Entity")
```

```
HS.Parent.AllowAdjs(" ")
```

Entity は、有効なエンティティまたは親メンバーの名前です。

この関数を現在のエンティティまたは親に適用するには、空白の文字列(" ")を使用します。

戻り値

指定されたエンティティの仕訳の転記が可能な場合に **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文が実行されます。

```
If HS.Entity.AllowAdjs("France") = TRUE then
```

```
...
```

```
End If
```

ApplicationName

ルールが実行されているアプリケーションの名前を戻します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.ApplicationName
```

戻り値

ルールが実行されているアプリケーションの名前。

例

```
strApp=HS.ApplicationName
```

CalculateExchangeRate

ある通貨から別の通貨への為替レートを計算します。

構文

```
HS.CalculateExchangeRate (ScenarioID, YearID, PeriodID, EntityID,  
RateAccountID, FromCurrencyID, ToCurrencyID)
```

表 11-7 CalculateExchangeRate 関数の構文

パラメータ	説明
ScenarioID	シナリオの ID。

表 11-7 (続き) CalculateExchangeRate 関数の構文

パラメータ	説明
<i>YearID</i>	年の ID。
<i>PeriodID</i>	期間の ID。
<i>EntityID</i>	エンティティの ID。
<i>RateAccountID</i>	レート勘定の ID。
<i>FromCurrencyID</i>	換算元の通貨の ID。
<i>ToCurrencyID</i>	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 11-8 CalculateRate 関数の構文

パラメータ	説明
<i>ScenarioID</i>	シナリオの ID。
<i>YearID</i>	年の ID。
<i>PeriodID</i>	期間の ID。
<i>EntityID</i>	エンティティの ID。
<i>ValueID</i>	値の ID。
<i>RateAccountID</i>	レート勘定の ID。

戻り値

指定されたエンティティの為替レート。

CalcStatus

指定された視点の算出ステータスを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

- 配賦

構文

```
HS.CalcStatus("Scenario.Year.Period.Entity.Value")
```

表 11-9 CalcStatus 関数の構文

パラメータ	説明
<i>Scenario</i>	有効なシナリオ・メンバーの名前。
<i>Year</i>	有効な年。
<i>Period</i>	有効な期間。
<i>Entity</i>	有効なエンティティ・メンバーの名前。
<i>Value</i>	有効な値メンバーの名前。

戻り値

指定された視点の算出ステータスを示す文字列。有効なステータス・コードを次に示します。

表 11-10 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ディメンションのデータは、変更されていません。
OK ND	OK - データなし。データは計算済ですが、データのないセルは計算に含まれていません。
OK SC	OK - システム変更済。指定したシナリオ、年、期間、エンティティおよび値ディメンションのデータに影響する変更が実行されています。たとえば、新しいルール・ファイル、メタデータ・ファイルまたはメンバー・リスト・ファイルがロードされたか、通貨レートが変更されました。値 10,000 など、データ自体は変更されていません。ディメンション・メンバー属性などに変更が行われています。たとえば、勘定科目に関連付けられている小数点以下の桁数が 2 に設定されている場合、値を 10,000.00 に変更する必要があります。
CH	計算が必要。指定されたシナリオ、年、期間、エンティティおよび値ディメンションの少なくとも 1 つが変更されているか、メタデータ・パラメータまたはルールが変更されています。その結果、計算が実行されていないので、このディメンションのその他のデータ・セルが最新ではない可能性があります。基本レベルのエンティティの場合は、データ・セルがデータ入力か、データ・ファイルのロードで入力された可能性があります。エンティティの場合は、データ・セルが仕訳の転記で入力された可能性があります。

表 11-10 (続き) 算出ステータス・コード

ステータスのコード	説明
CH ND	計算が必要 - データなし。これは、セルで計算が実行される初回を示します。
TR	換算が必要。選択された値ディメンションのメンバーは、エンティティのデフォルト通貨ではなく、換算された値が最新のものではない可能性があります。
TR ND	換算が必要 - データなし。これは、セルで換算が実行される初回を示します。
CN	連結が必要。次の変更が行われたため、指定したシナリオ、年、期間、エンティティおよび値ディメンションのデータが最新ではない可能性があります： <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ディメンションのデータが、管理者によってロックされました。手動で変更することも、計算で変更することもできなくなりました。 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ディメンションにデータがありません。
NoAccess	ユーザーが、指定したディメンションのメンバーへのアクセス権を持っていません。

例

この例では、指定された視点のステータスが「OK」のときに、**If...Then** と **End If** の間にあるすべての文が実行されます。

```
If HS.CalcStatus("S#Actual.Y#2014.P#January.E#Connecticut.
V#<EntityCurrency>") = "OK" Then
```

```
...
```

```
End If
```

CellTextUnitItem

連結、計算または換算時に処理するセル・テキスト単位アイテムを戻します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

構文

```
HS.OpenCellTextUnit("", "[Default]", "Entity", "Ascending")
```

戻り値

The entry for the specified item.

例

```
Set MyCellTextUnit = HS.OpenCellTextUnit("", "[Default]", "Entity",
"Ascending")

NumItems= MyCellTextUnit.GetNumItems

For i = 0 to NumItems - 1

Scenario = MyCellTextUnit.Item(i).Scenario

Year = MyCellTextUnit.Item(i).Year

Period = MyCellTextUnit.Item(i).Period

Entity = MyCellTextUnit.Item(i).Entity

Value = MyCellTextUnit.Item(i).Value

Account = MyCellTextUnit.Item(i).Account

ICP = MyCellTextUnit.Item(i).ICP

Flow = MyCellTextUnit.Item(i).Custom("Flows")

Nature = MyCellTextUnit.Item(i).Custom("Nature")

...

...

...

CellText = MyCellTextUnit.Item(i).CellText

POV = MyCellTextUnit.Item(i).POV

Next
```

Clear

勘定科目、ICP およびカスタム・メンバーの組合せからデータを除去します。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Clear "Account.ICP.Custom1.Custom2.Custom3.Custom4"
```

表 11-11 Clear 関数の構文

パラメータ	説明
<i>Account</i>	有効な勘定科目メンバーの名前。
<i>ICP</i>	有効な ICP メンバーの名前。
<i>Custom1</i> 、 <i>Custom2</i> 、 <i>Custom3</i> 、 <i>Custom4</i>	カスタム 1、カスタム 2、カスタム 3 およびカスタム 4 の有効なメンバーの名前。

現在の視点(現在のエンティティ、期間、シナリオ、値、ビューおよび年のメンバー)と交差するすべてのセルのデータを除去するには、次の例のように、**A#**に続けて **ALL** キーワードを入力します。

```
HS.Clear "A#ALL"
```

セルおよびカスタムまたは ICP のディメンションのすべての交差をクリアするには、**ALL** キーワードを使用するか、**A#**文字を省略します。この例では、**A#**文字を省略して、カスタム 1 ディメンションの **GolfBalls** メンバーと交差するすべての勘定科目からデータをクリアします。

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

戻り値

なし。

例

この例では、**Sales** 勘定科目がカスタム・ディメンションの **GolfBalls** メンバーと交差する部分に保管されているデータをクリアします。

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

Con

[Proportion]および[Elimination]の値ディメンションのメンバーにデータを入力します。この関数は、連結ルールで使用できます。

構文

Hs.Con (*"DestPOV", Factor, "Nature")

表 11-12 Con 関数の構文

パラメータ	説明
<i>DestPOV</i>	次の要素の組合せです。 <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
<i>Factor</i>	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。
<i>Nature</i>	<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 表からレポートされます。要件によっては、RTS/RTD 表エントリの量を減らすために、[Proportion] HS.Con エントリではなく、[Elimination] HS.Con エントリの仕訳データを作成する方が適切な場合もあります。</p> <p>RTS/RTD 表・エントリを生成する Nature パラメータを使用すると、データベースのサイズが増大し、連結のパフォーマンスに影響する可能性があります。</p>

戻り値

なし。

例

```
Call HS.Con ("V#[Elimination]",-1*dPCon,"")
```

```
Call HS.Con ("V#[Elimination]",-1*dPCon, "Elimination")
```

```
Call HS.Con ("V#[Elimination]"PCON, "DefaultConsolidation")
```

Consol1、Consol2、Consol3

指定された `parent.entity` ノードの `Consol1`、`Consol2` または `Consol3` 勘定科目から値を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 配賦

構文

シナリオ、年、期間メンバーと、`parent.entity` 形式で指定したエンティティ・メンバーを組み合わせます。

```
HS.Node.Consoln ("S#Scenario.Y#Year.P#Period.E#Parent.Entity")
```

```
HS.Node.Consoln ("")
```

ノート:

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 11-13 `Consol1`、`Consol2`、`Consol3` の関数の構文

パラメータ	説明
<i>Scenario</i>	有効なシナリオ・メンバーの名前。
<i>Year</i>	有効な年。
<i>Period</i>	有効な期間。
<i>Parent.Entity</i>	<code>Parent.Entity</code> 形式の、有効なノードの名前。

戻り値

`Consol1`、`Consol2`、`Consol3` のシステム勘定科目の値。

例

この例では、**Actual**(実績)シナリオから **Group1.Ent1** ノードの **Consol1** の値を取得します。

```
dVar1 = HS.Node.Consol1 ("S#Actual.E#Group1.Ent1")
```

ConsolidateYTD

現在のシナリオ・メンバーまたは指定されたシナリオ・メンバーが年次累計で連結されるかどうかを調べます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Scenario.ConsolidateYTD("Scenario")
```

```
HS.Scenario.ConsolidateYTD("")
```

「Scenario」は有効なシナリオ・メンバーの名前です。

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

戻り値

シナリオが累計メソッドを使用して連結される場合は **TRUE** となり、期別メソッドを使用して連結される場合は **FALSE** となるブール式。

例

この例では、**Actual**(実績)シナリオが年次累計メソッドを使用して連結されるように設定されている場合、**If...Then** と **End If** の文の間にあるすべての文が実行されます。

```
If HS.Scenario.ConsolidateYTD("Actual") = TRUE Then
```

```
...
```

```
End If
```

ContainsCellText

指定されたセルにセル・テキストが含まれているかどうかを調べます。この関数は、次のタイプのルールで使用できます:

- 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。ディメンションを指定していない場合は、次のデフォルト値が使用されます。

- Account - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- Scenario - 現在のシナリオ・メンバー
- Entity - 現在のエンティティ・メンバー
- Value - 現在の値メンバー
- Year および Period - 現在のメンバー

ノート:

この関数を計算ルールで使用すると、デフォルト・エントリが適用されます。配賦ルール、換算ルールまたは連結ルールで使用する場合は、完全なサブキューブの Account/ICP/Custom POV を指定する必要があります。シナリオ、年、期間、ビュー、エンティティおよび値ディメンションは、デフォルトで現在のメンバーに設定されます。

戻り値

指定したデータ・セルに、任意のセル・テキスト・ラベルのセル・テキストが含まれる場合は TRUE になるブール式; 含まれない場合は FALSE。

例

この例では、指定されたセルに、任意のセル・テキスト・ラベルのセル・テキストが含まれていない場合に、If...Then と End If の各行の間にある文が実行されます。

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

```
...
```

```
End If
```

 ノート:

セル・テキスト・ラベルを使用している場合、既存のセル・テキスト・ラベル([Default]を含む)のいずれかにセル・テキスト・エントリが含まれると、この関数は値 True を返します。

ContainsCellTextWithLabel

指定されたラベルのセル・テキストがセルに入っているかどうかを調べ、セル・テキスト情報を更新します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.ContainsCellTextWithLabel ("POVExpression", "CellTextLabel")
```

POVExpression は、メンバーの組合せです。ディメンションを指定しないと、次のデフォルト値が使用されます。

- Account - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- Scenario - 現在のシナリオ・メンバー
- Entity - 現在のエンティティ・メンバー
- Value - 現在の値メンバー
- Year および Period - 現在のメンバー

CellTextLabel は、デフォルトのセル・テキスト・ラベル([Default])か、ロードされた有効なセル・テキスト・ラベルの 1 つです。

 ノート:

この関数を計算ルールで使用すると、デフォルト・エントリが適用されます。配賦ルール、換算ルールまたは連結ルールで使用する場合は、完全なサブキューブの Account/ICP/Custom POV を指定する必要があります。シナリオ、年、期間、ビュー、エンティティおよび値ディメンションは、デフォルトで現在のメンバーに設定されます。

戻り値

指定したデータ・セルに、指定されたセル・テキスト・ラベルのセル・テキストが含まれる場合は **TRUE** になるブール式; 含まれない場合は **FALSE** になります。

例

この例では、指定されたセルに、指定されたセル・テキスト・ラベルのセル・テキストが含まれていない場合に、**If...Then** と **End If** の各行の間にある文が実行されます。

```
If HS.ContainsCellTextWithLabel ("A#Sales.C1#Prod1.C2#Region1",  
"Rating") = "False" then ... End If
```

Currency

アプリケーションまたは現在の値ディメンションのメンバーの通貨を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.AppSettings.Currency
```

```
HS.Value.Currency
```

戻り値

アプリケーションまたは値メンバーの通貨の名前を示す文字列。

例

この例では、アプリケーションの通貨が **Euro** の場合に、**If...Then** と **End If** の間にあるすべての文が実行されます。

```
If HS.AppSettings.Currency = "Euro" Then
```

```
...
```

```
End If
```

CustomTop

現在のメンバーまたは指定された勘定科目メンバーの **CustomTopMember** を戻します。この関数は、次のタイプのルールで使用できます:

- 計算

- 換算
- 連結
- 配賦

構文

```
HS.Account.CustomTop("Member","", "CustomDimName")
```

```
HS.Account.CustomTop("", "CustomDimName")
```

ここで、*CustomDimName* はカスタム・ディメンションで、*Member* は勘定科目ディメンション・メンバーです。指定されていないメンバーは、**Sub Consolidate** サブルーチンでのみ適切です。

ノート:

関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

戻り値

指定した勘定科目メンバーの **CustomTopMember**。

例

```
HS.Account.CustomTop("Prod", "Sales")
```

DataUnitItem

連結、計算または換算時に処理するデータ単位アイテムを戻します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

構文

```
HS.OpenDataUnit("")
```

戻り値

指定したアイテムのメンバー・ラベル。

例

```
Set MyDataUnit = HS.OpenDataUnit("")

NumItems= MyDataUnit.GetNumItems

For i = 0 to NumItems - 1

Account = MyDataUnit.Item(i).Account

ICP = MyDataUnit.Item(i).ICP

Flow = MyDataUnit.Item(i).Custom("Flows")

Nature = MyDataUnit.Item(i).Custom("Nature")

...

...

...

Data = MyDataUnit.Item(i).Data

POV = MyDataUnit.Item(i).POV

Next
```

Decimal

指定した勘定科目の小数点以下の桁数を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Account.Decimal("AccountName")
```

```
HS.Account.Decimal("")
```

ノート:

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 11-14 Decimal 関数の構文

パラメータ	説明
<i>AccountName</i>	勘定科目の有効なメンバーの名前
<i>Var1</i>	勘定科目メンバーを示す VBScript 変数

戻り値

勘定科目に割り当てられる小数点以下の桁数の値を表す整数。有効な値は 0 から 9 です。

例

この例では、Sales 勘定の小数点以下の桁数を 2 に設定している場合に、If...Then と End If の間にあるすべての文が実行されます。

```
If HS.Account.Decimal("Sales") = 2 Then
```

```
...
```

```
End If
```

DefaultFreq

現在のシナリオのメンバーまたは指定されたシナリオのメンバーのデフォルトの期間単位を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

- 配賦

構文

```
HS.Scenario.DefaultFreq("Scenario")
```

```
HS.Scenario.DefaultFreq("")
```

ノート:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

Scenario は、有効なシナリオ・メンバーの名前です。

戻り値

シナリオのデフォルトの期間単位を示す文字列。

例

この例では、**Actual** シナリオのデフォルトの期間単位が **YTD** の場合に、**If...Then** と **End If** の間にあるすべての文が実行されます。

```
If HS.Scenario.DefaultFreq("Actual") = "YTD" Then
```

```
...
```

```
End If
```

DefaultParent

現在のメンバーまたは指定されたメンバーのデフォルトの親を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.<Object>.DefaultParent("Member")
```

```
HS.<Object>.DefaultParent("")
```

```
HS.Custom("Label").DefaultParent("Member")
```

```
HS.Custom("Label").DefaultParent("")
```

ここで、*Member* は、有効なディメンション・メンバーです。

ノート:

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 11-15 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"> 勘定科目 Custom1...4 カスタム(カスタム・ディメンション・ラベル) エンティティ シナリオ
Member	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> 勘定科目 Custom1...4 カスタム エンティティ シナリオ

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 11-16 DefaultTranslate 関数の構文

パラメータ	説明
<i>dRateForBalanceAccounts</i>	レートの数字
<i>dRateForFlowAccounts</i>	レートの数字
<i>bUserPVAForFlowAccounts</i>	TRUE または FALSE
<i>bUsePVAForBalanceAccounts</i>	TRUE または FALSE

戻り値

なし。

例

この例では、親メンバーが **United States** の場合に、**If...Then** と **End If** の間のすべての文が実行されます。

```
If HS.Parent.Member="UnitedStates" Then  
  
    HS.DefaultTranslate .25, .27, True, False  
  
End If
```

DefaultView

現在のシナリオのメンバーまたは指定されたシナリオのメンバーのデフォルトのビューを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Scenario.DefaultView("Scenario")  
  
HS.Scenario.DefaultView("")
```

 ノート:

この関数を現在のメンバーに適用するには、空白の文字列("")を使用します。

Scenario は、有効なシナリオ・メンバーの名前です。

戻り値

指定されたシナリオのデフォルトのビューを示す文字列。有効な値は YTD と Periodic です。

例

この例では、**Actual**(実績)シナリオのデフォルトのビューが YTD である場合に、If...Then と End If の間にあるすべての文が実行されます。

```
If HS.Scenario.DefaultView("Actual") = "YTD" Then
```

```
...
```

```
End If
```

DefCurrency

現在のエンティティまたは親メンバー、または指定のエンティティまたは親メンバーのデフォルト通貨を取得します。エンティティを指定すると、エンティティの通貨が戻されます。親通貨を取得するには、親エンティティを指定する必要があります。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.<Object>.DefCurrency("Entity")
```

```
HS.<Object>.DefCurrency("")
```

 ノート:

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 11-17 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"> エンティティ Parent
Entity	有効なエンティティ・メンバーの名前

戻り値

指定したエンティティまたは親のデフォルト通貨を含む文字列。

例

この例では、**Connecticut** というエンティティと **EastRegion** というエンティティのデフォルト通貨が同じでない場合に、**If...Then** と **End If** の間にあるすべての文が実行されます。

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

...

```
End If
```

Down

指定された **parent.entity** ノードの直接出資比率を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

シナリオ、年、期間メンバーと、**parent.entity** 形式で指定したエンティティ・メンバーを組み合わせます。

```
HS.Node.Down("S#Scenario.Y#Year.P#Period.E#Parent.Entity")
```

```
HS.Node.Down("")
```

 ノート:

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 11-18 DOWn 関数の構文

パラメータ	説明
<i>Scenario</i>	有効なシナリオ・メンバーの名前
<i>Year</i>	有効な年
<i>Period</i>	有効な期間
<i>Parent.Entity</i>	Parent.Entity 形式の、有効なノードの名前

戻り値

直接出資比率を示す数値。

例

```
dVar1 = HS.Node.DOWn("S#Actual.Y#2014.P#Q1.E#Group1.Ent1")
```

Dynamic

計算が必要な動的勘定科目の式を指定します。この関数は、動的ルールで使用できます。

この関数は、同じサブキューブ内のデータのみを参照できます。別のサブキューブのデータを参照する必要がある場合は、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、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。

 ノート:

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 = A#AccumUnits.I#ICP None.C1#None.C2#None /  
HS.View.PeriodNumber"
```

構文

```
HS.Dynamic "DestPOV = Expression"
```

表 11-19 Dynamic 関数の構文

パラメータ	説明
<i>DestPOV</i>	「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。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。
<i>Expression</i>	計算式

**ノート:**

Dynamic 関数では、IF...THEN 文はサポートされません。

戻り値

なし。

例

Sub Dynamic

```
HS.Dynamic "A#GM% = A#GM/(A#Sales * 100)/HS.View.PeriodNumber"
```

End Sub

GM%勘定科目の予想結果は次のとおりです。

Custom1	売上高	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 11-20 Exp 関数の構文

パラメータ	説明
<i>DestPOV</i>	データの挿入先を識別する宛先視点。使用に関する次のルールに注意してください: データベースに不要な値を移入することを避けるには、データを配置する場所の定義について、ルールをできるだけ明確にする必要があります。適切な方法は、勘定科目ディメンションに、明確に定義された ICP およびカスタム・ディメンションの交差を含めることです。この明確に定義された交差では、 Oracle Hyperion 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。
<i>Expression</i>	計算式。

データの宛先は等号の左側にあり、割り当てられるデータは右側にあります。この例は、UnitsSold 勘定科目およびカスタム・ディメンションの[None]メンバーと交差するセルを 40000 に設定します。

```
HS.Exp "A#UnitsSold.C1#[None].C2#[None].C3#[None].C4#[None]" _
& " = 40000"
```

等号の右側で、勘定科目式の文字を使用してディメンション・メンバーを表すことができます。したがって、セルのグループのデータをセルの別のグループに代入できます。この例は、Taxes 勘定科目およびカスタム・ディメンションの[None]メンバーと交差するセルを、Sales 勘定科目および指定されたカスタム・ディメンションと交差するセルのデータの 6 パーセントに設定します。

```
HS.Exp "A#Taxes.C1#[None].C2#[None].C3#[None].C4#[None]" _
& " = A#Sales.C1#AllProducts.C2#AllCustomers.C3#[None]" _
& ".C4#[None] * .06"
```

💡 ヒント:

1 つの Exp 文に複数の勘定科目を設定できます。複数の勘定科目の同時設定を参照してください。

ディメンションの交差に関する考慮事項

Exp がデータを取得し、Exp がデータを配置するディメンションの交差を指定しない場合は、ソース交差と宛先交差は次の要素によって決定されます。

- 宛先。内部取引パートナ・ディメンションまたはカスタム・ディメンションのメンバーが等号の左側に指定されていない場合は、Exp によって、勘定科目およびディメンションの有効な各交差にデータが挿入されます。宛先勘定科目を指定していない場合は、Oracle Hyperion Financial Management によって、現在の視点に有効なすべての勘定科目にデータが挿入されます。[複数の勘定科目の同時設定](#)を参照してください。
- ソース。ディメンションのメンバーが等号の右側に指定されていない場合は、次のようないくつかの可能性がります。
 - ディメンションのメンバーが 1 つのみの場合、Exp はこのメンバーとソース勘定科目の交差からデータを取得します。
 - ディメンションにソース勘定科目との有効な交差が 1 つのみある場合、Exp はこの交差からデータを取得します。
 - ディメンションにソース勘定科目と交差するメンバーが複数ある場合、データのソース交差は次のように等号の左側によって決定されます。
 - * メンバーが左側に指定されている場合は、Exp は、このメンバーとソース勘定科目の交差からデータの取得を試みます。
 - * メンバーが左側に指定されていない場合は、Exp は、宛先勘定科目およびディメンションのメンバーの有効な各交差へのデータの挿入を試みます。Exp は、メンバーおよびソース勘定科目の対応する交差から宛先交差のデータを取得します。

ノート:

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している詳細な例については、[Exp およびディメンションの交差に関する考慮事項](#)を参照してください。

期間および年のキーワード

動的なルールを作成するには、メンバー名かわりに[表 1](#)のキーワードを使用して、年または期間の宛先ディメンションのメンバーを表します:

表 11-21 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。

表 11-21 (続き) 期間および年のキーワードと説明

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。

 ノート:

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"
```

 ノート:

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ず。つまり、データとして **0.00** が含まれている勘定科目のステータスは **NoData** にはなりません。

Exp 内での他の関数の配置

関数が単一の数値を戻す場合は、**Exp** 関数内のその関数をネストできます。ただし、**String** 引数が含まれている関数をネストすると、**String** を引用符で囲えません。次の例では、**NumBase** 関数が **Exp** 関数内でネストされているため、その **String** 関数は引用符で囲えません。

```
HS.Exp "A#SalesAlloc = A#Sales/HS.Entity.NumBase(Regional)"
```

複数の勘定科目の同時設定

現在の視点と交差するすべての勘定科目にデータを挿入するには、勘定科目式で **All** を使用します。これを使用して、すべての勘定科目の期首残高を設定できます。次の例では、現在の期間が最初の期間であるかどうかを **IsFirst** 関数で検証します。最初の期間であると、**Exp** は、現在の期間の各勘定科目の値を前年の最後の期間の勘定科目の値に設定します。

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

```
    HS.Exp "A#ALL = A#ALL.Y#PRIOR.P#LAST"
```

```
End If
```

勘定科目およびカスタム・ディメンションまたは内部取引パートナー(**ICP**)ディメンションのすべての交差にデータを挿入するには、**All** キーワードを使用するか、または **A#**文字を指定しません。**A#**文字を指定しない次の例では、カスタム **1** ディメンションの **GolfBalls** メンバーと交差する各勘定科目にデータが挿入されます。**GolfBalls** と勘定科目の個々の有効な交差の場合は、前の期間の交差の金額が、現在の期間の交差に配置されます。

```
HS.Exp "C1#GolfBalls = C1#GolfBalls.P#PRIOR"
```

戻り値

なし。

例

この例では、StateTax 勘定科目の金額を設定します。2014 年の Sales 勘定科目の金額に、2014 年の StateRate 勘定科目のレートを乗算して計算します。

```
HS.Exp "A#StateTax = A#Sales.Y#2014 * A#StateRate.Y#2014"
```

Exp およびディメンションの交差に関する考慮事項

次の例は、[ディメンションの交差に関する考慮事項](#)に記載されている考慮事項を示しています。次のタイプの交差が取り上げられています。

- ソース勘定科目と宛先勘定科目に対して、すべての交差が有効である。[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。[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。[右側に 1 つの有効なメンバーがある場合](#)を参照してください。

次のすべての例で、Member001、Member002、Member003 という名前のカスタム 1 ディメンションのメンバーとともに、TargAcct および SourceAcct という名前の勘定科目が使用されます。表 1 に、これらすべての例のソース交差データを示します:

表 11-22 ディメンション交差のデータの例

Member	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 ディメンションに有効な交差が複数あった場合、Exp は SourceAcct と Member001 の交差からデータの取得を試みます。これが無効な交差であると、Exp は宛先勘定科目を変更しません。

GetCell

セルのデータを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.GetCell("POVExpression")
```

POVExpression は、有効な視点(POV)です。

戻り値

指定したセルにあるデータ。

 ノート:

関数によって複数の値が戻された場合はエラーが発生します。

例

この例では、Sales 勘定とカスタム 1 ディメンションの GolfBalls メンバーの交差にある値を **dData** 変数に割り当てます。

```
Dim dData
```

```
dData = HS.GetCell("A#Sales.I#[ICP  
None].C1#Golfballs.C2#Customer2.C3#Increases.C4#[None]")
```

GetCellNoData

セルに含まれているデータを取得し、セルにデータが含まれているかどうかを示します。この関数は、次のタイプのルールで使用できます:

- 計算

- 換算
- 連結
- 配賦

構文

```
HS.GetCellNoData("POV",Var1)
```

表 11-23 GetCellNoData 関数の構文

パラメータ	説明
POV	有効な視点。
Var1	セルにデータがあるかどうかを示す変数。

戻り値

戻される値は、セルの内容によって異なります。

- セルに実際のデータが含まれている場合は、データ値が戻され、*Var1* に戻されるブール値は **FALSE** になります。
- セルにデータが含まれていない場合は、データ値として **0** が戻され、*Var1* に戻されるブール値は **TRUE** になります。
- セルに計算結果データが含まれている場合は、その値が戻され、*Var1* に戻されるブール値は **FALSE** になります。

▲ 注意:

引数によって GetCellNoData に複数の値を戻させると、エラーが発生します。

例

この例では、Sales 勘定の金額が *dSalesAmt* 変数に割り当てられます。Sales 勘定にデータがない場合は、**If Then** 文と **End If** 文間にあるすべての文が実行されます。

```
dSalesAmt = HS.GetCellNoData("A#Sales.I#[ICP  
None].C1#[None].C2#[None].C3#[None].C4#[None]",bIsNoData)
```

```
If bIsNoData = TRUE then
```

```
...
```

```
End If
```

GetCellRealData

セルに含まれているデータを取得し、セルに実数データが含まれているかどうかを示します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.GetCellRealData("POV", Var1)
```

表 11-24 GetCellRealData 関数の構文

パラメータ	説明
<i>POV</i>	有効な視点。
<i>Var1</i>	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

視点からデフォルトのセル・テキストを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.GetCellText ("POVExpression")
```

POVExpression は、有効な視点(POV)です。

戻り値

視点(POV)の[Default]ラベル・セル・テキスト。

例

```
HS.GetCellText ("S#Actual.Y#2014.P#January.E#Connecticut.V#<Entity  
Currency>.A#Sales.I#[ICP None].C1#[None].C2#[None].C3#[None].C4#[None]")
```

ノート:

セル・テキスト・ラベルを使用している場合、この関数は、セル・テキスト・ラベル[Default]のセル・テキストを取得します。

GetCellTextWithLabel

指定したセル・テキスト・ラベルの視点からセル・テキストを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.GetCellTextWithLabel ("POVExpression", "CellTextLabel")
```

