

# Oracle® Hyperion Profitability and Cost Management

## Administrator's Guide

リリース 11.1.2.3

### Profitability and Cost Management Administrator's Guide, 11.1.2.3

Copyright © 2008, 2013, Oracle and/or its affiliates. All rights reserved.

著者: EPM 情報開発チーム

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

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

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

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

#### U.S. GOVERNMENT RIGHTS:

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

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

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

---

# 目次

---

ドキュメントのアクセシビリティについて .....	9
<b>第 I 部 Profitability and Cost Management の概要 .....</b>	<b>11</b>
<b>第 1 章 概要 .....</b>	<b>13</b>
Profitability and Cost Management のアクセシビリティ .....	14
Profitability タイプ .....	14
標準 Profitability .....	14
詳細 Profitability .....	15
追加の製品コンポーネント .....	15
管理タスク .....	16
インストール後の最初の Profitability and Cost Management アプリケーションの 作成 .....	17
Profitability and Cost Management の起動 .....	18
<b>第 2 章 セキュリティの管理およびユーザーの認可 .....</b>	<b>21</b>
ユーザーの設定およびプロビジョニングについて .....	21
セキュリティの役割の割当て .....	23
標準 Profitability のセキュリティの役割 .....	24
詳細 Profitability のセキュリティの役割 .....	26
Profitability and Cost Management の変更点の監査 .....	28
出力ログ・ファイル .....	31
Oracle Diagnostic Logging (ODL)ファイル .....	32
<b>第 II 部 標準 Profitability モデルの操作 .....</b>	<b>33</b>
<b>第 3 章 標準 Profitability について .....</b>	<b>35</b>
製品アーキテクチャの理解 .....	35
データベースの処理 .....	36
<b>第 4 章 Profitability and Cost Management の標準的な次元およびメンバーの操作 .....</b>	<b>39</b>
標準的な Profitability の次元とメタデータについて .....	39
次元タイプ .....	41
メジャー次元 .....	41
ドライバ・メジャー .....	42

費用レイヤー配賦メジャー	44
収益レイヤー配賦メジャー	45
レポート・メジャー	47
AllocationType 次元	48
別名次元	49
ビジネス次元	50
POV 次元	51
属性次元	51
次元ソート順および密度の設定	52
次元ソート順の推奨事項	53
次元ソート順の設定	53

<b>第 5 章 Performance Management Architect を使用した標準 Profitability アプリケーションおよび次元の管理</b>	55
アプリケーションおよび次元の操作	55
Essbase の次元設定の最適化	56
メタデータのインポート	57
Profitability and Cost Management の次元プロパティおよびメンバー・プロパティ	58
階層プロパティ・タイプの設定	63
標準 Profitability アプリケーションの作成	64
標準 Profitability の次元の選択	65
ウィザードを使用した標準 Profitability の次元の追加	67
標準 Profitability 次元の手動での追加	69
標準 Profitability アプリケーションの設定の変更	70
次元ライブラリを使用したアプリケーション設定の変更	71
アプリケーション・ウィザードを使用したアプリケーション設定の変更	71
POV 次元のプロパティの設定	72
子メンバーの順序変更	73
Performance Management Architect でのアプリケーションの検証と配置	74
Profitability and Cost Management の検証	74
ウィザードを使用したアプリケーションの検証と配置	76
Performance Management Architect を使用したアプリケーションの検証と配置	77
データの同期	79
<b>第 6 章 標準 Profitability のインポート・ステージング・テーブル</b>	81
インポート・ステージング・テーブルの使用	81
標準 Profitability のインポート・データベース・テーブルの作成	82
HPM_STG_STAGE	82

HPM_STG_POV .....	83
HPM_STG_DRIVER .....	85
HPM_STG_DRIVER_SEL .....	87
HPM_STG_DRIVER_EXCEP .....	88
HPM_STG_ASSIGNMENT .....	90
HPM_STG_ASGN_RULE_SEL .....	91
<b>第 7 章 標準 Profitability のモデル定義データのエクスポート .....</b>	<b>95</b>
HPM_EXP_STAGE .....	95
HPM_EXP_POV .....	96
HPM_EXP_DRIVER .....	97
HPM_EXP_DRIVER_SEL .....	98
HPM_EXP_DRIVER_EXCEP .....	99
HPM_EXP_ASSIGNMENT .....	100
HPM_EXP_ASGN_RULE_SEL .....	101
<b>第 III 部 詳細 Profitability の操作 .....</b>	<b>103</b>
<b>第 8 章 詳細 Profitability の製品アーキテクチャの理解 .....</b>	<b>105</b>
<b>第 9 章 詳細 Profitability データベースの操作 .....</b>	<b>107</b>
<b>第 10 章 製品スキーマ .....</b>	<b>109</b>
<b>第 11 章 モデル・データ・スキーマ .....</b>	<b>111</b>
製品スキーマへのモデル・データ・スキーマ・テーブル権限の付与 .....	111
モデル・データ・スキーマのテーブル構造要件 .....	112
データベース・ビューの登録 .....	112
<b>第 12 章 詳細 Profitability and Cost Management の次元とメタデータの操作 .....</b>	<b>115</b>
詳細 Profitability の次元とメタデータについて .....	115
詳細 Profitability の次元タイプ .....	117
別名次元 .....	117
ビジネス次元 .....	118
POV 次元 .....	119
属性次元 .....	120
次元ソート順および密度の設定 .....	120
次元ソート順の推奨事項 .....	121
次元ソート順の設定 .....	121
<b>第 13 章 Performance Management Architect を使用した詳細 Profitability アプリケーションおよび次元の管理 .....</b>	<b>123</b>
アプリケーションおよび次元の操作 .....	123

Essbase の次元設定の最適化 .....	124
メタデータのインポート .....	125
詳細 Profitability and Cost Management の次元プロパティおよびメンバー・プロパティ .....	126
詳細 Profitability アプリケーションの作成 .....	129
詳細 Profitability アプリケーションのタイプの選択 .....	130
詳細 Profitability の次元の選択 .....	130
アプリケーション設定の変更 .....	135
Performance Management Architect でのアプリケーションの検証と配置 .....	138
詳細 Profitability and Cost Management の検証 .....	139
ウィザードを使用したアプリケーションの検証と配置 .....	140
Performance Management Architect を使用したアプリケーションの検証と配置 .....	141
<b>第 14 章 詳細 Profitability の計算 .....</b>	<b>143</b>
計算前および計算後のカスタム・スクリプト .....	143
HPM_SQL_SCRIPT .....	144
カスタム・スクリプトの作成 .....	144
データ転送を行うための ODBC データ・ソースの作成 .....	146
高度な計算オプション .....	146
ドライバ操作タイプ .....	146
その他のプロセス・タイプ .....	147
<b>第 15 章 詳細 Profitability のインポート・ステージング・テーブル .....</b>	<b>149</b>
詳細 Profitability のインポート・データベース・テーブルの作成 .....	150
HPM_STGD_POV .....	150
HPM_STGD_DRIVER .....	151
HPM_STGD_DRIVER_SEL .....	153
HPM_STGD_DRIVER_EXCEP .....	154
HPM_STGD_ASGN_RULE_SEL .....	155
HPM_STGD_CALCRULE_SNGLSRC .....	156
HPM_STGD_CALCRULE_CALCMSRS .....	157
HPM_STGD_CALCRULE_MULTISRC .....	159
<b>第 16 章 詳細 Profitability のモデル定義データのエクスポート .....</b>	<b>161</b>
HPM_EXPD_STAGE .....	161
HPM_EXPD_POV .....	163
HPM_EXPD_DRIVER .....	163
HPM_EXPD_DRIVER_SEL .....	164
HPM_EXPD_DRIVER_EXCEP .....	165
HPM_EXPD_ASGN_RUL_SEL .....	166

HPM_EXPD_CALCRULE_SNGLSRC .....	167
HPM_EXPD_CALCRULE_CALCMSRS .....	168
HPM_EXPD_CALCRULE_MULTISRC .....	169
アプリケーションの統計の生成 .....	170
<b>付録 A. Profitability and Cost Management へのデータのインポート .....</b>	<b>173</b>
ステージング・テーブルについて .....	174
インポート構成の作成 .....	175
インポート構成の変更 .....	177
インポート構成の削除 .....	178
インポート構成の実行 .....	178
インポートしたデータの確認 .....	179
<b>付録 B. EPM System ライフサイクル管理を使用したデータの移行 .....</b>	<b>181</b>
ライフサイクル管理について .....	181
ライフサイクル管理に対するデフォルトのタイムアウト設定の変更 .....	182
<b>付録 C. Profitability and Cost Management コンポーネントのバックアップ .....</b>	<b>185</b>
<b>付録 D. Essbase 命名規則 .....</b>	<b>187</b>
生成された計算スクリプトの命名規則 .....	187
アプリケーションとデータベースの Essbase 命名規則 .....	188
次元、メンバーおよび別名の Essbase 命名規則 .....	189
属性計算次元の Essbase 命名規則 .....	192
<b>索引 .....</b>	<b>195</b>





---

# ドキュメントのアクセシビリティについて

---

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

## Access to Oracle Support

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



---

## 第 I 部

# Profitability and Cost Managementの概要

---

Profitability and Cost Management の概要の内容 :

- 概要
- セキュリティの管理およびユーザーの認可



# 1

## 概要

### この章の内容

Profitability and Cost Management のアクセシビリティ .....	14
Profitability タイプ .....	14
追加の製品コンポーネント.....	15
管理タスク.....	16
インストール後の最初の Profitability and Cost Management アプリケーションの作成 .....	17
Profitability and Cost Management の起動 .....	18

Oracle Hyperion Profitability and Cost Management は、Oracle Hyperion Enterprise Performance Management Workspace からアクセスする分析アプリケーションで、費用分解、消費ベースの費用計算およびシナリオ再生を使用することにより、コストと収益を正確に測定、配賦および管理し、事業区分の収益性を計算し、収益性を測定するために使用します。

Profitability and Cost Management は、EPM Workspace の一部分であり、様々なコンポーネントを使用してアプリケーションを構築および管理します:

- EPM Workspace を使用して、Profitability and Cost Management にアクセスし、アプリケーションを構築するための他のコンポーネントを管理し、セキュリティおよび製品レポートを制御します。Profitability アプリケーションのタイプについては、14 ページの「Profitability タイプ」を参照してください。
- Oracle Hyperion Shared Services を使用して、ユーザー・アカウントを作成および管理します。これには、ユーザーがアクセスできるモデルを決定するセキュリティの役割の定義が含まれます。
- Oracle Hyperion EPM Architect を使用して、Profitability and Cost Management で使用する共有次元とローカル次元の両方を構築および管理します。次元ライブラリを介して、Oracle Hyperion Planning などの他のアプリケーションにすでに存在する共有次元および共有メンバーを使用してアプリケーションを構築できます。

Performance Management Architect を使用することにより、次元はアプリケーション内に共有またはローカルとして存在できます:

- 共有次元は、Performance Management Architect の共有ライブラリ内に存在し、複数のアプリケーションが使用できます。
- ローカル次元は、Profitability and Cost Management などの 1 つのアプリケーション内にもみ存在する、分離された独立次元です。これらの次元は、これらが作成されているアプリケーションに対してのみ使用されます。

このガイドでは、モデルのアプリケーションと次元を管理するために Performance Management Architect を使用すると仮定します。ただし、Performance Management Architect を使用するかわりに、Profitability アプリケーションを使用することもできます。

Performance Management Architect のかわりに Profitability アプリケーションを使用する場合、この管理ガイドで Performance Management Architect に関して説明するすべての手順について、ユーザー・ガイドの Profitability アプリケーションに関する項を参照する必要があります。

- Oracle Essbase またはリレーショナル・データベースを使用して、アプリケーション・アウトラインを作成し、計算スクリプトを格納および実行します。
- Oracle Hyperion Enterprise Performance Management System ライフサイクル管理を使用して、製品環境およびオペレーティング・システム間でアプリケーション、多次元データベース、リポジトリまたは個別アーチファクトを移行します。
- Oracle Hyperion Reporting and Analysis、Oracle Hyperion Financial Reporting、Oracle Hyperion Web Analysis、または Microsoft Excel などのサードパーティの製品を使用して、計算結果のレポートを作成します。

## Profitability and Cost Management のアクセシビリティ

Profitability and Cost Management には、主要な機能を対象としたキーボード・ショートカットが用意されています。

アクセシビリティ機能の詳細は、Oracle Hyperion Profitability and Cost Management アクセシビリティ・ガイドを参照してください。これらの機能は、Profitability and Cost Management に固有のものであります。

### Profitability タイプ

Profitability and Cost Management には、収益性の評価で使用される 2 つの異なるタイプのアプリケーションがあります。

- [14 ページの「標準 Profitability」](#)
- [15 ページの「詳細 Profitability」](#)

### 標準 Profitability

標準 Profitability では、プロセスのあらゆる段階における原価および収益の資金フローに従って、資金の拠出元と用途を決定する標準拠出金分析に重点が置かれます。

標準 Profitability and Cost Management モデルを使用すると、モデル全体の直接のコントリビューション・データを監視および制御できます。原価および収益の両方について入力金額、原価と収益のフロー、資金の最終的な宛先を追跡すること

によって、リソースを最大限に活用し収益性を簡単に示すことができるようになります。計算結果は個々のコスト・センターまたは勘定科目に転記されます。

標準 Profitability モデルのデータは、Essbase の多次元データベースとリレーショナル・データベースの両方に格納されます。Performance Management Architect でモデルを作成し、次元および次元メンバーを使用して組織内の勘定科目、活動および操作の階層を定義します。AllocationType 次元は、Performance Management Architect からインポートされます。この次元を使用して、費用および収益を正しく配賦したり、直接配賦および配賦システムを保管します。

標準 Profitability の操作の詳細は、第 3 章「標準 Profitability について」を参照してください。

## 詳細 Profitability

詳細 Profitability では、収益性分析のために、プールまたはレートをシングル・ステップで単一のソースから宛先に配賦します。詳細 Profitability は、アーティファクト・ストレージのモデル、計算、レポート・ビューにリレーショナル・データベースを利用します。

詳細 Profitability and Cost Management モデルは、ユーザー定義のスキーマを利用してリレーショナル・テーブルを既存のデータおよび関連する参照テーブルに編成し、そのデータを拡張します。詳細 Profitability モデルのデータは、リレーショナル・データベースのみに格納されます。

Performance Management Architect でモデルを作成し、ビジネス次元、別名、メジャーなどを組織で定義します。Profitability and Cost Management では、詳細 Profitability モデルを構築できるように、データがアプリケーションにマッピングされます。アプリケーションはきわめて大きなボリュームでも処理できます。

アプリケーションは階層構造を使用せず、ソースと宛先の単一の組み合わせの間におけるフローですべての配賦を処理します。配賦は、標準 Profitability の場合のように AllocationType を作成するのではなく、制限された MeasuresDetailed 次元を通じて処理されます。MeasuresDetailed 次元には、すべての配賦を処理する限られたメンバーが含まれます。

詳細 Profitability の操作の詳細は、第 8 章「詳細 Profitability の製品アーキテクチャの理解」を参照してください。

## 追加の製品コンポーネント

Profitability and Cost Management を使用して、モデルを作成し、配賦を実行します。Profitability and Cost Management の機能と柔軟性を拡張するには、表 1 に示されている製品を使用します。

表 1 Profitability and Cost Management の製品コンポーネント

製品	説明
Essbase	標準 Profitability の場合のみ、Profitability and Cost Management アプリケーションのデータを格納および計算します。

製品	説明
Oracle Essbase Administration Services	標準 Profitability の場合のみ、Essbase Administration Services Console を使用して Essbase サーバーに接続します。 複数の Essbase アプリケーションおよびデータベースを設計、開発、維持および管理するために使用します。
Performance Management Architect	次元およびアプリケーションを管理します
EPM Workspace	Profitability and Cost Management および他の製品に移動し、アプリケーションを管理します
Shared Services	Profitability and Cost Management の外部システムからユーザーをプロビジョニングします Profitability and Cost Management アプリケーションを移行します
Oracle Hyperion Smart View for Office	Profitability and Cost Management のデータを Microsoft Excel スプレッドシートに入力してレポートを作成します
Financial Reporting	Web または印刷配布用のレポートおよびチャートを作成します
Web Analysis	Web クライアントを使用して分析用のアド・ホックのレポートおよびチャートを作成します

## 管理タスク

Profitability and Cost Management 管理者または admin 役割を使用すると、次のタスクを実行できます:

- Shared Services を使用して、ユーザー・アカウントを作成および管理します。
- Shared Services の認証のためユーザーをプロビジョニングします。
- Performance Management Architect における共有ライブラリおよび Profitability and Cost Management の次元とメンバーを管理します。
- 多次元の Essbase データベースの生成。
- モデル・ステージ、ドライバおよび視点(POV)を作成、更新および削除します。
- ドライバの選択、割当て、割当てルール、および割当てルールの選択を作成、更新および削除します。
- 計算ルールを作成、更新および削除します。
- モデルを計算および再計算します。
- モデル・データを表示および変更します。
- トレース配賦を表示します。
- Profitability and Cost Management のモデル・コンポーネントのバックアップおよび復元を行います。
- ライフサイクル管理ユーティリティを使用して、環境間でアプリケーションを転送します。ある環境(開発またはテスト環境など)から別の環境(本番環境など)へデータのレベルを上げます。
- ビジネス・オブジェクトに加えられた変更を監視します。



Profitability and Cost Management 管理者は、EPM Workspace を介してモデル・メタデータおよびデータも管理します。

次の項を参照してください:

- [第 4 章「Profitability and Cost Management の標準的な次元およびメンバーの操作」](#)
- [第 12 章「詳細 Profitability and Cost Management の次元とメタデータの操作」](#)

## インストール後の最初の Profitability and Cost Management アプリケーションの作成

Profitability and Cost Management は、EPM Workspace の一部分であり、共通アプリケーションを使用してアプリケーションおよびセキュリティを管理します。インストール後に、複数の手順を実行して最初の Profitability and Cost Management アプリケーションを作成する必要があります。アプリケーションが作成された後、データを Profitability and Cost Management にインポートする必要があります。

**注：** マルチバイト文字は Performance Management Architect では使用できますが、Profitability and Cost Management ではサポートされません。

非 Unicode の標準 Profitability アプリケーションでは、計算用またはレポート用データベースの名前にマルチバイト文字を使用しないでください。データベース名には ASCII 文字のみを使用します。

接頭辞と次元名の長さの合計が 80 文字を超えると、Essbase への配置が失敗します。

▶ インストール後に最初の Profitability and Cost Management アプリケーションを作成する手順は、次のとおりです:

**1** 次の製品がインストール、構成および実行されていることを確認します:

- EPM Workspace
- Shared Services
- Performance Management Architect
- Essbase (標準 Profitability の場合のみ)
- Profitability and Cost Management

**注：** このリストは、Profitability and Cost Management を使用するための最小限のインストール要件を示します;ただし、追加の製品はいつでもインストールできます。

インストールの詳細は、『Oracle Hyperion Enterprise Performance Management System インストールおよび構成ガイド』および『Oracle Hyperion Enterprise Performance Management System インストール概要』を参照してください。

- 2 EPM Workspace のメイン・メニューから、「ナビゲート」、「管理」、「アプリケーション・ライブラリ」の順に選択します。
- 3 「アプリケーション・ライブラリ」から、「ファイル」、「新規」、「アプリケーション・ウィザード」の順に選択し、新しい Profitability and Cost Management アプリケーションを作成します。

新しいアプリケーションの作成の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』で該当する項を参照してください。

- 標準 Profitability アプリケーションの場合、64 ページの「標準 Profitability アプリケーションの作成」を選択します。
- 詳細 Profitability アプリケーションの場合、129 ページの「詳細 Profitability アプリケーションの作成」を選択します。

- 4 EPM Workspace のメイン・メニューで、「ナビゲート」、「管理」、「次元ライブラリ」の順に選択し、Performance Management Architect にアクセスして、最初のアプリケーションに必要な次元を作成します。

次元の作成の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

- 5 「アプリケーション・ライブラリ」から、新しいアプリケーション名を右クリックし、「検証」を選択します。

検証タスクが送信され、メタデータが確認されます。ジョブ・ステータス・メッセージにあるリンクをクリックして検証タスクのジョブ・コンソールを表示し、「添付ファイル」の下にあるログ・ファイル内の警告またはエラーを表示します。アプリケーションを配置する前に、エラーを修正する必要があります。

- 6 「アプリケーション・ライブラリ」から、新しいアプリケーション名を右クリックし、「配置」を選択します。

アプリケーション名および選択されている次元が Profitability and Cost Management サーバーに転送されます。『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

- 7 EPM Workspace のメイン・メニューで、「ナビゲート」、「アプリケーション」、「Profitability」の順に選択し、新しいアプリケーションを選択します。

**注：** 新しく作成したアプリケーションが見つからない場合、「リフレッシュ」を選択して、手順 7 を繰り返します。

- 8 「タスク領域」で、「モデルの管理」、「ステージング・テーブルのインポート」の順に選択し、モデルの構築を開始するために必要なデータをインポートします。付録 A 「Profitability and Cost Management へのデータのインポート」を参照してください。

## Profitability and Cost Management の起動

Profitability and Cost Management は、EPM Workspace を介してのみアクセスできません。

▶ Profitability and Cost Management にアクセスするには:

1 次のアプリケーションが構成および実行されていることを確認します:

- EPM Workspace
- Shared Services
- Performance Management Architect
- Essbase (標準 Profitability の場合のみ)
- Profitability and Cost Management

詳細は、Hyperion EPM System インストールおよび構成ガイドおよび『Oracle Hyperion Enterprise Performance Management System インストール概要』を参照してください。

2 Web ブラウザで、EPM Workspace の Web ページにアクセスします。

デフォルトでは、この URL は `http://server name:19000/workspace/` です。ここで、server name は Oracle HTTP Server (OHS) サーバー名です。

3 EPM Workspace のユーザー名とパスワードを入力します。

**注:** ユーザー名とパスワードは両方とも大/小文字が区別されます。

4 「ログオン」をクリックします。

EPM Workspace のメイン・ページが表示されます。

5 「ナビゲート」、「アプリケーション」、「Profitability」の順に選択し、表示するアプリケーションを選択します。



# 2

## セキュリティの管理および ユーザーの認可

### この章の内容

ユーザーの設定およびプロビジョニングについて .....	21
セキュリティの役割の割当て .....	23
Profitability and Cost Management の変更点の監査.....	28
出力ログ・ファイル .....	31
Oracle Diagnostic Logging (ODL)ファイル .....	32

## ユーザーの設定およびプロビジョニングについて

Profitability and Cost Management を操作する前に、管理者は、ユーザーとグループを設定し、それぞれに適したセキュリティの役割を割り当てる必要があります。セキュリティの役割ごとに指定される権限により、ユーザーまたはグループがアクセスできる機能およびデータが決まります。構成時に、『Oracle Hyperion Enterprise Performance Management System インストールおよび構成ガイド』の説明に従って、認証ノードとして Shared Services を選択してください。

セキュリティとユーザー・プロビジョニングの管理については、2つのガイドがあります:

- セキュリティに関する技術情報は、『Oracle Hyperion Enterprise Performance Management System セキュリティ管理ガイド』を使用してください:
  - SSL (一方向、双方向、SSL オフロード、SSL の終了)
  - シングル・サインオン
  - デフォルトの EPM System の SSO
  - セキュリティ・エージェント
  - カスタム・ログイン
  - カスタム認証モジュール
  - EPM System 保護のガイドライン
- ユーザー・プロビジョニングの設定および管理方法の詳細は、『Oracle Hyperion Enterprise Performance Management System ユーザーおよび役割セキュリティ・ガイド』を使用してください:
  - Oracle Hyperion Shared Services Console
  - ユーザー・ディレクトリ

- アプリケーションおよびアプリケーション・グループ
- 委任されたユーザー管理
- ネイティブ・ディレクトリの管理
- プロビジョニングの管理
- EPM System のプロビジョニング

EPM Workspace を介して Profitability and Cost Management のユーザーおよびグループを作成、管理およびプロビジョニングします。プロビジョニング・プロセスでは、Shared Services と Profitability and Cost Management の両方が構成されて実行されている必要があります。外部認証を使用すると、アプリケーション間の通信がシームレスになり、ユーザーを簡単かつ正確にプロビジョニングできるようになります。

次の手順は、ユーザーとグループを設定およびプロビジョニングするためのプロセスの概要を示します:

1. EPM Workspace で、「ナビゲート」、「管理」、「Shared Services Console」の順に選択して Shared Services の画面にアクセスします。

**注：** 初めてログオンする場合、製品の管理者(admin)ユーザーが自動的に作成されます。

2. admin 役割が割り当てられているユーザーをプロビジョニング・マネージャの役割に設定します。『Oracle Hyperion Enterprise Performance Management System ユーザーおよび役割セキュリティ・ガイド』を参照してください。
3. ユーザーを作成します。『Oracle Hyperion Enterprise Performance Management System ユーザーおよび役割セキュリティ・ガイド』を参照してください。
4. 適切なセキュリティの役割を使用してユーザーをプロビジョニングし、必要なプロジェクトおよびアプリケーションにアクセスします。

---

**注意** ユーザーが EPM Workspace にアクセスして Profitability and Cost Management の外部でタスクを実行する必要がある場合、ライフサイクル管理など、EPM Workspace の役割を使用して選択したユーザーをプロビジョニングすることも必要です。

---

5. 必要に応じて、グループを作成します。グループ内のグループをプロビジョニングできます。『Oracle Hyperion Enterprise Performance Management System ユーザーおよび役割セキュリティ・ガイド』を参照してください。
6. EPM Workspace にログオンし、Profitability and Cost Management を開きます。
7. ユーザーが Profitability and Cost Management にログオンでき、プロビジョニングされたプロジェクトおよびアプリケーションを表示できることを確認します。

ユーザーとグループの設定およびプロビジョニングの詳細は、『Oracle Hyperion Enterprise Performance Management System ユーザーおよび役割セキュリティ・ガイド』を参照してください。

# セキュリティの役割の割当て

Profitability and Cost Management では、各ユーザー ID はセキュリティの役割に割り当てられています。

- 管理者(admin は Shared Services へのログオン時のデフォルトのセキュリティの役割)
- パワー・ユーザー
- 対話型ユーザー
- 表示ユーザー

割り当てられたセキュリティの役割は、ユーザーが使用可能なアクセス・レベルと権限を決定します。ユーザーには複数の役割を付与できます。特定のアクションに対する権限は、アクションが開始される時点でチェックされます。

**注：** Shared Services Console では、少なくとも 1 人のユーザーにプロビジョニング・マネージャの役割を手動で割り当てる必要があります。プロビジョニング・マネージャの役割を使用すると、ユーザーは、そのアプリケーションの他のユーザーにセキュリティの役割を割り当てることができます。『Oracle Hyperion Enterprise Performance Management System ユーザーおよび役割セキュリティ・ガイド』参照してください

ユーザーをグループに割り当てるには、このユーザーが存在し、セキュリティの役割が割り当てられている必要があります。ユーザーのグループにアクセス・レベルが割り当てられると、そのグループのすべてのメンバーに同等のセキュリティ・アクセス権が付与されます。特定ユーザーに必要なアクセス権によっては、割り当てられたセキュリティのレベル設定を変更して、アクセス権の範囲を変更できます。たとえば、パワー・ユーザーのセキュリティ権限を持つグループに割り当てられた表示ユーザーは、より高いセキュリティ・レベルになります。

表 2 で説明されているように、ユーザーがタスクフローの開始や監視を行う場合は、追加の Shared Services の役割が必要です。

---

**注意** ユーザーが EPM Workspace のアプリケーション作成者、Lifecycle Management によるステージング・テーブルのインポートまたはエクスポート、または Performance Management Architect の次元エディタなどの他の製品にアクセスする必要がある場合、これらのセキュリティの役割は個別に割り当てる必要があります。『Oracle Hyperion Enterprise Performance Management System ユーザーおよび役割セキュリティ・ガイド』を参照してください。

---

以下の項で詳細に説明しているセキュリティ役割の詳細は、Profitability and Cost Management に固有です。すべてのセキュリティの役割の詳細については、『Oracle Hyperion Enterprise Performance Management System ユーザーおよび役割セキュリティ・ガイド』を参照してください。

## 標準 Profitability のセキュリティの役割

表 2 標準 Profitability and Cost Management のセキュリティの役割

セキュリティの役割	役割のタイプ	説明
管理者 (admin)	パワー	<ul style="list-style-type: none"> <li>● ユーザー・アカウントとセキュリティ役割の作成および維持のほか、Shared Services を使用したユーザーのプロビジョニング</li> <li>● Essbase データベースの生成</li> <li>● アプリケーション・プリファレンスの設定と維持</li> <li>● 共通次元とメンバーの選択を目的とした、Performance Management Architect を使用したモデル・データベースの構築</li> <li>● ステージ、ドライバ、POV、ドライバ選択、割当て、アプリケーション・プリファレンスなどのモデルにおける要素の作成および維持</li> <li>● POV コピー、計算、検証、データ入力、トレース配賦の実行</li> <li>● Essbase への配置と計算スクリプトの生成</li> </ul> <p><b>注意</b> Profitability and Cost Management の管理者には、Essbase ASO および BSO の配置を実行するために、Essbase のアクセス権限も割り当てられている必要があります。</p> <ul style="list-style-type: none"> <li>● データのインポートおよびエクスポート</li> <li>● ライフサイクル管理ユーティリティを使用した、開発またはテスト環境から本番環境などの別の環境へのデータのレベル上げ。</li> <li>● Profitability and Cost Management のモデル・コンポーネントのバックアップおよび復元。</li> <li>● ビジネス・オブジェクトに加えられた変更を監視します。</li> <li>● クエリーを作成、編集、コピー、削除および起動します。</li> <li>● Performance Management Architect のかわりにアプリケーション・マネージャを使用して、新しい Profitability and Cost Management アプリケーションを作成および管理します。</li> </ul>
パワー・ユーザー	パワー	<ul style="list-style-type: none"> <li>● ステージ、ドライバ、POV、ドライバ選択、割当て、アプリケーション・プリファレンスなどのモデルにおける要素の作成および維持。</li> <li>● POV コピー、計算、検証、データ入力、トレース割当ての実行。</li> <li>● データのインポートおよびエクスポート</li> <li>● Essbase への配置と計算スクリプトの生成。</li> </ul> <p><b>注意</b> Profitability and Cost Management のパワー・ユーザーには、データを Essbase に入力したり、Essbase ASO および BSO の配置を実行したりするために Essbase のアクセス権限(アプリケーションの作成やアプリケーション・マネージャなど)も割り当てられている必要があります。</p> <ul style="list-style-type: none"> <li>● クエリーを作成、編集、コピー、削除および起動します。</li> </ul> <p><b>注：</b> パワー・ユーザーは、タスクの実行に特定のセキュリティ役割を必ずしも必要としません。たとえば、パワー・ユーザーが「計算」画面から計算を実行する場合、このアクションによってタスクフローが内部で作成されて実行されます。パワー・ユーザーは、「タスクフローの管理」タスクから直接このタスクにアクセスする場合以外は、このタスクの実行に「タスクフローの管理」役割を必要としません。</p>



セキュリティの役割	役割のタイプ	説明
対話型ユーザー	対話型	<ul style="list-style-type: none"> <li>● すべてのモデリング画面の表示</li> <li>● データ入力画面でのデータの表示および変更</li> <li>● トレース配賦の表示</li> <li>● クエリーを起動します。</li> </ul>
表示ユーザー	対話型	<p>次の関数に対して表示のみのアクセス権を持ちます:</p> <ul style="list-style-type: none"> <li>● 配賦のトレース</li> <li>● アプリケーション・プリファレンス</li> <li>● モデル・ステージ、ドライバ、および POV</li> </ul>
タスクフローの管理	Shared Services の役割	<p>タスクフローの作成と編集に必要。</p> <p>詳細は、『Oracle Hyperion Enterprise Performance Management System ユーザーおよび役割セキュリティ・ガイド』を参照してください。</p>
タスクフローの実行	Shared Services の役割	<p>ユーザーによるタスクフローの実行と表示のみを可能にするために必要。この役割を割り当てられたユーザーは、タスクフローの作成や編集を行うことはできません。</p> <p>詳細は、『Oracle Hyperion Enterprise Performance Management System ユーザーおよび役割セキュリティ・ガイド』を参照してください。</p>

## 詳細 Profitability のセキュリティの役割

表 3 詳細 Profitability and Cost Management のセキュリティの役割

セキュリティの役割	役割のタイプ	説明
管理者 (admin)	管理者	<ul style="list-style-type: none"> <li>● アプリケーション・プリファレンスの設定と維持</li> <li>● 共通次元とメンバーの選択を目的とした、Performance Management Architect を使用したモデル・データベースの構築</li> <li>● レポート・ビューを作成してリレーショナル・データベースに配置</li> <li>● 次の機能を作成、読取り(表示)、更新、削除:             <ul style="list-style-type: none"> <li>○ ステージ</li> <li>○ ドライバ</li> <li>○ POV</li> <li>○ ドライバの関連付け</li> <li>○ 割当て</li> <li>○ アプリケーション・プリファレンス</li> <li>○ 計算ルール</li> <li>○ ジョブ・ライブラリおよびステータス</li> <li>○ テーブル登録</li> </ul> </li> <li>● 次のタスクを実行:             <ul style="list-style-type: none"> <li>○ POV コピー</li> <li>○ 検証</li> <li>○ 配置</li> <li>○ 計算</li> <li>○ ジョブの停止</li> </ul> </li> <li>● ライフサイクル管理ユーティリティを使用した、開発またはテスト環境から本番環境などの別の環境へのデータのレベル上げ。</li> <li>● データのインポートおよびエクスポート</li> <li>● Profitability and Cost Management のモデル・コンポーネントのバックアップおよび復元。</li> <li>● ビジネス・オブジェクトに加えられた変更を監視します。</li> <li>● Essbase レポート用データベースを配置、更新および置換し、データを転送します。</li> </ul> <p><b>注意</b> Profitability and Cost Management の管理者には、Essbase ASO の配置を実行するために、Essbase のアクセス権限も割り当てられている必要があります。</p> <ul style="list-style-type: none"> <li>● Performance Management Architect のかわりにアプリケーション・マネージャを使用して、新しい Profitability and Cost Management アプリケーションを作成および管理します。</li> </ul>

セキュリティの役割	役割のタイプ	説明
パワー・ユーザー	パワー	<ul style="list-style-type: none"> <li>● ユーザー・アカウントとセキュリティ役割の作成および維持のほか、Shared Services を使用したユーザーのプロビジョニング</li> <li>● レポート・ビューを作成してリレーショナル・データベースに配置</li> <li>● 次の機能を作成、読取り(表示)、更新、削除: <ul style="list-style-type: none"> <li>○ ステージ</li> <li>○ ドライバ</li> <li>○ POV</li> <li>○ ドライバの関連付け</li> <li>○ 割当て</li> <li>○ アプリケーション・プリファレンス</li> <li>○ 計算ルール</li> <li>○ ジョブ・ライブラリおよびステータス</li> <li>○ テーブル登録</li> </ul> </li> <li>● 次のタスクを実行: <ul style="list-style-type: none"> <li>○ POV コピー</li> <li>○ 検証</li> <li>○ 計算</li> <li>○ ジョブの停止</li> <li>○ Essbase レポート用データベースを配置、更新および置換し、データを転送します。</li> <li>○ 配置</li> </ul> </li> </ul> <p><b>注意</b> Profitability and Cost Management のパワー・ユーザーには、データを Essbase に入力したり、Essbase ASO の配置を実行したりするために Essbase のアクセス権限(アプリケーションの作成やアプリケーション・マネージャなど)も割り当てられている必要があります。</p> <p><b>注:</b> パワー・ユーザーは、タスクの実行に特定のセキュリティ役割を必ずしも必要としません。たとえば、パワー・ユーザーが「計算」画面から計算を実行する場合、このアクションによってタスクフローが内部で作成されて実行されます。パワー・ユーザーは、「タスクフローの管理」タスクから直接このタスクにアクセスする場合以外は、このタスクの実行に「タスクフローの管理」役割を必要としません。</p>
対話型ユーザー	対話型	<ul style="list-style-type: none"> <li>● 次の機能を表示(読取り): <ul style="list-style-type: none"> <li>○ ステージ</li> <li>○ ドライバ</li> <li>○ POV</li> <li>○ ドライバの関連付け</li> <li>○ 割当て</li> <li>○ アプリケーション・プリファレンス</li> <li>○ 計算ルール</li> <li>○ ジョブ・ライブラリおよびステータス</li> <li>○ テーブル登録</li> </ul> </li> </ul>

セキュリティの役割	役割のタイプ	説明
表示ユーザー	対話型	<ul style="list-style-type: none"> <li>● 次の機能を表示(読取り): <ul style="list-style-type: none"> <li>○ ステージ</li> <li>○ ドライバ</li> <li>○ POV</li> <li>○ ドライバの関連付け</li> <li>○ 割当て</li> <li>○ アプリケーション・プリファレンス</li> <li>○ ジョブ・ライブラリおよびステータス</li> <li>○ テーブル登録</li> </ul> </li> </ul>
タスクフローの管理	Shared Services の役割	<p>タスクフローの作成と編集に必要。</p> <p>詳細は、『Oracle Hyperion Enterprise Performance Management System ユーザーおよび役割セキュリティ・ガイド』を参照してください。</p>
タスクフローの実行	Shared Services の役割	<p>ユーザーによるタスクフローの実行と表示のみを可能にするために必要。この役割を割り当てられたユーザーは、タスクフローの作成や編集を行うことはできません。</p> <p>詳細は、『Oracle Hyperion Enterprise Performance Management System ユーザーおよび役割セキュリティ・ガイド』を参照してください。</p>

## Profitability and Cost Management の変更点の監査

Shared Services Console の監査確認を使用してアクティビティおよび変更点を監視してから、結果が詳細に記載された監査レポートを生成できます。

使用可能な監査レポートには、次の3つのタイプがあります:

- セキュリティ・レポート
- アーティファクト・レポート
- 構成レポート

監査レポートには、選択した監査領域に関するアクティビティの詳細が記載されています。これには、次の情報が含まれます:

- 日付
- アプリケーション
- ユーザー
- アーティファクトのタイプと名前
- 実行されたタスク

レポートを生成するには、次の手順に説明されているように、先に監査を使用可能にする必要があります。これらのレポートは、CSV ファイルとしてエクスポートできます。『Oracle Hyperion Enterprise Performance Management System セキュリティ管理ガイド』を参照してください。

▶ 監査機能を使用可能にするには:

- 1 EPM Workspace から、「ナビゲート」、「管理」、「Shared Services Console」の順に選択します。
- 2 Shared Services Console から、「管理」>、「監査の構成」を順に選択します。

監査構成画面が表示されます。

Select Tasks	
<input checked="" type="checkbox"/>	Shared Services
<input checked="" type="checkbox"/>	BPMA-11.1.2
<input checked="" type="checkbox"/>	Analytic Server -11.1.2
<input checked="" type="checkbox"/>	Analytic Services Application -11.1.2
<input checked="" type="checkbox"/>	Profitability-11.1.2
<input checked="" type="checkbox"/>	Stage
<input checked="" type="checkbox"/>	Driver
<input checked="" type="checkbox"/>	POV
<input checked="" type="checkbox"/>	Driver selection
<input checked="" type="checkbox"/>	Assignment
<input checked="" type="checkbox"/>	Cubes
<input checked="" type="checkbox"/>	Scripts
<input checked="" type="checkbox"/>	Group operations
<input checked="" type="checkbox"/>	Group import operations
<input checked="" type="checkbox"/>	Deployment Metadata

- 3 「監査の使用可能」を選択します。

「グローバル設定の上書きを許可」および「タスクの選択」リストがアクティブ化されます。

- 4 「タスクの選択」で、監査を使用可能にするアプリケーションの領域を選択します。領域全体を選択することも、各オプションを展開して監視する手順を個別に選択することもできます。

表 4 Profitability and Cost Management の監査タスク

監査領域	使用可能な監査タスク
ステージ	<ul style="list-style-type: none"><li>● ステージの作成</li><li>● ステージの変更</li><li>● ステージの削除</li><li>● ステージのエクスポート</li><li>● ステージのインポート</li></ul>
ドライバ	<ul style="list-style-type: none"><li>● ドライバの作成</li><li>● ドライバの変更</li><li>● ドライバの削除</li><li>● ドライバのエクスポート</li><li>● ドライバのインポート</li></ul>

監査領域	使用可能な監査タスク
POV	<ul style="list-style-type: none"> <li>● POV の作成</li> <li>● POV の変更</li> <li>● POV の削除</li> <li>● POV のエクスポート</li> <li>● POV のインポート</li> </ul>
ドライバの選択	<ul style="list-style-type: none"> <li>● ドライバ選択の作成</li> <li>● ドライバ選択の変更</li> <li>● ドライバ選択の削除</li> </ul>
割当て	<ul style="list-style-type: none"> <li>● 割当ての作成</li> <li>● 割当ての変更</li> <li>● 割当ての削除</li> </ul>
割当てルール	<ul style="list-style-type: none"> <li>● 割当てルールの作成</li> <li>● 割当てルールの変更</li> <li>● 割当てルールの削除</li> <li>● ルール選択の作成</li> <li>● ルール選択の削除</li> </ul>
キューブ(標準 Profitability のみ)	<ul style="list-style-type: none"> <li>● 計算キューブの配置</li> <li>● レポート・キューブの配置</li> <li>● データの転送</li> </ul>
システム(標準 Profitability のみ)	<ul style="list-style-type: none"> <li>● システムの計算</li> </ul>
スクリプト	<ul style="list-style-type: none"> <li>● 配賦スクリプトの配置</li> <li>● システムスクリプトの配置</li> <li>● POV コピー・スクリプトの配置</li> <li>● 配賦スクリプトの実行</li> <li>● システムスクリプトの実行</li> <li>● POV コピー・スクリプトの実行</li> </ul>
操作のグループ化	<ul style="list-style-type: none"> <li>● 割当てのコピー</li> <li>● 割当ての削除</li> <li>● 割当てルールの選択の削除</li> <li>● ドライバ選択のコピー</li> <li>● ドライバ選択のルールの削除</li> <li>● ドライバ選択例外の削除</li> </ul>
インポート操作のグループ化	<ul style="list-style-type: none"> <li>● ステージのインポート</li> <li>● POV のインポート</li> <li>● ドライバのインポート</li> <li>● ドライバ選択のインポート</li> <li>● 割当てのインポート</li> </ul>

監査領域	使用可能な監査タスク
エクスポート操作のグループ化	<ul style="list-style-type: none"> <li>● ステージのエクスポート</li> <li>● POVのエクスポート</li> <li>● ドライバのエクスポート</li> <li>● ドライバ選択のエクスポート</li> <li>● 割当てのエクスポート</li> </ul>
ライフサイクル管理(LCM)	<ul style="list-style-type: none"> <li>● ファイルからの LCM ロード</li> <li>● ファイルへの LCM 保存</li> <li>● LCM エクスポート</li> <li>● LCM インポート</li> </ul>

5 「OK」をクリックします。

監査構成が保存されたことを確認するメッセージが表示されます。

6 「はい」をクリックします。

7 **オプション:** 『Oracle Hyperion Enterprise Performance Management System セキュリティ管理ガイド』の説明に従って、監査結果を確認する監査レポートを生成します。

