Oracle® Essbase Studio

User's Guide

リリース 11.1.2.3.000

ORACLE

ENTERPRISE PERFORMANCE
MANAGEMENT SYSTEM

Essbase Studio User's Guide, 11.1.2.3.000

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 19

第 1 章 Essbase Studio の概要 21

Essbase Studio の紹介 21

Essbase Studio ワークフロー 22

Essbase Studio ユーザー・インタフェース 23

データ・ソース・ナビゲータ 24

作業領域 25

メタデータ・ナビゲータ 25

Essbase Studio へのログイン 26

Essbase Studio サーバーへの再接続 26

第 2 章 管理 29

管理の概要 29

Essbase Studio カタログ・データベースの設定 29

Essbase Studio カタログおよびデータのアップグレード 31

Essbase Studio に対して SSL を使用可能にする 32

サーバー・プロパティ・ファイルの構成 33

サーバー・プロパティ 35

catalog.url 36

server.css.URL 38

catalog.db 39

catalog.username 40

catalog.password 40

server.hss.bpmApplication 40

server.datafile.dir 41

server.essbase.streamingCubeBuilding 42

server.timeoutPeriod 43

server.queueSize 43

server.threadCount 43

server.resourceCount 44

server.sql.fetchSize 44

server.tempDir 45

server.charset 45
server.readLockTimeOut 46
server.writeLockTimeOut 46
server.essbase.TPTapi 47
server.essbase.disableDistinct 47
server.httpPort 48
transport.port 48
server.runInBackground 49
server.essbase.blindShare 50
oracle.jdbc.ReadTimeout 50
server.jdbc.teradata.properties 51
data-source-type.cache.size 51
data-source-type.pool.maxsize 52
server.essbase.uniqueMemberFromCaptionBinding 53
catalog.pool.size 54
サーバー・プロパティ・ファイルの例 54
Oracle の例 54
IBM DB2 の例 55
Microsoft SQL Server の例 56
ロギングの構成 57
Essbase Studio サーバーの起動ファイルの操作 57
UNIX システムでの ODBC の構成 59
.odbc.ini へのシンボリック・リンクの作成 60
odbcinst.ini へのドライバ記述子の追加 60
odbc.ini ファイルへの DSN の追加 61
Oracle の ODBC 設定の例 63
IBM DB2 UDB の ODBC 設定の例 63
MySQL の ODBC 設定の例 64
Microsoft SQL Server の ODBC 設定の例 64
Netezza の ODBC 設定の例 64
Teradata の ODBC 設定の例 65
JDBC ドライバの構成 66
Essbase Studio サーバーおよび Essbase Studio コンソールの開始と停止 67
UNIX 上におけるフォアグラウンドでの Essbase Studio サーバーの開始 68
Essbase Studio サーバーのコマンド 69
Essbase Studio カタログ・データベースのエクスポートとインポート 72
カタログのエクスポートおよびインポートについて 72
XML ファイルへのカタログ・データベースのエクスポート 73
カタログ・データベース全体のエクスポート 74

選択したカタログ要素のエクスポート 74
カタログ XML ファイルのインポート 76
仮想メモリーの構成 78
配置履歴の消去 79
再ホストされた EPM System 製品への参照の更新 80
再ホストされた Essbase サーバーへの参照の更新について 80
再ホストされた Essbase サーバーへの参照の更新 81

第 3 章 ユーザー管理およびセキュリティ 83
Shared Services と Essbase Studio の併用について 83
Shared Services の Essbase Studio の役割 83
Shared Services Console の起動およびログイン 84
Shared Services ユーザーへのアクセス権の割当て 85

第 4 章 データ・ソース接続 87
概要 87
接続ウィザードを使用した接続の作成 88
接続パラメータの定義 88
接続に含めるテーブルの選択 89
ミニスキーマ・オプションの選択 89
ミニスキーマの移入 90
メタデータ要素の作成 90
リレーショナル・ソースへの接続の作成 90
リレーショナル・ソースの接続パラメータの定義 91
リレーショナル・ソースのテーブルの選択 94
リレーショナル・ソースのミニスキーマ・オプションの選択 95
リレーショナル・ソースのミニスキーマの作成 96
リレーショナル・ソース用のミニスキーマ作成のスキップ 96
リレーショナル・ソースのミニスキーマへの移入 96
リレーショナル・ソースのメタデータ要素の作成 98
Oracle BI EE 次元の作成 98
Oracle BI EE のキューブ・スキーマおよび Essbase モデルの作成 100
Essbase への接続の作成 101
Performance Management Architect データ・ソースへの接続の作成 102
Performance Management Architect ソースの接続パラメータの定義 102
Performance Management Architect ソース用のメタデータ要素の作成 103
テキスト・ファイルのデータ・ソースへの接続の作成 104
テキスト・ファイルのデータ・ソースについて 104
テキスト・ファイル・ソースの接続パラメータの定義 105
テキスト・ファイルのモデリング 107

テキスト・ファイル・ソースのミニスキーマ・オプションの選択 108
テキスト・ファイル・ソースのミニスキーマの作成 109
テキスト・ファイル・ソース用のミニスキーマ作成のスキップ 109
テキスト・ファイル・ソースのミニスキーマの移入 109
テキスト・ファイル・ソースからのメタデータ要素の作成 110
データ・ソース接続の同期 111
データ・ソース接続の増分更新の実行 111
リレーショナル・データ・ソースの増分更新の実行 111
テキスト・ファイルのデータ・ソースの増分更新の実行 113
データ・ソース接続からのテーブルの削除 115
データ・ソース接続のテーブルのリフレッシュ 116
テーブルのリフレッシュ 116
結合の追加 117
データ・ソース接続の操作 118
データ・ソース接続でのイントロスペクションの実行 118
データ・ソース接続プロパティの編集 118
ソース・テーブルおよび列のプロパティの表示 121
リレーショナル・ソース・テーブルおよびユーザー定義テーブルのプロパティの表示 121
リレーショナル・ソース列およびユーザー定義テーブル列のプロパティの表示 122
ユーザー定義テーブルの基となるステートメントの表示 122
テキスト・ファイルのソース・ファイルのプロパティの表示 122
テキスト・ファイル・ソースの列のプロパティの表示 123
サンプル・データの表示 123
接続リストのリフレッシュ 125
接続の削除 125
わかりやすい名前の表示 125
ユーザー定義テーブルの作成 126
ユーザー定義テーブルについて 127
ユーザー定義テーブルの例 127

第 5 章 ミニスキーマ 129
ミニスキーマについて 129
ミニスキーマの作成または編集 130
ミニスキーマの全般プロパティの設定 130
ミニスキーマでのテーブルの追加または削除 131
ミニスキーマの操作 132
ミニスキーマの保存 133
ミニスキーマを開く 133

ミニスキーマ・テーブルおよび列のプロパティの表示 133
ミニスキーマからの要素の削除 134
ミニスキーマでの結合の追加または削除 134
手動での結合の追加または編集 134
検査による結合の追加 137
ミニスキーマでのテーブルの追加 138
サンプル・データの表示 138
ミニスキーマ・テーブルへの色の適用 139
ミニスキーマ・オブジェクトからのメタデータ要素の作成 139
ミニスキーマの表示 140
ミニスキーマ作業領域の使用 141
ミニスキーマ作業領域の最大化と最小化 143
ミニスキーマ作業領域のズーム・インおよびズーム・アウト 143
ミニスキーマでのテーブルの配置 144
ミニスキーマのプロパティの編集 144
ミニスキーマ・リストのリフレッシュ 145
ミニスキーマの削除 145

第 6 章 イントロスペクション 147
イントロスペクションの概要 147
イントロスペクション・ウィザードでのミニスキーマ・オプションの選択 148
イントロスペクション・ウィザードでのファクト・テーブルの選択 148
イントロスペクション・ウィザードでの次元テーブルの選択 149
イントロスペクション・ウィザードでの階層の選択 150

第 7 章 メタデータ要素 153
メタデータ要素について 153
メタデータ要素の作成または編集 154
次元要素および導出テキスト・メジャーの作成または編集 154
次元要素の作成 154
次元要素の一般プロパティの定義または編集 156
次元要素の基となる式の作成 157
次元要素のキー・バインディング・オプションの選択 159
導出テキスト・メジャーの作成または編集 162
指定した別名セットに対する別名セット・バインディングの作成または編集 163
テキスト・リストの作成または編集 164
日付要素の作成 165
メタデータ・フォルダの作成または編集 165
メタデータ要素の操作 166

メタデータ要素のコピー 166
メタデータ要素の名前変更 166
メタデータ要素の削除 167
メタデータ要素のサンプル・データの表示 167
系列の表示 167

第 8 章 別名セット 169
別名セットについて 169
別名セットの操作 170
別名セットの作成 170
手動による別名セットのバインディングの作成 170
検査による別名セットのバインディングの作成 171
別名セットの編集 172
別名セットの次元要素およびバインディングのソート 173
別名セットのバインディングの変更 173
別名セットへのバインディングの追加 174
別名セットからのバインディングの削除 174
別名セットの管理 175
別名セットのコピーまたは移動 175
別名セットの名前変更 175
別名セットの削除 176
別名セットのエクスポート 176

第 9 章 階層 177
階層について 177
メジャー階層について 178
標準階層およびメジャー階層の作成 180
階層での遅延キー・バインディング列の使用 182
階層の例 184
シングルチェーン階層 185
マルチチェーン階層 185
共有メンバーを持つマルチチェーン階層(代替階層) 185
属性次元を持つマルチチェーン階層 186
再帰階層 187
物理およびメタデータ要素から構築された階層 188
DATE 型のメタデータ要素から構築された時間階層 189
カレンダ階層の作成 189
グレゴリオ暦カレンダ階層 190
会計カレンダ階層 190
リテール・カレンダ階層 192

ISO カレンダ階層 193
製造カレンダ階層 194
時間レベルの定義 195
日付属性の定義 196
リンク値属性 196
階層の編集 197

第 10 章 キューブ・スキーマ 199
キューブ・スキーマの概要 199
キューブ・スキーマの作成または編集 199
キューブ・スキーマのメジャーおよび階層の選択 200
階層のプレビュー 201
キューブ・スキーマ・オプションの設定 202
データ・ロード・マッピングの定義 204
既存のキューブ・スキーマからの Essbase モデルの作成 206

第 11 章 Essbase プロパティ 207
モデル・プロパティ 207
Essbase モデルについて 207
Essbase モデルの概要 208
「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス 209
モデルの全般プロパティの設定 209
標準データ・ロード SQL の上書き 211
データ・ロード SQL の上書き編集のガイドライン 212
OLAP の概要 213
XOLAP の概要 214
XOLAP へのモデルの指定 216
Essbase モデルの別名セットの選択 217
モデルの属性の定義 218
属性の概要 218
可変属性の概要 219
属性メンバー名のフォーマットの設定 221
属性計算メンバー名のフォーマットの設定 221
属性のブール、日付および数値の範囲の指定 223
次元のプロパティ 224
次元の概要 224
次元タイプ 225
時間次元の使用ルール 225
会計次元の使用ルール 226
「次元プロパティ」タブへのアクセス 226

次元の全般プロパティの設定 226
世代およびレベルの命名 227
次元の編集 228
次元タイプの選択 229
動的時系列の使用 230
次元の保管方法の選択 231
2 パス計算の選択 231
2 パス計算の概要 232
データの保管方法の選択 232
解決順の選択 233
会計次元プロパティの編集 234
タイム・バランスの選択 234
タイム・バランスの概要 235
差異レポートの作成方法の選択 235
スキップ・オプションの選択 236
次元への式の追加 237
式について 237
次元の別名の表示および編集 238
次元へのユーザー定義属性の割当て 239
UDA の概要 240
アウトライン構築オプションの選択 240
階層ストレージ設定の選択 241
重複するメンバー設定の移動 242
共有メンバーの前に実メンバーを配置 242
データ・ロードの最適化 243
次元の順序付け 244
メンバーのプロパティ 245
メンバーの概要 245
「メンバーのプロパティ」タブへのアクセス 246
メンバーの全般プロパティの設定 246
メンバーの全般プロパティの指定 246
属性としてのメンバーの選択 247
可変属性の履歴テーブルの設定 247
メンバーの可変属性の設定 249
メンバー情報の編集 253
子メンバーの集計メソッドの選択 253
子メンバーの集計の概要 254
メンバーを計算するときの解決順の選択 255
解決順の概要 255

2 パス計算オプションの選択 256
メンバー・データの保管方法の選択 256
「ラベルのみ」オプションを持つ集約ストレージ 257
集約ストレージ・オプションの選択 258
集約ストレージのガイドライン 258
会計次元のメンバーの編集 260
タイム・バランスの選択 260
スキップ・オプションの選択 261
差異レポートの作成方法の選択 262
データ・ロード・スケーリングの指定 263
型付きメジャーの指定 263
メンバーへの式の追加 264
メンバーの別名の編集 265
別名の概要 266
別名セットの表示 267
別名の変更およびリフォーマット 267
検索ルールを作成して別名を変更およびリフォーマットする方法 . . . 268
検索ルールを変更して別名を変更およびリフォーマットする方法 . . . 269
別名の大文字/小文字の変更 270
別名のスペースのリフォーマット 270
別名への接頭辞の追加 271
別名への接尾辞の追加 272
メンバーへのユーザー定義属性の割当て 273
メンバーの変換 274
メンバー名変換シーケンス 274
メンバーへの接頭辞の追加 275
メンバーへの接尾辞の追加 276
メンバー名の変更およびリフォーマット 277
検索ルールを作成してメンバー名を変更およびリフォーマットする方法 277
検索ルールを変更してメンバー名を変更およびリフォーマットする方法 278
メンバーの大文字/小文字の変更 279
メンバーのスペースのリフォーマット 280
Essbase モデル・プロパティの操作 280
モデルの表示 281
モデル作業領域を開く 281
モデル作業領域の使用 281
モデル・プロパティの検証 282
プロパティへの変更の確認 283

モデルの参照 283
ツール・チップの使用 284

第 12 章 モデルの再同期 287
モデルの再同期について 287
モデルの再同期の使用 288
メタデータ要素 - モデルの再同期 288
モデルの再同期 288

第 13 章 キューブ配置 291
キューブ配置について 291
配置シナリオとストリーミング・キューブ・ビルディング・プロパティの注意点 292
日付メジャーの配置 294
キューブの配置 295
Essbase サーバー接続の作成 295
キューブ配置のための接続情報の提供 297
配置オプションの設定 299
キューブ配置の増分ロードの設定 304
配置の進捗状況および結果の表示 305

第 14 章 配置済アプリケーション 307
配置履歴の表示 307
キューブ・リンケージの更新 308
キューブ・リンケージについて 308
個々のキューブのキューブ・リンケージを更新する前に 309
個々のキューブのキューブ・リンケージの更新 309
Essbase Studio の再ホスト後のキューブ・リンケージの更新 311

第 15 章 系列 313
系列の概要 313
系列作業領域を開く 315
系列作業領域の使用 315

第 16 章 ドリルスルー・レポート 317
ドリルスルー・レポートの概要 317
ドリルスルー・レポートの操作 318
ドリルスルー・レポートの作成 319
レポートの交差レベルの指定 319
交差レベルの設定場所の指定 320
再帰階層の交差レベルの指定 321
レポート・タイプの定義とレポートのカスタマイズ 321

リレーショナル・ソースのレポートの定義およびカスタマイズ 322
リレーショナル・レポート・タイプの定義とドリルスルー・レポートの列の指定 322
ドリルスルー・レポート列のソート順の定義 324
テンプレート SQL の定義 325
リレーショナル・ソースの行ガバナーの指定 326
リレーショナル・ソースのドリルスルー・レポート・フィルタの指定 327
リレーショナル・ソースのレポートのテスト 328
テスト・シナリオの例–キャプションおよびキー・バインディングが異なる場合 329
テスト・シナリオの例–再帰階層 331
URL へのレポートの定義およびカスタマイズ 332
サンプル URL テンプレート 333
サンプル FDM URL テンプレート 334
サンプル Oracle BI EE URL テンプレート 336
Java メソッドへのレポートの定義およびカスタマイズ 337
ドリルスルー・レポートと Essbase モデルの関連付け 340

第 17 章 検索とサーチ 343
検索について 343
メタデータ要素の検索 343
メタデータ要素のサーチ 344

第 18 章 プリファレンス 347
一般プリファレンスの設定 347
スキーマ・プリファレンスの設定 347
Essbase モデルのプリファレンスの設定 348

第 19 章 Integration Services カタログ移行 351
カタログ移行の概要 351
モデルの移行 351
メタアウトラインの移行 352
階層のマッピング 352
メジャー列のマッピング 353
Essbase エクスポート・モデルの作成 353
別名のマッピング、ユーザー定義メンバーおよびレポート 353
「EIS カタログの移行」ダイアログ・ボックスへのアクセス 353
EIS モデルおよびメタアウトラインの移行 354
ガイドラインおよび制限 356
メタアウトラインとモデルの移行のガイドライン 356

移行されないプロパティ 357

付録 A. 制限とガイドライン 359
概要 359
カタログおよびデータ・ソースのガイドライン 359
カタログ・データベースにつき 1 つの Essbase Studio サーバー 360
カタログ・アクセスのガイドライン 360
カタログおよびデータ・ソース権限のガイドライン 360
Oracle クライアント・ドライバのガイドライン 361
カタログ・エクスポート・ファイルに含まれないパスワード 361
XML のインポートに関する制限 361
MySQL の制限 362
Microsoft SQL Server の制限 362
データ・ソースとしての Excel ファイルはサポートされていない 362
Oracle Call Interface (OCI)の制限 362
列のデータ型のサポート 362
Essbase Studio カタログのアップグレード 363
ユーザー定義 SQL の検証 364
ユーザー定義 SQL のデータ・ソース権限 364
カタログおよびデータ・ソースの一般的な制限事項とガイドライン 364
イントロスペクションに関する制限 364
メタデータ要素使用のガイドラインと制限 365
導出テキスト・メジャーに関する制限 365
サイクル依存関係のガイドライン 365
Oracle BI EE のビジネス・モデル次元要素のキー・バインディングとキャプション・バインディングの生成ルール 365
階層の使用方法のガイドラインと制限 366
階層のガイドライン 366
標準階層およびメジャー階層に関する制限 367
カレンダ階層に関する制限 367
キューブ・スキーマに関する制限 367
Essbase プロパティの編集および用途に関する制限 368
Essbase モデルの再構築のガイドライン 368
カスタム・データ・ロード SQL のガイドライン 369
重複するメンバー名のサポートの制限事項 369
XOLAP の機能のガイドライン 369
独立次元のバインディングに関する制限 371
可変属性編集のガイドライン 371
テキスト・ファイルのデータ・ソース・メンバーの変換の制限 372
Essbase ビジネス・モデル・ソースに基づく Essbase モデル 372

Essbase モデルの再同期のガイドラインと制限 372
キューブ配置の制限とガイドライン 373
キューブ配置に関する一般的な制限 373
管理サービス・コンソールでのルール・ファイルの制限 374
XOLAP キューブの Essbase Studio サーバーで生成された MaxL について 374
Essbase Studio ルール・ファイルでの MaxL の Import Dimensions ステートメントの使用 375
ドリルスルー・レポートの制限とガイドライン 375
ドリルスルー操作の一般的な制限事項 375
ドリルスルーにおけるポート番号の制限 376

付録 B. Essbase Studio の命名規則 377
データベースとアプリケーションの命名規則 377
メタデータ要素の命名規則 378
Essbase 命名ルールの制限を受けるメタデータ要素 378
メタデータ要素の命名規則 379
メタデータ要素の命名の一般的なガイドライン 379
使用できない文字 380
予約語 380
ドリルスルー・レポートの命名ガイドライン 380
Essbase 命名規則の例外 381

付録 C. CPL リファレンス 383
CPL 式の概要 383
オペランド 383
構文の要素 383
接続 385
2 つのレベルを持つデータベース 385
3 つのレベルを持つデータベース 385
フラット・ファイル(1 レベル) 385
ユーザー定義テーブル 385
クラス 386
定数 386
演算子 388
計算の順序 388
グループ化 389
例 389
論理演算子 389
例 391

算術演算子 391
例 392
文字列演算子 393
例 393
CPL SQL 関数 393
日付 393
dayOfMonth 393
month 394
monthName 394
monthShortName 395
quarter 395
quarterAsString 395
weekday 396
weekdayName 396
weekdayShortName 397
年 397
yearShort 398
文字列 398
contains 398
index 399
leftStr 399
length 400
lower 400
lTrim 400
rightStr 401
rTrim 401
soundex 402
substr 402
trim 403
upper 403
数値 404
abs 404
exp 404
ln 404
log10 405
pow 405
sqrt 406

用語集 407

索引 433

ドキュメントのアクセシビリティについて

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

Access to Oracle Support

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

1 Essbase Studioの概要

この章の内容

Essbase Studio の紹介 21
Essbase Studio ワークフロー 22
Essbase Studio ユーザー・インタフェース 23
Essbase Studio へのログイン 26
Essbase Studio サーバーへの再接続 26

Essbase Studio の紹介

Oracle Essbase Studio は、データのモデリング、キューブの設計、分析アプリケーションの構築などに関連するタスクを実行するための単一の環境を提供することにより、キューブの構築を容易にします。Essbase Studio は、キューブの構築アクティビティを単一のインタフェースに統合することで、アウトラインの構築およびデータのロードに一貫したプラットフォームを提供します。

Essbase Studio では、ウィザード主体のユーザー・インタフェースを使用して、様々なデータ・ソース・タイプのモデリングをサポートします。Oracle Essbase アプリケーションは通常そこから構築されるため、Essbase Studio をすべてのキューブ関連データのモデリングを実行できる単一ポイントにすることができます。

共通のメタデータ・リポジトリ、つまりカタログでは、企業で構築されるすべての Essbase アプリケーションに関連するすべてのメタデータが取り込まれ、最低限の粒度でメタデータを再使用できます。これらのカタログは、Essbase Studio に共通のメタデータに関する情報を提供し、それらを企業全体の様々なアプリケーション間で共有します。

Essbase Studio では、リレーショナル・データベース、Oracle Business Intelligence Enterprise Edition、URL、カスタム SQL および Java メソッドなど、いくつかのドリルスルー・オプションがサポートされています。ドリルスルーの機能は、データ・セルおよびメンバー・セルでサポートされており、一致するメタデータ・コンテキストを持つキューブに動的にリンクします。

また、Essbase Studio では、メタデータ関係の豊富なグラフィック表示を使用した系列の追跡もサポートされています。これにより、ユーザーはアプリケーションの系列を、メタデータ・コンポーネントからその元となるデータ・ソースまでたどることができます。

Essbase Studio ワークフロー

表 1 は、Essbase Studio を使用するときのワークフローを示してます。

表 1　Essbase Studio ワークフロー

手順	タスク	詳細情報
1.	**データ・ソース接続を作成します。接続ウィザード**を使用して、各種のデータ・ソースに接続します。データ・ソースは、リレーショナル・データベース、Oracle BI EE ソース、Oracle Hyperion EPM Architect アプリケーションまたはテキスト・ファイルのデータ・ソースなどです。	● 88 ページの「接続ウィザードを使用した接続の作成」 ● 90 ページの「リレーショナル・ソースへの接続の作成」 ● 102 ページの「Performance Management Architect データ・ソースへの接続の作成」 ● 104 ページの「テキスト・ファイルのデータ・ソースへの接続の作成」 ● 111 ページの「データ・ソース接続の同期」 ● 118 ページの「データ・ソース接続の操作」 ● 126 ページの「ユーザー定義テーブルの作成」
2.	**メタデータ要素を作成または編集**。メタデータ要素は、データ・ソース接続の作成時またはその後に作成できます。メタデータ要素は、Essbase Studio での使用を目的としてすでに保管およびカタログが作成されている他のメタデータから作成することもできます。	● 154 ページの「メタデータ要素の作成または編集」 ● 166 ページの「メタデータ要素の操作」
3.	**別名セットの操作**。別名セットを使用すると、1 つのメンバーに対して複数の別名を指定できます。別名セットは、特定の名前付き別名セットをメンバー名にマッピングします。	● 170 ページの「別名セットの操作」 ● 172 ページの「別名セットの編集」 ● 175 ページの「別名セットの管理」
4.	**階層の作成**。Essbase Studio では、階層を使用して次元をモデリングします。任意の次元を使用して、レポート用のデータを集計できます。階層のレベルは、次のいずれかの方法で結合されている異なるソース・データベース・テーブルから取得できます。 ● リレーショナル・データ・ソースの物理結合単位。 ● Essbase Studio のミニスキーマで作成された結合単位。	● 180 ページの「標準階層およびメジャー階層の作成」 ● 189 ページの「カレンダ階層の作成」 ● 197 ページの「階層の編集」
5.	**キューブ・スキーマの作成または編集**。キューブ・スキーマでは、階層、メジャーおよびメジャー階層を選択し、階層にある個々のメンバーのデータ・ロードの上書きを指定します。また、キューブ・スキーマから Essbase モデルを作成するためのオプションもあります。	● 199 ページの「キューブ・スキーマの作成または編集」 ● 206 ページの「既存のキューブ・スキーマからの Essbase モデルの作成」