POVExpression は、メンバーの組合せです。ディメンションを指定していない場合は、次のデフォルト値が使用されます。

- Account - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- Scenario - 現在のシナリオ・メンバー
- Entity - 現在のエンティティ・メンバー
- Value - 現在の値メンバー
- Year および Period - 現在のメンバー

CellTextLabel は、デフォルトのセル・テキスト・ラベル([Default])か、ロードされた有効なセル・テキスト・ラベルの 1 つです。

ノート:

この関数を計算ルールで使用すると、デフォルトのディメンション・メンバー値が適用されます。配賦ルール、換算ルールまたは連結ルールで使用する場合は、Account、ICP および Custom1 から 4 のメンバーを指定する必要があります。シナリオ、年、期間、ビュー、エンティティおよび値は、デフォルトで現在のメンバーに設定されます。

戻り値

POV のセル・テキストおよび指定されたセル・テキスト・ラベル。

例

```
HS.GetCellTextWithLabel("A#Sales.I#[ICP  
None].C1#[None].C2#[None].C3#[None].C4#[None]","Rating")
```

GetCellType

セル・タイプを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

ノート:

勘定科目タイプは、すべて大文字です。

構文

```
HS.GetCellType("POVExpression")
```

POVExpression は、有効な視点(POV)です。

戻り値

特定のセルのタイプ。

次のタイプがあります:

- ASSET(資産)
- LIABILITY(負債)
- REVENUE(収益)
- EXPENSE(費用)
- FLOW(フロー)
- BALANCE(残高)
- BALANCERECURRING(経常残高)
- CURRENCYRATE(通貨レート)
- GROUPLABEL(グループラベル)

例

この例では、セル・タイプが **EXPENSE(費用)**かどうかをチェックします。**EXPENSE(費用)**の場合は、**If...Then** 文と **End If** 文間にあるすべての文が実行されます。

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

```
...
```

```
End If
```

GetCustomLabelArray

配列のカスタム・ディメンション・ラベルのリストを戻します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.GetCustomLabelArray (Names, Aliases)
```

戻り値

カスタムのショート名およびカスタムのロング名の 2 つの配列を返します。

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。

ノート:

この関数は、4 つ以下のカスタム・ディメンションにのみ、使用できます。4 つより多いカスタム・ディメンションには、[DataUnitItem](#) を使用してください。

構文

```
DataUnit.GetItem (lItem, strAccount, strICP, strCustom1, strCustom2,  
strCustom3, strCustom4, dData)
```

表 11-25 GetItem 関数の構文

パラメータ	説明
<i>lItem</i>	レコードの番号。
<i>strAccount</i>	勘定科目ディメンションの有効なメンバーの名前。
<i>strICP</i>	ICP ディメンションの有効なメンバーの名前。
<i>strCustom1</i>	カスタム 1 ディメンションの有効なメンバーの名前。
<i>strCustom2</i>	カスタム 2 ディメンションの有効なメンバーの名前。
<i>strCustom3</i>	カスタム 3 ディメンションの有効なメンバーの名前。
<i>strCustom4</i>	カスタム 4 ディメンションの有効なメンバーの名前。
<i>dData</i>	指定したセルにあるデータ。

戻り値

勘定科目メンバー、ICP メンバー、カスタム 1...4 メンバー、およびデータを含む配列。

例

```
Call DataUnit.GetItem(lItem, strAccount, strICP, strCustom1, strCustom2,  
strCustom3, strCustom4, dData)
```

```
If dData = 0 Then
```

```
...
```

```
End If
```

GetItemIDs2

ディメンション ID を使用して、連結する個々のレコードを取得します。この関数は、連結ルールで使用できます。

ノート:

この関数は、4 つ以下のカスタム・ディメンションにのみ、使用できます。4 つより多いカスタム・ディメンションには、[GetItemIDs2ExtDim](#) を使用してください。

構文

```
DataUnit.GetItemIDs2(lItem, lAccountID, lICPID, lCustom1ID, lCustom2ID,  
lCustom3ID, lCustom4ID, dData)
```

表 11-26 GetItemIDs2 関数の構文

パラメータ	説明
<i>lItem</i>	レコードの番号。
<i>lAccountID</i>	勘定科目ディメンションの有効なメンバーの ID 番号。
<i>lICPID</i>	ICP ディメンションの有効なメンバーの ID 番号。
<i>lCustom1ID</i>	カスタム 1 ディメンションの有効なメンバーの ID 番号。
<i>lCustom2ID</i>	カスタム 2 ディメンションの有効なメンバーの ID 番号。
<i>lCustom3ID</i>	カスタム 3 ディメンションの有効なメンバーの ID 番号。

表 11-26 (続き) GetItemIDs2 関数の構文

パラメータ	説明
<i>lCustom4ID</i>	カスタム 4 ディメンションの有効なメンバーの ID 番号。
<i>dData</i>	指定したセルにあるデータ。

戻り値

勘定科目、ICP、カスタム 1...4 のデータを含む変数。

例

```
Call DataUnit.GetItemIDs2(lItem, lAccount, lICP, lCustom1, lCustom2,
lCustom3, lCustom4, dData)
```

```
If dData = 0 Then
```

```
...
```

```
End If
```

GetItemIDs2ExtDim

ディメンション ID を使用して、連結する個々のレコードを取得します。この関数は、連結ルールで使用できます。

ノート:

この関数は、4 つより多いカスタム・ディメンションがある場合に使用されます。

構文

```
DataUnit.GetItemIDs2ExtDim(lItem, lAccountID, lICPID, lCustomID, dData)
```

表 11-27 GetItemIDs2 関数の構文

パラメータ	説明
<i>lItem</i>	レコードの番号。
<i>lAccountID</i>	勘定科目ディメンションの有効なメンバーの ID 番号。

表 11-27 (続き) GetItemIDs2 関数の構文

パラメータ	説明
<i>lICPID</i>	ICP デイメンションの有効なメンバーの ID 番号。
<i>lCustomID</i>	カスタム・デイメンションの有効なメンバーの ID 番号。
<i>dData</i>	指定したセルにあるデータ。

戻り値

デイメンションとメンバーの両方の ID を含む 2 列の配列。

例

```
Call DataUnit.GetItemIDs2(lItem, lAccount, lICP, lCustom5, dData)
```

```
If dData = 0 Then
```

```
...
```

```
End If
```

GetNumItems

連結するレコードの数を取得します。この関数は、連結ルールで使用できます。

構文

```
NumItems = DataUnit.GetNumItems
```

戻り値

データ単位内のレコードの数。

例

```
Set dataUnit = HS.OpenDataUnit("")

lNumItems = dataUnit.GetNumItems

for lItem = 0 to lNumItems - 1

' Get the next item from the Data Unit

Call dataUnit.GetItem(lItem, strAccount, strICP, strCustom1,
strCustom2, strCustom3, strCustom4, dData)
```

GetNumLID

指定した視点のライン・アイテムの詳細の数を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.GetNumLID("POVExpression")
```

POVExpression は有効な POV の組合せです。ディメンションを指定しないと、次の値が使用されます。

- Account - [none]
- ICP - [ICP None]
- Custom1...4 - [None]
- Scenario - 現在のシナリオ・メンバー
- Entity - 現在のエンティティ・メンバー
- Value - <エンティティ 通貨>
- Year および Period - 現在のメンバー

 **ノート:**

無効な交差を指定すると、0 が戻されます。

戻り値

指定したセルのライン・アイテムの詳細の数。

例

この例では、指定したセルにライン・アイテムの詳細が入力されていないと、If...Then 行と End If 行の間にあるすべての文が実行されます。

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

```
...
```

```
End If
```

GetRate

現在の視点または指定した視点の換算レートを取得します。この関数は、次のタイプのルールで使用できます。

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.GetRate("POVExpression")
```

POVExpression は有効なデータの視点(POV)です。



ノート:

レート勘定を指定する必要があります。

戻り値

指定した視点の為替レート。

例

```
dVar1 = HS.GetRate("S#Actual.Y#2014.P#March.V#Euro.E#Connecticut.A#AvgRate")
```

GetSubmissionGroup

セルのプロセス管理の送信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

プロセス管理の送信グループを表す整数です。有効な値は、**1** から **99** です。

例

```
dVar1=HS.GetSubmissionGroup("A#Sales.C1#Golfballs.C2#Tennisballs.C3#Soccerballs.C4#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** から **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、または親ディメンション・メンバーの持株会社を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Entity.Holding("Entity")
```

```
HS.Entity.Holding("")
```

```
HS.Parent.Holding("Entity")
```

```
HS.Parent.Holding("")
```

Entity は、有効なエンティティ・ディメンションまたは親メンバーの名前です。空白の文字列("")を使用して、この関数を現在のメンバーに割当てできます。

戻り値

指定されたエンティティ・メンバーまたは親の持株会社の名前を示す文字列。

例

この例では、**Europe** がエンティティ **France** の持株会社である場合、**If...Then** 行と **End If** 行の間にあるすべての文が実行されます。

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

```
...
```

```
End If
```

ICPTopMember

現在の勘定科目ディメンション・メンバー、または指定した勘定科目メンバーの **ICPTopMember** を取得します。この関数は、次のタイプのルールで使用できます。

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Account.ICPTopMember("AccountName")
```

```
HS.Account.ICPTopMember("")
```

```
HS.Account.ICPTopMember(Var1)
```

ノート:

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 11-28 ICPTopMember 関数の構文

パラメータ	説明
<i>AccountName</i>	勘定科目ディメンションの有効なメンバーの名前。
<i>Var1</i>	VisualBasic 変数。

戻り値

勘定科目の ICPTopMember の名前を示す文字列。

例

この例では、Sales 勘定の ICPTopMember が TotalProd である場合、If...Then 文と End If 文の間にあるすべての文が実行されます。

```
If HS.Account.ICPTopMember("Sales") = "TotalProd" Then
```

```
...
```

```
End If
```

ICPWeight

アプリケーションの ICP 加重を取得します。値ディメンションの[ICP Top]メンバーに集約される、内部取引パートナ・エンティティ [ICP Entities]の額の割合です。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算

- 連結
- 配賦

構文

```
HS.AppSettings.ICPWeight
```

戻り値

ICPTopMember に集約される ICP エンティティの割合。この値は、小数で表した百分率の値 (1.0 が 100 パーセント) です。

例

この例では、現在のアプリケーションの ICP Weight が 1 である場合に、If...Then 行と End If 行の間にあるすべての文が実行されます。

```
If HS.AppSettings.ICPWeight = 1 Then
```

```
...
```

```
End If
```

IDFromMember

指定したメンバーの ID 番号を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.<Object>.IDFromMember("Element")
```

表 11-29 IDFromMember 関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> Account Custom1...4 Custom (カスタム・ディメンション・ラベル) Entity ICP Parent Period Scenario Value Year View
Element	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> Account Custom1...4 Custom Entity ICP Parent Period Scenario Value Year

戻り値

指定したメンバーの ID 番号。メンバー・エントリが有効なメンバーではない場合、戻り値は-1 になります。

例

この例では、Connecticut というメンバーの ID 番号を取得します。

```
1EntityID = HS.Entity.IDFromMember("Connecticut")
```

```
1CustomID=HS.Custom("Prod").IDFromMember("P3000-Phones")
```

ImpactStatus

指定したデータ単位のステータスを「変更済」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 11-30 ImpactStatus 関数の構文

パラメータ	説明
<i>Scenario</i>	シナリオ・ディメンションの有効なメンバーの名前。
<i>Year</i>	有効な年。
<i>Period</i>	有効な期間。
<i>Entity</i>	エンティティ・ディメンションの有効なメンバーの名前。
<i>Value</i>	値ディメンションの有効なメンバーの名前。

ノート:

指定された POV が、処理されるデータ単位と同じシナリオ、年、期間およびエンティティである場合、エラーが発生し、データ単位には影響を与えません。ルールの実行時にターゲット期間がロックされる場合、VBScript エラーが戻され、データ単位には影響を与えません。

戻り値

なし。

例

```
HS.ImpactStatus "S#Actual.Y#2014.P#January"
```

Input

基本エンティティ・レベルで入力できるデータ・ポイントに対して親エンティティの<エンティティ通貨>にデータ入力できます。これには、IsCalculated としてフラグ設定されているデータ・ポイントおよび NoInput として設定されているデータ・ポイントは含まれません。エンティティ通貨の値ディメンションのみがサポートされます。この関数を使用すると、親エンティティ・レベルの値は、子のコントリビューションの合計に等しくなりません。

この関数は親エンティティ・レベルでの入力を可能にするため、子のコントリビューションは親エンティティのエンティティ通貨の値ディメンションにロール・アップしません。ただし、親子コントリビューション値が保管され、コントリビューション調整の値ディメンションへの仕訳の転記はできます。この関数は入力ルールで使用できます。

▲ 注意:

親エンティティ・レベルで入力できる勘定科目では、親エンティティのエンティティ通貨の値メンバーの値が、すべての子のコントリビューションの合計に等しくならぬ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 **Oracle Hyperion Financial Management** イプシロンに基づいてチェックします。この関数は、すべてのタイプのルールで使用できます。

-0.00000000000001 から 0.00000000000001 の差は、ゼロ差分と見なされます。

構文

```
BooleanValue = HS.IsAlmostEqual(Value1, Value2)
```

戻り値

渡された値が等しい場合は **TRUE** となるブール式、等しくない場合は **FALSE** となるブール式。

例

```
Dim BoolVal

Dim Value1

Dim Value2

Value1 = 10.1299999999

Value2 = 10.13

BoolVal = HS.IsAlmostEqual(Value1, Value2)

If BoolVal = true Then

    'do processing

Else

    'do Processing

End If
```

IsBase

現在のメンバーまたは指定したメンバーが、アプリケーションまたは指定した親の基本メンバーであるかどうかを調べます。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結
- 配賦

 **ノート:**

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)
```

ノート:

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 11-31 IsBase 関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> Account Custom1...4 Custom (カスタム・ディメンション・ラベル) Entity Parent
Parent	有効な親メンバー。 Parent は、Node と共に使用する場合にのみ必須です。
Element	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> Account Custom1...4 カスタム Entity Parent
Entity	エンティティ・ディメンションの有効なメンバーの名前。
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。

 ノート:

Node オブジェクトを使用した場合は、要素が指定した親のアクティブな基本メンバーかどうか調べられます。

戻り値

要素が指定した親の下位にある基本メンバーである場合に **TRUE** となり、親が指定されていない状態ではアプリケーションの基本メンバーとなるブール式。要素が基本メンバーではない場合は **FALSE** になります。

ノードの場合は、要素が指定した視点の親の下位にあるアクティブな基本エンティティである場合は **TRUE** になります。要素がアクティブな基本エンティティでない場合は **FALSE** になります。

例

この例では、Connecticut が EastRegion の下位にある基本エンティティであると、If...Then 行と End If 行の間にあるすべての文が実行されます。

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

```
...
```

```
End If
```

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

```
...
```

```
End If
```

IsCalculated

現在の勘定科目ディメンション・メンバーまたは指定した勘定科目メンバーが計算済勘定科目であるかどうかを調べます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Account.IsCalculated("Account")
```

Account は、有効な勘定科目メンバーの名前です。

```
HS.Account.IsConsolidated("")
```

ノート:

Sub Consolidate サブルーチンで関数を使用している場合にのみ、空白の文字列("")を使用して、この関数を現在のメンバーに適用できます。

戻り値

勘定科目が計算された勘定科目の場合は **TRUE** となるブール式、そうでない場合は **FALSE** となるブール式。

例

この例では、売上勘定が計算されると、**If...Then** 文と **End If** 文間にあるすべての文が実行されます。

```
If HS.Account.IsCalculated("Sales") = TRUE Then
```

```
...
```

```
End If
```

IsChild

現在のメンバーまたは指定したメンバーが、指定した親の子であるかどうかを調べます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

ノート:

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")
```

ノート:

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 11-32 IsChild 関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> Account Custom1...4 Custom (カスタム・ディメンション・ラベル) Entity Parent
Parent	親メンバーの名前(親は必須です。)
Element	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> Account Custom1...4 カスタム Entity Parent
Entity	エンティティ・ディメンションの有効なメンバーの名前。
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年
Period	有効な期間

 ノート:

Node オブジェクトを使用した場合は、メンバーが、指定した親のアクティブな子かどうか調べられます。

戻り値

要素が指定した親の子である場合に **TRUE** となり、そうでない場合に **FALSE** となるブール式。

ノードの場合は、要素が指定した親のアクティブな子である場合に **TRUE** となり、そうでない場合に **FALSE** となります。

例

この例では、Connecticut が EastRegion の子であると、If...Then 行と End If 行の間にあるすべての文が実行されます。

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

```
...
```

```
End If
```

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

```
...
```

```
End If
```

IsConsolidated

現在の勘定科目ディメンション・メンバーまたは指定した勘定科目メンバーが連結勘定科目であるかどうかを調べます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Account.IsConsolidated("Account")
```

Account は、有効な勘定科目メンバーの名前です。

```
HS.Account.IsConsolidated("")
```

ノート:

Sub Consolidate サブルーチンで関数を使用している場合にのみ、空白の文字列("")を使用して、この関数を現在のメンバーに適用できます。

戻り値

勘定科目が親勘定科目に連結される場合に **TRUE** となり、そうでない場合に **FALSE** となるブール式。

例

この例では、売上勘定科目が連結されると、**If...Then** 文と **End If** 文間にあるすべての文が実行されます。

```
If HS.Account.IsConsolidated("Sales") = TRUE Then
```

```
...
```

```
End If
```

IsDescendant

現在のメンバーまたは指定したメンバーが、指定した親の子孫であるかどうかを調べます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

ノート:

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)
```



ノート:

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 11-33 IsDescendant 関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> Account Custom1...4 Custom (カスタム・ディメンション・ラベル) Entity Parent
Parent	有効な親メンバーの名前。親は必須です。
Element	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> Account Custom1...4 カスタム Entity Parent
Entity	エンティティ・ディメンションの有効なメンバーの名前。
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。

 ノート:

ノードをオブジェクトとして使用する場合は、メンバーが指定した親のアクティブな子孫であるかどうか判断されます。

戻り値

要素が指定した親の子孫である場合に **TRUE** となり、そうでない場合に **FALSE** となるブール式。

ノードの場合は、要素が指定した親のアクティブな子孫である場合に **TRUE** となり、そうでない場合に **FALSE** となります。

例

この例では、**Connecticut** が **Regional** の子孫であると、**If...Then** 行と **End If** 行の間にあるすべての文が実行されます。

```
If HS.Entity.IsDescendant("Regional","Connecticut") = TRUE Then
```

```
...
```

```
End If
```

```
If HS.Custom("Prod").IsDescendant("All_Phones", P3000_Phones")
```

```
...
```

```
End If
```

IsFirst

現在の期間または年が、アプリケーションの最初の期間または年であるかどうかを調べます。現在の期間または年がアプリケーションの最初の期間または年であるかどうかの判別に、現在のシナリオのデフォルト期間単位が使用されます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.<Object>.IsFirst
```

<Object>は、次のキーワードのいずれかです。

- Period
- Year

戻り値

現在の期間または年がアプリケーションの最初の期間または年である場合に **TRUE** となり、そうでない場合に **FALSE** となるブール式。

例

この例では、現在の期間が最初の期間であると、**If...Then** 文と **End If** 文間にあるすべての文が実行されます。

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

```
...
```

```
End If
```

IsICP

現在の勘定科目ディメンション・メンバーまたはエンティティ・ディメンション・メンバー、あるいは指定した勘定科目メンバーまたはエンティティ・メンバーが内部取引パートナー(ICP)であるかどうかを調べます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.<Object>.IsICP("Element")
```

```
HS.<Object>.IsICP("")
```

ノート:

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 11-34 IsICP 関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> Account Entity
Element	勘定科目ディメンションまたはエンティティ・ディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> Account Entity

戻り値

勘定科目またはエンティティのメンバーが内部取引パートナーである場合に TRUE となり、そうでない場合に FALSE となるブール式。

例

この例では、売上勘定科目が内部取引パートナーであると、If...Then 行と End If 行の間にあるすべての文が実行されます。

```
If HS.Account.IsICP("Sales") = TRUE Then
```

```
...
```

```
End If
```

IsLast

現在の期間または年がアプリケーションの最後の期間または年であるかどうかを調べます。この判別には、現在のシナリオのデフォルト期間が使用されます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.<Object>.IsLast
```

<Object>は、次のキーワードのいずれかです。

- Period
- Year

戻り値

現在の期間または年がアプリケーションの最後の期間または年である場合に **TRUE** となり、そうでない場合に **FALSE** となるブール式。

例

この例では、現在の期間が最後の期間であると、**If...Then** と **End If** の文の間にあるすべての文が実行されます。

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

```
...
```

```
End If
```

IsTransCur

現在の値ディメンション・メンバーが、換算済の通貨メンバーであるかどうかを調べます。この関数は計算ルールで使用できます。

構文

```
HS.Value.IsTransCur
```

戻り値

現在の値メンバーが換算された通貨メンバーである場合に **TRUE** となり、そうでない場合に **FALSE** となるブール式。

例

この例では、値メンバーが換算済の通貨メンバーであると、**If...Then** 文と **End If** 文間にあるすべての文が実行されます。

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

```
...
```

```
End If
```

IsTransCurAdj

現在の値ディメンション・メンバーが、換算済の通貨調整メンバーであるかどうかを調べます。この関数は計算ルールで使用できます。

構文

```
HS.Value.IsTransCurAdj
```

戻り値

現在の値メンバーが換算された通貨調整メンバーである場合に **TRUE** となり、そうでない場合に **FALSE** となるブール式。

例

この例では、値メンバーが換算済の通貨調整メンバーであると、**If...Then** 文と **End If** 文間にあるすべての文が実行されます。

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

```
...
```

```
End If
```

IsValidDest

指定した視点が有効な宛先かどうかを判別します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

ノート:

この関数では、セルが計算されたセルであるかどうかはチェックされません。

構文

```
HS.IsValidDest ("POVExpression")
```

POVExpression は視点(POV)です。ディメンションを指定しないと、次のデフォルト値が使用されます。

- **Account - Sub Consolidate** サブルーチンで使用される場合は現在の勘定科目メンバー。それ以外の場合は、勘定科目は必須です。
- **Custom** および **ICP - Sub Consolidate** サブルーチンで現在のメンバーが使用されます。それ以外の場合は、勘定科目の **TopMember** が使用されます。

- **Scenario** - 現在のシナリオ・メンバー
- **Entity** - 現在のエンティティ・メンバー
- **Value** - 現在の値メンバー
- **Year** および **Period** - 現在のメンバー

戻り値

指定した視点が有効な宛先である場合は **TRUE**、そうでない場合は **FALSE** となるブール値。

例

この例では、指定した宛先が有効であると、**If...Then** 文と **End If** 文間にあるすべての文が実行されます。

```
If HS.IsValidDest("A#Sales.I#CT.C1#P1.C2#R1.C3#[None].C4#[None]") =  
TRUE Then  
  
    ...  
  
End If
```

IsZero

渡された値がゼロに近いかどうかを、事前に定義されている **Oracle Hyperion Financial Management** イプシロンに基づいてチェックします。この関数は、すべてのタイプのルールで使用できます。

この関数は、ゼロとの正確な比較のかわりに推奨されます。正確な比較では、浮動小数点算術によって、無視できる $1 \times 10 \text{ E-}10$ より小さいエラーを引き起こします。

次の代わりに:

```
Difference = Value1 - Value2  
  
If Difference = 0 Then  
  
    'process where Difference = 0  
  
Else  
  
    'process where Difference <> 0  
  
End If
```

次を使用します:

```
Difference = Value1 - Value2
```

```
If HS.IsZero(Difference) Then
```

```
    'process where Difference = 0
```

```
Else
```

```
    'process where Difference <> 0
```

```
End If
```

構文

```
BooleanValue = HS.IsZero(Value)
```

戻り値

渡された値がゼロに近い場合に **TRUE** になるブール値。それ以外の場合は、**FALSE** になります。

例

```
Dim BoolVal

Dim Value

Value = 0.000000001

BoolVal = HS.IsZero(Value)

If BoolVal = true Then

    'do processing

Else

    'do Processing

End If
```

List

指定したリストにある要素を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.<Object>.List("Parent","Listname")
```

```
HS.Node.List("Parent","Listname"."S#Scenario.Y#Year.P#Period")
```

```
HS.Custom("Dimension").List("Parent","Listname")
```

表 11-35 List 関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> Account Custom1...4 Custom (カスタム・ディメンション・ラベル) Entity Parent ICP Scenario
Parent	有効な親メンバーの名前。
Listname	有効なシステム・リストまたはユーザー定義リストの名前。 ノードの場合は、有効なエンティティ・システム・リストが必須。
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。

戻り値

指定したリストのすべての要素が含まれている配列。ノードの場合は、リスト内のアクティブな要素のみ。

例

この例では、現在の勘定科目のユーザー定義リスト **MyBaseList** の要素を取得します。

```
HS.Account.List("", "MyBaseList")
```

この例では、**TotalAssets** 勘定科目のシステム・リスト[Base]の要素を取得します。

```
HS.Account.List("TotalAssets", "[Base]")
```

```
Hs.Custom("Prod").List("Products", "[Base]")
```

Member

現在のメンバーの名前を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.<Object>.Member
```

<Object>は、次のキーワードのいずれかです。

- Entity
- Parent
- Period
- Scenario
- Value
- Year
- View

戻り値

現在のメンバー名が含まれている文字列。

値オブジェクトの場合、メンバーは、値に関連付けられている通貨ではなく現在の値メンバーの名前を戻します。たとえば、現在の値メンバーがエンティティ通貨で、値が USD 通貨に関連付けられている場合、`HS.Parent.Member` は USD ではなく、エンティティ通貨を戻します。

ヒント:

現在の値メンバーに設定されている通貨を取得するには、`DefCurrency` 関数を使用します。

例

この例では、現在のエンティティが `California` であると、`If...Then` と `End If` の文の間にあるすべての文が実行されます。

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

```
...
```

```
End If
```

MemberFromID

指定した ID 番号のディメンション・メンバーを取得します。この関数は、次のタイプのルールで使用できます:

- 計算

- 換算
- 連結
- 配賦

構文

HS.<Object>.MemberFromID(*ElementID*)

HS.Custom(*Dimension*).MemberFromID(*ElementID*)

表 11-36 MemberFromID 関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> • Account • Custom1...4 • Custom (カスタム・ディメンション・ラベル) • Entity • ICP • Parent • Period • Scenario • Value • Year • View
Parent	有効な親メンバー。
ElementID	次のいずれかのディメンションの有効なメンバーの ID 番号。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> • Account • Custom1...4 • カスタム • Entity • ICP • Parent • Period • Scenario • Value • Year

戻り値

ディメンション・メンバー名。

例

この例では、ID 番号 001 のメンバーを取得します。

```
strEntity = HS.Entity.MemberFromID(001)
```

```
strCustom = HS.Custom("Prod").MemberFromID(001)
```

Method

指定したメンバーの連結メソッドを取得します。ゼロ以外の値が複数あると、その中の最初に見つかった値が戻されます。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Node.Method("POVExpression")
```

POVExpression は、シナリオの組合せです。年、期間およびエンティティのメンバー。

戻り値

指定した視点の連結メソッドを示す文字列。

例

この例では、視点のメソッドが GLOBAL であると、If...Then 文と End If 文間にあるすべての文が実行されます。

```
If HS.Node.Method("S#Actual.Y#2014.P#January.E#Regional.Connecticut")  
= "GLOBAL" Then
```

```
...
```

```
End If
```

NoInput

特定のセルまたはセルのスライスにユーザーがデータを入力できないようにします。この関数は、入力または計算済ではない勘定科目がある場合に便利です。

計算済勘定科目を設定すると、実際にユーザーは計算済勘定科目に入力できなくなります。ただし、データ入力の一部のディメンション交差には可能で、その他のディメンション交差には不可である勘定科目を持っている場合は、**NoInput** を使用できます。この関数は、**NoInput** ルールで使用できます。

構文

```
HS.NoInput "POVExpression"
```

POVExpression は視点(POV)です。

戻り値

なし。

例

この例では、**Sales** 勘定科目と **2014** 年の **Budget** シナリオが交差するセルへの入力が禁止されます:

```
Sub NoInput
```

```
    HS.NoInput "S#Budget.Y#2014.A#Sales"
```

```
End Sub
```

NoRound

後に続くすべての **Exp** ステートメントの丸め処理をオフにします。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 配賦

ヒント:

Round 関数の引数として **0** を入力しても、丸め処理をオフにできます。たとえば、**HS.Round(0)** で丸め処理がオフになります。

構文

```
HS.NoRound
```

戻り値

なし。

例

この例では、**SalesRound** 勘定のセルに挿入される金額を小数第 2 位で丸め、次に **NoRound** 関数を使用して、**SalesNoRound** 勘定のセルに挿入される金額の丸め処理をオフにします。

```
HS.Round 0.1
```

```
HS.Exp "A#SalesRound" = "A#Sales"
```

```
HS.NoRound
```

```
HS.Exp "A#SalesNoRound" = "A#Sales"
```

NumBase

現在のメンバーまたは指定したメンバーの基本メンバーの数を取得します。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結
- 配賦

ノート:

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)
```

 ノート:

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 11-37 NumBase 関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> Account Custom1...4 Custom (カスタム・ディメンション・ラベル) Entity Parent Node
Element	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> Account Custom1...4 カスタム 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>ディメンション全体の基本メンバーの数を取得するには、次の例のように、ALL を引用符で囲んで指定します。</p> <pre>iAcctBase = HS.Account.NumBase("ALL")</pre>
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。
Entity	エンティティ・ディメンションの有効なメンバーの名前。

Exp 関数に NumBase 関数を組み込むこともできます。NumBase 関数を組み込む場合は、次の例のように、NumBase の引数を引用符で囲まないでください。

```
HS.Exp "A#AverageSales = A#Sales/HS.Entity.NumBase(Regional) "
```

戻り値

基本メンバーの数を識別する LONG。NODE の場合は、指定したメンバーのアクティブな基本要素の数を取得します。

 ノート:

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"A#SalesAlloc = A#Sales/  
HS.Custom("Prod").NumBase("TotalProducts")
```

```
...
```

```
End If
```

Number

現在の期間の番号を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.View.PeriodNumber
```

戻り値

現在の期間の番号。

例

この例では、現在の期間が最初の期間であると、**If...Then** と **End If** の文の間にあるすべての文が実行されます。

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

```
...
```

```
End If
```

NumChild

現在のディメンション・メンバーまたは指定したメンバーの子メンバーの数を取得します。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結
- 配賦

ノート:

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)
```

ノート:

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 11-38 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・ディメンション・ラベル) Entity Node Parent
Element	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> Account カスタム 1...4 カスタム Entity Node Parent Account オブジェクトと Custom オブジェクトを使用する場合は、メンバーを指定する必要があります。空白の文字列は使用できません。ディメンション全体の子メンバーの数を取得するには、次の例のように、ALL を引用符で囲って指定します。 iAcctBase = HS.Account.NumChild("ALL")
Scenario	シナリオ・ディメンションの有効なメンバーの名前。
Year	有効な年。
Period	有効な期間。
Entity	エンティティ・ディメンションの有効なメンバーの名前。

NumChild 関数を Exp 関数に組み込むことができます。NumChild 関数を組み込む場合は、NumChild の引数を引用符で囲わないでください。

戻り値

子メンバーの数を識別する文字列。NODE の場合は、指定したメンバーのアクティブな子の数を取得します。

例

この例では、Regional エンティティ直下にエンティティの平均売上高を保管する SalesChild という名前の勘定科目がアプリケーションに含まれています。SalesChild

の金額を計算するために、Sales 勘定の金額を、Regional 直下の子の数で割っています。

```
HS.Exp "A#SalesChild = A#Sales/HS.Entity.NumChild(Regional)"
```

NumCustom

アプリケーションに定義されているカスタム・ディメンションの合計数を戻します。たとえば、5つのカスタム・ディメンションを作成して、1つのカスタム・ディメンションを削除し、2つのカスタム・ディメンションを追加する場合、この関数で戻される値は、アプリケーションに定義されているカスタム・ディメンションの合計数である6です。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.NumCustom
```

戻り値

アプリケーションのカスタム・ディメンションの数。

例

```
nCustoms = HS.NumCustom
```

NumDescendant

現在のディメンション・メンバーまたは指定したメンバーの子孫の数を取得します。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結
- 配賦

ノート:

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)
```

ノート:

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 11-39 NumDescendant 関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> Account Custom1...4 Custom (カスタム・ディメンション・ラベル) Entity Node Parent

表 11-39 (続き) NumDescendant 関数の構文

パラメータ	説明
<i>Element</i>	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> Account Custom1...4 カスタム Entity Node Parent Account オブジェクトと Custom オブジェクトを使用する場合は、メンバーを指定する必要があります。空白の文字列は使用できません。ディメンション全体の子孫の数を取得するには、次の例のように、ALL を引用符で囲って指定します。 iAcctBase = HS.Account.NumDescendant ("ALL")
<i>Scenario</i>	シナリオ・ディメンションの有効なメンバーの名前。
<i>Year</i>	有効な年。
<i>Period</i>	有効な期間。
<i>Entity</i>	エンティティ・ディメンションの有効なメンバーの名前。

戻り値

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。

 **ノート:**

1 つの分岐に子孫エンティティが 2 回表示される場合は、2 個あるとみなされます。

例

この例では、エンティティ **France** に子孫がないと、**If...Then** 文と **End If** 文間にあるすべての文が実行されます。

```
If HS.Entity.NumDescendant("France") = 0 Then
```

```
...
```

```
End If
```

```
If HS.Custom("Product").NumDescendant("P3000-Phones") = 0 Then
```

```
...
```

```
End If
```

NumPerInGen

処理中の現在の期間に対する世代の期間の数を取得します。この関数は、**Dynamic SUB** 関数で使用できます。

構文

```
HS.Period.NumPerInGen
```

戻り値

ビューの期間の数を表す値。

例

```
HS.Dynamic "A#MarginPct=A#GrossMargin/HS.Period.NumPerInGen"
```

たとえば、現在の期間が 4 月で、4 月がカレンダー・ファイルの 4 番目の世代にある場合(月ごとの世代)は、月ごとの世代の期間の数は 12 です。現在の期間が、カレンダー・ファイルの 3 番目の世代にある Q2 である場合(四半期世代)は、期間の数は 4 です。

月ごとの世代(4 番目の世代):

1 月、2 月、3 月、4 月、5 月、6 月、7 月、8 月、9 月、10 月、11 月、12 月

この世代の期間の数として 12 が戻されます。

四半期世代(3 番目の世代):

Q1、Q2、Q3、Q4

この世代の期間の数として 4 が戻されます。

半期世代(2 番目の世代):

HY1、HY2

この世代の期間の数として 2 が戻されます。

年次世代(1 番目の世代):

年

この世代の期間の数として 1 が戻されます。

NumPeriods

指定したシナリオの期間単位に対して定義されている期間の数を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Scenario.NumPeriods("ScenarioName")
```

または

```
HS.Scenario.NumPeriods(" ")
```

または

```
HS.Scenario.NumPeriods(Var1)
```

戻り値

期間単位に対する期間の数を表す数値。たとえば、シナリオが月次の場合は、期間の数として 12 が戻されます。シナリオが四半期ごとの場合は、期間の数として 4 が戻されます。

例

この例では、実績シナリオの期間単位に対して定義されている期間の数が戻されます。

```
HS.Scenario.NumPeriods("Actual")
```

OpenCellTextUnit

複数のセルのセル・テキストを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

セル・テキストを取得する **POV** の一部を指定できます(例: **POV** 全体ではなく、シナリオと年)。ページ・ディメンション・メンバー(シナリオ、年、期間、値、エンティティ)を指定しない場合、現在のメンバーのセル・テキストのみが戻されます。サブキューブ・ディメンション(勘定科目、**ICP**、カスタム)を指定しない場合、すべての基本および親メンバーのセル・テキストが戻されます。シナリオおよび年のメンバーを指定して、現在のサブキューブではないサブキューブのセル・テキストを取得できます。期間、値および/またはエンティティのメンバー・リストを指定して、複数のサブキューブのセル・テキストを取得することもできます。セル・テキスト・ラベルを取得するよう指定し、ディメンション・メンバーまたはセル・テキスト・ラベルの昇順または降順で、セル・テキストをソートするかどうかを指定することもできます。ソートの値を指定しない場合、セル・テキスト単位でソートが実行されず、データベース内の順序に基づいて結果が戻されます。

構文

```
Set CTU =  
HS.OpenCellTextUnit(POVExpression,Label(s),Sort_Dimension,Sort_Order)
```

POVExpression は **POV**、*Label* はセル・テキスト・ラベル(なし、1 つまたは複数)、*Sort_Order* は昇順または降順です。

戻り値

指定した **POV** のすべてのセル・テキストおよびラベルの情報が戻されます。

例

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014","", "", "")
```

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014.P{.[Base]}.A{.  
[Base]}", "", "Period", "Ascending")
```

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014.E{.[Base]}.P{.  
[Base]}", "CouponRate", "LABEL", "Ascending")
```

```
Set CTU = HS.OpenCellTextUnit("S#Actual.Y#2014.E{.[Base]}.P{.  
[Base]}.A#Purchases", "", "Brands", "Descending")
```

複数のラベルのセル・テキストを取得する場合は、カンマ区切り値としてラベルを入力できます。ラベルの値を指定しない場合、すべてのラベルが取得されます。

例

```
Set CTU =  
HS.OpenCellTextUnit("A#Sales", "Label-1,Label-2", "Label", "Descending")
```

OpenDataUnit

連結、計算または換算時に処理するデータ単位を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

構文

```
HS.OpenDataUnit(POVExpression)
```

ここで、*POVExpression* は POV です。POV 式の一部として、関数では Account、ICP、C1、C2、C3 および C4 のユーザー定義リストとシステム・リストがサポートされています。

戻り値

この関数を **Sub Consolidate** サブルーチンで使用すると、データのあるすべてのレコードが戻されますが、勘定科目は連結勘定として指定されている勘定科目のみが戻されます。

Sub Calculate または **Sub Translate** サブルーチンで使用すると、連結勘定として設定されている勘定科目を含め、データのあるすべてのレコードが戻されます。

ノート:

勘定科目は、`IsConsolidated` 属性の値が `TRUE` である場合に連結されます。

例

```
Set DataUnit=HS.OpenDataUnit{"A{TotalRev.[Base]}.C1{C1Top.[Base]}.C2{MyC2List}.C3#[None]"}
```

OpenDataUnitSorted

計算、換算または連結時に処理するデータ単位を、指定した順序にデータをソートして取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結

構文

```
HS.OpenDataUnitSorted(POVExpression, <dimension to be sorted>, Ascending or Descending)
```

ここで、*POVExpression* は POV 文字列、*<dimension to be sorted>* はディメンション名の文字列であり、次の 6 つのディメンションのいずれかになります：

"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("")
```



ノート:

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 11-40 PCon 関数の構文

パラメータ	説明
シナリオ	シナリオ・ディメンションの有効なメンバーの名前。
年	有効な年。
期間	有効な期間。
エンティティ	エンティティ・ディメンションの有効なメンバーの名前。

戻り値

メンバーの連結比率。

例

この例では、指定した視点の連結比率が取得されます。

```
Pcon = HS.Node.PCon("S#Actual.Y#2014.P#January.E#Regional.Connecticut")
```

PEPU

エクイティ・ピックアップ(EPU)表から出資比率を取得します。この関数は、エクイティ・ピックアップ・ルールで使用されます。

構文

```
HS.PEPU("S#.Y#.P#",Owner,Owned)
```

戻り値

EPU 表の出資比率。

例

```
HS.PEPU(S#Actual.Y#2014.P#Jan, Group, CT)
```

または

```
""
```

```
HS.PEPU("", "", "")
```

デフォルト・パラメータ: 値が空白の場合、この関数は現在のシナリオ、年、期間のエンティティの組合せの出資比率を戻します。

例

```
Sub EquityPickUp()
```

```
Owned=Hs.Entity.Owned
```

```
OwnerCurrencyTotl=Hs.Entity.DefCurrency & "Total"
```

```
Hs.Clear "A#Inv.C4#EPU.I#" & Owned
```

```
Hs.Exp "A#Inv.C4#EPU.I#" & Owned & "=A#EQ.C4#C3Tot.I#[ICPTot].E#" &
```

```
Owned & ".V#" & OwnerCurrencyTotl & "*" & Hs.PEPU (,,)
```

```
End Sub
```

PeriodNumber

取得中のデータのビューの期間番号を取得します。この関数は、Dynamic SUB 関数で使用できます。

構文

```
HS.View.PeriodNumber
```

戻り値

ビューの期間の数を表す値。

例

```
Hs.Dynamic "A#MarginPct=A#GrossMargin/HS.View.PeriodNumber"
```

期間	期別	YTD	QTD	HYTD
1月	1	1	1	1
2月	1	2	2	2
3月	1	3	3	3
Q1	1	1	1	1
4月	1	4	1	4
5月	1	5	2	5
6月	1	6	3	6
Q2	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
Q3	1	3	1	1
10月	1	10	1	4
11月	1	11	2	5
12月	1	12	3	6
Q4	1	4	1	2
HY2	1	2	2	1
年	1	1	1	1

PlugAcct

現在の勘定科目メンバーまたは指定した勘定科目の調整勘定科目を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Account.PlugAcct ("Account")
```

```
HS.Account.PlugAcct ("")
```

Account は、有効な勘定科目ディメンション・メンバーの名前です。

ノート:

Sub Consolidate サブルーチンで関数を使用している場合にのみ、空白の文字列("")を使用して、この関数を現在のメンバーに適用できます。

戻り値

メンバーの調整勘定科目の名前を表す文字列。

例

この例では、**Sales** 勘定の調整勘定科目が **Plug1** であると、**If...Then** 文と **End If** 文間にあるすべての文が実行されます。

```
If HS.Account.PlugAcct ("Sales") = "Plug1" Then
```

```
...
```

```
End If
```

POwn

現在のメンバーまたは指定したメンバーの最大出資比率を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Node.POwn ("S#Scenario.Y#Year.P#Period.E#Entity")
```

```
HS.Node.POwn ("")
```

 ノート:

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 11-41 POwn 関数の構文

パラメータ	説明
<i>Scenario</i>	シナリオ・ディメンションの有効なメンバーの名前。
<i>Year</i>	有効な年。
<i>Period</i>	有効な期間。
<i>Entity</i>	エンティティ・ディメンションの有効なメンバーの名前。

戻り値

メンバーの出資比率。

例

この例では、指定した視点の出資比率が取得されます。

```
POwn = HS.Node.POwn("S#Actual.Y#2014.P#January. E#Regional.Connecticut")
```

PVAForBalance

BALANCE 勘定科目(ASSET および LIABILITY 勘定科目)のデフォルト換算メソッドを指定します。この関数は、換算中に通貨換算のアプリケーション・デフォルトを上書きします。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.AppSettings.PVAForBalance
```

戻り値

BALANCE 勘定が期別値(PVA)換算メソッドを使用する場合に TRUE となり、為替レート値 (VAL)換算メソッドを使用する場合に FALSE となるブール式。

例

この例では、アプリケーションの **BALANCE** 勘定で期別値換算メソッドが使用されると、**If...Then** 文と **End If** 文間にあるすべての文が実行されます。

```
If HS.AppSettings.PVAForBalance = TRUE Then
```

```
...
```

```
End If
```

PVAForFlow

FLOW 勘定(**REVENUE** および **EXPENSE** 勘定)のデフォルト換算メソッドを指定します。この関数は、換算中に通貨換算のアプリケーション・デフォルトを上書きします。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.AppSettings.PVAForFlow
```

戻り値

FLOW 勘定が期別値(**PVA**)換算メソッドを使用する場合に **TRUE** となり、為替レート値(**VAL**)換算メソッドを使用する場合に **FALSE** となるブール式。

例

この例では、アプリケーションの **FLOW** 勘定科目で為替レート換算メソッドの値が使用されると、**If...Then** と **End If** の間にあるすべての文が実行されます。

```
If HS.AppSettings.PVAForFlow = FALSE Then
```

```
...
```

```
End If
```

RateForBalance

BALANCE 勘定(ASSET および LIABILITY 勘定科目)で使用するデフォルト換算レート勘定科目を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.AppSettings.RateForBalance
```

戻り値

BALANCE 勘定で使用するデフォルトの換算レートが含まれているレート勘定科目を表す文字列。

例

この例では、BALANCE 勘定のデフォルト換算レート勘定科目が **Rate1** であると、If...Then 文と End If 文間にあるすべての文が実行されます。

```
If HS.AppSettings.RateForBalance = "Rate1" Then
```

```
...
```

```
End If
```

RateForFlow

FLOW 勘定(REVENUE および EXPENSE 勘定)で使用するデフォルト換算レート勘定を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.AppSettings.RateForFlow
```

戻り値

FLOW 勘定で使用するデフォルトの換算レートが含まれているレート勘定科目を表す文字列。

例

この例では、FLOW 勘定のデフォルト換算レート勘定がレート 2 であると、If...Then 文と End If 文間にあるすべての文が実行されます。

```
If HS.AppSettings.RateForFlow = "Rate2" Then
```

```
...
```

```
End If
```

ReviewStatus

指定した視点の確認ステータスを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.ReviewStatus("S#Scenario.Y#Year.P#Period.E#Entity.V#Value")
```

```
HS.ReviewStatus("")
```



ノート:

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 11-42 ReviewStatus 関数の構文

パラメータ	説明
シナリオ	シナリオ・ディメンションの有効なメンバーの名前。
年	有効な年
期間	有効な期間。

表 11-42 (続き) ReviewStatus 関数の構文

パラメータ	説明
エンティティ	エンティティ・ディメンションの有効なメンバーの名前。
値	値ディメンションのメンバーの名前。

戻り値

メンバーの確認ステータスを示す文字列。有効な確認ステータスは次のとおりです。

- 未開始
- 第 1 パス
- 確認レベル 1-10
- 送信済
- 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「送信済」であると、If...Then 文と End If 文間にあるすべての文が実行されます。

```
If HS.ReviewStatus("") = "Submitted" Then
```

```
...
```

```
End If
```

ReviewStatusUsingPhaseID

プロセス管理の提出フェーズ ID を使用して、指定した視点の確認ステータスを取得します。

構文

```
HS.ReviewStatusUsingPhaseID("S#Scenario.Y#Year.P#Period.E#Entity", n)
```

n は、プロセス管理の提出フェーズを表す整数です。有効な値は、1 から 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。

- 未開始

- 第 1 パス
- 確認レベル 1-10
- 送信済
- 承認済
- 発行済
- サポートされていません

例

```
HS.ReviewStatusUsingPhaseID("S#Actual.Y#2014.P#January.E#Connecticut",3  
)
```

Round

Exp 関数から戻されたデータを丸めます。丸め処理の位置を引数で指定します。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算
- 連結

ヒント:

計算ルールで、状況に応じて丸め処理の位置を変える必要がある場合は、**Round** を含む複数のステートメントを使用できます。

構文

```
HS.Round(Unit)
```

Unit は丸め処理の係数です。値 **1** は、最も近い整数に丸めます。値 **0.1** は、最も近い 10 分の 1 の値に丸めます。値 **0** は丸め処理をオフにします。

この引数に **0** を指定すると、計算ルール内の後続のすべての **Exp** 関数に対して丸め処理がオフになります。この構文は、**HS.NoRound: HS.Round(0)** と同じ効果を持ちます。

注意:

勘定科目の **NumDecimalPlaces** 属性によって、小数点の右側の最大桁数が決まります。**Round** 関数は、この属性を上書きしません。

戻り値

なし。

例

この例では、**SalesRound** 勘定のセルに挿入される金額を小数第 2 位で丸め、次に **NoRound** を使用して、**SalesNoRound** 勘定のセルに挿入される金額の丸め処理をオフにします。

```
HS.Round(0.1)
```

```
HS.Exp"A#SalesRound = A#Sales"
```

```
HS.NoRound
```

```
HS.Exp"A#SalesNoRound = A#Sales"
```

Scale

指定した通貨の位取りを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Currency.Scale("Currency")
```

```
HS.Currency.Scale(Var1)
```

表 11-43 Scale 関数の構文

パラメータ	説明
<i>Currency</i>	有効な通貨の名前。
<i>Var1</i>	VisualBasic 変数。

戻り値

指定した通貨の位取りを示す数字(0-9)。小数点の位置を指定して、通貨の金額を表示および保管するときの単位を指定します。戻り値は次のとおりです。

- 0 = 一
- 1 = 十
- 2 = 百
- 3 = 千

- 4 = 1 万
- 5 = 10 万
- 6 = 百万
- 7 = 1 千万
- 8 = 1 億
- 9 = 十億

例

この例では、フランス・フラン(**FF**)の位取りが **3** であると、**If...Then** 文と **End If** 文間にあるすべての文が実行されます。

```
If HS.Currency.Scale("FF") = 3 Then
```

```
...
```

```
End If
```

SecurityAsPartner

エンティティまたは親が内部取引パートナーとして使用される場合に、指定されたエンティティまたは親に割り当てられるセキュリティ・クラスを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Entity.SecurityAsPartner("Entity")
```

```
HS.Entity.SecurityAsPartner("")
```

```
HS.Entity.SecurityAsPartner(Var1)
```

```
HS.Parent.SecurityAsPartner("Entity")
```

 ノート:

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 11-44 SecurityAsPartner 関数の構文

パラメータ	説明
<i>Entity</i>	エンティティ・ディメンションの有効なメンバーの名前。
<i>Var1</i>	VisualBasic 変数。

戻り値

ICP として使用する場合に、エンティティまたは親に割り当てられているセキュリティ・クラスを示す文字列。

例

この例では、内部取引パートナーとして使用されるとき **Class1** が **France** のセキュリティ・クラスである場合に、**If...Then** と **End If** の文間にある文が実行されます。

```
If HS.Entity.SecurityAsPartner("France") = "Class1" Then
```

```
...
```

```
End If
```

```
If HS.Parent.SecurityAsPartner("France") = "Class1" Then
```

```
...
```

```
End If
```

SecurityClass

指定されたメンバーのセキュリティ・クラスを取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.<Object>.SecurityClass("Element")
```

```
HS.<Object>.SecurityClass("")
```

```
HS.<Object>.SecurityClass(Var1)
```

```
HS.Custom(Dimension).SecurityClass("Element")
```

ノート:

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 11-45 SecurityClass 関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> • Account • Scenario • Entity • Parent • Custom1...4 • Custom (カスタム・ディメンション・ラベル)
Element	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> • Account • Scenario • Entity • 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** の文間にある文が実行されます。

```
If HS.Account.SecurityClass("Cash") = "Class1" Then
```

```
...
```

```
End If
```

```
If HS.Custom(Dimension).SecurityClass(Member) Then
```

```
...
```

```
End If
```

SetCellTextWithLabel

指定された **POV** およびセル・テキスト・ラベルのセル・テキスト情報を更新します。この関数は、次のタイプのルールで使用できます：

計算

構文

```
HS.SetCellTextWithLabel("POVExpression", "CellTextLabel", "CellTextString")
```

POVExpression はメンバーの組合せで、**CellTextLabel** はデフォルトのセル・テキスト・ラベル(**[Default]**)か、ロードされた有効なセル・テキスト・ラベルの 1 つです。**CellTextString** は、書き込まれるテキスト文字列です。**Account**、**ICP** および **Custom POV** メンバーを指定する必要があります。シナリオ、年、期間、ビュー、エンティティおよび値は、デフォルトで現在のメンバーに設定されます。

戻り値

なし

例

```
HS.SetCellTextWithLabel "A#Asset.I#[ICP  
None].C1#[None].C2#[None].C3#[None].C4#[None]", "Rating", "AAA"
```

SetData

個々のレコードを設定します。この関数は、次のタイプのルールで使用できます：

- 計算
- 換算

構文

```
HS.SetData lView, lAccount, lICP, lCustom1, lCustom2,
lCustom3, lCustom4, dData, bAddToExistingData
```

表 11-46 SetData 関数の構文

パラメータ	説明
<i>lView</i>	0 = シナリオ・ビュー 1 = 期別ビュー 2 = 年次累計ビュー
<i>lAccount</i>	データを設定する勘定科目の ID 番号。
<i>lICP</i>	データを設定する ICP の ID 番号。
<i>lCustom1...4</i>	データを設定するカスタム・ディメンションの ID 番号。
<i>dData</i>	設定するデータ値。
<i>bAddToExistingData</i>	TRUE = データを累積する FALSE = データを置換する

戻り値

なし。

例

```
HS.SetData 2, 002, , , , , 25000, TRUE
```

SetDataWithPOV

データをノードまたは通貨キューブに挿入します。この関数は、次のタイプのルールで使用できます。

- 計算
- 換算

構文

```
HS.SetDataWithPOV POV, dData, bAddToExistingDataInCache
```

表 11-47 SetData 関数の構文

パラメータ	説明
<i>POV</i>	有効な POV
<i>dData</i>	設定するデータ値。

表 11-47 (続き) SetData 関数の構文

パラメータ	説明
<i>bAddToExistingData</i>	TRUE = データを累積する FALSE = データを置換する

戻り値

なし。

例

```
HS.SetDataWithPOV "W#YTD.A#Asset.I#[ICP
None].C1#None.C2#None.C3#None.C4#None",25000,TRUE
```

SubmissionGroup

ディメンション・メンバーのプロセス管理の送信グループを取得します。

構文

```
HS.<Dimension>.SubmissionGroup(Dimension member)
```

ここで、*Dimension* は、次のいずれかのディメンションです: 勘定科目、カスタム 1...4 またはカスタム

```
HS.Account.SubmissionGroup(Account)
```

```
HS.Custom1.SubmissionGroup(Custom1)
```

```
HS.Custom2.SubmissionGroup(Custom2)
```

```
HS.Custom3.SubmissionGroup(Custom3)
```

```
HS.Custom4.SubmissionGroup(Custom4)
```

```
HS.Custom(Custom Dimension Label).SubmissionGroup((CustomDimensionLabel))
```

戻り値

ディメンション・メンバーの送信グループを表す整数。有効な値は、1 から 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("")
```



ノート:

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 11-48 SupportsProcessManagement 関数の構文

パラメータ	説明
<i>Scenario</i>	有効なシナリオ。
<i>Var1</i>	シナリオメンバーを表す VBScript 変数。

戻り値

シナリオでプロセス管理が使用可能な場合は **TRUE**、そうでない場合は **FALSE** となるブール値。

例

この例では、実績シナリオでプロセス管理が使用可能である場合に、If と End If の間にあるすべての文が実行されます。

```
If HS.Scenario.SupportsProcessManagement("Actual") = "TRUE" then
```

```
...
```

```
End IF
```

SupportsTran

内部取引をサポートするアプリケーションの勘定科目を指定します。この関数は、取引ルールのみで使用できます。

構文

```
HS.SupportsTran "POVExpression"
```

POVExpression は、勘定科目、カスタム 1-4、シナリオ、エンティティおよび年の各メンバーの組合せです。

戻り値

なし。

例

```
HS.SupportsTran "S#ActMon.A#RecltIC.C1#Closing"
```

SwitchSign

現在のカスタム・メンバーまたは指定されたカスタム・メンバーの貸方が借方に切り替えられているかどうかを調べます。この関数は、次のルールを使用して、貸方/借方の符号を逆にします。

- ASSET から LIABILITY へ
- LIABILITY から ASSET へ
- EXPENSE から REVENUE へ
- REVENUE から EXPENSE へ
- BALANCE から FLOW へ
- FLOW から BALANCE 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)
```

ノート:

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 11-49 SwitchSign 関数の構文

パラメータ	説明
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> • Custom1 • Custom2 • Custom3 • Custom4 • Custom
Member	カスタム・ディメンションの有効なメンバーの名前。

戻り値

カスタム・メンバーの貸方が借方に切り替わる場合に TRUE、そうでない場合に FALSE となるブール式。

例

この例では、貸方と借方が切り替わった場合に、If...Then と End If の文の間にあるすべての文が実行されます。

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

```
...
```

```
End If
```

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

```
...
```

```
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タイプが切り替えられているかどうかを調べます。この関数は、次のルールを使用して、カスタム・ディメンション・メンバーの勘定科目タイプを変更します。

- ASSET から EXPENSE へ
- EXPENSE から ASSET へ
- LIABILITY から REVENUE へ
- REVENUE から LIABILITY へ
- BALANCE から FLOW へ
- FLOW から BALANCE 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("")
```

 ノート:

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 11-50 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・ディメンションの有効なメンバーの名前。

戻り値

カスタム・メンバーの勘定科目タイプ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タイプが切り替わる場合に、If...Then と End If の文の間にあるすべての文が実行されます。

```
If HS.Custom1.SwitchType("") = TRUE Then
```

```
...
```

```
End If
```

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

```
...
```

```
End If
```

Trans

年次累計メソッドを使用して通貨を換算します。この関数は換算ルールで使用できません。

構文

```
HS.Trans("DestPOV","SourcePOV","Rate1","Rate2")
```

表 11-51 Trans 関数の構文

パラメータ	説明
<i>DestPOV</i>	宛先視点。宛先は、勘定科目、カスタム 1...4、ICP のメンバーの組合せを指定できます。指定されていない各ディメンションに対して、ディメンションの有効なすべてのメンバーに書き込まれます。指定された各ディメンションに対して、指定されたメンバーのみに書き込まれます。
<i>SourcePOV</i>	ソース視点。ソースはディメンションの組合せを指定できます。勘定科目、カスタム 1...4 および ICP のディメンションが指定されていない場合は、Destination メンバーと同じメンバーが使用されます。シナリオ、年、期間およびエンティティのディメンションが指定されていない場合は、現在のメンバーが使用されます。値ディメンションが指定されていない場合は、<EntityCurrTotal>メンバーが使用されます。ソース視点空白の場合は、ソース視点として宛先視点を使用されます。
<i>Rate1-2</i>	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、 デフォルトの換算方法 を参照してください。

戻り値

なし。

例

この例では、Rate1 勘定科目のレートを使用して、年次累計メソッドにより Sales 勘定科目が換算されます。

```
HS.Trans("A#Sales", "A#LastYearSales", "A#Rate1", "")
```

TransPeriodic

期別メソッドを使用して通貨を換算します。この関数は換算ルールで使用できます。

構文

```
HS.TransPeriodic("DestPOV","SourcePOV","Rate1","Rate2")
```

表 11-52 TransPeriodic 関数の構文

パラメータ	説明
<i>DestPOV</i>	宛先視点。宛先は、勘定科目、カスタム 1-4、ICP のメンバーの組合せを指定できます。指定されていない各ディメンションに対して、ディメンションの有効なすべてのメンバーに書き込まれます。指定された各ディメンションに対して、指定されたメンバーのみに書き込まれます。
<i>SourcePOV</i>	ソース視点。ソースはディメンションの組合せを指定できます。勘定科目、カスタム 1、カスタム 2、カスタム 3、カスタム 4 および ICP のディメンションが指定されていない場合は、 Destination メンバーと同じメンバーが使用されます。シナリオ、年、期間およびエンティティのディメンションが指定されていない場合は、現在のメンバーが使用されます。値が指定されていない場合は、 EntityCurrTotal メンバーが使用されます。ソースが空白の場合は、ソースとして宛先が使用されます。
<i>Rate1-2</i>	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、 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(strElement)
```

```
HS.<Object>.UD1...3("")
```

```
HS.Custom(Dimension).UD1(Member)
```

ノート:

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 11-53 UD1-3 関数の構文

Parameter	Description
<Object>	次のオブジェクトのキーワードのいずれか: <ul style="list-style-type: none"> Account Entity Parent Scenario Custom1...4 Custom
要素	次のいずれかのディメンションの有効なメンバーの名前。選択したオブジェクトによって異なります。 <ul style="list-style-type: none"> Account Entity Parent Scenario Custom1...4 Custom

戻り値

メンバー用に保存されたユーザー定義のテキストを含む文字列。

例

この例では、UD1 勘定科目のユーザー定義テキストが **History** である場合に、**If...Then** と **End If** の文間にある文が実行されます。

```
If HS.Account.UD1(strAccount) = "History" Then
```

```
...
```

```
End If
```

```
If HS.Custom("Product").UD1("P3000-Phones") = "History" Then
```

```
...
```

```
End If
```

ValidationAccount

アプリケーションの検証勘定を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.AppSettings.ValidationAccount
```

戻り値

アプリケーションの検証勘定の名前を示す文字列。

例

アプリケーションの検証勘定科目が **MyAccount** である場合に、**If...Then** と **End If** の文間にある文が実行されます。

```
If HS.AppSettings.ValidationAccount = "MyAccount" Then  
  
    ...  
  