## 出力ログ・ファイル

管理者は、EPM Workspace 全体を通じてログ・ファイルを生成して、技術者がシステムまたは環境の問題を識別したり、開発者がレポートまたは API の問題をデバッグするのに役立てることができます。

Profitability and Cost Management に関する情報は、次のログ・ファイルで調べることができます:

表 5 Profitability and Cost Management のログ・ファイル

ログ・ファイル	説明
hpcm.log	<p>Profitability and Cost Management は、アプリケーションまたはサーバーから送信されるアプリケーション固有のメッセージを収集するアプリケーション、サーバー側のログ・ファイルを生成します。</p> <p>このログ・ファイルは、デフォルトでは C:\oracle\Middleware\user_projects\domains\EPMSys\servers\Profitability0\logs にあります。</p> <p>このログ・ファイルへのアクセスについては、システム管理者にお問い合わせください。</p>
SharedServices_Security_Client.log	<p>Shared Services のクライアント側のログ・ファイルで、Profitability and Cost Management と Common Security Services とのハンドシェイクに関する詳細を提供します。このログ・ファイルは、デフォルトでは C:\oracle\Middleware\user_projects\domains\EPMSys\servers\Profitability0\logs にあります。</p>

他の関連する製品とアプリケーションのログ・ファイルについては、『Oracle Hyperion Enterprise Performance Management System インストールおよび構成ガイド』を参照してください。

ログ・ファイルで取得する詳細のレベルを変更するには、『Oracle Hyperion Enterprise Performance Management System インストールおよび構成トラブルシューティング・ガイド』のシステム・ログの使用に関する項を参照してください。

## Oracle Diagnostic Logging (ODL)ファイル

各 EPM Workspace 製品のすべての構成ロギング・ファイルについて、Web アプリケーションの Oracle Diagnostic Logging (ODL)の場所は一元化されています。

Profitability and Cost Management の場合、構成ロギング・ファイルは次の場所にあります:

C:\oracle\Middleware\user\_projects\domains\EPMSys\config  
\fmwconfig\servers\Profitability0。ファイル名は logging.xml です。

構成ロギング・ファイルの追加情報については、『Oracle Hyperion Enterprise Performance Management System インストールおよび構成トラブルシューティング・ガイド』の EPM ログの使用に関する項を参照してください。



---

## 第 II 部

# 標準 Profitability モデルの操作

---

標準 Profitability モデルの操作の内容：

- 標準 Profitability について
- Profitability and Cost Management の標準的な次元およびメンバーの操作
- Performance Management Architect を使用した標準 Profitability アプリケーションおよび次元の管理
- 標準 Profitability のインポート・ステージング・テーブル
- 標準 Profitability のモデル定義データのエクスポート



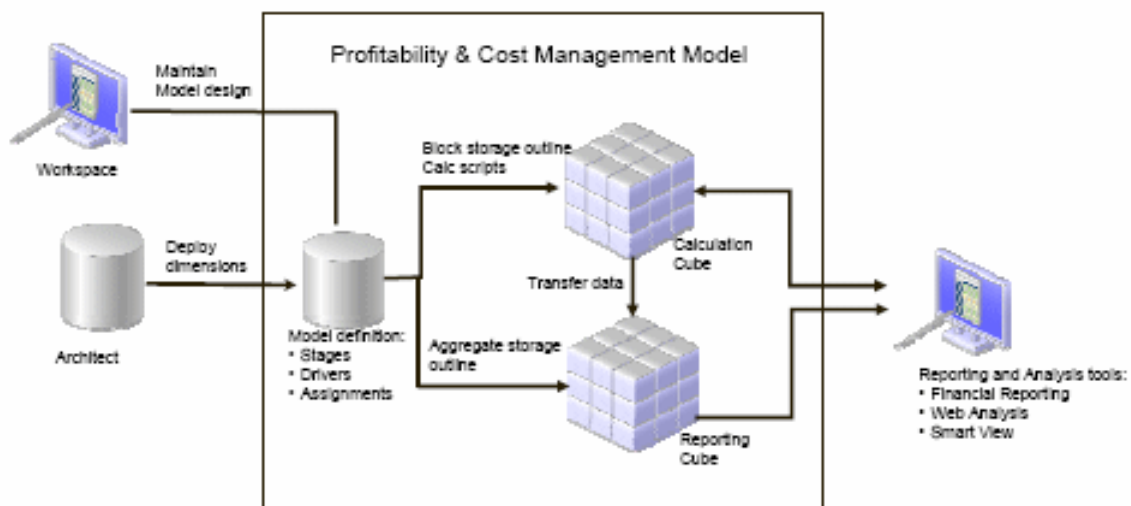
## この章の内容

製品アーキテクチャの理解.....	35
データベースの処理.....	36

## 製品アーキテクチャの理解

EPM Workspace を介してアクセスされる Profitability and Cost Management は、Essbase の最上位に位置する分析アプリケーションです。ビジネス・ユーザーは、このアプリケーションを使用して、収益性および費用管理を目的としてビジネスをモデリングできます。このモデル情報を使用することにより、スクリプト言語を理解しなくても、収益性および費用分析用として調整可能な Essbase データベースを作成できます。

図 1 Profitability and Cost Management の標準的な製品アーキテクチャ



Profitability and Cost Management では、アプリケーションのメタデータとセキュリティを一元的に管理するために Performance Management Architect と Shared Services を使用します。

アプリケーション管理者は、Performance Management Architect を使用して Profitability and Cost Management 次元を作成します。ユーザー・アクセスは、Shared Services を使用して一元的に管理されます。次元メタデータは、準備が整うと、

Profitability and Cost Management アプリケーションまたはモデルに配置されます。Performance Management Architect の次元は、複数のモデルが共有できます。

モデル設計には、モデルの Essbase コンポーネントに必要な Essbase のアウトラインおよび計算スクリプトを生成するのに必要な情報が含まれます。各モデルは、次のデータベースにアクセスする必要があります:

- Performance Management Architect から配置される次元メタデータを含むモデル設計が保管されるリレーショナル・データベース
- 計算データベース(BSO)およびレポート・データベース(ASO)を含む Essbase データベース。

**注:** 複数のモデルを保管するのに必要なデータベースは 1 つのみです。

レポート・データベースの結果は、Oracle Hyperion Financial Reporting、Oracle Hyperion Web Analysis および Oracle Hyperion Smart View for Office などのレポート・ツールや分析ツールで表示できます。

## データベースの処理

Profitability and Cost Management では、リレーショナル・データベースと Essbase データベースの両方が必要です。これらのデータベースおよび接続情報の詳細は、アプリケーションの「モデルの要約」の「システム情報」タブで指定します。詳細は、『Oracle Hyperion Profitability and Cost Management ユーザー・ガイド』を参照してください。

アプリケーションを配置すると、Performance Management Architect により、Profitability and Cost Management のリレーショナル・データベースにテーブルと適切な値が移入されます。Essbase アプリケーションは、モデルの配置中に Profitability and Cost Management に作成されます。各データベースには、表 6 に示すように、特定の情報が保管されます。

表 6 Profitability and Cost Management のデータベース

データベース・タイプ	保管される情報のタイプ
Essbase	<ul style="list-style-type: none"><li>● 次元階層</li><li>● 費用データ</li><li>● 収益データ</li><li>● ドライバ・データ</li><li>● 通貨レート</li><li>● 計算スクリプト</li></ul>

データベース・タイプ	保管される情報のタイプ
製品リレーショナル・データベース	<ul style="list-style-type: none"> <li>● 次元</li> <li>● ステージ定義</li> <li>● POV 定義</li> <li>● ドライバ定義</li> <li>● ドライバ選択</li> <li>● 割当て</li> <li>● 割当てルール</li> <li>● 割当てルールの選択</li> <li>● モデル・プリファレンス</li> </ul>

Profitability and Cost Management では、リレーショナル・データベースに定義されているドライバの定義、ドライバの選択および割当ての情報を使用して、Essbase アプリケーションで計算スクリプトを作成します。この計算スクリプトは、必要なデータ配賦を実行します。



# 4

## Profitability and Cost Managementの標準的な次元およびメンバーの操作

### この章の内容

標準的な Profitability の次元とメタデータについて .....	39
次元タイプ.....	41
メジャー次元.....	41
AllocationType 次元.....	48
別名次元.....	49
ビジネス次元.....	50
POV 次元.....	51
属性次元.....	51
次元ソート順および密度の設定 .....	52

### 標準的な Profitability の次元とメタデータについて

次元およびメタデータは、Profitability and Cost Management 管理者によって Performance Management Architect を使用して作成および管理されます。Profitability and Cost Management では、一般的な次元およびメンバーを使用することにより、Planning などの他の製品で一般的なデータを使用および転送し、時間と労力を削減し、正確さを増すことができます。次元とメタデータの作成および管理の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

Profitability and Cost Management では、Performance Management Architect で作成された次元およびメンバーを使用して、ビジネス・モデルの様々な構造要素を表します:

- AllocationType 次元。直接配賦データと配賦システムデータの両方を保管します
- 次元。ソースや宛先のメジャー次元などのモデルおよびシナリオの構造を実現します。
- ビジネス次元。部署、一般会計の勘定科目、アクティビティ、顧客、製品などの、ビジネス固有のモデル要素を表します
- 視点(POV)次元。期間、シナリオおよびバージョンなど
- 別名次元(必要な場合)
- 属性次元(必要な場合)

次元ごとに、次元タイプと次元名の両方を指定する必要があります:

- 次元タイプは、選択した次元に定義された機能を使用可能にする次元のプロパティです。41 ページの「次元タイプ」を参照してください。
- 次元名は、組織およびビジネスに関して次元のコンテンツを識別するために割り当てられます。たとえば、「勘定科目タイプ」の次元には、「一般会計」や「勘定科目表」などの次元名を付ける場合があります。次元名に次元タイプを反映する必要はありませんが、反映することもできます。命名の制限については、付録 D「Essbase 命名規則」を参照してください。

Performance Management Architect を使用することにより、次元はアプリケーション内に共有またはローカルとして存在できます:

- 共有次元は、Performance Management Architect の共有ライブラリ内に存在し、複数の製品およびアプリケーションが使用できます。
- ローカル次元は、Profitability and Cost Management タイプのアプリケーションなどの 1 つのアプリケーション・インスタンス内にのみ存在する、分離された独立次元です。これらの次元は、これらが作成されているアプリケーション内でのみ使用され、同じタイプの別のアプリケーションであっても表示または使用できません。

Performance Management Architect を介して、別の製品内に存在する次元およびメンバーを選択したり、そのモデル専用の新しい次元およびメンバーを作成します。Profitability and Cost Management モデルに対して次元およびメンバーが選択された後、これらは次元ライブラリで自動的に使用可能になります。配置後は、Profitability and Cost Management アプリケーションで次元およびメンバーを使用できます。

Performance Management Architect には、システム生成とユーザー定義の両方の次元およびメンバーが存在する必要があります。次元には、英数字や計算値が含まれる場合があります。

---

**注意** 作成できる次元およびメンバーの数に制限はありませんが、次元構造が大きくなると、パフォーマンス上の問題が発生します。

---

Profitability and Cost Management モデルの次元のソート順は、処理および計算機能を最大限に高めるために特定の順序で設定する必要があります。53 ページの「次元ソート順の設定」を参照してください。

次元とメンバーの作成および管理の手順は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。次元およびメンバーの命名規則については、付録 D「Essbase 命名規則」を参照してください。

Profitability and Cost Management の次元の詳細は、次の各項を参照してください:

- 41 ページの「メジャー次元」
- 48 ページの「AllocationType 次元」
- 49 ページの「別名次元」
- 50 ページの「ビジネス次元」



- 51 ページの「POV 次元」
- 51 ページの「属性次元」

## 次元タイプ

- システム次元(メジャー次元や AllocationType 次元など)は、この項に記載されている要件に基づいて作成する必要があります:
  - 41 ページの「メジャー次元」
  - 48 ページの「AllocationType 次元」
- 別名次元は、次元の定義に役立つ代替の名前、説明、言語またはその他のアイテムを割り当てるために使用されます。
- ビジネス次元は、部署、総勘定元帳の勘定科目、アクティビティ、顧客または製品などのモデル内のビジネス要素を説明するために作成されます。50 ページの「ビジネス次元」を参照してください。
- 属性次元を使用すると、次元メンバーの属性または品質に基づく分析が可能になります。属性は、製品のサイズや色など、データの特性を示します。
- ユーザー定義属性(UDA)を使用すると、関連付けられた別の次元のメンバーの分類に使用できるテキスト属性に基づいた分析が可能になります。たとえば、New Products という UDA を追加できます。
- POV 次元は、年、シナリオまたは期間など、モデルに固有の視点またはバージョンを示します。モデルごとに少なくとも 1 つの POV 次元が必要です。

また、バージョン次元タイプを作成して、同じ POV の異なるバージョンを保持できます。バージョンは、同じモデルの異なるバージョンの追跡に使用され、変更やモデルに対する異なる戦略の影響を評価できます。
- 一部の Performance Management Architect の次元タイプは、Profitability and Cost Management のモデルで使用できます:
  - 勘定科目
  - エンティティ
  - 国
  - 通貨

これらの次元タイプの使用の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

**注：** 次元アウトラインを定義する場合、命名用として使用できない制限文字があります。『Oracle Essbase データベース管理者ガイド』の Essbase 命名規則に関する項を参照することをお勧めします。

## メジャー次元

メジャー次元には、モデルの構築、検証および計算に必要なメンバーが含まれています。

メジャー次元は、次の2つのいずれかの方法を使用して Performance Management Architect で作成されます:

- ユーザーがアプリケーション・ウィザードを使用して Profitability and Cost Management アプリケーションを作成し、「ローカル次元の自動作成」オプションを選択します。これにより、メジャー・タイプ次元が自動的に生成されます。
- ユーザーがアプリケーション・ウィザードから「ブランク・アプリケーションの作成」を選択することにより、アプリケーションを手動で作成します。ユーザー独自の次元を作成し、メジャー次元タイプを選択する必要があります。

メジャー次元に必要な変更できないメンバー以外にも、ユーザーは、メンバー「UserDefinedDriverMeasures」の下の階層にユーザー定義のドライバ・メジャー(つまりサブ階層)を追加できます。これらの次元メンバーはユーザーが変更できません。

**注:** 必要であれば、メジャー・タイプ次元に別の名前を指定できます。

メジャー次元内では、メジャーのタイプを次のカテゴリに分けることができます:

- ドライバ・メジャーは、ドライバのタイプと式の作成に使用されます。42 ページの「[ドライバ・メジャー](#)」を参照してください
- 費用レイヤー配賦メジャーは、計算済費用および入力済費用の配賦の制御に使用されます。44 ページの「[費用レイヤー配賦メジャー](#)」を参照してください。
- 収入レイヤー配賦メジャーは、計算済収入および入力済収入の配賦の制御に使用されます。45 ページの「[収益レイヤー配賦メジャー](#)」を参照してください。
- レポート・メジャーは、計算済の値および入力済の値を使用して、モデルの総費用および総収益のレポートを生成するために使用します。レベル0以外のすべてのレポート・メジャーが計算されます。47 ページの「[レポート・メジャー](#)」を参照してください。

## ドライバ・メジャー

表7 ドライバ・メジャー

メンバー名	別名	説明	計算済または入力
FixedDriverValue	FixedDV	固定ドライバ値パラメータを必要とするドライバ・タイプで使用するデフォルトのメジャー	入力
Rate	Rate	レート・パラメータを必要とするドライバ・タイプで使用するデフォルトのメジャー	入力

メンバー名	別名	説明	計算済または入力
Quantity	Qty	数量パラメータを必要とするドライバ・タイプで使用するデフォルトのメジャー	入力
Weight	Weight	加重パラメータを必要とするドライバ・タイプで使用するデフォルトのメジャー	入力
Percentage	Percent	パーセンテージ・ドライバ・タイプで使用するデフォルトのメジャー	入力
CalculatedDriverValue	CDV	割当てで使用されるドライバ式の結果であるメジャー	計算済
TotalDriverValue	TDV	配賦式 DV/TDV の分母として使用されるメジャー	計算済
EffectiveTotalDriverValue	EffTDV	ドライバの定義時に「アイドルを許可」ボックスが選択されているドライバの有効なドライバ合計を保管するために使用されるメジャー。	計算済
OverrideTotalDriverValue	OvrdTDV	配賦式の数値として TotalDriverValue メジャーを上書きするユーザー入力値です。	入力
TotalDriverValueAfterReciprocals	TDVAftRcp	相互割当てにソースが含まれるときに非相互ステージ内配賦およびステージ後配賦の配賦式の数値として使用されるメジャー	計算済
IdleDriverValue	IdleDV	IdleCost の計算用のドライバ値(DV)として使用されるメジャー	計算済
UserDefinedDriverMeasures	N/A	<p>メンバー UserDefinedDriverMeasures は、アプリケーション固有のユーザー定義のドライバ・メジャーが保管される場所です。</p> <p>ASOMember DataStorage および BSOMember DataStorage プロパティは、次のように設定します:</p> <ul style="list-style-type: none"> <li>- このメンバーに子がない場合、<b>StoreData</b> に設定します。</li> <li>- このメンバーにメンバーが子として追加されるときに、これらの子に集計記号 IGNORE がある場合、<b>LabelOnly</b> に設定します。</li> </ul> <p><b>注:</b> すべてのドライバ・メジャーはアウトライン内で一意である必要があります。アウトライン内の次元では既存のドライバ・メジャーの名前を他のメンバーの名前として使用しないでください(システム次元、POV 次元およびビジネス次元を含む); これらを使用すると、「データの入力」画面に値が正しく表示されません。</p>	N/A

# 費用レイヤー配賦メジャー

表 8 費用レイヤー配賦メジャー

メンバー名	別名	説明	計算済または入力
UnassignedCost	UnAsgCost	すべての割当ておよびアイドル計算が完了した後のソース交差の費用の残り。	計算済
CostAssigned	CostAsg	ソースからステージ後宛先および非相互ステージ内宛先へ割り当てられる費用合計	計算済
CostAssignedIntraStage	CostAsgInt	相互宛先を除く、ステージ内宛先へ割り当てられる費用合計	計算済
CostAssignedPostStage	CostAsgPost	ステージ後宛先へ割り当てられる費用合計	計算済
OverDrivenCost	OverDrivenCost	標準基準ドライバの場合、割り当てられる費用合計が NetCostForAssignment より大きい場合、超過した金額は OverDrivenCost に転記されます。	計算済
IdleCost	IdleCost	アイドル費用は、ドライバのタイプによって異なる方法で生成されます: <ul style="list-style-type: none"> <li>実績基準ドライバの場合、アイドル費用は次の配賦式を使用して生成されます: <math>IdleDriverValue / OverrideTotalDriverValue</math></li> <li>標準基準ドライバの場合、割り当てられる費用合計が NetCostForAssignment より小さいとアイドル費用が生成されます。</li> </ul>	計算済
NetCostForAssignment	NetCostAsg	前ステージ割当て、ステージ内割当ておよび相互割当てをすべて計上した後で割当てに使用できる総費用 DataStorage (BSO) プロパティを「StoreData」に設定します。	計算済
GrossReceivedCost	GrRecCost	相互費用およびユーザー入力費用を除く、ステージ前割当ておよびステージ内割当てから割り当てられたすべての費用合計 DataStorage (BSO) プロパティを「StoreData」に設定します。	計算済
StandardCostRate	StandardCostRate	標準基準ドライバの場合、ユーザーは標準の費用レートを割り当て、標準基準の費用ドライバの計算で使用する値を次のように入力します: $CostReceivedPriorStage = StandardCostRate * TotalDriverValue$	入力
CostInput	CostInput	交差のユーザー入力の費用値	入力

メンバー名	別名	説明	計算済または入力
CostReceived	CostRec	相互割当てから発生した費用を除く、ステージ前割当ておよびステージ内割当てによって交差に割り当てられたすべての費用合計  DataStorage (BSO)プロパティを「StoreData」に設定します。	計算済
CostReceivedPriorStage	CostRecPri	ステージ前からの割当て時に受け取られた費用合計	計算済
CostReceivedIntraStage	CostRecInt	相互割当てを除く、ステージ内割当て時に受け取られたすべての費用合計	計算済
NetReciprocalCost	NetRcpCost	ステージ後宛先および非相互ステージ内宛先へ割当て可能な金額に対する相互割当ての正味の影響  DataStorage (BSO)プロパティを「StoreData」に設定します。	計算済
ReciprocalCostAssigned	RcpCostAsg	相互割当てから受け取る費用を除く、相互宛先へ割り当てられる費用合計	計算済
ReciprocalCostReceived	RcpCostRec	相互宛先から受け取る費用合計	計算済
ReciprocalIntermediate Cost	RcpIntCost	同時等式が適用された後だが相互調整が行われる前に交差について計算される中間値	計算済
CostPerDrvUnit	Cost Per Driver Unit	このメジャーは、AllocationMeasures の子です。  この式では、割当て費用(CostAssigned)を全ドライバ値の合計(TotalDriverValue)で除算して、ドライバ値の1単位当たりの費用を計算します。	計算済
UnitCost	Unit Cost。	このメジャーは、AllocationMeasures の子です。  この式では、ソース交差の費用金額(NetCostForAssignment)をユーザーが入力した数量で除算して、1単位当たりの費用を計算します。	計算済

## 収益レイヤー配賦メジャー

表 9 収益レイヤー配賦メジャー

メンバー名	別名	説明	計算済/入力
UnassignedRevenue	UnAsgRev	すべての割当ておよびアイドル計算が完了した後のソース交差の収益の残り	計算済

メンバー名	別名	説明	計算済/入力
RevenueAssigned	RevAsg	ソースからステージ後宛先および非相互ステージ内宛先へ割り当てられる収益合計	計算済
OverDrivenRevenue	OverDrivenRevenue	標準基準ドライバの場合、収益合計が <code>NetRevenueForAssignment</code> より大きい場合、超過した金額は <code>OverDrivenRevenue</code> に転記されます。	計算済
RevenueAssignedIntraStage	RevAsgInt	相互宛先を除く、ステージ内宛先へ割り当てられる収益合計	計算済
RevenueAssignedPostStage	RevAsgPos	ステージ後宛先へ割り当てられる収益合計	計算済
IdleRevenue	IdleRev	アイドル収益は、ドライバのタイプによって異なる方法で生成されます: <ul style="list-style-type: none"> <li>実績基準ドライバの場合、アイドル収益は次の配賦式を使用して生成されます:  <math display="block">\text{IdleDriverValue} / \text{OverrideTotalDriverValue}.</math> </li> <li>標準基準ドライバの場合、割り当てられる費用合計が <code>NetRevenueForAssignment</code> より小さいと収益費用が生成されます。</li> </ul>	計算済
NetRevenueForAssignment	NetRevAsg	前ステージ割当て、ステージ内割当ておよび相互割当てをすべて計上した後に使用可能な総収入 DataStorage (BS0)プロパティを「StoreData」に設定します。	計算済
GrossReceivedRevenue	GrRecRev	相互割当て収益およびユーザー入力収益を除く、ステージ前割当ておよびステージ内割当てから割り当てられたすべての収益合計 DataStorage (BS0)プロパティを「StoreData」に設定します。	計算済
StandardRevenueRate	StandardRevenueRate	標準基準ドライバの場合、ユーザーは標準の収益レートを割り当て、標準基準の収益ドライバの計算で使用する値を次のように入力します: $\text{RevenueReceivedPriorStage} = \text{StandardRevenueRate} * \text{TotalDriverValue}$	入力
RevenueInput	RevInput	交差のユーザー入力の収益値 収益カテゴリを <code>RevenueInput</code> 下の階層として定義し、保管します。 DataStorage (BS0)プロパティを「StoreData」に設定します。	入力
RevenueReceived	RevRec	相互費用から発生した収益を除く、ステージ前割当ておよびステージ内割当てによって交差に割り当てられたすべての収益合計 DataStorage (BS0)プロパティを「StoreData」に設定します。	計算済
RevenueReceivedPriorStage	RevRecPri	ステージ前からの割当て時に受け取られた収益合計	計算済

メンバー名	別名	説明	計算済/入力
RevenueReceivedIntraStage	RecRecInt	相互割当てを除く、ステージ内割当て時に受け取られたすべての収益合計	計算済
NetReciprocalRevenue	NetRcpRev	ステージ後宛先および非相互ステージ内宛先へ割当て可能な収益金額に対する相互割当ての正味の影響 DataStorage (BS0)プロパティを「StoreData」に設定します。	計算済
ReciprocalRevenueAssigned	RcpRevAsg	相互割当てから受け取る収益を除く、相互宛先へ割り当てられる収益合計	計算済
ReciprocalRevenueReceived	RcpRevRec	相互宛先から受け取る収益合計	計算済
ReciprocalIntermediateRevenue	RcpIntRev	同時等式が適用された後だが相互調整が行われる前に交差について計算される中間値	計算済

## レポート・メジャー

表 10 レポート・メジャー

メンバー名	別名	説明	計算済または入力
Profit	Profit	選択した交差について計算された収益の値。この値は、次の計算の結果です: $\text{NetRevenueForAssignment} - \text{NetCostForAssignment}$	計算済
GrossCost	GrossCost	使用可能なすべての入力を含む、交差の費用合計: <ul style="list-style-type: none"> <li>● 入力値</li> <li>● ステージ前割当て</li> <li>● 相互割当てを含むステージ内割当て。</li> </ul> この計算は、交差の真の費用合計を表します。	計算済
StandardCost		標準基準ドライバの場合、 $\text{StandardCostRate} * \text{TotalDriverValue}$ で計算された費用	計算済
StandardRevenue		標準基準ドライバの場合、 $\text{StandardRevenueRate} * \text{TotalDriverValue}$ で計算された収益	計算済
InitialCost	InitialCost	入力費用およびステージ前からの割当て時に受け取られた費用を含む、ステージ内費用または相互費用が計算される前の交差の費用。	計算済
NetCostAfterIntraStage	NetCostAftInt	すべてのステージ内費用割当てを含む、交差の費用	計算済



メンバー名	別名	説明	計算済または入力
GrossRevenue	GrossRev	使用可能なすべての入力を含む、交差の収益合計: <ul style="list-style-type: none"> <li>● 入力値</li> <li>● ステージ前割当て</li> <li>● 相互割当てを含むステージ内割当て。</li> </ul> <p>この計算は、交差の真の収益合計を表します。</p>	計算済
InitialRevenue	InitialRev	入力収益およびステージ前からの割当て時に受け取られた収益を含む、ステージ内収益または相互収益が計算される前の交差の収益。	計算済
NetRevenueAfterIntra Stage	NetRevAftInt	すべてのタイプのステージ内収益割当てを計上した後の交差の収益	計算済

## AllocationType 次元

AllocationType 次元は、直接配賦データと配賦システムデータの両方を保管するために使用します。レポートを作成する場合、AllocationType 次元を使用して、取得する配賦データのタイプを指定できます。

**注：** 必要であれば、AllocationType タイプの次元に別の名前を指定できます。

ユーザーがウィザードを使用してアプリケーションを作成し、「ローカル次元の自動作成」を選択すると、AllocationType 次元は自動的に生成されます。「ブランク・アプリケーションの作成」を選択する場合、ユーザー独自の次元を作成し、AllocationType 次元タイプを選択する必要があります。

この次元は変更できず、Performance Management Architect または Profitability and Cost Management では表示されません。

Profitability and Cost Management によって作成される Essbase アウトラインでは、AllocationType 次元には次のメンバーが含まれます:

- **AllAllocations** には、次の子メンバーが含まれます:
  - **DirectAllocation** には、モデル内で指定したソース交差と宛先交差間で直接配賦された計算済データが保管されます。直接配賦は、ユーザーによって割当てとして定義される必要があります。
  - **GenealogyAllocation** には、モデル内の様々な関連交差間の間接リンクで計算された配賦システムが保管されます。

システム配賦は、ユーザーによって直接は定義されませんが、複数の直接配賦のために存在します。たとえば、A-B-C 配賦システムデータが存在するのは、A から B(A-B)および B から C(B-C)への直接配賦が存在するためです。



- **SysAllocVar1** には、仮想リンクでのステージ内割当ての値が保管されます。この値は、同じステージの別のノードに配賦されており、DirectAllocation メンバーですすでに使用可能な値の一部です。
- **SysAllocVar2** は、DirectAllocation、Genealogy Allocation および SysAllocVar1 のソース・リンクの合計を取得するために使用されます。
- **SysAllocVar3** には、システム内で使用される計算済システムデータが保管されます。このメンバーは、レポートには使用しないでください。
- **TotalAllocation** は、DirectAllocation、GenealogyAllocation および SysAllocVar3 のソース・リンクの合計を動的に計算します。
- **IndirectAllocation** は、GenealogyAllocation および SysAllocVar3 のソース・リンクの合計を動的に計算します。

---

**注意** この次元のシステム・メンバーは編集しないでください。変更を行うと、データの損失やモデルの破損が生じる可能性があります。

---

## 別名次元

別名は、次元の定義に役立つ代替の名前、説明、言語またはその他のアイテムです。たとえば、システム内で顧客番号を参照できますが、画面上に顧客名を表示する別名を割り当てることにより、この顧客を簡単に識別できるようになります。1つ以上の別名を勘定科目、通貨、エンティティ、シナリオ、期間、バージョン、年およびユーザー定義の次元メンバーに割り当てることができます。

**注：** 同じ次元内で重複するメンバー名または別名は使用できません。

Profitability and Cost Management では、別名を Performance Management Architect に設定する必要があります。別名次元の作成の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

---

**注意** アプリケーションが Profitability and Cost Management に再配置されるまでは、別名の関連付けを Performance Management Architect で削除しても、モデルからは削除されません。

---

インストール後、Default 別名テーブルが使用可能になります。再配置後は、「ドライバの選択」や「割当て」など、共通メンバー・セクタを使用するすべての画面で別名を表示できます。別名を検索およびフィルタ処理するには、先にコンテキスト・メニューで「別名の表示」を選択します。





**注：** 別名ビューは「ドライバの選択」セクタには含まれません(このセクタには、ドライバの選択を追加または変更するとき、「ドライバ定義」画面または「ドライバの選択」画面からアクセスできます)。

特定の次元が Essbase で複製されている場合、別名が複製される場合があります。


▶ 別名を表示するには:

1 アプリケーションから、共通メンバー・セレクタを使用する任意の画面(「ドライバの選択」、「割当て」、「データの入力」、「配賦のトレース」など)を選択します。共通メンバー・セレクタを含む該当する画面が表示され、使用可能なすべてのメンバーが表示されます。

2 オプション: メンバーを選択するには:

- 「ドライバの選択」で、「選択」または「追加」をクリックします。
- 「ドライバ・ルール」タブおよび「例外」タブで、「新しい例外の追加」をクリックし、次元を選択してから「選択」をクリックします。
- 「配賦のトレース」で、「選択」をクリックします。

「メンバーの選択」ダイアログ・ボックスが開き、使用可能なすべてのメンバーが表示されます。

3 共通メンバー・セレクタで、「コンテキスト・メニュー」ボタンをクリックし、「別名の表示」を選択します。

別名がメンバー・リストに表示されます。

**注:** 「コンテキスト・メニュー」から「別名の表示」を選択したときに、別名が割り当てられていない場合、メンバー名が大かっこで囲まれて表示されます。たとえば、メンバー名 Product は、[Product]と表示されます。

## ビジネス次元

ビジネス次元は、製品、顧客、地域、従業員など、モデルの各ステージ内のビジネス固有のオブジェクトを示します。これらの次元およびメンバーは、Performance Management Architect で作成されます。

ビジネス次元は、次の次元タイプの一部または全部を使用し、1つ以上のステージまたはモデルに適用される場合があります:

- 汎用
- 勘定科目
- エンティティ
- 国

Essbase アウトラインが配置されると、Profitability and Cost Management アプリケーションでビジネス次元がタイプのない基本次元または汎用次元として作成されます。この機能を使用すると、Profitability and Cost Management で、Planning などの他のアプリケーションで定義されている次元メンバーおよび階層を再利用できます。

**注:** この次元タイプは、集約ストレージ・アウトラインには適用されません。

ビジネス次元を作成する場合、次の要件が適用されます:

- 次元の Gen1 メンバーの次のプロパティは LABEL\_ONLY に設定する必要があります:
  - DataStorage(BSO)
  - DataStorage(ASO)
- 通常、Gen1 次元名の下にある最初の世代 2 の子は、ALL メンバーに設定されます。たとえば、Departments 次元の場合は AllDepartments になります。  
プライマリ階層は、最初の世代 2 の子の下でホストされます。割当モデリングでは最初の世代 2 階層のみが使用され、この階層に共有メンバーを含めることはできません。
- 追加の世代 2 メンバーは代替階層をホストできますが、これらの階層は割当モデリングでは使用されません。次元が代替階層をホストする場合は、次元の階層タイプを「使用可能」に設定し、最初の世代 2 メンバーの階層タイプを「保管済」、共有メンバーを含む代替階層の世代 2 メンバーを「動的」に設定します。  
これらの代替階層は、Profitability and Cost Management モデリング画面には表示されず、Essbase でのみ表示できます。
- NoMember メンバーが必要です。階層内の最後の世代 2 の子は常に NoMember であるとともに、集計が IGNORE (~) に設定されている必要があります。

**注:** NoMember メンバーは、Performance Management Architect によって自動的に追加されるため、ADS ファイルまたはフラット・ファイルに作成する必要はありません。このメンバーは削除しないでください。

## POV 次元

POV 次元は、年、シナリオまたは期間など、モデルに固有の視点またはバージョンを示します。次元は、組織の要件を反映するようにカスタマイズできます。たとえば、四半期、月、季節ごとのグループなどで POV を構成できます。

各モデルには 1 つ以上の POV 次元が必要で、最大 4 つの POV 次元を作成できます。POV 次元は、モデルの Performance Management Architect プロパティ・グリッドで設定します。

バージョン次元も使用可能で、モデルの異なるインスタンスの作成に使用されます。このバージョンは、仮定シナリオを実行して、戦略やビジネス・オプションを試行できるよう変更可能です。バージョンを変更することで、機能を実装し、最善の方策を決定するために結果を比較できます。

## 属性次元

属性次元は、ビジネス次元に関連付けられた次元の特殊なタイプで、関連付けられたその他の次元のメンバーを分類する際に使用可能なメンバーが含まれます。属性次元は、製品のサイズ、色など、データの特性を示します。

これらの属性を使用して、次元メンバーの属性または品質に基づいてデータを分析できます。属性次元は、割当てルールの作成時に宛先交差のフィルタ処理にも使用されます。

**注：** 命名の制限については、[付録 D 「Essbase 命名規則」](#) を参照してください。

2つの属性次元タイプが Performance Management Architect に存在します：

- 属性次元：
  - ブール、日付、数値、テキストなど、異なる構造を使用して属性を作成できます。
  - 属性には階層があり、集計値にロールアップすることができます。
  - 所定の属性次元の1つの属性のみを1つのメンバーに関連付けできます。
- ユーザー定義属性次元(UDA)：
  - 属性は、テキストを使用して作成する必要があります。
  - UDA には階層がないため、合計を生成するためのレポートに容易に使用することはできません。
  - 複数の UDA を1つのメンバーに関連付けることができます。

モデルやレポートの要件により、属性次元の利点はタイプごとに異なります。属性次元の使用の詳細は、『Oracle Essbase データベース管理者ガイド』を参照してください。

これらの属性を使用して、次元メンバーの属性または品質に基づいてデータを分析できます。属性次元は、割当てルールの作成時に宛先交差のフィルタ処理にも使用されます。

命名の制限については、[付録 D 「Essbase 命名規則」](#) を参照してください。

## 次元ソート順および密度の設定

Dimension Sort Order プロパティは、Profitability and Cost Management によって生成される Essbase アウトラインにおける次元の順序を制御します。次元ソート順は、別名または UDA を除く、モデル内のすべての次元に対して設定する必要があります。

---

**注意** 次元のソート順を空のままにすると、検証は失敗します。

---

Dimension Sort Order プロパティは、Performance Management Architect で設定され、Essbase アウトラインの生成に使用するための配置時に Profitability and Cost Management に渡されます。詳細は、[53 ページの「次元ソート順の設定」](#) を参照してください。

モデルの次元ソート順設定は、Performance Management Architect で検証されます。[70 ページの「標準 Profitability アプリケーションの設定の変更」](#) を参照してください。

「次元のソート順設定」ダイアログ・ボックスから、次元の密度を設定することもできます。

## 次元ソート順の推奨事項

次の推奨事項を参照して、次元ソートを設定することをお勧めします:

- 別名と UDA を除く、モデル内のすべての次元ソート順が設定されている。

**注:** 別名次元と UDA 次元は、Profitability and Cost Management と Essbase には次元として存在しないため、次元ソート順では無視されている。

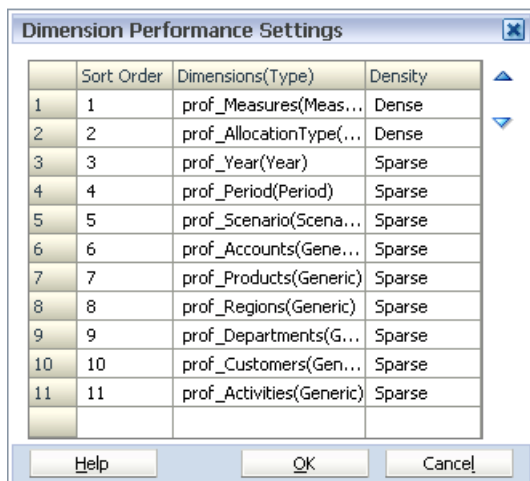
- 次元ソート順が連続しており、一意かつ 1 以上である必要があります。
- メジャー次元がデフォルトで 1 に設定されている。
- AllocationType 次元がデフォルトで 2 に設定されている。
- ビジネス次元と POV 次元が 3 以上に設定されている。
- 属性次元が常に最後の次元としてソートされている。たとえば、12 個の連続した次元のうち 4 個が属性次元である場合、属性次元は 9、10、11 および 12 に設定する必要があります。




## 次元ソート順の設定

モデル内のすべての次元の処理順序は、Dimension Sort Order プロパティを使用して次元レベルで設定する必要があります。次元ソート順の制限を満たす必要があります; 制限を満たさない場合、モデルの検証は失敗します。制限の完全なリストについては、53 ページの「次元ソート順の推奨事項」を参照してください。

Performance Management Architect で次元ソート順を設定する方法は 2 つあります:

- すべての次元のソート順を一度に設定するには、「次元パフォーマンスの設定」ダイアログ・ボックスを使用します。
  - 1 つの次元の個々の次元ソート順を一度に設定するには、次元ライブラリのプロパティ・グリッドを使用します。
- ▶ 次元パフォーマンスの設定ダイアログ・ボックスを使用して次元ソート順を設定するには:
- 1 EPM Workspace から、「ナビゲート」、「管理」、「次元ライブラリ」の順に選択し、共有ライブラリを表示します。
  - 2 アプリケーション名を右クリックし、「次元パフォーマンスの設定」を選択します。



- 3 53 ページの「次元ソート順の設定」で説明されているように、次元を選択し、上矢印  上矢印および下矢印  下矢印を使用して各次元を正しいソート順に移動します。「ソート順」に、次元の元の場所の番号が表示されます。
  - 4 オプション: 選択した次元の「密度」で、オプションを表示するセルをダブルクリックし、次元の適切な密度を選択します。56 ページの「Essbase の次元設定の最適化」を参照してください。
  - 5 「OK」をクリックします。
- ▶ 個々の次元の次元ソート順を設定するには:
- 1 EPM Workspace から、「ナビゲート」、「管理」、「次元ライブラリ」の順に選択します。
  - 2 Profitability and Cost Management アプリケーションで、次元ソート順を設定する次元を選択します。
  - 3 プロパティ・グリッドで、Dimension Sort Order プロパティを選択し、「値」にソートの順序に必要な数字を入力します。53 ページの「次元ソート順の設定」を参照してください。
  - 4 「保存」アイコン  をクリックします。

# 5

## Performance Management Architectを使用した標準 Profitabilityアプリケーション および次元の管理

### この章の内容

アプリケーションおよび次元の操作.....	55
Essbase の次元設定の最適化.....	56
メタデータのインポート.....	57
Profitability and Cost Management の次元プロパティおよびメンバー・プロパティ.....	58
階層プロパティ・タイプの設定.....	63
標準 Profitability アプリケーションの作成.....	64
標準 Profitability の次元の選択.....	65
標準 Profitability アプリケーションの設定の変更.....	70
Performance Management Architect でのアプリケーションの検証と配置.....	74
データの同期.....	79

## アプリケーションおよび次元の操作

Profitability and Cost Management では、Performance Management Architect を使用して、収益性モデルに使用される Essbase アウトラインを構築するための次元を選択します。すべての次元およびメンバーは、Performance Management Architect で作成され、モデルを構築するために Profitability and Cost Management アプリケーションにインポートされます。

**注：** ライフサイクル管理は、モデル・データのインポートやエクスポートだけでなく、Performance Management Architect 内のアプリケーションのインポートやエクスポートにも使用できます。

Performance Management Architect では、次のタスクを実行できます：

- 次元の作成、編集およびコピー
- 別名の設定
- 次元の関連付けの作成、表示および削除
- メンバーの作成、表示、名前変更および削除
- プロパティ値の編集



- Profitability and Cost Management へのアプリケーションの配置
- トランザクション・ログの表示
- Profitability and Cost Management と他のアプリケーション、Essbase データベース(ASO および BSO)、外部ソース(フラット・ファイル)、およびインタフェース・テーブル間のデータの同期化

Performance Management Architect を使用したメタデータおよび次元の操作の詳細は、次の項を参照してください:

- [56 ページの「Essbase の次元設定の最適化」](#)
- [57 ページの「メタデータのインポート」](#)
- [58 ページの「Profitability and Cost Management の次元プロパティおよびメンバー・プロパティ」](#)
- [64 ページの「標準 Profitability アプリケーションの作成」](#)
- [74 ページの「Performance Management Architect でのアプリケーションの検証と配置」](#)
- [79 ページの「データの同期」](#)

## Essbase の次元設定の最適化

一般的な Profitability and Cost Management アプリケーションには、1つのメジャー次元、1つの AllocationType 次元、複数の POV 次元、および多数のビジネス次元が含まれます。Profitability and Cost Management では、ビジネス次元が複数のステージで使用される場合、ビジネス次元が重複します。このプロセスにより、アプリケーションによって生成される Essbase 計算キューブ・アウトラインはより疎になり、計算スクリプトの実行時のパフォーマンスに影響を与える可能性があります。

Oracle Essbase Administration Services (EAS) コンソールを使用して、生成された計算キューブ・アウトラインの次元ストレージ・タイプを直接変更することにより、ある程度の最適化は簡単に実現できます。

---

**注意** 「次元ストレージ・タイプ」プロパティの変更は、データベース管理者 (DBA) のみが実行する必要があります。指示の詳細は、『Oracle Essbase Administration Services 開発者ガイド』を参照してください。

---

各次元の「次元ストレージ・タイプ」プロパティの設定に関するデフォルトの推奨事項は、次のとおりです:

- メジャー次元および AllocationType 次元は密に設定します
- ビジネス次元および POV 次元はすべて疎に設定します

このデフォルト設定の場合、Essbase 計算キューブのブロック・サイズは約 3K になり、疎次元の次元性に基づく多数の潜在的なブロックが発生します。最大ステー



ジの最も密な次元(データの存在に基づく)は密に設定する必要があります。その場合、AllocationType 次元およびメジャー次元は疎に設定する必要があります。