手順	タスク	詳細情報
6.	**Essbase プロパティの操作**。 カスタマイズした Essbase アプリケーションを作成するには、Essbase のモデル・プロパティ、次元プロパティおよびメンバー・プロパティを編集します。	• 207 ページの「モデル・プロパティ」 • 224 ページの「次元のプロパティ」 • 245 ページの「メンバーのプロパティ」 • 280 ページの「Essbase モデル・プロパティの操作」
7.	**キューブの配置**。キューブ・スキーマ・ウィザードを使用して構築した Essbase モデルは、Essbase サーバーへのキューブの配置に使用されます。キューブ配置は、アウトラインを構築して、アプリケーションおよびデータベースにデータをロードするためのロード・オプションを設定するプロセスです。構築するアウトラインは、管理サービス・コンソールで編集できます。作成したアプリケーションおよびデータベースには、Oracle Hyperion Smart View for Office からアクセスできます。	• 295 ページの「キューブの配置」
8.	**配置したアプリケーションの表示およびキューブ・リンケージの更新**。Essbase モデルの配置履歴を表示して、Essbase サーバーの様々なインスタンスに配置されたすべての子キューブのリストを表示できます。キューブ・リンケージは、Essbase キューブに格納される次の情報を意味します: • キューブの配置に使用される Essbase Studio サーバーとポート • キューブのベースとなったモデル • キューブ構築の基になったデータ・ソースのログイン情報	• 307 ページの「配置履歴の表示」 • 308 ページの「キューブ・リンケージの更新」
9.	**ドリルスルー・レポートの作成および操作**。ドリルスルー・レポートを使用して、外部リレーショナル・データ・ソースから直接取得したデータを表示するスプレッドシート・レポートを作成します。	• 318 ページの「ドリルスルー・レポートの操作」 • 319 ページの「ドリルスルー・レポートの作成」 • 340 ページの「ドリルスルー・レポートと Essbase モデルの関連付け」

Essbase Studio ユーザー・インタフェース

Essbase Studio ユーザー・インタフェースは、次の固定できる 3 つのメイン領域で構成されます:

- 「データ・ソース・ナビゲータ」(24 ページの「データ・ソース・ナビゲータ」で説明)
- 作業領域(25 ページの「作業領域」で説明)
- 「メタデータ・ナビゲータ」(25 ページの「メタデータ・ナビゲータ」で説明)

さらに、コンソール・メッセージが表示または固定されるように選択できます。

データ・ソース・ナビゲータ

「データ・ソース・ナビゲータ」は、デフォルトで Essbase Studio コンソールの右ペインに表示され、作成した接続の宛先の物理データ・ソースがリストされます。この構造は、物理ツリーと呼ばれます。ここから、データ・ソース接続を作成するための「接続ウィザード」を起動できます。各データ・ソース接続について、関連付けられているミニスキーマも表示されます。ミニスキーマとは、1 つ以上のデータ・ソース接続から選択したテーブルをグラフィカルに表示したものです。ミニスキーマは、データ・ソース接続の作成時に作成したり、後で作成したりできます。

データ・ソースに対して実行できるその他のタスクは次のとおりです:

- データ・ソースにある選択したテーブルのサンプル・データの表示
- データ・ソース、テーブルおよび列の基本プロパティの表示
- Microsoft SQL Server データ・ソースにあるテーブルのわかりやすい名前の表示
- データ・ソース接続の削除
- データ・ソース・リストのリフレッシュ
- イントロスペクション、階層候補を識別するためのデータ・ソース接続の分析
- 既存のデータ・ソースの増分更新
- ユーザー定義テーブルの作成
- ミニスキーマの作成、編集および管理。

  ミニスキーマは、作成した接続の接続先となるデータ・ソースをグラフィカルに表示したものです。ミニスキーマは、データ・ソース接続の作成時に作成したり、後で作成したりできます。ミニスキーマには、データ・ソース接続のテーブルのサブセットまたはすべてのテーブルを含められます。また、ミニスキーマには、複数のデータ・ソースのテーブルを含めることもできます。

  - ソース、テーブルおよび列のプロパティなどのミニスキーマのプロパティの表示
  - 物理データ・ソースにあるテーブルのサンプル・データの表示
  - 1 つ以上のデータ・ソース接続のテーブルを追加または削除する場合のミニスキーマの編集
  - 結合の追加または編集
  - ミニスキーマの削除
  - ミニスキーマ・リストのリフレッシュ

データ・ソース接続およびミニスキーマの詳細は、次を参照してください:

- 第 4 章「データ・ソース接続」
- 第 5 章「ミニスキーマ」

作業領域

デフォルトで Essbase Studio コンソールの中央ペインにある作業領域は、メタデータ要素と、ソースおよびメタデータ要素のグラフィックの表示および操作に使用します。

作業領域に表示されるオブジェクトは次のとおりです:

- 129 ページの「ミニスキーマについて」で説明するミニスキーマ
- 第 11 章「Essbase プロパティ」で説明する Essbase モデル
- 第 15 章「系列」で説明する系列ビュー
- 第 16 章「ドリルスルー・レポート」で説明するドリルスルー・レポート
- 第 9 章「階層」で説明する階層エディタ
- 123 ページの「サンプル・データの表示」の第 4 章「データ・ソース接続」トピックで説明するサンプル・データ
- 307 ページの「配置履歴の表示」で説明する配置履歴

メタデータ・ナビゲータ

「メタデータ・ナビゲータ」は、デフォルトで Essbase Studio コンソールの左ペインに表示され、次の内容を含みます:

- データ・ソース接続を作成するときに物理データ・ソースから導出したメタデータ要素。

  新しいデータ・ソース接続を作成するたびに、接続したデータ・ソースの物理要素からメタデータ要素を作成できます。これらのメタデータ要素は、デフォルトで、データ・ソースのテーブルと列を模したフォルダ構造に表示されます。

  「メタデータ・ナビゲータ」では、複数のデータ・ソース接続からメタデータ要素を表示し、さらにサポートされているタイプのデータ・ソースからもオブジェクトを表示します。たとえば、リレーショナル、テキスト・ファイル、Oracle BI EE、Performance Management Architect など様々なソースからのデータにアクセスする必要があるとします。これらのソースへの接続を作成すると、各データ・ソースから導出したメタデータ要素が「メタデータ・ナビゲータ」のツリー構造に表示されます。デフォルトの各フォルダが作成されて、データ・ソースごとにメタデータ要素が格納されます。さらにオプションで、データ・ソース作成プロセス時にフォルダを作成し、メタデータ要素をツリー形式で詳細に整理できます。
- 「メタデータ・ナビゲータ」で、導出したメタデータ・アーティファクトから作成したメタデータ要素。

  導出したメタデータ・アーティファクトから、さらに次のようなメタデータ要素を作成できます:

  - フォルダ
  - 次元要素
  - 導出テキスト・メジャー

- 標準階層
- メジャー階層
- カレンダ階層
- テキスト・リスト
- キューブ・スキーマ
- Essbase モデル
- ドリルスルー・レポート

メタデータ要素は、Essbase で操作するキューブ・スキーマとモデルの作成に使用されます。Essbase モデルは、Essbase キューブの作成と配置に使用されます。

Essbase Studio へのログイン

Essbase Studio を開始すると、「ログイン」ダイアログ・ボックスが表示されます。

➤ Essbase Studio にログインするには:

1 **「サーバー」に、Essbase Studio サーバーがインストールされているコンピュータの名前を指定します。例:**

```
aspen3
```

2 **「ユーザー」名と「パスワード」を入力します。**

3 **オプション:「このユーザーを記憶する」を選択します。**

後続のログインでは、ドロップダウン・リストからユーザー名を選択できます。

4 **「ログイン」をクリックします**

ログインすると、Essbase Studio コンソール・ウィンドウのタイトル・バーに次の情報が表示されます:

- ユーザー名
- サーバー名
- ユーザーの役割

これで、Essbase Studio の操作を開始する準備ができました。

Essbase Studio サーバーへの再接続

Essbase Studio サーバーと Essbase Studio コンソール間の接続は、コンソールのアイドル状態が 1 時間(または第 2 章「管理」の説明に従って指定した長さの時間)を超えると、失われる可能性があります。

➤ Essbase Studio サーバーと Essbase Studio コンソール間の接続を復元するには:

1 **Essbase Studio コンソールで、「ツール」>「サーバーへの再接続」を選択して、「ログイン」ダイアログ・ボックスを表示します。**

「サーバー」フィールドと「ユーザー」フィールドは読取り専用です。

2 **「パスワード」にパスワードを入力して、「ログイン」をクリックします。**

2 管理

この章の内容

管理の概要 29
Essbase Studio カタログ・データベースの設定 29
Essbase Studio カタログおよびデータのアップグレード 31
Essbase Studio に対して SSL を使用可能にする 32
サーバー・プロパティ・ファイルの構成 33
ロギングの構成 57
Essbase Studio サーバーの起動ファイルの操作 57
UNIX システムでの ODBC の構成 59
JDBC ドライバの構成 66
Essbase Studio サーバーおよび Essbase Studio コンソールの開始と停止 67
UNIX 上におけるフォアグラウンドでの Essbase Studio サーバーの開始 68
Essbase Studio サーバーのコマンド 69
Essbase Studio カタログ・データベースのエクスポートとインポート 72
仮想メモリーの構成 78
配置履歴の消去 79
再ホストされた EPM System 製品への参照の更新 80

管理の概要

Oracle Hyperion Enterprise Performance Management System インストーラを使用して Essbase Studio をインストールした後、Oracle Hyperion Enterprise Performance Management System コンフィグレータを構成します。構成を変更する必要があるときは常に EPM System コンフィグレータを使用してください。ただし、一部の Essbase Studio 構成ファイルは手動で編集できます。

Essbase Studio カタログ・データベースの設定

Essbase Studio を使用する前に、Essbase Studio カタログ・データベース(カタログとも呼ばれる)を設定する場所を決定する必要があります。カタログ・データベースは、すべての Oracle Enterprise Performance Management System 製品に対して機能する 1 つのデータベース・インスタンス内に含めることができます。または、データベース・サーバー・コンピュータにカタログ用の特別な専用リレーショナル・データベースを作成することもできます。

カタログは、Essbase Studio のメタデータ・リポジトリです。Essbase Studio ユーザーが Essbase Studio でどのような種類のメタデータ要素(次元要素、階層、キューブ・スキーマおよび Essbase モデルなど)を作成しても、これらの要素のメタデータは Essbase Studioカタログに保管されます。

構成プロセス(インストール後)では、EPM System コンフィグレータを使用して、指定したカタログ・データベースを使用するように Essbase Studio を構成します。カタログ・データベースは、すべての EPM System 製品用の 1 つのデータベース・インスタンスの一部であっても、Essbase Studio カタログを保持するために特別に作成されたデータベースであっても構いません。

カタログ・データベースについての情報は、`MIDDLEWARE_HOME/user_projects/EPM_ORACLE_INSTANCE_NAME/BPMS/bpms1/bin` ディレクトリ内の `server.properties` に保管されます。このファイルには、このカタログ・データベースの名前、場所、ユーザー名および暗号化されたパスワードが含まれます。Essbase Studio を起動すると、`server.properties` が読み取られ、カタログ・データベースの場所およびセキュリティ・ログイン情報が特定されます。

どのサポートされているリレーショナル・データベースを使用しても、Essbase Studio カタログ・データベースを保持できます。

注意:

- 構成プロセスにおいて、カタログ・データベース・ユーザーを指定する必要があります。カタログ・データベース・ユーザーは、少なくとも Essbase Studio カタログ・データベースへの書込み権限を持っている必要があります。
- 各 Essbase Studio サーバー・インスタンスに独自のカタログ・データベースが必要です。
- Essbase Studio カタログ・データベースに Oracle を使用する場合は、カタログ・データベースへの接続に使用するユーザーに次の権限を設定することをお薦めします。

```
open_cursors=nnn SCOPE=MEMORY
```

nnn >= 300 です。

すべての EPM System 製品に対して 1 つのデータベース・インスタンスを使用している場合、このデータベースの設定の詳細は、Oracle Hyperion Shared Services のドキュメントを参照してください。

カタログに専用データベースを使用している場合、次の手順を完了します。

➤ Essbase Studio の専用カタログ・データベースを設定するには:

1 **RDBMS で、空のデータベース・スキーマを作成します。**

このデータベース・スキーマは、Essbase Studio カタログ・データベースとなります。

構成後は、これが `server.properties` ファイルの `catalog.db` プロパティで指定されるデータベースとなります。

2 **RDBMS で、カタログのデータベース・ユーザーに指定されるユーザーに少なくとも書込み権限を与えます。**

構成後、このユーザーは、`server.properties` ファイルの `catalog.username` および `catalog.password` プロパティで指定されるユーザーになります。このユーザーは、少なくとも `catalog.db` プロパティで指定されるデータベースへの書込み権限を持つ必要があります。これらのプロパティは、構成プロセスで設定されます。

カタログ・データベースに Shared Services リポジトリを使用している場合、同じ権限の要件が適用されます。

3 **Essbase Studio のインストール後に、EPM System コンフィグレータを実行して、カタログ・データベースに必要な情報を提供します。**

4 **構成後に、`MIDDLEWARE_HOME/user_projects/EPM_ORACLE_INSTANCE_NAME/BPMS/bpms1/bin` で、カタログ・データベースに関連する情報を確認します:**

- `catalog.db`-手順 1 で作成したカタログ・データベースの名前です。

  注： IBM DB2 または Microsoft SQL Server データベースのカタログ・データベースを構成した場合、データベース名にスキーマ名を付加する必要があります。たとえば、`catalog.db=esbstudio.dbo` のように指定します。ここで、`esbstudio` はデータベース名、`dbo` はスキーマ名です。

- `catalog.username`-カタログ・データベースに対して少なくとも書込み権限を持つユーザーです。40 ページの「catalog.username」を参照してください。
- `catalog.password`-カタログ・データベース・ユーザー用の暗号化されたパスワードです。40 ページの「catalog.password」を参照してください。

終了したら `server.properties` を閉じます。

Essbase Studio サーバーでは、起動時に `server.properties` ファイルが読み取られ、このカタログ・データベースの情報が取得されます。構成オプションの詳細は、33 ページの「サーバー・プロパティ・ファイルの構成」を参照してください。

Essbase Studio カタログおよびデータのアップグレード

Essbase Studio をリリース 11.1.1.x から現在のリリースに移行するには、アップグレードを実行します。リリース 11.1.2.x から現在のリリースに移行するには、メンテナンス・リリースを適用します。

Essbase Studio のアップグレードまたはメンテナンス・リリースの手順は EPM System の構成プロセスの一部です。詳細は、Oracle Enterprise Performance Management System インストールおよび構成ガイドを参照してください。ガイドに記載されている順序で、使用している環境に適用可能なタスクを実行してください。

構成後、Essbase Studio で次のタスクを実行してください:

1. アップグレードまたはメンテナンス・リリースの適用を実行するには、`reinit` コマンドを実行する必要があります。`reinit` コマンドの実行の詳細は、Oracle Enterprise Performance Management System インストールおよび構成ガイドを参照してください。
2. Essbase サーバーを移動した場合は、Essbase 接続が新しいサーバーの場所を指定するように、81 ページの「再ホストされた Essbase サーバーへの参照の更新」の説明に従って再ホスト手順を実行します。
3. Essbase Studio サーバーを移動した場合は、古いサーバーの場所を参照している配置済アプリケーションのために、308 ページの「キューブ・リンケージの更新」の説明に従ってキューブ・リンケージを更新します。
4. テキスト・ファイル・データ・ソースの場合:
   - EPM System コンフィグレータに指定されているデフォルトの `EPM_ORACLE_INSTANCE/BPMS/bpms1/datafiles` の場所を変更することで、以前のリリースからの場所、または Essbase Studio データ・ファイルのレプリケートされた場所を指定します。詳細は、Oracle Enterprise Performance Management System インストールおよび構成ガイドを参照してください。
   - 118 ページの「データ・ソース接続プロパティの編集」で説明されているように、任意のテキスト・ファイル接続の接続プロパティを編集して、テキスト・ファイルの新しい場所を指すようにします。
   - 構成後、テキスト・ファイルを別の場所に移動する場合は、`server.properties` ファイルの `server.datafile.dir` プロパティを編集し、テキスト・ファイルの新しい場所を指すように、任意のテキスト・ファイル接続の接続プロパティを編集します。41 ページの「server.datafile.dir」および118 ページの「データ・ソース接続プロパティの編集」をそれぞれ参照してください。
5. Performance Management Architect 次元サーバーを移動した場合は、データ・ソース接続プロパティの編集の説明に従って、古いサーバーの場所を指定する118 ページの「データ・ソース接続プロパティの編集」次元サーバー接続の接続プロパティを編集します。

Essbase Studio に対して SSL を使用可能にする

Essbase は、Secure Socket Layer (SSL)モードおよび非 SSL モードで動作するよう配置されています。SSL モードでは、Essbase サーバーと Essbase Studio サーバー間のすべての通信がデータ・セキュリティを確保するために暗号化されます。Essbase コンポーネントのデフォルトの配置では、自己署名の証明書がインストールされ、主にテスト目的で SSL 通信が使用可能になります。本番環境では SSL 対応の Essbase に、よく知られたサードパーティ認証局(CA)の証明書を使用することをお薦めします。詳細は、Oracle Enterprise Performance Management System インストールおよび構成ガイドを参照してください。

サーバー・プロパティ・ファイルの構成

インストール時に作成されたデフォルトの server.properties ファイルには、Essbase Studio の実行に必要なサーバー・プロパティが含まれています。

構成時に、EPM System コンフィグレータによって、選択したサーバー構成設定が Essbase Studio の server.properties ファイルに書き込まれます。

インストール後に、サーバー・プロパティを編集または追加できます。

注： server.properties ファイルは、MIDDLEWARE_HOME/user_projects/EPM_ORACLE_INSTANCE_NAME/BPMS/bpms1/bin にあります。

MIDDLEWARE_HOME については、Oracle Enterprise Performance Management System Installation and Configuration Readme を参照してください。

➤ server.properties ファイルを編集するには:

1. **Windows のメモ帳など、任意のテキスト・エディタを使用して、ファイルをテキスト・フォーマットで編集します。**
2. **ファイルの個別の行に各設定を入力します。**

   各行をセミコロンで終了する必要はありません。
3. **ファイル名が server.properties であることを確認します。**
4. **Essbase Studio の server ディレクトリにファイルを保存します。**
5. **サーバー・プロパティ・ファイルを変更したら、Essbase Studio サーバーを停止して再起動します。**

   Essbase Studio サーバーによって起動時に一度プロパティ・ファイルが読み込まれます。

server.properties で構成できるプロパティの例を次に示します。各プロパティの説明と例は、35 ページの「サーバー・プロパティ」を参照してください。

```
   catalog.url=
   database tag
   \://
   host
   :
   port
   /
server.css.URL=server.css.URL=http\://
   server
   .us.abcxyzco.com\:
   port
   /interop/framework/getCSSConfigFile
catalog.db=
   catalog database name

catalog.username=
   catalog user ID
```

```
catalog.password=
   encrypted catalog password

server.hss.bpmApplication=BPM\:
   BPM application ID

server.datafile.dir=
   path to flat file directory

server.essbase.streamingCubeBuilding=
   true|false

server.timeoutPeriod=
   number of seconds

server.queueSize=
   size of the task queue

server.threadCount=
   number of worker threads

   server.resourceCount=
   number of tasks that can be executed concurrently

server.sql.fetchSize=
   fetch size

server.tempDir=
   path to temporary directory

server.charset=
   charset

server.readLockTimeOut=
   number of seconds

server.writeLockTimeOut=
   number of seconds

server.essbase.TPTapi=
   true|false

server.essbase.disableDistinct=
   true|false

server.httpPort=
   port number

transport.port=
   port number

server.runInBackground=
   true|false

server.essbase.blindShare=
   true|false
```

```
oracle.jdbc.ReadTimeout=
   milliseconds


   data-source-type
   .pool.maxsize=
   maximum size of the connection pool


   data-source-type
   .cache.size=
   cache size

server.essbase.uniqueMemberFromCaptionBinding=
   true|false

catalog.pool.size=
   connection pool size for catalog connection
```

注： RDBMS によって異なりますが、この構文には例外が存在します。54 ページの「サーバー・プロパティ・ファイルの例」を参照してください。

サーバー・プロパティ

構成可能なサーバー・プロパティは次のとおりです:

- 36 ページの「catalog.url」
- 38 ページの「server.css.URL」
- 39 ページの「catalog.db」
- 40 ページの「catalog.username」
- 40 ページの「catalog.password」
- 40 ページの「server.hss.bpmApplication」
- 41 ページの「server.datafile.dir」
- 42 ページの「server.essbase.streamingCubeBuilding」
- 43 ページの「server.timeoutPeriod」
- 43 ページの「server.queueSize」
- 43 ページの「server.threadCount」
- 44 ページの「server.resourceCount」
- 44 ページの「server.sql.fetchSize」
- 45 ページの「server.tempDir」
- 45 ページの「server.charset」
- 46 ページの「server.readLockTimeOut」
- 46 ページの「server.writeLockTimeOut」

- 47 ページの「server.essbase.TPTapi」
- 47 ページの「server.essbase.disableDistinct」
- 48 ページの「server.httpPort」
- 48 ページの「transport.port」
- 49 ページの「server.runInBackground」
- 50 ページの「server.essbase.blindShare」
- 50 ページの「oracle.jdbc.ReadTimeout」
- 51 ページの「server.jdbc.teradata.properties」
- 51 ページの「data-source-type.cache.size」
- 52 ページの「data-source-type.pool.maxsize」
- 53 ページの「server.essbase.uniqueMemberFromCaptionBinding」
- 53 ページの「server.essbase.uniqueMemberFromCaptionBinding」
- 54 ページの「catalog.pool.size」

catalog.url

カタログ・データベースの URL。

構文

```
catalog.url=
database_tag
\://
host
:
port
/
```

`host` は、要件に応じてサーバー名またはサーバー名にドメイン名を加えた名前になります。

注意

- 次のデータベース・タグがサポートされます:
  - `oracle`-Oracle
  - `db2`-IBM DB2
  - `sqlserver`-Microsoft SQL Server
- カタログ・データベースが Oracle ID (SID)を使用する Oracle の場合、次のように Oracle SID が `catalog.url` パラメータに追加されます:

```
catalog.url=oracle\://
host
\:
port
```

```
\:
OracleSID
```

- カタログ・データベースが Oracle サービス名を使用する Oracle の場合、次のように Oracle サービス名が `catalog.url` パラメータに追加されます:

```
catalog.url=oracle\://
host
\:
port
/
OracleServiceName
```

- カタログ・データベースが Oracle RAC インスタンス内にある場合、`catalog.url` パラメータは次のようになります:

```
     catalog.url=
oracle\:// (DESCRIPTION=(LOAD_BALANCE=on)
(ADDRESS=(PROTOCOL=TCP)(HOST=
     host1
     )(PORT=
     port-of-host1

[ (ADDRESS=(PROTOCOL=TCP)(HOST=
     host2
     )(PORT=
     port-of-host2
     ))
[ (ADDRESS=(PROTOCOL=TCP)(HOST=
     hostN
     )(PORT=
     port-of-hostN
     )) ]...]
(CONNECT_DATA=(SERVICE_NAME=
     service
     )))
```

- カタログ・データベースが IBM DB2 の場合は、次の構文を使用してデータベース名が `catalog.url` パラメータに追加されます:

```
database tag
://
host
:
port
;databaseName=
database name
```