End If
```

ValidationAccountEx

プロセス管理の提出フェーズの検証勘定科目を取得します。

構文

```
HS.AppSettings.ValidationAccountEx(n)
```

n は、プロセス管理の提出フェーズを表す整数です。有効な値は **1** から **9** です。

戻り値

プロセス管理の提出フェーズの検証勘定科目名を示す文字列。

例

この例では、提出フェーズ **5** に定義された検証勘定を戻します。

```
HS.AppSettings.ValidationAccountEx(5)
```

XBRLTags

指定された勘定科目メンバーに割り当てられた **XBRL** を取得します。この関数は、次のタイプのルールで使用できます:

- 計算
- 換算
- 連結
- 配賦

構文

```
HS.Account.XBRLTags("Account")
```

```
HS.Account.XBRLTags("")
```

 ノート:

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 11-54 XBRLTags 関数の構文

パラメータ	説明
勘定科目	有効な勘定科目。

戻り値

指定した勘定科目の XBRL タグを表す文字列。

12

カスタム関数

この項では、管理レポート関数およびプランニング関数用に Oracle Hyperion Financial Management で使用できる内部 HS カスタム関数を示します。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 12-1 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 で同等のもの
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	プロシージャ
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 12-2 プランニング関数

カスタム関数	説明	パラメータ	関数のタイプ
Units_Rates	単位 * レート (C=A*B)	Unit_Rates (Description, Units, Rates)	プロシージャ
Custom_Alloc	カスタム・ディメンションに配賦します	Custom_Alloc (Destination, Source, Factor, FactorN, FactorD, Elimination)	プロシージャ
Increase_Decrease	勘定科目をある比率で増減します	Increase_Decrease (Destination, Source, Factor, Scale, Inverse)	プロシージャ
Pro_Rata_Ratio	2つの勘定科目の比率	Pro_Rata_Ratio (Destination, SourceN, SourceD)	プロシージャ
Spread	合計金額を年のすべての期間に分散します	Spread (Destination, Source, Factor, FactorN, FactorD, Temp, Per)	プロシージャ

管理レポート関数

このセクションでは、使用可能な管理レポートのカスタム関数をリストします。

Average

多数の期間に渡る勘定科目の平均値を計算します。

戻り値

HS.EXP 関数の一部として使用される正しい式を表す文字列が戻されます。

構文

Average (*PointOfView*, *Periods*)

表 12-3 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、2014 年 1 月、2 月、3 月に次の値が戻されます。処理中のシナリオに設定されたデフォルト・ビューは YTD です。

表 12-4 Average 関数の例

勘定科目	Oct2013	Nov2013	Dec2013	Jan2014	Feb2014	Mar2014
A#Sales	9,000	10,500	11,700	800	1,900	3,200
Average ("A#Sales", "YTD")	N/A	N/A	N/A	800	950	1,067
Average ("A#Sales", "Periodic")	N/A	N/A	N/A	800	950	1,200
Average ("A#Sales", "3")	N/A	N/A	N/A	1,167	1,033	1,067

サンプル・スクリプト

```
' sample statement written in the calling routine
```

```
Sub Calculate()
```

```
  Hs.Exp "A#AVG_SALES = "& Average("A#Sales", "12")
```

```
End Sub
```

```
' programming of the AVERAGE function
```

```
FUNCTION Average(strPOV, strPERIOD)
```

```
  DIM nPERIOD
```

```
  DIM strCUM
```

```
  DIM i
```

```
  strPOV = UCASE(strPOV)
```

```
  strPERIOD = UCASE(strPERIOD)
```

```
  IF strPERIOD = "PERIODIC" THEN
```

```
    IF HS.PERIOD.ISFIRST = TRUE THEN
```

```
      nPERIOD = 1
```

```
    ELSE
```

```
      nPERIOD = 2
```

```
    END IF
```

```
  ELSEIF strPERIOD = "YTD" THEN
```

```
    nPERIOD = HS.PERIOD.NUMBER()
```

```
  ELSEIF CINT(strPERIOD) > 0 THEN
```

```
    nPERIOD = CINT(strPERIOD)
```

```
  ELSE
```

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 12-5 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
ビュー	次のいずれかの値にする必要があります: "" "" (二重引用符) - 処理中のシナリオに定義されているデフォルト・ビューに基づきます(YTD または Periodic)。 YTD - ユーザーにより年次累積オプションが指定され、シナリオに設定されたデフォルト・ビューが上書きされます。 Periodic - 期別オプションが指定され、シナリオに設定されたデフォルト・ビューが上書きされます。
NumPeriod	累計する現在のシナリオ内の期間数を表す整数で、現在の期間から開始されます。 NumPeriod は 0 または負数で、関数は現在の年の初めから集計します。

詳細説明

この関数は、指定された期間の合計、または指定された勘定科目の年次累計の合計を計算します。デフォルトでは、累積されたデータのビューはシナリオのデフォルトです;ただし、フロー・タイプの勘定科目の場合は上書きできます。

- View パラメータが YTD の場合、この関数では年次累計値が累計されます。
- View パラメータが Periodic の場合は、期間の値が累計されます。
- View パラメータが空白("")の場合は、シナリオのデフォルト・ビューを使用してデータが累計されます。

例

CASH 勘定科目では、Cumulative 関数で使用する Number パラメータに基づいて、2014 年 1 月、2 月、3 月に次の値が戻されます。

SALES 勘定科目では、Cumulative 関数で使用する View および Number パラメータの両方に基づいて、2014 年 1 月、2 月、3 月に次の値が戻されます。処理中のシナリオに設定されたデフォルト・ビューは YTD です。

表 12-6 Cumulative 関数の例

勘定科目	Oct2013	Nov2013	Dec2013	Jan2014	Feb2014	Mar2014
A#Cash	1,000	1,500	1,200	800	1,100	1,300
Cumulative("A#Cash","0")	N/A	N/A	N/A	800	1,900	3,200
Cumulative("A#Cash","3")	N/A	N/A	N/A	3,500	3,100	3,200
A#Sales	9,000	10,500	11,700	800	1,900	3,200
Cumulative("A#Sales","0")	N/A	N/A	N/A	800	2,700	5,900
Cumulative("A#Sales","Periodic","0")	N/A	N/A	N/A	800	1,900	3,200
Cumulative("A#Sales","Periodic","3")	N/A	N/A	N/A	3,500	3,100	3,200

サンプル・スクリプト

```
' sample statement written in the calling routine
```

```
Sub Calculate()
```

```
HS.EXP "A#TOT_Cash =" &Cumulative("A#Cash","" ,0)
```

```
End Sub
```

```
' programming of the Cumulative function
```

```
Function Cumulative(StrPov, StrVIEW, nPERIOD)
```

```
DIM strCUM
```

```
DIM i
```

```
IF nPERIOD <= 0 THEN
```

```
nPERIOD = HS.PERIOD.NUMBER() - 1
```

```
ELSE
```

```
nPERIOD = nPERIOD - 1
```

```
END IF
```

```
IF strVIEW = "" THEN
```

```
strVIEW = HS.SCENARIO.DEFAULTVIEW("")
```

```
END IF
```

```
strPOV = UCASE(strPOV)
```

```
strVIEW = UCASE(strVIEW)
```

```
IF strVIEW = "PERIODIC" THEN
```

```
strVIEW = ".W#PERIODIC"
```

```
ELSEIF strVIEW = "YTD" THEN
```

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 12-7 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
ビュー	次のいずれかの値にする必要があります: "" (二重引用符) - 処理中のシナリオに定義されているデフォルト・ビューに基づきます(YTD または Periodic)。 YTD - 年次累積オプションが指定され、シナリオに設定されたデフォルト・ビューが上書きされます。 Periodic - 期別オプションが指定され、シナリオに設定されたデフォルト・ビューが上書きされます。

詳細説明

この関数は、現在の期間の値と開始値の差を計算します。(現在 - 開始時)

開始値の導出方法は、関数に渡される *View* パラメータによって異なります。

- *View* パラメータが YTD の場合、開始値は、前年の最後の期間から取得されます。
- *View* パラメータが Periodic の場合、開始値は、現在の年の前の期間から取得されます。現在の期間が年の最初の期間である場合、開始値は、前年の最後の期間から取得されます。
- *View* パラメータが空白("")の場合、開始値は、シナリオのデフォルトのデータ・ビューに基づきます。

例

CASH 勘定科目では、Difference 関数で使用される *View* パラメータに基づいて、2014 年 1 月、2 月、3 月に次の値が戻されます。処理中のシナリオに設定されたデフォルト・ビューは YTD です。Difference 関数は、現在の期間の値から開始値を減算します。

表 12-8 Difference 関数の例

勘定科目	Dec2013	Jan2014	Feb2014	Mar2014
A#Cash	900	1,200	1,100	1,500
Difference("A#Cash" N/A , ""		300	200	600
Difference("A#Cash" N/A , "YTD")		300	200	600
Difference("A#Cash" N/A , "Periodic")		300	-100	400

サンプル・スクリプト

```
' sample statement written in the calling routine

Sub Calculate()

Hs.Exp = "A#DiffCash" & Difference("A#Cash", "YTD")

End Sub

' programming of the DIFFERENCE function

FUNCTION DIFFERENCE(strPOV,strVIEW)

IF strVIEW = "" THEN

strVIEW = HS.SCENARIO.DEFAULTVIEW ("" )

END IF

strPOV = UCASE(strPOV)
strVIEW = UCASE(strVIEW)

IF strVIEW = "PERIODIC" THEN

DIFFERENCE = ""&" "&strPOV"-& strPOV & ".P#PRIOR" &"")""

ELSEIF strVIEW = "YTD" THEN

DIFFERENCE = "" ("&strPOV &"-" & strPOV & ".Y#PRIOR.P#LAST" &"") ""

ELSE

EXIT FUNCTION

END IF

END FUNCTION
```

DSO - 売上債権回転日数

全数法を使用して、当期借方の売上回収日数が計算されます。

戻り値

このルーチンは、当期取引の借方内における売上回収日数を表す単一の値を計算します。含まれる DSO サブルーチンでは、次のことが仮定されています：

- 借方も売上も正数です。
- ルーチンでは **HS.GETCELL** 関数を使用されるため、指定されるパラメータは完全に定義された視点(勘定科目/C1/C2/C3/C4/ICP など)である。
- ルーチンでは、可能なかぎりさかのぼって日数が計算されます。ただし、任意の期間の期別売上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 12-9 DSO 関数の構文

パラメータ	有効な値
strDSO	カスタム・ディメンションと内部取引ディメンションを含む完全に定義された勘定科目。この勘定科目は計算の宛先です。
strDEBTOR	カスタム・ディメンションと内部取引ディメンションを含む完全に定義された勘定科目。この勘定科目は当期取引の借方のソースです。
strSALES	カスタム・ディメンションと内部取引ディメンションを含む完全に定義された勘定科目。この勘定科目は売上のソースです。 特に、期間単位への参照は除外されます。
strDIP	カスタム・ディメンションと内部取引ディメンションを含む完全に定義された勘定科目。この勘定科目はその期間の日数のソースです。 これは、【なし】エンティティであるとみなされません。

詳細説明

このルーチンは、現在の期間の **Debtors** 勘定科目(パラメータ 2)および **Sales** 勘定科目(パラメータ 3)の値を取得し、それらを比較します。どちらかがゼロまたは負数の場合は計算が止まります。借方の値が累積売上の値を超える連続した各期間(現在の期間からさかのぼる)では、ルーチンにより、**Period** 勘定科目(パラメータ 4)の日数に指定されているとおりに、その期間の日数が累計に追加されます。

このようにして、すべての借方の値が全数法で処理されると、最後の期間の日数は、期別売上値に対する失効前借方の一部として計算されます。

こうして、累計が宛先の勘定科目(パラメータ 1)に転記されます。

例

この例では、表示されている月の合計未処理日数を計算しています。

表 12-10 DSO 関数の例

Month	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 12-11 Opening 関数の構文

パラメータ	有効な値
PointOfView	A#CLOSE.C1#[None].I#[ICP Top]など、勘定科目、カスタム 1 から 4、ICP メンバーの有効な組合せ
View	次のいずれかの値にする必要があります: "" (二重引用符) - 処理中のシナリオに定義されているデフォルト・ビューに基づきます(YTD または Periodic)。 YTD - 年次累積オプションが指定され、シナリオに設定されたデフォルト・ビューが上書きされます。 Periodic - 期別オプションが指定され、シナリオに設定されたデフォルト・ビューが上書きされます。

詳細説明

この関数は、指定された勘定科目の開始値を計算します。開始値の導出方法は、**View** パラメータによって異なります。

- **View** パラメータが **YTD** の場合、開始値は、前年の最後の期間から取得されます。
- **View** パラメータが **Periodic** の場合、開始値は、現在の年の前の期間から取得されます。現在の期間が年の最初の期間である場合、開始値は、前年の最後の期間から取得されます。
- **View** パラメータが空白("")の場合、開始値は、シナリオのデフォルトのデータ・ビューに基づきます。

例

FA_COST 勘定科目では、**Opening** 関数で使用される **View** パラメータに基づいて、2014 年 1 月、2 月、3 月に次の値が戻されます。処理中のシナリオに設定されたデフォルト・ビューは **YTD** です。

表 12-12 Opening 関数の例

勘定科目	Dec2013	Jan2014	Feb2014	Mar2014
A#FA_COST	900	1,200	1,100	1,500
Opening("A#FA_COST N/A T", "")		900	900	900
Opening("A#FA_COST N/A T", "YTD")		900	900	900
Opening("A#FA_COST N/A T", "Periodic")		900	1,200	1,100

サンプル・スクリプト

```
' sample statement written in the calling routine

Sub Calculate()

Hs.Exp "A#Open_FA_Cost = " & Opening("A#FA_Cost", "YTD")

End Sub

' programming of the OPENING function

FUNCTION OPENING(strPOV,strVIEW)

IF strVIEW = "" THEN

strVIEW = HS.SCENARIO.DEFAULTVIEW ( "")

END IF

strPOV = UCASE(strPOV)
strVIEW = UCASE(strVIEW)

IF strVIEW = "PERIODIC" THEN

OPENING = strPOV & ".P#PRIOR"

ELSEIF strVIEW = "YTD" THEN

OPENING = strPOV & ".Y#PRIOR.P#LAST"

ELSE

EXIT FUNCTION

END IF

END FUNCTION
```

Rate

親と子の間で関連する為替レートを計算し、値を乗数として戻します。

戻り値

この関数は、通常、換算セクションで **HS.EXP** 関数の一部として使用される値を戻します。

構文

Rate (*ExchangeRate*, *TriangulationCurrency*)

表 12-13 Rate 関数の構文

パラメータ	有効な値
ExchangeRate	勘定科目文字列として指定された CurrencyRate タイプのメイン勘定科目で、 A#EOP_RATE などのカスタム・ディメンションや内部取引ディメンションへの参照は含まれません
TriangulationCurrency	文字列または二重引用符(" ")のいずれかの有効な通貨ラベル。通貨を指定する際は、カスタム・ディメンションを参照する必要はありません。

詳細説明

- この関数は、親と子の間で関連する為替レートを計算し、値を乗数として戻します。この値は、*TriangulationCurrency* パラメータに基づいて計算されます。
- TriangulationCurrency* パラメータが有効な通貨ラベルの場合、クロス・レートはこの通貨に基づきます。
- TriangulationCurrency* パラメータが空白(" ")の場合は有効な直接レートが検索され、検出されないときには、アプリケーションの通貨に対してトライアングレーションが使用されます。
- レート値が見つからない場合は、関数により **1** が戻されます。

後続の表には、データの検索方法と、その検索が行われる順序が示されています。順序は、(1)のように、カッコで囲んだ数値で表されています。いずれの場合も、検索は最初に子エンティティで実行され、データが見つからない場合に[None]エンティティで検索されます。

次の表では、子の通貨または親の通貨がトライアングレーション通貨と同一であり、トライアングレーションが空白の場合には、アプリケーションの通貨と同一です。

表 12-14 Rate の例 - トライアングレーション通貨と同一の場合

カスタム 2 ディメンションのレート		カスタム 1 ディメンションのレート	
		子	親
カスタム 2 ディメンシ ョンのレート	子		(2)
親	(1)		

次の表では、トライアン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 12-15 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 2 ディメンションのレート	カスタム 1 ディメンションのレート		トライアンギュレーション
	子	親	
カスタム 2 ディメンションのレート			(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 12-16 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 2 ディメンションのレート	カスタム 1 ディメンションのレート		アプリケーション
	子	親	
カスタム 2 ディメンションのレート		(2)	(4)
親 アプリケーション	(1)		(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 12-17 Rate 関数の例

Rate	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

```

```

sRATE,sCCUR,sPCUR,s3rdCUR)

END IF

END IF

END IF

END SELECT

IF bRET THEN

RATE = retValue

ELSE

RATE = 1

END IF

END FUNCTION

FUNCTION GETVALUECP (sENTITY,sVALUE,sRATE,sCCUR,sPCUR)

' this sub-function is used when comparing direct rates between child
and parent

GETVALUECP = FALSE

' check if data exists for direct rate child to parent. If it does
return it.
' if no direct child to parent rate check for indirect parent to child

```

```
rate...
' return the inverse of the indirect rate.

IF HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & sPCUR & sENTITY) <> 0 THEN

sVALUE = CDBL(HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & sPCUR & sENTITY))

GETVALUECP = TRUE

ELSEIF HS.GETCELL(sRATE & ".C1#" & sPCUR & ".C2#" & sCCUR & sENTITY) <> 0
THEN

sVALUE = CDBL(1 / HS.GETCELL(sRATE & ".C1#" & sPCUR & ".C2#" & sCCUR &
sENTITY))

GETVALUECP = TRUE

END IF

END FUNCTION

FUNCTION GETVALUE3(sENTITY,sVALUE,sRATE,sCCUR,sPCUR,s3rdCUR)

' this sub-function is used when triangulating
' check if data exists for direct rate child to triangulation...
' ... if it does return the direct relative rate child to parent...
' if no direct child to triangulation rate check for indirect triangulation
```

```
to child rate...
' ... return the inverse of the indirect relative rates.

GETVALUE3 = FALSE

IF HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & s3rdCUR & sENTITY) <>
0 THEN

sVALUE = CDBL(HS.GETCELL(sRATE & ".C1#" & sCCUR & ".C2#" & s3rdCUR &
sENTITY) / HS.GETCELL(sRATE & ".C1#" & sPCUR & ".C2#" & s3rdCUR &
sENTITY))

GETVALUE3 = TRUE

ELSEIF HS.GETCELL(sRATE & ".C1#" & s3rdCUR & ".C2#" & sCCUR & sENTITY)
<> 0 THEN

sVALUE = CDBL(HS.GETCELL(sRATE & ".C1#" & s3rdCUR & ".C2#" & sPCUR &
sENTITY) / HS.GETCELL(sRATE & ".C1#" & s3rdCUR & ".C2#" & sCCUR &
sENTITY))

GETVALUE3 = TRUE

END IF

END FUNCTION
```

ビジネス・ルール関数

このセクションでは、ビジネス・ルール・カスタム関数をリストします。

Custom_Alloc

この関数は、配賦の基本として係数 **POV** を使用し、ソースの視点(**POV**)を宛先 **POV** に割り当てます。また、消去 **POV** に対する合計配賦額の転記を逆にするオプションもあります。この関数は、カスタム・ディメンションの配賦用に設計されています。

戻り値

戻り値はありません。

構文

```
Custom_Alloc(Destination, Source, Factor, FactorN, FactorD,
Elimination)
```

表 12-18 Custom_Alloc 関数の構文

パラメータ	有効な値
Destination	勘定科目、ICP、カスタム 1 から 4 のメンバーの有効な組合せである有効な宛先 POV。
Source	ディメンション・メンバーの有効な組合せである有効なソース POV。Source は配賦される金額です。
Factor	有効なソース POV。Factor は、配賦係数の格納に使用される勘定科目です。
FactorN	有効なソース POV。FactorN は、配賦の基準として使用される分子係数です。
FactorD	有効なソース POV。FactorD は、配賦の基準として使用される分母係数です。
Elimination	有効なソース POV。Elimination は空の文字列("")にすることが可能で、その場合、このパラメータは無視されます。Elimination パラメータが設定されている場合、Destination POV に転記される金額には-1 が乗算され、消去 POV に転記されます。

詳細説明

この関数は、配賦の基本として係数 POV を使用し、ソースの POV を宛先 POV に割り当てます。また、消去 POV に対する合計配賦額の転記を逆にするオプションもあります。この関数は、カスタム・ディメンションの配賦用に設計されています。

Factor パラメータには、**FactorD** で除算された **FactorN** の結果が格納されます。これは、係数が、現在のエンティティ以外のエンティティを参照できるようにするために必要です。

ソース POV のエンティティが親である場合、子レベルで計算を実行する前に、その親を連結する必要があります。親の通貨が子の通貨と異なる場合、関連するすべての通貨の換算も、子レベルで計算を開始する前に実行する必要があります。

呼出しルーチンに変数を設定し、Destination、Source、Factor、FactorN、FactorD および Elimination POV を定義する Custom_Alloc に渡すことをお勧めします。また、呼出しルーチンの変数名を Custom_Alloc 関数と同一になるように設定することもお勧めします。

Elimination パラメータは空の文字列("")にすることが可能で、その場合、このパラメータは無視されます。**Elimination** パラメータが設定されている場合、宛先 POV に転記される金額には-1 が乗算され、消去 POV に転記されます。

例

Total Sales に対する Products Sales の比率に基づいて、Telephone 勘定科目が Products に配賦されます。配賦額の逆数が Allocations 勘定科目に転記されます。

表 12-19 Custom_Alloc 関数の例

勘定科目	Jan2014	Feb2014	Mar2014
A#Telephone.C1#[None]	100	300	400
A#Sales.C1#Product1	1000	1000	1000
A#Sales.C1#Product2	1000	2000	3000
A#Sales.C1#TotalProducts	2000	3000	4000
Custom_Alloc (A#Telepho ne", "A#Telephone.C1#[None]", "A#Factor", A#Sales", "A#Sales.C1#TotalProd ucts", "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#Product 1	50	100	100
A#Telephone.C1#Product 2	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文。
- 呼出しルーチンに設定され、Custom_Alloc 関数に渡される変数。
- Custom_Alloc 関数と同一になるように設定された呼出しルーチンの変数名。

```
Sub Calculate()
```

```
Dim Destination
```

```
Dim Source
```

```
Dim Elimination
```

```
Dim Factor  
Dim FactorN  
Dim FactorD  
Dim C1list  
Dim 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 12-20 Increase_Decrease 関数の構文

パラメータ	有効な値
Destination	勘定科目、ICP、カスタム 1 から 4 のメンバーの有効な組合せである有効な宛先 POV。
Source	ディメンション・メンバーの有効な組合せである有効なソース POV。Source は配賦される金額です。
Factor	有効なソース POV、定数または変数。
Scale	1 または 100 の整数値。Factor は Scale で除算されます。
Inverse	TRUE または FALSE。TRUE の場合は、Factor の符号が逆になります。これは、Factor が正数として格納される場合に減少させるときに使用できます(逆も同様)。FALSE の場合は、格納された Factor の符号を使用して、増加させるか減少させるかが決定されます。

詳細説明

この関数は、宛先 POV を比率係数で増加または減少させます。比率係数は、ソース POV、VBScript 定数、または VBScript 変数から取得されます。

一般的に、ソース POV は宛先 POV と同一ですが、別にすることも可能です。

Scale パラメータは、必要な場合に、係数を小さくするために使用されます。これは、係数がソース POV が取得され、調整されていないフォーム(50%が 0.50 ではなく 50 として格納される)に格納される場合に適用されます。

Inverse パラメータは、係数の符号を逆にするために使用されます。これは、係数がソース POV から取得され、絶対値として格納される場合に適用されます。Inverse パラメータが TRUE に設定されている場合、係数には-1 が乗算されます。Inverse パラメータが FALSE に設定されている場合、係数は-1 で乗算されません。

例

この例では、Telephone 勘定科目が 10%増加されます。

表 12-21 Increase_Decrease 関数の例

勘定科目	Jan2014	Feb2014	Mar2014
A#Telephone	100	300	400
A#Factor/C1[None]	10	10	10
	N/A	N/A	N/A
<pre>Increase_Decrease (" A#Telephone", "A#Telephone", "A#Factor.C1#[None] ",100,False)</pre>			

表 12-21 (続き) Increase_Decrease 関数の例

勘定科目	Jan2014	Feb2014	Mar2014
A#Telephone	110	330	440

INCREASE_DECREASE 関数から戻される結果は次のとおりです:

```
HS.EXP "A#Telephone = A#Telephone * (1+ (A#Factor.C1#[None]/100))"
```

サンプル・スクリプト

- 呼出しルーチンで記述されたサンプル文。
- 呼出しルーチンに設定され、Increase_Decrease 関数に渡される変数。
- Increase_Decrease 関数と同一になるように設定された呼出しルーチンの変数名。

```
Sub Calculate()

Dim Destination
Dim Source
Dim Factor
Dim Scale
Dim Inverse

Destination = "A#Telephone"
Source = "A#Telephone"
Factor = "A#Factor.C1#[None]"
```

```

Scale = "100"
Inverse = False

Call Increase_Decrease(Destination,Source,Factor,Scale,
Inverse)

End Sub

' Beginning of the Increase_Decrease function

Sub Increase_Decrease(Destination,Source,Factor,Scale,Inverse)

If Inverse = False Then

HS.EXP Destination & " = " & Source & " *

(1 + (" & Factor & " / " & Scale & "))"

Else

HS.EXP Destination & " = " & Source & " *

(1 + ((" & Factor & " * -1) / " & Scale & ))"

End If

End Sub

```

Pro_Rata_Ratio

この関数は、2つのソース POV ($C = A / B$)の比率を計算します。

戻り値

戻り値はありません。

構文

```
Pro_Rata_Ratio(Destination,SourceN,SourceD)
```

表 12-22 Pro_Rata_Ratio 関数の構文

パラメータ	有効な値
Destination	勘定科目、ICP、カスタム 1 から 4 のメンバーの有効な組合せである有効な宛先 POV。
SourceN	ディメンション・メンバーの有効な組合せである有効なソース POV。SourceN は比率計算の分子です。
SourceD	有効なソース POV。SourceD は比率計算の分母です。

詳細説明

この関数は、2 つのソース POV ($C = A / B$) の比率を計算します。

ベスト・プラクティスとして、呼出しルーチンに変数を設定し、Destination、SourceN および SourceD POV を定義する Pro_Rata_Ratio 関数に渡されるようにすることをお勧めします。また、呼出しルーチンの変数名を Pro_Rata_Ratio 関数と同一になるように設定することもお勧めします。

親メンバーの加重平均比率は自動的に計算されません。親メンバーの値は、子の値の集約として表示されます。この結果、親メンバーの値は数学的には正しくなくなります。そのため、Ratio 勘定科目では、集約を無効にしておくことをお勧めします。

例

MarginPct 勘定科目は、GrossMargin/TotalRevenues の値を戻します。

表 12-23 Pro_Rata_Ratio 関数の例

勘定科目	Jan2014	Feb2014	Mar2014
A#GrossMargin	1000	100	750
A#TotalRevenues	2000	400	1000
	0.50	0.25	0.75
Pro_Rata_Ratio("A#GrossMargin", "#TotalRevenues")			

PRO_RATA_RATIO 関数から戻される結果は次のとおりです:

```
HS.EXP "A#MarginPct = A#GrossMargin / A# TotalRevenues"
```

サンプル・スクリプト

このスクリプトには、次の情報が含まれます:

- 呼出しルーチンで記述されたサンプル文。
- 呼出しルーチンに設定され、Pro_Rata_Ratio 関数に渡される変数。

- **Pro_Rata_Ratio** 関数と同一になるように設定された呼出しルーチンの変数名。

```

Sub Calculate()

Dim Destination 'Destination POV
Dim SourceN     'Source Numerator POV
Dim SourceD     'Source Denominator POV

Destination = "A#MarginPct"
SourceN     = "A#GrossMargin"
SourceD     = "A#TotalRevenues    "

Call Pro_Rata_Ratio(Destination,SourceN,SourceD)

End Sub

' Beginning of the Pro_Rata_Ratio function

Sub Pro_Rata_Ratio(Destination,SourceN,SourceD)

HS.EXP Destination & " = " & SourceN & " / " & SourceD

End Sub

```

Spread

この関数は、ソース勘定科目の単一の期間値(たとえば、**P#[Year]**)を、プロファイル勘定科目(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配賦します。

戻り値

戻り値はありません。

構文

```
Spread(Destination,Source,Factor,FactorN,FactorD,Temp,Per)
```

表 12-24 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。

表 12-24 (続き) Spread 関数の構文

パラメータ	有効な値
Source	ディメンション・メンバーの有効な組合せである有効なソース POV。ソース POV には、P#[Year]などの単一の期間を含める必要があります。単一の期間数が分散される数です。
Factor	有効なソース POV。 <i>Factor</i> は、配賦係数の格納に使用される勘定科目です。
FactorN	有効なソース POV。 <i>FactorN</i> は、分散配賦の基準として使用される分子係数です。
FactorD	有効なソース POV。 <i>FactorD</i> は、分散配賦の基準として使用される分母係数です。
Temp	有効な宛先勘定科目。 <i>Temp</i> は、ソース値を一時的に格納する勘定科目です。
Per	January など、タイムフレームの最初の期間の名前を定義する期間文字列。 <i>Temp</i> 値は最初の期間に格納され、パラメータは計算でこの値を参照する必要があります。

詳細説明

この関数は、Profile POV (Revenue プロファイル、4-4-5 など)に定義されているプロファイルに基づいて、宛先 POV のすべての期間にソース POV の単一の期間値 (P#[Year]など)を配賦します。

時間ベースの配賦は、まず 1 年間の金額が入力されてから、適切なプロファイルに基づいて期間全体に金額が配賦される、予算策定アプリケーションに特に適しています。

ソース POV には、単一の期間を含める必要があります。期間は通常 P#[Year]ですが、P#January など、任意の単一の期間にすることが可能です。

ソース POV の値は、計算により一時勘定科目に格納されます。ソース勘定科目と宛先勘定科目が通常同一であるため、このようにする必要があります。これが当てはまる場合、P#[Year]の値は、計算が 1 つの期間から次の期間へ進むにつれて変わります。そのため、まず値を格納し、すべての期間で参照できるようにする必要があります。

呼出しルーチンに変数を設定し、Destination、Source、Profile、Temp および Period1 パラメータを定義する Spread 関数に渡すことをお勧めします。また、呼出しルーチンの変数名を Spread 関数と同一になるように設定することもお勧めします。

例

Telephone 勘定科目の年の値が、4-4-5 という四半期ごとの比率を使用して期間全体に配賦されます。

SPREAD 関数から戻される結果は次のとおりです:

```
HS.EXP "A#TempTelephone.C1#[None] = A#Telephone.C1#[None].P#[Year]" (Where  
Period.Number = 1)
```

```
HS.EXP "A#Telephone.C1#[None] = A#TempTelephone P#January *  
E.Globals.A#Profile445.C1#[None].P#Cur /  
E.Globals.A#Profile445.C1#[None].P#[Year]"
```

サンプル・スクリプト

このスクリプトには、次の情報が含まれます:

- 呼出しルーチンで記述されたサンプル文。
- 呼出しルーチンに設定され、**Spread** 関数に渡される変数。
- **Spread** 関数と同一になるように設定された呼出しルーチンの変数名。

```
Sub Calculate()
```

```
Dim Destination  
Dim Source  
Dim Factor  
Dim FactorN  
Dim FactorD  
Dim Temp  
Dim Per
```

```
Source = "A#Telephone.C1#[None].P#[Year]"  
Destination = "A#Telephone.C1#[None]"  
Factor = "A#Factor.C1#[None]"  
FactorN = "E#Globals.A#Profile445.C1#[None].P#CUR"  
FactorD = "E#Globals.A#Profile445.C1#[None].P#[Year]"
```

```

Temp = "A#TempTelephone.C1#[None]"
Per = "January"

Call Spread(Destination,Source,Factor,
FactorN,FactorD,Temp,Per)

End Sub

' Beginning of the Spread function

Sub Spread(Destination,Source,Factor,FactorN,FactorD,Temp,Per)

If HS.Period.Number = 1 Then

HS.Exp Temp & " = " & Source

End If

HS.Clear Factor
HS.EXP Factor & " = " & FactorN & " / " & FactorD

HS.Clear Destination
HS.EXP Destination & " = " & Temp & ".P#" & Per & " * " & Factor

End Sub

```

Units_Rates

この関数は、2つのソース POV ($C = A * B$)の積を計算します。

戻り値

戻り値はありません。

構文

```
Units_Rates(Destination,Units,Rates)
```

表 12-25 Units_Rates 関数の構文

パラメータ	有効な値
Destination	勘定科目、ICP、カスタム 1 から 4 のメンバーの有効な組合せである有効な宛先 POV。
Units	ディメンション・メンバーの有効な組合せである有効なソース POV。
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 12-26 販売単位数の関数の例

勘定科目	Jan2014	Feb2014	Mar2014
A#UnitsSold	1000	2000	5000
A#Price	1.25	1.00	0.50
	1250	2000	2500
Units_Rates("A#UnitsSold", A#Price)			

UNITS_RATES 関数から戻される結果は次のとおりです:

```
HS.EXP "A#Sales = A#UnitsSold * A#Price"
```

サンプル・スクリプト

このスクリプトには、次の情報が含まれます:

- 呼出しルーチンで記述されたサンプル文。
- 呼出しルーチンに設定され、Units_Rates 関数に渡される変数。

- **Units_Rates** 関数と同一になるように設定された呼出しルーチンの変数名。

```

Sub Calculate()

Dim Destination
Dim Units
Dim Rates

Destination = "A#Sales"
Units = "A#UnitsSold"
Rates = "A#Price"

Call Units_Rates(Destination,Units,Rates)

End Sub

' Beginning of the Units_Rates function

Sub Units_Rates(Destination,Units,Rates)