**注：** このブロック・サイズは、Essbase のベスト・プラクティスの推奨事項の制限内に収まる必要があります。詳細は、『Oracle Essbase Administration Services 開発者ガイド』を参照してください。

## 例

最大ステージは、ステージ内の潜在的なノードの最大数によって定義されます。

一般的な Profitability and Cost Management モデルの最後のステージが顧客 X 製品(顧客 x 製品の収益性の計算用)として定義されており、これが最大ステージである場合、このステージ内の顧客次元または製品次元を密に設定できます。関連するメジャー次元および AllocationType 次元は疎に設定します。顧客を密に設定するか製品を密に設定するかの決定は、管理しているドライバのデータの密度に基づいて行います。

たとえば、前のステージ内の様々なソースからこのステージに配賦するために使用するドライバが販売量であり、製品次元より顧客次元のほうにより密なデータが含まれる(多くの顧客に対して少ない製品が販売される)場合、顧客を密に設定できます。少ない顧客に対して多くの製品が販売される場合、製品を密に設定できます。いずれの場合も、Essbase ブロック・サイズは、Essbase のベスト・プラクティスの推奨事項の制限内に収まる必要があります。

**注：** 属性の関連付けがある次元は密に設定できません。Essbase では、属性の関連付けは疎次元としてのみ設定できます。

## メタデータのインポート

Performance Management Architect の次元ライブラリは、次元および階層を表示、作成および管理するための、管理者用の中心的な場所です。他のアプリケーションと共有する既存の次元を使用することも、自分のモデル専用のローカル次元を作成することもできます。

次元ライブラリを使用するには、次元エディタのセキュリティの役割がある必要があります。この役割では、次元およびメンバーの作成、削除、変更、インポート・プロファイルの作成、トランザクション・ログの実行などのすべての次元ライブラリ機能へのアクセスが可能です。『Oracle Hyperion Enterprise Performance Management System ユーザーおよび役割セキュリティ・ガイド』を参照してください。

Performance Management Architect には、2つのタイプの次元があります：

- 共有次元は、共有ライブラリにリンクされており、次元に対して行われたすべての変更を継承します。
- ローカル次元は、共有ライブラリからアプリケーションへコピーされます。ローカル次元は、共有ライブラリ内の次元に対して行われる将来の変更を継承しません。

---

**注意** Profitability and Cost Management のすべてのプロパティはローカル値です。プロパティが他の階層に存在する場合に、この値を変更しても自動的に継承されません。1つの階層でプロパティを変更しても、他の階層に値が継承されるとは仮定できません。

---

次元ライブラリからこれらのプロパティを編集できます:

- アプリケーション
- 次元
- メンバー

**注:** 時間、通貨および国用の標準次元は、すべての製品を対象として Performance Management Architect で使用できます。

▶ 次元ライブラリにアクセスするには、「ナビゲート」、「管理」、「次元ライブラリ」の順に選択します。

次元およびメンバーを操作する方法の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

## Profitability and Cost Management の次元プロパティおよびメンバー・プロパティ

Profitability and Cost Management の次元およびメンバーのプロパティは、表 11 にアルファベット順で表示され、次の情報が含まれます:

- プロパティ・ラベル。プロパティにわかりやすい表示名を提供します。該当する場合、関連データベース・タイプが名前に付加されます(ASO または BSO)。データベース・タイプが指定されていない場合、プロパティは両方のタイプに適用されます。
- 各プロパティの説明
- プロパティに一意の識別子を提供し、インポートおよびバッチ・クライアントでデータを更新する際に使用するプロパティ名。

次元ライブラリでプロパティを選択すると、ドロップダウン・リストまたはデータ入力ボックスを表すプロパティを変更できます。

次元およびメンバーを操作する方法の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

---

**注意** Profitability and Cost Management のすべてのプロパティはローカル値です。ある階層でプロパティを変更しても、他の階層に値が継承されるわけではありません。

---

表 11 Profitability and Cost Management の次元プロパティおよびメンバー・プロパティ

プロパティ・レベル	説明	プロパティ名
別名	<p>選択した次元メンバーの別名を入力します。別名は、配置したアプリケーションに表示される代替次元メンバー名です。</p> <ul style="list-style-type: none"> <li>● 現在の次元メンバーのデフォルトの別名を割り当てるかまたは変更する場合は、既存の別名をクリックして新規名前を入力します。適切な命名規則に従います。すべてのデータベースには、「Default」という別名テーブルがあります。</li> <li>● 現在の次元メンバーの別名を割り当てるかまたは変更する場合は、別名テーブルの横にある既存の別名をクリックして新規名前を入力します。適切な命名規則に従います。</li> </ul>	
アプリケーション・タイプ	<p>アプリケーション・タイプを選択します:</p> <ul style="list-style-type: none"> <li>● 標準 Profitability の場合は「全般」</li> <li>● 詳細 Profitability の場合は「詳細」</li> </ul>	ApplicationType
属性	<p>次元メンバーの特性を入力します。</p> <p>たとえば、従業員次元メンバーには、名前、年齢、または住所の各属性がある場合があります。Product 次元のメンバーはサイズ、味などの複数の属性を持つ場合があります。</p>	Attributes
属性タイプ	<p>属性タイプを選択したメンバーに関連付けます。メンバーのクエリーおよびフィルタに使用できる次元メンバーに割り当てられた属性の属性値を作成します。</p> <p>属性タイプ:</p> <ul style="list-style-type: none"> <li>● テキスト</li> <li>● ブール</li> <li>● 日付</li> <li>● 数値</li> </ul> <p>たとえば、「カフェイン入り」の属性タイプはブール値になります。データ値は True または False です。</p>	AttributeDataType
コメント	<p>次元またはメンバーのコメントを入力します。コメントには最大 255 文字を指定できます。デフォルトでは、テキスト・ボックスには現在のコメントが表示されます(存在する場合)。</p> <p>属性次元またはメンバーにコメントを割り当てられません。</p> <p><b>注:</b> Unicode 使用可能データベースでは、最大 80 文字まで許可されます。</p>	Comment

プロパティ・ラベル	説明	プロパティ名
集計	<p>メンバー集計プロパティは、子を親にロールアップする方法を指定します。</p> <p>現在のメンバーが次元または属性ではない場合は、そのメンバーに割り当てる集計演算子を選択します:</p> <ul style="list-style-type: none"> <li>● + (加算) - デフォルト</li> <li>● - (減算)</li> <li>● * (乗算)</li> <li>● / (除算)</li> <li>● % (パーセント)</li> <li>● ~ (集計では無視する)</li> <li>● ^ (集計しない)</li> <li>● NotUsed</li> </ul> <p><b>注:</b> 集約ストレージ・アウトラインでは、連結演算子の使用に一部制限があります。詳細は、『Oracle Essbase データベース管理者ガイド』を参照してください。</p>	Consolidation
データ・ストレージ(ASO)	<p>次元およびメンバーの ASO ストレージ・オプションを選択します:</p> <ul style="list-style-type: none"> <li>● StoreData - データは次元とともに保管されます。</li> <li>● ShareData - このメンバーに関連付けられているデータは共有できます。ShareData プロパティは、メンバーのみに適用されません。次元ルート・メンバーは共有できません。</li> <li>● NeverShare - この次元に関連付けられているデータは、単一の子を持つ親などのように、暗黙の共有関係がある場合にも共有できません。この場合、データは親および子で複製されます。</li> </ul> <p><b>注:</b> このオプションは、集約ストレージ・アウトラインに保管された階層には適用されません。</p> <ul style="list-style-type: none"> <li>● LabelOnly - この次元に関連付けられているデータはありません。</li> </ul>	<ul style="list-style-type: none"> <li>● ASODimensionDataStorage (次元ルート・メンバーの場合)</li> <li>● ASOMemberDataStorage (次元メンバーの場合)</li> </ul>
データ・ストレージ(BSO)	<p>次元およびメンバーの BSO ストレージ・オプションを選択します:</p> <ul style="list-style-type: none"> <li>● StoreData - データは次元とともに保管されます。</li> <li>● ShareData - このメンバーに関連付けられているデータは共有できます。ShareData プロパティは、メンバーのみに適用されません。次元ルート・メンバーは共有できません。</li> <li>● NeverShare - この次元に関連付けられているデータは、単一の子を持つ親などのように、暗黙の共有関係がある場合にも共有できません。この場合、データは親および子で複製されます。</li> <li>● LabelOnly - この次元に関連付けられているデータはありません。</li> <li>● DynamicCalcAndStore - この次元に関連付けられているデータは、ユーザーが要求するまで計算されません。計算された後、データは保管されます。</li> <li>● DynamicCalc - この次元に関連付けられているデータは、ユーザーが要求するまで計算されません。データは保管されず、要求の処理が完了した後で破棄されます。</li> </ul>	<ul style="list-style-type: none"> <li>● BSODimensionDataStorage (次元ルート・メンバー用)</li> <li>● BSOMemberDataStorage (次元メンバー用)</li> </ul>

プロパティ・ラベル	説明	プロパティ名
次元式 (ASO)	セルをダブルクリックし、セレクト・ボタンをクリックして、メモ・エディタを開きます。式を入力し、「OK」をクリックします。 適切な ASO MDX 構文を使用します。	ASODimensionFormula (次元ルート・メンバー用)
次元式 (BSO)	セルをダブルクリックし、セレクト・ボタンをクリックして、メモ・エディタを開きます。式を入力し、「OK」をクリックします。 適切な BSO CALCULATOR 構文を使用します。	BSODimensionFormula (次元ルート・メンバー用)
次元解決順	選択した次元のソリューション・シーケンスに数値を入力します。たとえば、この次元を 2 番目に解決する場合は、「2」を入力します。	DimensionSolveOrder
次元ソート順	数値を順番に入力して Profitability and Cost Management によって生成された Essbase アウトライン内の次元の順序を設定します。 たとえば、この次元が Essbase アウトラインの 2 番目の次元である場合は、2 を入力します。 次元ソート順は、別名次元と UDA 次元を除き、モデル内のすべての次元に設定されている必要があります。 次元ソート順が連続しており、一意かつ 1 以上である必要があります。 53 ページの「次元ソート順の設定」に概略されているように、ソート順を設定します。	DimensionSortOrder
次元ストレージ・タイプ	「次元ストレージ・タイプ」プロパティは、Essbase のブロック・ストレージ(BSO)データベースにのみ使用されます。次元に必要なストレージのタイプを選択します: <ul style="list-style-type: none"><li>● 密</li><li>● 疎</li></ul>	DimensionStorageType
階層タイプ (次元のみ)	「階層タイプ」プロパティは、Essbase の集約ストレージ(ASO)データベースにのみ使用されます。レポート用データベースが Profitability and Cost Management によって生成されます。 次元の階層のタイプを設定します: <ul style="list-style-type: none"><li>● 次の集計記号を使用する次元メンバーには「保管済」を選択します:<ul style="list-style-type: none"><li>○ +(加算)</li><li>○ ~(無視)</li></ul><b>注:</b> メンバーの親が LABEL_ONLY に設定されている場合、非集計演算子または無視(~)は、保管済階層にのみ使用できます。</li><li>● 加算や無視などの集計記号を使用する次元メンバーの場合、または次元メンバーに式が含まれる場合は「動的」を選択します。</li><li>● 代替階層をサポートする場合は、「使用可能」を選択します</li></ul>	DimensionHierarchyType

プロパティ・ラベル	説明	プロパティ名
階層タイプ (メンバーのみ)	<p>メンバー元の階層のタイプを設定します:</p> <ul style="list-style-type: none"> <li>● 保管(最初の Gen2 の子用)</li> <li>● 動的(2 番目以降の Gen2 の子用。2 番目の Gen2 メンバーは代替階層をホスト可能)</li> <li>● なし(使用不可)</li> </ul>	HierarchyType
集約でのレベルの使用	<p>このプロパティは、ASO アウトライン内の保管済階層をホストする Gen1 または Gen2 の階層メンバーに適用されます。</p> <p>「DEFAULT」に設定します</p> <p>このプロパティの追加情報については、『Oracle Essbase データベース管理者ガイド』を参照してください。</p>	
メンバー式(ASO)	<p>セルをダブルクリックし、セレクト・ボタンをクリックして、メモ・エディタを開きます。式を入力し、「OK」をクリックします。</p> <p>適切な ASO MDX 構文を使用します。</p>	ASOMemberFormula
メンバー式(BSO)	<p>セルをダブルクリックし、セレクト・ボタンをクリックして、メモ・エディタを開きます。式を入力し、「OK」をクリックします。</p> <p>適切な BSO (BSO CALCULATOR)構文を使用します。</p>	BSOMemberFormula
メンバー解決順 (メンバーのみ)	<p>選択したメンバーのソリューション・シーケンスに数値を入力します。たとえば、このメンバーを 2 番目に解決する場合は、「2」を入力します。</p> <p>このプロパティは、ASO データベースのみに適用されます。</p> <p>解決順が 0 のメンバーは、その次元の解決順序を継承します。</p> <p>解決順が同じメンバーは、次元ソート順プロパティで別の値が指定されていないかぎり、その次元がデータベース・アウトラインに出現する順序で評価されます。</p> <p>解決順のないメンバーは、解決順のあるメンバーの後に評価されます。</p>	MemberSolveOrder
POV 次元	<p>この次元を選択したモデルの POV 次元として設定する場合は、このチェック・ボックスを選択します。</p>	IsPOVDimension
POV 表示順	<p>複数の POV 次元がある場合は、1、2、3 などの数値を入力して、各 POV 次元の表示順を設定します。</p>	POVDisplayOrder

プロパティ・ラベル	説明	プロパティ名
プライマリ・レベルの加重	<p>プライマリ・レベルの加重は、集約操作中のビュー選択エンジンで選択できるレベルを制限します。これらの設定は、Essbase (ASO) アプリケーションのみに適用されます。</p> <p>次のオプションのいずれかを選択します:</p> <ul style="list-style-type: none"> <li>「Default」 - ビュー選択エンジンは選択するレベルを自由に決定できます。これは、新規の階層に対する初期値です。</li> <li>「AllLevels」 - ビュー選択エンジンは階層のすべてのレベルを集約対象とみなします。</li> <li>「NoAggregation」 - ビュー選択エンジンはどのレベルも集約用には選択できません。</li> <li>「TopLevelOnly」 - ビュー選択エンジンは次元の最上位レベルのみを集約対象とみなします。</li> <li>「BottomTop」 - ビュー選択エンジンは、次元の最上位レベルおよび最下位レベルのみを集約対象とみなします。</li> <li>「BottomLevelOnly」 - ビュー選択エンジンは次元の最下位レベルのみを集約対象とみなします。</li> </ul>	PrimaryLevelWeighting
2パス計算 (次元のみ)	BSO データベースの場合のみ、このチェック・ボックスを選択すると、アウトラインを介した2番目のパスのメンバーが計算されます。	TwoPassCalc (次元ルート・メンバー用)
2パス計算 (メンバーのみ)	BSO データベースのみの場合、アウトラインを通じて2番目のパスでメンバーを計算するためのチェック・ボックスを選択します。	TwoPassCalc
UDA	<p>選択したメンバーの UDA メンバーを選択します。</p> <p>UDA 次元は、Performance Management Architect に存在し、UDA が追加される先のベースに関連付けられている必要があります。</p> <p>最大 80 文字(英数字)の UDA メンバーを選択できます。</p> <p><b>注：</b> Unicode 使用可能データベースでは、最大 80 文字まで許可されます。</p>	UDA
Unicode	「Unicode」を選択して、アプリケーションを Unicode モードに設定します。Unicode モードのアプリケーションは複数の文字セットをサポートします。	Unicode

## 階層プロパティ・タイプの設定

「Hierarchy Type」プロパティは、Essbase の集約ストレージ・データベースにのみ使用されます。(レポート・キューブが Profitability and Cost Management によって生成されます。)

その他の集計タイプ記号(加算および無視以外)を使用できるようにするには、階層タイプを「動的」に設定する必要があります。



**注：** 次元メンバーに式がある場合は、階層タイプを「動的」に設定する必要があります。

▶ 階層タイプを設定するには:

- 1 EPM Workspace から、「ナビゲート」、「管理」、「次元ライブラリ」の順に選択します。
- 2 アプリケーションの下で次元を選択します(「別名」、「属性」、「UDA」を除く)。
- 3 プロパティ・グリッドで、適切な階層タイプ・プロパティを選択します:
  - 次の集計記号を使用する次元メンバーには「保管済」を選択します:
    - + ADDITION
    - ~ IGNORE (LABEL ONLY メンバーの下のみ。)
  - サポートされているすべての集計記号を使用する必要のある次元メンバー、または式のある次元メンバーを含む次元には「動的」を選択します。
- 4 Profitability and Cost Management、Essbase の順にアプリケーションを再配置します。

## 標準 Profitability アプリケーションの作成

次の 2 つの方法のいずれかを使用して、Performance Management Architect で標準 Profitability アプリケーションを作成できます:

- アプリケーションおよび次元を自動的に作成する「アプリケーション・ウィザード」を使用して Profitability and Cost Management アプリケーションを作成します。
- ブランク・アプリケーションを作成し、次元を手動で選択して名前を付けます。

▶ Profitability and Cost Management アプリケーションを作成するには:

- 1 フラット・ファイルのインポートまたは Performance Management Architect インタフェース・テーブルのインポートを使用して Performance Management Architect に新規共有ライブラリを移入します。

---

**注意** アプリケーションに含めるビジネス次元(汎用、勘定科目、エンティティ、時間、国など)は、アプリケーションを作成する前に、次元ライブラリに追加する必要があります; これを行わないと、「アプリケーション・ウィザード」での選択に次元を使用できません。

---

- 2 EPM Workspace から、「ナビゲート」、「管理」、「アプリケーション・ライブラリ」の順に選択します。

アプリケーション・ライブラリが表示されます。

- 3 「ファイル」、「新規」、「アプリケーション」の順に選択します

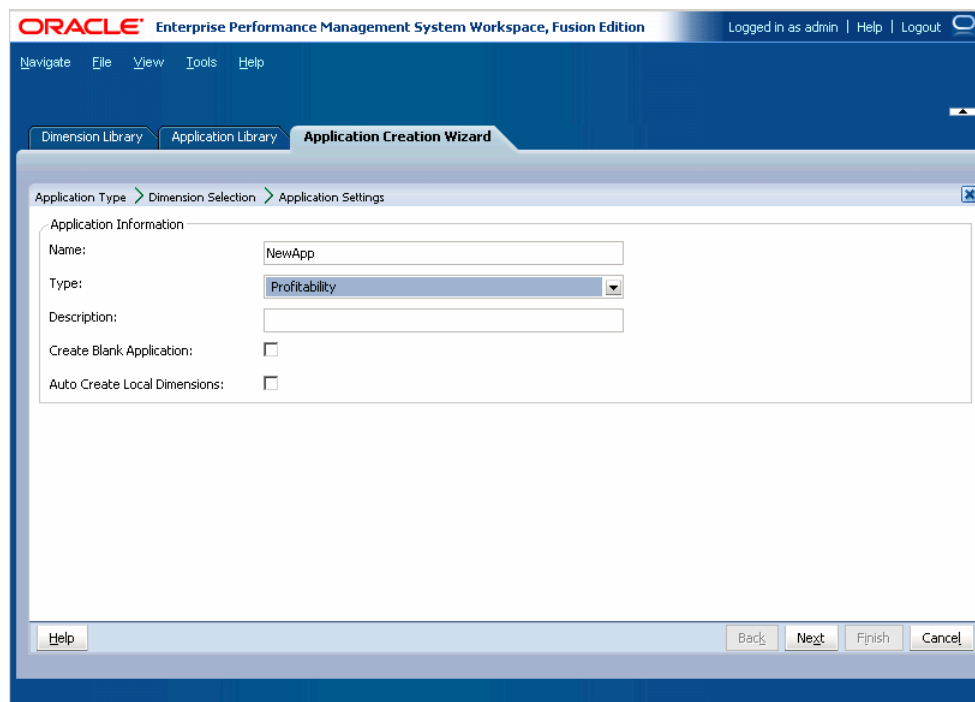
ウィザードの最初の「アプリケーション・タイプ」画面が表示されます。



- 4 「名前」で、アプリケーション名を入力します。

名前は7文字以内で命名し、「&」(アンパサンド)を含む特殊文字を使用しないでください。付録D「Essbase 命名規則」を参照してください。

- 5 「タイプ」で、「Profitability」を選択します。



- 6 オプション: 「説明」に、説明を入力します。

- 7 オプション: 次元を手動でブランク・アプリケーションに追加するには、「ブランク・アプリケーションの作成」を選択し、「終了」をクリックします。69 ページの「標準 Profitability 次元の手動での追加」を参照してください。

- 8 オプション: すべての必須次元を自動的に作成するには、「ローカル次元の自動作成」を選択します。

「ローカル次元の自動作成」を選択すると、アプリケーションに必要なすべての次元について新規次元が自動的に作成されます。各新規次元の次元名は、次元タイプにカッコで囲んだ(新規)が付いた名前になります。

ウィザードの2番目の画面である「次元の選択」が表示されます。

- 9 「次」をクリックします。65 ページの「標準 Profitability の次元の選択」を参照してください。

## 標準 Profitability の次元の選択

標準 Profitability アプリケーションを作成した後、アプリケーションに含める次元を選択する必要があります。

標準 Profitability and Cost Management アプリケーションに必要な次元は次のとおりです:

- メジャー次元には、ドライバ・メジャー、レポート・メジャー、配賦メジャーなど、モデルの構築、検証および計算に必要なメンバーが含まれています。

ユーザーは、メンバー「UserDefinedDriverMeasures」の下の階層にユーザー定義のドライバ・メジャー(つまりサブ階層)を追加できます。

- AllocationType 次元は、直接配賦および系統配賦の保管に使用されます。
- POV 次元により、年、ステータス、シナリオなど、選択した期間のモデル情報に関する具体的なビューを参照できます。ユーザーは、少なくとも1つの視点(POV)次元を定義する必要があります。また、POV バージョンを作成することもできます。
- ユーザーは、少なくとも1つのビジネス次元を定義する必要があります。ビジネス(またはユーザー定義の)次元には、製品タイプ、販売地域、製造プロセス、総勘定元帳、給与計算、部署など、ビジネスまたは組織の要件に特に関連する情報が保管されるメンバーが含まれています。

ビジネス次元には、次の次元タイプの一部または全部が含まれ、1つ以上のステージまたはモデルに適用される場合があります:

- 汎用
- 勘定科目
- エンティティ
- 時間
- 国

**注:** これらのビジネス次元は Profitability and Cost Management アプリケーションの一部として含められますが、Essbase アウトラインが配置されると、タイプのない基本または汎用次元として作成されます。

- 属性次元は、ビジネス次元に関連付けられた特殊なタイプの次元です。属性は、製品のサイズや色など、データの特性を示します。

Performance Management Architect には、2つの属性次元タイプがあります:

- 属性次元:
  - 属性は、Boolean、Date、Numeric、Text などの異なる構造を使用して作成できます。
  - 属性には階層があり、集計値にロールアップすることができます。
  - 1つのメンバーに1つの属性のみ関連付けることができます。
- ユーザー定義属性次元(UDA):
  - 属性は、テキストを使用して作成する必要があります。
  - UDA には階層がないため、合計を生成するためのレポートに容易に使用することはできません。
  - 複数の UDA を1つのメンバーに関連付けることができます。

各タイプの属性次元には、モデルの要件とレポート・ニーズに応じて異なる利点があります。属性次元の操作の詳細は、『Oracle Essbase データベース管理者ガイド』を参照してください。

- 別名はオプションです。モデルで別名を使用する場合のみ必要です。命名の規則については、付録 D「Essbase 命名規則」を参照してください。

---

**注意** 同じ次元内で重複するメンバー名または別名は使用できません。

---

**注：** モデル・ステージは次元ではないため、Performance Management Architect では使用できません。Profitability and Cost Management では、ステージはモデルに追加され、次元を論理的な手順またはステージに整理するために使用されます。

アプリケーションを作成した後、適切な手順を使用してアプリケーションに追加する次元を選択します：

- ウィザードを使用する場合は、[67 ページの「ウィザードを使用した標準 Profitability の次元の追加」](#)を参照してください。
- ブランクのアプリケーションを作成した場合は、[69 ページの「標準 Profitability 次元の手動での追加」](#)を参照してください。

## ウィザードを使用した標準 Profitability の次元の追加

アプリケーション・ウィザードを使用すると、必要なすべての次元が自動的に表示されます。正確な一致がある場合は、その次元タイプの次元列に自動的に入力されます。Profitability and Cost Management の必要な次元タイプは、自動的に分類されて、次の影付きヘッダーとともに表示されます：

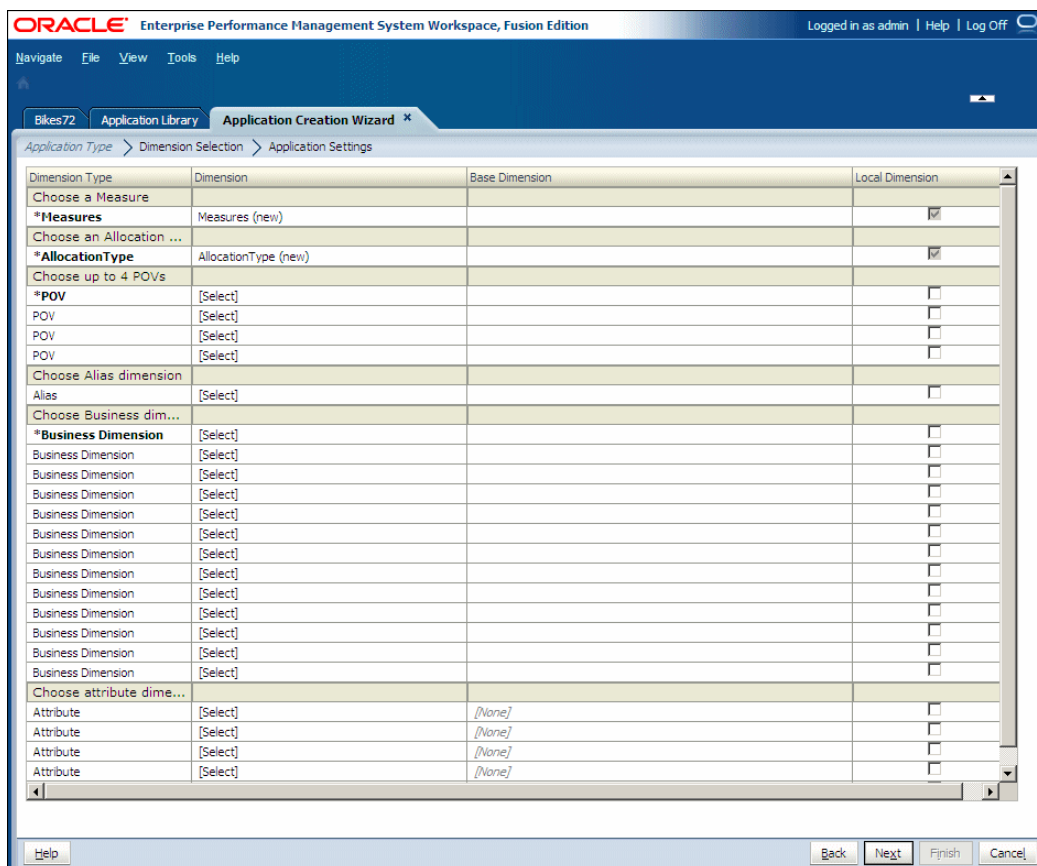
- メジャー次元
- AllocationType 次元
- POV 次元
- 別名次元(オプション)
- ビジネス次元
- 属性次元(オプション)

「ローカル次元の自動作成」を選択した場合、必要な次元ごとに新規ローカル次元が作成されます。各新規次元の名前は、次元タイプにかっこで囲まれた(New)が付いたものになります。たとえば、Account (New)となります。

勘定科目、エンティティ、時間、国などアプリケーションに含めるビジネス次元は、アプリケーションを作成する前に次元ライブラリに追加する必要があります；そうしないと、「アプリケーション・ウィザード」で次元を選択できません。名前について制限されている単語および文字を確認するには、[付録 D「Essbase 命名規則」](#)を参照してください。

▶ 次元を定義するには：

- 1 「次元の選択」タブ(アプリケーション・ウィザードの手順 2)の「次元のタイプ」で、必要な次元タイプを確認します。



- 2 既存の各次元をアプリケーションに追加するには、「次元」列の「[選択]」をクリックすると、その次元タイプに使用可能な次元のドロップダウン・リストが表示されます。

関連次元タイプの次元のみが表示されます。たとえば、次元をメジャー次元タイプにマッピングしている場合は、メジャー次元のみがリストに表示されます。

- 3 ドロップダウン・リストから次元を選択します。
- 4 オプション: 新規次元を作成します。新規次元を作成するには:
- 1 「次元」列で、「[選択]」をクリックすると、その次元タイプに使用可能な次元のドロップダウン・リストが表示されます。
  - 2 ドロップダウン・リストから、「[新規次元の作成]」を選択します。

「新規次元の追加」ダイアログ・ボックスが表示されます。

**Add New Dimension**

Dimension

Name:

Description:

Type:

Buttons: Help, OK, Cancel

3. 新規次元の「名前」と「説明」を入力します。「タイプ」は、次元タイプに基づいて自動的に選択されます。
4. 「OK」をクリックします。
5. **手順 3 と手順 4** を繰り返し、新規次元を選択または作成します。

**注：** 次の画面に移動する前に、すべての必須次元を、次元または[該当なし]にマップする必要があります。

6. **オプション:** 次元をローカル次元として指定するには、各行の最後にある「ローカル次元」チェック・ボックスを選択します。
7. 「次」をクリックします。  
「アプリケーション設定」画面が表示されます。
8. 必要に応じて、アプリケーション設定を変更します。[70 ページの「標準 Profitability アプリケーションの設定の変更」](#)を参照してください。

## 標準 Profitability 次元の手動での追加

ブランク・アプリケーションを作成する場合は、次元ライブラリを使用して次元を選択します。

次元ビューには、次の 3 つのペインがあります：

- **共有ライブラリ** - 左側に表示されます。共有ライブラリで現在使用可能なすべての次元が表示されます。
- **アプリケーション** - 中央部に表示され、展開すると現在のアプリケーションのすべての次元の名前がリスト表示されます。
- **プロパティ・グリッド** - 右側に表示されます。選択した次元またはメンバーのすべてのプロパティを表示します。

次元ライブラリの使用の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

▶ ブランク・アプリケーションに次元を追加するには：

1. 次元ライブラリで、共有ライブラリからアプリケーションに次元をドラッグします。

「次元のコピー」ダイアログ・ボックスが表示されます。


**ヒント：** 新規ローカル次元を作成するには、アプリケーション・ビューでアプリケーション名を右クリックし、「次元の作成」を選択します。「新規次元の追加」ダイアログ・ボックスが表示されます。

2. 「ローカル次元」または「共有次元」を選択します。

**ヒント：** アプリケーションにメンバーのみを追加するには、共有ライブラリで次元を右クリックして、「アプリケーション・ビューに追加」を選択します。

- 必要なすべての次元をアプリケーションに追加します。(メジャー、AllocationType、POV など。)

展開されたアプリケーションにすべての次元が表示されます。

**ヒント:** 共有ライブラリにメンバーを追加し、アプリケーションに次元を追加した場合、「リフレッシュ」ボタン  をクリックすると、アプリケーションに新規メンバーが表示されます。

- 「NoMember」の「集計プロパティ値」「」を「~」(無視)に設定します。
- 次元を順序変更して、NoMember が最後の世代 2 の位置に移動されるようにします。73 ページの「子メンバーの順序変更」を参照してください。
- 必要に応じて、アプリケーション設定を変更します。70 ページの「標準 Profitability アプリケーションの設定の変更」を参照してください。

## 標準 Profitability アプリケーションの設定の変更

### サブトピック

- [次元ライブラリを使用したアプリケーション設定の変更](#)
- [アプリケーション・ウィザードを使用したアプリケーション設定の変更](#)
- [POV 次元のプロパティの設定](#)
- [子メンバーの順序変更](#)

次元を作成して選択した後で、アプリケーションのプロパティ、関連付けおよび他のアプリケーション設定を更新します。ブランク・アプリケーションを作成し、次元ライブラリを使用してアプリケーション設定を変更する場合、変更を加えるためにウィザードではなく次元ライブラリを使用するという点を除けば、同様のオプションを使用できます。

**注:** 次元の関連付けは、共有階層に必要ですが、すべてのモデルで必要というわけではありません。

アプリケーション設定を変更する次の各手順を参照してください:

- ブランク・アプリケーションの場合は次元ライブラリを使用します。71 ページの「[次元ライブラリを使用したアプリケーション設定の変更](#)」を参照してください。
- 自動生成アプリケーションの場合は「アプリケーション・ウィザード」を使用します。71 ページの「[アプリケーション・ウィザードを使用したアプリケーション設定の変更](#)」を参照してください。

## 次元ライブラリを使用したアプリケーション設定の変更

次元ライブラリには、共有ライブラリ、アプリケーションおよびプロパティ・グリッドが表示されます。右クリック・メニューを使用して、アプリケーション次元を変更し、次のアクションを実行します:

- Profitability and Cost Management モデルの場合は、次のタスクを実行します:
  - POV 次元のプロパティ、および必要に応じて、複数の POV 次元の POV 表示順を設定します。  
[72 ページの「POV 次元のプロパティの設定」](#)を参照してください。
  - すべてのモデル次元の次元ソート順を設定します。  
[53 ページの「次元ソート順の設定」](#)を参照してください。
  - NoMember の順序を変更し、このメンバーをリスト上の最終世代 2 のメンバーとして表示します。  
[73 ページの「子メンバーの順序変更」](#)を参照してください
- 必要に応じて、Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide に記載されている手順を使用して次元を変更します:

## アプリケーション・ウィザードを使用したアプリケーション設定の変更

ウィザードを使用してアプリケーションを作成する場合、アプリケーションの次元およびメンバーが左側に表示され、プロパティ・グリッドが右側に表示され、検証および配置オプションが下部に表示されます。

右クリック・メニューを使用してアプリケーション次元を変更し、次のタスクを実行します:

- Profitability and Cost Management モデルの場合は、次のタスクを実行します:
  - POV 次元のプロパティ、および必要に応じて、複数の POV 次元の POV 表示順を設定します。  
[72 ページの「POV 次元のプロパティの設定」](#)を参照してください。
  - すべてのモデル次元の次元ソート順を設定します。  
[53 ページの「次元ソート順の設定」](#)を参照してください。
  - NoMember の順序を変更し、このメンバーをリスト上の最終世代 2 のメンバーとして表示します。  
[73 ページの「子メンバーの順序変更」](#)を参照してください
- 必要に応じて、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』に記載されている手順を使用して次元を変更します:
  - 次元別名を設定します。
  - アプリケーション・メンバーシップを表示します。
  - 次元の関連付けを表示します。



- 次元を同期化します。
- メンバーを作成します。
- 共有メンバーを挿入します。
- 孤立メンバーを管理します。

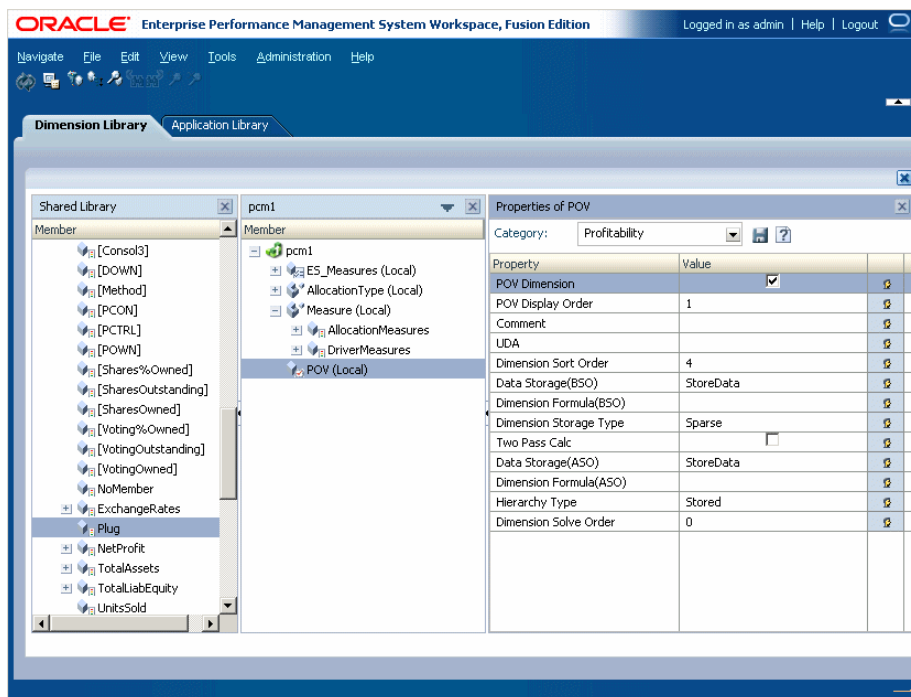
## POV 次元のプロパティの設定

Profitability and Cost Management モデルごとに、少なくとも 1 つの POV(視点)次元を設定する必要があります。POV 次元は、任意のカスタム値に設定できますが、通常は、年、期間、シナリオなどの期間を示します。また、POV バージョンを作成することもできます。

複数の POV 次元が指定されている場合、POV の表示順序も設定してこれらの次元の計算にシーケンスを付ける必要があります。Performance Management Architect で設定された POV の表示順序は、Profitability and Cost Management で自動的に取得されます。

▶ POV 次元プロパティを設定するには:

- 1 次元ライブラリで、最初の POV 次元(年など)を選択します。
- 2 プロパティ・グリッドで「カテゴリ」の下「収益性」を選択します。
- 3 「POV 次元」を選択します。



- 4 オプション: 期間やシナリオなど、他の POV 次元を選択するには、手順 1 から手順 3 を繰り返します。
- 5 オプション: 複数の POV 次元が選択された場合、POV 次元ごとに「POV の表示順序」を設定します。

「POV の表示順序」セルをダブルクリックして、選択した POV 次元のシーケンス番号を入力します。たとえば、年、期間およびシナリオが POV 次元とし



て設定されている場合、年の POV の表示順序に 1、期間に 2、およびシナリオに 3 を表示順序として設定します。

## 子メンバーの順序変更

モデル内のすべてのビジネス次元については、NoMember がメンバー・リストの最後の世代 2 メンバーとして表示されるようにリストを順序変更する必要があります; これを行わないと、モデルの検証は失敗します。

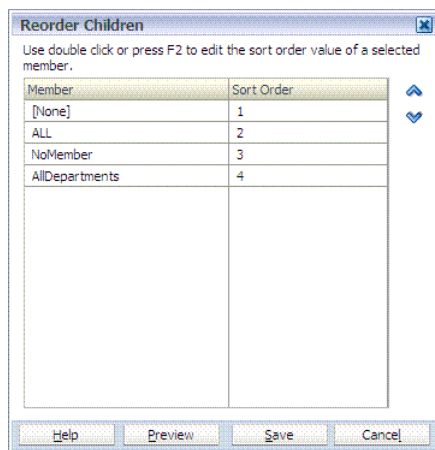
NoMember が最後の位置に残っていれば、特定の要件にあわせてメンバーの順序も変更できます。

**注：** この手順はビジネス次元のみで必要です。

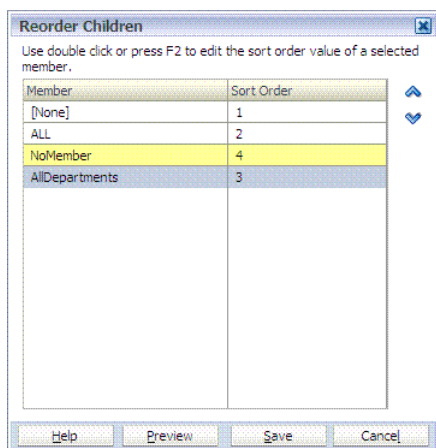
▶ 子を順序変更するには:

- 1 EPM Workspace から、「ナビゲート」、「管理」、「次元ライブラリ」の順に選択し、共有ライブラリを表示します。
- 2 次の説明に従って、記録する次元を選択します:
  - 共有次元の場合、「共有ライブラリ」の下で、順序変更する次元を右クリックして「子の順序変更」を選択します。
  - ローカル次元の場合、「アプリケーション」列の下で、順序変更する次元を右クリックして「子の順序変更」を選択します。

「子の順序変更」ダイアログ・ボックスが表示されます。



- 3 「ソート順」で、メンバーの順序を変更するセルをダブルクリックし、新しいソート順の場所の番号を入力します。



**注：** NoMember 次元が最後の世代 2 メンバーとしてリストの最後に移動したことを確認します; 最後がない場合、配置は失敗します。

- 4 手順 3 を繰り返して、すべてのメンバーを順序付けます。
- 5 変更をプレビューするには、「プレビュー」をクリックします。
- 6 「保存」をクリックします。

## Performance Management Architect でのアプリケーションの検証と配置

Performance Management Architect で Profitability and Cost Management アプリケーションを作成した後、アプリケーションを検証して Profitability and Cost Management に配置する必要があります。

次の項を参照してください:

- [74 ページの「Profitability and Cost Management の検証」](#)
- [76 ページの「ウィザードを使用したアプリケーションの検証と配置」](#)
- [77 ページの「Performance Management Architect を使用したアプリケーションの検証と配置」](#)

**注：** アプリケーションの再配置時に個別メンバー、複数のメンバーまたはメンバーの組合せの追加、名前変更、親変更または削除が可能です。

Performance Management Architect を使用して Profitability and Cost Management アプリケーションを作成および配置する方法の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

## Profitability and Cost Management の検証

Profitability and Cost Management では、次の条件が検証されます:

表 12 Profitability の検証

検証レベル	検証
アプリケーション	<ul style="list-style-type: none"> <li>● アプリケーションの名前は 7 文字以下にする必要があります、特殊文字は使用できません。</li> <li>● 少なくとも 1 つの次元を POV タイプに設定する必要があります。最大 4 つの次元を POV 次元としてマークできます。各 POV 次元クラスは 1 回のみ使用できます。</li> <li>● 少なくとも 1 つのビジネス次元が定義されている必要があります。</li> <li>● 少なくとも 1 つのメジャー次元が定義されている必要があります。</li> <li>● 少なくとも 1 つの AllocationType 次元が定義されている必要があります。</li> <li>● タイプが勘定科目の次元が 1 つしかありません。</li> <li>● タイプがエンティティの次元が 1 つしかありません。</li> <li>● アプリケーション名に Essbase の特殊文字および予約語が含まれていません。</li> </ul>
次元	<ul style="list-style-type: none"> <li>● ビジネス次元のルート・メンバーには、「LabelOnly」に設定された ASO および BSO データ・ストレージが必要です。</li> <li>● 次元ソート順が、別名次元と UDA 次元を除き、モデル内のすべての次元に設定されており、次の条件を満たしています： <ul style="list-style-type: none"> <li>○ 別名次元と UDA 次元を除く、モデル内のすべての次元に次元ソート順が設定されている必要があります。</li> </ul> <p style="margin-left: 40px;">注： 別名次元と UDA 次元は、次元ソート順では無視されます。</p> <ul style="list-style-type: none"> <li>○ 次元ソート順は連続している必要があります。</li> <li>○ メジャー次元がデフォルトで 1 に設定されています。</li> <li>○ AllocationType 次元がデフォルトで 2 に設定されています。</li> <li>○ POV 次元とビジネス次元が 3 以上に設定されています。</li> <li>○ 属性次元は最後の次元としてソートされています。</li> </ul> <p style="margin-left: 40px;">たとえば、12 個の連続した次元のうち 4 個が属性次元である場合、属性次元は 9、10、11 および 12 に設定する必要があります。</p> </li> <li>● ビジネス次元には少なくとも 2 つの世代 2 メンバーが必要です。</li> <li>● POV 次元には少なくとも 1 つのメンバーが必要です。</li> <li>● 属性次元は次の条件を満たす必要があります： <ul style="list-style-type: none"> <li>○ 属性次元のレベル 0 のメンバーのみが属性として割当て可能です。</li> <li>○ 属性は基本次元の同じレベルのメンバーにのみ割当て可能です。</li> <li>○ 属性次元は疎次元にのみ関連付けられます。</li> </ul> </li> </ul>