例

- SID を使用している Oracle:

```
catalog.url=oracle\://sequoia.efgco.com\:1521\:bpm
```

```
catalog.url=oracle\://sequoia\:1521\:bpm
```

- Oracle サービス名を使用している Oracle:

```
oracle://aspen123.us.xyzco.com\:1521/epmdb.us.xyzco.com
```

```
oracle://aspen123\:1521/epmdb.us.xyzco.com
```

- Oracle RAC:

```
catalog.url=oracle\://(DESCRIPTION=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)
(HOST=host1)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=host2)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=service)))
```

- IBM DB2:

```
catalog.url=db2\://cypress.abcco.com\:50000;databaseName=EPM_Db
```

```
catalog.url=db2\://cypress\:50000;databaseName=EPM_Db
```

- Microsoft SQL Server:

```
catalog.url=sqlserver\://aspen3.us.abcxyzco.com\:1433
```

```
catalog.url=sqlserver\://aspen3\:1433
```

特定の RDBMS の例については、54 ページの「サーバー・プロパティ・ファイルの例」を参照してください。

server.css.URL

Shared Services サーバーの URL です。

構文

```
server.css.URL=http\://
hostname
.
```

```
domain name
.com\:
port
/interop/framework/getCSSConfigFile
```

例

```
server.css.URL=http\://aspen3.us.abcxyzco.com\:28080/interop/framework/
getCSSConfigFile
```

catalog.db

Essbase Studio のメタデータ・リポジトリに設定されているリレーショナル・データベースの名前です(カタログ・データベースまたはカタログとも呼ばれます)。

構文

```
catalog.db=
catalog database name
```

注： カタログ・データベース名またはスキーマ名が数字で開始している場合、catalog.db のプロパティの設定時に、データベース名を引用符(" ")で囲む必要があります。

例

```
catalog.db=esbstudio
```

IBM DB2 ユーザーおよび Microsoft SQL Server ユーザーの場合、次の構文を使用して catalog.db パラメータにスキーマ名を含める必要があります:

```
catalog.db=
catalog database name
.
schema name
```

カタログ・データベース名またはスキーマ名が数字で開始している場合、データベース名が引用符(" ")で囲まれていることを確認してください。カタログ・データベース名の正しい定義と誤った定義の例を次に示します。

誤った例-catalog.db=123a

正しい例-catalog.db="123a"

誤った例-catalog.db=123a.user1

正しい例-catalog.db="123a".user1

誤った例-catalog.db=a123.1user

正しい例-catalog.db=a123."1user"

catalog.username

カタログ・データベースのユーザーのユーザー ID です。このユーザーには少なくともカタログへの書込みアクセス権必要です。

構文

```
catalog.username=
catalog user ID
```

例

```
catalog.username=root
```

catalog.password

catalog.username で指定されるユーザーの暗号化パスワードです。

注： 暗号化パスワード文字列は、EPM System コンフィグレータにより生成されます。このプロパティを編集する場合は、暗号化パスワード文字列を使用する必要があります。

構文

```
catalog.password=
encrypted catalog password
```

例

```
catalog.password=A627FC9A6DEA834C1FA777217871D09E
```

server.hss.bpmApplication

EPM System の構成プロセスで Shared Services により Essbase Studio サーバーに割り当てられるアプリケーション ID 番号です。

EPM System コンフィグレータでは、このプロパティは Essbase Studio サーバーの登録時に自動的に設定されます。Shared Services の別のインスタンスを指定するように server.hss.bpmApplication のプロパティを変更するには、EPM System コンフィ

グレータを実行して、新しい Shared Services のインスタンスを登録する必要があります。

構文

```
server.hss.bpmApplication=BPM\:
application identifier
```

例

```
server.hss.bpmApplication=BPM\:29696
```

server.datafile.dir

Essbase Studio Server のデータ・ソースとして機能するフラット・ファイルのルート・ディレクトリを定義します。テキスト・ファイルのデフォルトの格納場所は次のとおりです:

```
EPM_ORACLE_INSTANCE
/BPMS/bpms1/datafiles
```

フラット・ファイルがある場所へのフル・パスを指定して、デフォルトのディレクトリを上書きできます。

構文

```
server.datafile.dir=
path to flat file directory
```

例

- **Windows:**

  テキスト・ファイルはローカルの Essbase Studio Server マシンに格納されます:

  ```
  server.datafile.dir=C:\\EssbaseStudio\\text_file_sources
  ```

  テキスト・ファイルは UNC (汎用命名規則)のパスに格納されます:

  ```
  server.datafile.dir=\\\\svr33\\cedar5\\EssbaseStudio\\text_file_sources
  ```

  円記号(バックスラッシュ)(\)は特殊文字であるため、その前に円記号(バックスラッシュ)(エスケープ・シーケンス)を付ける必要があります。

- **UNIX:**

```
server.datafile.dir=/vol1/cedar5/EssbaseStudio/text_file_sources
```

注意

- テキスト・ファイルのデータ・ソース・ディレクトリのパスの長さには、次の制限があります:
  - ネイティブ・モード: 121 バイト
  - Unicode モード: 1028 バイト
- `server.datafile.dir` が変更された場合、変更を有効にするには Essbase Studio サーバーを再起動する必要があります。

server.essbase.streamingCubeBuilding

TRUE に設定すると、Essbase キューブ配置がストリーミング・モードで実行されます。

デフォルト値は FALSE で、この場合、Essbase Studio は非ストリーミング・モードで動作します。

構文

```
server.essbase.streamingCubeBuilding=
true|false
```

注意

非ストリーミング・モードとは、キューブ配置時に Essbase Studio サーバーが ODBC 接続を使用して外部データ・ソースをクエリーすることを意味します。

ストリーミング・モードとは、キューブ配置時に Essbase Studio サーバーが外部データ・ソースを直接クエリーすることを意味します。

`server.essbase.streamingCubeBuilding` が FALSE に設定されている場合、キューブ配置ウィザードで配置時にストリーミング・モードを選択できます(297 ページの「キューブ配置のための接続情報の提供」の手順 5 を参照)。

IBM DB2 ユーザー: 接続ウィザードの「接続の定義」ページで、データ・ソースに「暗号化なし」以外の認証メソッドを指定した場合、`server.essbase.streamingCubeBuilding` プロパティを true に設定し、ストリーミング・モードでの配置を有効にする必要があります。認証メソッドの指定の詳細は、91 ページの「リレーショナル・ソースの接続パラメータの定義」を参照してください。

例

```
server.essbase.streamingCubeBuilding=true
```

server.timeoutPeriod

Essbase Studio コンソールがアイドル状態を保つことができる総時間数(秒)です。この時間が経過すると Essbase Studio サーバーへの接続が解除されます。

デフォルト値は、3600 秒(1 時間)です。

構文

```
server.timeoutPeriod=
number of seconds
```

例

```
server.timeoutPeriod=7200
```

server.queueSize

キュー・サイズは、タスク数で設定します。このキューには、Essbase Studio サーバーでの実行を待機しているタスクが保持されます。

デフォルトのキュー・サイズは、200 タスクです。

構文

```
server.queueSize=
size of the task queue
```

例

```
server.queueSize=250
```

server.threadCount

Essbase Studio サーバーに割り当てられるワーカー・スレッドの数。

構文

```
server.threadCount=
number of worker threads
```

注意

次のワーカー・スレッドがあります:

- リスナー。Essbase Studio クライアントからの要求があるかどうかリスニングするスレッド。

- クライアントからタスクを取得するスレッド。
- タスクそのもの。Essbase Studio サーバーで実行されるタスク。

デフォルトのスレッド数は 30 です。

注： Essbase Studio クライアントからタスクをリスニングし、取得するワーカー・スレッド数を決定する際、ワーカー・スレッド数に44 ページの「server.resourceCount」で設定されたリソース数よりも大きい数値を設定する必要があります。

例

```
server.threadCount=40
```

server.resourceCount

リソースの最大数を設定します。リソース数によって、Essbase Studio サーバーが同時に実行できるタスクの数が定義されます。

デフォルトのリソース数は 20 です。

構文

```
server.resourceCount=
number of tasks that can be executed concurrently
```

例

```
server.resourceCount=35
```

server.sql.fetchSize

JDBC ドライバのフェッチ・バッファのデフォルト・サイズのサイズをレコード数で再定義します。

Essbase Studio サーバーのデフォルトのバッファ・サイズは 1000 レコードです。

構文

```
server.sql.fetchSize=
fetch size in number of records
```

例

```
server.sql.fetchSize=500
```

server.tempDir

ルール・ファイルやエラー・ファイルなど、Essbase Studio サーバーで作成された一時ファイルのディレクトリを指定します。このディレクトリは必ず存在している必要があります。

デフォルト値によって、ディレクトリ./ess_japihome/data への相対パスが定義されます。これは、Essbase Studio のインストール時に作成されます。

構文

```
server.tempDir=
path to temporary directory
```

キューブの配置時に「名前が長すぎます」というエラーが戻される場合、server.tempDir プロパティのエントリを server.properties ファイルに追加して、より短いディレクトリ・パスを指定します。例:

```
server.tempDir=C:\\studiotemp
```

例

```
server.tempDir=$USER_HOME/Temp
```

server.charset

サーバー・ログ・ファイルにあるすべてのメッセージを変換するために Essbase Studio サーバーによって使用される文字セットを指定します。

デフォルトの文字セットは utf-8 です。

構文

```
server.charset=
charset
```

例

```
server.charset=US-ASCII
```

server.readLockTimeOut

Essbase Studio カタログ・データベースの情報の読取り要求を行った場合に、プロセスが待機するタイム・アウトまでの秒数を指定します。

デフォルト値は 120 秒です。

このプロパティは、46 ページの「server.writeLockTimeOut」で説明されている `server.writeLockTimeOut` プロパティと連動します。

構文

```
server.readLockTimeOut=
number of seconds
```

注意

複数のユーザーが同じ Essbase Studio カタログ・データベースにアクセスしている場合、ユーザーのアクションによっては、別のユーザーがカタログ・データベースにアクセスできなくなる場合があります。そのようなユーザーのアクションは次のとおりです:

- メタデータ要素の作成
- メタデータ要素の変更
- データ探索(テーブルの選択やメタデータ要素の作成などデータ・ソース接続の作成、またはデータ・ソースの増分更新など)

前述のリストにあるアクションのいずれかをユーザーが実行すると、Essbase Studio サーバーにより、他のユーザーの同じカタログ・データベースへの要求が、`server.readLockTimeOut` で設定した期間ブロックされます。この期間が経過してもサーバーが要求を実行できない場合、要求の処理が中断したことをユーザーに通知するメッセージが表示されます。

Essbase Studio がキャッシュからオブジェクトをフェッチすることで要求を実行できる場合は、その処理を試みます。たとえば、ユーザーが表示する階層を選択して、その階層がキャッシュにある場合、Essbase Studio はユーザーにその階層を表示します。

例

```
server.readLockTimeOut=90
```

server.writeLockTimeOut

Essbase Studio カタログ・データベースへの書込み要求を行うときに、プロセスの待機がタイム・アウトになるまでの秒数を指定します。

デフォルト値は 120 秒です。

このプロパティは、46 ページの「server.readLockTimeOut」で説明されているように、`server.readLockTimeOut` プロパティとともに動作します。

構文

```
server.writeLockTimeOut=
number of seconds
```

注意

46 ページの「server.readLockTimeOut」の注意セクションを参照してください。

例

```
server.writeLockTimeOut=150
```

server.essbase.TPTapi

Teradata ユーザーのみを対象とします。

TRUE に設定すると、Teradata Parallel Transporter API が使用可能になり、ODBC 接続と比較してより高速なデータ・ロードのパフォーマンスを得られます。

デフォルト値は FALSE です。

Teradata Parallel Transporter の詳細は、『Oracle Essbase SQL インタフェース・ガイド』を参照してください。

構文

```
server.essbase.TPTapi=
true|false
```

例

```
server.essbase.TPTapi=true
```

server.essbase.disableDistinct

TRUE に設定すると、ユーザーはメンバー・ロード・クエリーの DISTINCT フィルタを使用不可にできます。

デフォルト値は FALSE です。

構文

```
server.essbase.disableDistinct=
true|false
```

注意

- デフォルトでは、メンバー・ロードの実行時に、Essbase Studio サーバーによって DISTINCT キーワードが追加され、重複レコードが除外されます。
- **IBM DB2 ユーザー:** SELECT DISTINCT ステートメントでの LONG VARCHAR データ型の処理における IBM DB2 の制限により、キューブの配置が失敗します。これを回避するには、`server.essbase.disableDistinct` を true に設定します。

例

```
server.essbase.disableDistinct=true
```

server.httpPort

Essbase Studio サーバーがリスニングする HTTP ポート。HTTP ポートは、Essbase Studio サーバーが Performance Management Architect と通信を行うために使用されます。

デフォルトの HTTP ポート番号は 12080 です。

Essbase Studio サーバーのコンピュータ上で稼動する他のアプリケーションまたはプログラムとポートが競合する場合は、競合するアプリケーション側のポート番号を変更することをお薦めします。

ただし、`server.httpPort` の設定の変更が必要な場合は、`server.httpPort` プロパティのエントリを `server.properties` ファイルに追加することで変更が可能になります。

構文

```
server.httpPort=
port number
```

例

```
server.httpPort=12080
```

transport.port

Essbase Studio サーバーがリスニングする TCP ポートを指定します。

デフォルトのポート番号は 5300 です。

Essbase Studio サーバーのコンピュータ上で稼動する他のアプリケーションまたはプログラムとポートが競合する場合は、`transport.port` プロパティのエントリを `server.properties` ファイルに追加することで Essbase Studio サーバーのポート番号を変更できます。

構文

```
transport.port=
port number
```

例

この例では、ポート番号がポート 3000 に変更されます。

```
transport.port=3000
```

注意

ポートの競合が発生した場合は、EPM System コンフィグレータを実行して、Essbase Studio サーバーへのポートの割当てを変更することをお薦めします。

EPM System コンフィグレータを実行せずに `transport.port` プロパティを手動で変更する場合は、コンソールへのログイン時に Essbase Studio サーバーのポート番号を入力する必要があります。

たとえば、aspen という名前のマシン上で稼働する Essbase Studio サーバーのインスタンスについて、ポートの割当てをデフォルトからポート番号 1234 に変更した場合は、Essbase Studio コンソールへのログイン時に「ログイン」ダイアログ・ボックスの「サーバー」フィールドに次のように入力します:

```
aspen:1234
```

server.runInBackground

true に設定されている場合、Essbase Studio サーバーはバックグラウンドで実行されます。サーバー・コンソールのコマンドの表示およびサーバー・コンソールのヒントはブロックされ、ユーザーはサーバー・コンソールでコマンドを入力できません。

デフォルト値は false です。ただし、UNIX 上で Essbase Studio サーバーを実行する場合、このプロパティはインストール時に自動的に true に設定されます。

構文

```
server.runInBackground=true|false
```

例

```
server.runInBackground=true
```

注： UNIX 上で Essbase Studio サーバーをフォアグラウンドで実行する方法については、68 ページの「UNIX 上におけるフォアグラウンドでの Essbase Studio サーバーの開始」を参照してください。

server.essbase.blindShare

`server.essbase.blindShare` プロパティを true に設定すると、重複メンバーを共有メンバーとして追加できるようになります。

また、true に設定すると、次のような特殊なケースにおいての、重複メンバーの共有メンバーとしての追加が可能になります:

- 重複メンバーがプライマリ・メンバーと同じ列からのメンバーではない場合
- 重複メンバーがレベル 0 以外のクラスからのメンバーである(ただし、アウトライン内ではレベル 0 にある)場合

デフォルト値は FALSE です。

重複メンバーを共有メンバーとして追加するには、`server.essbase.blindShare` プロパティのエントリを `server.properties` ファイルに追加します。

構文

```
server.essbase.blindShare=true|false
```

例

```
server.essbase.blindShare=true
```

oracle.jdbc.ReadTimeout

ソケットからの読取り時の読取りタイムアウトです。ミリ秒単位で指定します。

Oracle データ・ソースを使用していてタイムアウトが頻繁に発生する場合は、`server.properties` ファイルにエントリを追加することで、このオプションの値を増加できます。

デフォルト値は 600000 ミリ秒です。

構文

```
oracle.jdbc.ReadTimeout=
milliseconds
```

例

```
oracle.jdbc.ReadTimeout=900000
```

server.jdbc.teradata.properties

Teradata データ・ソースへの接続プロパティを設定する場合。

構文

```
server.jdbc.teradata.properties=property1=value1,property2=value2...
```

例

```
server.jdbc.teradata.properties=charset=UTF16
```

注： 文字セット UTF16 を指定します

data-source-type.cache.size

接続プール内の物理接続の内部キャッシュ。内部キャッシュは、常に開いている物理接続から構成されます。一部またはすべての接続を同時に使用できます。

`data-source-type.cache.size` プロパティと `data-source-type.pool.maxsize` プロパティが連携して、データ・ソース接続の接続プーリングを制御します。接続プールにより、複数のタスクが同じデータ・ソース接続を同時にクエリーできます。

接続のキャッシュのデフォルト・サイズは 5 です。

52 ページの「data-source-type.pool.maxsize」も参照してください。

構文

```
data-source-type.
cache.size=
number of physical connections in connection pool
```

`data-source-type` に対して有効な値は次のとおりです:

- Oracle - `oracle`
- IBM DB2 - `db2`
- Microsoft SQL Server - `sqlserver`
- MySQL - `mysql`
- Netezza - `netezza`
- Oracle BI EE - `obiee`
- Teradata - `teradata`

データ・ソース・タイプ別の例

```
oracle.cache.size=15
```

```
db2.cache.size=10

sqlserver.cache.size=15

mysql.cache.size=20

netezza.cache.size=25

obiee.cache.size=5

teradata.cache.size=15
```

data-source-type.pool.maxsize

接続プールの最大接続数。必要な接続数が data-source-type.cache.size で指定した接続数を超えている場合、Essbase Studio サーバーは、data-source-type.pool.maxsize で指定した値に達するまで、一時的な接続を開きます。必要な接続数が減ると、一時接続が破棄されます。

data-source-type.pool.maxsize プロパティと data-source-type.cache.size プロパティが連携して、データ・ソース接続の接続プーリングを制御します。接続プールにより、複数のタスクが同じデータ・ソース接続を同時にクエリーできます。

接続プール内の接続の最大数のデフォルトは 10 です。

51 ページの「data-source-type.cache.size」も参照してください。

構文

```
data-source-type.
pool.maxsize=
maximum size of the connection pool
```

data-source-type に対して有効な値は次のとおりです:

- Oracle - oracle
- IBM DB2 - db2
- Microsoft SQL Server - sqlserver
- MySQL - mysql
- Netezza - netezza
- Oracle BI EE - obiee
- Teradata - teradata

データ・ソース・タイプ別の例

```
Oracle.pool.maxsize=20
```

```
db2.pool.maxsize=20

sqlserver.pool.maxsize=20

mysql.pool.maxsize=20

netezza.pool.maxsize=20

obiee.pool.maxsize=10

teradata.pool.maxsize=20
```

server.essbase.uniqueMemberFromCaptionBinding

true に設定されている場合、Essbase にメンバーをロードする際に、一意のメンバー名がキャプション・バインディングから取得されます。デフォルトは false です。

キー・バインディングとキャプション・バインディングが異なり、`uniqueMemberFromCaptionBinding` が false に設定されている場合、ドリルスルー・レポートが正しく機能するには、「Essbase モデル・プロパティ」ダイアログ・ボックスの「全般」タブで「重複するメンバー名のサポート」チェック・ボックスが選択されている必要があります。

構文

```
server.essbase.uniqueMemberFromCaptionBinding=
true
|
false
```

注意

`server.essbase.uniqueMemberFromCaptionBinding=false` の場合にキューブを配置する際、Essbase にメンバーをロードするときには次の動作がデフォルトです:

- 一意のメンバー名のアウトラインの場合、メンバー名はキー・バインディング式から取得されます。
- 重複するメンバー名のアウトラインの場合、メンバー名はキャプション・バインディング式から取得されます。
- 一意および重複するメンバー名のアウトラインの場合は、次のようになります:
  - メンバー名の変換(接頭辞/接尾辞など)は指定どおりに行われます
  - データ・ロード最適化(別名の最適化)は指定どおりに行われます

server.essbase.uniqueMemberFromCaptionBinding=true の場合に次元要素のキャプション・バインディングとキー・バインディングが異なると、この次元要素を含むキューブからドリルスルー・レポートを実行できません。

例

```
server.essbase.uniqueMemberFromCaptionBinding=true
```

catalog.pool.size

Essbase Studio カタログ接続の接続プール・サイズ。

このプロパティは、Essbase Studio カタログ・インスタンスに接続する際、Essbase Studio サーバーがリレーショナル・データベースに対して開く物理接続の数を定義します。

デフォルトのプール・サイズは 10 です。

構文

```
catalog.pool.size=
connection pool size for Essbase Studio catalog connection
```

例

```
catalog.pool.size=10
```

サーバー・プロパティ・ファイルの例

次のような例があります:

- 54 ページの「Oracle の例」
- 55 ページの「IBM DB2 の例」
- 56 ページの「Microsoft SQL Server の例」

Oracle の例

Oracle ID(SID)を使用している場合は、次の構文を使用して catalog.url パラメータに Oracle SID を追加する必要があります:

```
catalog.url=oracle\://
host
\:
port
\:
OracleSID
```

Oracle サービス名を使用している場合は、次の構文を使用して、Oracle サービス名を catalog.url パラメータに追加する必要があります:

```
catalog.url=oracle\://
host
\:
port
/
OracleServiceName
```

Oracle データベースと Oracle SID を使用した server.properties ファイルの構成例を次に示します:

```
   catalog.url=oracle\://sequoia.xyzco.com\:1521\:bpm
server.css.URL=http\://spruce.xyzco.com\:28080/interop/framework/getCSSConfigFile
catalog.db=esbstudio
catalog.username=root
catalog.password=A627FC9A6DEA834C1FA777217871D09E
server.hss.bpmApplication=BPM\:29696
server.datafile.dir=./data/flatfiles
server.essbase.streamingCubeBuilding=true
server.timeoutPeriod=7200
server.queueSize=250
server.threadCount=40
server.resourceCount=30
server.sql.fetchSize=100
server.tempDir=C:/Windows/Temp
server.charset=windows-1251
server.readLockTimeOut=90
server.writeLockTimeOut=150
oracle.pool.maxsize=20
oracle.cache.size=15
server.essbase.uniqueMemberFromCaptionBinding=true
```

IBM DB2 の例

DB2 データベースをカタログ・データベースとして使用する場合、次の構文を使用してデータベース名を catalog.url パラメータに追加する必要があります:

```
database tag
://
hostname
:
port
;databaseName=
database name
```

また、DB2 ユーザーは次の構文を使用して、スキーマ名を catalog.db パラメータに追加する必要もあります:

```
catalog.db=
catalog database name
.
schema name
```

DB2 データベースを使用した server.properties の構成例を次に示します:

```
   catalog.url=db2\://cypress\:50000;databaseName=BPM_Db
server.css.URL=http\://aspen2.xyzco.com\:28080/interop/framework/getCSSConfigFile
catalog.db=bpm.ROOT
catalog.username=root
catalog.password=A627FC9A6DEA834C1FA777217871D09E
server.hss.bpmApplication=BPM\:29696
server.datafile.dir=$USER_HOME/data/FlatFileDir
server.essbase.streamingCubeBuilding=true
server.timeoutPeriod=7200
server.queueSize=250
server.threadCount=50
server.resourceCount=35
server.sql.fetchSize=500
server.tempDir=$USER_HOME/Temp
server.readLockTimeOut=150
server.writeLockTimeOut=150
db2.pool.maxsize=20
db2.cache.size=15
server.essbase.uniqueMemberFromCaptionBinding=true
```

Microsoft SQL Server の例

Microsoft SQL Server ユーザーの場合、次の構文でスキーマ名を catalog.db パラメータに含める必要があります:

```
catalog.db=
catalog database name
.
schema name
```