HS.EXP Destination & " = " & Units & " * " & Rates

End Sub

```

Calculation Manager を使用したルールを作成

Oracle Hyperion Calculation Manager モジュールは、Oracle Hyperion Financial Management の計算ルールを作成するための共通のユーザー・インター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

次の役割は、Oracle Hyperion Financial Management における Oracle Hyperion Calculation Manager アクセスで使用できます。

- ルール管理者 - ルール・オブジェクト、テンプレートおよび変数の作成、変更および削除や、ルール・セットの検証およびデプロイなど、指定されたアプリケーションに対して Calculation Manager のあらゆるタスクを実行できます
- ルール・デザイナー - ルール・オブジェクトを作成し、このオブジェクトを変更または削除できます
- ルール・ビューア - ルール・オブジェクトを表示および検証できます

Oracle Hyperion Enterprise Performance Management Workspace から Calculation Manager にアクセスするには、ルール管理者、ルール・デザイナーまたはルール・ビューアのセキュリティ役割が必要です。

ルールをデプロイするには、ルール管理者のセキュリティ役割が必要です。

セキュリティ役割の詳細は、*Oracle Enterprise Performance Management System ユーザー・セキュリティ管理ガイド*を参照してください。

Calculation Manager でのアプリケーションの操作

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、Oracle Hyperion Calculation Manager を使用してルールを設計し、Oracle Hyperion Financial Management アプリケーションにデプロイすることもできます。

ノート:

ルールを Calculation Manager にデプロイした後で VB Script ルールをロードしようとする、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。

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 設計者ガイド*またはオンライン・ヘルプを参照してください。

Calculation Manager へのルールの移行

以前のリリースの既存の VB Script ルール(.rle)ファイルがある場合、それらのファイルをインポートして Oracle Hyperion Calculation Manager に移行できます。*Oracle Hyperion Calculation Manager 設計者ガイド*のルールの管理に関する項を参照してください。

関数セレクトタでの VB 関数のサポート

Oracle Hyperion Calculation Manager の関数セレクトタでは次の VB 関数をサポートします。

ノート:

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	指定された配列の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、ゼロから始まる一次元配列を戻します。
UBound	指定された配列の次元の最も大きいサブスクリプトを戻します。

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。

関数	説明
DateDiff	2 つの日付間の間隔数を返します。
DatePart	指定された日付の指定された部分を返します
DateSerial	指定された年、月および日の日付を返します。
Day	月の日(1 から 31 まで)を表す数字を返します。
Month	年の月(1 から 12 まで)を表す数字を返します。
MonthName	指定された月の名前を返します。

数学

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

String

関数	説明
InStr	別の文字列内で、ある文字列が最初に出現する位置を返します。検索は文字列の最初の文字から始めます。
InStrRev	別の文字列内で、ある文字列が最初に出現する位置を返します。検索は文字列の最後の文字から始めます。
LCase	指定された文字列を小文字に変換します。
Left	文字列の左側から指定された数の文字を返します。
Len	文字列内の文字数を返します。
Mid	文字列から指定された数の文字を返します。
Right	文字列の右側から指定された数の文字を返します。
StrComp	2 つの文字列を比較し、比較の結果を表す値を返します。
Trim	文字列の左と右の両側にあるスペースを除去します。
UCase	指定された文字列を大文字に変換します。

Financial Management の特別な VB Script 関数

これらの特別な関数は、Oracle Hyperion Calculation Manager の UI で配列やループを処理するために、Oracle Hyperion Financial Management 用に実装されました。

Range

値	ループ変数	VBScript 生成
@Range(1-50)	i	Dim i(50) I(1)=1 I(2)=2 I(3)=3 .. I(50)=50
@Range(5-10)	i	Dim i(6) I(1)=5 I(2)=6 I(3)=7 I(4)=8 I(5)=9 I(6)=10
@Range(1,3-5,7-9)	i	Dim i(8) I(1)=1 I(2)=3 I(3)=4 I(4)=5 I(5)=7 I(6)=8 I(7)=9

For / ForStep

値	ループ変数	VBScript 生成	コメント
@For(2,10)	アイテム	For Item=2 to 10	New @ForLoop @For(from, to)
@ForStep(2,10,2)	アイテム	For Item =2 to 10 step 2	New @ForStep loop function @ForStep(from, to, step) 逆のステップが必要な場合は、 ステップの前に負数の 符号(-)を追加します。例: @ForStep(2,10,-2)

ExitFor

@ExitFor - ループの終了

VB Script の次の文が生成されます。

For each *element* in group

[*statements*]

Exit For

statements]

Next [*element*]

または

For *counter*=start To end [*Step step*]

[*statements*]

Exit For

[*statements*]

Next [*counter*]

ExitSub

@ExitSub - ルールの終了

VB Script の次の文が生成されます。

Sub *name* [(*x,y*)]

statements]

ExitSub

[*statements*]

EndSub

ReDim

1 つ以上の動的配列変数を再次元化し、それらのストレージ・スペースを再割り当てします。オプションの **Preserve** キーワードを使用すると、配列を再次元化するときに配列の内容を現状のままに維持できます。

{*VarArrayX*(5)} = @Redim

{*VarArrayXY*(5,9)} = @Redim

次の VB 文が生成されます。

Redim *VarArrayX*(5)

Redim *VarArrayXY*(5,9))

RedimPreserve

{*VarArrayXY*(5)} = @RedimPreserve

または

{*VarArrayXY*(5,9)} = @RedimPreserve

または

{*VarArrayXY*(5,{})} = @RedimPreserve

次の VB 文が生成されます。

RedimPreserve *VarArrayX*(5)

または

RedimPreserve *VarArrayXY*(5,9)

または

RedimPreserve VarArrayXY(5,i)

内部取引の管理

次も参照:

- [内部取引の設定](#)
- [理由コードの管理](#)
- [内部取引のモニタリング](#)
- [エンティティのロックとロック解除](#)
- [内部取引の要約の表示](#)

内部取引の設定

内部取引は、組織内の 2 つのエンティティの間での取引です。Oracle Hyperion Financial Management では、勘定科目とカスタム・ディメンションにまたがって、内部取引の詳細を追跡して調整できます。内部取引モジュールでは、内部取引勘定科目の差異の特定、レポート作成、調整を効率的に行えます。

内部取引の管理ページを使用して、内部取引を処理します。取引の処理の詳細は、*Oracle Hyperion Financial Management ユーザーズ・ガイド*を参照してください。

内部取引を入力する前に、次の設定手順を実行する必要があります:

- 内部取引の期間を開きます。[内部取引の期間を開く](#)を参照してください。
- 内部取引ルールを定義して、アプリケーションにロードします。HS.SupportsTran 関数は、内部取引をサポートするアプリケーションの勘定科目、シナリオおよびカスタム・ディメンションを指定します。[ルール・ファイルの作成](#)および[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。[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。[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。[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「締切り済」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。[「転記前に照合」オプションの設定](#)および[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、「**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、「**アクション**」、「**期間の設定の保存**」の順に選択します。

 **ヒント:**

変更を保存せずに期間を元のステータスに復元するには、「**復元**」をクリックするか、または「**アクション**」、「**復元**」の順に選択します。

照合許容差を設定する

自動照合と手動照合のプロセスに期間別の照合許容差を設定できます。勘定科目と手動照合の許容差に値を設定できます。取引 ID (TID)許容差の場合は、値、パーセントまたはその両方を指定できます。

取引 ID にパーセントを入力すると、エンティティの取引の合計とパートナーの取引の合計で小さい方の値が使用され、その値にパーセントが適用されて、許容値が生成されます。

たとえば、TID123 のエンティティ A で 3 つの取引を持ち、これらの取引の合計が 1000 であるとして、TID123 のパートナー B が合計 1020 の 5 つの取引を持ちます。エンティティの合計とパートナーの合計の差は 20 です。ただし、3%の許容差を指定すると、小さい方の合計の 3%(1000x3%)が計算され、結果が 30 になります。その値を差と比較すると、許容差内にあるため、取引は一致とみなされます。

パーセント以外に、たとえば値 15 を入力すると、パーセント値が入力された値と比較され、最も小さい方の値が比較に使用されます。この例では、エンティティの合計とパートナーの合計の差は 20、パーセントの許容差は 30、値の許容差は 15 です。これは、許容差内とみなされないため、取引は一致しません。

TID の値とパーセントを空白のままにすることもできます。いずれかがゼロ値の場合、または両方が空白の場合、取引の差がゼロの取引のみが照合されます。

勘定科目の許容差と手動照合の許容差の場合、照合許容差は、アプリケーションの通貨で表され、入力された値はアプリケーションの通貨の位取りで表されます。照合プロセス時に、各取引がアプリケーションの通貨に変換され、合計値の差が期間に設定された照合許容差と比較されます。比較は 1 の単位で実行されます。

取引を TID 許容値と比較する場合は、TID/RID の照合プロセス時に、取引はアプリケーションの通貨に換算されません。

「転記前に照合」オプションの設定

期間ごとに、「転記前に照合」オプションを設定できます。照合/検証オプションでは、取引の転記前に取引の照合ステータスを確認する必要があるかどうかを定義し、エンティティをロックするか期間を閉じる前に実行する必要がある検証タイプを定義します。

「転記前に照合」オプションを選択する場合は、転記する前に、取引を一致させるか理由コードを割り当てる必要があります。期間を閉じる前、またはエンティティをロックする前には、一致した取引または理由コードのある不一致の取引すべてを転記する必要があります。

「制限」オプションを選択する場合は、転記する前に取引を一致させる必要がありますが、転記が戻されている取引を含む期間を閉じることや、エンティティをロックすることは可能です。

表 14-1 「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。 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。 また、期間を閉じるかエンティティをロックする前に、一致した取引または理由コードのある不一致取引のすべてが転記されているかどうかをチェックされます。
制限	「転記前に照合」オプションを「制限」に設定すると、次のいずれかの条件が満たされた場合に取引を転記できます： <ul style="list-style-type: none"> 取引のステータスが「一致」である。 ステータスが「不一致」の取引に有効な理由コードが含まれている。 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。 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。

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、「**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:

- [理由コードの追加](#)
- [理由コードの編集](#)
- [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。取引のロードの詳細は、*Oracle Hyperion Financial Management ユーザーズ・ガイド*を参照してください。

理由コードを追加するには:

1. 「**連結**」、「**内部取引**」、「**理由コード**」の順に選択します。
2. 「**行の追加**」をクリックするか、「**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。

 **ノート:**

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。

 **ノート:**

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」 または 「すべて削除」 をクリックするか、「アクション」、「行の削除」 または 「すべて削除」 の順に選択します。
4. 「はい」 をクリックして、理由コードを削除します。

内部取引のモニタリング

内部取引のモニター機能を使用して、内部取引照合プロセスをモニターします。多数の内部取引が期間内に入力されると、すべての取引が正常に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モニターする必要があります。内部取引のモニター機能を使用すると、内部取引プロセスを開始した内部取引パートナーを簡単に見つけることができます。

内部取引をモニターするには、「内部取引管理者」のセキュリティ役割が割り当てられている必要があります。この役割は、プロセス・ステータス、ロック・ステータス、エンティティの詳細、内部取引の要約を表示し、電子メール・アラートを実行できます。

内部取引のモニター・ページには、プロセス・ステータスとロック・ステータスが付いた内部取引エンティティのリストが表示されます。エンティティ、プロセス・ステータスまたはロック・ステータスで表示をフィルタできます。エンティティは、内部取引のモニター詳細情報にリンクされます。エンティティをクリックすると、「一致」、「不一致」、「未照合」などのステータス別に転記済と未転記の取引数が表示されます。

表 14-2 内部取引のエンティティ・ステータス

ステータス	説明
未開始	エンティティには、視点のシナリオ、年および期間に内部取引がありません。
開始済	エンティティには、視点のシナリオ、年および期間に少なくとも 1 つの内部取引があります。
ロック不可	その期間に「転記前に照合」を選択したが、一致した取引も理由コードのある不一致の取引も転記しなかった場合、エンティティのステータスは「ロック不可」になります。

リスト内のエンティティをクリックして、照合ステータスで分類されたエンティティの転記済および転記を戻し済の合計取引数を表示できます。

たとえば、エンティティ A の横にあるプラス(+)記号をクリックして展開すると、ページには、エンティティ A の転記済の取引と転記を戻し済の取引の数がステータス別に表示されます。「未照合」列で転記を戻し済の取引の値をクリックすると、フィルタが適用された内部取引の管理ページにリンクされ、ステータスが「未照合」のエンティティ A の転記を戻した取引が表示されます。同時に複数のエンティティを展開して、その詳細を表示できます。

内部取引のモニター・ページで任意のエンティティに電子メール・アラートを送信できます。*Oracle Hyperion Financial Management ユーザーズ・ガイド*を参照してください。

内部取引をモニターするには:

1. 「連結」、「内部取引」、「モニター」の順に選択します。
2. 視点バーで、シナリオ、年、期間を選択します。
3. 「表示オプション」リストから、次のオプションを選択します:
 - ラベルを使用してエンティティ情報を表示するには、「ラベル」を選択します。
 - 説明を使用してエンティティ情報を表示するには、「説明」を選択します。
 - ラベルと説明を使用してエンティティ情報を表示するには、「両方」を選択します。
4. 「フィルタ」リストから、「エンティティ」に、ステータスをモニターするエンティティを入力または参照します。

 ノート:

「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、またはコンテキスト・メニューから「**電子メール・アラート**」を選択するか、または「**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。

 ノート:

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・ステータスの詳細は、*Oracle Hyperion Financial Management ユーザーズ・ガイド*を参照してください。

「内部取引管理者」のセキュリティ役割が割り当てられている場合、プロセス・ステータス、ロック・ステータス、エンティティの詳細、内部取引の要約を表示し、電子メール・アラートを実行できます。

エンティティがロックされると、新規の内部取引を入力できません。また、既存の取引を削除または変更することもできません。ロックされたエンティティに取引を転記すること、転記を戻すこと、またはロックされたエンティティを含む取引の照合ステータスを更新することはできません。したがって、パートナ・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と、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。

複数の期間のロックを解除するには、**[Ctrl]**キーを使用して期間をハイライトし、「**ロック解除**」を選択します。**[Shift]**キーを使用して複数の期間をハイライトしてロックを解除することはできません。

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。

プロセス管理の提出フェーズの管理

プロセス管理は、財務データの確認と承認を管理します。データは確認目的でプロセス単位に編成されており、特定のシナリオ、年、期間、エンティティおよび値ディメンションのデータが組み合されています。確認サイクル中に、レベルの移行、送信、承認、却下および公開などのアクションをプロセス単位に実行できます。

プロセス管理では、データ・プロセス単位を異なる提出フェーズに分割し、データのサブセットを処理できます。確認プロセス時に、プロセス単位全体ではなく、フェーズのプロセス単位ごとにレベルを上げることができます。これにより、シナリオを追加して確認プロセスを適用する必要がなくなりました。

プロセス管理の提出フェーズを管理するには、**UseSubmissionPhase** アプリケーション属性が有効になっている必要があり、アプリケーション管理者または確認監督者である必要があります。

提出フェーズの定義

確認プロセス要件は、期間ごとに異なる場合があります。たとえば、月締めサイクルは、1月と2月の貸借対照表と損益計算書のデータで単一フェーズの確認プロセスを必要とする場合があります。3月などの四半期月の場合、四半期締めの確認プロセスは、貸借対照表、損益計算書および補足のデータで複数フェーズの送信サイクルを必要とします。

確認プロセス要件は、シナリオごとに異なる場合もあります。たとえば、実績シナリオは、確認のために送信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送信し、フェーズ 2 で補足データを送信します。予算シナリオでは、フェーズ 1 で内部取引データを送信し、フェーズ 2 で貸借対照表と損益計算書の勘定科目を送信し、フェーズ 3 で補足データを送信します。

提出フェーズを設定するには、次のタスクを実行します：

- アプリケーションとディメンション・メタデータの属性を設定して、提出フェーズを使用します。[アプリケーション設定の定義](#)を参照してください。
 - **UseSubmissionPhase** アプリケーション属性を Y に設定します。デフォルトでは、提出フェーズの設定が無効なため、この属性を設定してアプリケーションのフェーズ送信を有効にする必要があります。
 - 必要に応じて、**SupportSubmissionPhaseForAccounts**、**SupportSubmissionPhaseForCustom** または **SupportSubmissionPhaseForICP** 属性を設定します。プロセス管理に有効にするディメンション(勘定科目、カスタムおよび ICP メンバー)を定義する必要があります。たとえば、アプリケーションで勘定科目ディメンションによる送信が必要で、カスタム・ディメンションまたは内部取引(ICP)ディメンションからは必要としない場合は、**SupportSubmissionPhaseForAccounts** 属性を選択します。少なくとも1つのディメンションを有効にする必要があります。

- 検証勘定を割り当てます。[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。
フェーズ送信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。フェーズ送信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。
- 送信グループを定義して、送信グループをディメンション・メンバーに割り当てます。[送信グループの設定](#)を参照してください。
- 送信グループを提出フェーズに割り当てます。[送信グループのフェーズへの割当て](#)を参照してください。

送信グループの設定

アプリケーションごとに、送信グループに含めるディメンション・メンバーを決定する必要があります。たとえば、送信グループ 1 に現金勘定科目および投資勘定科目を定義できます。

メタデータ・ファイルで、送信グループに次の属性を設定します: **SubmissionGroup=0** から **99** または **<blank>**。

デフォルトは空白です。**<blank>**値にすると、デフォルトの値 **1** になります。

送信グループを **0** に設定すると、勘定科目が確認プロセスに含まれなくなります。

複数のディメンションがフェーズ送信に使用される場合、ディメンション・メンバーの最大のグループ割当てによってセル送信グループ割当てが決定されます。送信グループを割り当てる前に、必要なすべてのグループ割当てを検討する必要があります。

例 1:

Account=2

C1=1

C2=2

C3=1

C4=1

ICP=1

これらのディメンションの最大送信グループ数が 2 であるため、セルの送信グループ値は 2 となります。

例 2:

Account=1

C1=3

C2=2

C3=5

C4=1

ICP=3

これらのディメンションの最大送信グループ数が 5 であるため、セルの送信グループ値は 5 となります。

送信グループおよびフェーズの例

フェーズのアプリケーションとディメンション・メンバーの属性を設定し、ディメンション・メンバーの送信グループを定義したら、勘定科目の送信グループを各提出フェーズに割り当てることができます。割り当ては、プロセス管理をサポートするシナリオにのみ適用されます。ただし、割り当ては、シナリオと期間によって実行される必要があります。

これらの例は、サンプルの送信グループと提出フェーズへのそれらの割り当てを示します。

勘定科目	送信グループ
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)	送信グループ
[なし]	1
Golf Balls	7
Tennis Balls	8
Soccer Balls	9

基本勘定科目は、親勘定科目から送信グループを継承しません。親勘定科目は、子の送信グループを前提にしません。送信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 勘定科目は送信グループ 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送信グループの割当てを示します。

表 15-1 例: 提出フェーズへの送信グループの割当て

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送信を使用します。

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、送信グループ割当て 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、送信グループ割当て 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、送信グループの割当て 5 と 6 の全勘定科目となります。

送信グループのフェーズへの割当て

各提出フェーズに送信グループを割り当てることができます。割当ては、プロセス管理をサポートするシナリオにのみ適用されます。グループが指定されていない場合、プロセス管理はグループ内のディメンショナル・メンバーに適用されません。

プロセス管理の送信グループを割り当てるには、アプリケーション管理者または確認監督者である必要があります。

送信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、データを送信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クリアされます。

提出フェーズの割当てをスキップできます。たとえば、グループを提出フェーズ 2 に割り当てないで、提出フェーズ 1 および 3 に割り当てることができます。提出フェーズに割り当てられていないグループは、確認プロセスの一部とみなされません。これらのディメンショナル・メンバーは、確認レベルのセキュリティの確認をせずに、該当するセキュリティ・クラスのアクセス権を持つすべてのユーザーが使用できます。データを入力する前に、未割当てのセルをプロセス管理で開始する必要はありません。

送信グループを提出フェーズに割り当てるには:

1. 「連結」、「データ」、「提出フェーズ」の順に選択します。

デフォルトでは、すべてのフェーズが表示されます。右側の「オプション」ペインで、非表示にするフェーズをクリアできます。

2. シナリオを変更するには、POV でシナリオ・ディメンションをクリックし、シナリオを選択して「OK」をクリックします。
3. 各フェーズ列で、その提出フェーズのグループを入力し、[Enter]を押します。
 - 複数のグループを入力するには、区切りとしてカンマを使用します。
 - グループの範囲を指定するには、区切りとしてマイナス記号を使用します。
 - すべてのグループを示すには、「すべて」を指定します。

4. 「送信」をクリックするか、「アクション」、「送信」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、「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。

未割当ての送信グループの表示

「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送信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送信グループは、確認プロセスの一部ではありません。未割当てグループを表示すると、グループが間違っ確認プロセスから見落とされていないかを確認できます。

どの提出フェーズにも割り当てられていないディメンション・メンバーに割り当てられたグループと、どのディメンション・メンバーにも割り当てられていない提出フェーズに割り当てられたグループが表示されます。グループがディメンション・メンバーとフェーズのいずれにも割り当てられていない場合、そのグループは表示されません。

たとえば、アプリケーションで 1-10 の送信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

未割当て送信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。

電子メール・アラートの管理

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

電子メール・アラートは、標準の SMTP(Simple Mail Transfer Protocol)を使用して送信されるため、インターネット電子メールで動作するすべての電子メール・システムでアラートを使用できます。アラートを使用するには、Financial Management 構成ユーティリティを実行する場合、電子メール設定を構成し、SMTP サーバー名を指定する必要があります。Oracle Enterprise Performance Management System インストールおよび構成ガイドを参照してください。

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

電子メール・アラートを送受信する前に、アプリケーションでユーザーとデータのセキュリティ権限を設定する必要があります。アラートのシナリオおよびエンティティに割り当てられたセキュリティ・クラスでは、電子メール・アラートをサポートする必要があります。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の詳細は、Oracle Enterprise Performance Management System ユーザー・セキュリティ管理ガイドを参照してください。

プロセス管理のアラートの設定

プロセス・コントロール・モジュールを設定して、プロセス・コントロールでのステータスの変更に基づいて電子メール・アラートを生成できます。アラートは、「第 1 パス」、「確認レベル 1」から「確認レベル 10」、「送信済」、「承認済」、または「発行済」のアクションに対して設定できます。

プロセス単位が「開始していません」レベルの場合、またはサインオフ・アクションに対しては、電子メール・アラートは生成されません。

プロセス管理の電子メール・アラートを設定するには:

1. メタデータ・ファイルにおいて、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。

ノート:

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 16-1 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「送信者」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「送信者」の役割のユーザーに通知されます。
送信済	「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送信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「送信者」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。

 ノート:

「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の詳細は、*Oracle Enterprise Performance Management System ユーザー・セキュリティ管理ガイド*を参照してください。

内部取引の電子メール・アラートを設定するには:

1. 内部取引の電子メール・アラートを受信セキュリティ役割にユーザーを割り当てます。
2. セキュリティ・クラスのセキュリティ・アクセス・セクションで、アラートを有効にするには、アラートのサポート・オプションを Yes の Y に設定します。例:
[Default];User1@NativeDirectory;All;Y。

すべての条件を満たすユーザーが内部取引モジュールまたは内部取引パートナー照合レポート・モジュールから電子メール・アラートを受信します。

内部取引の電子メール・アラートの生成については、*Oracle Hyperion Financial Management ユーザーズ・ガイド*を参照してください。

A

構成設定

設定モジュールでは、Oracle Hyperion Financial Management の構成設定を表示および変更できます。デフォルトで、インストール時に、設定表には推奨される設定が移入されます。

必要に応じて、設定値を変更できます。たとえば、メモリーの使用状況に基づいて値を変更したり、パフォーマンスを改善する必要があります。このような値を更新する上での考慮事項は、[パフォーマンスの最適化](#)を参照してください。

特定のアプリケーションの設定を上書きできます。システム・レベルの値を変更するには、**Financial Management** 管理者のセキュリティ役割が必要です。アプリケーション・レベルの値を変更するには、**Financial Management** 管理者とアプリケーション管理者の両方のセキュリティ役割が必要です。

設定リストをソートして設定を検索し、Excel ワークシートにエクスポートできます。設定の上書きを削除することはできませんが、システムによって作成された設定は削除できません。

特定アプリケーションの設定のランタイム値を表示することもできます。[有効な設定の表示](#)を参照してください。

次のトピックを参照してください:

- [使用可能な構成設定](#)
- [構成設定の変更](#)
- [値の上書き](#)
- [設定表の表示の変更](#)
- [設定の検索](#)
- [有効な設定の表示](#)
- [設定のエクスポート](#)
- [設定の削除](#)

使用可能な構成設定

表 A-1 構成設定

設定	説明	値
AllowOverlappingConsolidationOverride	ユーザーが重複する連結プロセスを実行できるかどうかを指定します。	<p>0 - ユーザーは重複する連結を実行できません。現在実行中で重複するものがある場合、新しい連結プロセスを開始できません。</p> <p>1 - ユーザーは重複する連結を実行できます。別の連結プロセスが現在実行中で重複する場合、新しい連結を開始するかどうかを尋ねる警告メッセージが表示されます。</p> <p>デフォルト値: 1</p>
AutoClearDeadTasks	完了したタスクをシステムから自動的にクリアするかどうか	<p>0 - 完了したタスクをシステムに残します</p> <p>1 - 完了したタスクをシステムからクリアします</p> <p>デフォルト値: 1</p>
AutoClearEAFlatfileTasksAfterSeconds	「タスクの実行」からデータの抽出タスクを自動的にクリアし、アプリケーション・サーバーからファイルを削除するまでシステムが待機する秒数	<p>範囲: 600 秒(10 分) - 864,000 秒(10 日)</p> <p>デフォルト値: 86,400 秒</p>
BRCRowFetchSize	表のすべての繰返しの行フェッチ・サイズ。LCM アプリケーション・スナップショットのエクスポートでのみ使用されます。	<p>範囲: 1000 - 100000 行</p> <p>デフォルト値: 5000</p>
BRCThreadPoolCount	スレッド・プール・サイズ - LCM アプリケーション・スナップショットのインポートおよびエクスポートの処理に使用されるスレッド数	<p>範囲: 10 - 100 スレッド</p> <p>デフォルト値: 15</p>

表 A-1 (続き) 構成設定

設定	説明	値
ConcurrentCalculationWaitTimeInSecond	<p>同時計算の最大数が実行されている場合に、ユーザーが新しい計算を起動すると、システムはタイムアウトまで待機し、POV 詳細とともに、タイムアウトに関するメッセージをシステム・ログに記録します。</p> <p>同時計算の最大数が実行されていて</p> <p>ConcurrentCalculationWaitTimeInSecond = -1 の場合に、ユーザーが新しい計算を起動すると、システムはいずれかの計算が終了するまで無制限に待機します。</p> <p>この設定は各サーバーのアプリケーションごとに管理者が変更できますが、15 秒より短く設定できません。</p>	デフォルト値: 30 秒
ConsolidationMultiThreadingScheme	<p>連結プロセスの実行方法に関連する複数の設定に使用します。各設定は、他の設定に影響を与えることなく有効化または無効化できます。</p> <p>複数のオプションを選択し、選択したオプションの合計にパラメータを設定できます。たとえば、オプション 2 および 4 を使用するには、値を 6 に設定します。値 0 は、3 つのオプションのいずれも有効になっていない制限されたマルチスレッドでの従来の連結動作を表します。この設定は、サーバーまたはアプリケーション・レベル(あるいはその両方)で上書きできます。</p>	<p>有効値:</p> <p>1 - 数値が親エンティティに連結されるまでは、連結プロセスですべての子エンティティを [Parent] の値に達するまで計算および換算することが可能です。</p> <p>2 - マルチスレッドの親レベル・エンティティへの連結プロセスが可能です。これにより、大規模な連結においてパフォーマンスが大幅に向上します。</p> <p>4 - 全ベースレベルのエンティティの初期計算をスキップします。これらのエンティティは、親エンティティを処理するときに必要に応じて計算されます。</p> <p>8 - 連結プロセスで、「すべてを連結」および「データのあるものすべてを連結」での最後の失敗によって影響を受けた子のみを連結することが可能です。</p> <p>デフォルト値: 2</p>
DataSize	表コンポーネントのフェッチ・サイズ	<p>範囲: 25 - 2,000</p> <p>デフォルト値: 500</p>

表 A-1 (続き) 構成設定

設定	説明	値
DefaultAdminPage	連結管理のデフォルトの開始ページ	有効値: 次のページのいずれか: システム・メッセージ、アプリケーション、システム設定、システム・ユーザー、システム管理、タスクの監査、データの監査、タスク・フローの管理、タスクフロー・ステータス、DSN の構成 デフォルト値: システム・メッセージ
DefaultColFetchSize	データ・フォームおよびデータ・グリッドのデフォルトの列フェッチ・サイズ	範囲: 25 - 2,000 デフォルト値: 50
DefaultRowFetchSize	データ・フォームおよびデータ・グリッドのデフォルトの行フェッチ・サイズ	範囲: 25 - 2,000 デフォルト値: 250
DSSStartupOption	アプリケーションをいつ開始および停止するかを決定します。 この設定は、グローバルに設定するか、アプリケーションおよびサーバー・レベルで上書きできます(あるいはその両方)。	0 - 最初のユーザーがアプリケーションを開こうとしたときに、オン・デマンドで開始します。最後のユーザーがアプリケーションをログオフした後に停止します。この設定は、必要などきのみアプリケーションを稼働させる場合に有用です。アーカイブ用のアプリケーションは、常時稼働している必要はありません。 1 - アプリケーションは Java サーバーの起動時に開始します。最後のユーザーがシステムからログオフした後も、アプリケーションは稼働し続けます。この設定は、アプリケーションを常時稼働させ、ユーザーがアプリケーションを開く際の待機時間が長くないようにする場合に有用です。 2 - アプリケーションはオン・デマンドで開始します。最後のユーザーがシステムからログオフした後も、アプリケーションは稼働し続けます。 デフォルト値: 0
EnableRulesLogging	ルールのロギングを有効にするかどうか。ルールの問題を特定するためのデバッグの目的で使用します。	0 - ロギングを無効にします 1 - ロギングを有効にします デフォルト値: 0

表 A-1 (続き) 構成設定

設定	説明	値
EnableRulesProfiling	ルール・プロファイリングの有効化。アプリケーション・レベルで、またはグローバルに値を上書きできます。	0 - ルール・プロファイリングを無効にします 1 - ルール・プロファイリングを有効にします デフォルト値: 0
EnableRunningTasksMaskUserNames	管理者以外のユーザーが表示する場合に、「タスクの実行」でユーザー名をマスクするかどうかを指定します。	0 - ユーザー名のマスクングを無効にします 1 - ユーザー名のマスクングを有効にします デフォルト値: 0
IcmSystemReportTranslationMode	通貨換算を必要とする内部取引照合システム・レポートの実行時、デフォルトでは換算された値がデータベースに書き込まれます。これにより、レポートの実行時間が長くなり、データベース・サイズが大幅に増加する可能性があります。この設定を使用すると、換算された値の処理方法を制御できます。 換算モード = 0 (サブ換算を実行し、データを保管します) 換算モード = 1 (サブ換算を実行し、データを保管しません) 換算モード = 2 (デフォルト換算を使用してオンザフライ換算を実行し、サブ換算を実行せず、データを保管しません)	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。 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。ノート: CN ステータスの親エンティティも換算されます。 2 - オンザフライ、デフォルト換算を使用します。デフォルト換算のみを実行します(サブ換算は実行しません)。変更はデータベースに書き込まれません。ノート: CN ステータスの親エンティティは換算されません。 デフォルト値: 0
MaxDataCacheSizeInMB	セルを格納するためにアプリケーション・サーバーによって割り当てられる最大メモリー (MB)	範囲: 500 以上の値にする必要があります。 デフォルト値: 4,500
MaxFileSelectionOnLoad	複数ファイル選択コントロールで選択できる最大ファイル数	値は 1 以上にする必要があります。 デフォルト値: 10
MaxNumConcurrentCalculations	各サーバーのアプリケーションごとに可能な同時計算の最大数。この設定は各サーバーのアプリケーションごとに管理者が変更できます。	デフォルト値: 8
MaxNumConcurrentConsolidations	各サーバーのアプリケーションごとに可能な同時連結の最大数。設定値を超えて実行される連結は、スケジュールされた連結としてキューに入ります。この設定は、グローバルに設定するか、アプリケーションのサーバー・レベルで上書きできます。	範囲: 範囲にハード・リミットはありません。値は正数である必要があります。 デフォルト値: 8

表 A-1 (続き) 構成設定

設定	説明	値
MaxNumConcurrentConsolidationsPerAction	ユーザー・アクション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。	デフォルト値: 8
MaxNumCubesInRAM	FreeLRU アルゴリズムのサブキューブの最大数	範囲: 100 以上の値にする必要があります。 デフォルト値: 60,000
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。値は正数である必要があります。 デフォルト値: 30,000,000
MaxNumRetriesOfBaseLevelCalculation	連結プロセスがベースレベル・エンティティに対する計算を再試行する最大回数。この設定により、連結プロセス中に基本レベル・エンティティの計算を、指定した再試行の回数まで、またはこれらのエンティティのステータスが OK になるまで、複数回繰り返すことができます。 これは、ある基本レベル・エンティティで実行されているルールの HS.ImpactStatus 関数が、すでに計算済である別のエンティティに影響を与え、最上位エンティティのステータスが OK でない場合に使用されます。 例: カリフォルニアとコネチカットは、米国の親エンティティにロール・アップする基本レベル・エンティティです。米国の計算中に、 SubCalculate で HS.ImpactStatus を使用する場合、コネチカット・エンティティが影響を受けます。連結を実行した後、コネチカット子孫が変更されているため、米国の親エンティティのステータスは OK ではありません。この設定を 2 に設定した場合、計算処理が 2 回再試行されます。2 回目の計算の試行で、カリフォルニアは OK、コネチカットは計算されて米国にロール・アップされ、米国エンティティのステータスは OK です。3 つまたは 4 つのレベルの依存関係がある場合は、必要に応じて何度でも再試行するように設定を調整できます。	範囲: 1 - 4 デフォルト値: 0

表 A-1 (続き) 構成設定

設定	説明	値
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、メモリの断片化を低減できます。通常、データ・キャッシュは必要に応じて増加され、一度に最大 25MB 増加します。	範囲: 100 - 5,000 デフォルト値: 2,250
NumConsolidationThreads	連結ごとに可能な最大連結スレッド数。値を減らすと、システムによるリソースの使用が制限されるため、連結のパフォーマンスが低下します。システム上に、Oracle Hyperion Financial Management で使用できる同数の CPU コアが必要です。	範囲: 範囲にハード・リミットはありません。1 以上の値にする必要があります。 デフォルト値: 8
NumCubesLoadedBeforeCheckingLRU	FreeLRU アルゴリズムをトリガーするために、RAM にロードする必要があるキューブの数。2 つの条件のいずれかを満たした (NumMinutesBeforeCheckingLRU を満たしたか、NumCubesLoadedBeforeCheckingLRU に達した) 場合に、FreeLRU がトリガーされます。	範囲: 範囲にハード・リミットはありません。値は正数である必要があります。 デフォルト値: 100
NumDataLoadsAllowed	各サーバーのアプリケーションごとに可能な最大同時データ・ロード・タスク数。この設定は、グローバルに設定するか、アプリケーションのサーバー・レベルで上書きできます。	範囲: 範囲にハード・リミットはありません。値は正数である必要があります。 デフォルト値: 8
NumEAExportsAllowed	各サーバーのアプリケーションごとに可能な「データの抽出」タスクの最大同時数。この設定は、グローバルに設定するか、アプリケーションのサーバー・レベルで上書きできます。	範囲: 範囲にハード・リミットはありません。値は正数である必要があります。 デフォルト値: 8
NumEAThreads	データ抽出ごとのスレッド数	範囲: 範囲にハード・リミットはありません。値は正数である必要があります。 デフォルト値: 8
NumMinutesBeforeCheckingLRU	FreeLRU アルゴリズムの制限のチェック間隔(分単位)	1 以上の値にする必要があります。 デフォルト値: 15
NumReportsAllowed	レポート・タイプごと、サーバーごと、アプリケーションごとに可能な最大同時レポート・タスク数。	デフォルト値: 8

表 A-1 (続き) 構成設定

設定	説明	値
NumThreadsToUseWhenUpdatingCalcStatusSystemWasChanged	メタデータのロード後に算出ステータスを更新する際に使用するスレッドの数	1 以上の値にする必要があります。 デフォルト値: 16
SessionManagerTimeoutInMS	Financial Management でセッション・マネージャによって使用されるタイムアウト制限 (ミリ秒)	60,000 以上の値にする必要があります。 デフォルト値: 1,200,000
OverrideUserFetchSizeWhenOpening	データ・フォームおよびデータ・グリッドのすべてのユーザーについてフェッチ・サイズを上書きするかどうかを指定します。	0 - すべてのユーザーのフェッチ・サイズを上書きしません 1 - すべてのユーザーのフェッチ・サイズを上書きします デフォルト値: 0
WebformDoCalculateOnSubmit	ユーザーがデータ・フォームで「データの送信」をクリックする際、データの送信後に自動的に計算するかどうか。	0 - 自動的に計算しません 1 - 自動的に計算します デフォルト値: 0

構成設定の変更

設定を変更するには、管理者のセキュリティ権限を持っている必要があります。システム所有の設定の場合、値以外の列は変更できません。

「ノート」列を使用してコメントを入力できます(たとえば、値の変更理由など)。行の「ノート」アイコンの上にマウスを置くと、設定の既存のノートを表示できます。

設定が変更されると、「設定」表が、変更したユーザー名および変更日時を表示するように更新されます。

青で表示される設定は、ユーザーが変更可能な設定がデフォルト値と一致しておらず、値が更新されたことを示します。

構成設定を変更するには:

1. 「ナビゲート」、「管理者」、「連結管理」の順に選択します。
2. 「管理タスク」から、「システム」を展開し、「設定」をクリックします。
3. 「設定」表から、設定の値を変更します。

ヒント:

元の値にリセットするには、「リセット」をクリックして、確認プロンプトから「はい」をクリックします。

4. オプション: コメントを入力するには、その行の「ノート」アイコンをクリックしてコメントを入力し、「OK」をクリックします。
5. 「保存」をクリックします。
6. 変更を表示するには、「リフレッシュ」をクリックします。

値の上書き

サーバーおよびアプリケーションの設定のデフォルト値を上書きできます。設定がグローバルの場合、上書きすることはできません。

設定をコピーするには:

1. 「設定」表から、コピーする設定を選択します。
2. 「上書き」をクリックするか、「アクション」、「上書き」の順に選択するか、設定を右クリックして「上書き」を選択します。
3. 「上書き」ダイアログ・ボックスで、設定値を変更し、設定を適用するサーバーおよびアプリケーションを選択します。
4. **オプション:** コメントを入力するには、その行の「ノート」アイコンをクリックしてコメントを入力し、「OK」をクリックします。
5. 「適用して閉じる」をクリックして、新しい値を適用します。

作成した新しい設定は、表内に太字で表示され、選択された状態になっています。

オプション: 別の設定を上書きするには、「適用して別に上書き」をクリックします。

6. 「保存」をクリックします。

設定表の表示の変更

任意の設定列をソートでき、また、列を並べ替えることができます。

表示する列を選択するには:

1. 「設定」リストから、「表示」、「列」の順にクリックし、表示する列を選択するか、「すべて表示」を選択します。
2. **オプション:** 列を表示または非表示にするには、「表示」、「列」、「列の管理」の順に選択し、矢印キーを使用して列を「非表示」または「表示可能」列リストに移動します。
3. **オプション:** 列のソート順を変更するには、ヘッダー・アイコンをクリックして、「昇順にソート」または「降順にソート」を選択します。

設定の検索

選択した基準に基づいて設定を検索できます。「設定の管理」タブで、設定名、設定を適用するサーバーおよびアプリケーション、設定の更新日、設定を作成または更新したユーザーを検索できます。

「有効な設定」タブでは、アプリケーションに対して、サーバーが実行時に使用した値を参照できます。設定は、名前、サーバー、アプリケーション別に検索できます。

設定を検索するには:

1. 「検索」フィールドから、検索する基準を選択するか入力します。
2. すべての基準との一致の場合は「すべて」を、選択した基準のいずれかとの一致の場合は「任意」を選択します。
3. 「検索」をクリックします。

 **ヒント:**

「検索」設定をリセットするには、「検索」の隣の「リセット」ボタンをクリックします。

有効な設定の表示

「有効な設定」タブから、アプリケーション設定のランタイム値を表示できます。

有効な設定を表示するには:

1. 「**設定の管理**」から、サーバー名およびアプリケーション名を選択します。
2. 「**有効な設定**」をクリックするか、「**アクション**」、「**有効な設定**」の順に選択します。

有効な設定は、選択したアプリケーションおよびサーバーに基づいています。上書きされたデフォルト値は、青で表示されます。

3. メインの設定リストに戻るには、「**設定の管理**」をクリックするか、「**アクション**」、「**設定の管理**」の順に選択します。

設定のエクスポート

設定のリストをエクスポートして、Excel ワークシートに保存できます。

設定をエクスポートするには:

1. 「設定の管理」で「**エクスポート**」をクリックするか、「**アクション**」、「**エクスポート**」の順に選択します。

現在の検索に基づいて行がエクスポートされます。

2. ダウンロードの指示に従い、ファイルを Excel ワークシートとしてダウンロードします。

設定の削除

上書きされた設定は削除できます。システムで作成された設定は削除できません。

設定を削除するには:

1. 「設定」表から、削除する設定行を選択します。
2. 「**選択した項目の削除**」をクリックするか、「**アクション**」、「**選択した項目の削除**」の順に選択するか、設定を右クリックして「**選択した項目の削除**」を選択します。
3. 確認のプロンプトで「**はい**」をクリックします。

B

パフォーマンスの最適化

この章では、Oracle Hyperion Financial Management アプリケーション、データベース管理および一般的なオペレーティング・システム の概念をよく理解していることを前提としています。実際の実装および環境は、ビジネス要件、Financial Management のデータセット、ネットワーク・トポロジおよびハードウェアの使用状況に基づいて大きく異なります。したがって、このようなガイドラインを自身の実装にどのように適合させるかを検討する必要があります。テスト結果とパフォーマンスの数値はすべて、チューニングの概念を示すための例としてのみ使用されます。

▲ 注意:

不適切な設定と構成は、Financial Management の動作を妨げる可能性があります。

チューニング設定を実装する前に、エンドツーエンド・パフォーマンス・テストを実行してデフォルトの構成に対するベースライン・パフォーマンス・データを取得し、チューニング設定を増分変更してからパフォーマンス・データを収集することをお勧めします。構成の変更によってシステム・パフォーマンスが向上し、低下しないことを確認する必要があります。

サポートされているプラットフォームおよびコンポーネントは、Oracle Enterprise Performance Management System サポート対象プラットフォームのマトリックスを参照してください: [Oracle EPM サポート対象プラットフォーム・マトリックス](#)。

パフォーマンスの概要

Oracle Hyperion EPM System Performance の概要

Oracle Hyperion EPM System のパフォーマンスを最大化するには、すべてのコンポーネントをモニター、分析およびチューニングする必要があります。この項では、パフォーマンスをモニターするために使用できるツールと、Oracle Hyperion Financial Management のパフォーマンスを最適化するための技法について説明します。

一般的な EPM インストール・ディレクトリの参照

この章では次のディレクトリ参照を使用します。

ミドルウェア・ホーム

ミドルウェア・ホームは、Oracle WebLogic Server ホーム、および必要に応じて 1 つ以上の Oracle ホーム (EPM Oracle ホームを含む) で構成されています。デフォルトのインストール・ディレクトリは Oracle\Middleware です。この章全体を通して、ミドルウェア・ホームの場所を `MIDDLEWARE_HOME` と呼びます。

EPM Oracle ホーム

Oracle ホームには、特定の製品をホストするのに必要なインストール・ファイルが含まれます。Oracle ホームは、ミドルウェア・ホームのディレクトリ構造内にあります。デフォルトの EPM Oracle ホームの場所は `MIDDLEWARE_HOME\EPMSys11R1` です。この章全体を通して、EPM Oracle ホームの場所を `EPM_ORACLE_HOME` と呼びます。

EPM Oracle インスタンス

さらに、構成時に、一部の製品は構成中に定義された EPM Oracle インスタンスにコンポーネントをデプロイします。EPM Oracle インスタンスのデフォルトの場所は `MIDDLEWARE_HOME\user_projects\epmsystem1` です。この章全体を通して、EPM Oracle インスタンスの場所を `EPM_ORACLE_INSTANCE` と呼びます。

Financial Management のレコードおよびサブキューブ

Oracle Hyperion Financial Management のレコードは、ディメンション・メンバーの特定の交差についてすべての基本期間のデータを保持します。

サブキューブは、次のディメンションの同じメンバーを共有するすべてのセルで構成されます。

- シナリオ
- 年
- エンティティ
- 値

サブキューブには、通貨サブキューブとノード・サブキューブの 2 タイプがあります。これらのタイプのサブキューブは、エンティティおよび値のディメンション・メンバーの使用方法において異なります。