検証レベル	検証
メンバー	<ul style="list-style-type: none"> <li>● ASO および BS0 データ・ストレージのみを定義できます。</li> <li>● 共有メンバーは、最初の世代 2 メンバーでは許可されていません。</li> <li>● 共有メンバーは、アウトライン順序で常に対応する基本メンバーの後に表示される必要があります。</li> <li>● NoMember は、すべてのビジネス次元に対して最後の世代 2 メンバーとして設定する必要がありますが、プロパティ・グリッドで「無視(-)」に設定する必要があります。</li> </ul> <p><b>注：</b> この要件は、POV、メジャー、AllocationType、別名、UDA または属性次元に適用されません。</p> <ul style="list-style-type: none"> <li>● 次のいずれかの方法によって、共有メンバーを動的階層に置く必要があります： <ul style="list-style-type: none"> <li>○ 世代 1 メンバーが HierarchyType=dynamic です</li> <li>○ 世代 1 メンバーが HierarchyType=HierarchiesEnabled であり、共有メンバーの世代 2 の祖先が HierarchyType=dynamic です</li> </ul> </li> <li>● 次元のメンバーに重複するメンバー名または別名はありません。</li> <li>● メンバー名に Essbase の特殊文字および予約語が含まれていません。</li> </ul>

## ウィザードを使用したアプリケーションの検証と配置

アプリケーション・ウィザードを使用している場合は、アプリケーションの作成を終了する前にアプリケーションを検証して、すべてのエラーを修正します。必要に応じて、ウィザードの検証および配置オプションを使用せずに、後で配置オプションをアプリケーション・ライブラリで使用することもできます。

▶ ウィザードを使用してアプリケーションを検証および配置するには:

1 アプリケーション・ウィザードの「アプリケーション設定」画面で、「検証」をクリックします。

すべてのエラーはグリッドに表示され、エラー・タイプとエラー・メッセージが示されます。

2 **オプション:** 検証後にアプリケーションを配置するには、「終了時に配置」を選択します。

3 検証エラーが発生した場合、アプリケーションを配置する前にすべての問題を修正します。

4 「終了」をクリックします。

「終了時に配置」を選択した場合、「配置」ダイアログ・ボックスが表示されます。

**注：** 共有次元を変更した場合、その共有次元を使用するすべてのアプリケーションが影響を受けます。変更内容を適用するには、アプリケーションを再配置する必要があります。

5 アプリケーションを配置します。

# Performance Management Architect を使用したアプリケーションの検証と配置

▶ Profitability and Cost Management アプリケーションを Performance Management Architect で検証および配置するには:

- 1 EPM Workspace から、「ナビゲート」、「管理」、「アプリケーション・ライブラリ」の順に選択し、アプリケーション・ライブラリを表示します。
- 2 アプリケーション名を右クリックし、「検証」を選択して、Performance Management Architect に新たに作成した Profitability and Cost Management アプリケーションを検証します。Profitability and Cost Management の検証条件は、74 ページの「Profitability and Cost Management の検証」を参照してください。

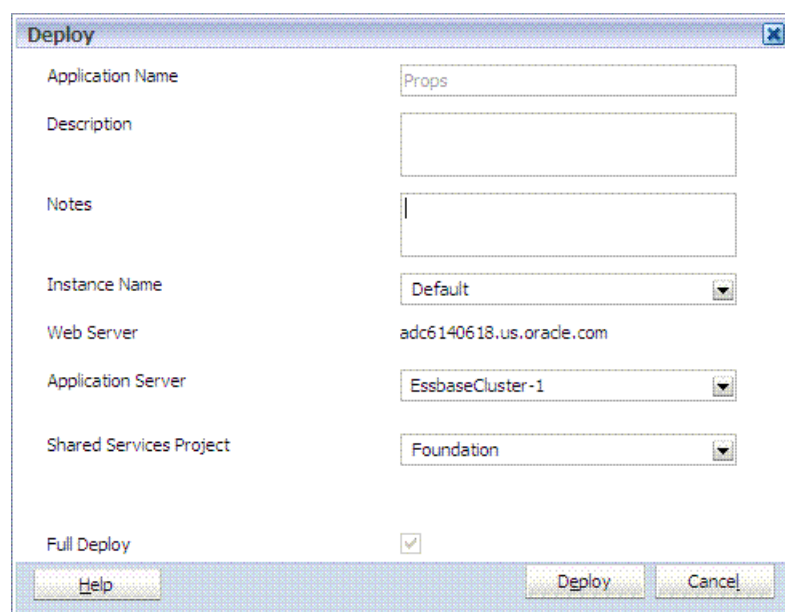
検証が完了すると、「ジョブ・タスク」ダイアログ・ボックスが表示され、関連付けられたジョブ番号が表示されます。

- 3 関連付けられたジョブのリンクをクリックすると、ジョブ・コンソールが表示されます。
- 4 ジョブ・コンソールの要約を確認します。エラーがある場合、「添付ファイル」の下にある検証ログのリンクをクリックして、すべての検証エラーまたはメッセージの完全リストを表示します。
- 5 検証ログ内のエラーをすべて修正します。
- 6 エラーがなくなり検証に成功するまで、手順 2 から手順 5 を繰り返すと、アプリケーションの配置に成功します。

これで、アプリケーションは配置可能になりました。

- 7 アプリケーション・ライブラリで、アプリケーション名を右クリックし、ドロップダウン・リストから「配置」を選択します。

アプリケーションは、配置前に検証されます。検証エラーは、すべてジョブ・コンソールに表示されます。配置前にすべての検証エラーを修正する必要があります。検証が完了すると、「配置」ダイアログ・ボックスが表示されます。



## 8 配置に必要な情報を指定します:

- **オプション:** 「ノート」で、このアプリケーションの配置に関する備考を追加します。これらの備考はジョブ・コンソールのジョブの説明に追加されます。このテキスト・フィールドには200文字まで入力でき、任意の文字を含めることができます。
- 「インスタンス名」で、アプリケーションの配置先の Profitability and Cost Management インストール名を選択します。

Profitability and Cost Management アプリケーションは、単一マシンまたはクラスタの一部として複数のマシンにインストールできます。各インストールまたはインスタンスは、「インスタンス名」ドロップダウン・リストに表示されます。初回インストール時には、コンフィグレータ・ツールで使用されるインスタンスの名前がデフォルトで指定されます。

- 「アプリケーション・サーバー」を選択します。
- 「Shared Services プロジェクト」で、このアプリケーションを割り当てる「Profitability and Cost Management」アプリケーション・グループを選択します。
- **オプション:** 次のようにして、「完全配置」を選択するかどうかを決定します:
  - アプリケーションの初めての配置の場合、Performance Management Architect はすべての次元を Profitability and Cost Management に自動的に送信するため、「完全配置」チェックボックスは無効になっています。
  - 後続の配置で「完全配置」が選択されている場合、Performance Management Architect はすべての次元を Profitability and Cost Management に送信します。
  - 「完全配置」が選択されていない場合、Performance Management Architect は変更された次元のみを Profitability and Cost Management に送信します。次元が変更されていない場合、その次元は配置に含まれません。

## 9 「配置」をクリックします。

検証ジョブが送信されたことを確認するジョブ・タスク・ウィンドウが表示され、ジョブ ID が表示されます。

## 10 「ジョブ・タスク」・ウィンドウでリンクをクリックすると、ジョブ・ステータスが表示されます。

ジョブを完了すると、新規アプリケーションが作成され、Profitability and Cost Management に配置されたことを示すメッセージが「詳細」に表示されます。新規アプリケーションを選択できるようになります。

**注:** アプリケーションの配置が正常に終了していない場合、アプリケーション・ライブラリで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クリーンアップ」を選択します。このオプションにより、Shared Services と Profitability and Cost Management の両方に対して強制的に削除コールが実行され、アプリケーションのすべてのトレースが削除されて、Performance Management Architect 内のアプリケーションが未配置の状態にリセットされます。

# データの同期

データの同期によって、Oracle アプリケーション、インタフェース・テーブルおよび外部ファイル間でデータを同期およびマップできます。

Performance Management Architect のデータの同期モジュールを使用して、宛先としての Oracle Hyperion Financial Management、Planning、Profitability and Cost Management、Essbase(ASO および BSO)、および次のソース間でデータを同期化できます:

- Oracle Hyperion Financial Management
- Planning
- Profitability and Cost Management
- Essbase(ASO および BSO)
- 外部ソース(フラット・ファイル)
- インタフェース・テーブル

データの同期は現在、BSO データベースからのデータ転送のみをサポートしています。

**注:** Essbase アプリケーションとデータベースの名前は、Profitability and Cost Management アプリケーションの初回配置後、最初に指定された名前です。BSO データベースの場合、データベース名はアプリケーション名に"C"を付加したものです。この名前ではない場合、データの同期は失敗します。

データの同期の詳細は、『Oracle Hyperion Enterprise Performance Management Architect バッチ・クライアント・ユーザー・ガイド』を参照してください。





# 6

## 標準Profitabilityのインポート・ステージング・テーブル

### この章の内容

インポート・ステージング・テーブルの使用 .....	81
標準 Profitability のインポート・データベース・テーブルの作成 .....	82
HPM_STG_STAGE.....	82
HPM_STG_POV.....	83
HPM_STG_DRIVER.....	85
HPM_STG_DRIVER_SEL.....	87
HPM_STG_DRIVER_EXCEP.....	88
HPM_STG_ASSIGNMENT.....	90
HPM_STG_ASGN_RULE_SEL.....	91

## インポート・ステージング・テーブルの使用

リレーショナル・データベースから Profitability and Cost Management にモデル・データをインポートするには、Profitability and Cost Management 製品スキーマとは別のデータベース・スキーマに一連のステージング・テーブルを作成する必要があります。(詳細 Profitability のモデル・データ・スキーマとして使用されるスキーマを再利用することもできます)。これらのステージング・テーブルに、インポートされるアプリケーション・アーティファクトの詳細を移入します。

**注意** 製品スキーマを変更しないでください。

MS SQL Server および Oracle Database では、ステージング・データベース・スクリプトを使用できます。適したスクリプトを使用して、新しいデータベースにステージング・テーブルを作成します:

この付録のスキーマ・テーブルを使用してプロシージャとステージング・テーブルを作成します:

- 82 ページの「標準 Profitability のインポート・データベース・テーブルの作成」
- 82 ページの「HPM\_STG\_STAGE」
- 83 ページの「HPM\_STG\_POV」
- 85 ページの「HPM\_STG\_DRIVER」
- 87 ページの「HPM\_STG\_DRIVER\_SEL」

- 90 ページの「HPM\_STG\_ASSIGNMENT」
- 91 ページの「HPM\_STG\_ASGN\_RULE\_SEL」

## 標準 Profitability のインポート・データベース・テーブルの作成

MS SQL Server および Oracle Database では、ステージング・データベース・スクリプトを使用できます。標準 Profitability でテーブルを作成するには、create\_staging.sql スクリプトを使用します。

▶ ステージング・テーブルを作成するには:

- 1 製品スキーマ外に、新しい Oracle データベース・スキーマまたは MS SQL Server データベース・スキーマを作成します。
- 2 データベースのタイプに応じて、デフォルトの場所で適切な SQL スクリプトの場所にナビゲートします:
  - %EPM\_ORACLE\_HOME%/products/Profitability/database/Common/MSSQLServer
  - %EPM\_ORACLE\_HOME%/products/Profitability/database/Common/Oracle
- 3 create\_staging.sql スクリプトを実行します。

## HPM\_STG\_STAGE

HPM\_STG\_STAGE テーブルは、モデル内のステージのリストを示し、各ステージの表示順、接頭辞、関連する次元およびドライバ次元を確認できます。

依存関係: なし

表 13 HPM\_STG\_STAGE

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	integer(38,0)	はい	一意のレコード ID
name	nvarchar (80)	varchar2(80 CHAR)	はい	モデル・ステージの名前。
description	nvarchar (255)	varchar2(255 CHAR)		ステージの目的の簡単な説明。
stage_order	identity (38,0)	integer(38,0)	はい	計算スクリプトの生成および計算において、モデリング(割当ておよびルール作成)時に選択したステージがモデル内で使用されるシーケンス位置。
stage_prefix	nvarchar (80)	varchar2(80 CHAR)	はい	ステージ名を区別するための接頭辞。

フィールド	SQL Server データ型	Oracle データ型	必須	説明
dim1_name	nvarchar (80)	varchar2(80 CHAR)	はい	ステージに含まれる最初の次元の名前。
dim2_name	nvarchar (80)	varchar2(80 CHAR)		ステージに含まれる 2 番目の次元の名前 (該当する場合)。
dim3_name	nvarchar (80)	varchar2(80 CHAR)		ステージに含まれる 3 番目の次元の名前 (該当する場合)。
driver_dim_name	nvarchar (80)	varchar2(80 CHAR)	はい	ステージのドライバとして選択された次元の名前。
ls_intra	nvarchar (1)	varchar2(255 CHAR)	はい	モデル・ステージ内でステージ内割当てを可能にするには、「Y」(はい)を入力します。  デフォルトは「N」(いいえ)です。このフィールドは、アプリケーションから移入されます。
last_upload_date	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻。
exception	nvarchar (255)	varchar2(255 CHAR)		このテーブルのインポート時に発生したエラーを詳述するメッセージ。
created_userid	nvarchar (32)	varchar2(32 CHAR)	更新時にインポート・プログラムによって移入されず。	最後のインポートを開始したユーザーの ID。
created_timestamp	timestamp (6)	timestamp (6)	更新時にインポート・プログラムによって移入されず。	最後のインポートの日付と時刻。この値は、更新時にインポート・プログラムによって移入されます。
modified_userid	nvarchar(32)	varchar2(32 CHAR)	更新時にインポート・プログラムによって移入されず。	最後のインポートを変更したユーザーの ID。
modified_timestamp	timestamp (6)	timestamp (6)	更新時にインポート・プログラムによって移入されず。	変更されたインポートの日付と時刻。

## HPM\_STG\_POV

HPM\_STG\_POV テーブルには、視点(POV)に含まれる次元メンバーの各組合せの状態が保管されます。

依存関係: なし

表 14 HPM\_STG\_POV

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	integer(38,0)	はい	Unique record ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	最初の POV 次元メンバーの名前。
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		2 番目の POV 次元メンバーの名前(該当する場合)。
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		3 番目の POV 次元メンバーの名前(該当する場合)。 必要に応じて、別の次元およびメンバー名を追加できます。
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		4 番目の POV 次元メンバーの名前(該当する場合)。 必要に応じて、別の次元およびメンバー名を追加できます。
pov_state	nvarchar (80)	varchar2(80 CHAR)	はい	POV の現在のステータス: <ul style="list-style-type: none"> <li>● ドラフト</li> <li>● 発行済</li> <li>● アーカイブ済</li> </ul>
last_upload_date	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻。
exception	nvarchar (255)	varchar2(255 CHAR)		このテーブルのインポート時に発生したエラーを詳述するメッセージ。
created_userid	nvarchar (32)	varchar2(32 CHAR)	更新時にインポート・プログラムによって移入されます。	最後のインポートを開始したユーザーの ID。
created_timestamp	timestamp (6)	timestamp (6)	更新時にインポート・プログラムによって移入されます。	最後のインポートの日付と時刻。この値は、更新時にインポート・プログラムによって移入されます。
modified_userid	nvarchar(32)	varchar2(32 CHAR)	更新時にインポート・プログラムによって移入されます。	最後のインポートを変更したユーザーの ID。
modified_timestamp	timestamp (6)	timestamp (6)	更新時にインポート・プログラムによって移入されます。	変更されたインポートの日付と時刻。

## HPM\_STG\_DRIVER

HPM\_STG\_DRIVER テーブルは、ドライバ・タイプ、表示順序、レイヤーおよびメジャー次元のメンバーなど、ドライバに関する詳細を示します。

依存関係: なし

表 15 HPM\_STG\_DRIVER

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	integer(38,0)	はい	一意のレコード ID
name	nvarchar (80)	varchar2(80 CHAR)	はい	ドライバの名前。
description	nvarchar (255)	varchar2(255 CHAR)		選択したドライバの目的の説明。
driver_type	nvarchar (80)	varchar2(80 CHAR)	はい	ドライバ・タイプ: <ul style="list-style-type: none"><li>● 均等</li><li>● 標準</li><li>● 標準加重</li><li>● 標準可変</li><li>● 加重可変</li><li>● 固定および可変</li><li>● パーセンテージ</li><li>● カスタム</li></ul>
fixed_member	nvarchar (80)	varchar2(80)		標準ドライバ・メジャー FixedDriverValue に対応するメジャー次元のメンバー。
fixed_location	nvarchar (30)	varchar2(30)		標準ドライバ・メジャー FixedDriverValue の場所。
rate_member	nvarchar (80)	varchar2(80)		標準ドライバ・メジャー Rate に対応するメジャー次元のメンバー。
rate_location	nvarchar (30)	varchar2(30)		標準ドライバ・メジャー Rate の場所。
weight_member	nvarchar (80)	varchar2(80)		標準ドライバ・メジャー Weight に対応するメジャー次元のメンバー。
weight_location	nvarchar (30)	varchar2(30)		標準ドライバ・メジャー Weight の場所。
volume_member	nvarchar (80)	varchar2(80)		標準ドライバ・メジャー Volume に対応するメジャー次元のメンバー。
volume_location	nvarchar (30)	varchar2(30)		標準ドライバ・メジャー Volume の場所。

フィールド	SQL Server データ型	Oracle データ 型	必須	説明
custom_ formula	nvarchar (500)	varchar2(500 CHAR)		ユーザーによって作成されたドライバの式。この式は、Essbase 計算スクリプト構文を使用して作成する必要があります。『Oracle Hyperion Profitability and Cost Management ユーザー・ガイド』を参照してください。
cost_layer	nvarchar (30)	varchar2(30 CHAR)		「はい」を入力して、ドライバ・レイヤーを「費用」に設定します。これにはモデルの費用値が含まれます。  注： cost_layer、revenue_layer、またはその両方を選択する必要があります。
revenue_ layer	nvarchar (30)	varchar2(30 CHAR)		「はい」を入力して、ドライバ・レイヤーを「収益」に設定します。これにはモデルの収益値が含まれます。  注： cost_layer、revenue_layer、またはその両方を選択する必要があります。
allow_idle_ cost	nvarchar (1)	varchar2(1 CHAR)		このドライバでアイドル費用を使用可能にするかどうかを決定する適切な値を入力します：  <ul style="list-style-type: none"> <li>このドライバでアイドル費用を使用可能にする場合、「Y」(はい)を入力します。</li> <li>このドライバでアイドル費用を使用不可能にする場合、「N」(いいえ)を入力します。これがデフォルトです。</li> </ul>
driver_basis_ type	nvarchar (80)	varchar2(80 CHAR)	はい	必要なドライバ基準を選択します：  <ul style="list-style-type: none"> <li>実績基準</li> <li>標準基準</li> </ul> 注： 標準基準ドライバは、均等または「パーセンテージ」のドライバ・タイプでは使用できません。
priority	identity	integer(38,0)	はい	ステージ内の配賦を指定した順序で実行できるようドライバの計算の優先度を入力します。  優先度が最低のドライバが最初に処理されます。デフォルトでは、優先度は 100 に設定されており、最も高い優先度は 1 です。数値は連続していなくてもかまいません。優先度が同じドライバは特定の順序で実行されるわけではありません。正の整数のみが有効です。
last_upload_ date	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻。
exception	nvarchar (255)	varchar2(255 CHAR)		このテーブルのインポート時に発生したエラーを詳述するメッセージ。
created_ userid	nvarchar (32)	varchar2(32 CHAR)	更新時に インポート・プログラムによって移入され ます。	最後のインポートを開始したユーザーの ID。

フィールド	SQL Server データ型	Oracle データ型	必須	説明
created_timestamp	timestamp (6)	timestamp (6)	更新時にインポート・プログラムによって移入されます。	最後のインポートの日付と時刻。この値は、更新時にインポート・プログラムによって移入されます。
modified_userid	nvarchar(32)	varchar2(32 CHAR)	更新時にインポート・プログラムによって移入されます。	最後のインポートを変更したユーザーの ID。
modified_timestamp	timestamp (6)	timestamp (6)	更新時にインポート・プログラムによって移入されます。	変更されたインポートの日付と時刻。

## HPM\_STG\_DRIVER\_SEL

HPM\_STG\_DRIVER\_SEL テーブルには、選択したドライバ・ルールについて、POV とソース・ステージ・ドライバ次元メンバーに関する詳細が示されます。

依存関係:

- HPM\_STG\_DRIVER
- HPM\_STG\_STAGE
- HPM\_STG\_POV

表 16 HPM\_STG\_DRIVER\_SEL

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	integer(38,0)	はい	一意のレコード ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した視点(POV)の最初の次元メンバーの名前。
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		視点(POV)の 2 番目の次元メンバーの名前(該当する場合)。
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		視点(POV)の 3 番目の次元メンバーの名前(該当する場合)。
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		視点(POV)の 4 番目の次元メンバーの名前(該当する場合)。

フィールド	SQL Server データ型	Oracle データ型	必須	説明
driver_dim_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択したドライバ次元のメンバーの名前
layer_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した視点(POV)のレイヤーの名前: <ul style="list-style-type: none"> <li>● 費用収益</li> <li>● 収益</li> </ul>
src_stage_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した視点(POV)のモデル・ステージの名前。
driver_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択したドライバ次元メンバーのドライバの名前。
last_upload_date	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻。
import_exception	nvarchar (255)	varchar2 (255 CHAR)		このテーブルのインポート時に発生したエラーを詳述するメッセージ。
created_userid	nvarchar (32)	varchar2 (32 CHAR)	更新時にインポート・プログラムによって移入されます。	最後のインポートを開始したユーザーの ID。
created_timestamp	timestamp (6)	timestamp (6)	更新時にインポート・プログラムによって移入されます。	最後のインポートが開始された日付と時刻。この値は、更新時にインポート・プログラムによって移入されます。
modified_userid	nvarchar(32)	varchar2 (32 CHAR)	更新時にインポート・プログラムによって移入されます。	最後のインポートを変更したユーザーの ID。
modified_timestamp	timestamp (6)	timestamp (6)	更新時にインポート・プログラムによって移入されます。	変更されたインポートの日付と時刻。

## HPM\_STG\_DRIVER\_EXCEP

HPM\_STG\_DRIVER\_EXCEP テーブルには、選択したドライバ例外について、POV とソース・ステージ交差に関する詳細が示されます。

表 17 HPM\_STG\_DRIVER\_EXCEP

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	integer(38,0)	はい	一意のレコード ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した視点(POV)の最初の次元メンバーの名前。



フィールド	SQL Server データ型	Oracle データ型	必須	説明
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		視点(POV)の 2 番目の次元メンバーの名前(該当する場合)。
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		視点(POV)の 3 番目の次元メンバーの名前(該当する場合)。
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		視点(POV)の 4 番目の次元メンバーの名前(該当する場合)。
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	ソース・ステージの最初の次元メンバーの名前
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		ソース・ステージの 2 番目の次元メンバーの名前(該当する場合)。
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		ソース・ステージの 3 番目の次元メンバーの名前(該当する場合)。
layer_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した視点(POV)のレイヤーの名前: <ul style="list-style-type: none"> <li>● 費用収益</li> <li>● 収益</li> </ul>
src_stage_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した視点(POV)のソース・ステージの名前。
driver_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した交差のドライバの名前。
last_upload_date	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻。
import_exception	nvarchar (255)	varchar2 (255 CHAR)		このテーブルのインポート時に発生したエラーを詳述するメッセージ。
created_userid	nvarchar (32)	varchar2 (32 CHAR)	更新時にインポート・プログラムによって移入されます。	最後のインポートを開始したユーザーの ID。
created_timestamp	timestamp (6)	timestamp (6)	更新時にインポート・プログラムによって移入されます。	最後のインポートが開始された日付と時刻。この値は、更新時にインポート・プログラムによって移入されます。
modified_userid	nvarchar(32)	varchar2 (32 CHAR)	更新時にインポート・プログラムによって移入されます。	最後のインポートを変更したユーザーの ID。
modified_timestamp	timestamp (6)	timestamp (6)	更新時にインポート・プログラムによって移入されます。	変更されたインポートの日付と時刻。

# HPM\_STG\_ASSIGNMENT

HPM\_STG\_ASSIGNMENT テーブルは、各割当てに関する詳細を示します。これには、ソース・ステージ、POV、レイヤー、ソース次元メンバー、宛先ステージおよび宛先次元メンバーが含まれます。

依存関係:

- HPM\_STG\_POV
- HPM\_STG\_STAGE
- HPM\_STG\_DRIVER

表 18 HPM\_STG\_ASSIGNMENT

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	integer(38,0)	はい	一意のレコード ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した視点(POV)の最初の次元メンバーの名前。
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		視点(POV)の 2 番目の次元メンバーの名前(該当する場合)。
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		視点(POV)の 3 番目の次元メンバーの名前(該当する場合)。
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		視点(POV)の 4 番目の次元メンバーの名前(該当する場合)。
layer_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した視点(POV)のレイヤーの名前: <ul style="list-style-type: none"><li>● 費用収益</li><li>● 収益</li></ul> 注: 「費用」または「収益」を選択する必要があります。
src_stage_name	nvarchar (80)	varchar2(80 CHAR)	はい	割当てのソース・ステージの名前。
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	ソース・ステージの最初の次元メンバーの名前。
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	はい(ソース・ステージ/宛先ステージにこの次元がある場合)	ソース・ステージの 2 番目の次元メンバーの名前(該当する場合)。
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	はい(ソース・ステージ/宛先ステージにこの次元がある場合)	ソース・ステージの 3 番目の次元メンバーの名前(該当する場合)。
dst_stage_name	nvarchar (80)	varchar2(80 CHAR)	はい	割当ての宛先ステージの名前。

フィールド	SQL Server データ型	Oracle データ型	必須	説明
dst_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	宛先ステージの最初の次元メンバーの名前。
dst_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		宛先ステージの 2 番目の次元メンバーの名前(該当する場合)。
dst_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		宛先ステージの 3 番目の次元メンバーの名前(該当する場合)。
last_upload_date	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻。
exception	nvarchar (255)	varchar2 (255 CHAR)		このテーブルのインポート時に発生したエラーを詳述するメッセージ。
created_userid	nvarchar (32)	varchar2 (32 CHAR)	更新時にインポート・プログラムによって移入されます。	最後のインポートを開始したユーザーの ID。
created_timestamp	timestamp (6)	timestamp (6)	更新時にインポート・プログラムによって移入されます。	最後のインポートが開始された日付と時刻。この値は、更新時にインポート・プログラムによって移入されます。
modified_userid	nvarchar(32)	varchar2 (32 CHAR)	更新時にインポート・プログラムによって移入されます。	最後のインポートを変更したユーザーの ID。
modified_timestamp	timestamp (6)	timestamp (6)	更新時にインポート・プログラムによって移入されます。	変更されたインポートの日付と時刻。

## HPM\_STG\_ASGN\_RULE\_SEL

HPM\_STG\_ASGN\_RULE\_SEL テーブルには、選択したステージの割当てを制御するルールのソース・ステージおよび次元メンバーに関する詳細が保管されます。

依存関係:

- HPM\_STG\_POV
- HPM\_STG\_STAGE
- HPM\_STG\_DRIVER

表 19 HPM\_STG\_ASSGN\_RULE\_SEL

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	integer(38,0)	はい	一意のレコード ID

フィールド	SQL Server データ型	Oracle データ 型	必須	説明
pov_dim1_ member_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した視点(POV)の最初の次元メンバ ーの名前。
pov_dim2_ member_name	nvarchar (80)	varchar2(80 CHAR)		視点(POV)の 2 番目の次元メンバ ーの名前(該当する場合)。
pov_dim3_ member_name	nvarchar (80)	varchar2(80 CHAR)		視点(POV)の 3 番目の次元メンバ ーの名前(該当する場合)。
pov_dim4_ member_name	nvarchar (80)	varchar2(80 CHAR)		視点(POV)の 4 番目の次元メンバ ーの名前(該当する場合)。
layer_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した視点(POV)のレイヤーの名前:  ● 費用収益 ● 収益  注: 「費用」または「収益」が必須 です。
src_stage_name	nvarchar (80)	varchar2(80 CHAR)	はい	割当てルール選択のソース・ステー ジの名前。
src_dim1_ member_name	nvarchar (80)	varchar2(80 CHAR)	はい	ソース・ステージの最初の次元メン バーの名前。
src_dim2_ member_name	nvarchar (80)	varchar2(80 CHAR)	はい(ソース・ス テージに適切な次 元がある場合)。	ソース・ステージの 2 番目の次元メン バーの名前(該当する場合)。
src_dim3_ member_name	nvarchar (80)	varchar2(80 CHAR)	はい(ソース・ス テージに適切な次 元がある場合)。	ソース・ステージの 3 番目の次元メン バーの名前(該当する場合)。
rule_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した割当てを制御するルールの名 前。このルールは、ターゲット・デー タベース内に存在する必要があります。
last_upload_ date	timestamp (6)	timestamp (6)		最後のインポートの日付。
import_ exception	nvarchar (255)	varchar2 (255 CHAR)		このテーブルのインポート時に発生し たエラーを詳述するメッセージ。
created_ timestamp	timestamp (6)	timestamp (6)	更新時にインポ ート・プログラム によって移入され ます。	最後のインポートの日付と時刻。
created_userid	nvarchar (32)	varchar2 (32 CHAR)	更新時にインポ ート・プログラム によって移入され ます。	最後のインポートを開始したユーザ ーの ID。
modified_userid	nvarchar (32)	varchar2 (32 CHAR)	更新時にインポ ート・プログラム によって移入され ます。	再インポートを開始したユーザーの ID。

フィールド	SQL Server データ型	Oracle データ 型	必須	説明
modified_ timestamp	timestamp (6)	timestamp (6)	更新時にインポート・プログラムによって移入され ます。	再インポートの日付と時刻。



# 7

## 標準Profitabilityのモデル定義 データのエクспорт

### この章の内容

HPM_EXP_STAGE .....	95
HPM_EXP_POV .....	96
HPM_EXP_DRIVER .....	97
HPM_EXP_DRIVER_SEL .....	98
HPM_EXP_DRIVER_EXCEP .....	99
HPM_EXP_ASSIGNMENT .....	100
HPM_EXP_ASGN_RULE_SEL .....	101

モデルを作成したら、データベース・ビューの出力として、モデル・アーティファクトを表示するようデータベースに問い合わせできます。

管理者は、ステージング・テーブルで使用される列をミラー化するデータベース・ビューをシステム・データベース内に作成できます。このビューは、システムに保管されるモデル・データを示します:

- ステージ(HPM\_EXP\_STAGE)
- POV(HPM\_EXP\_POV)
- ドライバ(HPM\_EXP\_DRIVER)
- ドライバ選択(HPM\_EXP\_DRIVER\_SEL)
- ドライバ例外(HPM\_EXP\_DRIVER\_EXCEP)
- 割当て(HPM\_EXP\_ASSIGNMENT)
- 割当てルール選択(HPM\_EXP\_ASGN\_RULE\_SEL)

## HPM\_EXP\_STAGE

HPM\_EXP\_STAGE ビューは、すべてのアプリケーションのモデル内で定義されたステージのリストを取得します。このビューには、ステージに対して定義されているすべてのその他の属性も表示されます。

表 20 HPM\_EXP\_STAGE

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar (80)	varchar2 (80)	アプリケーションの名前

フィールド	SQL Server データ型	Oracle データ型	説明
id	identity	integer(38,0)	Profitability and Cost Management データベースに保存されたステージ ID
name	nvarchar (80)	varchar2(80 CHAR)	モデル・ステージの名前。
description	nvarchar (255)	varchar2(255 CHAR)	ステージの目的の簡単な説明。
stage_order	integer	integer	計算スクリプトの生成および計算において、モデリング(割当ておよびルール作成)時に選択したステージがモデル内で使用されるシーケンス位置。
stage_prefix	nvarchar (80)	varchar2(80 CHAR)	ステージ名を区別するための接頭辞。
dim1_name	nvarchar (80)	varchar2(80 CHAR)	ステージに含まれる最初の次元の名前。
dim2_name	nvarchar (80)	varchar2(80 CHAR)	ステージに含まれる 2 番目の次元の名前(該当する場合)。
dim3_name	nvarchar (80)	varchar2(80 CHAR)	ステージに含まれる 3 番目の次元の名前(該当する場合)。
driver_dim_name	nvarchar (80)	varchar2(80 CHAR)	ステージのドライバとして選択された次元の名前。
ls_intra	nvarchar (1)	varchar2 (1 CHAR)	ステージ内割当てが許可されているかどうかを決定する次のフラグ: <ul style="list-style-type: none"> <li>「Y」(はい)は、モデル・ステージ内でステージ内割当てを使用できることを示します。</li> <li>「N」(いいえ)は、モデル・ステージ内でステージ内割当てを使用できないことを示します。</li> </ul>

## HPM\_EXP\_POV

HPM\_EXP\_POV ビューは、視点(POV)に含まれる次元メンバーの各組合せの状態を取得します。

表 21 HPM\_EXP\_POV

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar (80)	varchar2(80 CHAR)	アプリケーションの名前
id	identity	integer(38,0)	Profitability and Cost Management の製品データベースに保存された POV 次元メンバーのグループ ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	最初の POV 次元メンバーの名前。



フィールド	SQL Server データ型	Oracle データ型	説明
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	2 番目の POV 次元メンバーの名前(該当する場合)。
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	3 番目の POV 次元メンバーの名前(該当する場合)。 必要に応じて、別の次元およびメンバー名を追加できません。
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	4 番目の POV 次元メンバーの名前(該当する場合)。 必要に応じて、別の次元およびメンバー名を追加できません。
pov_state	nvarchar (80)	varchar2(80 CHAR)	POV の現在のステータス: <ul style="list-style-type: none"> <li>● ドラフト</li> <li>● 発行済</li> <li>● アーカイブ済</li> </ul>

## HPM\_EXP\_DRIVER

HPM\_EXP\_DRIVER ビューは、ドライバ・タイプ、式およびレイヤーを含む、ドライバに関する詳細を取得します。

表 22 HPM\_EXP\_DRIVER

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar (80)	varchar2(80 CHAR)	アプリケーションの名前
id	identity	integer	Profitability and Cost Management データベースに保存されたドライバ ID
driver_name	nvarchar (80)	varchar2(80 CHAR)	ドライバの名前。
description	nvarchar (255)	varchar2 (255 CHAR)	選択したドライバの目的の説明。
display_order	integer	integer	モデル内のすべてのドライバのリスト内のドライバの表示位置。
driver_type	nvarchar (80)	varchar2(80 CHAR)	ドライバ・タイプ: <ul style="list-style-type: none"> <li>● 均等</li> <li>● 標準</li> <li>● 標準加重</li> <li>● 標準可変</li> <li>● 加重可変</li> <li>● 固定および可変</li> <li>● パーセンテージ</li> <li>● カスタム</li> </ul>

フィールド	SQL Server データ型	Oracle データ 型	説明
complete_ formula	nvarchar (500)	varchar2 (4000 CHAR)	ユーザーによってドライバ・タイプのために作成された式、または事前定義済ドライバ・タイプの変換済の式。この式は、Essbase 計算スクリプト構文を使用して作成する必要があります。『Oracle Hyperion Profitability and Cost Management ユーザー・ガイド』を参照してください。
cost_layer	nvarchar (30)	varchar2 (3 CHAR)	ドライバ・レイヤーは、モデルの費用値が含まれる費用レイヤーに設定されます。 <b>注：</b> cost_layer、revenue_layer、またはその両方を選択する必要があります。
revenue_layer	nvarchar (30)	varchar2 (3 CHAR)	ドライバ・レイヤーは、モデルの収入値が含まれる収入レイヤーに設定されます。 <b>注：</b> cost_layer、revenue_layer、またはその両方を選択する必要があります。
allow_idle_ cost	nvarchar (1)	varchar2 (1 CHAR)	このドライバにアイドル費用が許可されているかどうかを決定するフラグ: <ul style="list-style-type: none"><li>● 「Y」 (はい)は、このドライバにアイドル費用が許可されていることを示します。</li><li>● 「N」 (いいえ)は、このドライバにアイドル費用が許可されないことを示します。</li></ul>
driver_basis_ type	nvarchar (80)	varchar2(80 CHAR)	必要なドライバ基準を選択します: <ul style="list-style-type: none"><li>● 実績基準</li><li>● 標準基準</li></ul> <b>注：</b> 標準基準ドライバは、均等または「パーセンテージ」のドライバ・タイプでは使用できません。
priority	integer	integer(38,0)	ステージ内の配賦を指定した順序で実行できるようドライバの計算の優先度を入力します。  優先度が最低のドライバが最初に処理されます。デフォルトでは、優先度は 100 に設定されており、最低(または最初)の優先度は 0 です。数値は連続していなくてもかまいません。優先度が同じドライバは特定の順序で実行されるわけではありません。正の整数のみが有効です。

## HPM\_EXP\_DRIVER\_SEL

HPM\_EXP\_DRIVER\_SEL ビューは、すべてのアプリケーションのモデル内で定義されたドライバ選択ルールに関する情報を取得します。

表 23 HPM\_EXP\_DRIVER\_SEL

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar (80)	varchar2(80 CHAR)	アプリケーションの名前
id	identity	integer(38,0)	一意のレコード ID

フィールド	SQL Server データ型	Oracle データ型	説明
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した視点(POV)の最初の次元メンバーの名前。
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	視点(POV)の 2 番目の次元メンバーの名前(該当する場合)。
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	視点(POV)の 3 番目の次元メンバーの名前(該当する場合)。
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	視点(POV)の 4 番目の次元メンバーの名前(該当する場合)。
driver_dim_member_name	nvarchar (80)	varchar2(80 CHAR)	ドライバ次元メンバーの名前
layer_name	nvarchar (80)	varchar2(80 CHAR)	選択した視点(POV)のレイヤーの名前: <ul style="list-style-type: none"> <li>● 費用収益</li> <li>● 収益</li> </ul>
src_stage_name	nvarchar (80)	varchar2(80 CHAR)	選択した視点(POV)のモデル・ステージの名前。
driver_name	nvarchar (80)	varchar2(80 CHAR)	選択したドライバ次元メンバーのドライバの名前。

## HPM\_EXP\_DRIVER\_EXCEP

HPM\_EXP\_DRIVER\_EXCEP ビューは、すべてのアプリケーションのモデル内に定義されているドライバ選択の例外を取得します。

表 24 HPM\_EXP\_DRIVER\_EXCEP

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar (80)	varchar2(80 CHAR)	アプリケーションの名前
id	identity	integer(38,0)	Profitability and Cost Management データベースに保存されたドライバ選択の例外。
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した視点(POV)の最初の次元メンバーの名前。
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	視点(POV)の 2 番目の次元メンバーの名前(該当する場合)。
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	視点(POV)の 3 番目の次元メンバーの名前(該当する場合)。
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	視点(POV)の 4 番目の次元メンバーの名前(該当する場合)。

フィールド	SQL Server データ型	Oracle データ型	説明
layer_name	nvarchar (80)	varchar2(80 CHAR)	選択した視点(POV)のレイヤーの名前: <ul style="list-style-type: none"> <li>● 費用収益</li> <li>● 収益</li> </ul>
src_stage_name	nvarchar (80)	varchar2(80 CHAR)	選択した視点(POV)のモデル・ステージの名前。
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	選択したステージの最初の次元のメンバー名。
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	選択したステージの 2 番目の次元のメンバー名(該当する場合)。
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	選択したステージの 3 番目の次元のメンバー名(該当する場合)。
driver_name	nvarchar (80)	varchar2(80 CHAR)	選択した交差のドライバの名前。

## HPM\_EXP\_ASSIGNMENT

HPM\_EXP\_ASSIGNMENT ビューは、各割当てに関する詳細を取得します。これには、ソース・ステージ、POV、レイヤー、ソース次元メンバー、宛先ステージおよび宛先次元メンバーが含まれます。

表 25 HPM\_EXP\_ASSIGNMENT

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar (80)	varchar2 (80)	アプリケーションの名前
id	identity	integer(38,0)	Profitability and Cost Management データベースに保存された割当て ID
pov_dim1_member_name	nvarchar (80)	varchar2 (80)	選択した視点(POV)の最初の次元メンバーの名前。
pov_dim2_member_name	nvarchar (80)	varchar2 (80)	視点(POV)の 2 番目の次元メンバーの名前(該当する場合)。
pov_dim3_member_name	nvarchar (80)	varchar2 (80)	視点(POV)の 3 番目の次元メンバーの名前(該当する場合)。
pov_dim4_member_name	nvarchar (80)	varchar2 (80)	視点(POV)の 4 番目の次元メンバーの名前(該当する場合)。
layer_name	nvarchar (80)	varchar2 (80)	選択した視点(POV)のレイヤーの名前: <ul style="list-style-type: none"> <li>● 費用</li> <li>● 収益</li> </ul>
src_stage_name	nvarchar (80)	varchar2 (80)	割当てのソース・ステージの名前。

フィールド	SQL Server データ型	Oracle データ 型	説明
src_dim1_member_name	nvarchar (80)	varchar2 (80)	ソース・ステージの最初の次元メンバーの名前。
src_dim2_member_name	nvarchar (80)	varchar2 (80)	ソース・ステージの 2 番目の次元メンバーの名前(該当する場合)。
src_dim3_member_name	nvarchar (80)	varchar2 (80)	ソース・ステージの 3 番目の次元メンバーの名前(該当する場合)。
dst_stage_name	nvarchar (80)	varchar2 (80)	割当ての宛先ステージの名前。
dst_dim1_member_name	nvarchar (80)	varchar2 (80)	宛先ステージの最初の次元メンバーの名前。
dst_dim2_member_name	nvarchar (80)	varchar2 (80)	宛先ステージの 2 番目の次元メンバーの名前(該当する場合)。
dst_dim3_member_name	nvarchar (80)	varchar2 (80)	宛先ステージの 3 番目の次元メンバーの名前(該当する場合)。

## HPM\_EXP\_ASGN\_RULE\_SEL

HPM\_EXP\_ASGN\_RULE\_SEL ビューは、選択したステージの割当てを制御するルールのソース・ステージおよび次元メンバーに関する詳細を取得します。