SQL Server データベースを使用した server.properties の構成例を次に示します:

```
   catalog.url=sqlserver\://aspen3.us.abcxyzco.com\:1433
server.css.URL=http\://pine4.us.abcxyzco.com\:28080/interop/framework/
getCSSConfigFile
catalog.db=esbstudio.root
catalog.username=root
catalog.password=A627FC9A6DEA834C1FA777217871D09E
server.hss.bpmApplication=BPM\:29696
server.datafile.dir=C:/FlatFileSource
server.essbase.streamingCubeBuilding=true
server.timeoutPeriod=7200
```

```
server.queueSize=200
server.threadCount=35
server.resourceCount=25
server.sql.fetchSize=500
server.tempDir=C:/Windows/Temp
server.charset=windows-1251
server.readLockTimeOut=150
server.writeLockTimeOut=150
sqlserver.pool.maxsize=20
sqlserver.cache.size=15
server.essbase.uniqueMemberFromCaptionBinding=true
```

ロギングの構成

Essbase Studio のロギングでは、Oracle Diagnostic Logging (ODL)フレームワークが使用されます。

リリース 11.1.2 以降、ロギングに関連する次の Essbase Studio サーバー・プロパティは非推奨になりました:

- `logger.file`
- `logger.limit`
- `logger.count`
- `com.hyperion.cp.handlers=com.hyperion.cp.util.LoggerFileHandler`

プロパティ `com.hyperion.cp.level` は、ODL に付属の `logging.xml` ファイルで設定されるようになり、Essbase Studio の `server.properties` ファイルでは設定されなくなりました。

Essbase Studio およびその他の EPM System 製品のロギングの構成の詳細は、Oracle Technology Network で入手可能な Oracle Enterprise Performance Management System Installation and Configuration Troubleshooting Guide を参照してください。

Essbase Studio サーバーの起動ファイルの操作

Essbase Studio サーバーの起動ファイルには、Essbase Studio サーバーを起動するために必要なコマンドが含まれています。

構成プロセスの際、EPM System コンフィグレータによって、Essbase Studio がインストールされているコンピュータに関する情報が起動ファイルに書き込まれます。その結果、Essbase Studio サーバーを正しく起動できます。起動ファイルを手動で編集しないでください。

起動ファイルは、デフォルトで次の場所にインストールされます:

```
MIDDLEWARE_HOME
/user_projects/
EPM_ORACLE_INSTANCE_NAME
/BPMS/bpms1/bin
```

Windows では、ファイル名は startBPMS_bpms1_Server.bat です。UNIX では、startBPMS_bpms1_Server.sh です。

例

Windows 2003 オペレーティング・システムの startBPMS_bpms1_Server.bat ファイルの例を次に示します:

```
   pushd "C:\Oracle\Middleware\user_projects\epmsystem1\bin"
call setEnv.bat
set JAVA_OPTIONS=-DESSBASE_STUDIO_INSTANCE="C:\Oracle\Middleware\user_projects
 \epmsystem1/BPMS\bpms1" -DsuppressAPSProductInfo=true
set JAVA_HOME=C:\Oracle\Middleware\jdk160_11\jre
call %EPM_ORACLE_HOME%/products/Essbase/EssbaseStudio/Server/startServer.bat
```

Oracle ユーザーの場合

Oracle のカタログ・データベースまたはリレーショナル・ソースを使用していて、Essbase Studio サーバーから Oracle RDBMS への物理接続を開けない場合は、タイム・ゾーン構成に問題がある可能性があります。

➤ タイム・ゾーン構成の問題を解決するには:

1 **Essbase Studio サーバーを起動するバッチ・ファイルまたはシェルスクリプトを探します。**

- 11.1.1.x の場合、このファイルは次の場所にあります:

```
HYPERION_HOME
/products/Essbase/EssbaseStudio/Server/startServer.bat|sh
```

- 11.1.2.1.x の場合、このファイルは次の場所にあります:

```
EPM_ORACLE_HOME
/products/Essbase/EssbaseStudio/Server/startServer.bat|sh
```

- 11.1.2.2.x の場合、このファイルは次の場所にあります

```
MIDDLEWARE_HOME
/user_projects/
EPM_ORACLE_INSTANCE_NAME
/BPMS/bpms1/bin/start_BPMS_bpms1_Server.bat|sh
```

2 **ファイルを手動で更新し、JVM パラメータ** user.timezone **を追加します。**

たとえば、元のエントリが次のようであるとします:

```
%JAVA_HOME%\bin\java" -Xms128m -Xmx768m %JAVA_OPTIONS% -jar server.jar
```

更新後、このエントリは次のとおりです:

```
        %JAVA_HOME%\bin\java" -Xms128m -Xmx768m -Duser.timezone="2:00"  %JAVA_OPTIONS
% -jar server.jar
```

3 Essbase Studio サーバーを再起動します。

UNIX システムでの ODBC の構成

Windows 環境では、ODBC 管理者を使用してデータ・ソースへの ODBC 接続を構成します。UNIX システムでは ODBC 管理者を使用できないため、手動で構成を行う必要があります。AIX、HP-UX、Solaris および Linux で ODBC を構成するには、次の手順を実行する必要があります。

- 60 ページの「.odbc.ini へのシンボリック・リンクの作成」

  Essbase Studio でデータ・ソースとして使用されるすべてのデータベースに対して、.odbc.ini へのシンボリック・リンクを作成します。

- 60 ページの「odbcinst.ini へのドライバ記述子の追加」

  UNIX 上で MySQL Enterprise、Netezza または Teradata をデータ・ソースとして使用している場合は、ODBC ドライバをインストールして odbcinst.ini にドライバのエントリを追加する必要があります。サポートされているその他のデータ・ソースの場合は、ODBC ドライバがインストールされ、odbcinst.ini に構成されています。

- 61 ページの「odbc.ini ファイルへの DSN の追加」

  Essbase Studio でデータ・ソースとして使用されるすべてのデータベースに対して、odbc.ini に ODBC 接続を構成する必要があります。

注： **Oracle BI EE ユーザー:** UNIX および Linux 用の ODBC 構成の手順は、Oracle Enterprise Performance Management System インストールおよび構成ガイドに記載されています。

注： **MySQL ユーザー:** MySQL Enterprise バージョンがサポートされています。MySQL Community バージョンはサポートされていません。

注： Essbase Studio では、スペースやピリオド(.)などの特殊文字を含むデータ・ソース・テーブル名および列名はサポートされていません。サポートされていない文字の完全なリストは、付録 B「Essbase Studio の命名規則」を参照してください。

フラット・ファイル・データ・ソース: Essbase Studio では、フラット・ファイル・データ・ソースに ODBC 構成は不要です。

.odbc.ini へのシンボリック・リンクの作成

➤ .odbc.ini へのシンボリック・リンクを作成するには:

1 **UNIX ユーザー・ホーム・ディレクトリから**$ESSBASEPATH/bin/.odbc.ini **へのシンボリック・リンクを作成します; 次に例を示します:**

```
ln -s
$EPM_ORACLE_HOME
/common/ODBC/Merant/x.x/odbc.ini /home/myfolder/.odbc.ini
```

2 **MySQL、Netezza および Teradata の場合:** 60 ページの「odbcinst.ini へのドライバ記述子の追加」を続行します。

odbcinst.ini へのドライバ記述子の追加

EPM System とともにインストールされる ODBC ドライバの場合、ドライバ記述子は odbcinst.ini に自動的に追加されます。EPM System によってインストールされないドライバの場合は、ドライバ記述子を odbcinst に手動で追加する必要があります。

➤ **MySQL、Netezza および Teradata の場合:** odbcinst.ini にドライバ記述子を追加するには:

1 **適切な** odbcinst.ini **を探します。**

32 ビットの場合:

```
$EPM_ORACLE_HOME
/common/ODBC/Merant/6.1/odbcinst.ini
```

64 ビットの場合:

```
$EPM_ORACLE_HOME
/common/ODBC-64/Merant/6.1/odbcinst.ini
```

2 **適切なドライバ記述子エントリを追加します。**

例:

MySQL

```
[ODBC Drivers]
MySQL ODBC 3.51 Driver =Installed

...

[MySQL ODBC 3.51 Driver]
Description = ODBC 3.51 for MySQL
DRIVER = /usr/lib/libmyodbc3.so
SETUP = /usr/lib/libmyodbc3S.so
```

```
UsageCount = 2
```

Netezza

```
      [ODBC Drivers]
Netezza=Installed

[Netezza]
Driver=/usr/local/nz/lib/libnzodbc.so
Setup=/usr/local/nz/lib/libnzodbc.so
```

Teradata

```
      [ODBC Drivers]
Teradata=Installed

[Teradata]
Driver=/usr/odbc/drivers/tdata.so
Setup=/usr/odbc/drivers/tdata.so
```

3 ファイルを保存し、61 ページの「odbc.ini ファイルへの DSN の追加」を続行します。

odbc.ini ファイルへの DSN の追加

データ・ソース・データベースのデータ・ソース名(DSN)は odbc.ini ファイルに構成します。ファイルのあるセクションに、ODBC データ・ソースの名前と説明を追加します。ファイルの新しく作成された別のセクションに、ODBC ドライバのパス、ファイル名およびその他すべての必要なドライバ設定を指定します。

EPM System のインストール・プログラムにより、products/common/ODBC/Merant/6.1 および products/common/ODBC-64/Merant/6.1 ディレクトリにサンプルの odbc.ini ファイルがインストールされます。このファイルには、サポートされている ODBC ドライバの汎用 ODBC 接続と構成情報が含まれます。このファイルを開始ポイントとして使用し、データ・ソース・データベースに対して使用する ODBC ドライバをマップします。

ヒント： この項の手順では、odbc.ini ファイルを手動で編集して DSN を構成する方法を示します。

➤ odbc.ini ファイルに ODBC データ・ソースを追加するには:

1 **EPM System の共通コンポーネントがインストールされているコンピュータで、適切な odbc.ini ファイルを探して開きます。**

vi $ODBCINI コマンドを使用して odbc.ini ファイルとステートメントを編集します:

32 ビットの場合:

```
$EPM_ORACLE_HOME
/common/ODBC/Merant/6.1/odbc.ini
```

64 ビットの場合:

```
$EPM_ORACLE_HOME
/common/ODBC-64/Merant/6.1/odbc.ini
```

2 **[ODBC Data Sources]で始まるセクションを探し、データ・ソース名と説明の新しい行を追加します; 次に例を示します:**

```
mydata=DataDirect 6.1 Oracle Wire Protocol
```

3 **かっこで囲んだ新規 DSN で新しい行を作成し、ファイルに新しいセクションを追加します; 次に例を示します:**

```
[mydata]
```

4 **データ・ソース名に続く行に、このデータ・ソースに必要な ODBC ドライバのフルパスとファイル名、およびその他に必要なデータベース情報を追加します。**

特定の RDBMS のガイドラインとして、次の項に記載されている例を使用します。

注: ODBC ドライバ・ファイルが、"Driver="設定に指定した場所に実際に存在することを確認します。

5 **odbc.ini の編集が終了したら、ファイルを保存してテキスト・エディタを終了します。**

構成の検証のヒント

- ivtestlib (32 ビット)または ddtestlib (64 ビット)を実行して、環境が適切な ODBC ドライバ・ファイルを実行するように設定されていることを検証できます。

  たとえば、ivtestlib を実行し、編集した odbc.ini ファイルの Driver=の後にパスとファイル名を貼り付けます。

- odbc.ini ファイルの[ODBC]ヘッダーの下で、次のパラメータ設定が指定どおりに行われていることを確認します:
  - InstallDir - ドライバのインストール・ディレクトリへのフルパス。この設定に、<installDir>などの変数記述子を使用することはできません。
  - TraceDll - /lib ディレクトリへのフルパス。この設定に、<traceDll>などの変数記述子を使用することはできません。

特定の RDBMS のガイドラインとして、次の項に記載されている例を使用します。

各 RDBMS またはフラット・ファイル・データ・ソースの odbc.ini ファイルおよび ODBC ドライバ設定の詳細は、products/common/ODBC/Merant/6.0/books または products/common/ODBC-64/Merant/6.0 ディレクトリの DataDirect Connect ODBC Reference を参照してください。ベンダーから提供された ODBC ドライバ設定の詳細は、ベンダーから提供された ODBC ドライバのインストール・ガイドを参照してください。

フラット・ファイル・データ・ソース: Essbase Studio では、フラット・ファイル・データ・ソースに ODBC 構成は不要です。

Oracle の ODBC 設定の例

次の例では、Oracle (Solaris)の oradata という名前のデータ・ソース・データベースに接続するために、DataDirect Wire Protocol ドライバを使用して odbc.ini を編集する方法を示します。

```
   [ODBC Data Sources]
oradata=DataDirect 6.1 Oracle Wire Protocol

...

[oradata]
Driver=<$EPM_ORACLE_HOME>/common/ODBC/Merant/6.1/lib/ARora25.so
HostName=oraclehost
SID=ORADB
PortNumber=1521
```

IBM DB2 UDB の ODBC 設定の例

次の例では、IBM DB2 (AIX)の db2data という名前のデータ・ソース・データベースに接続するために、DataDirect Wire Protocol ドライバを使用して odbc.ini を編集する方法を示します。

```
   [ODBC Data Sources]
db2data=DataDirect 6.1 DB2 Wire Protocol

...

[db2data]
Driver=<$EPM_ORACLE_HOME>/common/ODBC/Merant/6.1/lib/ARdb225.so
Database=DB2DB
IpAddress=db2host
TcpPort=50000
```

MySQL の ODBC 設定の例

次の例では、MySQL の ODBC_MySQL という名前のデータ・ソース・データベースに接続するために、ベンダーから提供された ODBC ドライバを使用して odbc.ini を編集する方法を示します。

```
   [ODBC Data Sources]
ODBC_MySQL=MyODBC 3.51 Driver DSN

...

[ODBC_MySQL]
Driver = /usr/local/lib/libmyodbc3.so
Description = Connector/ODBC 3.51 Driver DSN
SERVER = localhost
PORT = 3306
USER = root
Password =
Database = test
OPTION =
SOCKET =
```

Microsoft SQL Server の ODBC 設定の例

次の例では、Microsoft SQL Server 2005 (HP-UX)の sqldata というデータ・ソース・データベースに接続するために、DataDirect Wire Protocol ドライバを使用して odbc.ini を編集する方法を示します。

```
   [ODBC Data Sources]
sqldata=DataDirect 6.1 SQL Server Wire Protocol

...

[sqldata]
Driver=<$EPM_ORACLE_HOME>/common/ODBC/Merant/6.1/lib/ARsqls25.so
Database=SQLDB
Address=mssqlhost,1433
EnableQuotedIdentifiers=1
```

Netezza の ODBC 設定の例

次の例では、Netezza の NZSQL という名前のデータ・ソース・データベースに接続するために、ベンダーから提供された ODBC ドライバを使用して odbc.ini を編集する方法を示します。

```
   [ODBC Data Sources]
NZSQL = NetezzaSQL
```

```
...

[NZSQL]
Driver = /usr/local/nz/lib/libnzodbc.so
Description = NetezzaSQL ODBC
Servername = <123.4.5.6>
Port = 5480
Database = system
Username = admin
Password = password
ReadOnly = false
ShowSystemTables = false
LegacySQLTables = false
LoginTimeout = 0
QueryTimeout = 0
DateFormat = 1
NumericAsChar = false
SQLBitOneZero = false
StripCRLF = false
```

Teradata の ODBC 設定の例

次の例では、Teradata の terasource という名前のデータ・ソース・データベースに接続するために、ベンダーから提供された ODBC ドライバを使用して odbc.ini を編集する方法を示します。

Solaris および AIX

```
   [ODBC Data Sources]
terasource=Teradata data source

...

[terasource]
Driver=/usr/odbc/drivers/tdata.so #teradata installation path of .so file
DBCName=abcd0072.us.xyzco.com
PortNumber=1025
```

HP-UX

```
   [ODBC Data Sources]
terasource=Teradata data source

...

[terasource]
Driver=/usr/odbc/drivers/tdata.sl #teradata installation path of .sl file
DBCName=abcd0072.us.xyzco.com
PortNumber=1025
```

JDBC ドライバの構成

キューブの配置中に、Essbase Studio をストリーミング・モードで実行すると、Essbase Studio サーバーは JDBC ドライバを使用して外部データ・ソースにクエリーを直接実行します。

Oracle、Oracle BI EE、IBM DB2 および Microsoft SQL Server 用の JDBC ドライバは、Essbase Studio のインストール時にインストールされます。

MySQL、Netezza および Teradata の場合は、次の項の説明に従って、製造元から JDBC ドライバを入手し、インストールする必要があります。

注： Essbase Studio をインストールして構成したら、適用可能な JDBC ドライバに対して、この項のタスクを実行します。

MySQL

➤ MySQL JDBC ドライバを設定するには:

1 **MySQL の Web サイトから、適切なバージョンのファイルを次のディレクトリにダウンロードします:**

```
EPM_ORACLE_HOME
/products/Essbase/EssbaseStudio/Server
```

ダウンロードするドライバ・ファイルの名前は、ダウンロードしたバージョンにより異なります。

2 **ファイル `mysql-connector-java.jar` の名前を変更します。**

名前を変更するまで、Essbase Studio はドライバ・ファイルを認識しません。

Netezza

➤ Netezza JDBC ドライバを設定するには:

1 **Netezza の Web サイトから、適切なバージョンのファイルを次のディレクトリにダウンロードします:**

```
EPM_ORACLE_HOME
/products/Essbase/EssbaseStudio/Server
```

ダウンロードするドライバ・ファイルの名前は、ダウンロードしたバージョンにより異なります。

2 **ファイル `nzjdbc3.jar` の名前を変更します。**

名前を変更するまで、Essbase Studio はドライバ・ファイルを認識しません。

Teradata

➤ Teradata ドライバを設定するには:

1 Teradata の Web サイトから、Teradata JDBC ドライバのライブラリ・ファイル `tdgssconfig.jar` および `terajdbc4.jar` をダウンロードします。

2 次のディレクトリにファイルを抽出します:

```
EPM_ORACLE_HOME
/products/Essbase/EssbaseStudio/Server
```

注: 互換性の問題や不明な点は、直接 Teradata にお問い合せください。

注: キューブを非ストリーミング・モードで配置する場合は、Essbase サーバーを実行しているマシンにデータ・ソース ODBC ドライバをインストールする必要があります。

Essbase Studio サーバーおよび Essbase Studio コンソールの開始と停止

EPM System コンフィグレータを使用した構成が完了し、`server.properties` ファイルの手動構成を実行したら、Essbase Studio サーバーおよびコンソールを開始できます。

スクリプトを使用して Essbase Studio サーバーを起動または停止するときは、`EPM_ORACLE_INSTANCE/bin` にある `.bat` または `.sh` スクリプトのみを使用してください。

`EPM_ORACLE_HOME/products/Essbase/EssbaseStudio/Server` にある `.bat` または `.sh` ファイルを使用して Essbase Studio サーバーを起動または停止しないでください。

Essbase Studio サーバーおよびコンソールを含むすべての EPM System 製品の開始と停止に関する情報は、Oracle Enterprise Performance Management System インストールおよび構成ガイドに記載されており、Oracle Technology Network (OTN)にある EPM System のドキュメント・ライブラリで入手できます:

http://www.oracle.com/technology/documentation/epm.html

UNIX 上で Essbase Studio サーバーをフォアグラウンドで実行する方法は、68 ページの「UNIX 上におけるフォアグラウンドでの Essbase Studio サーバーの開始」を参照してください。

UNIX 上におけるフォアグラウンドでの Essbase Studio サーバーの開始

UNIX 上では、デフォルトで Essbase Studio サーバーはバックグラウンドで実行されます。フォアグラウンドで Essbase Studio サーバーを実行するには、次の項目を編集します:

- サーバー・プロパティ `server.runInBackground`。
- 次の環境変数:
  - EPM_ORACLE_HOME=<value>
  - EPM_ORACLE_INSTANCE=<value>
  - JAVA_HOME=$EPM_ORACLE_HOME/../jdk160_11/jre
  - JAVA_OPTIONS=-DESSBASE_STUDIO_INSTANCE\= $EPM_ORACLE_INSTANCE/BPMS/bpms1 -DsuppressAPSProductInfo=<T|F>
- Essbase Studio の起動シェル・スクリプト `startServer.sh`。

➤ UNIX 上において、Essbase Studio サーバーをフォアグラウンドで開始するには:

1 **Essbase Studio の `server.properties` ファイルで、`server.runInBackground` プロパティを false に設定するか、コメントアウトします。**

このプロパティはデフォルトで true に設定されます。

`server.properties` ファイルは、`Oracle/Middleware/user_projects/epmsystem1/BPMS/bpms1/bin/server.properties` にあります。

注: このプロパティの詳細は、49 ページの「server.runInBackground」を参照してください。

2 **`startServer.sh` を実行する環境で、次の変数を表示されているとおりに設定します:**

```
EPM_ORACLE_INSTANCE=/<install path>/Oracle/Middleware/user_projects/epmsystem1

EPM_ORACLE_HOME=/<install path>/Oracle/Middleware/EPMSystem11R1

JAVA_HOME="${EPM_ORACLE_HOME}/../jdk160_11/jre"

JAVA_OPTIONS="-DESSBASE_STUDIO_INSTANCE=${EPM_ORACLE_INSTANCE}/BPMS/bpms1
```

```
-DsuppressAPSProductInfo=true"
```

3 **Essbase Studio の** `startServer.sh` **シェルを次のように編集します。**

`startServer.sh` ファイルは`$EPM_ORACLE_HOME/products/Essbase/EssbaseStudio/Server/startServer.sh` にあります。

1. ファイルの最後の行を見つけます:

```
nohup "${JAVA_HOME}/bin/java" -Xms128m -Xmx768m $JAVA_OPTIONS
-jar "${EPM_ORACLE_HOME}/products/Essbase/EssbaseStudio/
Server/server.jar" >/dev/null &
```

2. 行の先頭にある nohup、行の最後にある NULL への出力である STDOUT 命令(>/dev/null)、およびバックグラウンド処理コマンド(&)を削除します。次はその例です:

```
"${JAVA_HOME}/bin/java" -Xms128m -Xmx768m $JAVA_OPTIONS -jar "$
{EPM_ORACLE_HOME}
/products/Essbase/EssbaseStudio/Server/server.jar"
```

4 **次のステートメントを実行して Essbase Studio サーバーを開始します:**

```
./startServer.sh
```

Essbase Studio サーバーのコマンド

Essbase Studio サーバー・コマンドは、要求マネージャの観点から、サーバーが実行中のタスクに関する情報およびサーバーのステータスを通知します。要求マネージャは、Essbase Studio サーバーへのタスク・フローとタスクの実行を制御します。また、タスク・フローを変更するコマンドもあります。

Essbase Studio サーバー・コマンドのリストは、起動時にサーバー・ウィンドウに表示されます。これらのコマンドを直接サーバー・ウィンドウに入力すると、必要な応答を得られます。

一部のコマンドは、`server.properties` でデフォルトのサーバー設定を変更したり、指定した設定値を上書きする場合に使用します。サーバー・コマンドを使用して設定を変更しても、次の Essbase Studio サーバー・セッションで無効になります。サーバーの再起動後、変更したコマンド設定はデフォルトの値または `server.properties` で指定した設定値に戻ります。

サーバー・ウィンドウで[Enter]キーを押すと、使用可能なコマンドのリストが表示されます。

Essbase Studio サーバー・コマンドについて、次に説明します。

サーバー・コマンド

- **`version`**-画面上の Essbase Studio サーバーのバージョン情報を印刷します。

コマンド

version

例

```
version
```

戻り値

```
Oracle Essbase Studio Server version 11.1.2.0.00 Build Number 1206
```

- **dumps**-サーバー・スレッドのスタック・トレースを表示します。このコマンドは、主に開発環境で使用されます。

  コマンド

  dumps

  例

  ```
  dumps
  ```

  戻り値

  dumps コマンドを実行すると戻される情報部分の例を次に示します:

  ```
  thread: Finalizer    at java.lang.Object.wait(Native Method)    at
  java.lang.ref.ReferenceQueue.remove(Unknown Source)    at
  java.lang.ref.ReferenceQueue.remove(Unknown Source)    at java.lang.ref.Finalizer
  $FinalizerThread.run(Unknown Source) thread: Reference Handler    at
  java.lang.Object.wait(Native Method)    at java.lang.Object.wait(Unknown Source)
    at java.lang.ref.Reference$ReferenceHandler.run(Unknown Source) thread: Signal
  Dispatcher thread: Thread-1    at java.lang.Object.wait(Native Method)    at
  com.sun.jndi.ldap.pool.PoolCleaner.run(Unknown Source)
  ```