- 通貨サブキューブには、適用可能なノード以外の値ディメンション・メンバーを共有するセルが含まれます。通貨サブキューブの場合、エンティティ・メンバーの親は関係ありません。適用可能なノード以外の値ディメンション・メンバーは次のとおりです。
 - ユーザー定義の通貨のメンバー。ユーザー定義の通貨ごとに、3 つ組の値ディメンション・メンバーが 1 つ存在します。たとえば、アプリケーションに USD という通貨がある場合、その通貨の 3 つ組の値ディメンション・メンバーは USD、USD Adjs および USD Total となります。3 つ組は、引き続き 3 つの別個のサブキューブと見なされます。
 - エンティティのデフォルト通貨を指す 3 つ組。この 3 つ組は、<Entity Currency>、<Entity Curr Adjs> および <Entity Curr Total> の値メンバーで構成されます。
 - [None] 値メンバー。

 ノート:

親エンティティのデフォルト通貨(<Parent Currency>、<Parent Curr Adjs>および<Parent Curr Total>)を指すノード以外の値ディメンション・メンバーは、通貨サブキューブとは関係ありません。

- ノード・サブキューブには、共通ノードの値ディメンション・メンバーを共有するセルが含まれます。ノード・サブキューブの場合、親と子の両方のエンティティ・メンバーを指定する必要があります。ノードの値ディメンション・メンバーは次のとおりです。
 - [Contribution Total]
 - [Contribution Adjs]
 - [Contribution]
 - [Elimination]
 - [Proportion]
 - [Parent Total]
 - [Parent Adjs]
 - [Parent]

サブキューブは連結を目的とした自然単位のデータであるため、Financial Management の様々な場所で、データの移動および処理がサブキューブ・ベースで実行されます。

Financial Management のチューニングに関する推奨事項

Financial Management のパフォーマンス・チューニングは反復プロセスです。

 ノート:

チューニングは特定の本番ワークロードに対して実行する必要があります。ワークロードが Oracle Application Testing Suite (OATS)などのロード生成ツールによって生成された場合にチューニングを実行できます。

この項では、パフォーマンス・チューニングをすぐに開始できるように次のような領域をいくつか説明します。

- オペレーティング・システム・パラメータのチューニング
- HTTP Server パラメータのチューニング
- Oracle Database パラメータのチューニング

 ノート:

このリストはパフォーマンス・チューニングを開始するのに便利な手段ですが、チューニング対象領域の包括的なリストであるというわけではありません。チューニングによってパフォーマンスが向上する可能性がある場所を把握するには、実装内で特定のパフォーマンスの問題をモニターおよび追跡する必要があります。

パフォーマンスの問題の診断

パフォーマンスの問題が発生した場合、修正処理を実行する前にまず原因を特定することが重要です。問題について詳しい分析が実行されるまでパフォーマンス関連のパラメータ設定を変更したり、他のアクションを実行することはお勧めしません。

モニタリング・ツールの使用

診断プロセスの一環としてパフォーマンス・データを収集するには、モニタリング・ツールを使用することをお勧めします。Oracle Hyperion Financial Management アプリケーション・サーバー、Web サーバー、データベース・サーバーおよびネットワーク・レイヤーのモニタリングは、有用なパフォーマンス・データを提供します。

Linux 環境では、OS Watcher Black Box などのツールを使用してオペレーティング・システムおよび Financial Management プロセスをモニターできます。

Windows 環境では、Microsoft Performance Monitoring などのツールを使用して Financial Management アプリケーション・プロセスのパフォーマンスをモニタリングできます。Financial Management アプリケーション固有のカウンタを収集するようにパフォーマンス・モニターを構成するステップは、次の URL を参照してください:

パフォーマンス・モニターの構成

モニターするパフォーマンス・カウンタは次のとおりです:

サブシステム	カウンタ	ガイダンス
メモリー	Memory: Free System Page Table Entries	警告: 空きシステム・ページ表エントリ数が 8,000 未満の場合。 クリティカル: 空きシステム・ページ表エントリ数が 5,000 未満の場合。
メモリー	Memory: Available Mbytes	インストールされている物理メモリーの 20%から 25%を下回る必要があります。このような場合、ページング・アクティビティを注意深くモニターします。
メモリー	Memory: Page Reads/sec	このカウンタは常に 1,000 未満である必要があります。
プロセッサ	Processor: % Processor Time: _Total	合計プロセッサ稼働率は約 70%から 80%を下回る必要があります。
プロセッサ	Processor: % Processor Time: (N)	各プロセッサ・インスタンスは約 70%から 80%の稼働率を下回る必要があります。

サブシステム	カウンタ	ガイダンス
ディスク	PhysicalDisk: Avg. Disk sec/Transfer	約 25 ミリ秒を下回る必要があります。 一般ルール: Avg. Disk Seconds/Transfer(ディスク待機時間カウンタ)が 25 ミリ秒を大幅に上回る場合、ディスク・サブシステムは異常かつボトルネックです。このカウンタは問題の解決方法を示しません。問題があることのみを示します。
ディスク	PhysicalDisk: Average Disk Queue Length	平均値はディスクのスピンドル数を下回る必要があります。SAN を使用している場合、このカウンタを無視して待機時間カウンタの PhysicalDisk\Average Disk sec/Read および PhysicalDisk\Average Disk sec/Write に的を絞ります。
ディスク	PhysicalDisk: Average Disk sec/Read	平均値は 20 ミリ秒を下回る必要があります。スパイク(最大値)は 50 ミリ秒以下にしてください。
ディスク	PhysicalDisk: Average Disk sec/Write	平均値は 20 ミリ秒を下回る必要があります。スパイク(最大値)は 50 ミリ秒以下にしてください。
ネットワーク	Network Interface: Bytes Total/sec	100Mbps/秒のネットワーク・インターフェース・カード(NIC)の場合、6-7MB/秒を下回る必要があります。1000Mbps/秒の NIC の場合、60-70MB/秒を下回る必要があります。
ネットワーク	Network Interface: Packets Outbound Errors	常にゼロ(0)である必要があります。

プロセス	カウンタ/ガイダンス
Oracle Hyperion Financial Management: インスタンス: 1 JHsxServer 2. XFMDDataSource	<p>Process: % Processor Time - プロセスのプロセッサ稼働率は 90%を下回る必要があります。</p> <p>Process: Private Bytes - 特定のプロセス専用割り当てられたバイト数をレポートします。値はリークしているプロセスに対して上昇する傾向があります。</p> <p>Process: Working Set - プロセスに割り当てられた共有バイト数およびプライベート・バイト数をレポートします。値はリークしているプロセスに対して上昇する傾向があります。</p> <p>Process: Page Faults/sec - プロセスに起因するフォルト(ハードおよびソフトのフォルト)の合計数をレポートします。値はリークしているプロセスに対して上昇する傾向があります。</p> <p>Process: Page File Bytes - ページング・ファイルのサイズをレポートします。値はメモリー・リーク時に上昇する傾向があります。</p> <p>Process: Handle Count - オブジェクト用に開かれたアプリケーションで作成されるハンドルの数をレポートします。ハンドルは、アクセスする必要があるリソースの識別にプログラムで使用されます。このカウンタの値はメモリー・リーク時に上昇する傾向があります。</p> <p>Process: Virtual Bytes - プロセスで使用されている仮想アドレス空間の現在のサイズ(バイト単位)。perfmon プロセスの仮想バイト数が急速に増えてメモリーを決して解放しないと、アプリケーションのメモリー・リークを示します。</p> <p>Process: Virtual Bytes Peak - プロセスが常時使用している仮想アドレス空間の最大サイズ(バイト単位)。perfmon プロセスの仮想バイト数が急速に増えてメモリーを決して解放しないと、アプリケーションのメモリー・リークを示します。</p> <p>Process: Pool Nonpaged Bytes - 非ページ・プールのサイズ(バイト単位)。システム・メモリー(物理メモリー)の領域。</p>

リモート診断エージェント(RDA)の使用

リモート診断エージェント(RDA)とは、Perl プログラミング言語で記述されたエンジンによって実行される一連のコマンドライン診断スクリプトです。収集されたデータからは、問題の診断に役立つその環境の全体像がわかります。

RDA を実行すると、特に Oracle Hyperion Financial Management アプリケーションでサブキューブのサイズを決定するのに有用です。RDA は My Oracle Support の Web サイトから入手できます。スタート・ガイドは、次のナレッジ・ベースの記事を参照してください:

[ドキュメント 1100612.1](#)

参照アプリケーションの使用

参照アプリケーションとは、パフォーマンスの問題の診断に使用するアプリケーションです。参照アプリケーションでは、環境が適切にチューニングされたかどうかを判定するのに有用な内部タイミングとの比較ができる一連のタスクが実行されます。一部のタイプの参照アプリケーションを実行すると、問題の特定に役立ちます。しかし、Oracle Hyperion Financial Management のすべてのパフォーマンス特性を明らかにできるアプリケーションはありません。あるアプリケーションでは良好なパフォーマンスを示しても、別のアプリケーションでは低いパフォーマンスを示すことがとてもよくあります。組み込まれたパラメータには、データ量、サブキューブごとのレコード分布、エンティティの構造、通貨の数などがあります。

オペレーティング・システム・パラメータのチューニング

Windows パラメータのチューニング

Windows プラットフォームでは通常、デフォルトの TCP/IP 設定で十分です。ただし、高負荷の下では、*MaxUserPort* および *TcpTimedWaitDelay* の調整が必要になる場合があります。これらのパラメータは、アプリケーションが要求するユーザー・ポートの可用性を決定します。

パラメータ	デフォルト値	推奨値
TcpTimedWaitDelay このパラメータは、アプリケーションが TCP 接続を閉じた後にポートを再利用するために OS が待機する時間を制御します。デフォルト値は 4 分です。ユーザー負荷が高いと、このような制限を超えて、アドレス使用中の接続例外が発生することがあります。 このパラメータは Windows レジストリの次の場所で設定します: HKLM\System\CurrentControlSet\Services\Tcpip\Parameters 値の名前: TcpTimedWaitDelay 値の種類: DWORD データ: 30 (10 進数)	240	30

パラメータ	デフォルト値	推奨値
MaxUserPort 値の名前: MaxUserPort 値の種類: DWORD データ: 65534 (10 進数) Windows2008 以降では、指定した数のポートを設定するために次のコマンドを実行する必要があります。 <pre>netsh int ipv4 set dynamicportrange protocol=tcp startport=10100 numberofports=55433 store=persistent</pre> 指定のコマンドは、 support.microsoft.com の次の記事に基づいています。 動的ポート範囲 Windows の設定	5000	65534

Web サーバーのチューニング

Oracle Hyperion Financial Management Web サーバーと他の EPM コンポーネントの主なチューニング・ステップは、最初の EPM System 製品とともにコンピュータにインストールされ、コンピュータにインストールされるすべての製品の構成に使用される EPM System コンフィグレータに組み込まれています。詳細は、*Oracle Enterprise Performance Management System* インストレーションおよび構成ガイドを参照してください。

HFM Web のチューニング

HFM Web のタイムアウト・パラメータ

EPM 構成ファイル	デフォルト値	推奨値
Web サーバー・プラグインと WebLogic Oracle HTTP サーバーまたは IIS を使用してリクエストを WebLogic にプロキシする場合、次の設定が適用されます。使用される Web サーバー・タイプの選択肢は EPM System コンフィグレータから設定されます。パラメータは大文字と小文字が区別され、手動で追加する必要があります。		

EPM 構成ファイル	デフォルト値	推奨値
WLIOTimeoutSecs プラグインが WebLogic Server からの要求に対する応答を待機する時間の長さを定義します。	7,200	7,200
WLSocketTimeoutSecs 接続時のソケットのタイムアウトを秒単位で設定します。	600	600
(OHS Web サーバー・プラグイン) EPM_ORACLE_INSTANCE\httpConfig\ohs\config\OHS\ohs_component\mod_wl_ohs.conf hfmadf インスタンスに適用されます: 例: <pre> <LocationMatch ^/hfmadf> SetHandler weblogic-handler WeblogicCluster<WLCluster>:<port> WLIOTimeoutSecs 7200 Idempotent OFF WLSocketTimeoutSecs 600 </LocationMatch> </pre>		
(IIS Web サーバー・プラグイン) EPM_ORACLE_INSTANCE\httpConfig\VirtualHosts\hfmadf\iisproxy.ini WlForwardPath=/hfmadf PathTrim= WebLogicHost=<host> WebLogicPort=<port> KeepAliveEnabled=true KeepAliveSecs=20 WLIOTimeoutSecs=3600 Idempotent=OFF WLSocketTimeoutSecs=750		

EPM 構成ファイル	デフォルト値	推奨値
<p>WebLogic スタック・スレッド(WebLogic 管理 コンソール)</p> <p>HFM アプリケーション・プロセスが ロード・ファイルの処理でビジー状態 の場合、メタデータのロードなどの長 時間実行タスクがスタック状態のよ うに見えることがあります。このよ うな問題が発生した場合は、スタッ ク・スレッド時間の値を増やします。</p> <p>WebLogic コンソールから次の設定 を変更します。</p> <p>「ロックして編集」を選択します。</p> <p>「サーバー」を選択して 「HFMWeb(N)」をクリックします。</p> <p>「チューニング」タブを選択します。</p> <p>「スタック・スレッド最大時間」の値 を 1200 に変更します。</p> <p>「スタック・スレッド・タイマー間隔」 の値を 1200 に変更します。</p> <p>「変更のアクティブ化」を選択します。</p> <p>HFMWeb0 サーバーを再起動する必 要があります。</p> <p>HFMWeb(N)は HFMWeb0、 HFMWeb1 など、HFM Web サー ーがデプロイされている数と更新 しているサーバーによって決まりま す。</p>	<p>スタック・スレッド最大時間 600</p> <p>スタック・スレッド・タイマー間隔 60</p>	<p>1200</p> <p>1200</p>

HFM Web のチューニング・パラメータ

この項では、その他の HFM Web UI のチューニング・パラメータについて説明します。

チューニング・パラメータと場所

パラメータ	デフォルト値	推奨値
-------	--------	-----



**HFM ADF Web アプリケーションの
Java ヒープ・サイズ(Xms および
Xmx)**

HFM ADF Web アプリケーションの
各管理対象インスタンスで使用可能
なヒープの量。

Windows

HFM ADF Web は Windows サービスとしてインストールされます。次の場所にある **JVMOptionXX** オプションを編集すると、ヒープのパラメータをチューニングできます。

HKLM\SOFTWARE\Hyperion
Solutions\HFMWeb (N)
\HyS9FinancialMangementWeb

HFMWeb(N)は HFMWeb0、
HFMWeb1 など、HFM Web サー
バーがデプロイされている数と更新
しているサーバー・インスタンスによ
って決まります。

コンパクト・デプロイメントでは、す
べての EPM 製品に対して
EPMSystem(N) Web アプリケーシ
ョンが 1 つしかないため、HFM ADF
Web を個別にチューニングするこ
とはできません。構成オプションを含
む Windows レジストリの場所は、
HKLM\SOFTWARE\Hyperion
Solutions\EPMServer (N)
\HyS9EPMServer です。

Xms は初期ヒープ・サイズを設定し、**128m**
Xmx と同じサイズに設定する必要が
あります。

8192m

Windows

JVMOptionXX で-Xms を探します
データ型は REG_SZ です
データ値: -Xms<size>

パラメータ	デフォルト値	推奨値
Xmx はヒープの最大サイズを設定します。32 ビット・システムの場合は 1.8 GB 以下に設定します。64 ビットの場合は <i>使用可能な物理メモリの 75%以下に設定することをお勧め</i> します。 Windows JVMOptionXX で-Xmx を探します データ型は REG_SZ です データ値: -Xmx<size>	4096m	8192m

Financial Management アプリケーションのチューニング

一般に、単一のアプリケーション・サーバー上で一度に複数の Oracle Hyperion Financial Management アプリケーションを実行すると、すべてのアプリケーションのパフォーマンスに影響を及ぼします。アイドル状態のアプリケーションの稼働にはデータベース接続および CPU 時間が必要であるため、他のアプリケーションがアイドル状態であっても、一度に 3 つまたは 4 つを超えるアプリケーションを実行しようとしてください。

一般的にチューニングされた Financial Management の設定

構成設定の詳細なリストは、このガイドの付録 A を参照してください。構成設定を参照してください。

MaxNumDataRecordsInRAM

一般に、RAM で保持するレコード数を決定するため、MaxNumDataRecordsInRAM が最も重要な設定です。2 つの条件 NumMinutesBeforeCheckingLRU または NumCubesLoadedBeforeCheckingLRU のいずれかに達した場合に、メモリー内のレコード数がチェックされます。使用可能な構成設定を参照してください。

範囲: 範囲にハード・リミットはありません。

デフォルト値: 30,000,000

ノート:

RAM 内の合計レコード数がこの値を上回ると FreeLRU がコールされ、Oracle Hyperion Financial Management サーバーのメモリーを一部解放するために、メモリーからレコードが解放されます。「FreeLRUCachesIfMoreRAMIsNeeded によりデータ・キューブが解放されました」という情報メッセージがログに記録されます。

MinDataCacheSizeInMB

この値を高い数値に設定することによって、データ・キャッシュ増加の試行回数を減らすことができるため、メモリーの断片化が削減されます。通常、データ・キャッシュは必要に応じて増加され、一度に最大 25MB 増加します。

範囲: 範囲にハード・リミットはありません。

デフォルト値: 2,000

MaxDataCacheSizeInMB

この設定は、セル値およびセル・ステータスを格納するために Oracle Hyperion Financial Management アプリケーション・サーバーによって割り当てられるメモリーの最大量を制御します。システムでさらにメモリーが必要な場合、セル値およびセル・ステータスは LRU ロジックに基づいてディスクにページ・アウトされます。回避策として、キャッシュ・サイズを増します。

範囲: 500 以上の値にする必要があります。

デフォルト値: 4,500

ノート:

ページングが原因で連結操作のパフォーマンスが低下する場合は、**MaxDataCacheSizeInMB** を増やしてページングを最小限に抑えることができます。この値は **MaxNumDataRecordsInRAM** によって許可される合計メモリー使用量よりも大きい値にする必要があるため、セルが不必要にディスクにページ・アウトされません。**MaxDataCacheSizeInMB** 設定が小さすぎると、データ・レコードを格納するためのメモリーが不足してページングが開始されるため、システム・パフォーマンスが低下します。

MaxNumCubesInRAM

この設定は任意の時点でメモリーに保持されるキューブの数を制御します。2つの条件 **NumMinutesBeforeCheckingLRU** または **NumCubesLoadedBeforeCheckingLRU** のいずれかに達した場合に、メモリー内のキューブ数がチェックされます。この設定を小さくすると、疎アプリケーション(多くのエンティティを含むが、エンティティごとのレコードが相対的に少ないアプリケーション)の高いメモリー使用量を軽減できます。この設定は、連結やデータ取得など、すべてのデータ操作に影響を及ぼします。

範囲: 100-500,000

デフォルト値: 60,000

IcmSystemReportTranslationMode

通貨換算を必要とする内部取引照合システム・レポートの実行時、デフォルトでは換算された値がデータベースに書き込まれます。これにより、レポートの実行時間が長くなり、データベース・サイズが大幅に増加する可能性があります。この設定を使用すると、換算された値の処理方法を制御できます。

範囲: 0、1、2

デフォルト値: 0

有効なデータ・オプション:

- 0 - デフォルトの動作です。換算された値がデータベースに書き込まれます。
- 1 - オンザフライ計算ではルールでサブ換算を使用しますが、換算された値はデータベースに書き込まれません。ノート: CN ステータスの親エンティティも換算されます。
- 2 - オンザフライ計算ではデフォルト換算のみを実行します(サブ換算ルールなし)。変更はデータベースに書き込まれません。ノート: CN ステータスの親エンティティは換算されません。

NumConsolidationThreads

この設定は、連結のマルチスレッドを、アプリケーション・サーバーごとに制御します。

範囲: 範囲にハード・リミットはありません。値は正数である必要があります。

デフォルト値: 8

ノート:

値を下げると、システム・リソースの使用量が制限されて、連結のパフォーマンスが下がります。この値を上げると、CPU 使用率が高くなり、他のコンポーネントのパフォーマンスに影響を及ぼす可能性があります。

ヒント:

NumConsolidationThreads を 4 から 8 に増加した影響を評価するため、テストを複数回実行しました。次の結果は、1 か月の連結時間が短くなったことを示しています。

取引	92 ユーザー NumConsolidationThreads = 4 平均応答時間(秒)	92 ユーザー NumConsolidationThreads = 8 平均応答時間(秒)
01_Run_Consolidation_A_1105	10.11	3.22
02_Run_Consolidation_A_0005	16.15	9.47
03_Run_Consolidation_A_2205	7.75	3.19
04_Run_Consolidation_A_3305	18.67	9.17

取引	92 ユーザー NumConsolidationThreads = 4 平均応答時間(秒)	92 ユーザー NumConsolidationThreads = 8 平均応答時間(秒)
05_Run_Consolidation_B_11 05	8.21	3.14
06_Run_Consolidation_B_00 05	13.26	9.27
07_Run_Consolidation_B_22 05	7.69	6.20
08_Run_Consolidation_B_33 05	18.29	9.41
09_Run_Consolidation_C_00 05	30.59	22.08

 ノート:

テスト実行時の Financial Management アプリケーション・サーバーの CPU 使用率は、さらに厳しいワークロードへの対応に使用できる容量があって許容範囲内でした。

 ノート:

この値を増やす前に、すべてのアプリケーション・サーバーで必ずすべての設定を同じにします。実行中の同時連結とキューで待機中の連結を比較する場合、合計連結時間が実際に悪化する前に特定のサーバーで現在の連結がいくつ実行されるかを確認するためにテストする必要があります。

MaxNumConcurrentConsolidations

この設定は、可能な同時連結の数を、アプリケーション・サーバーごとに制御します。値を超えて実行される連結は、スケジュールされた連結としてキューに入ります。

範囲: 1 - 制限なし

デフォルト値: 8

シナリオ: 3つの Oracle Hyperion Financial Management アプリケーション・サーバーがある場合、各サーバーでは最大 8 つの同時連結を実行できますが、NumConsolidationsAllowed のデフォルト値により、1 つのアプリケーションに対して 3 つのサーバーでは実行中の同時連結が合計 8 つのみに制限されます。例: ユーザーが ServerA で 6 つの連結を送信した後、ServerB でさらに 2 つの連結を送信します。この 8 つの連結はすべて実行されます。それと同時に、ユーザーが ServerC で次の連結を送信した場合、これは前の 8 つのいずれかが終了するまで実行されません(「タスクの実行」ページで、ステータスが「開始(スケジュールによる)」になります)。

 ノート:

この値を増やす前に、すべてのアプリケーション・サーバーで必ずすべてのレジストリ設定を同じにします。実行中の同時連結とキューで待機中の連結を比較する場合、合計連結時間が実際に悪化する前に特定のサーバーで現在の連結がいくつ実行されるかを確認するためにテストする必要があります。

NumThreadsToUseWhenUpdatingCalcStatusSystemWasChanged

メタデータのロード時に、エンティティ 階層が変更(メンバーが移動、追加または削除)されると、使用中のメンバーはその子または親と整合性がなくなる可能性があります。Oracle Hyperion Financial Management では、シナリオおよび年の各組合せの算出ステータスを確認する必要があります。この設定を使用すると、単一のスレッドではなく複数のスレッドで算出ステータスを確認および更新できるようになります。

範囲: 1 - 制限なし

デフォルト値: 16

大規模アプリケーション用の Financial Management のメモリー設定

次の表に、使用可能な物理メモリーに応じたパラメータの推奨値を示します。これは、Oracle Hyperion Financial Management がマシンで実行されている唯一のメモリー集約型プロセスであり、Financial Management アプリケーションを 1 つしか実行していないということを前提としています。

 ノート:

複数のアプリケーションがアクティブの場合、サーバーに搭載されている物理メモリーの合計をアプリケーションの数で除算して、各アプリケーションの使用可能な物理メモリーを求めます。

使用可能な物理メモリー	MaxNumDataRecordsInRAM	MaxDataCacheSizeinMB
8GB	10,000,000	1,500
16GB	30,000,000	4,500
32GB	60,000,000	9,000
64GB	100,000,000	15,000
128GB	200,000,000	30,000

アプリケーション固有の設定

以前は環境レベルのみの設定であった一部の設定が、アプリケーション・レベルまで拡大されています。通常、アプリケーション・レベルでの Oracle Hyperion Financial Management のチューニングは、複数のアプリケーションを同じ Financial Management アプリケーション・サーバー上で実行する必要があるのに、頻繁に使用

されるアプリケーションが 1 つのみの場合に役立ちます。その場合、頻繁に使用されるアプリケーションをチューニングしてサーバーのメモリーの大部分を使用できるようにし、残りのアプリケーションではデフォルト値を使用してアプリケーションが利用できるメモリー量を制限します。Financial Management アプリケーション固有またはサーバー固有の設定を使用するには、「構成設定」の章での説明に従って上書き手順を使用します。値の上書きを参照してください。

 ノート:

上書きの優先順位は次のようになります:

1. アプリケーション固有設定が存在せず、インストール・レジストリ設定が存在する場合、インストール・レジストリ設定が使用されます。
2. アプリケーション固有設定が存在しない場合、サーバー・キーに定義された設定が使用されます。
3. アプリケーション固有設定もサーバー設定も定義されていない場合、デフォルト値が使用されます。設定およびデフォルト値については、[構成設定](#)を参照してください。

次の設定はアプリケーション固有設定でオーバーライドできます:

- AllowOverlappingConsolidationOverride
- AutoClearDeadTasks
- AutoClearEAFlatfileTasksAfterSeconds
- EnableRulesLogging
- EnableRunningTasksMaskUserNames
- DSStartupOption
- MaxDataCacheSizeInMB
- MaxNumConcurrentConsolidations
- MaxNumCubesInRAM
- MaxNumDataRecordsInRAM
- MaxNumRetriesOfBaseLevelCalculation
- MinDataCacheSizeInMB
- NumConsolidationThreads
- NumCubesLoadedBeforeCheckingLRU
- NumDataLoadsAllowed
- NumEAExportsAllowed
- NumEAThreads
- NumMinutesBeforeCheckingLRU
- NumThreadsToUseWhenUpdatingCalcStatusSystemWasChanged

Financial Management アプリケーション・サーバーのチューニング

Oracle Hyperion Financial Management アプリケーション・サーバーをチューニングする場合は、代表的なユーザーの同時実行性で主要ユーザー・アクティビティを測定するベースライン・テストから始める必要があります。レポート作成および Oracle Smart View for Office のユーザー・アクティビティを連結アクティビティを区別するために、よく Financial Management クラスタを複数使用しますが、その場合、様々なチューニングによる変更が、測定されるユーザー・タスクに基づいてサーバーごとに異なる影響を及ぼすことも考えられます。たとえば、主にレポート作成に使用される Financial Management アプリケーション・サーバーでは、NumConsolidationThreads を増やしても何の利点も見られませんが、連結を数多く実行しているサーバーでは連結時間に改善が見られます。同様に、主にレポート作成に使用されるアプリケーション・サーバーでは、すべてのレコードをメモリーに保持するのに十分高い値に MaxNumDataRecordsInRAM が設定されている場合、レポート作成を繰り返すとレスポンス時間が短くなる可能性があります。連結を数多く実行するサーバーでは、連結時間の改善は見込めません。アプリケーション・サーバーに付与する役割を決定するときに考慮する別のポイントは、CPU 速度、CPU コア数、RAM 容量および RAM 速度です。集約型ルールが設定された Financial Management アプリケーションを 1 つ実行し、主に連結の実行を専門とするアプリケーション・サーバーは通常、クロック速度が遅い CPU/コアの数を増やすのではなく、CPU クロック速度を速くして少なくとも 8 個のコアを使用すると、最高のパフォーマンスを示します。

Financial Management のチューニング例

この例は、CPU が 8 個、RAM が 64GB で、3 つの HFM アプリケーションをホストし、その 3 つのうち 1 つのみ(このアプリケーションを CORP と呼びます)が頻繁に使用される HFM アプリケーション・サーバーのチューニングを示しています。Oracle Hyperion Financial Management アプリケーション・サーバーは連結とレポート作成の両方のアクティビティに使用されます。

Financial Management のインストール時に、標準の構成設定がデフォルトで指定されます。「設定」モジュールでは、これらのデフォルト設定を表示し変更することができます。

この例では、MaxNumDataRecordsinRAM の合計値を 60,000,000、MaxDataCacheSizeinMB の合計値を 9,000 に維持する必要があります。3 つのアプリケーションのうち 2 つは頻繁に使用されないため、その 2 つのアプリケーションではデフォルト値を使用できるようにし、サーバー・リソースを最大限に利用するように CORP アプリケーションをチューニングします。

2 つのアプリケーションでデフォルト値を使用するため、MaxNumDataRecordsinRAM のデフォルト値は 30,000,000 です。つまり、CORP では MaxNumDataRecordsinRAM を 40,000,000 に設定できます。

2 つのアプリケーションでデフォルト値を使用するため、MaxDataCacheSizeinMB のデフォルト値は 4,500 です。つまり、CORP では MaxDataCacheSizeinMB を 6,000 に設定できます。

この例で CORP 用のチューニングを特別に検討するもう一方の値は次のようになります:

MinDataCacheSizeInMB – MaxDataCacheSizeinMB の値の半分(3,000)に設定します。

CORP についてアプリケーション・レベルでチューニングし、他の 2 つのアプリケーションでデフォルト値を使用できるようにしているため、CORP でデフォルト値とは異なる値を使用するために上書きを作成します。設定に対する更新は次のようになります:

MaxNumDataRecordsinRAM (40,000,000)

MaxDataCacheSizeinMB (6,000)

MinDataCacheSizeinMB (4,000)

これらの構成設定を更新するには、「ナビゲート」、「管理者」、「連結管理」、「システム」の順に選択し、「設定」を選択します

すべての構成設定の詳細は、[使用可能な構成設定](#)を参照してください。

大きなアプリケーションの設定の詳細は、[大規模アプリケーション用の Financial Management のメモリー設定](#)を参照してください。

アプリケーション・データベースのメンテナンス

Oracle Hyperion Financial Management とともに Oracle/SQL Server データベースを使用する場合は、次のベスト・プラクティスをお勧めします。

Financial Management の表、<appname>_DATA_AUDIT、<appname>_TASK_AUDIT および HFM_ERRORLOG の場合、次のベスト・プラクティスを実施することをお勧めします:

四半期ごと - 監査ログの確認、アーカイブおよび削除する業務。

半年ごと - システム・メッセージのアーカイブおよび表の切捨て。

ヒント:

これらの表が推奨レコード数(500,000)を超えて大きくなった場合にアクションを実行できるようにアラートを設定します。大規模な監査表によって Financial Management のパフォーマンスが重大な影響を受ける可能性があります。

基本的な設計上の考慮事項

- データ監査機能がビジネス要件に含まれていない場合、データ監査を無効にすることをお勧めします。データ監査表が 10GB を超えるアプリケーションでは、パフォーマンスの低下が認められています。

 ヒント:

メンバーの監査を無効にするには、メタデータ・ファイルですべてのメンバーについて **EnableDataAudit** 属性を **N** に変更します。

- 設計の悪いルール危険(ルールによるデータ爆発など)を回避するため、本番環境にロードする前にルールをテストする必要があります。システム・パフォーマンスを許容範囲に収めるには効率的なルールが重要です。
- **Oracle Hyperion Financial Management** では、サブキューブが **RAM** に格納されている間にそれを介してすべての処理を実行するため、サブキューブが大きいほどパフォーマンスへの影響が大きくなります。可能なかぎりサブキューブのサイズを最小化してください。たとえば、最適なシステム・パフォーマンスを確保するために、サブキューブが基本レコードの制限 **200,000** 件を超えないようにします。
- **Financial Management** アプリケーションでゼロをロードしたり、ゼロを計算したりすることはお薦めしません。ゼロはデータとして保管されるため、データベース・サイズが増えてパフォーマンスに影響する場合があります。**Financial Management** には、**1000** などの数値情報のみ格納してください。**Oracle Hyperion Financial Management ユーザーズ・ガイド**のデータ・ファイルに関する項を参照してください。
- **Financial Management** が正常にシャット・ダウンされない場合、再起動時に一時ファイルが残っていることがあります。最適なパフォーマンスを確保するには、**Financial Management** を起動する前に、**Financial Management Server** の作業フォルダから*.db.*の付くファイル名をすべて削除することをお薦めします。
- データ・グリッドまたはプロセス単位に複数のドキュメントを添付する場合、1つのセルに添付するドキュメントは**3**つ以下にすることをお薦めします。データベースに与えるパフォーマンスの影響を制限するために、各ドキュメントのサイズは**100K** 未満にする必要があります。

 ヒント:

アプリケーションを作成するときに、ユーザーごとにドキュメントの添付のサイズ制限や最大数を設定できます。この制限は、アプリケーション・メタデータの **AppSettings** 属性で設定できます。

- 「すべてを連結」オプションは、適切な状況下でのみ使用します。このオプションを使用すると、**NoData** のエンティティがスキップされないため、連結パフォーマンスに重大な影響を及ぼす可能性があります。

 ヒント:

ロジックまたは連結が必要なエンティティのみが更新されるため、「連結(変更したデータのみ連結)」が最も効率的なオプションです。「データのあるものすべてを連結」オプションは、メタデータの変更後にシステム・ステータスを「**OK SC**」から「**OK**」に更新するのに便利です。詳細は、**Oracle Hyperion Financial Management ユーザーズ・ガイド**の連結オプションを参照してください。

Financial Management 用の Oracle 11g データベースのチューニング

はじめに

Oracle Hyperion Financial Management は、アプリケーション・データを格納するためにリレーショナル・データベースを必要とします。各 Financial Management アプリケーションには、表、インデックス、ストアド・プロシージャなどの一連のオブジェクトが含まれます。このようなオブジェクトの数とサイズは、ユーザーのデータセット、アプリケーションの設計およびレポート要件に応じて異なるため、データベースを設定するために具体的なルール・セットを指定することは困難です。この項では、Oracle データベースへのデプロイ時に発生する 2 つの最も一般的な問題について説明します：

- 必要な数のデータベース接続をサポートするためのメモリーが不足している Oracle DB
- レポート作成および連結時の低パフォーマンス

これらの問題はいずれも不適切な Oracle パラメータの設定および構成に端を発します。大きすぎるシステム・グローバル領域(SGA)を作成すると、ユーザーの接続およびアクティビティをサポートするための空き物理メモリーの量が制限されます。反対に、小さすぎる SGA を作成すると、余計なディスク・アクセスが発生してパフォーマンスが低下します。

この項では、一般的なデータベースをモニタリングしてパフォーマンスを最大化するために適切な初期化設定を決定するプロセスを見ていきます。データセット、ワークロードおよびアプリケーションの設計に対する変更に対応するには、このプロセスを定期的に繰り返す必要があります。

Oracle 11g では、データベースをモニタリングおよびチューニングするプロセスが以前のバージョンよりもずっと簡単になっています。Oracle Enterprise Manager (Data Control と Grid Control の両方)を使用して、データベース・パフォーマンスをモニター、診断およびチューニングすることをお勧めします。正確なインスタンスの統計を取得するには、Oracle データベースの自動メンテナンス・タスクを有効にすることをお勧めします。

データベース起動直後にチューニングを実行しないことが非常に重要です。その時点で、バッファ・キャッシュは空であり、統計は収集されていません。常に、Financial Management アプリケーションのアクティビティ 期間後にデータベースをテストしてチューニングします。

一般的なパフォーマンスの問題

低パフォーマンスの最も一般的な原因は、失効しているか欠落している表の統計です。Oracle Hyperion Financial Management アプリケーションでは、新しい表を作成するか、既存の表をクリアしてデータセット全体をオンザフライでリロードまたは置換できます。Oracle 10g および 11g のデフォルトでは、統計を更新する必要がある表をチェックする夜間ジョブが実行されます。これは、場合によっては十分でないことがあります。次のアクティビティは、スキーマ統計を更新する必要があることを DBA に通知する必要がある一般的なエンド・ユーザーのアクティビティです。

- 新しい HFM アプリケーションを作成およびロード
- 既存のシナリオのデータをクリアおよびリロード

- 新しいシナリオ/年を開いてデータをロードするには **Financial Management** のルールによって移入されたシナリオが含まれます。

シナリオ/年の最初の月のデータがロードされた後では、後続の期間にデータがロードされても表の統計に影響はありません。

Oracle 11g Database のチューニング・ガイドライン

Oracle 初期化パラメータ

データベース・パフォーマンスを向上させるために、様々な初期化パラメータを細かくチューニングできます。この項では、**Oracle** データベースで **Oracle Hyperion Financial Management** のパフォーマンスに大きく影響することがわかっているパラメータについて説明します。

CURSOR_SHARING

CURSOR_SHARING は、同じカーソルを共有できる **SQL** 文のタイプを決定します。このパラメータが **EXACT** に設定されている場合、同一のテキストを含む文のみが同じカーソルを共有できます。このパラメータが **FORCE** に設定されている場合、一部のリテラルが異なり、それ以外は同一である文が **SQL** カーソルを共有および再使用できます。ただし、リテラルが文の意味に影響する場合は除きます。テストにより、このパラメータを値 **FORCE** に設定すると、連結およびレポート作成のパフォーマンスが大幅に向上する可能性があることがわかっています。これは、このパラメータを **FORCE** に設定すると、**Oracle** データベースでは **SQL** 文の解析に費やす時間が短くなり、必要なメモリーが少なくなるためです。

11.1.2.2.300 より前のすべてのリリースに対する推奨設定: **FORCE**

11.1.2.2.300 以降のリリースに対する推奨設定: **EXACT**

MEMORY_TARGET

MEMORY_TARGET および **MEMORY_MAX_TARGET** の 2 つは、**Oracle Database** リリース **11g** の新しいパラメータです。この 2 つのパラメータは、**Oracle** データベースの自動メモリー管理の使用方法を決定します。システム上のメモリーを管理するために自動メモリー管理の使用をお勧めします。この 2 つのパラメータをゼロ以外の値に設定することで、自動メモリー管理が有効になり、システム・グローバル領域(**SGA**)とインスタンス・プログラム・グローバル領域(**PGA**)の間で必要に応じてメモリーを再分散してターゲット・メモリー・サイズにチューニングされます。結果として、次のパラメータのサイズが自動的に設定されます:

SGA_TARGET

SGA_MAX_SIZE

DB_CACHE_SIZE

SHARED_POOL_SIZE

LARGE_POOL_SIZE

JAVA_POOL_SIZE

STREAMS_POOL_SIZE

PGA_AGGREGATE_TARGET

ただし、LOG_BUFFER は自動メモリー管理の影響を受けないため、引き続き手動でサイズを設定する必要があります。LOG_BUFFER のチューニングはこの項で後述します。

MEMORY_TARGET は SGA および PGA の合計メモリー・サイズを指定するため、高いパフォーマンスを得るには、比較的高い値に設定する必要があります。Oracle Hyperion Financial Management の連結およびレポート作成はメモリー集約型のタスクであるため、十分なメモリーが必要です。32 ビットのオペレーティング・システムで使用可能なアドレス空間(通常 2GB または 3GB)に制限を設定することを考慮して、このパラメータを少なくとも 1.2GB に設定することをお勧めします。一般に、値が高いほど Oracle データベースのパフォーマンスは良好になるため、仮想アドレス空間が不足することなく可能なかぎり高い値にこのパラメータを設定してください。

推奨設定: 最小 1.2GB、通常(環境に応じて)1.2GB 超の値。

Oracle 11g インスタンスへの割り当てるメモリー量を決定する場合、次の項を確認してください: [Oracle Database リリース 11g のプロセス数を計算する方法](#)。

MEMORY_MAX_TARGET

MEMORY_MAX_TARGET は、DBA が MEMORY_TARGET パラメータに設定できる最大値を指定します。上限として機能するため、MEMORY_TARGET を誤って高すぎる値に設定することはできません。また、再起動せず実行時に MEMORY_TARGET を増す場合に備えて、Oracle データベース・インスタンスのメモリーを予約します。したがって、MEMORY_MAX_TARGET を MEMORY_TARGET の値以上にする必要があります。

推奨設定: MEMORY_TARGET の値以上

SGA_TARGET

SGA_TARGET は、すべての SGA コンポーネントの合計サイズを指定します。自動メモリー管理が有効になっており、SGA_TARGET がゼロ以外の値に設定されている場合、この値は SGA の最小サイズとして機能します。

推奨設定: 自動メモリー管理が有効になっている場合は 0。それ以外の場合は、[Oracle Database リリース 11g のメモリー設定を決定する方法](#)を参照してください。

SGA_MAX_SIZE

SGA_MAX_SIZE は、インスタンスの存続期間を通した SGA の最大サイズを指定します。このパラメータは、SGA_TARGET の上限を設定します。自動メモリー管理が有効になっている場合、Oracle データベースでは SGA_MAX_SIZE を超えて SGA コンポーネントの合計サイズを増やすことはできません。

推奨設定: 自動メモリー管理が有効になっている場合はデフォルト設定を使用してください。それ以外の場合は、[Oracle Database リリース 11g のメモリー設定を決定する方法](#)を参照してください。

PGA_AGGREGATE_TARGET

PGA_AGGREGATE_TARGET は、インスタンスに接続されたすべてのサーバー・プロセスで使用できる合計 PGA メモリーを指定します。自動メモリー管理が有効になっており、

`PGA_AGGREGATE_TARGET` がゼロ以外の値に設定されている場合、この値は `PGA` の最小サイズとして機能します。

推奨設定: 自動メモリ管理が有効になっている場合は 0。それ以外の場合は、[Oracle Database リリース 11g のメモリ設定を決定する方法](#)を参照してください。

LOG_BUFFER

`LOG_BUFFER` は、REDO ログ・ファイルへの REDO エントリのバッファリングに使用されるメモリ量(バイト単位)を指定します。REDO ログ・エントリには、データベース・ブロック・バッファに対して行われた変更の記録が含まれています。Oracle Hyperion Financial Management は更新頻度の高いトランザクション・システムであり、データベースは常にログ・バッファを使用します。ログ・バッファのサイズを適切に設定すると、データベース・パフォーマンスを向上させることができます。一般に、`LOG_BUFFER` の値を大きくすると、特にトランザクションが長い場合や多数の場合に、REDO ログ・ファイルの I/O が減ります。バッファが小さすぎると、バッファがクリアされるまで待機してから新しい更新が追加されるため、データベースのパフォーマンスを向上させるには、このバッファを正しく設定することが重要です。

推奨設定: 8MB から始めます。[Oracle Database リリース 11g のメモリ設定を決定する方法](#)を参照してください。

OPTIMIZER_MODE

`OPTIMIZER_MODE` は、インスタンスの最適化方法を選択するためのデフォルト動作を設定します。オプティマイザ・モードは、オンライン・アプリケーションのレスポンスを最適にするために `FIRST_ROWS` に設定するか、バッチ操作の合計実行時間を最小限に抑えるために `ALL_ROWS` に設定できます。Oracle Hyperion Financial Management では戻されたデータセット全体のみを処理するため、合計実行時間を最小限に抑える方が適切です。

推奨設定: `ALL_ROWS`

OPTIMIZER_INDEX_COST_ADJ

`OPTIMIZER_INDEX_COST_ADJ` を指定すると、アクセス・パスの選択するときに索引を使いやすくするか否か、オプティマイザの動作をチューニングできます。つまり、オプティマイザが全表スキャンよりも索引アクセス・パスを選択する傾向を強めたり弱めたりできます。このパラメータのデフォルト値は 100 パーセントで、その場合、オプティマイザは索引アクセス・パスを通常コストで評価します。しかし、Oracle Hyperion Financial Management のトランザクションは通常、全表スキャン・パスよりも索引アクセス・パスの方を選ぶため、このパラメータの値を低くすることをお勧めします。

推奨設定: 50

PROCESSES

`PROCESSES` は、Oracle データベースに同時に接続できるオペレーティング・システムのユーザー・プロセスの最大数を指定します。Oracle Hyperion Financial Management は専用サーバーとのみ連携するため、各接続にはプロセスが必要です。各 Financial Management アプリケーションでは、Financial Management のデータベ

ース接続プール設定で指定された数に加えて、最低 200 (複数サーバー環境の場合は 400) のデータベース接続を必要とします。

推奨設定: [Oracle Database リリース 11g のプロセス数を計算する方法](#)を参照してください。

SESSIONS

このパラメータは、データベース・システムで作成できるセッションの最大数を指定します。すべてのログインにはセッションが必要であるため、このパラメータは実質的に Oracle データベースの同時ユーザーの最大数を決定します。デフォルト値は、 $1.1 * PROCESSES + 5$ です。デフォルト値を下回る値にこのパラメータを設定することはお薦めしません。

TRANSACTIONS

TRANSACTIONS は、同時トランザクションの最大数を指定します。トランザクションによっては再帰的であるため、このパラメータは SESSIONS より大きい値にし、SESSIONS は PROCESSES より大きい値にして再帰的なトランザクションを考慮する必要があります。デフォルト値は、 $1.1 * SESSIONS$ です。デフォルト値を下回る値にこのパラメータを設定することはお薦めしません。

OPEN_CURSORS

OPEN_CURSORS は、1 つのセッションで一度に保持できるオープン・カーソル(プライベート SQL 領域へのハンドル)の最大数を指定します。アプリケーションでオープン・カーソルが不足しない程度に十分高い OPEN_CURSORS の値に設定することが重要です。

OPEN_CURSORS で指定したカーソル数をセッションでオープンしなければ、この値を実際に必要な数より高く設定しても、オーバーヘッドは追加されません。

推奨設定: 5000

SESSION_CACHED_CURSORS

SESSION_CACHED_CURSORS は、キャッシュするセッション・カーソル数を指定します。同じ SQL 文に対して解析コールを繰り返すと、その文のセッション・カーソルがセッション・カーソル・キャッシュに移動されます。その後の解析コールではキャッシュ内にカーソルが見つかるため、カーソルを再オープンする必要がありません。Financial Management 接続もキャッシュされるため、Oracle Hyperion Financial Management アプリケーションのパフォーマンスはこのキャッシュからの恩恵を受けます。

推奨設定: 50

TRACE_ENABLED

TRACE_ENABLED は、Oracle データベースの実行履歴またはコード・パスのトレースを制御します。パラメータを TRUE に設定してこのオプションを有効にすると、データベースにオーバーヘッドがさらに追加されるため、通常の Financial Management アプリケーション環境ではお薦めしません。

推奨設定: FALSE

STATISTICS_LEVEL

STATISTICS_LEVEL は、データベースおよびオペレーティング・システムの統計収集のレベルを指定します。Oracle データベースでは、自己管理に関する意思決定などの様々な目的のために、これらの統計が収集されます。デフォルト設定の TYPICAL では、データベースの自己管理機能に必要なすべての主要統計が収集され、全体的なパフォーマンスが最適化されます。

推奨設定: TYPICAL

TIMED_STATISTICS

TIMED_STATISTICS は、時間に関連する統計情報を収集するかどうかを指定します。Oracle Database リリース 11.1.0.7.0 以降では、STATISTICS_LEVEL の値が TYPICAL または ALL に設定されていると、TIMED_STATISTICS パラメータの値を FALSE に設定できません。

推奨設定: TRUE

TIMED_OS_STATISTICS

TIMED_OS_STATISTICS は、クライアントからサーバーへの要求が発生したとき、または要求が完了したときに、Oracle でオペレーティング・システムの統計情報を収集する間隔(秒単位)を指定します。パラメータを 0 (ゼロ)より大きい数値に設定してこのオプションを有効にすると、アプリケーションのパフォーマンスが大幅に低下します。

推奨設定: 0

Oracle Database リリース 11g のメモリー設定を決定する方法

この項では、Oracle システム関連の統計をモニターおよび表示する方法と、Oracle データベースのメモリー・パラメータをチューニングする方法を説明します。最適なメモリー設定を決定する方法はいくつもありますが、望ましい方法は、メモリー・アドバイザ、SGA アドバイザ、共有プール・アドバイザ、バッファ・キャッシュ・アドバイザ、PGA アドバイザなどのメモリー・アドバイザを使用することです。このようなアドバイザを使用して次のタスクを実行するには、DBA 権限による Oracle ログインが必要です。次の問合せの大部分には、Oracle Enterprise Manager による同等のグラフィカル・インタフェースがあります。

合計メモリーのサイズ設定(MEMORY_TARGET)

MEMORY_TARGET は、SGA と PGA の両方を含め、Oracle システム全体の使用可能なメモリーを指定します。Oracle Database リリース 11g より前では、SGA および PGA を別個にチューニングする必要がありました。

データベースを Oracle 10g から 11g にアップグレードする場合、Oracle 10g データベースの SGA_TARGET と PGA_AGGREGATE_TARGET を加算して MEMORY_TARGET を求めることができます。

データベースを Oracle 9i から 11g にアップグレードする場合、DB_CACHE_SIZE、SHARED_POOL_SIZE、LARGE_POOL_SIZE、JAVA_POOL_SIZE などのすべての SGA コンポーネントと PGA_AGGREGATE_TARGET を加算して MEMORY_TARGET を求めることができます。

データベースが以前のバージョンからアップグレードされておらず、履歴参照がない場合、システム・リソースおよびシステム制限に応じて、このパラメータを 1 から 3GB に初期設定することをお勧めします。データベースをしばらくの間使用した後、このパラメータを次のように調整できます。(このチューニングは前述の 2 つのアップグレード・シナリオにも適用されます)。