表 26 HPM\_EXP\_ASGN\_RULE\_SEL

フィールド	SQL Server データ型	Oracle データ 型	説明
application_name	nvarchar (80)	varchar2 (80)	アプリケーションの名前
id	identity	integer(38,0)	一意のレコード ID
pov_dim1_member_name	nvarchar (80)	varchar2 (80)	選択した視点(POV)の最初の次元メンバーの名前。
pov_dim2_member_name	nvarchar (80)	varchar2 (80)	視点(POV)の 2 番目の次元メンバーの名前(該当する場合)。
pov_dim3_member_name	nvarchar (80)	varchar2 (80)	視点(POV)の 3 番目の次元メンバーの名前(該当する場合)。
pov_dim4_member_name	nvarchar (80)	varchar2(80)	視点(POV)の 4 番目の次元メンバーの名前(該当する場合)。
layer_name	nvarchar (80)	varchar2(80)	選択した視点(POV)のレイヤーの名前: <ul style="list-style-type: none"> <li>● 費用</li> <li>● 収益</li> </ul>
src_stage_name	nvarchar (80)	varchar2(80)	割当てルールのソース・ステージの名前。

フィールド	SQL Server データ型	Oracle データ 型	説明
src_dim1_member_ name	nvarchar (80)	varchar2(80)	ソース・ステージの最初の次元メンバーの名前。
src_dim2_member_ name	nvarchar (80)	varchar2(80)	ソース・ステージの 2 番目の次元メンバーの名前(該当 する場合)。
src_dim3_member_ name	nvarchar (80)	varchar2(80)	ソース・ステージの 3 番目の次元メンバーの名前(該当 する場合)。
rule_name	nvarchar (80)	varchar2(80)	選択した割当てのソース組合せを制御する割当てルール の名前。Profitability and Cost Management データベースに割 当てルールがあらかじめ定義されている必要があります。

---

## 第 III 部

# 詳細 Profitability の操作

---

詳細 Profitability の操作の内容 :

- 詳細 Profitability の製品アーキテクチャの理解
- 詳細 Profitability データベースの操作
- 製品スキーマ
- モデル・データ・スキーマ
- 詳細 Profitability and Cost Management の次元とメタデータの操作
- Performance Management Architect を使用した詳細 Profitability アプリケーションおよび次元の管理
- 詳細 Profitability の計算
- 詳細 Profitability のインポート・ステージング・テーブル
- 詳細 Profitability のモデル定義データのエクスポート



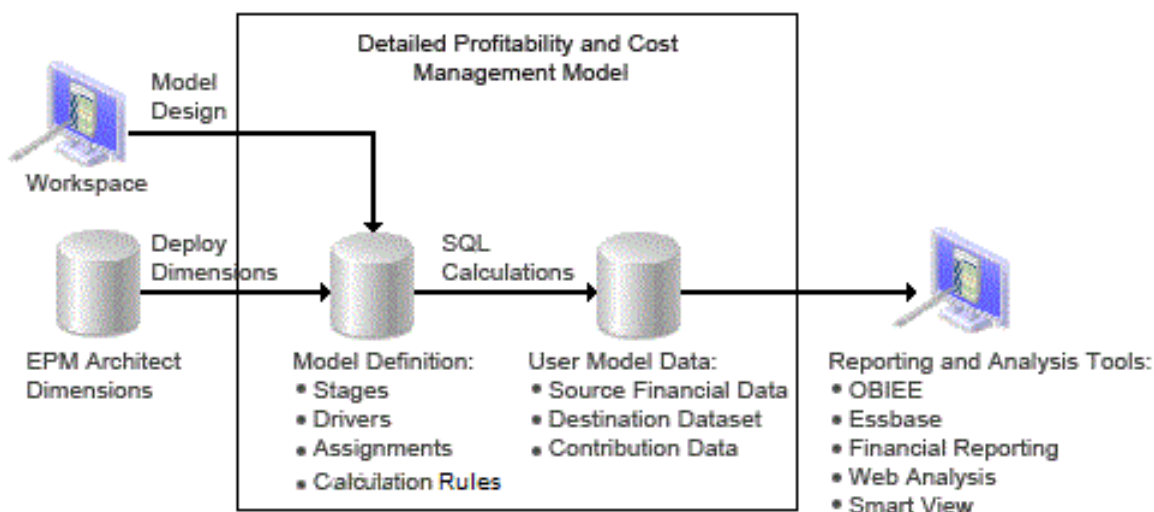


# 8

## 詳細Profitabilityの製品アーキテクチャの理解

詳細 Profitability は、EPM Workspace からアクセスされ、リレーショナル・データベースのデータを使用する分析アプリケーションです。ビジネス・ユーザーはこのアプリケーションを使用して、Profitability and Cost Management のためにビジネスをモデル化することができます。そのモデル情報を使用して、モデル情報、既存データ、およびアプリケーションによる計算結果が格納されるリレーショナル・データベースを作成することができます。詳細 Profitability モデルで入力および計算されるデータはどちらも、リレーショナル・データベースに格納されます。

Profitability and Cost Management では、アプリケーションのメタデータとセキュリティを一元的に管理するために Performance Management Architect と Shared Services を使用します。



アプリケーション管理者は、Performance Management Architect を使用して Profitability and Cost Management 次元を作成します。ユーザー・アクセスは、Shared Services を使用して一元的に管理されます。次元メタデータは、準備が整うと、Profitability and Cost Management アプリケーションまたはモデルに配置されます。Performance Management Architect の次元は、複数のモデルが共有できます。

モデル設計には、データベース内で計算を実行する際に必要な SQL 文を生成するために必要な情報が含まれます。各モデルは、データベース内の次のスキーマにアクセスできる必要があります。

- Performance Management Architect から配置された次元メタデータなどのモデル設計が格納されている、「製品スキーマ」と呼ばれるスキーマ。第 10 章「製品スキーマ」を参照してください。

- 既存のデータとモデル計算の結果が格納される、モデル・データ・スキーマと呼ばれるスキーマ。第 11 章「モデル・データ・スキーマ」を参照してください。

# 9

## 詳細 Profitability データベース の操作

詳細 Profitability アプリケーションの場合、次元データとモデル定義は、標準 Profitability アプリケーションの次元データとモデル定義を格納するのと同じリレーショナル・データベースのスキーマに格納されます。このスキーマは製品スキーマと呼ばれ、Profitability and Cost Management のインストール時に作成されます。次元データは、Performance Management Architect からアプリケーションを配置するとき、製品スキーマに移入されます。モデル定義は、モデルを構築する際にこのスキーマに格納されます。

詳細 Profitability アプリケーションの場合、配賦が実行される対象のビジネス・データも、リレーショナル・データベースに格納されます(標準 Profitability アプリケーションのように、Essbase には格納されない)。このデータが配置されるのは、モデル・データ・スキーマという別のデータベース・スキーマです。モデル・データ・スキーマはユーザー定義であり、製品スキーマと同じデータベース・インスタンスにある必要があります。Oracle と MS SQL Server データベースのみがサポートされます。

Performance Management Architect からアプリケーションを配置した後で、次元およびメンバーが列にマッピングされ、Profitability and Cost Management のアプリケーション・テーブルに登録されます。詳細 Profitability の次元の詳細は、[第 12 章「詳細 Profitability and Cost Management の次元とメタデータの操作」](#)を参照してください。

表 27 詳細 Profitability and Cost Management データベース・スキーマ

スキーマ・タイプ	保管される情報のタイプ
モデル・データ・スキーマ	<ul style="list-style-type: none"><li>● 次元階層</li><li>● 費用データ</li><li>● 収益データ</li><li>● ドライバ・データ</li><li>● 収益オブジェクト・データ</li><li>● 通貨レート</li></ul>

スキーマ・タイプ	保管される情報のタイプ
製品スキーマ	<ul style="list-style-type: none"><li>● 次元</li><li>● ステージ定義</li><li>● POV 定義</li><li>● ドライバ定義</li><li>● ドライバ選択</li><li>● 割当てルール</li><li>● 割当てルールの選択</li><li>● モデル・プリファレンス</li><li>● テーブル登録</li><li>● 計算ルール</li></ul>

標準 Profitability および詳細 Profitability アプリケーションは、Profitability and Cost Management のシステム・テーブルを保持し、インストール中に作成される同じ製品スキーマを共有します。Profitability and Cost Management の UI と詳細 Profitability 計算エンジンがこのスキーマに接続します。レポート・ビューは、モデルの開発およびモデルの計算中に製品スキーマに作成されます。

このスキーマへのアクセスは通常、アプリケーション管理者とデータベース管理者に制限されます。レポート・ビューの権限がモデル・データ・スキーマに自動的に付与され、そのシノニムがモデル・データ・スキーマに自動的に作成されます。シノニムは、その製品スキーマに対する権限を付与することなく、そのスキーマにアクセスするユーザーに権限を公開します。

インストール後、詳細 Profitability では次のシステム権限(またはそれを含む役割)を製品スキーマに付与する必要があります。

- Oracle データベースの場合、詳細 Profitability では次のシステム権限(またはそれを含む役割)を製品スキーマに付与する必要があります。
  - CREATE TABLE
  - CREATE VIEW
  - CREATE ANY SYNONYM
  - DROP ANY SYNONYM
- MSSQL Server の場合、詳細 Profitability では次のシステム権限(またはそれを含む役割)を製品スキーマに付与する必要があります。
  - CREATE TABLE
  - CREATE VIEW
  - 製品スキーマに対するモデル・データ・スキーマの ALTER または CONTROL(製品スキーマがモデル・データ・スキーマのオブジェクトを変更できる)

「例外:」製品スキーマが管理者ユーザー dbo または sa である場合、これらの権限がすでにあります。この場合、次の付与を実行する必要はありません。

**注:** HPCM 製品スキーマに関連付けられているユーザーに SYSADMIN 役割を付与しないでください。これにより、そのユーザーのデフォルト・スキーマが変更されますが、デフォルト・スキーマは HPCM 製品スキーマに設定しておく必要があります。

CREATE TABLE 権限と CREATE VIEW 権限を付与すると、Profitability and Cost Management で、モデルの開発中およびモデルの計算中にシステム生成のテーブルとレポート・ビューを作成できるようになります。

CREATE ANY SYNONYM 権限と DROP ANY SYNONYM 権限を付与すると、Profitability and Cost Management で、製品スキーマに作成されるシステム生成のレポート・ビューに対するモデル・データ・スキーマのシノニムを管理できるようになります。

**注：** これらのシステム権限を付与することが、組織でセキュリティ上問題になる場合は、これらの権限を留保し、システム生成のレポート・ビューに対するモデル・データ・スキーマに、手動でシノニムを作成することも可能です。このオプションを使用すると、モデル・データ・スキーマのみにアクセスするビジネス・データ・ユーザーにそれらを公開できます。

## この章の内容

製品スキーマへのモデル・データ・スキーマ・テーブル権限の付与.....	111
モデル・データ・スキーマのテーブル構造要件.....	112
データベース・ビューの登録.....	112

モデル・データ・スキーマは、既存のスキーマの場合も、詳細 Profitability アプリケーションをサポートするためにデータベース管理者によって作成されたスキーマの場合もあります。このスキーマには、詳細 Profitability アプリケーションが配賦を実行する対象のビジネス・データを含むテーブルおよびビューが保持されています。Profitability and Cost Management は製品スキーマにのみ直接接続します。アプリケーションでテーブルまたはビューを表示するためには、データベース管理者が製品スキーマにそれぞれのテーブル権限を付与する必要があります。

## 製品スキーマへのモデル・データ・スキーマ・テーブル権限の付与

「モデル・データの登録」でモデル・データ・スキーマのデータベース・オブジェクトを表示するには、まずテーブル権限を Profitability and Cost Management 製品スキーマに付与する必要があります。次の表のように各テーブル・タイプの権限を付与します:

テーブル・タイプ	権限
ソース・ステージ・テーブル(水平または垂直)	GRANT SELECT ON <table_name or view_name> TO <hpm_product_schema> WITH GRANT OPTION
宛先ステージ・テーブル	GRANT SELECT, UPDATE ON <table_name or view_name> TO <hpm_product_schema>
すべてのタイプの参照テーブル	GRANT SELECT ON <table_name or view_name> TO <hpm_product_schema>

**注：** Oracle データベースの場合は dbms\_stats の EXECUTE 権限を付与する必要があります。デフォルトでは、この権限はパブリックに付与されます。企業において統計の収集が制限されており、この権限がパブリックから取り消されている場合は、この権限を製品スキーマに割り当てる必要があります。

Performance Management Architect からアプリケーションを配置した後、詳細 Profitability アプリケーションで使用するモデル・データ・スキーマにテーブルを登録する必要があります。

Performance Management Architect で定義した次元とメジャーにその列をマッピングします。詳細 Profitability の次元の詳細は、[第 12 章「詳細 Profitability and Cost Management の次元とメタデータの操作」](#)を参照してください。

## モデル・データ・スキーマのテーブル構造要件

詳細 Profitability アプリケーションで使用するためにモデル・データ・スキーマのテーブルを登録するには、次の要件を満たす必要があります：

- **データベース・オブジェクト名:** 大文字であることが必要です。数字 0-9 および記号「\_」と「\$」を使用できます。小文字および他の記号はサポートされません。小文字や他の記号を含む既存のデータベース・オブジェクト名がある場合は、テーブル、ビューおよび列の名前を変更するために、許可された識別子を使用してビューを作成します。基礎となるテーブルのかわりにビューを登録することができます。
- **宛先ステージ・テーブルのみ:** 「WORKING」(大文字)という数値列を含む必要があります。
- **宛先ステージ・テーブルのみ:** テーブルの一意性を強制するために主キー制約が定義されている必要があります。

## データベース・ビューの登録

製品スキーマに適切な権限が付与されている場合には、モデル・データ・スキーマからデータベース・ビューを登録することもできます。任意の有効なデータベース・ビューを、ステージ・テーブルまたは参照テーブルとして登録できます。宛先ステージ・テーブルとして登録できるのは、単純な単一テーブルの更新可能なビューのみです。

データベース・ビューは次のような理由で役立ちます。

- モデル・データ・スキーマに格納されていないテーブルにアクセスする場合。これは特に参照テーブルで役に立ちます。
- 詳細 Profitability アプリケーションに公開されないように、基礎となるテーブルで行をフィルタ処理する、または列を非表示にする場合。
- 読取り専用テーブルのビュー定義で複雑な結合を非表示にし、アプリケーションでの使用を単純にする場合。

モデルの開発およびモデルの計算中に、Profitability and Cost Management はシステム生成のレポート・ビューを製品スキーマに作成します。Profitability and Cost Management はモデル・データ・スキーマに対する SELECT 権限も付与し、モデル・データ・スキーマにそのシノニムを作成しようとします。これにより、通常は管理者のみのアクセス用に予約されている製品スキーマに対する権限がないユーザーでも、スキーマにアクセスできるようになります。モデルの開発者とレポー



ト開発者に、このスキーマに対するアクセス権が付与されるのは、Profitability and Cost Management で必要とされるすべてのテーブル登録タスクと、使用するレポート・ツールでの物理モデルの設定を管理者が実行する予定ではない場合のみです。

**注：** Oracle ではデータベース・ユーザーとデータベース・スキーマが同じものですが、MS SQL Server ではそれぞれが別個です。MS SQL Server を使用する場合、モデル・データ・スキーマと同じ名前のユーザーを作成し、モデル・データ・スキーマへのアクセス権をそのユーザーに付与する必要があります。Profitability and Cost Management によって付与されたシステム生成ビューに対する権限は、モデル・データ・ユーザーを参照します。作成されたシノニムはモデル・データ・スキーマに追加されます。

SQL Server の場合、Profitability and Cost Management 製品およびドキュメントでのモデル・データ・スキーマの参照は、スキーマまたはユーザー(あるいは両方)を指します。

**注：** SQL Server: 宛先ステージ・テーブルと参照テーブルに、丸めエラーを防ぐために小数点以下が 4 桁以上であるメジャー列を作成する必要があります。

---

**注意** データまたはアーティファクトをインポートする前にモデル・データ・スキーマのバックアップを作成する必要があります。Oracle Enterprise Performance Management System Backup and Recovery Guide を参照してください。

---

詳細は、Oracle Hyperion Profitability and Cost Management User's Guide を参照してください。



# 12

## 詳細 Profitability and Cost Management の次元とメタデータの操作

### この章の内容

詳細 Profitability の次元とメタデータについて.....	115
詳細 Profitability の次元タイプ.....	117
別名次元.....	117
ビジネス次元.....	118
POV 次元.....	119
属性次元.....	120
次元ソート順および密度の設定.....	120

### 詳細 Profitability の次元とメタデータについて

詳細 Profitability では、詳細 Profitability アプリケーションの開始点として既存のリージョンナル・データベースを使用できます。

次元およびメタデータは、Profitability and Cost Management 管理者によって Performance Management Architect を使用して作成および管理されます。Profitability and Cost Management では、一般的な次元およびメンバーを使用することにより、Planning などの他の製品で一般的なデータを使用および転送し、時間と労力を削減し、正確さを増すことができます。次元とメタデータの作成および管理の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

Profitability and Cost Management では、Performance Management Architect で作成された次元およびメンバーを使用して、ビジネス・モデルの様々な構造要素を表します。詳細 Profitability アプリケーションの作成時に、次の次元が選択されます。

- MeasuresDetailed (必須)
- 少なくとも 1 つの Performance Management Architect POV 次元(必須)
- 少なくとも 1 つのビジネス次元(必須)
- 別名次元(オプション)
- 属性次元(オプション)

次元ごとに、次元タイプと次元名の両方を指定する必要があります:

- 次元タイプは、選択した次元に定義された機能を使用可能にする次元のプロパティです。41 ページの「次元タイプ」を参照してください。

- 次元名は、組織およびビジネスに関して次元のコンテンツを識別するために割り当てられます。たとえば、「勘定科目タイプ」の次元には、「一般会計」や「勘定科目表」などの次元名を付ける場合があります。次元名に次元タイプを反映する必要はありませんが、反映することもできます。命名の制限については、[付録 D「Essbase 命名規則」](#)を参照してください。

Performance Management Architect を使用することにより、次元はアプリケーション内に共有またはローカルとして存在できます:

- 共有次元は、Performance Management Architect の共有ライブラリ内に存在し、複数の製品およびアプリケーションが使用できます。
- ローカル次元は、Profitability and Cost Management タイプのアプリケーションなどの 1 つのアプリケーション・インスタンス内にのみ存在する、分離された独立次元です。これらの次元は、これらが作成されているアプリケーション内でのみ使用され、同じタイプの別のアプリケーションであっても表示または使用できません。
- MeasuresDetailed は、貸借一致などの検証アクティビティに必要な配賦メンバーが含まれている予約済の次元です。この次元には、ドライバ・メジャーが含まれていません。
- 次元。メジャー次元などのモデルおよびシナリオの構造を実現します。
- ビジネス次元。部署、一般会計の勘定科目、アクティビティ、顧客、製品などの、ビジネス固有のモデル要素を表します
- 視点(POV)次元。期間、シナリオおよびバージョンなど
- 別名次元(必要な場合)
- 属性次元(必要な場合)

Performance Management Architect を介して、別の製品内に存在する次元およびメンバーを選択したり、そのモデル専用の新しい次元およびメンバーを作成します。詳細 Profitability and Cost Management モデルに対して次元およびメンバーが選択された後、これらは次元ライブラリで自動的に使用可能になります。配置後は、Profitability and Cost Management アプリケーションで次元およびメンバーを使用できます。

Performance Management Architect には、システム生成とユーザー定義の両方の次元およびメンバーが存在する必要があります。次元には、英数字や計算値が含まれる場合があります。

---

**注意** 作成できる次元およびメンバーの数に制限はありませんが、次元構造が大きくなると、パフォーマンス上の問題が発生します。アプリケーションの配置後に追加の次元またはメンバーを作成する場合、新しいオブジェクトを使用するにはモデル・データ登録の再登録が必要になる場合があります。

---

次元とメンバーの作成および管理の手順は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

## 詳細 Profitability の次元タイプ

詳細 Profitability and Cost Management では、次のタイプの次元を使用できます:

- MeasuresDetailed は、貸借一致などの検証アクティビティに必要な配賦メンバーが含まれている制限付きの次元です。
- 別名次元は、次元の定義に役立つ代替の名前、説明、言語またはその他のアイテムを割り当てるために使用されます。これはオプションの次元であり、モデルで別名を使用する場合のみ必要です。

**注:** 重複メンバー名または別名は使用できません。

- ビジネス次元は、部署、総勘定元帳の勘定科目、アクティビティ、顧客または製品などのモデル内のビジネス要素を説明するために作成されます。アプリケーションに少なくとも1つのビジネス次元が必要です。
- 属性次元を使用すると、次元メンバーの属性または品質に基づく分析が可能になります。属性は、製品のサイズや色など、データの特性を示します。
- 一部の Performance Management Architect の次元タイプは、Profitability and Cost Management のモデルで使用できます:
  - 勘定科目
  - エンティティ
  - 国
  - 通貨

これらの次元タイプの使用の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

**注:** 次元アウトラインを定義する場合、命名用として使用できない制限文字があります。『Oracle Essbase データベース管理者ガイド』の Essbase 命名規則に関する項を参照することをお勧めします。

## 別名次元

別名は、次元の定義に役立つ代替の名前、説明、言語またはその他のアイテムです。たとえば、システム内で顧客番号を参照できますが、画面上に顧客名を表示する別名を割り当てることにより、この顧客を簡単に識別できるようになります。1つ以上の別名を勘定科目、通貨、エンティティ、シナリオ、期間、バージョン、年およびユーザー定義の次元メンバーに割り当てることができます。

**注:** 同じ次元内で重複するメンバー名または別名は使用できません。

Profitability and Cost Management では、別名を Performance Management Architect に設定する必要があります。別名次元の作成の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

---

**注意** アプリケーションが Profitability and Cost Management に再配置されるまでは、別名の関連付けを Performance Management Architect で削除しても、モデルからは削除されません。

---


インストール後、Default 別名テーブルが使用可能になります。再配置後は、ドライバの選択、割当て、データ入力およびトレーサビリティを含む共通メンバー・セクタを使用するすべての画面で別名を表示できます。別名を検索およびフィルタ処理するには、先にコンテキスト・メニューで「別名の表示」を選択します。

**注：** 別名ビューは「ドライバの選択」セクタには含まれません(このセクタには、ドライバの選択を追加または変更するときに、「ドライバ定義」画面または「ドライバの選択」画面からアクセスできます)。

特定の次元が Essbase で複製されている場合、別名が複製される場合があります。

▶ 別名を表示するには:

1 アプリケーションから、共通メンバー・セクタを使用する任意の画面(「ドライバの選択」、「割当て」、「データの入力」、「配賦のトレース」など)を選択します。共通メンバー・セクタを含む該当する画面が表示され、使用可能なすべてのメンバーが示されます。

2 共通メンバー・セクタで、「コンテキスト・メニュー」ボタン  をクリックし、「別名の表示」を選択します。

別名がメンバー・リストに表示されます。

**注：** 「コンテキスト・メニュー」から「別名の表示」を選択したときに、別名が割り当てられていない場合、メンバー名が大かっこで囲まれて表示されます。たとえば、メンバー名 Product は、[Product] と表示されません。

## ビジネス次元

ビジネス次元は、製品、顧客、地域、従業員など、モデルの各ステージ内のビジネス固有のオブジェクトを示します。これらの次元およびメンバーは、Performance Management Architect で作成されます。

ビジネス次元は、次の次元タイプの一部または全部を使用し、1つ以上のステージまたはモデルに適用される場合があります:

- 汎用
- 勘定科目
- エンティティ
- 国

Essbase アウトラインが配置されると、Profitability and Cost Management アプリケーションでビジネス次元がタイプのない基本次元または汎用次元として作成されます。この機能を使用すると、Profitability and Cost Management で、Oracle Hyperion Planning などの他のアプリケーションで定義されている次元メンバーおよび階層を再使用できます。

**注：** この次元タイプは、集約ストレージ・アウトラインには適用されません。

ビジネス次元を作成する場合、次の要件が適用されます：

- 次元の Gen1 メンバーの次のプロパティは LABEL\_ONLY に設定する必要があります：
  - DataStorage(BSO)
  - DataStorage(ASO)
- 通常、Gen1 次元名の下にある最初の世代 2 の子は、ALL メンバーに設定されます。たとえば、Departments 次元の場合は AllDepartments になります。  
プライマリ階層は、最初の世代 2 の子の下でホストされます。割当モデリングでは最初の世代 2 階層のみが使用され、この階層に共有メンバーを含めることはできません。
- 追加の世代 2 メンバーは代替階層をホストできますが、これらの階層は割当モデリングでは使用されません。次元が代替階層をホストする場合は、次元の階層タイプを「使用可能」に設定し、最初の世代 2 メンバーの階層タイプを「保管済」、共有メンバーを含む代替階層の世代 2 メンバーを「動的」に設定します。  
これらの代替階層は、Profitability and Cost Management モデリング画面には表示されず、Essbase でのみ表示できます。
- NoMember メンバーが必要です。階層内の最後の世代 2 の子は常に NoMember であるとともに、集計が IGNORE (~) に設定されている必要があります。

**注：** NoMember メンバーは、Performance Management Architect によって自動的に追加されるため、ADS ファイルまたはフラット・ファイルに作成する必要はありません。このメンバーは削除しないでください。

## POV 次元

POV 次元は、年、シナリオまたは期間など、モデルに固有の視点またはバージョンを示します。次元は、組織の要件を反映するようにカスタマイズできます。たとえば、四半期、月、季節ごとのグループなどで POV を構成できます。

各モデルには 1 つ以上の POV 次元が必要で、最大 4 つの POV 次元を作成できます。POV 次元は、モデルの Performance Management Architect プロパティ・グリッドで設定します。

バージョン次元も使用可能で、モデルの異なるインスタンスの作成に使用されます。このバージョンは、仮定シナリオを実行して、戦略やビジネス・オプション



を試行できるよう変更可能です。バージョンを変更することで、機能を実装し、最善の方策を決定するために結果を比較できます。

## 属性次元

属性次元は、ビジネス次元に関連付けられた次元の特殊なタイプで、関連付けられたその他の次元のメンバーを分類する際に使用可能なメンバーが含まれます。属性次元は、製品のサイズ、色など、データの特性を示します。

これらの属性を使用して、次元メンバーの属性または品質に基づいてデータを分析できます。属性次元は、割当てルールの作成時に宛先交差のフィルタ処理にも使用されます。

**注：** 命名の制限については、[付録 D 「Essbase 命名規則」](#) を参照してください。

2つの属性次元タイプが Performance Management Architect に存在します：

- 属性次元:
  - ブール、日付、数値、テキストなど、異なる構造を使用して属性を作成できます。
  - 属性には階層があり、集計値にロールアップすることができます。
  - 所定の属性次元の1つの属性のみを1つのメンバーに関連付けできます。
- ユーザ定義属性次元(UDA):
  - 属性は、テキストを使用して作成する必要があります。
  - UDA には階層がないため、合計を生成するためのレポートに容易に使用することはできません。
  - 複数の UDA を1つのメンバーに関連付けることができます。

モデルやレポートの要件により、属性次元の利点はタイプごとに異なります。属性次元の使用の詳細は、『Oracle Essbase データベース管理者ガイド』を参照してください。

これらの属性を使用して、次元メンバーの属性または品質に基づいてデータを分析できます。属性次元は、割当てルールの作成時に宛先交差のフィルタ処理にも使用されます。

命名の制限については、[付録 D 「Essbase 命名規則」](#) を参照してください。

## 次元ソート順および密度の設定

Dimension Sort Order プロパティは、Profitability and Cost Management によって生成される Essbase アウトラインにおける次元の順序を制御します。次元ソート順は、別名または UDA を除く、モデル内のすべての次元に対して設定する必要があります。

---

**注意** 次元のソート順を空のままにすると、検証は失敗します。

---



Dimension Sort Order プロパティは、Performance Management Architect で設定され、Essbase アウトラインの生成に使用するための配置時に Profitability and Cost Management に渡されます。詳細は、53 ページの「次元ソート順の設定」を参照してください。

モデルの次元ソート順設定は、Performance Management Architect で検証されます。70 ページの「標準 Profitability アプリケーションの設定の変更」を参照してください。

「次元のソート順設定」ダイアログ・ボックスから、次元の密度を設定することもできます。

## 次元ソート順の推奨事項

次の推奨事項を参照して、次元ソートを設定することをお勧めします:

- 別名と UDA を除く、モデル内のすべての次元ソート順が設定されている。  
**注:** 別名次元と UDA 次元は、Profitability and Cost Management と Essbase には次元として存在しないため、次元ソート順では無視されている。
- 次元ソート順が連続しており、一意かつ 1 以上である必要があります。
- メジャー次元がデフォルトで 1 に設定されている。
- AllocationType 次元がデフォルトで 2 に設定されている。
- ビジネス次元と POV 次元が 3 以上に設定されている。
- 属性次元が常に最後の次元としてソートされている。たとえば、12 個の連続した次元のうち 4 個が属性次元である場合、属性次元は 9、10、11 および 12 に設定する必要があります。

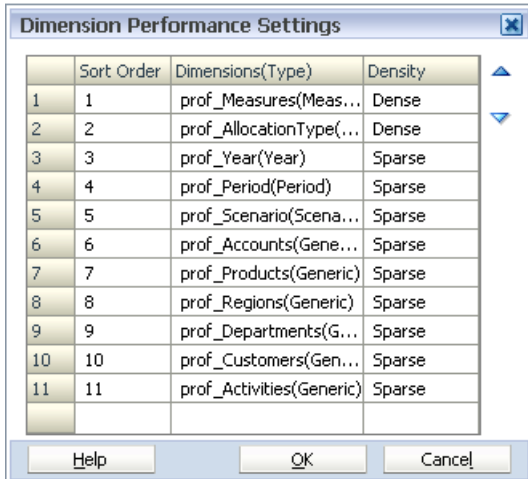
## 次元ソート順の設定



モデル内のすべての次元の処理順序は、Dimension Sort Order プロパティを使用して次元レベルで設定する必要があります。次元ソート順の制限を満たす必要があります; 制限を満たさない場合、モデルの検証は失敗します。制限の完全なリストについては、53 ページの「次元ソート順の推奨事項」を参照してください。

Performance Management Architect で次元ソート順を設定する方法は 2 つあります:


- すべての次元のソート順を一度に設定するには、「次元パフォーマンスの設定」ダイアログ・ボックスを使用します。
  - 1 つの次元の個々の次元ソート順を一度に設定するには、次元ライブラリのプロパティ・グリッドを使用します。
- ▶ 次元パフォーマンスの設定ダイアログ・ボックスを使用して次元ソート順を設定するには:
- 1 EPM Workspace から、「ナビゲート」、「管理」、「次元ライブラリ」の順に選択し、共有ライブラリを表示します。

- 2 アプリケーション名を右クリックし、「次元パフォーマンスの設定」を選択します。



- 3 53 ページの「次元ソート順の設定」で説明されているように、次元を選択し、上矢印  上矢印および下矢印  下矢印を使用して各次元を正しいソート順に移動します。「ソート順」に、次元の元の場所の番号が表示されます。
- 4 オプション: 選択した次元の「密度」で、オプションを表示するセルをダブルクリックし、次元の適切な密度を選択します。56 ページの「Essbase の次元設定の最適化」を参照してください。
- 5 「OK」をクリックします。

▶ 個々の次元の次元ソート順を設定するには:

- 1 EPM Workspace から、「ナビゲート」、「管理」、「次元ライブラリ」の順に選択します。
- 2 Profitability and Cost Management アプリケーションで、次元ソート順を設定する次元を選択します。
- 3 プロパティ・グリッドで、Dimension Sort Order プロパティを選択し、「値」にソートの順序に必要な数字を入力します。53 ページの「次元ソート順の設定」を参照してください。
- 4 「保存」アイコン  をクリックします。

# 13

## Performance Management Architectを使用した詳細 Profitabilityアプリケーション および次元の管理

### この章の内容

アプリケーションおよび次元の操作 .....	123
Essbase の次元設定の最適化 .....	124
メタデータのインポート .....	125
詳細 Profitability and Cost Management の次元プロパティおよびメンバー・プロパティ .....	126
詳細 Profitability アプリケーションの作成 .....	129
Performance Management Architect でのアプリケーションの検証と配置 .....	138

## アプリケーションおよび次元の操作

Profitability and Cost Management では、Performance Management Architect を使用して、収益性モデルに使用される Essbase アウトラインを構築するための次元を選択します。すべての次元およびメンバーは、Performance Management Architect で作成され、モデルを構築するために Profitability and Cost Management アプリケーションにインポートされます。

**注：** ライフサイクル管理は、モデル・データのインポートやエクスポートだけでなく、Performance Management Architect 内のアプリケーションのインポートやエクスポートにも使用できます。

Performance Management Architect では、次のタスクを実行できます：

- 次元の作成、編集およびコピー
- 別名の設定
- 次元の関連付けの作成、表示および削除
- メンバーの作成、表示、名前変更および削除
- プロパティ値の編集
- Profitability and Cost Management へのアプリケーションの配置
- トランザクション・ログの表示
- Profitability and Cost Management と他のアプリケーション、外部ソース(フラット・ファイル)、およびインタフェース・テーブル間のデータの同期化

Performance Management Architect を使用したメタデータおよび次元の操作の詳細は、次の項を参照してください:

- 124 ページの「Essbase の次元設定の最適化」
- 125 ページの「メタデータのインポート」
- 126 ページの「詳細 Profitability and Cost Management の次元プロパティおよびメンバー・プロパティ」
- 129 ページの「詳細 Profitability アプリケーションの作成」
- 138 ページの「Performance Management Architect でのアプリケーションの検証と配置」

## Essbase の次元設定の最適化

一般的な Profitability and Cost Management アプリケーションには、1 つの MeasuresDetailed 次元、複数の POV 次元、および多数のビジネス次元が含まれます。Profitability and Cost Management では、ビジネス次元が複数のステージで使用される場合、ビジネス次元が重複します。このプロセスにより、アプリケーションによって生成される Essbase 計算キューブ・アウトラインはより疎になり、計算スクリプトの実行時のパフォーマンスに影響を与える可能性があります。

Oracle Essbase Administration Services (EAS) コンソールを使用して、生成された計算キューブ・アウトラインの次元ストレージ・タイプを直接変更することにより、ある程度の最適化は簡単に実現できます。

---

**注意** 「次元ストレージ・タイプ」プロパティの変更は、データベース管理者 (DBA) のみが実行する必要があります。指示の詳細は、『Oracle Essbase Administration Services 開発者ガイド』を参照してください。

---

各次元の「次元ストレージ・タイプ」プロパティの設定に関するデフォルトの推奨事項は、次のとおりです:

- メジャー次元および AllocationType 次元は密に設定します
- ビジネス次元および POV 次元はすべて疎に設定します

このデフォルト設定の場合、Essbase 計算キューブのブロック・サイズは約 3K になり、疎次元の次元性に基づく多数の潜在的なブロックが発生します。最大ステージの最も密な次元(データの存在に基づく)は密に設定する必要があります。その場合、AllocationType 次元およびメジャー次元は疎に設定する必要があります。

**注:** このブロック・サイズは、Essbase のベスト・プラクティスの推奨事項の制限内に収まる必要があります。詳細は、『Oracle Essbase Administration Services 開発者ガイド』を参照してください。

### 例

最大ステージは、ステージ内の潜在的なノードの最大数によって定義されます。

一般的な Profitability and Cost Management モデルの最後のステージが顧客 X 製品 (顧客 x 製品の収益性の計算用)として定義されており、これが最大ステージである場合、このステージ内の顧客次元または製品次元を密に設定できます。関連するメジャー次元および AllocationType 次元は疎に設定します。顧客を密に設定するか製品を密に設定するかの決定は、管理しているドライバのデータの密度に基づいて行います。

たとえば、前のステージ内の様々なソースからこのステージに配賦するために使用するドライバが販売量であり、製品次元より顧客次元のほうにより密なデータが含まれる(多くの顧客に対して少ない製品が販売される)場合、顧客を密に設定できます。少ない顧客に対して多くの製品が販売される場合、製品を密に設定できます。いずれの場合も、Essbase ブロック・サイズは、Essbase のベスト・プラクティスの推奨事項の制限内に収まる必要があります。

**注：** 属性の関連付けがある次元は密に設定できません。Essbase では、属性の関連付けは疎次元としてのみ設定できます。

## メタデータのインポート

Performance Management Architect の次元ライブラリは、次元および階層を表示、作成および管理するための、管理者用の中心的な場所です。他のアプリケーションと共有する既存の次元を使用することも、自分のモデル専用のローカル次元を作成することもできます。

次元ライブラリを使用するには、次元エディタのセキュリティの役割がある必要があります。この役割では、次元およびメンバーの作成、削除、変更、インポート・プロファイルの作成、トランザクション・ログの実行などのすべての次元ライブラリ機能へのアクセスが可能です。Oracle Enterprise Performance Management System User Security Administration Guide を参照してください。

Performance Management Architect には、2つのタイプの次元があります：

- 共有次元は、共有ライブラリにリンクされており、次元に対して行われたすべての変更を継承します。
- ローカル次元は、共有ライブラリからアプリケーションへコピーされます。ローカル次元は、共有ライブラリ内の次元に対して行われる将来の変更を継承しません。

---

**注意** Profitability and Cost Management のすべてのプロパティはローカル値です。プロパティが他の階層に存在する場合に、この値を変更しても自動的に継承されません。1つの階層でプロパティを変更しても、他の階層に値が継承されるとは仮定できません。

---

次元ライブラリからこれらのプロパティを編集できます：

- アプリケーション
- 次元
- メンバー

**注：** 時間、通貨および国用の標準次元は、すべての製品を対象として Performance Management Architect で使用できます。

▶ 次元ライブラリにアクセスするには、「ナビゲート」、「管理」、「次元ライブラリ」の順に選択します。

次元およびメンバーを操作する方法の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

## 詳細 Profitability and Cost Management の次元プロパティおよびメンバー・プロパティ

Profitability and Cost Management の次元およびメンバーのプロパティは、表 28 にアルファベット順で表示され、次の情報が含まれます：

- プロパティ・ラベル。プロパティにわかりやすい表示名を提供します。該当する場合、関連データベース・タイプ(ASO)が名前に付加されます。データベース・タイプが指定されていない場合、プロパティは両方のタイプに適用されます。
- 各プロパティの説明
- プロパティに一意の識別子を提供し、インポートおよびバッチ・クライアントでデータを更新する際に使用するプロパティ名。

次元ライブラリでプロパティを選択すると、ドロップダウン・リストまたはデータ入力ボックスを表すプロパティを変更できます。

次元およびメンバーを操作する方法の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

**注意** Profitability and Cost Management のすべてのプロパティはローカル値です。ある階層でプロパティを変更しても、他の階層に値が継承されるわけではありません。

表 28 詳細 Profitability and Cost Management の次元プロパティおよびメンバー・プロパティ

プロパティ・ラベル	説明	プロパティ名
別名	選択した次元メンバーの別名を入力します。別名は、配置したアプリケーションに表示される代替次元メンバー名です。  現在の次元メンバーのデフォルトの別名を割り当てるかまたは変更する場合は、既存の別名をクリックして新規名前を入力します。適切な命名規則に従います。すべてのデータベースには、「Default」という別名テーブルがあります。	
アプリケーション・タイプ	アプリケーション・タイプを選択します： <ul style="list-style-type: none"><li>● 標準 Profitability の場合は「全般」</li><li>● 詳細 Profitability の場合は「詳細」</li></ul>	ApplicationType

プロパティ・ラベル	説明	プロパティ名
属性	色、サイズ、モデルなど次元メンバーの特性を入力します。 たとえば、従業員次元メンバーには、名前、年齢、または住所の各属性がある場合があります。Product 次元のメンバーはサイズ、味などの複数の属性を持つ場合があります。	Attributes
属性タイプ	属性タイプを選択したメンバーに関連付けます。メンバーのクエリおよびフィルタに使用できる次元メンバーに割り当てられた属性の属性値を作成します。 テキスト・タイプのみ現在はサポートされています。	AttributeDataType
コメント	次元またはメンバーのコメントを入力します。コメントには最大 255 文字を指定できます。デフォルトでは、テキスト・ボックスには現在のコメントが表示されます(存在する場合)。 属性次元またはメンバーにコメントを割り当てられません。 <b>注：</b> Unicode 使用可能データベースでは、最大 80 文字まで許可されます。	Comment
集計	メンバー集計プロパティは、子を親にロールアップする方法を指定します。 現在のメンバーが次元または属性ではない場合は、そのメンバーに割り当てる集計演算子を選択します： <ul style="list-style-type: none"> <li>● + (加算) - デフォルト</li> <li>● - (減算)</li> <li>● * (乗算)</li> <li>● / (除算)</li> <li>● % (パーセント)</li> <li>● ~ (集計では無視する)</li> <li>● ^ (集計しない)</li> <li>● NotUsed</li> </ul>	Consolidation
データ・ストレージ(ASO)	次元およびメンバーの ASO ストレージ・オプションを選択します： <ul style="list-style-type: none"> <li>● StoreData - データは次元とともに保管されます。</li> <li>● NeverShare - この次元に関連付けられているデータは、単一の子を持つ親などのように、暗黙の共有関係がある場合にも共有できません。この場合、データは親および子で複製されます。 <b>注：</b> このオプションは、集約ストレージ・アウトラインに保管された階層には適用されません。</li> <li>● LabelOnly - この次元に関連付けられているデータはありません。</li> <li>● ShareData - このメンバーに関連付けられているデータは共有できます。ShareData プロパティは、メンバーのみに適用されます。次元ルート・メンバーは共有できません。</li> </ul>	<ul style="list-style-type: none"> <li>● ASODimensionDataStorage (次元ルート・メンバーの場合)</li> <li>● ASOMemberDataStorage (次元メンバーの場合)</li> </ul>
次元式 (ASO)	セルをダブルクリックし、セレクタ・ボタンをクリックして、メモ・エディタを開きます。式を入力し、「OK」をクリックします。 適切な ASO MDX 構文を使用します。	ASODimensionFormula (次元ルート・メンバー用)



プロパティ・ラベル	説明	プロパティ名
次元解決順	選択した次元のソリューション・シーケンスに数値を入力します。たとえば、この次元を2番目に解決する場合は、「2」を入力します。	DimensionSolveOrder
次元ソート順	数値を順番に入力して Profitability and Cost Management によって生成された Essbase アウトライン内の次元の順序を設定します。 たとえば、この次元が Essbase アウトラインの2番目の次元である場合は、2を入力します。 次元ソート順は、別名次元と UDA 次元を除き、モデル内のすべての次元に設定されている必要があります。 次元ソート順が連続しており、一意かつ1以上である必要があります。 53 ページの「次元ソート順の設定」に概略されているように、ソート順を設定します。	DimensionSortOrder
階層タイプ (次元のみ)	「階層タイプ」プロパティは、Essbase の集約ストレージ(ASO)データベースにのみ使用されます。 詳細 Profitability の場合、この値は「使用不可」に設定されます。	DimensionHierarchyType
階層タイプ (メンバーのみ)	詳細 Profitability の場合、この値は「使用不可」に設定されます。	HierarchyType
集約でのレベルの使用	このプロパティは、ASO アウトライン内の保管済階層をホストする Gen1 または Gen2 の階層メンバーに適用されます。 詳細 Profitability の場合、この値は「デフォルト」に設定されます。	
メンバー式(ASO)	セルをダブルクリックし、セレクタ・ボタンをクリックして、メモ・エディタを開きます。式を入力し、「OK」をクリックします。 適切な ASO MDX 構文を使用します。	ASOMemberFormula
メンバー解決順 (メンバーのみ)	選択したメンバーのソリューション・シーケンスに数値を入力します。たとえば、このメンバーを2番目に解決する場合は、「2」を入力します。 このプロパティは、ASO データベースのみに適用されます。 解決順が0のメンバーは、その次元の解決順序を継承します。 解決順が同じメンバーは、次元ソート順プロパティで別の値が指定されていないかぎり、その次元がデータベース・アウトラインに出現する順序で評価されます。 解決順のないメンバーは、解決順のあるメンバーの後に評価されます。	MemberSolveOrder
POV 次元	この次元を選択したモデルの POV 次元として設定する場合は、このチェック・ボックスを選択します。	IsPOVDimension
POV 表示順	複数の POV 次元がある場合は、1、2、3 などの数値を入力して、各 POV 次元の表示順を設定します。	POVDisplayOrder