- **pconf**-指定したワーカー・スレッドの数、割り当てられたリソースの数、使用可能なリソースの数、キューのサイズ、使用中のキューの数など、Essbase Studio サーバーの要求マネージャ・パラメータの状態を表示します。

  ワーカー・スレッドについての説明は、43 ページの「server.threadCount」を参照してください。

  コマンド

  pconf

  例

  ```
  pconf
  ```

  戻り値

  pconf コマンドを実行すると戻される情報の例を次に示します。

  ```
  Request manager configuration:  Threads: 10, Resources: 5 (4 avail), Queue
  Size: 100 (10 in use)
  ```

- **squeue**-タスクの数の観点からキューのサイズを設定します。Essbase Studio サーバーによる実行待ちのタスクがキューに入れられます。

  キューのデフォルトのタスクの数は 200 です。

  コマンド

  squeue

  このコマンドを実行すると、新しいキュー・サイズの入力を要求されます。

  例

  ```
  squeue

  new queue size:
  250
  ```

  注： この設定を変更しても、次の Essbase Studio サーバー・セッションで無効になります。サーバーの再起動後、33 ページの「サーバー・プロパティ・ファイルの構成」で説明するように、squeue パラメータはデフォルトまたは server.queueSize で指定した値に戻ります。

- **sthd**-Essbase Studio サーバーに割り当てるワーカー・スレッドの数を設定します。

  ワーカー・スレッドについての説明は、43 ページの「server.threadCount」を参照してください。

  注： リスニングするワーカー・スレッドを考慮して Essbase Studio クライアントからタスクを取得するには、ワーカー・スレッドの数をリソースの数よりも大きく設定する必要があります。

  デフォルト設定は 30 スレッドです。

  コマンド

  sthd

  このコマンドを実行すると、新しいスレッド数の入力を要求されます。

  例

  ```
  sthd

  new thread count:
  45
  ```

  注： この設定を変更しても、次の Essbase Studio サーバー・セッションで無効になります。サーバーの再起動後、33 ページの「サーバー・プロパティ・ファイルの構成」で説明するように、sthd パラメータはデフォルトまたは server.threadCount で指定した値に戻ります。

- **sres**-リソースの数を設定します。これにより、同時に実行できるタスクの数が定義されます。

  デフォルト設定は 20 リソースです。

  コマンド

  sres

  このコマンドを実行すると、新しいリソース数の入力を要求されます。

  例

  ```
  sres

  new resource count:
   30
  ```

  注： この設定を変更しても、次の Essbase Studio サーバー・セッションで無効になります。サーバーの再起動後、33 ページの「サーバー・プロパティ・ファイルの構成」で説明するように、sres パラメータはデフォルトまたは server.resourceCount で指定した値に戻ります。

- **exit**-Essbase Studio サーバーを停止してサーバー・ウィンドウを閉じます。

  コマンド

  exit

  例

  ```
  exit
  ```

Essbase Studio カタログ・データベースのエクスポートとインポート

Essbase Studio は、カタログ・データベースまたは選択したカタログ・データベース・アーティファクトを XML ファイルにエクスポートするメカニズムを提供します。カタログを再作成するために、その XML ファイルを別のコンピュータにインポートできます。詳細は、次の各トピックを参照してください:

- 72 ページの「カタログのエクスポートおよびインポートについて」
- 73 ページの「XML ファイルへのカタログ・データベースのエクスポート」
- 76 ページの「カタログ XML ファイルのインポート」

カタログのエクスポートおよびインポートについて

Essbase Studio カタログ・データベースをあるコンピュータから別のコンピュータに移動する必要がある場合があります。Essbase Studio では、カタログ・データベー

スをファイルにエクスポートし、そのファイルを別のマシンにインポートしてカタログを再作成することが可能です。

プロセスは、エクスポート操作から開始します。「ファイル」メニューの「エクスポート」オプションを使用して、次の操作を実行できます:

- カタログ全体のエクスポート
- データ・ソース接続、次元要素、階層、別名セット、キューブ・スキーマ、Essbase モデル、ドリルスルー・レポートなど、個々のカタログ・アーティファクトをエクスポートします。

  次のように分類した個々のカタログ・アーティファクトをエクスポートします。

  - 共通メタデータ–次元要素、階層、別名セット、キューブ・スキーマ、ドリルスルー・レポートなどのアーティファクト。
  - 配置ベースのメタデータ–Essbase モデルや配置履歴などのアーティファクト。

「データ・ソース・ナビゲータ」または「メタデータ・ナビゲータ」で個々または複数のカタログ・アーティファクトを右クリックし、ショートカット・メニューから「エクスポート」オプションを選択することもできます。

選択するエクスポート方法にかかわらず、カタログ・データベースまたは選択したアーティファクトは XML ファイルにエクスポートされます。

次に、新しい場所またはマシンで XML ファイルをインポートします。ファイルのコンテンツ全体をインポートする必要があります。ファイルからアーティファクトを選択してインポートすることはできません。

注意

- Essbase Studio カタログ・データベースのエクスポート/インポートを実行するには、ユーザーが Shared Services に管理者としてプロビジョニングされている必要があります。
- 特定のカタログ・アーティファクトがエクスポートされた場合、作成される XML ファイルには、選択したカタログ・アーティファクトの子孫が含まれます。たとえば、階層が選択された場合は、その階層の系列にある次元要素とデータ・ソース接続もエクスポートされます。
- データ・ソース接続をエクスポートする際、ソース・データベースに対するパスワードは生成される XML ファイルに含まれません。XML ファイルをインポートした後で、118 ページの「データ・ソース接続プロパティの編集」の説明に従って、接続を編集してパスワードを追加し直します。

XML ファイルへのカタログ・データベースのエクスポート

カタログ・データベース全体をエクスポートすることも、選択したカタログ要素のみをエクスポートすることも可能です。手順は、次のトピックを参照してください:

- 74 ページの「カタログ・データベース全体のエクスポート」

- 74 ページの「選択したカタログ要素のエクスポート」

カタログ・データベース全体のエクスポート

共通メタデータと配置固有メタデータの両方が含まれる、カタログ・データベース全体のエクスポートを実行するには、次の手順を使用します。

➤ Essbase Studio カタログ・データベースを XML ファイルにエクスポートするには:

1. **Essbase Studio コンソールで、「ファイル」>「エクスポート」を選択します。**
2. **「エクスポート」ダイアログ・ボックスで、「参照」をクリックし、エクスポートしたカタログ XML ファイルを置く場所に移動します。**
3. **「ファイルの場所」に、XML ファイルの名前を入力します。**
4. **オプション:「ファイルの説明」にわかりやすいテキストを入力します。**
5. **「ファイル・コンテンツ」で、オプションを選択します:**
   - 「カタログ全体のエクスポート」

     カタログ全体には、共通メタデータと配置固有メタデータの両方が含まれます。

     カタログ全体をエクスポートするには、手順 6 に進みます。
   - 「一部のカタログのエクスポート」

     一部のカタログには、選択した共通メタデータに加えオプションの配置固有メタデータが含まれます。

     選択したカタログ要素をエクスポートするには、74 ページの「選択したカタログ要素のエクスポート」を参照してください。
6. **「OK」をクリックしてエクスポートを完了します。**

   カタログ・データベース全体が、手順 3 で指定した名前の XML ファイルにエクスポートされます。

   注 : エクスポート中にエラーが発生した場合は、メッセージ・ボックスにエラーの詳細が表示されます。
7. **カタログ・データベース XML ファイルが、手順 2 で指定した場所に存在することを確認します。**

これで、XML カタログ・データベースを新しい場所にインポートできます。76 ページの「カタログ XML ファイルのインポート」を参照してください。

選択したカタログ要素のエクスポート

選択した共通メタデータに加えオプションの配置固有メタデータが含まれる、一部のカタログのエクスポートを実行するには、次の手順を使用します。

➤ 選択したカタログ要素(一部のカタログ)を XML ファイルにエクスポートするには:

1 **次のいずれかのアクションを行います:**

- 73 ページの「XML ファイルへのカタログ・データベースのエクスポート」の手順 1 から手順 5 までを実行し、手順 5 で「一部のカタログのエクスポート」を選択します。
- 「メタデータ・ナビゲータ」で要素を右クリックするか、「データ・ソース・ナビゲータ」でデータ・ソースを右クリックし、ショートカット・メニューから「エクスポート」を選択します。

「一部のカタログのエクスポート」・オプションが選択され、「追加」ボタンが有効になっている「エクスポート」ダイアログ・ボックスが表示されます。

一部のカタログのエクスポートには、選択した共通メタデータに加え関連付けられているオプションの配置固有メタデータが含まれます。

カタログ全体をエクスポートするには、74 ページの「カタログ・データベース全体のエクスポート」の手順 6 に進みます。

2 **「追加」をクリックして、オプションを選択します:**

- 「メタデータの追加」

  「エクスポートする要素の選択」ダイアログ・ボックスに、「メタデータ・ナビゲータ」の内容が表示されます。次の手順を実行します:

  1. エクスポートするアイテムに移動します。
  2. 選択が終了したら、「OK」をクリックします。

- 「データ・ソースの追加」

  「エクスポートするデータ・ソースの選択」ダイアログ・ボックスに、「データ・ソース・ナビゲータ」の内容が表示されます。次の手順を実行します:

  1. エクスポートするデータ・ソースを選択します。
  2. 選択が終了したら、「OK」をクリックします。

エクスポートに対して選択したメタデータ要素とデータ・ソースは、「エクスポート」ダイアログ・ボックスの「ファイル・コンテンツ」部分の表にリストされます。

各要素のフルパスは、「名前」の下に表示されます。接続や階層などの要素のタイプは、「タイプ」の下に表示されます。

3 **オプション:「キューブ・スキーマに関連付けられた Essbase モデルを含める」チェック・ボックスを選択します。**

注: このオプションは、「一部のカタログのエクスポート」のオプションが選択されている場合のみ使用可能です。

キューブ・スキーマを持つ Essbase モデルをエクスポートすると、配置固有オブジェクトや配置履歴など、エクスポート・ファイルに大量の情報が追加さ

れます。配置情報は含めずに論理メタデータのみをエクスポートするには、「キューブ・スキーマに関連付けられた Essbase モデルを含める」オプションを選択解除します。

4 **オプション: エクスポートを実行する前に、「ファイル・コンテンツ」リストからアイテムを削除するには、「削除」 をクリックします。**

5 **エクスポートするすべてのメタデータおよびデータ・ソース要素について、この手順を繰り返します。**

6 **「OK」をクリックします。**

カタログから選択したアイテムが、手順 2 から手順 5 で指定した XML ファイルにエクスポートされます。

注： エクスポート中にメモリー不足エラーが発生した場合は、78 ページの「仮想メモリーの構成」の説明に従って、Essbase Studio コンソールの仮想メモリー設定を増やします。

これで、XML カタログ・データベース・アーティファクトを新しい場所にインポートできます。76 ページの「カタログ XML ファイルのインポート」を参照してください。

カタログ XML ファイルのインポート

インポートの準備

➤ カタログ XML ファイルのインポート手順を開始する前に、次のタスクを実行します:

1 **選択した方法を使用して、73 ページの「XML ファイルへのカタログ・データベースのエクスポート」または 74 ページの「選択したカタログ要素のエクスポート」で作成したカタログ XML ファイルをインポート先の場所またはコンピュータに移動します。**

たとえば、FTP プログラムを使用して、ファイルを別のコンピュータのターゲットの場所に移動できます。

2 **`server.properties` の `catalog.db` プロパティで、ターゲット・カタログ・データベースの場所が指定されていることを確認します。**

39 ページの「catalog.db」を参照してください。

新規のカタログの場合は、Essbase Studio のインストールに含まれている `catalog_schema.sql` スクリプトを使用して、ターゲット・カタログ・データベースとして機能する空のデータベースを作成します。スクリプトは次の場所にあります:

```
EPM_ORACLE_HOME
/products/Essbase/EssbaseStudio/server/database/common/
your_RDBMS
```

注： このタスクでは、カタログ RDBMS プログラムがターゲット・マシンにインストールされていることを前提とします。

インポート

この手順を開始する前に、インポートの準備のタスクを完了してください。

➤ Essbase Studio カタログ・データベース XML ファイルをインポートするには:

1 **Essbase Studio コンソールで、「ファイル」>「インポート」を選択します。**

2 **「インポート」ダイアログ・ボックスで、「参照」をクリックし、「ファイルのインポート」で、エクスポートしたカタログ XML ファイルの場所に移動します。**

3 **インポートする XML ファイルを選択し、「開く」をクリックして「インポート」ダイアログ・ボックスに戻ります。**

「ファイル名」、「ファイルの説明」および「ファイル・コンテンツ・タイプ」フィールドには、エクスポート・プロセスで入力した情報が移入されます。

4 **「カタログ・グループへのインポート用オプション」で、オプションを選択します:**

- 「要素を確認して上書きする」-要素がカタログ・データベースにすでに存在する場合は、XML ファイルの新しい要素で上書きされます。

  注： 「要素を確認して上書きする」オプションを選択して Essbase Studio カタログ XML ファイルをインポートする場合に、インポートする XML ファイルに含まれるデータ・ソース接続がターゲット・カタログにも存在する場合、ミニスキーマおよび他のオブジェクト(表やビュー、シノニムなど)はインポートされません。

  回避策: エクスポートされたデータ・ソースと既存のデータ・ソースに相違がある場合は、次のタスクを完了します: データ・ソース接続を右クリックして、「増分更新」を選択します。「増分更新」ダイアログ・ボックスで、データ・ソース接続に追加する表、ビューおよびシノニムを選択します。次に、データ・ソース接続内の個別の表、ビューおよびシノニムを右クリックして「リフレッシュ」を選択し、列を追加または削除してプロパティを更新します。

- 「要素を確認するが上書きしない」-要素がカタログ・データベースにすでに存在する場合は、その要素が保持され、XML ファイルの重複する要素は使用されません。
- 「要素を確認しないで、すべて新しい要素にする」-XML ファイルを使用してカタログ・データベースが作成されます。

5 **「OK」をクリックして、XML ファイル内の要素をカタログ・データベースにインポートします。**

XML ファイルは、`server.properties` ファイルの `catalog.db` で指定されたカタログ・データベースへの移入を行います(39 ページの「catalog.db」を参照)。

インポート中にエラーが発生した場合は、詳細が記載されたエラー・メッセージ・ボックスが表示されます。

注： インポート中にメモリー不足エラーが発生した場合は、78 ページの「仮想メモリーの構成」の説明に従って、Essbase Studio コンソールの仮想メモリー設定を増やします。

6 **新しくインポートしたデータ・ソース接続に、パスワードを追加し直します。**

データ・ソース接続をエクスポートする際、ソース・データベースに対するパスワードは生成される XML ファイルに含まれません。XML ファイルのインポート後、Essbase Studio コンソールにログインし、インポートした要素がホストされている Essbase Studio サーバーに接続します。次に、118 ページの「データ・ソース接続プロパティの編集」の説明のとおり、接続を編集してパスワードを追加し直します。

仮想メモリーの構成

大規模なカタログ・データベースのエクスポートや非常に複雑なキューブ配置などの操作では、使用される仮想メモリーの量が、Essbase Studio コンソールを実行しているマシンの仮想メモリー設定に限りなく近くなる可能性があります。

仮想メモリー設定は、オペレーティング・システムで指定されている範囲で増加させることが可能です。たとえば、Windows 32 ビット・プラットフォームでは、最大仮想メモリー設定は 2048MB です。

仮想メモリー設定を変更するには、次の手順を実行します。

➤ Essbase Studio コンソールの仮想メモリーを構成するには:

1 **`EPM_ORACLE_HOME/products/Essbase/EssbaseStudio/Console` で `EssbaseStudio.ini` を開きます。**

`EssbaseStudio.ini` ファイルには、デフォルトで次の 2 行が含まれています:

```
     -vmargs
-Xmx1024m
```

2 行目には仮想メモリー設定が指定されています。デフォルトで、仮想メモリーは 1024MB に設定されています。

2 **仮想メモリー設定を増やしてメモリーを増加します; 次に例を示します:**

```
     -Xmx1536m
```

この例では 1536MB の仮想メモリーを指定しています。

3 **変更した `EssbaseStudio.ini` ファイルを保存します。**

4 **Essbase Studio コンソールを再起動します。**

配置履歴の消去

Essbase モデルを配布するたびに、配置履歴データが収集され、Essbase Studio カタログに格納されます。カタログを別のリリースに移行する場合、Essbase Studio サーバーはカタログに含まれている配置履歴データを使用します。カタログ内の履歴が大きすぎる場合、移行を完了できません。

Essbase Studio では、モデルの配置履歴ファイル `cleanModelHistory` を消去するユーティリティが提供されています。

このコマンドを使用する前に、ファイルが何度も再配置されたモデルなどの大量の配置履歴ファイルがあるモデルを識別する必要があります。操作するモデルを特定したら、Essbase Studio のコマンド・ライン・クライアントから `cleanModelHistory` ユーティリティを実行します。

➤ `cleanModelHistory` ユーティリティを実行するには:

1 `MIDDLEWARE_HOME/user_projects/epmsystem1/bin` に移動します。

2 `start_BPMS_bpms1_CommandLineClient.bat|sh` を呼び出して、Essbase Studio コマンド・ライン・クライアントを開始します。

CPL Shell というコマンド・ウィンドウが表示されます。

3 プロンプトで、有効な Essbase Studio 管理者ユーザー名およびパスワードを入力します。

注： Essbase Studio の `cleanModelHistory` ユーティリティを使用するには、Essbase Studio の管理者権限を持っている必要があります。

4 次の構文を使用して、`cleanModelHistory` コマンドを実行します。

```
call 'util'::'cleanModelHistory'("MODEL_NAME", dataset:\'DATASET_NAME');
```

`"MODEL_NAME"`は、消去する履歴のモデル名です。

`'DATASET_NAME'`は、このモデルの構築に使用したキューブ・スキーマ名です。

例:

```
call 'util'::'cleanModelHistory'("ProdMktModel1", dataset:\'ProdMktCS');
```

指定したキューブの現在の@ "in-sync"エントリ以外の、モデルの配置履歴にあるすべてのエントリが消去されます。すべての失敗と現在以外の@成功が削除されます。

再ホストされた EPM System 製品への参照の更新

EPM System 製品を(アップグレード後などに)新しいホスト・マシンに移動する際は必ず、新しいホスト名とポート番号を反映させるために一部の製品に対する参照を更新する必要があります。

注： Essbase Studio を含むその他の EPM System 製品の再ホスト手順を実行する前に、Essbase の再ホスト手順を実行する必要があります。

- 再ホストされた Essbase サーバー・インスタンスまたはクラスタには、81 ページの「再ホストされた Essbase サーバーへの参照の更新」の手順を使用します。
- 再ホストされた Essbase Studio サーバー・インスタンスには、308 ページの「キューブ・リンケージの更新」の手順を使用します。
- 再ホストされた Performance Management Architect データ・ソースには、118 ページの「データ・ソース接続プロパティの編集」の手順を使用します

EPM System 製品の再ホストに関する完全な情報については、Oracle Enterprise Performance Management System インストールおよび構成ガイドを参照してください。

再ホストされた Essbase サーバーへの参照の更新について

Essbase サーバー接続および配置されたキューブは、基礎となる Essbase サーバー・インスタンスまたはクラスタに対するホスト、ポートまたはデータ暗号化情報の変更に影響されます。

再ホストされた Essbase サーバー・インスタンスおよびクラスタへの参照は、「Essbase 接続の再ホスト」・ダイアログ・ボックスで更新します。81 ページの「再ホストされた Essbase サーバーへの参照の更新」の説明に従って、新しいホストまたはクラスタ、ポートおよびデータ暗号化方法(必要な場合)を指定します。

注： Essbase Studio を含むその他の EPM System 製品の再ホスト手順を実行する前に、Essbase の再ホスト手順を実行する必要があります。

再ホストに関するその他のトピック

- 再ホストされた Essbase Studio サーバー・インスタンスには、308 ページの「キューブ・リンケージの更新」の手順を使用します。
- 再ホストされた Performance Management Architect データ・ソースには、118 ページの「データ・ソース接続プロパティの編集」の手順を使用します

EPM System 製品の再ホストに関する完全な情報については、Oracle Enterprise Performance Management System インストールおよび構成ガイドを参照してください。

再ホストされた Essbase サーバーへの参照の更新

注： Essbase Studio を含むその他の EPM System 製品の再ホスト手順を実行する前に、Essbase の再ホスト手順を実行する必要があります。

➤ 再ホストされた Essbase サーバー・インスタンスまたはクラスタへの参照を更新するには:

1 **Essbase Studio コンソールで、「ツール」、「Essbase 接続の再ホスト」の順に選択します。**

2 **「Essbase 接続の再ホスト」において、「Essbase 接続」で再ホストする接続を選択します。**

「カタログのホスト/ポート/暗号化」に、古いクラスタ名または Essbase ホストとポート番号、該当する場合はデータ暗号化方法が表示されます。

3 **「新しいホスト/ポート/暗号化」に、新しいクラスタ名、または新しいホスト名、ポート番号およびデータ暗号化方法(該当する場合)を入力します。**

注： Essbase サーバー・クラスタの場合、必要なものはクラスタ名のみです。ポート番号は必要ありません。デフォルトで、「新しいホスト/ポート/暗号化」列に新しいクラスタ名が表示されます。

次の構文を使用します:

- Essbase サーバー・インスタンスで暗号化なしの場合は、次のように入力します:

```
host
:
port
```

- Essbase サーバー・インスタンスで暗号化を使用する場合は、次のように入力します:

```
host
:
port
:ssl
```

- Essbase サーバー・クラスタで暗号化なしの場合は、次のように入力します:

```
cluster_name
```

4 **オプション:「新しいホスト/ポート/暗号化」に入力したエントリを検証するには、「接続のテスト」をクリックします。**

注： Essbase Studio では、エントリを入力しても検証されません。手順 3 で入力した情報を検証するには、「接続のテスト」をクリックする必要があります。

5 **現在選択されている Essbase サーバー接続に対して、「配置履歴の更新」オプションを選択します:**

- 「すべての配置履歴のホスト名およびポート番号を更新」 - 古い Essbase サーバー・ホストおよびポートのすべての参照を、手順 3 で指定した新しいホストとポートに置き換えます。

  これがデフォルトのオプションです。

- 「最後に成功した配置履歴を複製してコピーのみを更新」 - 最後に成功した配置履歴のリストのコピーを作成し、そのコピーを新しいホストとポートの情報および再ホストの日時で更新します。

ヒント： 複製のオプションを選択した場合、更新が完了すると、「メタデータ・ナビゲータ」に再ホストされた各 Essbase モデルが 2 つずつ表示されます: 1 つは古い host:port 情報とともに、もう 1 つは新しい host:port 情報とともに表示されます。

注： 配置履歴は、正常に再ホストされた Essbase サーバー接続に対してのみ更新されます。

6 **「更新」をクリックします。**

7 **再ホストされた各 Essbase サーバー接続の再ホスト・ステータスは、「Essbase 接続の再ホスト」・ダイアログ・ボックスの「ステータスの更新」列で確認します。**

再ホストに成功すると、「カタログのホスト/ポート/暗号化」列が新しいクラスタまたはホストとポート情報、および暗号化情報(該当する場合)で更新されます。再ホストに失敗すると、エラー・メッセージが表示されます。

注： その他の Essbase サーバー接続に同一の古い host:port が設定されている場合、1 つのインスタンスを再ホストするとその他すべてのインスタンスも再ホストされます。

3 ユーザー管理およびセキュリティ

この章の内容

Shared Services と Essbase Studio の併用について 83
Shared Services の Essbase Studio の役割 83
Shared Services Console の起動およびログイン 84
Shared Services ユーザーへのアクセス権の割当て 85

Shared Services と Essbase Studio の併用について

Essbase Studio のユーザー管理とセキュリティは、Shared Services で実現されます。Shared Services によって、ユーザー管理、ユーザー・プロビジョニングおよび外部認証が提供されます。プロビジョニングとは、役割およびアクセス権を Essbase Studio ユーザーに割り当てるプロセスのことです。

Shared Services 機能を実装している製品には、Shared Services クライアントとサーバー・ソフトウェアを実行している Shared Services サーバーへのアクセス、および Shared Services 専用のデータベースへのアクセスが必要です。

Shared Services の Essbase Studio の役割

役割により、ユーザーが実行できるタスクが決まります。役割は次の方法でグループ化できます:

- 製品固有の役割

  Essbase の役割の例は、データ・ソース管理者とメタデータ管理者です。これらの役割により、ユーザーが Essbase Studio アーティファクトと行う対話の種類を決定できます。特定のアーティファクトと行う各役割の対話は、Oracle Enterprise Performance Management System User Security Administration Guide の付録 Essbase Studio の役割を参照してください。

- Shared Services の役割

  Shared Services の役割の例は、プロジェクト・マネージャまたはプロビジョニング・マネージャです。Shared Services の役割の多くはグローバルです(役割がすべての Shared Services アプリケーションに適用されます)。Shared Services 役割の詳細は、Oracle Enterprise Performance Management System User Security Administration Guide を参照してください。

次の Essbase Studio の役割により、Essbase Studio でタスクを実行する権限の様々なレベルを提供します。

- 管理者 - キューブの配置とドリルスルー・レポートの実行を含む、すべての Essbase Studio タスクを実行します
- メタデータ管理者 - メタデータ要素の作成および維持に関連するすべてのタスクを実行します; キューブを配置します; ドリルスルー・レポートを実行します
- データ・ソース管理者 - データ・ソース接続の作成および維持に関連するすべてのタスクを実行します; ドリルスルー・レポートを実行します
- ビューア - すべての Essbase Studio データ・ソースおよびメタデータ要素を表示します; ドリルスルー・レポートを実行します

管理者およびメタデータ管理者は、Essbase へのキューブの配置を行う責任があるため、さらに次の役割のプロビジョニングが必要です:

- Shared Services 管理者(オプション)
- Shared Services プロジェクト・マネージャ(必須)

  キューブを配置するには、少なくともプロジェクト・マネージャの役割が必要です。
- Essbase 管理者(オプション)
- Essbase アプリケーションの作成/削除(必須)

  キューブを配置すると、情報が Essbase に書き込まれるため、Essbase への書込みを行うには、少なくとも「アプリケーションの作成/削除」の役割が必要です。

  キューブの配置後、新規の Essbase アプリケーションおよびデータベースにアクセスするために、Essbase Studio ユーザーのプロビジョニングを行う必要はありません。Essbase Studio から Essbase に配置するキューブ用に権限が継承されます。ただし、Essbase Studio で作成されるアプリケーションおよびデータベースにアクセスするには、Essbase ユーザーのプロビジョニングが必要です。

Shared Services Console の起動およびログイン

Oracle Hyperion Shared Services Console を起動する手順は、Oracle Enterprise Performance Management System User Security Administration Guide の Shared Services Console の起動のトピックを参照してください。

Shared Services Console を起動する場合、適切なユーザー権限でログインする必要があります。たとえば、Essbase Studio ユーザーのプロビジョニングを行うには、Shared Services 管理者としてログインする必要があります。

Shared Services ユーザーへのアクセス権の割当て

インストールおよび構成の後で、Essbase Studio の役割を Shared Services Console のユーザーおよびグループに割り当てます。

Oracle Hyperion Shared Services Console の Essbase Studio ユーザーを管理するには、Shared Services プロビジョニング・マネージャの役割がプロビジョニングされているユーザーとしてコンソールにログインする必要があります。

ユーザーをプロビジョニングするときに、Essbase Studio サーバーのプロジェクトの下に Essbase Studio の役割がリストされます。

注： キューブ配置を実行する Essbase Studio ユーザーまたはグループには、少なくとも Shared Services プロジェクト・マネージャ役割も割り当てられている必要があります。ユーザーにはアプリケーションまたは管理者の作成/削除などの Essbase サーバー役割も必要です。

Oracle Hyperion Shared Services では、親グループに 1 つ以上のサブグループが含まれる集約グループがサポートされています。サブグループでは、親グループの役割が継承されます。たとえば、親グループに Essbase Studio メタデータ管理者の役割をプロビジョニングした場合、サブグループ(およびそのグループのユーザー)は、メタデータ管理者の役割を継承します。

ユーザーとグループへのアクセス権の割当ての詳細は、Oracle Enterprise Performance Management System User Security Administration Guide のユーザーおよびグループのプロビジョニングのトピックと Essbase のプロビジョニングの章を参照してください。

4 データ・ソース接続

この章の内容

概要 87
接続ウィザードを使用した接続の作成 88
接続パラメータの定義 88
接続に含めるテーブルの選択 89
ミニスキーマ・オプションの選択 89
ミニスキーマの移入 90
メタデータ要素の作成 90
リレーショナル・ソースへの接続の作成 90
Essbase への接続の作成 101
Performance Management Architect データ・ソースへの接続の作成 102
テキスト・ファイルのデータ・ソースへの接続の作成 104
データ・ソース接続の同期 111
データ・ソース接続の操作 118
ユーザー定義テーブルの作成 126

概要

Essbase Studio では、「接続ウィザード」を使用して様々なタイプのデータ・ソースに接続します。データ・ソースは、リレーショナル・データベース、Oracle BI EE ソース、Performance Management Architect アプリケーションまたはテキスト・ファイルのデータ・ソースなどです。「接続ウィザード」により、モデリングを行うデータ・ソースを定義したり、オプションでミニスキーマを移入したり、カタログのメタデータ要素を作成したりできます。

また、「接続ウィザード」を使用して、後でキューブの配置時に使用できる Essbase サーバー・インスタンスへの接続を設定することもできます。

「Essbase Studio」をインストールしたら、まず接続ウィザードでデータ・ソースの設定を行います。メタデータ要素、階層、キューブなどのアプリケーション・オブジェクトを作成する前に、それらのデータ・ソースを定義する必要があります。「接続ウィザード」の最初のページで、データ・ソースの情報(データ・ソース・タイプ、ユーザー名、パスワード、サーバー名など)が収集されます。次にデータ・ソースがスクレーピングされて、データ・ソース・テーブル、列およびカタログの結合に関する情報が保管されます。ウィザードの後続の手順では、ミニスキーマの作成や移入を行ったり、カタログのメタデータ要素を作成したり、これらのメタデータ要素をデータ・ソースの物理要素にバインドしたりできます。

データ・ソース接続が作成されると、この接続は、削除するまで Essbase Studio に常に表示されます。次に Essbase Studio サーバーおよびコンソールにログインするときに、データ・ソースに再接続する必要はありません。

注： データ・ソース接続を作成または編集する場合、データ・ソースのデータベース・ユーザー名およびパスワードが必要になります。少なくとも接続先のデータ・ソース(データベース)への読取り権限を持つユーザー名を入力する必要があります。

開始方法は、88 ページの「接続ウィザードを使用した接続の作成」を参照してください。

接続ウィザードを使用した接続の作成

リレーショナルやテキストなどの様々なタイプのデータ・ソースへの接続を作成するには、「接続ウィザード」を使用します。「接続ウィザード」を使用すると、次のいくつかのタスクを完了できます:

- データ・ソース接続パラメータを定義する。

  88 ページの「接続パラメータの定義」を参照してください。
- オプションで、接続に含めるソース・テーブルまたはファイルを選択する。

  89 ページの「接続に含めるテーブルの選択」を参照してください。
- このデータ・ソースのミニスキーマを作成するかどうか選択する。

  89 ページの「ミニスキーマ・オプションの選択」を参照してください。

  ミニスキーマの作成を選択した場合、ミニスキーマの移入もできます。

  90 ページの「ミニスキーマの移入」を参照してください。
- データ・ソース要素からメタデータ要素を作成する。

  90 ページの「メタデータ要素の作成」を参照してください。

これらの手順により、次元要素、階層などのカスタマイズされたメタデータ要素を作成する基盤が提供されます。これらのメタデータ要素は、キューブ・スキーマや Essbase モデルの作成、およびキューブ配置の実行に使用できます。

注： 「接続ウィザード」の最初のページでデータ・ソース・パラメータを定義したら、ウィザードのその他のページの設定はオプションです。

接続パラメータの定義

「接続ウィザード」の最初のページで接続パラメータを定義します。定義するパラメータは、接続先となるデータ・ソースのタイプによって異なります。

- リレーショナル・データ・ソース - Oracle、IBM DB2、Microsoft SQL Server、MySQL、Netezza、Oracle BI EE (「プレゼンテーション層」または「ビジネス・モデル」オプション)、Oracle Real Application Clusters (Oracle RAC)、Teradata な

どのリレーショナル・ソースが含まれます。91 ページの「リレーショナル・ソースの接続パラメータの定義」を参照してください。

- Essbase サーバー・インスタンス-101 ページの「Essbase への接続の作成」を参照してください。
- 次元サーバー・ソース-Performance Management Architect ソースに使用されます。102 ページの「Performance Management Architect データ・ソースへの接続の作成」を参照してください。
- フラットまたはテキスト・ファイルのデータ・ソース-105 ページの「テキスト・ファイル・ソースの接続パラメータの定義」を参照してください。

接続に含めるテーブルの選択

リレーショナル・データ・ソースへのデータ・ソース接続を作成するとき、データ・ソースからすべてのテーブル、ビュー、別名およびシノニムを含めるか、またはテーブルのサブセットのみを含めることができます。

リレーショナル・データ・ソースには、Oracle、IBM DB2、Microsoft SQL Server、MySQL、Netezza、Oracle BI EE、Oracle RAC、Teradata などのリレーショナル・ソースが含まれます。91 ページの「リレーショナル・ソースの接続パラメータの定義」を参照してください。

ミニスキーマ・オプションの選択

ミニスキーマは、データ接続プロセスで作成できます。リレーショナル・データ・ソースから作成されるミニスキーマには、テーブル間の結合が含まれます。Oracle BI EE、次元サーバーまたはテキスト・ファイル・ソースから作成されるミニスキーマには、テーブル間の結合は含まれません。ただし、これらのソースから作成されるミニスキーマは、便利な視覚ツールになります。

- リレーショナル・データ・ソース - Oracle、IBM DB2、Microsoft SQL Server、MySQL、Netezza、Oracle BI EE、Oracle RAC、Teradata などのリレーショナル・ソースが含まれます。95 ページの「リレーショナル・ソースのミニスキーマ・オプションの選択」を参照してください。
- Essbase サーバー・ソース-101 ページの「Essbase への接続の作成」を参照してください。
- 次元サーバー・ソース-Performance Management Architect ソースが含まれます。102 ページの「Performance Management Architect データ・ソースへの接続の作成」を参照してください。
- フラットまたはテキスト・ファイルのデータ・ソース-108 ページの「テキスト・ファイル・ソースのミニスキーマ・オプションの選択」を参照してください。

ミニスキーマの移入

データ・ソース接続プロセス中にミニスキーマを作成する場合、ミニスキーマの移入に使用するソース要素を選択します。ビジネス・ニーズに応じて、すべての要素または要素のサブセットを選択できます。

- リレーショナル・データ・ソース - Oracle、IBM DB2、Microsoft SQL Server、MySQL、Netezza、Oracle BI EE、Oracle RAC、Teradata などのリレーショナル・ソースが含まれます。96 ページの「リレーショナル・ソースのミニスキーマへの移入」を参照してください。
- フラットまたはテキスト・ファイルのデータ・ソース-109 ページの「テキスト・ファイル・ソースのミニスキーマの移入」を参照してください。

メタデータ要素の作成

データ・ソース作成プロセス時に、データ・ソースの物理要素からメタデータ要素を作成できます。このメタデータ要素は、一般にデータ・ソースのテーブルと列を表し、カスタマイズした次元要素や階層など、後で他のメタデータ要素を作成するための基礎となります。

特定のデータ型に関するメタデータ要素の作成の詳細は、次のトピックを参照してください:

- リレーショナル・データ・ソース - Oracle、IBM DB2、Microsoft SQL Server、MySQL、Netezza、Oracle BI EE (「プレゼンテーション層」のみ)、Oracle RAC、Teradata などのリレーショナル・ソースが含まれます。98 ページの「リレーショナル・ソースのメタデータ要素の作成」を参照してください。

  Oracle BI EE (「ビジネス・モデル」のみ)については、98 ページの「Oracle BI EE 次元の作成」を参照してください。
- 次元サーバー・ソース-Performance Management Architect ソースが含まれます。103 ページの「Performance Management Architect ソース用のメタデータ要素の作成」を参照してください。
- フラット・ファイルまたはテキスト・ファイルのデータ・ソース-110 ページの「テキスト・ファイル・ソースからのメタデータ要素の作成」を参照してください。

注： このアクティビティは、Essbase データ・ソースには該当しません。

接続を作成したら、その接続情報は常に Essbase Studio に存在します。これ以降、Essbase Studio サーバーと Essbase Studio コンソールにログインするときにデータ・ソースに再接続する必要はありません。

リレーショナル・ソースへの接続の作成

リレーショナル・データ・ソースへ接続するには、このセクションの手順を使用します。

次のリレーショナル・データ・ソースへの接続が可能です:

- Oracle
- Oracle RAC
- IBM DB2
- Microsoft SQL Server
- MySQL
- Netezza
- Teradata

リレーショナル・ソースと同様に、Oracle BI EE ソース(「プレゼンテーション層」または「ビジネス・モデル」)への接続も可能です。

- プレゼンテーション層接続の場合、ソース内のすべての対応する物理テーブルを表示できません。かわりに、ビューを表示できます。
- ビジネス・モデル接続の場合は、OBI 次元が表示されます。

注: 適切な JDBC ドライバをインストールすると、MySQL、Netezza および Teradata をデータ・ソースとして使用できます。66 ページの「JDBC ドライバの構成」を参照してください。

次のトピックでは、リレーショナル・データ・ソースへの接続を作成するワークフローについて説明しています:

- 91 ページの「リレーショナル・ソースの接続パラメータの定義」。
- 94 ページの「リレーショナル・ソースのテーブルの選択」。
- 95 ページの「リレーショナル・ソースのミニスキーマ・オプションの選択」。
- 96 ページの「リレーショナル・ソースのミニスキーマへの移入」。
- 98 ページの「リレーショナル・ソースのメタデータ要素の作成」。

  Oracle BI EE ソースに接続する場合は、98 ページの「Oracle BI EE 次元の作成」を参照してください。

リレーショナル・ソースの接続パラメータの定義

➤ データ・ソースのパラメータを定義するには:

1. **「データ・ソース・ナビゲータ」で、物理ツリーの「データ・ソース」を右クリックして「新規」を選択し、「データ・ソース」を選択します。**

   「接続ウィザード」の「パラメータの定義」ページが表示されます。
2. **「接続名」に入力します。**
3. **オプションの「説明」を入力します。**
4. **適切な「データ・ソース・タイプ」を選択します。**

たとえば、Microsoft SQL Server データ・ソースへの接続を作成する場合は、ドロップダウン・リストから Microsoft SQL Server を選択します。

5 **Oracle BI EE ユーザー: 次のデータ・ソース接続には Oracle BI EE レイヤーを選択します:**

- 「プレゼンテーション層」 - ビジネス・モデルのカスタマイズされたメタデータのビューです。
- 「ビジネス・モデル」 - メタデータに対する物理データ・ソースのオブジェクトのマッピングです。

6 **Oracle RAC ユーザー: 「RAC サーバー・ノードの追加」ボタンをクリックし、Oracle RAC 接続の少なくとも 1 つのノードに対して次の情報を指定します。**

- 「サービス名」 - 新規の Oracle サービス名を入力するか、以前に入力したサービス名をドロップダウン・リストから選択します。
- 「サーバー名」 - 新規の Oracle RAC サーバー名を入力するか、以前に入力したサーバー名をドロップダウン・リストから選択します。
- 「ポート」 - 新規の Oracle RAC ポート番号を入力するか、以前に入力したポート番号をドロップダウン・リストから選択します。

データ・ソース接続に追加するすべての Oracle RAC サーバー・ノードについて、この手順を繰り返します。

7 **「サーバー名」で、データベースが常駐するサーバーの名前を入力するか、以前に入力したサーバー名をドロップダウン・リストから選択します。**

注： この手順は、Oracle RAC ユーザーには適用されません。

8 **Oracle ユーザー: Oracle データベースに接続する場合、「Oracle SID/サービス名」グループで、Oracle インスタンスの「SID」または「サービス名」に次のように入力します。**

9 **オプション: デフォルト以外のポート番号を使用するには、「ポート」の横にある「デフォルト」チェック・ボックスの選択を解除して、テキスト・ボックスに正しいポート番号を入力します。**

デフォルトのポート番号を使用する場合は、この手順を省略できます。

10 **選択したデータ・ソースの「ユーザー名」および「パスワード」を入力します。**

11 **「データベース名」で、接続先となるデータベースの名前を選択します。**

接続先のデータベースの名前がわからない場合は、「データベース名」の横にある「データベースのフェッチ」ボタンをクリックします。手順 7 で指定したサーバーで使用できるデータベースのリストからデータベースを選択します。

IBM DB2 ユーザーのみ: データベース名を入力する必要があります。選択対象のデータベースのリストは表示されません。

12 **IBM DB2 ユーザーのみ: 「認証メソッド」で、オプションを選択します:**

- 「暗号化なし」
- 「パスワードの暗号化」

- 「ユーザー ID とパスワードの暗号化」
- 「クライアント」

注： データ・ソースの認証メソッドを暗号化なし以外に指定した場合、キューブ配置をストリーミング・モードで実行する必要があります。ストリーミング・モードでの配置を有効にするには、`server.essbase.streamingCubeBuilding` プロパティを true に設定します。このプロパティの詳細は、42 ページの「server.essbase.streamingCubeBuilding」を参照してください。

13 オプション:「接続プール」の設定を変更します。

「プールの最大サイズ」と「キャッシュ・サイズ」が連携して、データ・ソース接続の接続プーリングを制御します。接続プールにより、複数のタスクが同じデータ・ソース接続を同時にクエリーできます。

1. 「プールの最大サイズ」に、接続プールの最大接続数を変更する値を入力します。

   「プールの最大サイズ」は、接続プールの最大接続数を示します。必要な接続数が「キャッシュ・サイズ」で指定した接続数を超えている場合、Essbase Studio サーバーは、「プールの最大サイズ」で指定した値に達するまで、一時的な接続を開きます。必要な接続数が減ると、一時接続が破棄されます。

   「プールの最大サイズ」に表示されているデフォルト値を変更するには、52 ページの「data-source-type.pool.maxsize」の説明に従って、サーバー・プロパティ `pool.maxsize` を変更します。

2. 「キャッシュ・サイズ」に、接続プール内の物理接続の内部キャッシュを変更する値を入力します。

   内部キャッシュは、常に開いている物理接続から構成されます。一部またはすべての接続を同時に使用できます。

   「キャッシュ・サイズ」に表示されているデフォルト値を変更するには、51 ページの「data-source-type.cache.size」の説明に従って、サーバー・プロパティ `cache.size` を変更します。

14 オプション:「接続のテスト」をクリックします。

ウィザードに入力した情報が正しい場合、正常に接続されたことを示すメッセージが表示されます。

ウィザードに入力した情報が正しくない場合、ログイン情報が無効であることを示すメッセージが表示されます。エラーを修正して、正常に接続されるまで再テストしてください。

15「次へ」または「終了」をクリックします。

「次へ」をクリックすると、94 ページの「リレーショナル・ソースのテーブルの選択」の説明のように、ウィザードの「テーブルの選択」ページに進みます。

リレーショナル・ソースのテーブルの選択

リレーショナル・ソースのデータ・ソースへの接続を作成するときに、すべての使用可能なテーブルまたはテーブルのサブセットをデータ・ソース接続に含めることができます。すべての使用可能なビュー、別名テーブルおよびシノニム、またはそれらのオブジェクトのサブセットを含めることもできます。選択が完了したら、Essbase Studio によってデータ・ソースからテーブル、列および結合に関する情報が取得されます。

注： この手順では、「テーブル」は、テーブル、ビュー、別名およびシノニムを表します。

注： Oracle BI EE データ・ソースの場合、この手順を使用するのはプレゼンテーション層ソースのみです(ビジネス・モデル・ソースでは使用しません)。

➤ リレーショナル・データ・ソース接続のテーブルを選択するには:

1 **「接続ウィザード」の「テーブルの選択」ページで、「使用可能なテーブル」の左にある 1 つ以上のボタンをクリックして、「使用可能なテーブル」ボックスに表示されるコンテンツを制御します:**

- 「テーブルの表示」
- 「ビューの表示」
- 「別名の表示」
- 「シノニムの表示」

これらはトグル・ボタンです。特定のテーブルのグループを非表示にするにはもう一度ボタンをクリックします。たとえば、「ビューの表示」をクリックしてデータベース・ビューを表示している場合、もう一度「ビューの表示」をクリックすると非表示になります。

2 **オプション:「使用可能なテーブル」に表示されるテーブルを制限するには、「フィルタ」を入力して「適用」をクリックします。**

たとえば、データベース・テーブルに、業績を示す perf など、特定のビジネス領域を表す接頭辞が付いている場合、次のようにフィルタを入力すれば、その領域のテーブルのみが戻されます:

```
perf*
```

フィルタは、表示するために選択したすべてのテーブル、ビュー、別名またはシノニムに適用されます。

3 **次のいずれかのアクションを行います:**

- データ・ソース接続に含めるテーブルを選択し、「選択したテーブルをデータ・ソースに追加」ボタンをクリックします。

  「データ・ソースのテーブル」ボックスに選択したテーブルが表示されます。

- 「データ・ソースのテーブル」ボックスにすべてのテーブルを追加するには、「すべてのテーブルをデータ・ソースに追加」ボタンをクリックします。

  「データ・ソースのテーブル」ボックスにすべてのテーブルが表示されます。

4 **オプション: 「データ・ソースのテーブル」ボックスからテーブルを削除するには、次のいずれかのタスクを実行します:**

- データ・ソース接続から除外するテーブルを選択し、「選択したテーブルをデータ・ソースから削除」ボタンをクリックします。

  「使用可能なテーブル」ボックスに選択したテーブルが表示されます。

- 「データ・ソースのテーブル」ボックスからすべてのテーブルを削除するには、「すべてのテーブルをデータ・ソースから削除」ボタンをクリックします。

  「使用可能なテーブル」ボックスにすべてのテーブルが表示されます。

5 **オプション: データ・ソースの作成プロセス中に他のユーザーによるカタログ・データベースへの書込みを禁止するには、「参照時にカタログをロック」を選択します。**

注: データ・ソースの作成には、データベースの取得も含まれるため、プロセスが完了するまで、他のユーザーによるソースまたはメタデータ要素の追加、変更または削除は禁止します。

6 **「次へ」または「終了」をクリックします。**

注: 「次へ」または「終了」をクリックすると、データ・ソースのスクレーピング・プロセスが開始します。多数のテーブルを接続に追加する場合、スクレーピング・プロセスに数分かかることがあります。

「次へ」をクリックすると、ウィザードの「ミニスキーマの選択」ページに移動します(95 ページの「リレーショナル・ソースのミニスキーマ・オプションの選択」を参照)。

注: 後でテーブルのサブセットのみが含まれているデータ・ソースにテーブルを追加する場合、データ・ソースの増分更新を実行して追加することもできます。111 ページの「データ・ソース接続の増分更新の実行」を参照してください。

リレーショナル・ソースのミニスキーマ・オプションの選択

「接続ウィザード」の「ミニスキーマの選択」ページで、次のアクションを実行できます:

- 新しいミニスキーマ・ダイアグラムの作成

注： リレーショナル・データ・ソースにアクセスしている場合、新しいミニスキーマ・ダイアグラムを作成するときに「イントロスペクションを使用した階層の検出」オプションを選択することもできます。このオプションを選択すると、データ・ソースが細部まで検査され、階層情報などのより詳細な情報が戻されます。フラット・ファイルまたはテキスト・ファイルのデータ・ソースにアクセスしている場合「イントロスペクションを使用した階層の検出」オプションは使用できません。

- ミニスキーマ・ダイアグラムの作成のスキップ

  96 ページの「リレーショナル・ソース用のミニスキーマ作成のスキップ」を参照してください。

- イントロスペクションを使用してデータ・ソース接続プロセスを完了します。

注： Oracle BI EE のデータ・ソースにアクセスしている場合、ミニスキーマを作成できますが、含めることができるのは自己結合(1 つのファイル内の列間の結合)のみです。

リレーショナル・ソースのミニスキーマの作成

ミニスキーマの作成方法は、イントロスペクション・オプションを選択する方法としない方法があります。このトピックでは、両方の手順を説明します。

➤ イントロスペクションを使用しないで新しいミニスキーマを作成するには:

1. **「ミニスキーマの選択」で、「新規スキーマ・ダイアグラムを作成します」を選択します。**
2. **ミニスキーマの名前を入力するか、表示されるデフォルトの名前を使用します。**
3. **「次へ」または「終了」をクリックします。**

   「次へ」をクリックすると、96 ページの「リレーショナル・ソースのミニスキーマへの移入」で説明されているように、ウィザードの「ミニスキーマの移入」ページに移動します。

リレーショナル・ソース用のミニスキーマ作成のスキップ

➤ ミニスキーマの作成をスキップするには:

1. **「ミニスキーマ・ダイアグラムをスキップ」を選択します。**
2. **「次へ」または「終了」をクリックします。**

   「次へ」をクリックすると、ウィザードの「メタデータ要素の作成」ページに移動します。このページの説明は、98 ページの「リレーショナル・ソースのメタデータ要素の作成」に記載されています。

リレーショナル・ソースのミニスキーマへの移入

ミニスキーマへの移入対象は、データ・ソースで使用可能なテーブルのリストから選択します。また、ミニスキーマ用に選択したテーブルにフィルタを適用したり、関連オブジェクトを追加したりすることもできます。

注： 94 ページの「リレーショナル・ソースのテーブルの選択」でデータ・ソースに含めるために選択したテーブルは、デフォルトで、接続ウィザードの「ミニスキーマの移入」ページにある「スキーマのテーブル」リストに含まれています。

➤ ミニスキーマを移入するには:

1 **「ミニスキーマの移入」で、ミニスキーマからテーブルを削除するには、「スキーマのテーブル」リストから、ミニスキーマに含めないテーブルを選択し、「選択したテーブルをスキーマから削除」ボタンをクリックします。**

テーブルが「使用可能なテーブル」リストに移動されます。

すべてのテーブルを「使用可能なテーブル」リストに移動するには、「スキーマからすべてのテーブルを削除」ボタンをクリックします。

2 **ミニスキーマにテーブルを追加するには、「使用可能なテーブル」リストから追加するテーブルを選択し、「選択したテーブルをスキーマに追加」ボタンをクリックします。**

テーブルが「スキーマのテーブル」リストに移動されます。

3 **オプション: 「使用可能なテーブル」にリストされているテーブルにフィルタを適用するには、「フィルタ」テキスト・ボックスにフィルタを入力し、「適用」をクリックします。**

注： フィルタは、「使用可能なテーブル」リストにリストされているテーブルのみに適用されます。

4 **オプション: ミニスキーマに関連オブジェクトを追加するには、「スキーマのテーブル」リストでテーブルを 1 つ以上選択し、「関連オブジェクトの追加」ボタンをクリックします。**

たとえば、テーブルを「スキーマのテーブル」リストに追加した場合、そのテーブルを選択して「関連オブジェクトの追加」をクリックすると、そのテーブルに結合されているテーブルがあればすべてリストに追加されます。

5 **オプション: 別のデータ・ソースからミニスキーマにテーブルを移入するには:**

1. 「接続」で、接続先とするデータ・ソースを選択します。
2. 「データベース」でデータベースを選択して、このミニスキーマ用にどのデータベースからテーブルを選択するかを指定します。
3. ミニスキーマへのテーブルの追加元であるすべてのデータ・ソースに対して、手順 1 から手順 4 までを繰り返します。

6 **「次へ」または「終了」をクリックします。**

「次へ」をクリックすると、ウィザードの「メタデータ要素の作成」ページに移動します。このページの説明は、98 ページの「リレーショナル・ソースのメタデータ要素の作成」に記載されています。

リレーショナル・ソースのメタデータ要素の作成

データ・ソースのテーブルと列からメタデータ要素を作成します。これらのメタデータ要素は、後で次元要素および階層の作成に使用でき、さらにそれらを使用してキューブ・スキーマの構築できます。

注： この手順は、「プレゼンテーション層」オプションを選択して作成されたOracle BI EE データ・ソースに適用されます。Oracle BI EE のデータ・ソースを「ビジネス・モデル」オプションを使用して作成した場合は、98 ページの「Oracle BI EE 次元の作成」を参照してください。

➤ メタデータ要素を作成するには:

1 **「メタデータ要素の作成」の「使用可能なソース・オブジェクト」リストで、「メタデータ・ナビゲータ」のこのデータ・ソースのフォルダに追加するテーブルと列を選択します。**

- すべてのテーブル内のすべての列に対してメタデータ要素を作成するには、「使用可能なソース・オブジェクト」リストの一番上のデータ・ソース名の横にあるチェック・ボックスを選択します。
- 選択したテーブルのすべての列に対してメタデータ要素を作成するには、追加するテーブルの名前の横にあるチェック・ボックスを選択します。
- 1 つのテーブルで選択した列に対してメタデータ要素を作成するには、そのテーブルを展開し、追加する列の名前の横にあるチェック・ボックスを選択します。

2 **オプション: デフォルト・フォルダ以外のフォルダにメタデータ要素を保管するには:**

1. 「参照」をクリックします。
2. 「フォルダの選択」で、メタデータ要素を保管するフォルダに移動し、必要に応じてフォルダを展開します。

   または、「新規フォルダ」をクリックしてテキスト・ボックスにフォルダ名を入力し、「OK」をクリックします。

3 **「終了」をクリックします。**

「メタデータ・ナビゲータ」にリストされた新しいメタデータ要素を表示します。

ミニスキーマを作成した場合、Essbase Studio コンソールのミニスキーマ作業領域に新しいミニスキーマが表示されます。

Oracle BI EE 次元の作成

「ビジネス・モデル」オプションを選択した場合、Oracle BI EE 接続の作成時に、Essbase Studio は次元のソース・データベースを参照します。Essbase Studio は、選択された次元から階層と階層に含まれる次元要素を導出し、「メタデータ・ナビゲータ」に追加します。

注： Essbase Studio が、作成された次元要素のキー・バインディング式とキャプション・バインディング式を生成する方法の詳細は、365 ページの「Oracle BI EE のビジネス・モデル次元要素のキー・バインディングとキャプション・バインディングの生成ルール」を参照してください。

➤ 次元要素と階層を含むメタデータ要素を作成するには:

1 **ウィザードの「OBIEE 次元の作成」ページで、「使用可能な OBI 次元」リストから、メタデータ要素を作成する次元(階層およびそれに関連する次元要素を含む)を選択します。**

- 「使用可能な OBI 次元」リストの一番上のデータ・ソース名の横のチェック・ボックスを選択すると、データ・ソース内のすべての使用可能な OBI 次元からメタデータ要素を作成します。
- メタデータ要素を作成する OBI 次元の横のチェック・ボックスのみを選択してください。

選択した次元は、「メタデータ・ナビゲータ」でこのデータ・ソースのフォルダに追加されます。

2 **「OBI ファクト・テーブルの選択」で、ファクト・テーブルとして指定するテーブルを選択します。**

- 「OBI ファクト・テーブルの選択」リストの一番上のデータ・ソース名の横のチェック・ボックスを選択すると、すべてのファクト・テーブル候補を指定します。
- ファクト・テーブルとして指定するファクト・テーブル候補の横のチェック・ボックスのみを選択してください。

注： Oracle BI EE バージョン 11.1.1.6 以降のソースを取得すると、「OBI ファクト・テーブルの選択」パネルにファクト・テーブルのみがリストされます。11.1.1.6 より前の Oracle BI EE バージョンでは、すべてのテーブル(標準テーブルおよびファクト・テーブル)がリストされます。

3 **オプション: デフォルト・フォルダ以外のフォルダにメタデータ要素を保管するには:**

1. 「参照」をクリックします。
2. 「フォルダの選択」で、メタデータ要素を保管するフォルダに移動し、必要に応じてフォルダを展開します。

   または、「新規フォルダ」をクリックしてテキスト・ボックスにフォルダ名を入力し、「OK」をクリックします。

4 **オプション: 「キューブ・スキーマと Essbase モデルの作成」チェック・ボックスを選択します。**

Oracle BI EE のファクト・テーブルおよび次元を使用すると、メジャーと階層を指定して、キューブ・スキーマおよび Essbase モデルを作成できます(100 ページの「Oracle BI EE のキューブ・スキーマおよび Essbase モデルの作成」を参照)。

5 **「次へ」をクリックして Oracle BI EE 要素からキューブ・スキーマを作成するか、「終了」をクリックします。**

「メタデータ・ナビゲータ」にリストされた新しいメタデータ要素(次元要素および階層)を表示します。

ミニスキーマを作成した場合、Essbase Studio コンソールのミニスキーマ作業領域にミニスキーマが表示されます。

Oracle BI EE のキューブ・スキーマおよび Essbase モデルの作成

98 ページの「Oracle BI EE 次元の作成」の手順 4 でキューブ・スキーマおよび Essbase モデルを作成するように選択した場合は、ウィザードのこのページで、作成した Oracle BI EE 次元要素および階層を使用してキューブ・スキーマを設定します。これにより Essbase モデルが自動的に作成されます。

キューブ・スキーマおよび Essbase モデルの詳細は、第 10 章「キューブ・スキーマ」および第 11 章「Essbase プロパティ」を参照してください。

➤ Oracle BI EE のファクト・テーブルと次元を使用してキューブ・スキーマおよび Essbase モデルを設定するには:

1 **ウィザードの「OBIEE キューブ・スキーマの作成」ページで、「使用可能なファクト・テーブル」リストから Oracle BI EE ファクト・テーブルを展開し、列を選択して「メジャー」リストに移動し、右矢印をクリックします。**

ファクト・テーブル・レベルで選択を行うことはできません。ただし、リストされたファクト・テーブルから 1 つ以上の列を選択することはできます。

注: また、ドラッグ・アンド・ドロップを使用して、「使用可能なファクト・テーブル」リストと「メジャー」リストの間で次元を移動することもできます。

2 **「使用可能な次元要素」リストで、Oracle BI EE 次元を選択して「階層」リストに移動し、右矢印をクリックします。**

注: また、ドラッグ・アンド・ドロップを使用して、「使用可能な次元要素」リストと「階層」リストの間で次元を移動することもできます。

3 **「終了」ボタンをクリックします。**

「メタデータ・ナビゲータ」にリストされた新しいメタデータ要素(次元要素、階層、キューブ・スキーマおよび Essbase モデル)を表示します。これらは、メタデータ要素のデフォルト・フォルダか、98 ページの「Oracle BI EE 次元の作成」の手順 3 で指定したフォルダのいずれかにあります。

注: この手順で作成した Essbase モデルでは、モデル・プロパティ「ASO ストレージ・モデル」および「重複するメンバー名のサポート」がデフォルトで有効になっています。

これらの要素は、他のリレーショナル・ソースから作成した要素と同じように操作できます。

Essbase への接続の作成

特定の Essbase サーバー・インスタンスまたは Essbase サーバー・クラスタへの接続を作成し、後でキューブ配置プロセス時に使用できます。

SSL データ暗号化オプションを選択することも可能です。SSL を使用すると、Essbase Studio サーバーと Essbase 間の通信が保護されます。

➤ Essbase インスタンスまたはクラスタへの接続を定義するには:

1 **「データ・ソース・ナビゲータ」で、物理ツリーの「データ・ソース」を右クリックして「新規」を選択し、「データ・ソース」を選択します。**

2 **「接続名」に入力します。**

3 **オプション: 「説明」を入力します。**

4 **「データ・ソース・タイプ」で、「Essbase サーバー」を選択します。**

5 **「パラメータ」の下にある「Essbase サーバー」に、この Essbase サーバー・インスタンスが存在するコンピュータの名前を入力するか、Essbase サーバーのクラスタ名を入力します。**

   注: クラスタ名では大文字と小文字が区別されます。デフォルトのクラスタ名は `EssbaseCluster-1` です。

6 **オプション: デフォルト以外のポート番号を使用するには、「ポート」の横にある「デフォルト」チェック・ボックスの選択を解除して、テキスト・ボックスに正しいポート番号を入力します。**

   デフォルト・ポート番号を使用する場合は、この手順を省略します。

7 **オプション: Essbase サーバー・クラスタに接続している場合は、「クラスタ」チェック・ボックスを選択します。**

   注: Essbase サーバー・クラスタは、Essbase 接続の作成前に設定しておく必要があります。クラスタの設定の詳細は、Oracle Enterprise Performance Management System インストールおよび構成ガイド、および Oracle Enterprise Performance Management System High Availability and Disaster Recovery Guide を参照してください。

8 **Essbase サーバー・インスタンスまたはクラスタの「ユーザー名」と「パスワード」を入力します。**

9 **「データ暗号化」方法を選択します:**

   - 「暗号化なし」 - Essbase Studio サーバーと Essbase サーバー・インスタンスまたはクラスタ間の通信のデフォルト・モード。
   - 「SSL」 - 接続中の Essbase サーバー・インスタンスまたはクラスタが、TCP/IP 経由の Secure Socket Layer (SSL)プロトコルに対応するよう構成されている場合はこの方法を選択します。

SSL を使用すると、Essbase Studio サーバーと Essbase 間の通信が保護されます。

注： SSL を選択し、Essbase サーバーが SSL をサポートするように構成されていない場合は、接続が確立されず、エラー・メッセージが表示されます。

10 「接続のテスト」をクリックします。

ウィザードに入力した情報が正しい場合、正常に接続されたことを示すメッセージが表示されます。

ウィザードに入力した情報が正しくない場合、ログイン情報が無効であることを示すメッセージが表示されます。エラーを修正して、正常に接続されるまで再テストしてください。

11 「終了」をクリックします。

「データ・ソース・ナビゲータ」に接続名が表示されます。後でキューブ配置時に、この名前を「Essbase サーバー接続」として選択できます。

Performance Management Architect データ・ソースへの接続の作成

次のトピックでは、Performance Management Architect データ・ソースへの接続を作成するためのワークフローを説明します:

- 102 ページの「Performance Management Architect ソースの接続パラメータの定義」。
- 103 ページの「Performance Management Architect ソース用のメタデータ要素の作成」。

Performance Management Architect ソースの接続パラメータの定義

➤ Performance Management Architect データ・ソースへの接続を定義するには:

1 「データ・ソース・ナビゲータ」で、物理ツリーの「データ・ソース」を右クリックして「新規」を選択し、「データ・ソース」を選択します。

2 「接続名」に入力します。

3 **オプション:** 「説明」を入力します。

4 「データ・ソース・タイプ」で「次元サーバー」を選択します。

Performance Management Architect は、次元サーバーの例の 1 つです。

5 「パラメータ」の下にある「サーバー名」に、Performance Management Architect 次元サーバーが存在するコンピュータの名前を入力します。

6 **オプション:** デフォルト以外のポート番号を使用するには、「ポート」の横にある「デフォルト」チェック・ボックスの選択を解除して、テキスト・ボックスに正しいポート番号を入力します。

デフォルト・ポート番号を使用する場合は、この手順を省略します。

7 この Performance Management Architect 次元サーバー・インスタンスの「ユーザー名」および「パスワード」を入力します。

8 「データ暗号化」方法を選択します:

- 「暗号化なし」 - これが、Essbase Studio サーバーおよび Performance Management Architect 次元サーバー間の通信のデフォルト・モードです。
- 「SSL」 - 接続中の Performance Management Architect 次元サーバー・インスタンスが、TCP/IP 経由の Secure Socket Layer (SSL)プロトコルに対応するよう構成されている場合はこの方法を選択します。

  SSL を使用すると、Essbase Studio サーバーと Performance Management Architect 次元サーバー間の通信が保護されます。

  注 : 「SSL」を選択していて、Performance Management Architect 次元サーバーが SSL をサポートするよう構成されていない場合は、接続が確立されず、エラー・メッセージが表示されます。

9 「接続のテスト」をクリックします。

ウィザードに入力した情報が正しい場合、正常に接続されたことを示すメッセージが表示されます。

ウィザードに入力した情報が正しくない場合、ログイン情報が無効であることを示すメッセージが表示されます。エラーを修正して、正常に接続されるまで再テストしてください。

10 「次へ」または「終了」をクリックします。

103 ページの「Performance Management Architect ソース用のメタデータ要素の作成」で説明されているように、「次へ」をクリックすると、ウィザードの「メタデータ要素の作成」ページに移動します。

「終了」をクリックした場合、「データ・ソース・ナビゲータ」でデータ・ソース名を展開すると、このデータ・ソースの次元およびメンバーが表示されます。

Performance Management Architect ソース用のメタデータ要素の作成

データ・ソース内の Performance Management Architect 物理テーブルからメタデータ要素を作成します。これらのメタデータ要素は、後で次元要素および階層の作成に使用でき、さらにそれらを使用してキューブ・スキーマの構築できます。

➤ メタデータ要素を作成するには:

1 「使用可能なソース・オブジェクト」リスト内の「メタデータ要素の作成」で、「メタデータ・ナビゲータ」内の、このデータ・ソース用フォルダに追加するオブジェクトを選択します。

- すべてのオブジェクトを「メタデータ・ナビゲータ」に追加するには、「使用可能なソース・オブジェクト」リスト上部のデータ・ソース名の横にあるチェック・ボックスを選択します。
- 選択した要素のみを「メタデータ・ナビゲータ」に追加するには、追加する要素の名前の横にあるチェック・ボックスを選択します。
- または、オブジェクトの特定の子を「メタデータ・ナビゲータ」に追加するには、要素を展開して、追加する子オブジェクトの横にあるチェック・ボックスを選択します。

2 **オプション: デフォルト・フォルダ以外のフォルダにメタデータ要素を保管するには:**

1. 「参照」をクリックします。
2. 「フォルダの選択」で、メタデータ要素を保管するフォルダに移動し、必要に応じてフォルダを展開します。

   または、「新規フォルダ」をクリックしてテキスト・ボックスにフォルダ名を入力し、「OK」をクリックします。

3 **「終了」をクリックします。**

「メタデータ・ナビゲータ」に、新しいメタデータ要素がリスト表示されていることを確認します。また、「データ・ソース・ナビゲータ」でデータ・ソース名を展開すると、このデータ・ソースの次元およびメンバーが表示されます。

テキスト・ファイルのデータ・ソースへの接続の作成

テキスト・ファイルのデータ・ソースの作成プロセスは、リレーショナル・データ・ソースの作成プロセスと少し異なります。フラット・ファイル・ソースへデータ・ソース接続を作成するには、104 ページの「テキスト・ファイルのデータ・ソースについて」を確認し、次の手順を完了します:

- 105 ページの「テキスト・ファイル・ソースの接続パラメータの定義」。
- 107 ページの「テキスト・ファイルのモデリング」。
- 108 ページの「テキスト・ファイル・ソースのミニスキーマ・オプションの選択」。
- 110 ページの「テキスト・ファイル・ソースからのメタデータ要素の作成」。

テキスト・ファイルのデータ・ソースについて

テキスト・ファイルのデータ・ソースは、データベース管理者またはシステム管理者が事前定義された場所に置くフラット・テキスト・ファイルから構成されます。指定されたフォルダから開始する 1 つの場所にテキスト・ファイルのソースを保持することで、管理者はユーザーがサーバー上の他のファイルにアクセスするのを制限して、セキュリティの手段を提供できます。

テキスト・ファイルのデフォルトの場所は、次の通りです:

EPM_ORACLE_HOME/products/Essbase/EssbaseStudio/Server/datafiles

注： server.properties ファイルの server.datafile.dir プロパティは、テキスト・ファイルの場所を定義するために使用します。テキスト・ファイルのデフォルトの場所を使用しない場合は、server.datafile.dir プロパティの指定について 41 ページの「server.datafile.dir」を参照してください。

テキスト・ファイルは、server.datafile.dir で指定された場所の個々のディレクトリに格納されています。各テキスト・ファイル・ディレクトリはテキスト・ファイルのデータ・ソースとみなされ、任意の数の個別テキスト・ファイルを含むことができます。

Essbase Studio は、データ・ソースの作成時にテキスト・ファイルの場所にアクセスします。「テキスト・ファイルの場所」ダイアログ・ボックスで、/<root>は server.datafile.dir で定義されたディレクトリの場所と等価です。このディレクトリ内のフォルダはテキスト・ファイルのデータ・ソースであり、ユーザーはここから選択できます。

テキスト・ファイル・ソースの接続パラメータの定義

ウィザードのこのページで、テキスト・ファイルのデータ・ソース接続パラメータを定義します。

注： Essbase Studio では、Microsoft Excel ファイルはデータ・ソースとしてサポートされていません。

➤ フラット・ファイルのデータ・ソースを作成するには:

1. **「データ・ソース・ナビゲータ」で、物理ツリーの「データ・ソース」を右クリックして「新規」を選択し、「データ・ソース」を選択します。**
2. **「接続名」に入力します。**
3. **オプション:「説明」を入力します。**
4. **「データ・ソース・タイプ」として「テキスト・ファイル」を選択します。**
5. **テキスト・ファイルのデータ・ソースの「場所」を指定し、「参照」をクリックします。**

   テキスト・ファイルのデータ・ソースの場所は、server.properties ファイルの server.datafile.dir プロパティを使用して指定されます。詳細は、104 ページの「テキスト・ファイルのデータ・ソースについて」を参照してください。
6. **「テキスト・ファイルの場所」ダイアログ・ボックスで、アクセスするテキスト・ファイルのデータ・ソースを含むディレクトリを選択します。**

注： データ・ソースごとに選択できるディレクトリは 1 つのみです。Essbase Studio では、同じディレクトリの異なるサブディレクトリにあるテキスト・ファイルはサポートされていません。

注： テキスト・ファイルのデータ・ソースのディレクトリ名と、個々のテキスト・ファイル名にスペースを含めることはできません。

7 **「選択したディレクトリのコンテンツ」の下にリストされたファイルを見直し、それが目的のテキスト・ファイルのデータ・ソースであることを確認します。**

8 **「テキスト・ファイルの場所」ダイアログ・ボックスでの選択が終了したら、「OK」をクリックして「接続ウィザード」の「接続パラメータの定義」ページに戻ります。**

9 **オプション:「レコードのスキップ」で、データ・ソースのテキスト・ファイルごとに、先頭からスキップするレコードの数を入力します。**

たとえば、データ・ソースのテキスト・ファイルには、各ファイルの先頭 3 行に識別コメントが含まれる場合があります。この場合、「レコードのスキップ」に 3 と入力します。

ここで選択した項目は、「接続ウィザード」の「テキスト・ファイルのモデリング」でファイル単位で上書きできます。

10 **オプション: スキップしたレコードの次の行に列名が含まれる場合、「最初の行の列名」を選択します。**

このチェック・ボックスの選択を解除すると、「接続ウィザード」によりデフォルトの名前(Col_0、Col_1 など)が各ファイルの列ごとに割り当てられます。この列名は、107 ページの「テキスト・ファイルのモデリング」の説明に従って、テキスト・ファイルをモデリングする場合にファイル単位で後から変更できます。

11 **次のいずれかの「区切り記号」を選択します:**

- 「カンマ」
- 「タブ」
- 「スペース」
- 「カスタム」-データ・ソースでカンマ、タブまたはスペース以外の区切り記号が使用されている場合、その区切り記号を指定します。

指定した区切り記号は、以降はテキスト・ファイル接続のデフォルトとして使用されます。ただし、ウィザードの「テキスト・ファイルのモデリング」ページで、特定のテキスト・ファイルのデータ・ソースに対してこの設定を変更できます。

12 **「次へ」をクリックしてウィザードの「テキスト・ファイルのモデリング」ページに戻ります(107 ページの「テキスト・ファイルのモデリング」を参照してください)。**

テキスト・ファイルのモデリング

ウィザードのこのページで、データ・ソース用にモデリングするテキスト・ファイルを選択します。各ファイルのレコードを個別に表示したり、ファイル単位でモデリング・タスクを実行したりできます。

➤ テキスト・ファイルをモデリングするには:

1 **テキスト・ファイル名の横にあるチェック・ボックスを選択し、名前をハイライトして下部のテキスト・ボックスにサンプル・レコードを表示します。**

Essbase Studio によって最大 19 レコードが下部のテキスト・ボックスに表示されます。

また、「すべて選択」ボタンをクリックしてデータ・ソースのファイルをすべて選択し、各ファイルの名前をハイライトしてレコードを表示することもできます。

注: すべてのチェック・マークの選択を解除するには、「全消去」をクリックします。選択されているチェック・ボックスをクリックして選択を解除することもできます。

2 **オプション:「フィルタ」テキスト・ボックスで、テキスト・ファイル・セットのフィルタを入力します。**

たとえば、接頭辞が prod のテキスト・ファイルをフィルタするには、`prod*.txt` と入力します。prod で始まるテキスト・ファイルのリストが表示されます。