```
SQL> select * from v$memory_target_advice order by memory_size;
```

```
MEMORY_SIZE MEMORY_SIZE_FACTOR ESTD_DB_TIME ESTD_DB_TIME_FACTOR VERSION
```

```
180 .5 458 1.344 0
```

```
270 .75 367 1.0761 0
```

```
360 1 341 1 0
```

```
450 1.25 335 .9817 0
```

```
540 1.5 335 .9817 0
```

```
630 1.75 335 .9817 0
```

```
720 2 335 .9817 0
```

MEMORY_SIZE_FACTOR が 1 の行には、MEMORY_TARGET 初期化パラメータで設定した現在のメモリのサイズと、現在のワークロードを完了するのに必要な DB 時間が表示されます。前後の行の結果には、代替 MEMORY_TARGET サイズの数値が表示されます。代替サイズごとに、データベースのサイズ係数(現在のサイズの倍数)と、MEMORY_TARGET パラメータが代替サイズに変更された場合に現在のワークロードを完了するための推定 DB 時間が表示されます。現在の MEMORY_TARGET サイズ(この例では 360)より合計メモリー・サイズが小さいほど、推定 DB 時間(ESTD_DB_TIME)が増えます。また、この例では、ESTD_DB_TIME 値が減らないため、合計メモリー・サイズを 450MB を超えて増やしても何も得られません。したがって、この例では、推奨 MEMORY_TARGET サイズは 450MB です。

SGA のサイズ設定(SGA_TARGET)

SGA は通常、自動メモリー管理が有効になっている場合、Oracle データベースによって自動的にチューニングされます。しかし、DBA が SGA のサイズを引き続きモニターし、最適なサイズになっているかどうかを調べることができます。

```
SQL> select * from v$sga_target_advice order by sga_size;
```

```
SGA_SIZE SGA_SIZE_FACTOR ESTD_DB_TIME ESTD_DB_TIME_FACTOR  
ESTD_PHYSICAL_READS
```

```
290 .5 448176 1.6578 1636103
```

```
435 .75 339336 1.2552 1636103
```

```
580 1 201866 1 513881
```

```
725 1.25 201866 1 513881
```

```
870 1.5 201866 1 513881
```

```
1015 1.75 201866 1 513881
```

```
1160 2 201866 1 513881
```

「合計メモリーのサイズ設定」の項と同様の分析に基づいた場合、SGA_TARGET の現在の設定はすでに最適です。

PGA のサイズ設定(PGA_AGGREGATE_TARGET)

SGA と同様に PGA も、自動メモリー管理が有効になっている場合、Oracle データベースによって自動的にチューニングされます。次の問合せは、PGA サイズが適切に設定されているかどうかをモニターするために使用できます。結果は、v\$memory_target_size および v\$sga_target_size の問合せ結果に似ています。

```
SQL> select * from v$pga_target_advice order by  
pga_target_for_estimate;
```

LOG_BUFFER のサイズ設定

システム・ビュー v\$sysstat の REDO ログ・バッファ割当て再試行回数の値には、ユーザー・プロセスが REDO ログ・バッファ内のスペースを待機した回数が反映されま

す。この値は適切にサイズ設定されたデータベースに対してゼロに近い必要があります。例:

```
select name, value

from v$sysstat

where name = 'redo buffer allocation retries'

NAME VALUE

redo buffer allocation retries 1021967
```

ログ・バッファに更新用のスペースがない場合、データベースは待機してから再試行する必要があります。この例では、データベースは合計 **1,021,967** 回再試行しています。パフォーマンスを向上させるには、**LOG_BUFFER** パラメータ値を増やします。この値はバイト単位で表され、ログ・ブロック・サイズ値(オペレーティング・システムのブロック・サイズ)の倍数である必要があります。**Oracle Hyperion Financial Management** アプリケーションの場合、**LOG_BUFFER** を最小値 **8MB** に設定した後、前述の問合せを使用してパフォーマンスをモニターし、必要に応じて増やします。**LOG_BUFFER** を増やす必要がある場合は、一度に **50%** ずつ増やすことをお勧めします。

Oracle Database リリース 11g のプロセス数を計算する方法

Oracle データベースに同時に接続できるユーザー・プロセスの数は、Oracle データベース・インスタンスの **PROCESSES** 初期化パラメータによって制限されます。デフォルトでは、各 **Oracle Hyperion Financial Management** アプリケーション・プロセスでは、単一サーバー環境に最低 **200** のデータベース接続が必要です。環境に複数の **Financial Management** サーバーが構成されている場合は、アプリケーションの整合性を確保するために、クラスター・コントローラ接続用の追加のプールが必要です。クラスター・コントローラ・プールはアプリケーション・プールと等しいため、**DBA** はアプリケーション・プロセスごとに **400** 接続を考慮する必要があります。

ノート:

クラスター内のサーバーの合計数とアプリケーションの合計数は、必要なデータベース接続の数に影響します。

次の例では、Oracle データベースに接続するプロセスの数を計算する方法を示します。単一の **Financial Management** アプリケーション・サーバーに 2 つのアプリケーションがあるとします。デフォルトの **Financial Management** のデータベース接続プール設定は **200** です。**Financial Management** のみに必要な Oracle データベース接続の最小数は **400** です。安全性を高めるためには、この数値に係数 **1.1** を乗算して、補助的な接続およびデータベースの一般的な使用量を考慮します。Oracle データベースにはバックグラウンド・プロセスもあることを考慮して、この数値に **20** を加算して **PROCESSES** の値を求めます。したがって、この場合、**PROCESSES** を **460** に設定する必要があります。

単一の HFM アプリケーション・サーバーの場合:

$PROCESSES = (\text{Financial Management 接続プール設定}) * (\text{Financial Management アプリケーションの数}) * 1.1 + 20$

2 つの Financial Management アプリケーション・サーバーで構成されるクラスタで、各サーバーに 2 つのアプリケーションがあるとします。Financial Management のデータベース接続プール設定は 200 で、クラスタ・コントローラ・プールも 200 となります。Financial Management のみに必要な Oracle データベース接続の最小数は今度は 1,600 です: $(200+200)*2*2=1600$ 。

複数サーバー環境の場合:

$PROCESSES = (\text{Financial Management 接続プール設定} + \text{Financial Management クラスタ・コントロール接続設定}) * (\text{Financial Management アプリケーションの数}) * (\text{Financial Management サーバーの数}) * 1.1 + 20$

その他の考慮事項

共有サーバーと専用サーバー

Oracle Hyperion Financial Management では、Oracle データベースへの接続がすべて専用サーバー・プロセスによって処理される必要があります。つまり、Financial Management は共有サーバー・プロセスとは連携しません。専用サーバー・プロセスでは、CPU およびメモリー・リソースの消費が多くなりますが、高いパフォーマンスが実現します。専用サーバーを使用するには、ネット・サービス名の値の接続記述子に `SERVER=DEDICATED` 句を含める必要があります。次に、専用サーバー・プロセス用に構成されたネット・サービスの例を示します。

```
HFMDB = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST =
hfm.oracle.com) (PORT = 1521)) (CONNECT_DATA =(SERVER = DEDICATED)
(SERVICE_NAME = HFMDB1) ) )
```

オンライン REDO ログ・ファイルのサイズ

データベースのライター・プロセスおよびアーカイバ・プロセスの動作は REDO ログのサイズによって決まるため、REDO ログ・ファイルのサイズがパフォーマンスに影響する可能性があります。通常、REDO ログ・ファイルの規模が大きくなると、パフォーマンスは向上します。規模が小さいログ・ファイルでは、チェックポイント・アクティビティが増加し、パフォーマンスが低下します。ただし、小規模のログ・ファイルと頻繁なチェックポイントによってリカバリ時間は短縮されます。そのため、毎日の業務効率がリカバリ時間を最小限に抑えることよりも重要である場合は、オンライン REDO ログ・ファイルを比較的大きい値に設定します。Oracle Hyperion Financial Management データベースの場合、数百 MB が通常サイズです。ただし、REDO ログ・ファイルのサイズを決定する望ましい方法は、`FAST_START_MTTR_TARGET` を有効にして、しばらくの間一般的なデータベース・ワークロードを実行することです。その後、次の問合せを実行して REDO ログの最適なサイズを取得します。

```
SQL> select optimal_logfile_size from v$instance_recovery;
```

MTTR ターゲットとオンライン REDO ファイルのチューニング方法の詳細は、『Oracle Database パフォーマンス・チューニング・ガイド』を参照してください。

表領域とセグメントの断片化

時間の経過とともに、表領域内でのオブジェクトに対する更新および削除により、新しいデータに再利用されるほど個々には大きくない空のスペースのポケットが作成される傾向があります。このような空のスペースを断片化された空きスペースと呼びます。断片化された空きスペースがあるオブジェクトは無駄なスペースが多くなり、データベースのパフォーマンスに影響を及ぼすことがあります。Oracle Hyperion Financial Management の連結では、広い範囲にわたって更新、挿入および削除を実行するため、表領域の断片化をモニターして定期的に最適化することは非常に重要です。このスペースを最適化して再利用する望ましい方法は、オンライン・セグメント縮小を実行することです。オンライン・セグメントの使用方法の詳細は、『Oracle Database 管理者ガイド』を参照するか、Oracle データベース・サポート・サービスに問い合わせてください。

インデックスの断片化

Oracle Hyperion Financial Management アプリケーションは通常、インデックスを数百または数千すら作成します。時間の経過とともにアプリケーション・データが変化すると、インデックスが断片化される場合があります。このようなインデックスを定期的にモニタリングして最適化すると、パフォーマンスを向上させることができます。ただし、インデックスの再構築は時間がかかるリソース集約型の操作です。アプリケーションの稼働中にインデックスを再構築することはお薦めしません。Oracle Enterprise Manager には、インデックスの統計をモニターするための使いやすいインタフェースが用意されています。Enterprise Manager を使用してインデックスをモニターおよび最適化する方法の詳細は、Oracle データベースのドキュメントを参照してください。

機能 DEFERRED_SEGMENT_CREATION の無効化

リリース 11.2 には、DEFERRED_SEGMENT_CREATION 機能が導入されています。すべてのインストールで、デフォルト設定はオンです。この機能により、TABLE create 文で表が実際には作成されなくなります。表は、データの行が挿入された後でのみ作成されます。

機能 DEFERRED_SEGMENT_CREATION の無効化

リリース 11.2 には、DEFERRED_SEGMENT_CREATION 機能が導入されています。すべてのインストールで、デフォルト設定はオンです。この機能により、TABLE create 文で表が実際には作成されなくなります。表は、データの行が挿入された後でのみ作成されます。表によってはインポート中に作成されないことがあるため、この機能が原因で、Oracle Hyperion Financial Management スキーマのエクスポートおよびインポート時に問題が発生する可能性があります。この機能を無効にすることをお薦めします。その場合、表を自動的に作成する必要があります。この機能を無効にするには、SYSTEM または SYS を使用してインスタンスにログインし、次のコマンドを発行します：

```
alter system set deferred_segment_creation=false;
```

この文の発行後に作成される表は自動的に作成されます。空の表が含まれるインスタンスがすでに存在する場合にアプリケーションをエクスポートするには、各表を強制的に作成するように個別に変更し、表が EXP (エクスポート) コマンドで使用されることを許可できます。

スキーマが空の表かどうかを判別するには、次の 2 つのコマンドのいずれかを実行します:

```
select segment_name, segment_type, extents from dba_segments where
extents < 1 and segment_type='TABLE' and owner='<hfm db schema>'
```

```
select table_name from all_tables where owner='<hfm db schema>' and
table_name not in (select segment_name from dba_segments where
owner='<hfm db schema>' and segment_type='TABLE' and extents>0)
```

空の表ごとに次のコマンドを発行します:

```
alter table <table_name> allocate extent
```

定期的なメンテナンスおよびチューニング・プラン

これまでの各項では、Oracle のメモリー・パラメータのサイズを正しく設定する一般的なプロセスについて概説しました。その性質上、パフォーマンス・チューニングは反復します。パフォーマンス・ブロックを 1 つ排除すると、別のブロックが現れることもあるため、パフォーマンスがすぐに向上しない可能性があります。したがって、このプロセスはパフォーマンスが許容範囲内になるまで繰り返す必要があります。

Oracle Hyperion Financial Management アプリケーション・データは全期間を通して絶えず変化するため、定期的なデータベースのメンテナンスおよびチューニング・プランは、先を見越して Oracle データベースのパフォーマンスをモニターおよびチューニングする際に役立ち、将来発生する可能性があるパフォーマンスの問題を回避します。詳細およびその他のチューニング・オプションは、Oracle データベース・サポートに問い合せてください。

よくある質問

どのオペレーティング・システムがサポートされていますか？

Oracle Enterprise Performance Management System サポート対象プラットフォームのマトリックス: [Oracle EPM サポート対象プラットフォーム・マトリックス](#)を参照してください。

リレーショナル・データベースは 64 ビットである必要がありますか？

データベースは、サポートされている DBMS のタイプとバージョンであれば、32 ビットと 64 ビットのいずれでもかまいません。

サードパーティ・ソフトウェアおよび拡張ソフトウェアを考慮する必要があります。デフォルトでは、Oracle Hyperion Financial Management のインストールで 64 ビットのオペレーティング・システムにインストールされるのは 64 ビットのソフトウェアのみです。つまり、64 ビットのクライアント・コンポーネントのみが Financial Management アプリケーション・サーバーにインストールされます。32 ビットの接続が必要な場合、32 ビットのクライアント・ソフトウェアが 64 ビットのアプリケーション・サーバーにインストールされないかぎり、コンポーネントが機能しないことがあります。詳細は、[Oracle Enterprise Performance Management System インストールガイド](#)および[構成ガイド](#)を参照してください。

64 ビットの Financial Management にはどのようなメモリー制限がありますか？

実際の問題として、64 ビットの Financial Management は仮想メモリーではなく物理メモリーによって制限されます。メモリー・パラメータを適切に調整した後であれば、使用可能な物理メモリーをすべて利用できます。

Financial Management 用にチューニングが必要なメモリー設定はありますか？

Financial Management のデフォルトのメモリー設定は、32 ビット環境での小規模から中規模サイズのアプリケーションに適しています。使用可能なメモリーを最大限に利用するために、月次のアプリケーションには次の設定をお勧めします。関連する設定は `MaxNumDataRecordsInRAM` および `MaxDataCacheSizeinMB` で、変更する必要があります。次の表に、使用可能なメモリーに応じたこれらのパラメータの推奨値を示します。この表では、Financial Management がマシンで実行されている唯一のメモリー集約型プロセスであり、Financial Management アプリケーションを 1 つしか実行していないということを前提としています。複数のアプリケーションがアクティブの場合、サーバーに搭載されている物理メモリーの合計を Financial Management アプリケーションの数で除算して、各アプリケーションの使用可能な物理メモリーを求めます。

使用可能な物理メモリー	MaxNumDataRecordsinRAM	MaxDataCacheSizeinMB
4	4,000,000	500
8	10,000,000	1500
16	30,000,000	4500
32	60,000,000	9000

例: RAM が 24 GB のサーバーで月次の Financial Management アプリケーションが 2 つアクティブであり、`MaxNumDataRecordsInRAM` の値は 22,500,000、`MaxDataCacheSizeinMB` の値は 3375 です。

平均で、月次のアプリケーションでは 1 データ・レコードに使用されるメモリー量は 112 バイトであり、週次のアプリケーションでは 1 データ・レコード当たり 472 バイトを使用し、日次のアプリケーションでは 1 データ・レコード当たり 3,296 バイトを使用します。

週次のアプリケーションの場合、`MaxNumDataRecordsInRAM` を 4 で除算します。`MaxDataCacheSizeinMB` については、最後の列の値を変更する必要はありません。

日次のアプリケーションの場合、`MaxNumDataRecordsInRAM` を 30 で除算します。`MaxDataCacheSizeinMB` については、最後の列の値を変更する必要はありません。