プロパティ・レベル	説明	プロパティ名
プライマリ・レベルの加重	<p>プライマリ・レベルの加重は、集約操作中のビュー選択エンジンで選択できるレベルを制限します。これらの設定は、Essbase (ASO) アプリケーションのみに適用されます。</p> <p>次のオプションのいずれかを選択します:</p> <ul style="list-style-type: none"> <li>「Default」 - ビュー選択エンジンは選択するレベルを自由に決定できます。これは、新規の階層に対する初期値です。</li> <li>「AllLevels」 - ビュー選択エンジンは階層のすべてのレベルを集約対象とみなします。</li> <li>「NoAggregation」 - ビュー選択エンジンはどのレベルも集約用に選択できません。</li> <li>「TopLevelOnly」 - ビュー選択エンジンは次元の最上位レベルのみを集約対象とみなします。</li> <li>「BottomTop」 - ビュー選択エンジンは、次元の最上位レベルおよび最下位レベルのみを集約対象とみなします。</li> <li>「BottomLevelOnly」 - ビュー選択エンジンは次元の最下位レベルのみを集約対象とみなします。</li> </ul>	PrimaryLevelWeighting
UDA	<p>選択したメンバーの UDA メンバーを選択します。</p> <p>UDA 次元は、Performance Management Architect に存在し、UDA が追加される先のベースに関連付けられている必要があります。</p> <p>最大 80 文字(英数字)の UDA メンバーを選択できます。</p> <p><b>注:</b> Unicode 使用可能データベースでは、最大 80 文字まで許可されます。</p>	UDA
Unicode	<p>アプリケーション・レベルで「Unicode」を選択し、アプリケーションを Unicode モードに設定します。Unicode モードのアプリケーションは複数の文字セットをサポートします。</p>	Unicode

## 詳細 Profitability アプリケーションの作成

次の 2 つの方法のいずれかを使用して、Performance Management Architect で詳細 Profitability アプリケーションを作成できます:

- アプリケーションおよび次元を自動的に作成する「アプリケーション・ウィザード」を使用して詳細 Profitability and Cost Management アプリケーションを作成します。
- ブランクの詳細 Profitability アプリケーションを作成し、次元を手動で選択して名前を付けます。

次の項では、詳細 Profitability アプリケーションの作成方法を順を追って説明します:

- [130 ページの「詳細 Profitability アプリケーションのタイプの選択」](#)
- [130 ページの「詳細 Profitability の次元の選択」](#)
- [135 ページの「アプリケーション設定の変更」](#)

## 詳細 Profitability アプリケーションのタイプの選択

▶ Profitability and Cost Management アプリケーションを作成するには:

- 1 フラット・ファイルのインポートまたは Performance Management Architect インタフェース・テーブルのインポートを使用して Performance Management Architect に新規共有ライブラリを移入します。

---

**注意** アプリケーションに含めるビジネス次元(汎用、勘定科目、エンティティ、時間、国など)は、アプリケーションを作成する前に、次元ライブラリに追加する必要があります;これを行わないと、「アプリケーション・ウィザード」での選択に次元を使用できません。

---

- 2 EPM Workspace から、「ナビゲート」、「管理」、「アプリケーション・ライブラリ」の順に選択します。  
アプリケーション・ライブラリが表示されます。
- 3 「ファイル」、「新規」、「アプリケーション」の順に選択します。  
ウィザードの最初の「アプリケーション・タイプ」画面が表示されます。
- 4 「名前」で、アプリケーション名を入力します。  
名前は7文字以内で命名し、「&」(アンパサンド)を含む特殊文字を使用しないでください。付録 D 「Essbase 命名規則」を参照してください。
- 5 「タイプ」で、「Profitability」を選択します。
- 6 オプション: 「説明」に、説明を入力します。
- 7 オプション: 次元を手動でブランク・アプリケーションに追加するには、「ブランク・アプリケーションの作成」を選択し、「終了」をクリックします。134 ページの「詳細 Profitability 次元の手動での追加」を参照してください。
- 8 オプション: 次元を作成するには、「ローカル次元の自動作成」を選択します。  
「ローカル次元の自動作成」を選択すると、MeasuresDetailed 次元が自動的に作成されます。  
ウィザードの2番目の画面である「次元の選択」が表示されます。
- 9 「Profitability」で、「Detailed アプリケーションとして作成」を選択します。
- 10 「次」をクリックします。130 ページの「詳細 Profitability の次元の選択」を参照してください。

## 詳細 Profitability の次元の選択

アプリケーションを作成した後、アプリケーションに含める次元を選択する必要があります。

Profitability and Cost Management アプリケーションに必要な次元は次のとおりです:

- MeasuresDetailed は、ステージの貸借一致および計算に必要な予約済の次元です。AllocationMeasures が含まれますが、ドライバ・メジャーは含まれません。

---

**注意** この次元は編集しないでください。変更を行うと、データの損失やモデルの破損が生じる可能性があります。

---

- ユーザーは、少なくとも1つの視点次元(POV)を定義する必要があります。モデル POVにより、選択した POV または期間のモデル情報に関する具体的なビューを参照できます。たとえば、POV 次元タイプには次のような次元がいくつか含まれる場合があります:
  - 年
  - 時間
  - 期間
  - シナリオ
  - 月
  - 財務四半期
  - 季節
- ユーザーは、少なくとも1つのビジネス次元を定義する必要があります。ビジネス次元は、ビジネス固有の部署、総勘定元帳の勘定科目、アクティビティ、場所、顧客または製品などのモデル内のビジネス要素を説明するために作成されます。

ビジネス次元には、次の次元タイプの一部または全部が含まれ、1つ以上のステージまたはモデルに適用される場合があります:

- 汎用
- 勘定科目
- エンティティ
- 時間
- 国

**注:** これらのビジネス次元は Profitability and Cost Management アプリケーションの一部として含められますが、Essbase アウトラインが配置されると、タイプのない基本または汎用次元として作成されます。

- 属性次元は、ビジネス次元に関連付けられた特殊なタイプの次元です。属性は、製品のサイズや色など、データの特性を示します。

Performance Management Architect には、2つの属性次元タイプがあります:

- 属性次元:
  - 属性は、テキストを使用して作成されます。
  - 属性には階層があり、集計値にロールアップすることができます。
  - 1つのメンバーに1つの属性のみ関連付けることができます。
- ユーザー定義属性次元(UDA):
  - 属性は、テキストを使用して作成する必要があります。
  - UDA には階層がないため、合計を生成するためのレポートに容易に使用することはできません。
  - 複数の UDA を1つのメンバーに関連付けることができます。

属性次元の操作の詳細は、『Oracle Essbase データベース管理者ガイド』を参照してください。

- 別名はオプションです。モデルで別名を使用する場合のみ必要です。命名の規則については、[付録 D「Essbase 命名規則」](#)を参照してください。

---

**注意** 同じ次元内で重複するメンバー名または別名は使用できません。

---

**注：** モデル・ステージは次元ではないため、Performance Management Architect では使用できません。Profitability and Cost Management では、ステージはモデルに追加され、次元を論理的な手順またはステージに整理するために使用されます。

アプリケーションを作成した後、適切な手順を使用してアプリケーションに追加する次元を選択します：

- ウィザードを使用する場合は、[132 ページの「ウィザードを使用した詳細 Profitability の次元の追加」](#)を参照してください。
- ブランクのアプリケーションを作成した場合は、[134 ページの「詳細 Profitability 次元の手動での追加」](#)を参照してください。

## ウィザードを使用した詳細 Profitability の次元の追加

アプリケーション・ウィザードを使用すると、必要なすべての次元が自動的に表示されます。正確な一致がある場合は、その次元タイプの次元列に自動的に入力されます。Profitability and Cost Management の必要な次元タイプは、自動的に分類されて、次の影付きヘッダーとともに表示されます：

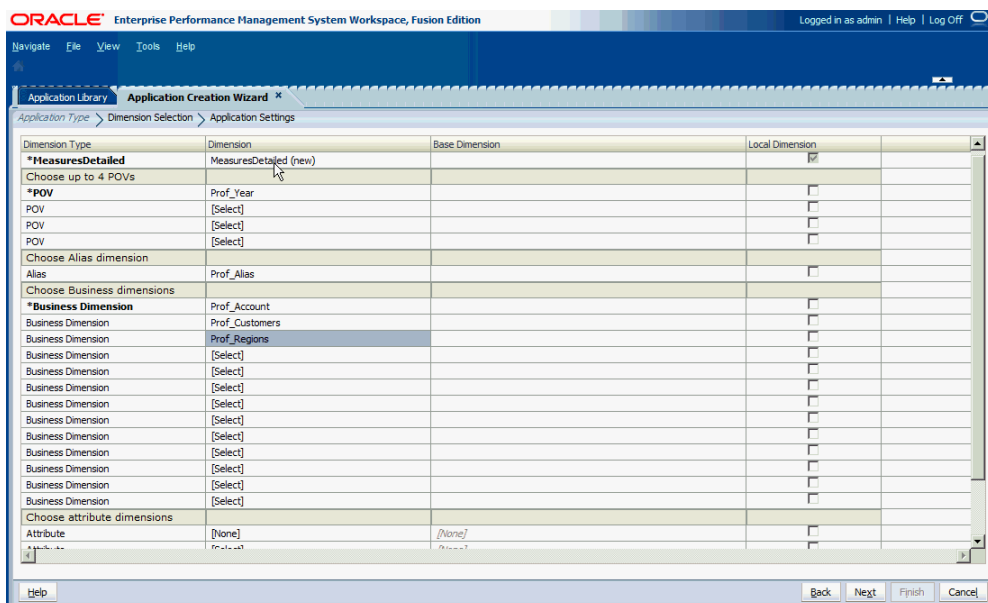
- MeasuresDetailed 次元
- POV 次元
- 別名次元(オプション)
- ビジネス次元
- 属性次元(オプション)

「ローカル次元の自動作成」を選択した場合、必要な次元ごとに新規ローカル次元が作成されます。各新規次元の名前は、次元タイプにかっこで囲まれた(New)が付いたものになります。たとえば、Account (New)となります。

勘定科目、エンティティ、時間、国などアプリケーションに含めるビジネス次元は、アプリケーションを作成する前に次元ライブラリに追加する必要があります；そうしないと、「アプリケーション・ウィザード」で次元を選択できません。名前について制限されている単語および文字を確認するには、[付録 D「Essbase 命名規則」](#)を参照してください。

▶ 次元を定義するには：

- 1 「次元の選択」タブ(アプリケーション・ウィザードの手順 2)の「次元のタイプ」で、必要な次元タイプを確認します。



2 「MeasuresDetailed」を選択します。

**注意** この次元は、詳細 Profitability アプリケーションに必須ですが、変更する必要があります。変更しないとアプリケーションが破損する可能性があります。

3 既存の各次元をアプリケーションに追加するには、「次元」列の「[選択]」をクリックすると、その次元タイプに使用可能な次元のドロップダウン・リストが表示されます。

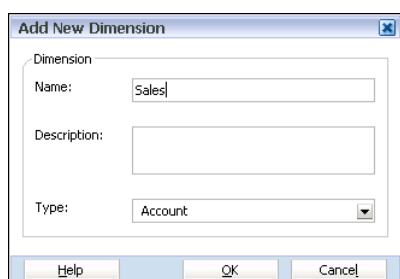
関連次元タイプの次元のみが表示されます。たとえば、次元をメジャー次元タイプにマッピングしている場合は、メジャー次元のみがリストに表示されます。

4 ドロップダウン・リストから次元を選択します。

5 オプション: 新規次元を作成します。新規次元を作成するには:

1. 「次元」列で、「[選択]」をクリックすると、その次元タイプに使用可能な次元のドロップダウン・リストが表示されます。
2. ドロップダウン・リストから、「[新規次元の作成]」を選択します。

「新規次元の追加」ダイアログ・ボックスが表示されます。



3. 新規次元の「名前」と「説明」を入力します。「タイプ」は、次元タイプに基づいて自動的に選択されます。
4. 「OK」をクリックします。

6 手順 2 から手順 4 を繰り返し、新規次元を選択または作成します。

**注：** 次の画面に移動する前に、すべての必須次元を、次元または[該当なし]にマップする必要があります。

- 7 **オプション:** 次元をローカル次元として指定するには、各行の最後にある「ローカル次元」チェック・ボックスを選択します。
- 8 「次」をクリックします。  
「アプリケーション設定」画面が表示されます。
- 9 必要に応じて、アプリケーション設定を変更します。135 ページの「[アプリケーション設定の変更](#)」を参照してください。

## 詳細 Profitability 次元の手動での追加

ブランク・アプリケーションを作成する場合は、次元ライブラリを使用して次元を選択します。

次元ビューには、次の3つのペインがあります:

- **共有ライブラリ** - 左側に表示されます。共有ライブラリで現在使用可能なすべての次元が表示されます。
- **アプリケーション** - 中央部に表示され、展開すると現在のアプリケーションのすべての次元の名前がリスト表示されます。
- **プロパティ・グリッド** - 右側に表示されます。選択した次元またはメンバーのすべてのプロパティを表示します。

次元ライブラリの使用の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

▶ ブランク・アプリケーションに次元を追加するには:

- 1 次元ライブラリで、共有ライブラリからアプリケーションに次元をドラッグします。

「次元のコピー」ダイアログ・ボックスが表示されます。


**ヒント：** 新規ローカル次元を作成するには、アプリケーション・ビューでアプリケーション名を右クリックし、「次元の作成」を選択します。  
「新規次元の追加」ダイアログ・ボックスが表示されます。

- 2 「ローカル次元」または「共有次元」を選択します。

**ヒント：** アプリケーションにメンバーのみを追加するには、共有ライブラリで次元を右クリックして、「アプリケーション・ビューに追加」を選択します。

- 3 必要なすべての次元をアプリケーションに追加します。

展開されたアプリケーションにすべての次元が表示されます。

**ヒント:** 共有ライブラリにメンバーを追加し、アプリケーションに次元を追加した場合、「リフレッシュ」ボタンをクリックすると、アプリケーションに新規メンバーが表示されます。

- 4 「NoMember」の「集計プロパティ値」「」を「~」(無視)に設定します。
- 5 次元を順序変更して、NoMember が最後の世代 2 の位置に移動されるようにします。137 ページの「子メンバーの順序変更」を参照してください。
- 6 必要に応じて、アプリケーション設定を変更します。135 ページの「アプリケーション設定の変更」を参照してください。

## アプリケーション設定の変更

### サブトピック

- [次元ライブラリを使用したアプリケーション設定の変更](#)
- [アプリケーション・ウィザードを使用したアプリケーション設定の変更](#)
- [POV 次元のプロパティの設定](#)
- [子メンバーの順序変更](#)

アプリケーションの作成の次の手順は、アプリケーションのプロパティ、関連付けおよび他のアプリケーション設定の変更です。ブランク・アプリケーションを作成し、次元ライブラリを使用してアプリケーション設定を変更する場合、変更を加えるためにウィザードではなく次元ライブラリを使用するという点を除けば、同様のオプションを使用できます。

**注:** 次元の関連付けは、共有階層に必要ですが、すべてのモデルで必要というわけではありません。

アプリケーション設定を変更する次の各手順を参照してください:

- ブランク・アプリケーションの場合は次元ライブラリを使用します。135 ページの「[次元ライブラリを使用したアプリケーション設定の変更](#)」を参照してください。
- 自動生成アプリケーションの場合は「アプリケーション・ウィザード」を使用します。136 ページの「[アプリケーション・ウィザードを使用したアプリケーション設定の変更](#)」を参照してください。

## 次元ライブラリを使用したアプリケーション設定の変更

次元ライブラリには、共有ライブラリ、アプリケーションおよびプロパティ・グリッドが表示されます。右クリック・メニューを使用して、アプリケーション次元を変更し、次のアクションを実行します:

- Profitability and Cost Management モデルの場合は、次のタスクを実行します:
  - POV 次元のプロパティ、および必要に応じて、複数の POV 次元の POV 表示順を設定します。

136 ページの「[POV 次元のプロパティの設定](#)」を参照してください。



- すべてのモデル次元の次元ソート順を設定します。  
120 ページの「次元ソート順および密度の設定」を参照してください
- NoMember の順序を変更し、このメンバーをリスト上の最終世代 2 のメンバーとして表示します。  
137 ページの「子メンバーの順序変更」を参照してください
- 必要に応じて、Oracle Hyperion Enterprise Performance Management Architect Administration Guide に記載されている手順を使用して次元を変更します。
  - 次元別名を設定します。
  - アプリケーション・メンバーシップを表示します。
  - 次元の関連付けを表示します。
  - 次元を同期化します。
  - メンバーを作成します。
  - 共有メンバーを挿入します。
  - 孤立メンバーを管理します。

## アプリケーション・ウィザードを使用したアプリケーション設定の変更

ウィザードを使用してアプリケーションを作成する場合、アプリケーションの次元およびメンバーが左側に表示され、プロパティ・グリッドが右側に表示され、検証および配置オプションが下部に表示されます。

右クリック・メニューを使用してアプリケーション次元を変更し、次のタスクを実行します:

- Profitability and Cost Management モデルの場合は、次のタスクを実行します:
  - POV 次元のプロパティ、および必要に応じて、複数の POV 次元の POV 表示順を設定します。  
136 ページの「POV 次元のプロパティの設定」を参照してください。
  - すべてのモデル次元の次元ソート順を設定します。  
120 ページの「次元ソート順および密度の設定」を参照してください
  - NoMember の順序を変更し、このメンバーをリスト上の最終世代 2 のメンバーとして表示します。  
137 ページの「子メンバーの順序変更」を参照してください
- 必要に応じて、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』に記載されている手順を使用して次元を変更します:

## POV 次元のプロパティの設定

少なくとも 1 つの POV(視点)次元を、詳細 Profitability and Cost Management モデルごとに設定する必要があります。POV 次元は、任意のカスタム値に設定できますが、通常は、年、期間、シナリオなどの期間を示します。また、POV バージョンを作成することもできます。



複数の POV 次元が指定されている場合、POV の表示順序も設定してこれらの次元の計算にシーケンスを付ける必要があります。Performance Management Architect で設定された POV の表示順序は、Profitability and Cost Management で自動的に取得されます。

▶ POV 次元プロパティを設定するには:

- 1 次元ライブラリで、最初の POV 次元(年など)を選択します。
- 2 プロパティ・グリッドで「カテゴリ」の下の「収益性」を選択します。
- 3 「POV 次元」を選択します。
- 4 **オプション:** 期間やシナリオなど、他の POV 次元を選択するには、**手順 1** から**手順 3**を繰り返します。
- 5 **オプション:** 複数の POV 次元が選択された場合、POV 次元ごとに「POV の表示順序」を設定します。

「POV の表示順序」セルをダブルクリックして、選択した POV 次元のシーケンス番号を入力します。たとえば、年、期間およびシナリオが POV 次元として設定されている場合、年の POV の表示順序に 1、期間に 2、およびシナリオに 3 を表示順序として設定します。

## 子メンバーの順序変更

モデル内のすべてのビジネス次元については、NoMember がメンバー・リストの最後の世代 2 メンバーとして表示されるようにリストを順序変更する必要があります; これを行わないと、モデルの検証は失敗します。

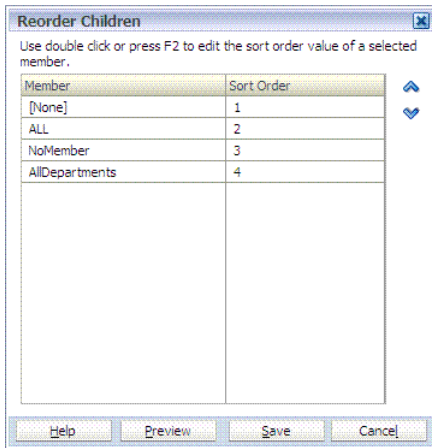
NoMember が最後の位置に残っていれば、特定の要件にあわせてメンバーの順序も変更できます。

**注:** この手順は、別名、AllocationType、メジャー、期間、シナリオ、年などのシステム次元では必要ありません。

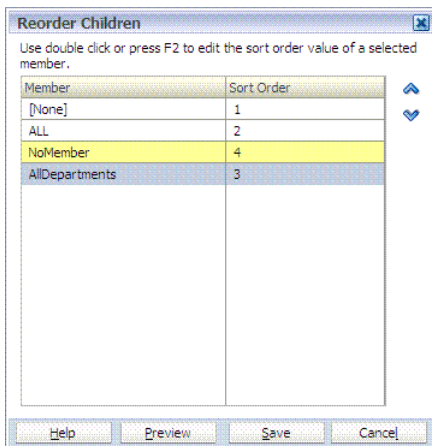
▶ 子を順序変更するには:

- 1 EPM Workspace から、「ナビゲート」、「管理」、「次元ライブラリ」の順に選択し、共有ライブラリを表示します。
- 2 次の説明に従って、記録する次元を選択します:
  - 共有次元の場合、「共有ライブラリ」の下で、順序変更する次元を右クリックして「子の順序変更」を選択します。
  - ローカル次元の場合、「アプリケーション」列の下で、順序変更する次元を右クリックして「子の順序変更」を選択します。

「子の順序変更」ダイアログ・ボックスが表示されます。



- 3 「ソート順」で、メンバーの順序を変更するセルをダブルクリックし、新しいソート順の場所の番号を入力します。



**注：** NoMember 次元が最後の世代 2 メンバーとしてリストの最後に移動したことを確認します; 最後でない場合、配置は失敗します。

- 4 手順 3 を繰り返して、すべてのメンバーを順序付けます。
- 5 変更をプレビューするには、「プレビュー」をクリックします。
- 6 「保存」をクリックします。

## Performance Management Architect でのアプリケーションの検証と配置

Performance Management Architect で詳細 Profitability and Cost Management アプリケーションを作成したら、アプリケーションを検証して詳細 Profitability に配置する必要があります。

次の項を参照してください:

- [139 ページの「詳細 Profitability and Cost Management の検証」](#)
- [140 ページの「ウィザードを使用したアプリケーションの検証と配置」](#)
- [141 ページの「Performance Management Architect を使用したアプリケーションの検証と配置」](#)

**注：** アプリケーションの再配置時に個別メンバー、複数のメンバーまたはメンバーの組合せの追加、名前変更、親変更または削除が可能です。

Performance Management Architect を使用して詳細 Profitability and Cost Management アプリケーションを作成および配置する方法の詳細は、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』を参照してください。

## 詳細 Profitability and Cost Management の検証

Profitability and Cost Management では、次の条件が検証されます：

表 29 詳細 Profitability の検証

検証レベル	検証
アプリケーション	<ul style="list-style-type: none"> <li>● アプリケーションの名前は 7 文字以下にする必要があり、特殊文字は使用できません。</li> <li>● 少なくとも 1 つの次元を POV タイプに設定する必要があります。最大 4 つの次元を POV 次元としてマークできます。各 POV 次元クラスは 1 回のみ使用できます。</li> <li>● 少なくとも 1 つのビジネス次元が定義されている必要があります。</li> <li>● 少なくとも 1 つの MeasuresDetailed 次元が定義されている必要があります。</li> <li>● タイプが勘定科目の次元が 1 つしかありません。</li> <li>● タイプがエンティティの次元が 1 つしかありません。</li> <li>● アプリケーション名に Essbase の特殊文字および予約語が含まれていません。</li> </ul>
次元	<ul style="list-style-type: none"> <li>● ビジネス次元のルート・メンバーには、「LabelOnly」に設定された ASO データ・ストレージが必要です。</li> <li>● 次元ソート順が、別名次元と UDA 次元を除き、モデル内のすべての次元に設定されており、次の条件を満たしています： <ul style="list-style-type: none"> <li>○ 別名次元と UDA 次元を除く、モデル内のすべての次元に次元ソート順が設定されている必要があります。</li> </ul> <p><b>注：</b> 別名次元と UDA 次元は、次元ソート順では無視されます。</p> <ul style="list-style-type: none"> <li>○ 次元ソート順は連続している必要があります。</li> <li>○ メジャー次元がデフォルトで 1 に設定されています。</li> <li>○ POV 次元とビジネス次元が 3 以上に設定されています。</li> <li>○ 属性次元は最後の次元としてソートされています。</li> </ul> <p>たとえば、12 個の連続した次元のうち 4 個が属性次元である場合、属性次元は 9、10、11 および 12 に設定する必要があります。</p> </li> <li>● ビジネス次元には少なくとも 2 つの世代 2 メンバーが必要です。</li> <li>● POV 次元には少なくとも 1 つのメンバーが必要です。</li> <li>● 属性次元は次の条件を満たす必要があります： <ul style="list-style-type: none"> <li>○ 属性次元のレベル 0 のメンバーのみが属性として割当て可能です。</li> <li>○ 属性は基本次元の同じレベルのメンバーにのみ割当て可能です。</li> <li>○ 属性次元は疎次元にのみ関連付けられます。</li> </ul> </li> </ul>

検証レベル	検証
メンバー	<ul style="list-style-type: none"> <li>● ASO データ・ストレージのみを定義できます。</li> <li>● 共有メンバーは、最初の世代 2 メンバーでは許可されていません。</li> <li>● 共有メンバーは、アウトライン順序で常に対応する基本メンバーの後に表示される必要があります。</li> <li>● NoMember は、すべてのビジネス次元に対して最後の世代 2 メンバーとして設定する必要がありますが、プロパティ・グリッドで「無視(-)」に設定する必要があります。</li> </ul> <p><b>注：</b> この要件は、POV、メジャー、別名、UDA または属性次元に適用されません。</p> <ul style="list-style-type: none"> <li>● 次のいずれかの方法によって、共有メンバーを動的階層に置く必要があります： <ul style="list-style-type: none"> <li>○ 世代 1 メンバーが HierarchyType=dynamic です</li> <li>○ 世代 1 メンバーが HierarchyType=HierarchiesEnabled であり、共有メンバーの世代 2 の祖先が HierarchyType=dynamic です</li> </ul> </li> <li>● 次元のメンバーに重複するメンバー名または別名はありません。</li> <li>● メンバー名に Essbase の特殊文字および予約語が含まれていません。</li> </ul>

## ウィザードを使用したアプリケーションの検証と配置

アプリケーション・ウィザードを使用している場合は、アプリケーションの作成を終了する前にアプリケーションを検証して、すべてのエラーを修正します。必要に応じて、ウィザードの検証および配置オプションを使用せずに、後で配置オプションをアプリケーション・ライブラリで使用することもできます。

▶ ウィザードを使用してアプリケーションを検証および配置するには:

- 1 アプリケーション・ウィザードの「アプリケーション設定」画面で、「検証」をクリックします。

すべてのエラーはグリッドに表示され、エラー・タイプとエラー・メッセージが示されます。

- 2 **オプション:** 検証後にアプリケーションを配置するには、「終了時に配置」を選択します。

- 3 検証エラーが発生した場合、アプリケーションを配置する前にすべての問題を修正します。

- 4 「終了」をクリックします。

「終了時に配置」を選択した場合、「配置」ダイアログ・ボックスが表示されます。

**注：** 共有次元を変更した場合、その共有次元を使用するすべてのアプリケーションが影響を受けます。変更内容を適用するには、アプリケーションを再配置する必要があります。

- 5 アプリケーションを配置します。141 ページの「[Performance Management Architect を使用したアプリケーションの検証と配置](#)」を参照してください。

# Performance Management Architect を使用したアプリケーションの検証と配置

▶ Profitability and Cost Management アプリケーションを Performance Management Architect で検証および配置するには:

- 1 EPM Workspace から、「ナビゲート」、「管理」、「アプリケーション・ライブラリ」の順に選択し、アプリケーション・ライブラリを表示します。
- 2 アプリケーション名を右クリックし、「検証」を選択して、Performance Management Architect に新たに作成した Profitability and Cost Management アプリケーションを検証します。Profitability and Cost Management Profitability の検証条件は、[139 ページの「詳細 Profitability and Cost Management の検証」](#)を参照してください。

検証が完了すると、「ジョブ・タスク」ダイアログ・ボックスが表示され、関連付けられたジョブ番号が表示されます。

- 3 関連付けられたジョブのリンクをクリックすると、ジョブ・コンソールが表示されます。
- 4 ジョブ・コンソールの要約を確認します。エラーがある場合、「添付ファイル」の下にある検証ログのリンクをクリックして、すべての検証エラーまたはメッセージの完全リストを表示します。
- 5 検証ログ内のエラーをすべて修正します。
- 6 エラーがなくなり検証に成功するまで、[手順 2](#) から [手順 5](#) を繰り返します。

これで、アプリケーションは配置可能になりました。

- 7 アプリケーション・ライブラリで、アプリケーション名を右クリックし、ドロップダウン・リストから「配置」を選択します。

アプリケーションは、配置前に検証されます。検証エラーは、すべてジョブ・コンソールに表示されます。配置前にすべての検証エラーを修正する必要があります。検証が完了すると、「配置」ダイアログ・ボックスが表示されます。

- 8 配置に必要な情報を指定します:

- **オプション:** 「ノート」で、このアプリケーションの配置に関する備考を追加します。これらの備考はジョブ・コンソールのジョブの説明に追加されます。このテキスト・フィールドには 200 文字まで入力でき、任意の文字を含めることができます。
- 「インスタンス名」で、アプリケーションの配置先の Profitability and Cost Management インストール名を選択します。

Profitability and Cost Management アプリケーションは、単一マシンまたはクラスタの一部として複数のマシンにインストールできます。各インストールまたはインスタンスは、「インスタンス名」ドロップダウン・リストに表示されます。初回インストール時には、コンフィグレータ・ツールで使用されるインスタンスの名前がデフォルトで指定されます。

- 「アプリケーション・サーバー」を選択します。

- 「Shared Services プロジェクト」で、このアプリケーションを割り当てる「Profitability and Cost Management」アプリケーション・グループを選択します。
- **オプション:** 次のようにして、「完全配置」を選択するかどうかを決定します:
  - アプリケーションの初めての配置の場合、Performance Management Architect はすべての次元を Profitability and Cost Management に自動的に送信するため、「完全配置」チェックボックスは無効になっています。
  - 後続の配置で「完全配置」が選択されている場合、Performance Management Architect はすべての次元を Profitability and Cost Management に送信します。
  - 「完全配置」が選択されていない場合、Performance Management Architect は変更された次元のみを Profitability and Cost Management に送信します。次元が変更されていない場合、その次元は配置に含まれません。

## 9 「配置」をクリックします。

検証ジョブが送信されたことを確認するジョブ・タスク・ウィンドウが表示され、ジョブ ID が表示されます。

## 10 「ジョブ・タスク」・ウィンドウでリンクをクリックすると、ジョブ・ステータスが表示されます。

ジョブを完了すると、新規アプリケーションが作成され、Profitability and Cost Management に配置されたことを示すメッセージが「詳細」に表示されます。新規アプリケーションを選択できるようになります。

**注:** アプリケーションの配置が正常に終了していない場合、アプリケーション・ライブラリで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クリーンアップ」を選択します。このオプションにより、Oracle Hyperion Shared Services と Profitability and Cost Management の両方に対して強制的に削除コールが実行され、アプリケーションのすべてのトレースが削除されて、Performance Management Architect 内のアプリケーションが未配置の状態にリセットされます。

## この章の内容

計算前および計算後のカスタム・スクリプト .....	143
データ転送を行うための ODBC データ・ソースの作成.....	146
高度な計算オプション.....	146
その他のプロセス・タイプ.....	147

モデルの検証後、モデルを計算できます。

「計算の管理」画面で、詳細 Profitability のユーザーは「カスタム・スクリプト」または「データ POV」などの処理オプションを選択します。

基本的な計算機能に加えて、管理ユーザーには追加の演算およびプロセス・タイプも表示されます。

## 計算前および計算後のカスタム・スクリプト

アプリケーションの計算前または計算後にタスクを処理する計算前および計算後のカスタム・スクリプトを作成し、モデルまたはステージ・データを操作することができます。

カスタム・スクリプトは、次のように実行されます。

- 計算前の SQL スクリプトは、モデル計算の前に実行されます。
- 計算後の SQL スクリプトは、モデル計算の後に実行されます。

カスタム・スクリプトは、モデル計算の前に配賦を準備する、またはモデル計算後のレポートの結果を強化するために実行されます。

SQL Developer またはサードパーティ製の任意の SQL ツールを使用して、単純な一連の SQL 文を、区切り文字で区切って作成します。カスタム・スクリプトは、製品スキーマ内で、データベースの HPM\_SQL\_SCRIPT テーブルの SCRIPT 列に格納されます。

計算パラメータを作成するとき、カスタム・スクリプトを選択します。「タスク領域」から、「計算」、「計算の管理」の順に選択し、必要な計算前または計算後のスクリプトを選択します。詳細な手順は、『Oracle Hyperion Profitability and Cost Management ユーザー・ガイド』を参照してください。

次の項を参照してください:

- 144 ページの「HPM\_SQL\_SCRIPT」



- 144 ページの「カスタム・スクリプトの作成」

## HPM\_SQL\_SCRIPT

このテーブルには、計算前および計算後のカスタム・スクリプトが格納されます。テーブルの 1 行ごとに、ステージ・モデルで実行できる 1 つまたは複数の SQL 文が格納されます。

**注：** HPM\_SQL\_SCRIPT に直接、スクリプトを入力しないでください。144 ページの「カスタム・スクリプトの作成」を参照してください。

表 30 HPM\_SQL\_SCRIPT

列	データ型	NULL 許容	説明
NAME	Varchar2 (80)	いいえ	カスタム・スクリプトの名前
APPLICATION_NAME	Varchar2 (8)	いいえ	このカスタム・スクリプトが作成された対象のアプリケーションの名前
MODEL_POV_NAME	Varchar2 (80)	はい	選択した POV が "Year": "2012", "Period": "January", "Scenario": "Actual" というフォーマットで表示されます。 <b>注意:</b> この列を NULL に設定した場合、どの「モデル POV」にもスクリプトを使用できます。 特定の POV を選択した場合、スクリプトはその POV にのみ使用できます。
SCRIPT_TYPE	Varchar2 (80)	はい	適切なスクリプト・タイプを選択します。 <ul style="list-style-type: none"> <li>● PRE (事前スクリプトを表示)</li> <li>● POST (事後スクリプトを表示)</li> <li>● EITHER (事前スクリプトまたは事後スクリプトを表示)</li> <li>● スクリプト・タイプが指定されなかった場合は、EITHER と想定されます。</li> </ul>
DESCRIPTION	Varchar2 (255)	はい	スクリプトの目的または内容の説明を入力します。
SCRIPT	LONG	いいえ	カスタム・スクリプトをここに入力します。 スクリプトの作成手順については、144 ページの「カスタム・スクリプトの作成」を参照してください。

## カスタム・スクリプトの作成

カスタム・スクリプトは、メモ帳、Textpad、Oracle SQL Developer などのテキスト・エディタで作成できます。スクリプトは、SQL 文 1 つだけの場合もありますが、それより長い場には、文と文の間に区切り文字が必要です。必要な区切り文字はスラッシュ (/) で、それだけを独立した行に指定します。このスクリプトではストアド・プロシージャを実行できます。



スクリプトは、144 ページの「HPM\_SQL\_SCRIPT」で説明されているように、HPM\_SQL\_SCRIPT テーブルの SCRIPT 列に貼り付けるか、その他の方法でロードします。HPM\_SQL\_SCRIPT に直接、スクリプトを入力しないでください。

---

**注意** 事前および事後のスクリプトを作成できるのは、HPM 製品スキーマに対するパスワードを持つシステム管理者のみです。

---

▶ カスタム・スクリプトを作成してロードするには:

- 1 メモ帳、Textpad、Oracle SQL Developer などのテキスト・エディタでカスタム・スクリプトを作成します。
- 2 Oracle SQL Developer を使用して、HPM\_SQL\_SCRIPT テーブルを開きます。テーブルは、HPM 製品スキーマ内で、製品テーブルと同じ場所にある必要があります。
- 3 新しい行を挿入します。
- 4 スクリプト・タイプなど、必要な値をテーブルに入力します。144 ページの「HPM\_SQL\_SCRIPT」を参照してください。
- 5 「編集」ダイアログの SCRIPT 列に、次のようにスクリプトをコピーします。
  - Oracle SQL Developer の場合、テキスト・エディタからスクリプトをコピーして貼り付けるか、更新文を使用して SCRIPT 列に移入します。これは Oracle に対する SQL スクリプトの例です:

```
        update my_table set my_column =5
    /
    update your_table set your_column =6
    /
    begin execute my_proc end
    /
```

- SQL Studio を使用する MS SQL Server の場合、Update 文を使用して SCRIPT 列に移入します。これは SQL Server に対する SQL スクリプトの例です:

```
        update my_table set my_column =5
    /
    update your_table set your_column =6
    /
    begin myproc; end
    /
```

---

**注意** 同じスクリプトで複数の SQL 文を指定する場合は、区切り文字「/」を間に含める必要があります。「/」文字は、各文の終わりごとに独立した行として入力してください。

---

# データ転送を行うための ODBC データ・ソースの作成

「計算の管理」タブのデータ転送オプションでは、「PROFITABILITY\_DS」という ODBC データ・ソースが必要です。これが、Profitability and Cost Management インスタンスをサポートする、Oracle or MS SQL Server データベース内の製品スキーマに接続します。

▶ この ODBC データ・ソースを作成するには:

- 1 アプリケーション・サーバーで、ODBC 管理ユーティリティを開きます: コマンドラインで `odbcad32` コマンドを実行します。
- 2 「システム DSN」タブを選択し、「追加」をクリックします。
- 3 適切な ODBC ドライバを選択し、「次へ」をクリックします。
- 4 「データ・ソース名」に「PROFITABILITY\_DS」を入力し、製品スキーマに接続するための接続の詳細を設定します。

## 高度な計算オプション

「計算の管理」画面で、管理者には「ドライバ操作タイプ」および「その他のプロセス・タイプ」タブが表示されますが、基本ユーザーは使用できません。

- [146 ページの「ドライバ操作タイプ」](#)
- [147 ページの「その他のプロセス・タイプ」](#)

---

**注意** これらのオプションは、経験豊富なデータベース管理者のみが使用するようになっています。「ドライバ操作タイプ」または「その他のプロセス・タイプ」のタイプを変更すると、アプリケーションに大きく影響したり、モデルまたはデータを破損したりする場合があります。

---

## ドライバ操作タイプ

「計算の管理」画面の「ドライバ操作タイプ」タブには、詳細 Profitability の現在のドライバ操作タイプが表示されます。

---

**注意** このタブでは新しいドライバ操作タイプを作成しないでください。このタブは、経験豊富なデータベース管理者のみが使用するようになっています。変更すると、アプリケーションに大きく影響したり、モデルまたはデータを破損したりする場合があります。

---

「ドライバ操作タイプ」は、サポートされるドライバ操作を拡張するための管理機能です。パフォーマンスの問題や一意ドライバの課題を解決することができます。これを使用するには、高度な SQL の知識があり、データベース管理や、Profitability and Cost Management の詳細 Profitability アプリケーションのドキュメント化されていない機能について深く理解していることが必要です。カスタマ・サービスの

リクエストに応じて Oracle から求められないかぎりこの機能は使用しないでください。

## その他のプロセス・タイプ

このリリースの詳細 Profitability では、2 つの代替計算プロセス・タイプを使用できます。

- Oracle Database 11g (デフォルト)
- Oracle Database 10g

Oracle Database 10g を使用するように構成されているシステムの場合は、「その他のプロセス・タイプ」タブを使用して、データベースに適した計算プロセス・タイプを選択します。

---

**注意** 「その他のプロセス・タイプ」タブは、経験豊富なデータベース管理者のみが使用するようにしてください。「その他のプロセス・タイプ」ではその他の変更を行わないようにしてください。アプリケーションに大きく影響したり、モデルまたはデータを破損したりする場合があります。


---

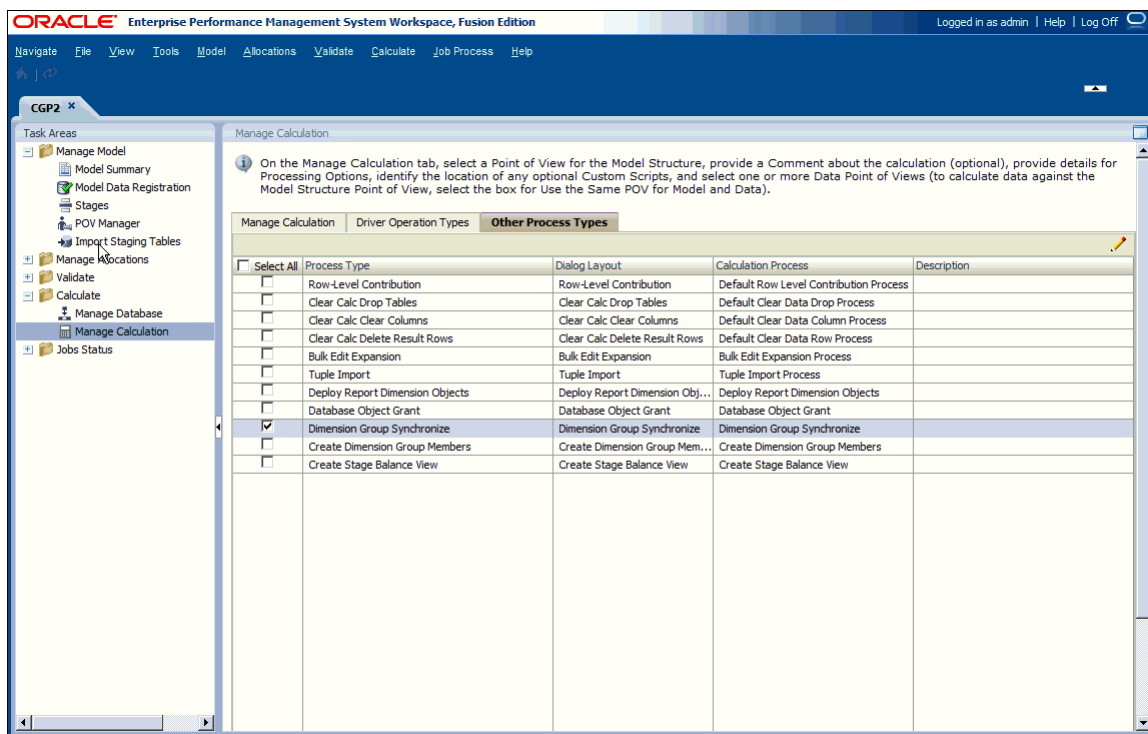
詳細 Profitability で使用している Oracle Database のバージョンに応じ、この画面を使用してデータベースを正しいバージョンに変更してください。

▶ ご使用の Oracle Database のバージョンを変更するには:

- 1 EPM Workspace で、「ナビゲート」、「アプリケーション」、「Profitability」の順に選択し、Oracle Database のバージョンを変更するアプリケーションを選択します。
- 2 「タスク領域」から、「計算」、「計算の管理」の順に選択し、「その他のプロセス・タイプ」タブを選択します。

**注:** このタブが表示されるのは、管理者ユーザーの役割が割り当てられている場合のみです。

- 3 「次元グループの同期」を選択し、「編集」ボタン  をクリックします。  
「その他のプロセス・タイプの定義」ダイアログ・ボックスが表示されます。



- 4 「計算プロセス」で、使用している Oracle Database の適切なオプションを選択します。
  - 次元グループの同期(Oracle 11g)
  - 次元グループの同期 10G (Oracle 10g)
- 5 「OK」をクリックして変更を保存します。

## この章の内容

詳細 Profitability のインポート・データベース・テーブルの作成 .....	150
HPM_STGD_POV .....	150
HPM_STGD_DRIVER .....	151
HPM_STGD_DRIVER_SEL .....	153
HPM_STGD_DRIVER_EXCEP .....	154
HPM_STGD_ASGN_RULE_SEL .....	155
HPM_STGD_CALCRULE_SNGLSRC .....	156
HPM_STGD_CALCRULE_CALCMSRS .....	157
HPM_STGD_CALCRULE_MULTISRC .....	159

リレーショナル・データベースから Profitability and Cost Management にモデル・データをインポートするには、Profitability and Cost Management 製品スキーマとは別のデータベース・スキーマに一連のステーjingング・テーブルを作成する必要があります。(詳細 Profitability のモデル・データ・スキーマとして使用されるスキーマを再利用することもできます)。これらのステーjingング・テーブルに、インポートされるアプリケーション・アーティファクトの詳細を移入します。

**注意** 製品スキーマを変更しないでください。

MS SQL Server および Oracle Database では、ステーjingング・データベース・スクリプトを使用できます。適したスクリプトを使用して、新しいデータベースにステーjingング・テーブルを作成します:

この付録のスキーマ・テーブルを使用してプロシージャとステーjingング・テーブルを作成します:

- 150 ページの「詳細 Profitability のインポート・データベース・テーブルの作成」
- 150 ページの「HPM\_STGD\_POV」
- 151 ページの「HPM\_STGD\_DRIVER」
- 153 ページの「HPM\_STGD\_DRIVER\_SEL」
- 154 ページの「HPM\_STGD\_DRIVER\_EXCEP」
- 155 ページの「HPM\_STGD\_ASGN\_RULE\_SEL」
- 156 ページの「HPM\_STGD\_CALCRULE\_SNGLSRC」

- 157 ページの「[HPM\\_STGD\\_CALCRULE\\_CALCMSRS](#)」
- 159 ページの「[HPM\\_STGD\\_CALCRULE\\_MULTISRC](#)」

## 詳細 Profitability のインポート・データベース・テーブルの作成

MS SQL Server および Oracle Database では、ステージング・データベース・スクリプトを使用できます。詳細 Profitability でテーブルを作成するには、`create_dp_staging.sql` スクリプトを使用します。

▶ ステージング・テーブルを作成するには:

- 1 製品スキーマ外に、新しい Oracle データベース・スキーマまたは MS SQL Server データベース・スキーマを作成します。
- 2 データベースのタイプに応じて、デフォルトの場所で `create_dp_staging.sql` を検索します:
  - `%EPM_ORACLE_HOME%/products/Profitability/database/Common/MSSQLServer`
  - `%EPM_ORACLE_HOME%/products/Profitability/database/Common/Oracle`
- 3 `create_dp_staging.sql` スクリプトを実行します。

## HPM\_STGD\_POV

HPM\_STGD\_POV テーブルには、視点(POV)に含まれる次元メンバーの各組合せの状態が保管されます。

表 31 HPM\_STGD\_POV

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	integer(38,0)	はい	一意のレコード ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した POV の最初の次元メンバーの名前
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 2 番目の次元メンバーの名前
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 3 番目の次元メンバーの名前
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 4 番目の次元メンバーの名前

フィールド	SQL Server データ型	Oracle データ型	必須	説明
pov_state	nvarchar (80)	varchar2(80 CHAR)	はい	POV の現在のステータス: <ul style="list-style-type: none"> <li>● ドラフト</li> <li>● 発行済</li> <li>● アーカイブ済</li> </ul>
last_upload_date	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻
exception	nvarchar (255)	varchar2(255 CHAR)		このテーブルのインポート時に発生したエラーを詳述するメッセージ。
created_userid	nvarchar (32)	varchar2(32 CHAR)	更新時にインポート・プログラムによって移入されま す	最後のインポートを開始したユーザーの ID
created_timestamp	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻
modified_userid	nvarchar (32)	varchar2(32 CHAR)	更新時にインポート・プログラムによって移入されま す	最後のインポートを変更したユーザーの ID
modified_timestamp	timestamp (6)	timestamp (6)		変更されたインポートの日付と時刻

## HPM\_STGD\_DRIVER

HPM\_STGD\_DRIVER テーブルは、ドライバ・タイプ、表示順序および式を含む、ドライバに関する詳細を示します。

表 32 HPM\_STGD\_DRIVER

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	integer(38,0)	はい	一意のレコード ID
name	nvarchar (80)	varchar2(80 CHAR)	はい	選択したドライバの名前
description	nvarchar (255)	varchar2(255 CHAR)		選択したドライバの目的の説明
式	nvarchar (500)	varchar2(500 CHAR)	はい	ユーザーによって作成されたドライバの式。この式は SQL 構文と、数学的に正しい演算を使用して作成する必要があります。

フィールド	SQL Server データ型	Oracle データ型	必須	説明
allow_idle_ cost	nvarchar (1)	varchar2 (1 CHAR)		このドライバでアイドル費用を使用可能にするかどうかを決定する適切な値を入力します:  <ul style="list-style-type: none"> <li>このドライバでアイドル費用を使用可能にする場合、「Y」(はい)を入力します。</li> <li>このドライバでアイドル費用を使用不可能にする場合、「N」(いいえ)を入力します。このオプションがデフォルトです。</li> </ul>
priority	integer(38,0)	integer(38,0)	はい	ステージ内の配賦を指定した順序で実行できるようドライバの計算の優先度を入力します。  優先度が最低のドライバが最初に処理されます。デフォルトでは、優先度は 100 に設定されており、最も高い優先度は 1 です。数値は連続していてもかまいません。  優先度が同じドライバは特定の順序で実行されるわけではありません。正の整数のみが有効です。
volume_ formula	nvarchar (4000)	varchar2(4000 CHAR)		式にレートとボリューム式の両方が定義されている、レートベースのドライバの場合のみ。
dst_ measure_ member_ name	nvarchar (80)	varchar2(80 CHAR)	はい	宛先テーブルにおける次元の名前
tdv_ measure_ member_ name	nvarchar (80)	varchar2(80 CHAR)		配賦式 Driver Value/TotalDriverValue (DV/TDV)で分母として使用されるメジャー
operation_ type	nvarchar (80)	varchar2(80 CHAR)	はい	ドライバ用に選択された操作のタイプ:  <ul style="list-style-type: none"> <li>レートベースの配賦</li> <li>比率ベースの配賦計算済メジャー</li> <li>比率ベースの配賦</li> <li>計算済メジャー</li> </ul>
last_upload_ date	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻
exception	nvarchar (255)	varchar2(255 CHAR)		このテーブルのインポート時に発生したエラーを詳述するメッセージ。
created_ userid	nvarchar (32)	varchar2(32 CHAR)	更新時に インポート・プログラムによって移入され ます	最後のインポートを開始したユーザーの ID
created_ timestamp	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻



フィールド	SQL Server データ型	Oracle データ型	必須	説明
modified_userid	nvarchar (32)	varchar2(32 CHAR)	更新時に インポート・プログラムによって移入されます	最後のインポートを変更したユーザーの ID
modified_timestamp	timestamp (6)	timestamp (6)		変更されたインポートの日付と時刻

## HPM\_STGD\_DRIVER\_SEL

HPM\_STGD\_DRIVER\_SEL テーブルには、選択したドライバ・ルールについて、POV とソース・ステージ・ドライバ次元メンバーに関する詳細が示されます。

表 33 HPM\_STGD\_DRIVER\_SEL

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	integer(38,0)	はい	一意のレコード ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した POV の最初の次元メンバーの名前
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 2 番目の次元メンバーの名前(該当する場合)
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 3 番目の次元メンバーの名前(該当する場合)
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 4 番目の次元メンバーの名前(該当する場合)
driver_dim_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択したドライバ次元のメンバーの名前
driver_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択したドライバの名前
last_upload_date	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻
import_exception	nvarchar (255)	varchar2(255 CHAR)		このテーブルのインポート時に発生したエラーを詳述するメッセージ。
created_userid	nvarchar (32)	varchar2(32 CHAR)	更新時にインポート・プログラムによって移入されます	最後のインポートを開始したユーザーの ID
created_timestamp	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻

フィールド	SQL Server データ型	Oracle データ型	必須	説明
modified_userid	nvarchar (32)	varchar2(32 CHAR)	更新時にインポート・プログラムによって移入され ます	最後のインポートを変更したユーザーの ID
modified_timestamp	timestamp (6)	timestamp (6)		変更されたインポートの日付と時刻

## HPM\_STGD\_DRIVER\_EXCEP

HPM\_STGD\_DRIVER\_EXCEP テーブルには、選択したドライバ例外について、POV とソース・ステージ交差に関する詳細が示されます。

表 34 HPM\_STGD\_DRIVER\_EXCEP

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	integer(38,0)	はい	一意のレコード ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した POV の最初の次元メンバーの名前
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 2 番目の次元メンバーの名前(該当する場合)
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 3 番目の次元メンバーの名前(該当する場合)
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 4 番目の次元メンバーの名前(該当する場合)
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	ソース・ステージの最初の次元メンバーの名前
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		ソース・ステージの 2 番目の次元メンバーの名前(該当する場合)
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		ソース・ステージの 3 番目の次元メンバーの名前(該当する場合)
src_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		ソース・ステージの 4 番目の次元メンバーの名前(該当する場合)
src_dim5_member_name	nvarchar (80)	varchar2(80 CHAR)		ソース・ステージの 5 番目の次元メンバーの名前(該当する場合)
driver_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択したドライバの名前
last_upload_date	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻
import_exception	nvarchar (255)	varchar2(255 CHAR)		このテーブルのインポート時に発生したエラーを詳述するメッセージ。

フィールド	SQL Server データ型	Oracle データ型	必須	説明
created_userid	nvarchar (32)	varchar2(32 CHAR)	更新時にインポート・プログラムによって移入されます	最後のインポートを開始したユーザーの ID
created_timestamp	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻
modified_userid	nvarchar (32)	varchar2(32 CHAR)	更新時にインポート・プログラムによって移入されます	最後のインポートを変更したユーザーの ID
modified_timestamp	timestamp (6)	timestamp (6)		変更されたインポートの日付と時刻

## HPM\_STGD\_ASGN\_RULE\_SEL

HPM\_STGD\_ASGN\_RULE\_SEL テーブルには、割当てルールの選択に関する詳細が示されます。

表 35 HPM\_STGD\_ASGN\_RULE\_SEL

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	integer(38,0)	はい	一意のレコード ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した POV の最初の次元メンバーの名前
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 2 番目の次元メンバーの名前
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 3 番目の次元メンバーの名前
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 4 番目の次元メンバーの名前
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	ソース・ステージの最初の次元メンバーの名前
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		ソース・ステージの 2 番目の次元メンバーの名前(該当する場合)
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		ソース・ステージの 3 番目の次元メンバーの名前(該当する場合)
src_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		ソース・ステージの 4 番目の次元メンバーの名前(該当する場合)

フィールド	SQL Server データ型	Oracle データ型	必須	説明
src_dim5_ member_name	nvarchar (80)	varchar2(80 CHAR)		ソース・ステージの 5 番目の次元メンバーの名前(該当する場合)
rule_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した割当てを制御するルールの名前。このルールは、ターゲット・データベース内に存在する必要があります。
last_upload_date	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻
import_exception	nvarchar (255)	varchar2(255 CHAR)		このテーブルのインポート時に発生したエラーを詳述するメッセージ。
created_userid	nvarchar (32)	varchar2(32 CHAR)	更新時にインポート・プログラムによって移入されます	最後のインポートを開始したユーザーの ID
created_ timestamp	timestamp (6)	timestamp (6)		最後のインポートの日付と時刻
modified_userid	nvarchar (32)	varchar2(32 CHAR)	更新時にインポート・プログラムによって移入されます	最後のインポートを変更したユーザーの ID
modified_ timestamp	timestamp (6)	timestamp (6)		変更されたインポートの日付と時刻

## HPM\_STGD\_CALCRLULE\_SNGLSRC

HPM\_STGD\_CALCRLULE\_SNGLSRC テーブルは、単一ソース計算ルールの作成または更新の詳細を提供します。

表 36 HPM\_STGD\_CALCRLULE\_SNGLSRC

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	NUMBER(38,0)	はい	各行の一意識別子。任意のシーケンス・ジェネレータの値を使用できます。
name	nvarchar (80)	VARCHAR2(80 CHAR)	はい	計算ルールの名前。
description	nvarchar (255)	varchar2(255 CHAR)		計算ルールの詳しい説明。
pov_dim1_ member_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した POV の最初の次元メンバーの名前
pov_dim2_ member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 2 番目の次元メンバーの名前

フィールド	SQL Server データ型	Oracle データ型	必須	説明
pov_dim3_ member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 3 番目の次元メンバーの名前
pov_dim4_ member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 4 番目の次元メンバーの名前
calculation_ sequence	identity	NUMBER(4,0)	はい	計算の際に計算ルールが処理される相対順序を指定する数値。これは 0 よりも大きくする必要があり、そうでない場合は挿入時にチェック制約エラーが発生します。
tag	nvarchar (255)	VARCHAR2(255)		この列に計算ルールの検索タグを指定します。
enabled_flag	nvarchar (1)	VARCHAR2(1 CHAR)	はい	
first_sequence_ priority	identity	NUMBER(38,0)	はい	このルールで実行されるドライバ範囲の最初のドライバ・シーケンス優先度を指定します。これは 0 よりも大きくする必要があり、そうでない場合は挿入時にチェック制約エラーが発生します。
last_sequence_ priority	identity	NUMBER(38,0)	はい	このルールで実行されるドライバ範囲の最後のドライバ・シーケンス優先度を指定します。これは FIRST_SEQUENCE_PRIORITY 以上である必要があり、そうでない場合は挿入時にチェック制約エラーが発生します。
all_drivers_flag	nvarchar (1)	VARCHAR2(1 CHAR)	はい	Y が計算の際に最初のシーケンス優先度フィールドおよび最後のシーケンス優先度フィールドの値を無視する場合。
last_upload_ date	timestamp	timestamp		この列はインポートによって入力され、インポートが最後に実行された日付を示します。
import_ exception	nvarchar (255)	varchar2(255 CHAR)		行のインポートに問題があった場合のエラー・コードが含まれます。NULL は、インポートが正常に完了したことを意味します。
created_userid	nvarchar (32)	varchar2(32 CHAR)		このテーブルのこの行を最初に作成したユーザーを識別します。これは HPCM では使用されません。
created_ timestamp	timestamp (6)	timestamp		この行が最初に作成された日時を識別します。これは HPCM では使用されません。
modified_userid	nvarchar (32)	varchar2(32 CHAR)		このテーブルのこの行を最初に作成したユーザーを識別します。これは HPCM では使用されません。
modified_ timestamp	timestamp (6)	timestamp		この行が最初に作成された日時を識別します。これは HPCM では使用されません。

## HPM\_STGD\_CALCRULE\_CALCMSRS

HPM\_STGD\_CALCRULE\_CALCMSRS テーブルは、計算済メジャー・ルールの作成または更新の詳細を提供します。

表 37 HPM\_STGD\_CALCRULE\_CALCMSRS

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	number(38,0)	はい	各行の一意識別子。任意のシーケンス・ジェネレータの値を使用できます。
name	nvarchar (80)	varchar2(80 CHAR)	はい	計算ルールの名前。
description	nvarchar (255)	varchar2(255 CHAR)		計算ルールの詳しい説明。
pov_dim1_ member_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した POV の最初の次元メンバーの名前
pov_dim2_ member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 2 番目の次元メンバーの名前
pov_dim3_ member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 3 番目の次元メンバーの名前
pov_dim4_ member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 4 番目の次元メンバーの名前
calculation_ sequence	number	number(4,0)	はい	計算の際に計算ルールが処理される相対順序を指定する数値。これは 0 よりも大きくする必要があり、そうでない場合は挿入時にチェック制約エラーが発生します。
tag	nvarchar (255)	varchar2(255)		この列に計算ルールの検索タグを指定します。
enabled_flag	nvarchar (1)	varchar2(1 CHAR)	はい	
driver_name1	nvarchar (1)	varchar2(1 CHAR)	はい	この計算ルールに関連付けるドライバ名。
driver_name2	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付けるドライバ名。
driver_name3	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付けるドライバ名。
driver_name4	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付けるドライバ名。
driver_name5	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付けるドライバ名。
dest_assign_rule_ name1	nvarchar (80)	varchar2(80 CHAR)	はい	この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_ name2	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_ name3	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_ name4	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_ name5	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付ける宛先割当てルール。
last_upload_date	timestamp (6)	timestamp		この列はインポートによって入力され、インポートが最後に実行された日付を示します。

フィールド	SQL Server データ型	Oracle データ型	必須	説明
import_exception	nvarchar (255)	varchar2(255 CHAR)		行のインポートに問題があった場合のエラー・コードが含まれます。NULL は、インポートが正常に完了したことを意味します。
created_userid	nvarchar (32)	varchar2(32 CHAR)		このテーブルのこの行を最初に作成したユーザーを識別します。これは HPCM では使用されません。
created_timestamp	timestamp (6)	timestamp		この行が最初に作成された日時を識別します。これは HPCM では使用されません。
modified_userid	nvarchar (32)	varchar2(32 CHAR)		このテーブルのこの行を最初に作成したユーザーを識別します。これは HPCM では使用されません。
modified_timestamp	timestamp (6)	timestamp		この行が最初に作成された日時を識別します。これは HPCM では使用されません。

## HPM\_STGD\_CALCRLU\_MULTISRC

HPM\_STGD\_CALCRLU\_MULTISRC テーブルは、複数ソース計算ルールの作成または更新の詳細を提供します。

表 38 HPM\_STGD\_CALCRLU\_CALCMSRS

フィールド	SQL Server データ型	Oracle データ型	必須	説明
id	identity	number(38,0)	はい	各行の一意識別子。任意のシーケンス・ジェネレータの値を使用できます。
name	nvarchar (80)	varchar2(80 CHAR)	はい	計算ルールの名前。
description	nvarchar (255)	varchar2(255 CHAR)		計算ルールの詳しい説明。
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	はい	選択した POV の最初の次元メンバーの名前
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 2 番目の次元メンバーの名前
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 3 番目の次元メンバーの名前
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		選択した POV の 4 番目の次元メンバーの名前
calculation_sequence	number	number(4,0)	はい	計算の際に計算ルールが処理される相対順序を指定する数値。これは 0 よりも大きくする必要があり、そうでない場合は挿入時にチェック制約エラーが発生します。
tag	nvarchar (255)	varchar2(255)		この列に計算ルールの検索タグを指定します。
enabled_flag	nvarchar (1)	varchar2(1 CHAR)	はい	

フィールド	SQL Server データ型	Oracle データ型	必須	説明
driver_name	nvarchar (80)	varchar2(80 CHAR)	はい	この計算ルールに関連付けるドライバ名。
src_assign_rule_name1	nvarchar (80)	varchar2(80 CHAR)	はい	この計算ルールに関連付けるソース割当てルール。
src_assign_rule_name2	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付けるソース割当てルール。
src_assign_rule_name3	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付けるソース割当てルール。
src_assign_rule_name4	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付けるソース割当てルール。
src_assign_rule_name5	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付けるソース割当てルール。
dest_assign_rule_name1	nvarchar (80)	varchar2(80 CHAR)	はい	この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_name2	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_name3	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_name4	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_name5	nvarchar (80)	varchar2(80 CHAR)		この計算ルールに関連付ける宛先割当てルール。
last_upload_date	timestamp (6)	timestamp		この列はインポートによって入力され、インポートが最後に実行された日付を示します。
import_exception	nvarchar (255)	VARCHAR2(255 CHAR)		行のインポートに問題があった場合のエラー・コードが含まれます。NULL は、インポートが正常に完了したことを意味します。
created_userid	nvarchar (32)	varchar2(32 CHAR)		このテーブルのこの行を最初に作成したユーザーを識別します。これは HPCM では使用されません。
created_timestamp	timestamp (6)	timestamp		この行が最初に作成された日時を識別します。これは HPCM では使用されません。
modified_userid	nvarchar (32)	varchar2(32 CHAR)		このテーブルのこの行を最初に作成したユーザーを識別します。これは HPCM では使用されません。
modified_timestamp	timestamp (6)	timestamp		この行が最初に作成された日時を識別します。これは HPCM では使用されません。



# 16

## 詳細Profitabilityのモデル定義 データのエクスポート

### この章の内容

HPM_EXPD_STAGE .....	161
HPM_EXPD_POV .....	163
HPM_EXPD_DRIVER .....	163
HPM_EXPD_DRIVER_SEL .....	164
HPM_EXPD_DRIVER_EXCEP .....	165
HPM_EXPD_ASGN_RUL_SEL .....	166
HPM_EXPD_CALCRULE_SNGLSRC .....	167
HPM_EXPD_CALCRULE_CALCMSRS .....	168
HPM_EXPD_CALCRULE_MULTISRC .....	169
アプリケーションの統計の生成 .....	170

モデルを作成したら、データベース・ビューの出力として、モデル・アーティファクトを表示するようデータベースに問い合わせできます。

管理者は、ステージング・テーブルで使用される列をミラー化するデータベース・ビューをシステム・データベース内に作成できます。このビューは、システムに保管されるモデル・データを示します。

## HPM\_EXPD\_STAGE

HPM\_EXPD\_STAGE ビューは、ステージの名前と順序、アプリケーション名、ソース・ステージおよび宛先ステージの次元を取得します。

表 39 HPM\_EXPD\_STAGE

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar (80)	varchar2(80 CHAR)	選択したアプリケーションの名前
id	identity	integer(38,0)	Profitability and Cost Management データベースで使用される ステージ ID
name	nvarchar (80)	varchar2(80 CHAR)	選択したソース・ステージの名前
description	nvarchar (255)	varchar2(255 CHAR)	選択したステージの目的の説明

フィールド	SQL Server データ型	Oracle データ型	説明
stage_order	integer	number (38,0)	計算スクリプトの生成および計算において、モデリング(割当ておよびルール作成)時に選択したステージがモデル内で使用されるシーケンス位置。
stage_prefix	nvarchar (80)	varchar2(80 CHAR)	ステージ名を区別するための接頭辞。
driver_dim_name	nvarchar (80)	varchar2(80 CHAR)	選択したドライバ次元の名前
is_intra	nvarchar (1)	varchar2(1 CHAR)	ステージ内割当てが許可されているかどうかを決定するフラグ: <ul style="list-style-type: none"> <li>● 「Y」(はい)は、モデル・ステージ内でステージ内割当てを使用できることを示します。</li> <li>● 「N」(いいえ)は、モデル・ステージ内でステージ内割当てを使用できないことを示します。</li> </ul>
dim1_name	nvarchar (80)	varchar2(80 CHAR)	ステージの最初の次元の名前
dim2_name	nvarchar (80)	varchar2(80 CHAR)	ステージに含まれる 2 番目の次元の名前(該当する場合)。
dim3_name	nvarchar (80)	varchar2(80 CHAR)	ステージに含まれる 3 番目の次元の名前(該当する場合)。
dim4_name	nvarchar (80)	varchar2(80 CHAR)	ステージに含まれる 4 番目の次元の名前(該当する場合)。
dim5_name	nvarchar (80)	varchar2(80 CHAR)	ステージに含まれる 5 番目の次元の名前(該当する場合)。
dim6_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim7_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim8_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim9_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim10_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim11_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim12_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim13_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim14_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim15_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim16_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim17_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim18_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim19_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim20_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim21_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前

フィールド	SQL Server データ型	Oracle データ型	説明
dim22_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim23_name	nvarchar (80)	wvarchar2(80 CHAR)	宛先ステージの次元の名前
dim24_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前
dim25_name	nvarchar (80)	varchar2(80 CHAR)	宛先ステージの次元の名前

## HPM\_EXPDP\_OV

HPM\_EXPDP\_OV ビューは、視点(POV)に含まれる次元メンバーの各組合せの状態を取得します。

表 40 HPM\_EXPDP\_OV

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar (80)	varchar2(80 CHAR)	選択したアプリケーションの名前
id	identity	integer(38,0)	一意のレコード ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の最初の次元メンバーの名前
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 2 番目の次元メンバーの名前
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 3 番目の次元メンバーの名前
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 4 番目の次元メンバーの名前
pov_state	nvarchar (80)	varchar2(80 CHAR)	POV の現在のステータス: <ul style="list-style-type: none"> <li>● ドラフト</li> <li>● 発行済</li> <li>● アーカイブ済</li> </ul>

## HPM\_EXPDP\_DRIVER

HPM\_STGD\_DRIVER ビューは、ドライバ・タイプ、表示順序および式を含む、ドライバに関する詳細を取得します。

表 41 HPM\_EXPDP\_DRIVER

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar (80)	varchar2(80 CHAR)	選択したアプリケーションの名前
id	identity	integer(38,0)	一意のレコード ID

フィールド	SQL Server データ型	Oracle データ型	説明
name	nvarchar (80)	varchar2(80 CHAR)	選択したドライバの名前
description	nvarchar (255)	varchar2 (255 CHAR)	選択したドライバの目的の説明
display_order	integer(38,0)	integer(38,0)	モデル内のすべてのドライバのリスト内のドライバの表示位置
式	nvarchar (4000)	varchar2 (4000 CHAR)	ユーザーによって作成されたドライバの式。この式は SQL 構文と、数学的に正しい演算を使用して作成する必要があります。
allow_idle_cost	nvarchar (1)	varchar2 (1 CHAR)	このドライバにアイドル費用が許可されているかどうかを決定するフラグ: <ul style="list-style-type: none"> <li>● 「Y」 (はい)は、このドライバにアイドル費用が許可されていることを示します。</li> <li>● 「N」 (いいえ)は、このドライバにアイドル費用が許可されないことを示します。</li> </ul>
priority	integer	number (38,0)	ステージ内の配賦を指定した順序で実行できるようドライバの計算の優先度を入力します。  優先度が最低のドライバが最初に処理されます。デフォルトでは、優先度は 100 に設定されており、最も高い優先度は 1 です。数値は連続していなくてもかまいません。  優先度が同じドライバは特定の順序で実行されるわけではありません。正の整数のみが有効です。
volume_ formula	nvarchar (4000)	varchar2(4000 CHAR)	レートベースのドライバの場合のみ、レートとボリューム式の両方が定義されている式。
dst_measure_ member_name	nvarchar (2000)	varchar2(2000 CHAR)	宛先テーブルにおける次元の名前
tdv_measure_ member_name	nvarchar (80)	varchar2(2000 CHAR)	配賦式 Driver Value/TotalDriverValue (DV/TDV)で分母として使用されるメジャー
operation_type	nvarchar (80)	varchar2(80 CHAR)	ドライバ用に選択された操作のタイプ: <ul style="list-style-type: none"> <li>● レートベースの配賦</li> <li>● 比率ベースの配賦</li> <li>● 計算済メジャー</li> </ul>

## HPM\_EXPD\_DRIVER\_SEL

HPM\_EXPD\_DRIVER\_SEL ビューは、POV に関する詳細とドライバ選択を取得します。

表 42 HPM\_EXPD\_DRIVER\_SEL

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar (80)	varchar2(80 CHAR)	選択したアプリケーションの名前
id	identity	integer(38,0)	一意のレコード ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の最初の次元メンバーの名前
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 2 番目の次元メンバーの名前(該当する場合)
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 3 番目の次元メンバーの名前(該当する場合)
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 4 番目の次元メンバーの名前(該当する場合)
driver_dim_member_name	nvarchar (80)	varchar2(80 CHAR)	選択したドライバ次元の名前
driver_name	nvarchar (80)	varchar2(80 CHAR)	選択したドライバの名前

## HPM\_EXPD\_DRIVER\_EXCEP

HPM\_EXPD\_DRIVER\_EXCEP ビューは、選択したドライバ例外について、POV に関する詳細とソース・ステージを取得します。

表 43 HPM\_EXPD\_DRIVER\_EXCEP

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar (80)	varchar2(80 CHAR)	選択したアプリケーションの名前
id	identity	integer(38,0)	一意のレコード ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の最初の次元メンバーの名前
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 2 番目の次元メンバーの名前(該当する場合)
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 3 番目の次元メンバーの名前(該当する場合)
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 4 番目の次元メンバーの名前(該当する場合)
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	ソース・ステージの最初の次元メンバーの名前
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	ソース・ステージの 2 番目の次元メンバーの名前(該当する場合)

フィールド	SQL Server データ型	Oracle データ型	説明
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	ソース・ステージの 3 番目の次元メンバーの名前(該当する場合)
src_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	ソース・ステージの 4 番目の次元メンバーの名前(該当する場合)
src_dim5_member_name	nvarchar (80)	varchar2(80 CHAR)	ソース・ステージの 5 番目の次元メンバーの名前(該当する場合)
driver_name	nvarchar (80)	varchar2(80 CHAR)	選択したドライバの名前

## HPM\_EXPD\_ASGN\_RULE\_SEL

HPM\_EXPD\_ASGN\_RULE\_SEL ビューは、割当てルールの選択に関する詳細を取得します。

表 44 HPM\_EXPD\_ASGN\_RULE\_SEL

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar (80)	varchar2(80 CHAR)	選択したアプリケーションの名前
id	identity	integer(38,0)	一意のレコード ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の最初の次元メンバーの名前
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 2 番目の次元メンバーの名前
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 3 番目の次元メンバーの名前
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 4 番目の次元メンバーの名前
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	ソース・ステージの最初の次元メンバーの名前
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	ソース・ステージの 2 番目の次元メンバーの名前
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	ソース・ステージの 3 番目の次元メンバーの名前
src_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	ソース・ステージの 4 番目の次元メンバーの名前
src_dim5_member_name	nvarchar (80)	varchar2(80 CHAR)	ソース・ステージの 5 番目の次元メンバーの名前

フィールド	SQL Server データ型	Oracle データ型	説明
rule_name	nvarchar (80)	varchar2(80 CHAR)	選択した割当てを制御するルールの名前。このルールは、ターゲット・データベース内に存在する必要があります。

## HPM\_EXPD\_CALCRLU\_SNGLSRC

HPM\_EXPD\_CALCRLU\_SNGLSRC ビューは、単一ソース計算ルールの詳細を取得します。

表 45 HPM\_EXPD\_CALCRLU\_SNGLSRC

フィールド	SQL Server データ型	Oracle データ型	説明
application_name		VARCHAR2(80 CHAR)	選択したアプリケーションの名前
id	identity	NUMBER(38,0)	各行の一意識別子。任意のシーケンス・ジェネレータの値を使用できます。
name	nvarchar (80)	VARCHAR2(80 CHAR)	計算ルールの名前。
description	nvarchar (255)	varchar2(255 CHAR)	計算ルールの詳しい説明。
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の最初の次元メンバーの名前
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 2 番目の次元メンバーの名前
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 3 番目の次元メンバーの名前
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 4 番目の次元メンバーの名前
calculation_sequence	integer	NUMBER(4,0)	計算の際に計算ルールが処理される相対順序を指定する数値。これは 0 よりも大きくする必要があり、そうでない場合は挿入時にチェック制約エラーが発生します。
tag	nvarchar (255)	VARCHAR2(255)	この列に計算ルールの検索タグを指定します。
enabled_flag	nvarchar (1)	VARCHAR2(1 CHAR)	
first_sequence_priority	integer	NUMBER(38,0)	このルールで実行されるドライバ範囲の最初のドライバ・シーケンス優先度を指定します。これは 0 よりも大きくする必要があり、そうでない場合は挿入時にチェック制約エラーが発生します。

フィールド	SQL Server データ型	Oracle データ型	説明
last_sequence_ priority	integer	NUMBER(38,0)	このルールで実行されるドライバ範囲の最後のドライバ・シーケンス優先度を指定します。これは FIRST_SEQUENCE_PRIORITY 以上である必要があり、そうでない場合は挿入時にチェック制約エラーが発生します。
all_drivers_flag	nvarchar (1)	VARCHAR2(1 CHAR)	Y が計算の際に最初のシーケンス優先度フィールドおよび最後のシーケンス優先度フィールドの値を無視する場合。

## HPM\_EXPD\_CALCRULE\_CALCMSRS

HPM\_EXPD\_CALCRULE\_CALCMSRS ビューは、計算済メジャー・ルールの詳細を取得します。

表 46 HPM\_EXPD\_CALCRULE\_CALCMSRS

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar (80)	VARCHAR2(80 CHAR)	選択したアプリケーションの名前
id	identity	NUMBER(38,0)	各行の一意識別子。任意のシーケンス・ジェネレータの値を使用できます。
name	nvarchar (80)	VARCHAR2(80 CHAR)	計算ルールの名前。
description	nvarchar (255)	varchar2(255 CHAR)	計算ルールの詳しい説明。
pov_dim1_member_ name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の最初の次元メンバーの名前
pov_dim2_member_ name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 2 番目の次元メンバーの名前
pov_dim3_member_ name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 3 番目の次元メンバーの名前
pov_dim4_member_ name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 4 番目の次元メンバーの名前
calculation_ sequence	integer	NUMBER(4,0)	計算の際に計算ルールが処理される相対順序を指定する数値。これは 0 よりも大きくする必要があり、そうでない場合は挿入時にチェック制約エラーが発生します。
tag	nvarchar (255)	VARCHAR2(255)	この列に計算ルールの検索タグを指定します。
enabled_flag	nvarchar (1)	VARCHAR2(1 CHAR)	
driver_name1	nvarchar (1)	VARCHAR2(1 CHAR)	この計算ルールに関連付けるドライバ名。
driver_name2	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付けるドライバ名。
driver_name3	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付けるドライバ名。



フィールド	SQL Server データ型	Oracle データ型	説明
driver_name4	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付けるドライバ名。
driver_name5	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付けるドライバ名。
dest_assign_rule_name1	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_name2	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_name3	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_name4	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_name5	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付ける宛先割当てルール。

## HPM\_EXPD\_CALCRULE\_MULTISRC

HPM\_EXPD\_MULTISRC ビューは、複数ソース計算ルールの詳細を取得します。

表 47 HPM\_EXPD\_CALCRULE\_MULTISRC

フィールド	SQL Server データ型	Oracle データ型	説明
application_name	nvarchar(80)	VARCHAR2(80 CHAR)	選択したアプリケーションの名前
id	identity	NUMBER(38,0)	各行の一意識別子。任意のシーケンス・ジェネレータの値を使用できます。
name	nvarchar (80)	VARCHAR2(80 CHAR)	計算ルールの名前。
description	nvarchar (255)	varchar2(255 CHAR)	計算ルールの詳しい説明。
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の最初の次元メンバーの名前
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 2 番目の次元メンバーの名前
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 3 番目の次元メンバーの名前
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	選択した POV の 4 番目の次元メンバーの名前
calculation_sequence	integer	NUMBER(4,0)	計算の際に計算ルールが処理される相対順序を指定する数値。これは 0 よりも大きくする必要があり、そうでない場合は挿入時にチェック制約エラーが発生します。

フィールド	SQL Server データ型	Oracle データ型	説明
tag	nvarchar (255)	VARCHAR2(255)	この列に計算ルールの検索タグを指定します。
enabled_flag	nvarchar (1)	VARCHAR2(1 CHAR)	
driver_name	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付けるドライバ名。
src_assign_rule_name1	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付けるソース割当てルール。
src_assign_rule_name2	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付けるソース割当てルール。
src_assign_rule_name3	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付けるソース割当てルール。
src_assign_rule_name4	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付けるソース割当てルール。
src_assign_rule_name5	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付けるソース割当てルール。
dest_assign_rule_name1	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_name2	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_name3	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_name4	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付ける宛先割当てルール。
dest_assign_rule_name5	nvarchar (80)	VARCHAR2(80 CHAR)	この計算ルールに関連付ける宛先割当てルール。

## アプリケーションの統計の生成

詳細アプリケーション・モデル計算統計スクリプトを詳細 Profitability アプリケーションと一緒に使用すると、モデルと、モデルの理解やデバッグに役立つモデル計算統計を生成できます。

このスクリプトは、HPCM 製品スキーマにテーブルを作成して統計を格納します。この統計は、モデルの計算結果の問題のトラブルシューティング、モデルの形と特徴の説明、およびモデルの計算パフォーマンスの評価に役立ちます。

dpmodecalcstats.sql スクリプトは次にあります:

```
%EPM_ORACLE_HOME%\OPatch\HPCM_11_1_2_2_351_14845052\files\products
\Profitability\database\Common\Oracle
```

(11.1.2.2.351 パッチがインストールされた後)。現在は Oracle データベース・インストールのみで使用できます。

このスクリプトは、SQL Developer 内で開いて実行するように設計されています。

▶ リクエストされたアプリケーションの統計を生成するには:

- 1 Oracle SQL Developer で `dpmodelcalcstats.sql` を開きます。
- 2 Oracle Fusion Performance Management 製品スキーマの接続を選択します。
- 3 スクリプトの 1 行目にカーソルを移動します。
- 4 「スクリプトの実行」をクリックするか、[F5]キーを押します。
- 5 モデル計算統計がリクエストされたアプリケーションの名前を入力します。

スクリプトによって、次の命名ルールを使用するテーブルが製品スキーマ内に作成されます:

HPMD\_<Application Name>\_DIAGSTAT\_<Unique Number>

HPMD\_BksDP20\_DIAGSTAT\_4129 テーブルの列の簡略な説明:

APPLICATION: アプリケーションの名前。

ITEM: 行の内容の簡略な説明。

RATIO\_DRV\_TOT: すべての比率ベース・ドライバに関連するアイテムの合計数。

RATE\_DRV\_TOT: すべてのレートベース・ドライバに関連するアイテムの合計数。

OBJC\_DRV\_TOT: すべてのオブジェクト計算ドライバに関連するアイテムの合計数。

TOTAL: すべての割当てに関連するアイテムの合計数。

NAME: アイテムに関連するアーティファクトまたはオブジェクトの名前。

次のタイプの診断統計アイテムがテーブルに含まれます:

「配賦テーブル:」 指定したアプリケーションの割当ての詳細をすべて含む配賦テーブルの名前。数が 1 よりも大きい場合、結果が間違っている可能性があります。計算の消去が実行されていないためです。

「割当て数:」 指定したアプリケーションのモデル内で識別された割当ての数。

「ソース・ノード数:」 モデル内で識別されたソース・ノード(交差)の数。

「データを含む割当て数:」 ソース・ステージ・ビジネス・オブジェクト内の 1 つ以上の行が潜在的なソース・ノードと一致する割当ての数。

「行更新の数:」 すべての割当てについて実行された Update 文によって変更された合計行数。

「割当てルールの数:」 少なくとも 1 つの割当てが割当てルールを参照する、モデル内に定義された割当てルール・アーティファクトの数。

「ドライバの数:」 少なくとも 1 つの割当てがドライバを参照する、モデル内に定義されたドライバ・アーティファクトの数。

「最大宛先行数:」 割当てによって宛先内で更新された行の最大数。

「宛先行数の中央値:」 割当てによって宛先内で更新された行数の中央値。標準

「宛先行数の偏差:」 割当てによって宛先内で更新された行数の標準偏差。

「ドライバ数:」 少なくとも1つの割当てによって参照されるドライバごとに1つのドライバ数アイテムがあります。このアイテムには、ドライバ・アーティファクトの名前と割当てによって参照される回数が含まれます。

「割当てルール数:」 少なくとも1つの割当てによって参照される割当てルールごとに1つの割当てルール数アイテムがあります。このアイテムには、割当てルール・アーティファクトの名前と割当てによって参照される回数が含まれます。



# Profitability and Cost Managementへのデータのインポート

## この付録の内容

ステージング・テーブルについて .....	174
インポート構成の作成 .....	175
インポート構成の変更 .....	177
インポート構成の削除 .....	178
インポート構成の実行 .....	178
インポートしたデータの確認 .....	179

Profitability and Cost Management にはデータを直接入力できます;ただし、データ入力には非常に時間がかかる場合があります。アプリケーションの移入を簡単に行うには、一連のインポート・ステージング・テーブルやインポート構成を使用してデータを Profitability and Cost Management に直接インポートできます。

モデル・データは複数のソースからインポートされます:

- モデル・メタデータおよび次元は、EPM Workspace を介して Oracle Hyperion EPM Architect の次元ライブラリからインポートされます。『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』のメタデータのインポートに関する項を参照してください。
- モデル定義データは、Profitability and Cost Management にインポートされます。このデータは、Excel などの別のアプリケーションで作成し、ステージング・テーブルを使用してインポートできます。
- モデルおよびエンドユーザー・データは Essbase から標準 Profitability アプリケーションに対してのみインポートおよびエクスポートが可能です。『Oracle Essbase データベース管理者ガイド』を参照してください。
- モデル・データおよびアプリケーションは、ライフサイクル管理を使用してインポートできます。Oracle Enterprise Performance Management System Lifecycle Management Guide を参照してください。

**注意** データをインポートする前に、EPM Workspace および Essbase でデータベースのバックアップ・ディレクトリを作成することをお勧めします。

インポート構成を作成し、インポートするテーブルおよびデータを指定する必要があります。この構成を保存することにより、複数回使用して同じデータ・セットをインポートできます。

# ステージング・テーブルについて

データが Profitability and Cost Management にインポートされると、ユーザーは、インポートを管理するために事前定義された構造を提供するステージング・テーブルを作成します。

リレーショナル・データベースから Profitability and Cost Management にモデル・データをインポートするには、Profitability and Cost Management データベース・テーブルが作成された場所から、別々のデータベース・スキーマに一連のステージング・テーブルを作成し、アプリケーションで使用できるように情報をフォーマットする必要があります。

---

**注意** 製品スキーマ内にインポート・ステージング・テーブルを作成しないでください。製品スキーマの変更はサポートされていないだけでなく、予測できない結果を招く場合があります。

---

これらのテーブルは、Oracle サーバーや SQL Server などのリレーショナル・データベースを使用して作成されます。これらを使用して、アプリケーションに簡単に対応できるフォーマットにデータを編成します。MS SQL Server および Oracle Database では、インストール後のインストール・フォルダにステージング・データベース・スクリプトが用意されています。デフォルトでは、この場所は %EPM\_ORACLE\_HOME%\products\Profitability\database\Common\ です。アプリケーション・タイプに適したスクリプトを使用して、新しいデータベースにステージング・テーブルを作成します:

- 標準 Profitability アプリケーションの場合は、create\_staging.sql スクリプトを使用します。
- 詳細 Profitability アプリケーションの場合は、create\_dp\_staging.sql スクリプトを使用します。

すべてのテーブルを同時に作成することも、インポートするテーブルのみを作成することもできます。次のデータ・グループのうち少なくとも 1 つを移入する必要があります:

- ステージ(標準 Profitability のみ)
- POV
- ドライバ
- ドライバの選択
- ドライバ例外
- 割当て(標準 Profitability のみ)
- ソース割当てルールに関連付け
- 計算ルール(詳細 Profitability のみ)

ステージング・テーブルは、Profitability and Cost Management 管理者(admin)により、これらのセクションで指定されているフォーマットを使用して作成されます。

- [第 6 章「標準 Profitability のインポート・ステージング・テーブル」](#)

- 第 15 章「詳細 Profitability のインポート・ステージング・テーブル」

データをインポートするには、適切なユーザーの役割およびセキュリティ権限が必要です。Oracle Enterprise Performance Management System User Security Administration Guide を参照してください。

## インポート構成の作成

データをインポートできるのは、次のすべての要素が使用可能である場合のみです:

- ソース・データベースおよび関連するステージング・テーブルとデータ
- データを受信するターゲット・アプリケーション
- インポートするテーブルおよびデータを定義するインポート構成。インポート構成は複数回使用できます。

インポートを合理化するために、異なる情報セットごとに個別の構成を作成することを検討してください。使用するインポート・グループを小さくすると、インポート時間を短縮できる他、静的情報を繰り返し更新しなくて済みます。たとえば、次のように、インポート構成のモデル要素をグループ化できます:

- まれに変更: POV およびステージ
- 頻繁に変更: ドライバ、ドライバの選択、ドライバ例外、割当て、割当てルールの選択、および計算ルール

構成が作成された後、データをアプリケーションにインポートする構成を実行します。

▶ インポート構成を作成する手順は、次のとおりです:

### 1 インポートのソースとして使用するデータベースを選択します。

インポート用としてフォーマットされている既存のデータベースをすることも、新しいブランク・データベースを作成することもできます。

データベースには命名規則はありません。

---

**注意** データまたはアーティファクトをインポートする前に、EPM Workspace および Essbase にデータベースのバックアップ・ディレクトリを作成することをお勧めします。支援が必要な場合は、管理者にお問い合わせください。

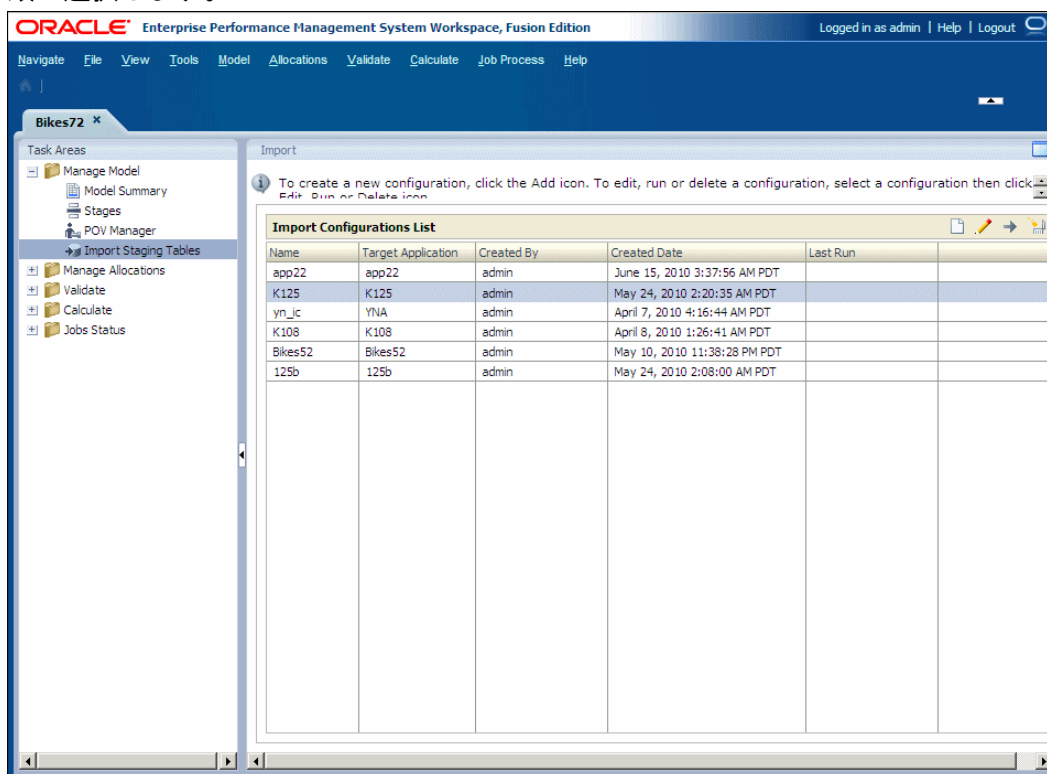
---


### 2 ソース・データベースに対して適切なスクリプトを実行し、ステージング・テーブルを自動的に作成します。

- 標準 Profitability アプリケーションの場合は、create\_staging.sql スクリプトを実行します。テーブルは第 6 章「標準 Profitability のインポート・ステージング・テーブル」で提供されたスキーマを使用して生成されます。
- 詳細 Profitability アプリケーションの場合は、create\_dp\_staging.sql スクリプトを実行します。テーブルは第 15 章「詳細 Profitability のインポー

ト・ステージング・テーブル」で提供されたスキーマを使用して生成されます。

- 3 モデル・データをソース・ステージング・テーブルにロードします。
- 4 ステージング・テーブルのエントリに明白な問題がないことを確認し、ソース・データベースから null 行を削除します。
- 5 ソース・データベースにアクセスできることを確認します。
- 6 EPM Workspace で、「ナビゲート」、「アプリケーション」、「Profitability」の順に選択し、Profitability and Cost Management モデルにアクセスするためのモデル名を選択します。
- 7 「タスク領域」で、「モデルの管理」、「ステージング・テーブルのインポート」の順に選択します。



- 8 「インポート」で、「新規インポート構成の追加」ボタン  をクリックして新しいインポート構成を作成します。

「データのインポート」ダイアログ・ボックスの手順 1 が表示されます。

- 9 「データ・ソースの詳細」で、ソース・ステージング・テーブルが存在するサーバーのアクセス詳細を入力します：
  - 「データベース・タイプ」で、ソース・データベースに使用されるデータベースのタイプを選択します。
  - 「ホスト」の下で、ソース・ステージング・テーブルが含まれるソース・データベースの場所へのパスを入力します。
  - 「ポート」の下で、ソース・データベースが関連付けられたポートの名前を入力します。
  - 「データベース名」の下で、ソース・データベースの名前を入力します。



- 10 必要に応じて、「ログイン情報」の下で、ソース・データベースの権限を持つユーザーの「ユーザー名」および「パスワード」を入力します。
- 11 「次へ」をクリックします。


「データのインポート」ダイアログ・ボックスの手順2が表示されます。
- 12 「構成」の下で、「ターゲット・アプリケーション」を選択します。

ターゲット・アプリケーションは、データのインポート先の Profitability and Cost Management アプリケーションです。
- 13 「ステージング・テーブルの詳細」の下で、インポートする1つまたは複数のステージング・テーブルを選択します。

既存のステージング・テーブルの名前が「テーブル名」の下にリストされ、各ステージング・テーブルのデータのタイプが「テーブル・タイプ」の下に表示されます。
- 14 「次」をクリックします。

「データのインポート」ダイアログ・ボックスの手順3が表示されます。
- 15 「構成の要約」の下で、このインポート構成を保管する「構成名」を入力します。


この構成に関連付けられたデータベース名が構成名の下に表示されます。構成名は80文字以上にすることはできません。
- 16 「終了」をクリックします。

インポート構成が「構成リストのインポート」に追加されます。これを再使用することにより、同じデータ・セットを何回もインポートできます。
- 17 **オプション:** 「構成リストのインポート」からインポート構成を選択し、「インポート構成の実行」ボタン  をクリックして、構成をインポートします。178 ページの「インポート構成の実行」を参照してください。

## インポート構成の変更

インポート構成を更新する際、ターゲット・アプリケーション、一連のステージング・テーブル、およびインポート構成名を変更できます。

- ▶ インポート構成を変更するには:
- 1 「タスク領域」で、「モデルの管理」、「ステージング・テーブルのインポート」の順に選択します。

インポート・ページが表示されます。
  - 2 「構成リストのインポート」で、変更するインポート構成を選択します。
  - 3 「インポート構成の編集」ボタン  をクリックします。

「データのインポート」ダイアログ・ボックスが表示されます。
  - 4 選択したインポート構成の接続情報およびログイン情報を確認し、「次へ」をクリックします。

- 5 **オプション:** 「ターゲット・アプリケーション」の下で、このインポート構成のターゲット・アプリケーションを選択します。
- 6 **オプション:** 「ステージング・テーブルの詳細」で、インポートするステージング・テーブルのセットを変更します。
- 7 「次へ」をクリックします。
- 8 **オプション:** 「構成の要約」の下で、変更した構成を保管する別々の「構成名」を入力し、接続 URL を確認します。
- 9 「終了」をクリックします。


変更した構成が保管され、いつでも実行できるようになります。178 ページの「[インポート構成の実行](#)」を参照してください。

## インポート構成の削除

▶ インポート構成を削除するには:

- 1 「タスク領域」で、「モデルの管理」、「ステージング・テーブルのインポート」の順に選択します。

インポート・ページが表示されます。

- 2 削除する構成を選択し、「インポート構成の削除」ボタン  をクリックします。

削除を確認するメッセージが表示されます。

- 3 「はい」をクリックします。

構成が削除されます。

## インポート構成の実行

インポート構成が作成された後、データをアプリケーションにインポートする構成を実行できます。


▶ インポート構成を実行するには:

- 1 ステージング・テーブルが存在するリレーショナル・データベースの接続情報があることを確認します。

- 2 「タスク領域」で、「モデルの管理」、「ステージング・テーブルのインポート」の順に選択します。

インポート・ページが表示されます。

- 3 「構成リストのインポート」で、実行するインポート構成を選択します。

- 4 「インポート構成の実行」ボタン  をクリックします。

「データのインポート」ダイアログ・ボックスが表示されます。データ接続の詳細がすべてリストされています。

- 5 「次へ」をクリックします。

- 6 「構成」の下で、ターゲット・データベースおよび選択したステージング・テーブルが正しいことを確認します。
- 7 「次へ」をクリックします。  
「構成の要約」が表示されます。構成名と接続 URI が示されます。
- 8 次のいずれかの操作を選択します:
  - インポートを即時実行するには、「今すぐ実行」をクリックします。
  - インポートを実行するのにより都合のよい日付と時間をスケジュールするには、「後で実行」をクリックします。
- 9 「終了」をクリックします。
  - 「今すぐ実行」を選択した場合、インポートが実行され、選択したデータがターゲット・アプリケーションに移入されます。
  - 「後で実行」を選択した場合、ジョブは保存されます。保存したインポート・ジョブを実行するには、「ジョブ・プロセス」、「タスクフローの管理」の順に選択します。この画面で、ジョブを実行したり、より都合のよい日付と時間でジョブを実行するようスケジュールできます。詳細は、『Oracle Hyperion Profitability and Cost Management ユーザー・ガイド』を参照してください。
- 10 インポートが完了したら、インポートしたデータを確認します。179 ページの「インポートしたデータの確認」を参照してください。

## インポートしたデータの確認

データをターゲット・データベースにインポートした後、データが正しく完全にインポートされたことを確認する必要があります。

▶ インポートしたデータを確認するには:

- 1 Oracle Hyperion Enterprise Performance Management Workspace で、「ナビゲート」、「アプリケーション」、「Profitability」の順に選択し、Profitability and Cost Management モデルにアクセスするためのアプリケーション名を選択します。
- 2 ターゲット・アプリケーションでモデルを開き、インポートされたと推測されるデータを確認します。  
たとえば、ステージをインポートした場合、「モデルの管理」、「ステージ」の順に選択し、すべてのステージの情報が存在していて正しいことを確認します。
- 3 インポートしたステージ・テーブルごとに「例外」列を確認し、エラーや警告があるかどうか確認します。
- 4 ソース・ステージング・テーブルのエラーを修正し、インポートを再実行します。  
インポート時にエラーが生成されなくなるまで手順 2 から手順 3 を繰り返します。



# B

## EPM System ライフサイクル管理を使用したデータの移行

### この付録の内容

ライフサイクル管理について .....	181
ライフサイクル管理に対するデフォルトのタイムアウト設定の変更.....	182

## ライフサイクル管理について

ライフサイクル管理は、製品環境およびオペレーティング・システムでのアプリケーション、リポジトリ、または個々のアーチファクトの移行方法として一貫性のある方法を、Oracle Hyperion Enterprise Performance Management System 製品に提供します。

ライフサイクル管理の機能を使用して、次のタスクを実行できます:

- アプリケーションおよびフォルダの表示
- アーティファクトの検索
- アプリケーションから他のアプリケーションへの直接的な移行
- ファイル・システム間の移行
- 移行定義ファイルの保存とロード
- 選択されたアーティファクトの表示
- 移行の監査
- 移行のステータスの表示
- ファイル・システム上で変更をすばやく行うための、個々のアーチファクトのインポートおよびエクスポート

通常、Shared Services Console のライフサイクル管理インタフェースは、ライフサイクル管理をサポートするすべての EPM System 製品で一貫していますが、Oracle Enterprise Performance Management System 製品の場合、ライフサイクル管理インタフェースでは異なるアーチファクト・リストとエクスポートおよびインポート・オプションが表示されます。

Oracle Hyperion Shared Services Console にライフサイクル管理インタフェースを提供する他に、別の方法で移行元から移行先にアーティファクトを移行できる、ライフサイクル管理ユーティリティというコマンドライン・ユーティリティがあります。ライフサイクル管理ユーティリティは、Windows タスク・スケジューラなどのサードパーティ製のスケジューリング・サービスや Oracle Enterprise Manager と併用できます。

最後に、ライフサイクル管理アプリケーション・プログラミング・インタフェース(API)があります。これにより、ユーザーはライフサイクル管理機能のカスタマイズおよび拡張ができます。

ライフサイクル管理および Profitability and Cost Management のアーティファクトに関する詳細情報については、『Oracle Hyperion Enterprise Performance Management System ライフサイクル管理ガイド』を参照してください。

## ライフサイクル管理に対するデフォルトのタイムアウト設定の変更

Oracle Hyperion Enterprise Performance Management System ライフサイクル管理を使用して大きいモデルをインポートする場合、Oracle WebLogic Server のデフォルトのタイムアウト設定で指定された時間よりもインポート・プロセスに時間がかかることがあります。この問題を回避するには、デフォルトのタイムアウト設定をリセットする必要があります。

▶ WebLogic Server のデフォルトのタイムアウトを変更するには:

### 1 次にナビゲート:

```
%Middleware_HOME%\user_projects\epmsystem1\httpConfig\ohs\config
\OHS\ohs_component\mod_wl_ohs.conf
```

### 2 この構成ファイルでセクション「LocationMatch/profitability」を確認します:

```
<LocationMatch /profitability>
SetHandler weblogic-handler
PathTrim /
KeepAliveEnabled ON
KeepAliveSecs 20
WLIOTimeoutSecs 3000
Idempotent OFF
WeblogicCluster servername:6756
</LocationMatch>
```

```
SetHandler weblogic-handler
```

```
PathTrim
```

```
KeepAliveEnabled ON
```

```
KeepAliveSecs 20
```

```
WLIOTimeoutSecs 3000
```

```
WeblogicCluster servername:6756
```

### 3 LocationMatch/profitability セクションに次の行を追加します:

```
WLIOTimeoutSecs 3000
Idempotent OFF
```

4 次にナビゲート:

```
%Middleware_HOME%\user_projects\epmsystem1\httpConfig\ohs\config  
\OHS\ohs_component\httpd.conf
```

5 次のテキストに示すように、「Timeout」を「3000」に設定します:

```
# Timeout: The number of seconds before receives and sends time  
out. Timeout 3000
```

**注:** 前述のサーバー・タイムアウトは推奨の制限で、使用しているアプリケーション・サーバーのドキュメントに指定されている特定のタイムアウト設定に合うように変更できます。







# Profitability and Cost Managementコンポーネントのバックアップ

データの安全性と整合性を確保するために定期的にバックアップする必要がある Profitability and Cost Management データベースが複数あります:

- 運用データ・ストア
- インポート・ステージング・テーブル
- ブロック・ストレージ・オプション(BSO)データベース

バックアップの頻度は、データベースの変更量や、組織の一般基準によって決まります。

表 48 には、バックアップが必要なデータベース、そのタイプ、およびバックアップ・オプションの提案が記載されています。

**注意** 現在、ライフサイクル管理(LCM)を使用してインポート・ステージング領域または運用データ・ストアをバックアップすることはできません。

表 48 Profitability and Cost Management データベースのバックアップ要件

Database	データベース・タイプ	バックアップ・オプション
インポート・ステージング領域	リレーショナル・データベース(RDB)	次のオプションを含む、任意の RDB で使用可能な標準バックアップ技術: <ul style="list-style-type: none"><li>● データベース・スクリプト</li><li>● スケジューラ・スクリプト</li><li>● バックアップ・ツール(TOAD など)</li><li>● SQL Server または Oracle からのバックアップ手順。</li></ul>
運用データ・ストア	リレーショナル・データベース(RDB)	次のオプションを含む、任意の RDB で使用可能な標準バックアップ技術: <ul style="list-style-type: none"><li>● データベース・スクリプト</li><li>● スケジューラ・スクリプト</li><li>● バックアップ・ツール(TOAD など)</li><li>● SQL Server または Oracle からのバックアップ手順。</li></ul>

Database	データベース・タイプ	バックアップ・オプション
ブロック・ストレージ・オプション(BSO)と集約ストレージ・オプション(ASO)データベースおよび割当てルール(標準 Profitability の場合のみ)	Essbase	標準 Essbase バックアップ技術を使用する必要があります。バックアップ手順と復元手順の詳細は、『Oracle Essbase データベース管理者ガイド』を参照してください。
詳細 Profitability のモデル・データ・スキーマ	リレーショナル・データベース(RDB)	<p>次のオプションを含む、任意の RDB で使用可能な標準バックアップ技術:</p> <ul style="list-style-type: none"> <li>● データベース・スクリプト</li> <li>● スケジューラ・スクリプト</li> <li>● バックアップ・ツール(TOAD など)</li> <li>● SQL Server または Oracle からのバックアップ手順。</li> </ul>

バックアップ手順と復元手順の詳細は、Oracle Enterprise Performance Management System Backup and Recovery Guide を参照してください。



# Essbase命名規則

## この付録の内容

生成された計算スクリプトの命名規則.....	187
アプリケーションとデータベースの Essbase 命名規則.....	188
次元、メンバーおよび別名の Essbase 命名規則.....	189
属性計算次元の Essbase 命名規則.....	192

## 生成された計算スクリプトの命名規則

Oracle Hyperion Profitability and Cost Management によって Essbase スクリプトが生成される場合、これらのスクリプトは自動的に特定の命名規則を使用して名前が付けられます。ドライバ計算スクリプトと配賦計算スクリプトは、ステージごとに1つのスクリプトに結合されます。スクリプト名は、次のフォーマットで作成されます:

スクリプト名の文字列 = スクリプトの接尾辞 + POV識別子 + ステージ順序番号 + "\_" + インデックス;

- スクリプトの接尾辞は、スクリプトのタイプに基づいています。次の表は、接尾辞のリストを示します。

表 49 計算スクリプトの接尾辞

スクリプトのタイプ	費用レイヤー	収益レイヤー
ステージ間配賦	"a"	"r"
ステージ内配賦	"i"	"t"

- POV 識別子は、POV ID に基づいており、最大3桁まで使用できます。POVごとにスクリプトが生成されて識別されます。
- ステージ順序番号は、ソース・ステージの順序番号です(1、2、3など)。
- スクリプトの分割のために複数のスクリプトが生成される場合、「\_インデックス」には、同じタイプ、POV、ステージおよびレイヤーに対して数値シーケンスが表示されます。最初が\_01、次が\_02のように続きます。

### サンプル・スクリプト名

- a3682\_01.csc は、POV 識別子 368、ソース・ステージ 2、および費用レイヤーの計算スクリプトを示します。

- t4533\_02.csc は、POV 識別子 453、ソース・ステージ 3、および収益レイヤーのステージ内計算スクリプトを示します。

## アプリケーションとデータベースの Essbase 命名規則

アプリケーションとデータベースの名前を作成する場合、単語を表示するとおりに大文字と小文字を区別して名前を入力します。アプリケーション名またはデータベース名は、入力したとおりに作成されます。たとえば、すべて大文字で名前を入力した場合(例: NEWAPP)、Essbase によって自動的に、最初が大文字で後は小文字に変換されることはありません(例: Newapp)。

**注：** 次のリストは、制限の一部を示します。すべての制限の完全なリストについては、Oracle Essbase Database Administrator's Guide を参照することをお勧めします。

次の命名規則は、アプリケーションとデータベースに名前を付ける際に適用されます:

- Unicode モード以外のアプリケーションおよびデータベース名で使用できるのは、8 バイト未満です。
- Unicode モードのアプリケーションおよびデータベース名で使用できるのは、30 文字未満です。
- 名前ではスペースを使用しないでください。
- 名前の中に次の特殊文字を使用しないでください:
  - \*(アスタリスク)
  - +(プラス記号)
  - \ (円記号(バックスラッシュ))
  - ?(疑問符)
  - [] (大かっこ)
  - "(二重引用符)
  - :(コロン)
  - ;(セミコロン)
  - ,(カンマ)
  - '(一重引用符)
  - =(等号)
  - /(スラッシュ)
  - >(大なり記号)
  - タブ
  - <(小なり記号)

- |(縦棒)
- .(ピリオド)
- 集約ストレージ・データベースの場合のみ、アプリケーション名またはデータベース名に次の語を使用しないでください:
  - DEFAULT
  - LOG
  - METADATA
  - TEMP

## 次元、メンバーおよび別名の Essbase 命名規則

次元アウトラインを定義する場合、次元、メンバーおよび別名の命名用として使用できない制限文字があります。この項では、最も一般的な制限文字のリストを示します;ただし、完全なリストについて Oracle Essbase Database Administrator's Guide の Essbase 命名規則に関する項を参照することをお勧めします。

**注:** 次のリストは、制限の一部を示します。すべての制限の完全なリストについては、Oracle Essbase Database Administrator's Guide を参照することをお勧めします。

次元、メンバーおよび別名を命名する場合は、次の命名規則に従ってください:

- Unicode モード以外の次元、メンバー、または別名では、80 バイト未満を使用してください。
- Unicode モードの次元、メンバー、または別名では、80 文字未満を使用してください。
- 大/小文字は、大/小文字の区別が使用可能になっている場合のみ区別してください。大/小文字の区別を使用可能にするには、Oracle Essbase Database Administrator's Guide のアウトラインのプロパティの設定に関する項を参照してください。
- 次元名、メンバー名、別名、説明では、HTML タグは使用しないでください。
- 名前の中に、引用符、ピリオド、大かっこ、円記号(バックスラッシュ)またはタブを使用しないでください。

---

**注意** 大かっこは使用できますが、集約ストレージ・アウトラインに変換する場合にエラーの原因となることがあるため、ブロック・ストレージ・アウトラインでは推奨されていません。

---

- 同じ次元内で重複するメンバー名または別名は使用できません。
- 次元名またはメンバー名の先頭に、次の文字を使用しないでください:
  - アット・マーク(@)

- 円記号(バックスラッシュ)(\)
- 大かっこ([])
- カンマ(,)
- ダッシュ
- ハイフン
- マイナス記号(-)
- 等号(=)
- 小なり記号(<)
- 丸かっこ(() )
- ピリオド(.)
- プラス記号(+)
- 一重引用符(')
- 引用符(")
- アンダースコア(\_)
- 縦棒(|)
- 名前の先頭または末尾にスペースを入れないでください。Essbase ではこれらは無視されます。
- メンバー名ではスラッシュ(/)は使わないでください。
- カスタム・カレンダーの期間で、接頭辞にスペースは使用できません。
- 次元名またはメンバー名に、次のアイテムを使用しないでください。
  - 計算スクリプト・コマンド、演算子またはキーワード。コマンドの完全なリストについては、Oracle Essbase Database Administrator's Guide を参照してください。
  - レポート・ライターのコマンド
  - 関数名および関数の引数
  - 動的時系列が使用可能な場合、履歴、年、季、期間、四半期、月、週または日は使用しません。
  - 他の次元の名前、他のメンバーの名前(メンバーが共有でない場合)、世代名、レベル名およびデータベースでの別名。
- 次の語を使用しないでください:
  - ALL
  - AND
  - ASSIGN
  - AVERAGE
  - CALC
  - CALCMBR
  - COPYFORWARD

- CROSSDIM
- CURMBRNAME
- DIM
- DIMNAME
- DIV
- DYNAMIC
- EMPTYPARM
- EQ
- EQOP
- EXCEPT
- EXP
- EXPERROR
- FLOAT
- FUNCTION
- GE
- GEN
- GENRANGE
- GROUP
- GT
- ID
- IDERROR
- INTEGER
- LE
- LEVELRANGE
- LOOPBLOCK
- LOOPPARMS
- LT
- MBR
- MBRNAME
- MBRONLY
- MINUS
- MISSING
- MUL
- MULOP
- NE

- NON
- NONINPUT
- NOT
- OR
- PAREN
- PARENPARM
- PERCENT
- PLUS
- RELOP
- SET
- SKIPBOTH
- SKIPMISSING
- SKIPNONE
- SKIPZERO
- TO
- TOLOCALRATE
- TRAILMISSING
- TRAILSUM
- UMINUS
- UPPER
- VARORXMBR
- XMBRONLY
- \$\$UNIVERSE\$\$
- #MISSING
- #MI

## 属性計算次元の Essbase 命名規則

Essbase によって属性計算次元に作成されるメンバー(Sum、Count、Min、Max および Avg)は、予約語とみなされません。これは、これらの名前を属性計算次元で変更してから、属性次元または標準次元で標準名を使用できるためです。

アウトラインに一意のメンバー・アウトラインのタグが付けられている場合、Sum、Count、Min、Max および Avg をメンバー名として使用しないようにしてください。たとえば、Max を標準次元として使用してから、属性次元を作成すると、Essbase によってこの属性計算次元に Max メンバーが作成され、Essbase によって重複名が検出され、次のエラー・メッセージが戻されます:



"Analytic Serverエラー(1060115): 属性の計算次元/メンバー名はすでに使用されています。"

アウトラインに重複メンバー・アウトラインのタグが付けられており、Sum、Count、Min、Max および Avg を基本メンバーとして使用する前に属性次元(したがって属性計算次元)が存在している場合、Oracle Essbase では重複名が許可されません。ただし、属性次元を作成する前に合計、Sum、Count、Min、Max および Avg を基本メンバーとして使用する場合、重複名は許可されません。



# 索引

## 記号

- 2 パス計算, 58, 126
  - 2 パス計算(次元のみ), 59, 126
  - 2 パス計算(メンバーのみ), 59, 126

## A - Z

- AllAllocations 次元, 48
- AllocationType 次元, 41, 48, 65, 130
- ALL メンバー, 50, 118
- CalculatedDriverValue ドライバ, 42
- CostAssigned, 44
- CostAssignedIntraStage, 44
- CostAssignedPostStage, 44
- CostInput, 44
- CostPerDrvUnit ドライバ, 44
- CostReceived, 44
- CostReceivedIntraStage, 44
- CostReceivedPriorStage, 44
- create.sql, 175
- CREATE ANY SYNONYM, 109
  - 製品スキーマ, 109
  - モデル・データ・スキーマ, 111
- CREATE TABLE, 109
- CREATE VIEW, 109
- create\_dp\_staging.sql, 173
- create\_staging.sql, 173, 175
- DirectAllocation 次元, 48
- DROP ANY SYNONYM, 109
- EffectiveTotalDriverValue, 42
- EPM Workspace, 15
- Essbase
  - Essbase の操作, 36
  - 最適化設定, 56, 124
  - データ・ストレージ, 15
  - データベース, 35
  - 命名規則, 187
- Essbase Administration Services, 15
- Financial Reporting, 15
- FixedDriverValue ドライバ, 42
- GenealogyAllocation 次元, 48
- GrossCost, 47
- GrossReceivedCost, 44
- GrossReceivedRevenue, 45
- GrossRevenue, 47
- hpcm.log, 31
- HPM\_SQL\_SCRIPT
  - 詳細 Profitability, 143
- IdleCost, 44
- IdleDriverValue ドライバ, 42
- IdleRevenue, 45
- IndirectAllocation, 48
- InitialCost, 47
- InitialRevenue, 47
- MeasuresDetailed 次元
  - 詳細 Profitability, 117
- NetCostAfterIntraStage, 47
- NetCostForAssignment, 44
- NetReciprocalCost, 44
- NetReciprocalRevenue, 45
- NetRevenueAfterIntraStage, 47
- NetRevenueForAssignment, 45
- noalias
  - 詳細 Profitability, 117
  - 標準 Profitability, 49
- NoMember, 50, 118
  - 順序の変更, 73, 137
- ODBC データ・ソース、転送, 146
- Oracle Diagnostic Logging (ODL), 32
- Oracle Essbase Administration Services, 15
- OverDrivenRevenue, 45
- OverrideTotalDriverValue ドライバ, 42
- Percentage ドライバ, 42
- POV 次元, 59, 126
- POV 表示順, 59, 126
- Profit, 47
- Profitability アプリケーションのタイプ

詳細 Profitability, 15  
 標準 Profitability, 14  
 Quantity ドライバ, 42  
 ReciprocalIntermediateRevenue, 45  
 ReciprocalIntermediateCost, 44  
 Rate ドライバ, 42  
 ReciprocalCostAssigned, 44  
 ReciprocalCostReceived, 44  
 ReciprocalRevenueAssigned, 45  
 ReciprocalRevenueReceived, 45  
 RevenueAssigned, 45  
 RevenueAssignedIntraStage, 45  
 RevenueAssignedPostStage, 45  
 RevenueInput, 45  
 RevenueReceived, 45  
 RevenueReceivedIntraStage, 45  
 RevenueReceivedPriorStage, 45  
 root メンバー, 50, 118  
 Shared Services, 15  
 SharedServices\_Security\_Client.log, 31  
 Smart View, 15  
 SQL スクリプト  
   , 143  
   create.sql, 175  
   create\_staging.sql, 175  
   init.sql, 175  
 StandardCostRate, 44  
 StandardRevenueRate, 45  
 SysAllocVar1, 48  
 SysAllocVar2, 48  
 SysAllocVar3, 48  
 TotalDriverValueAfterReciprocals ドライバ, 42  
 TotalAllocation, 48  
 TotalDriverValue ドライバ, 42  
 UDA, 51, 58, 59, 120, 126。「属性次元」も参照  
 UnassignedCost, 44  
 UnassignedRevenue, 45  
 Unicode, 59, 126  
 UnitCost ドライバ, 44  
 URL, 18, 19  
 UserDefinedDriverMeasures, 41, 44  
 Web Analysis, 15  
 WebLogic  
   デフォルトのタイムアウト設定, 182  
 Weight ドライバ, 42  
 Workspace, 15

## あ行

アクセシビリティ, 14  
 アクセス権, 23  
 アプリケーション  
   Profitability and Cost Management の起動, 18  
   アプリケーション設定の変更  
     詳細 Profitability, 135  
     標準 Profitability, 70  
   アプリケーションのクリーンアップ, 77, 141  
   移行, 181  
   インストール後のアクセス, 17  
   詳細 Profitability, 15  
   次元の追加, 69, 134  
   配置, 77, 141  
   標準 Profitability, 14  
   命名規則, 187  
   要件, 65, 130  
 アプリケーションのクリーンアップ, 77, 141  
 アーティファクト・レポート, 28  
 インポート  
   インポート構成の削除, 178  
   インポート構成の作成, 175  
   インポート構成の実行, 178  
   インポート構成の変更, 177  
   インポートしたデータの確認, 179  
   インポート・テーブル, 174  
   メタデータ, 57, 125  
 インポート・ステージング・テーブル  
   バックアップ, 185  
 インポート・テーブルのスキーマ, 174  
 インポート用テーブル, 174  
 疎設定, 56, 124  
 運用データ・ストア  
   バックアップ, 185  
 エラー・メッセージ  
   ODL, 32  
 エンティティの次元, 41  
   タイプ, 65, 130  
 オーバードライブ費用, 44

## か行

階層タイプ, 58, 126  
   階層タイプ(次元のみ), 59, 126  
   階層タイプ(メンバーのみ), 59, 126  
 加重, 59, 126  
 カスタム・スクリプト, 143  
 監査, 28

- 勘定科目次元, 41
  - タイプ, 65, 130
- 完全配置, 77, 141
- 管理者(admin)のセキュリティの役割, 23
  - タスク, 16
- 共有次元, 39, 57, 125
- 共有ライブラリ, 39
- 国次元, 41, 57, 125
- 国次元タイプ, 65, 130
- 計算
  - 計算前および計算後のスクリプト, 143
  - 詳細 Profitability
    - 高度なオプション・ドライバ, 146
- 計算スクリプト
  - 命名規則, 187
- 計算データベース, 35
- 権限
  - 製品スキーマ, 109
- 検証, 74, 139, 141
  - アプリケーションの配置, 76
    - 詳細 Profitability, 140
  - 次元の順序付け, 52, 120
  - 条件, 74, 138
- 構成
  - インポート, 175
  - インポートの実行, 178
- 構成レポート, 28
- コメント, 59, 126
- 子メンバー
  - 順序の変更, 73, 137
- さ行**
- 最適化
  - Essbase, 56, 124
- 視点(POV)
  - POV 次元, 58, 65, 126, 130, 136
    - 標準 Profitability, 72
  - POV 表示順, 58, 126, 136
    - 標準 Profitability, 72
  - 次元, 41, 51, 119
    - 詳細 Profitability, 117
  - バージョン次元, 41
    - 詳細 Profitability, 117
- 収益レイヤー配賦メジャー, 41
- 集計, 58, 59, 126
- 集約でのレベルの使用, 59, 126
- 詳細 Profitability
  - 概要, 15
- 時間次元, 41, 57, 125
  - タイプ, 65, 130
- 次元, 65, 130
  - AllAllocations, 48
  - AllocationType, 41, 48
  - DirectAllocation, 48
  - Essbase の設定の最適化, 56, 124
  - GenealogyAllocation, 48
  - MeasuresDetailed
    - 詳細 Profitability, 117
  - POV, 51, 119
  - アプリケーションへの追加, 69, 134
  - エンティティ, 41
- 勘定科目, 41
- 共有, 39
  - 共有およびローカル, 57, 125
- 最適化設定, 56, 124
  - 詳細 Profitability, 117
- 時間、国および通貨, 57, 125
- 次元ソート順, 58, 126
- 次元ライブラリ, 55
- 順序, 52, 120
  - ソート順, 53, 121
  - ソート順の推奨事項, 53, 121
- 属性, 41
  - 詳細 Profitability, 117, 120
  - 標準 Profitability, 51
- タイプ, 41
  - 詳細 Profitability, 117
- バージョン, 41, 51, 119
- パフォーマンス設定, 52, 120
- ビジネス, 41, 50, 118
  - 詳細 Profitability, 117
- プロパティ, 58, 126
- 別名
  - 詳細 Profitability, 117
  - 標準 Profitability, 49
- 密度, 52, 120
- 命名規則, 187
- メジャー, 41
- 要件, 50, 118
- ローカル, 39
- 次元解決順, 58, 59, 126
- 次元式, 58, 126
  - 次元式(BSO), 59
- 次元ストレージ・タイプ, 58, 59, 126

次元ソート順, 52, 58, 59, 120, 126  
 次元タイプ, 58, 126  
   AllocationType, 65, 130  
   POV, 65, 130  
   POV 次元, 136  
     標準 Profitability, 72  
   POV 表示順, 136  
     標準 Profitability, 72  
   属性, 65, 130  
   ビジネス, 65, 130  
   別名, 65, 130  
   メジャー, 65, 130  
 次元の解決順, 59, 126  
 次元の管理  
   次元ライブラリ, 55  
 次元メンバー  
   IndirectAllocation, 48  
   SysAllocVar1, 48  
   SysAllocVar2, 48  
   SysAllocVar3, 48  
   TotalAllocation, 48  
   命名規則, 187  
 次元ライブラリ, 55  
 スキーマ  
   詳細 Profitability  
     製品, 109  
     モデル・データ, 111  
   製品, 105, 107  
   モデル・データ, 105, 107  
 ステージ  
   インポート, 174  
   データ定義のエクスポート, 95  
   ビューのエクスポート, 161  
 ステージング・テーブル, 174  
   作成, 175  
   バックアップ, 185  
 製品スキーマ, 105, 107  
   詳細 Profitability, 109  
   付与, 109  
 セキュリティ役割, 21  
 セキュリティ・レポート, 28  
 設定  
   アプリケーション設定の変更  
     詳細 Profitability, 135  
     標準 Profitability, 70  
   疎, 56, 124  
   最適化, 56, 124

次元ソート順, 52, 120  
 次元のパフォーマンス設定, 52, 120  
 デフォルトのタイムアウト, 182  
 密, 56, 124  
 密度, 52, 120  
 操作タイプ  
   詳細 Profitability, 146  
 ソート順, 52, 120  
   推奨事項, 53, 121  
   設定, 53, 121  
 属性  
   属性タイプ, 59, 126  
   プロパティ, 59, 126  
 属性次元, 41, 65, 130  
   詳細 Profitability, 117, 120  
   標準 Profitability, 51

## た行

タイムアウト  
   WebLogic, 182  
   デフォルト設定の変更, 182  
 対話型ユーザーのセキュリティの役割, 23  
 タスク, 16  
 代替階層  
   階層タイプの設定, 59, 126  
 通貨次元, 41, 57, 65, 125, 130  
 デフォルト設定  
   タイムアウト, 182  
 データ・ストレージ, 15, 58, 126  
   データ・ストレージ(ASO), 59  
 データ転送、ODBC データ・ソース, 146  
 データの同期, 79  
 データベース  
   概要, 36  
   計算データベース(BSO), 35  
   命名規則, 187  
   リレーショナルおよび Essbase, 35, 36  
   レポート用データベース(ASO), 35  
 データベース・ビュー  
   モデル定義データ, 95  
 同期, 79  
 ドライバ  
   インポート, 174  
   詳細 Profitability  
     タイプ, 146  
   ドライバ基準タイプ, 85  
   ドライバの関連付け

インポート, 174  
 ドライバ・メジャー, 41  
 CalculatedDriverValue, 42  
 CostPerDrvUnit, 44  
 EffectiveTotalDriverValue, 42  
 FixedDriverValue, 42  
 IdleDriverValue, 42  
 OverrideTotalDriverValue, 42  
 Percentage, 42  
 Quantity, 42  
 Rate, 42  
 StandardCostRate, 44  
 TotalDriverValueAfterReciprocals, 42  
 TotalDriverValue, 42  
 UserDefinedDriverMeasures, 44  
 Weight, 42  
 テーブル, 42  
 ドライバ優先度シーケンス, 85

## は行

### 配置

検証条件, 76

詳細 Profitability, 140

### 配賦メジャー

CostAssigned, 44

CostAssignedIntraStage, 44

CostAssignedPostStage, 44

CostInput, 44

CostReceived, 44

CostReceivedIntraStage, 44

CostReceivedPriorStage, 44

GrossReceivedCost, 44

GrossReceivedRevenue, 45

IdleCost, 44

IdleRevenue, 45

NetCostForAssignment, 44

NetReciprocalCost, 44

NetReciprocalRevenue, 45

NetRevenueForAssignment, 45

ReciprocalIntermediateRevenue, 45

ReciprocalIntermediateCost, 44

ReciprocalCostAssigned, 44

ReciprocalCostReceived, 44

ReciprocalRevenueAssigned, 45

ReciprocalRevenueReceived, 45

RevenueAssigned, 45

RevenueAssignedIntraStage, 45

RevenueAssignedPostStage, 45

RevenueInput, 45

RevenueReceived, 45

RevenueReceivedIntraStage, 45

RevenueReceivedPriorStage, 45

StandardCostRate, 44

UnassignedCost, 44

UnassignedRevenue, 45

収益レイヤー配賦メジャーのテーブル, 45

費用レイヤーおよび収益レイヤー, 41

費用レイヤー配賦メジャーのテーブル, 44

汎用ビジネス次元, 50, 118

### バックアップ

インポート・ステー징・テーブル, 185

運用データ・ストア, 185

ブロック・ストレージ・オプション(BSO)

データベース, 185

割当てルール, 185

バージョン次元, 41, 51, 119

パワー・ユーザーのセキュリティの役割, 23

### 表示順序

POV 次元, 136

標準 Profitability, 72

### 標準 Profitability

概要, 14

表示ユーザーのセキュリティの役割, 23

費用レイヤー配賦メジャー, 41

ビジネス次元, 41, 50, 65, 118, 130

詳細 Profitability, 117

### 付与

モデル・データ・スキーマ, 111

ブロック・ストレージ・オプション(BSO)データベース

バックアップ, 185

プライマリ・レベルの加重, 59, 126

### プロパティ

編集, 57, 125

### 別名

次元

詳細 Profitability, 117

標準 Profitability, 49

次元の選択, 65, 130

プロパティ, 59, 126

命名規則, 187

要件, 65, 130

## ま行

密設定, 56, 124  
 密度, 52, 120  
 命名規則, 187  
     データベースのインポート, 175  
 メジャー  
     ドライバ・メジャー, 41  
     配賦メジャー, 41  
     レポート作成, 41  
 メジャー次元, 41, 65, 130  
 メンバー  
     順序の変更, 73, 137  
     プロパティ, 58, 126  
 メンバー解決順, 58, 126  
     メンバー解決順(次元のみ), 59, 126  
     メンバー解決順(メンバーのみ), 59, 126  
 メンバー式, 58, 126  
     メンバー式(ASO), 59, 126  
     メンバー式(BSO), 59, 126  
 モデル  
     モデル定義データのエクスポート, 95  
 モデル定義データのエクスポート, 95  
 モデル・データ・スキーマ, 105, 107  
     詳細 Profitability, 111  
     付与, 109

## や行

役割, 23  
 優先度シーケンス, 85  
 ユーザー  
     セキュリティ役割, 21  
     認可, 21  
     プロビジョニング, 21  
 ユーザーのプロビジョニング, 21

## ら行

ライフサイクル管理, 181  
     概要, 181  
     コマンドライン・ユーティリティ(CLU),  
         181  
     デフォルトのタイムアウト設定, 182  
 ライフサイクル管理のコマンドライン・ユー  
 ティリティ(CLU), 181  
 リレーショナル・データベース, 35, 36  
 レポート  
     監査, 28  
     レポート・メジャー, 47

レポート・データベース, 35  
 レポート・メジャー, 41  
     GrossCost, 47  
     GrossRevenue, 47  
     InitialCost, 47  
     InitialRevenue, 47  
     NetCostAfterIntraStage, 47  
     NetRevenueAfterIntraStage, 47  
     Profit, 47  
     テーブル, 47  
 ログイン URL, 19  
 ログ・ファイル, 31  
     hpcm.log, 31  
     ODL, 32  
     SharedServices\_Security\_Client.log, 31  
     検証, 17  
 ログ・ファイルのエラー・メッセージ, 31  
 ローカル次元, 39, 57, 125

## わ行

割当てルール  
     インポート, 101, 174  
     バックアップ, 185  
     割当てルール選択のインポート, 91