3 **オプション: 必要に応じて該当のファイルにモデリングを実行し、テキスト・ファイルの横にあるチェック・ボックスを選択してファイル名をハイライトします。**

1. 「レコードのスキップ」で、データ・ソースの各テキスト・ファイルの先頭からスキップするレコードの数を入力します。

   たとえば、データ・ソースのテキスト・ファイルには、各ファイルの最初の 3 レコードに識別用のコメントが含まれていることがあります。この場合、「レコードのスキップ」に 3 と入力します。

   注: スキップできるレコード数は、最大 1000 行までです。

2. レコードをスキップした後の最初の行に列名が含まれている場合、「最初の行の列名」を選択します。

   このチェック・ボックスを選択しない場合、接続ウィザードによって、デフォルト名が各ファイルの各列に割り当てられます。

3. 次のいずれかの「区切り記号」を選択します:

   - 「カンマ」
   - 「タブ」
   - 「スペース」
   - 「カスタム」-データ・ソースでカンマ、タブまたはスペース以外の区切り記号が使用されている場合、その区切り記号を指定します。

4 **オプション: 列のデータ型を上書きするには、作業領域の列のヘッダーをクリックして、新しいデータ型を選択するためのコンテキスト・メニューを表示します。**

使用可能なデータ型は次のとおりです:

- テキスト
- 整数
- 大きい整数
- 小数

5 **オプション: 列を非表示にするには、作業領域の列のヘッダーをクリックしてコンテキスト・メニューを表示し、「列を非表示」を選択します。**

作業領域のすでに非表示になっている列を表示するには、コンテキスト・メニューから「非表示列を表示」を選択します。

6 **オプション: 選択したファイルの複数列の列名およびデータ型を変更するには:**

1. 最初の列のヘッダーをクリックして、コンテキスト・メニューから「プロパティ」を選択し、「列のプロパティ」ダイアログ・ボックスを起動します。
2. 列名を変更するには、新しい名前を「名前」テキスト・ボックスに入力します。
3. 「データ型」グループから、選択した列の列タイプを選択します。

   注: データ型をテキストから整数に変更すると、サンプル・データを表示するときにエラーになりますが、キューブの配置は正常に完了しています。
4. 「次へ」をクリックして、ファイルの次列のプロパティを表示します。
5. 必要な列のデータ型をすべて変更するまで、手順 6.2 から手順 6.4 を繰り返し、「OK」をクリックして「テキスト・ファイルのモデリング」ダイアログ・ボックスに戻ります。

7 **108 ページの「テキスト・ファイル・ソースのミニスキーマ・オプションの選択」で説明されているように、「次へ」をクリックして「接続ウィザード」の「ミニスキーマの選択」ページを表示します。**

テキスト・ファイル・ソースのミニスキーマ・オプションの選択

テキスト・ファイルのデータ・ソースのミニスキーマを作成できます。サポートされるのはファイル内の自己結合のみですが、テキスト・ファイルのデータ・ソースのミニスキーマはファイルの表示に便利です。

「接続ウィザード」の「ミニスキーマの選択」ページで、次のアクションを実行できます:

- 新しいミニスキーマ・ダイアグラムの作成; ミニスキーマには、自己結合(1 つのファイル内の列間の結合)のみ含まれます。ファイル間の結合は含まれません。

109 ページの「テキスト・ファイル・ソースのミニスキーマの作成」を参照してください。

- ミニスキーマ・ダイアグラムの作成のスキップ

  109 ページの「テキスト・ファイル・ソース用のミニスキーマ作成のスキップ」を参照してください。

注： テキスト・ファイルでは、「イントロスペクションを使用した階層の検出」オプションはサポートされません。

テキスト・ファイル・ソースのミニスキーマの作成

➤ ミニスキーマを作成するには:

1 **「ミニスキーマの選択」で、「新規スキーマ・ダイアグラムを作成します」を選択します。**

2 **ミニスキーマの名前を入力するか、表示されるデフォルトの名前を使用します。**

3 **「次へ」または「終了」をクリックします。**

「次へ」をクリックすると、ウィザードの「ミニスキーマの移入」ページへ移動します。このページの説明は、109 ページの「テキスト・ファイル・ソースのミニスキーマの移入」を参照してください。

テキスト・ファイル・ソース用のミニスキーマ作成のスキップ

➤ ミニスキーマの作成プロセスをスキップするには、「ミニスキーマの選択」の「ミニスキーマ・ダイアグラムをスキップ」オプションが選択されていることを確認し、「次へ」をクリックします。

110 ページの「テキスト・ファイル・ソースからのメタデータ要素の作成」で説明されているように、「接続ウィザード」の「メタデータ要素の作成」ページが表示されます

テキスト・ファイル・ソースのミニスキーマの移入

テキスト・ファイルのデータ・ソース用にミニスキーマを作成したら、データ・ソース内の使用可能な物理テーブル(ファイル)のリストから選択して、ミニスキーマに移入します。また、ミニスキーマ用に選択したファイルにフィルタを適用したり、関連オブジェクトを追加することもできます。

➤ ミニスキーマを移入するには:

1 **「ミニスキーマの移入」で、「スキーマのテーブル」リストから、ミニスキーマに含めないテーブルを選択します。次いで「選択したテーブルをスキーマから削除」ボタンをクリックして、テーブルを「使用可能なテーブル」リストに移動します。**

すべてのテーブルを「使用可能なテーブル」リストに移動するには、「スキーマからすべてのテーブルを削除」ボタンをクリックします。

2 **「スキーマのテーブル」リストにテーブルを追加するには、追加するテーブルを「使用可能なテーブル」リストから選択し、「選択したテーブルをスキーマに追加」ボタンをクリックします。**

すべてのテーブルをミニスキーマに追加するには、「スキーマにすべてのテーブルを追加」ボタンをクリックします。

3 **オプション: 「使用可能なテーブル」にリストされているテーブルにフィルタを適用するには、「フィルタ」テキスト・ボックスにフィルタを入力し、「適用」をクリックします。**

注: フィルタは、「使用可能なテーブル」リストにリストされているテーブルのみに適用されます。

注: 「関連オブジェクトの追加」ボタンは、テキスト・ファイルのデータ・ソースのファイルには適用されません。

4 **オプション: ミニスキーマに別のデータ・ソースからオブジェクトを移入するには:**

1. 「接続」で、接続先とするデータ・ソースを選択します。
2. 「データベース」で、ミニスキーマへのオブジェクトの移入元になるデータベースを選択します。
3. このミニスキーマへのオブジェクトの追加元になるデータ・ソースすべてに対して、手順 1 から手順 4 までを繰り返します。

5 **「次へ」または「終了」をクリックします。**

110 ページの「テキスト・ファイル・ソースからのメタデータ要素の作成」で説明されているように、「次へ」をクリックすると、ウィザードの「メタデータ要素の作成」ページに移動します。

テキスト・ファイル・ソースからのメタデータ要素の作成

テキスト・ファイルのデータ・ソースのファイルのコンテンツからメタデータ要素を作成します。これらのメタデータ要素は、後で次元要素および階層の作成に使用でき、さらにそれらを使用してキューブ・スキーマの構築できます。

➤ テキスト・ファイルのデータ・ソースの物理オブジェクトから次元要素を作成するには:

1 **「メタデータ要素の作成」で、要素を選択して、「メタデータ・ナビゲータ」にあるこのデータ・ソースのフォルダに追加します。**

- すべてのテーブル内のすべての列に対してメタデータ要素を作成するには、「使用可能なソース・オブジェクト」リストの一番上のデータ・ソース名の横にあるチェック・ボックスを選択します。

- 選択したテーブルのすべての列に対してメタデータ要素を作成するには、追加するテーブルの名前の横にあるチェック・ボックスを選択します。
- 1 つのテーブルで選択した列に対してメタデータ要素を作成するには、そのテーブルを展開し、追加する列の名前の横にあるチェック・ボックスを選択します。

2 **オプション: デフォルト・フォルダ以外のフォルダに次元要素を保管するには:**

1. 「参照」をクリックします。
2. 「フォルダの選択」で、「メタデータ・ナビゲータ」を使用して次元要素を保管するフォルダを選択します。必要に応じてフォルダを展開します。

   または、「新規フォルダ」をクリックしてフォルダ名を入力し、「OK」をクリックします。

3 **「終了」をクリックします。**

メタデータ要素を作成した場合、「メタデータ・ナビゲータ」にメタデータ要素が表示されます。

次元要素や階層などのその他のメタデータ要素を作成する準備が整いました。

データ・ソース接続の同期

個々のデータ・ソース接続に対して次のタスクを実行します:

- 111 ページの「データ・ソース接続の増分更新の実行」
- 115 ページの「データ・ソース接続からのテーブルの削除」
- 116 ページの「データ・ソース接続のテーブルのリフレッシュ」

データ・ソース接続の増分更新の実行

物理データ・ソースで使用可能なテーブルのサブセットのみ含むようにデータ・ソース接続を設定できます。この接続にさらにデータ・ソース・テーブルを追加することを後で決定した場合は、データ・ソースの増分更新を実行して、テーブルを接続に追加できます。

リレーショナルまたはテキスト・ファイルのデータ・ソースに対して増分更新を実行できます。次のトピックで説明されています:

- 111 ページの「リレーショナル・データ・ソースの増分更新の実行」
- 113 ページの「テキスト・ファイルのデータ・ソースの増分更新の実行」

リレーショナル・データ・ソースの増分更新の実行

この手順は、リレーショナル・データ・ソースにのみ適用されます。テキスト・ファイルのデータ・ソースを処理する場合は、113 ページの「テキスト・ファイルのデータ・ソースの増分更新の実行」を参照してください。

注： この手順では、「テーブル」は、テーブル、ビュー、別名およびシノニムを表します。

➤ リレーショナル・データ・ソースの増分更新を実行するには:

1 **「データ・ソース・ナビゲータ」で、物理ツリーの「データ・ソース」の下の、テーブルを追加するリレーショナル・データ・ソースを右クリックして、「増分更新」を選択します。**

「増分更新」ダイアログ・ボックスが表示されます。

2 **「使用可能なテーブル」の左にあるボタンをクリックして、「使用可能なテーブル」ボックスに表示されるコンテンツを制御します:**

- 「テーブルの表示」
- 「ビューの表示」
- 「別名の表示」
- 「シノニムの表示」

これらはトグル・ボタンです。特定のテーブルのグループを非表示にするにはもう一度ボタンをクリックします。たとえば、「ビューの表示」をクリックしてデータベース・ビューを表示している場合、もう一度「ビューの表示」をクリックすると非表示になります。

3 **「使用可能なテーブル」ボックスで、次のいずれかのアクションを実行します:**

- データ・ソース接続に含めるテーブルを選択して、「選択したテーブルをデータ・ソースに追加」をクリックします。

  「データ・ソースのテーブル」ボックスに選択したテーブルが表示されます。

- 「データ・ソースのテーブル」ボックスにテーブルを追加するには、「すべてのテーブルをデータ・ソースに追加」をクリックします。

  「データ・ソースのテーブル」ボックスにすべてのテーブルが表示されます。

注： 多数のテーブルを接続に追加する場合、増分更新プロセスに数分かかることがあります。

4 **オプション: データ・ソースに追加したテーブルを削除するには、「データ・ソースのテーブル」ボックスで、次のいずれかのアクションを実行します:**

- データ・ソース接続から除外するテーブルを選択して、「選択したテーブルをデータ・ソースから削除」をクリックします。

  削除するために選択したテーブルが「使用可能なテーブル」ボックスに表示されます。

- 新規に追加したすべてのテーブルをデータ・ソースから除外するには、「すべてのテーブルをデータ・ソースから削除」をクリックします

  これで、追加したテーブルがデータ・ソース接続から削除されます。

注： 現在の増分更新セッションで追加したテーブルのみを削除できます。すでにデータ・ソースに含まれているテーブルは削除できません。「データ・ソースのテーブル」ボックスで使用不可になっているテーブルは削除できません。

5 **オプション: 使用可能なデータ・ソース・テーブルの一部のみを表示するには、「フィルタ」を入力します。**

たとえば、データベース・テーブルに、Performance を表す perf など、特定のビジネス領域の接頭辞が付いている場合、次のようなフィルタを入力して、その領域のテーブルのみを戻すことができます:

```
perf*
```

6 **データ・ソースへのテーブルの追加が終了したら、「OK」をクリックします。**

テキスト・ファイルのデータ・ソースの増分更新の実行

この手順はテキスト・ファイルのデータ・ソースのみに適用されます。リレーショナル・データ・ソースで作業する場合、111 ページの「リレーショナル・データ・ソースの増分更新の実行」を参照してください。

注： この手順はテキスト・ファイルのデータ・ソースのみに適用されます。

➤ テキスト・ファイルのデータ・ソースの増分更新を実行するには:

1 **「データ・ソース・ナビゲータ」で、物理ツリーの「データ・ソース」の下の、テーブルを追加するテキスト・ファイル・データ・ソースを右クリックして、「増分更新」を選択します。**

「増分更新」ダイアログ・ボックスが表示されます。

2 **オプション: 「フィルタ」テキスト・ボックスで、テキスト・ファイル・セットのフィルタを入力します。**

たとえば、接頭辞が prod のテキスト・ファイルをフィルタするには、`prod*.txt` と入力します。prod で始まるテキスト・ファイルのリストが表示されます。

3 **テキスト・ファイルのリストで、データ・ソースに追加するテキスト・ファイルの横にあるチェック・ボックスを選択します。**

4 **次の手順に従って、選択したファイルで必要に応じてモデリングを実行します:**

1. リストで選択したテキスト・ファイルの名前の 1 つをハイライトします。
2. 「レコードのスキップ」で、データ・ソースの各テキスト・ファイルの先頭からスキップするレコードの数を入力します。

   たとえば、データ・ソースのテキスト・ファイルには、各ファイルの最初の 3 レコードに識別用のコメントが含まれていることがあります。この場合、「レコードのスキップ」に 3 と入力します。

   注： スキップできるレコード数は、最大 1000 行までです。

3. レコードをスキップした後の最初の行に列名が含まれている場合、「最初の行の列名」を選択します。

   このチェック・ボックスを選択しない場合は、Essbase Studio によって、デフォルト名が各ファイルの各列に割り当てられます。

5 **次のいずれかの「区切り記号」を選択します:**

- 「カンマ」
- 「タブ」
- 「スペース」
- 「カスタム」-データ・ソースでカンマ、タブまたはスペース以外の区切り記号が使用されている場合、その区切り記号を指定します。

6 **オプション: 列のデータ型を上書きするには、作業領域の列のヘッダーをクリックして、新しいデータ型を選択するためのコンテキスト・メニューを表示します。**

使用可能なデータ型は次のとおりです:

- テキスト
- 整数
- 大きい整数
- 小数

7 **オプション: 列を非表示にするには、作業領域の列のヘッダーをクリックしてコンテキスト・メニューを表示し、「列を非表示」を選択します。**

作業領域のすでに非表示になっている列を表示するには、コンテキスト・メニューから「非表示列を表示」を選択します。

8 **オプション: 選択したファイルの複数列の列名およびデータ型を変更するには:**

1. 最初の列のヘッダーをクリックして、コンテキスト・メニューから「プロパティ」を選択し、「列のプロパティ」ダイアログ・ボックスを起動します。
2. 列名を変更するには、新しい名前を「名前」テキスト・ボックスに入力します。
3. 「データ型」グループから、選択した列の列タイプを選択します。

   注: データ型をテキストから整数に変更すると、サンプル・データを表示するときにエラーになりますが、キューブの配置は正常に完了しています。

4. 「次へ」をクリックして、ファイルの次列のプロパティを表示します。
5. 必要な列のデータ型がすべて変更されるまで手順 6.2 から手順 6.4 を繰り返し、「OK」をクリックして「増分更新」ダイアログ・ボックスに戻ります。

9 **データ・ソースに追加するテキスト・ファイルの選択とモデリングが終了したら、「OK」をクリックして「増分更新」ダイアログ・ボックスを閉じます。**

データ・ソース接続からのテーブルの削除

使用しなくなったテーブルまたは特定のデータ・ソース接続で使用しないテーブルの削除を検討します。

テーブルは、そのテーブルに依存している Essbase Studio メタデータ要素がないかぎり、データ・ソース接続から削除できます。たとえば、テーブルが、Essbase Studio カタログに格納されている階層の構築に使用されている場合は、先に依存階層およびその要素を削除してからでないと、そのテーブルを削除できません。

注： 削除できるのは、Oracle BI EE データ・ソースを含むリレーショナル・データ・ソースのテーブルと、フラット・ファイル・データ・ソースのファイルのみです。

➤ データ・ソース接続からテーブルを削除するには:

1 **「データ・ソース・ナビゲータ」で、物理ツリーの「データ・ソース」の下のデータ・ソース接続に属するテーブルを選択します。**

複数のテーブルを選択するには、「[Shift]」キーまたは「[Ctrl]」キーを使用します。

注： データ・ソース接続からすべてのテーブルを削除する場合は、データ・ソース接続を削除して(125 ページの「接続の削除」を参照)、新規の接続を作成することをお薦めします。

2 **右クリックして「削除」を選択します。**

3 **メタデータ要素で使用されているテーブルまたは他のテーブルへの結合があるテーブルがない場合は、「削除の確認」ダイアログ・ボックスで「はい」をクリックします。**

選択したテーブルに依存しているメタデータ要素がない場合は、「削除の確認」ダイアログ・ボックスが表示されます。

4 **選択したテーブルが他のメタデータ要素で使用されているか、他のテーブルへの結合がある場合は、次のアクションを実行します:**

1. 「削除」ダイアログ・ボックスの「配置エラーおよび警告」セクションにリストされているメタデータ要素を確認し、「取消し」をクリックします。

ヒント： 「取消し」をクリックする前に、「エラー/警告の保存」ボタンをクリックし、テーブルのリストをテキスト・ファイルとして保存します。このファイルは、手順 4.2 で参照用として使用できます。

2. 示されているメタデータ要素およびすべての関連メタデータ要素を削除します(167 ページの「メタデータ要素の削除」を参照)。

注： 系列ビューアを使用すると、Essbase モデルおよびキューブ配置など、依存しているメタデータ要素を簡単に確認できます。「削除」ダイアログ・ボックスを閉じたら、「メタデータ・ナビゲータ」でメタデータ要素を右クリックし、「系列の表示」を選択します。

3. 依存しているメタデータ要素をすべて削除したら、「データ・ソース・ナビゲータ」で削除するテーブルを選択して「削除」を選択し、プロセスを繰り返します。
4. 選択したテーブルに、他のテーブルとの結合が含まれている場合、「削除」ダイアログ・ボックスの「テーブル・ステータス」で、削除するテーブル名の横のチェック・ボックスを選択し、「削除」をクリックします。

   依存しているメタデータ要素が他にない場合、選択したテーブル、および選択したテーブルにリストされている結合は削除されます。

5 **「データ・ソース・ナビゲータ」でテーブルが削除されていることを確認します。**

データ・ソース接続のテーブルのリフレッシュ

データ・ソース接続の作成後にデータ・ソースで行われたスキーマ変更を検出するには、テーブルをリフレッシュします。リフレッシュで検出されるスキーマ変更には、次のものがあります:

- 新しい列
- 削除された列
- 列のデータ型の変更; たとえば、varchar から integer など

テーブルは、接続レベルとテーブル・レベルでリフレッシュできます。テーブル・レベルでリフレッシュした場合、リフレッシュは選択したテーブルでのみ実行されます。

注: リフレッシュできるのは、Oracle BI EE データ・ソースを含むリレーショナル・データ・ソースのみです。

次のトピックでは、リフレッシュ・プロセスについて説明します:

- 116 ページの「テーブルのリフレッシュ」
- 117 ページの「結合の追加」

テーブルのリフレッシュ

注: この手順は、Oracle BI EE データ・ソースを含むリレーショナル・データ・ソースにのみ適用されます。

➤ データ・ソース接続テーブルをリフレッシュするには:

1 **次のいずれかのアクションを行います:**

- データ・ソース接続の 1 つ以上のテーブルを右クリックします。

  複数のテーブルを選択するには、「[Shift]」キーまたは「[Ctrl]」キーを使用します。
- データ・ソース接続名を右クリックします。

2 **ショートカット・メニューから「リフレッシュ」を選択します。**

物理データ・ソースと同期していないテーブルには、「リフレッシュ」ダイアログ・ボックスでフラグが設定されています。

注： 選択したテーブルまたは接続が物理データ・ソースとすでに同期している場合は、メッセージが表示され、リフレッシュ・アクションは実行されません。

3 **「リフレッシュ」ダイアログ・ボックスで、次の手順を実行します:**

1. 「変更されたテーブル」で、テーブルを選択します。
2. 「テーブルの列を変更しました」で、情報を確認します:
   - 「列名」
   - 「状態」 - 列が新規、更新などのいずれであるか
   - 「データ型」
   - 「変更の詳細」
3. 「変更されたテーブル」にリストされている各テーブルの変更を確認した後で、リフレッシュするテーブルを選択し、「リフレッシュ」をクリックします。

   注： 旧列がまだ Essbase Studio メタデータ要素で使用されている場合は、テーブルをリフレッシュできません。テーブルをリフレッシュする前に、そのような要素を削除する必要があります。「リフレッシュ」ダイアログ・ボックスに、リフレッシュできないテーブルが、影響を受けるメタデータ要素とともに表示されます。
4. いずれかの結合がリフレッシュ後に無効になる場合は、「リフレッシュ」ダイアログ・ボックスで確認します。
5. 削除する結合が含まれているテーブルを選択し、「OK」をクリックします。

   テーブルに関連付けられているすべての結合が削除されます。結合を個別に選択することはできません。

   または、「すべての結合を選択」チェック・ボックスを選択します。

   注： 削除する結合を含むテーブルは明示的に選択する必要があります。リフレッシュ時に自動的には削除されません。

4 **117 ページの「結合の追加」の手順を実行して、プロセスを完了します。**

結合の追加

116 ページの「テーブルのリフレッシュ」の手順を完了した後で、結合を追加できます。データ・ソースがリフレッシュされた後、Essbase Studio は選択する潜在的な結合のリストを表示します。

注： この手順は、Oracle BI EE データ・ソースを含むリレーショナル・データ・ソースにのみ適用されます。

➤ リフレッシュ・プロセスの一部として新しい結合を追加するには:

1 **116 ページの「テーブルのリフレッシュ」の手順を実行します。**

リフレッシュ手順を実行した後で、「新規結合」ダイアログ・ボックスに、潜在的な新しい結合のリストが表示されます。

2 **「新規結合」ダイアログ・ボックスで、データ・ソース接続のリフレッシュ・プロセスの一部として追加する結合を選択します。**

3 **「OK」をクリックします。**

更新されたテーブルを示す成功メッセージが表示されます。

データ・ソース接続の操作

「データ・ソース・ナビゲータ」で次のタスクを実行します:

- 118 ページの「データ・ソース接続でのイントロスペクションの実行」。
- 118 ページの「データ・ソース接続プロパティの編集」。
- 125 ページの「接続リストのリフレッシュ」。
- 125 ページの「接続の削除」。
- 125 ページの「わかりやすい名前の表示」

データ・ソース接続でのイントロスペクションの実行

イントロスペクションは、物理データ・ソースにメタデータ要素があるかどうかを検査する方法です。イントロスペクションを実行するとき、データ・ソースの構造情報が検査され、ファクト・テーブル、次元テーブル、階層、別名および属性が検出されます。イントロスペクションから導出されたメタデータ要素は、キューブ・スキーマの作成と、オプションで Essbase モデルの作成に使用されます。

イントロスペクションは、データ・ソース接続の作成プロセス中に実行できます。また、イントロスペクションは、既存のデータ・ソース接続で実行できます。

➤ 既存のデータ・ソース接続に対するイントロスペクションを開始するには、「データ・ソース・ナビゲータ」で物理ツリーの「データ・ソース」から、イントロスペクションを実行するデータ・ソースを右クリックして「イントロスペクション」を選択します。

イントロスペクション実行の手順は、第 6 章「イントロスペクション」を参照してください。

データ・ソース接続プロパティの編集

データ・ソース接続情報を編集できます。

注： 次に示す個々の手順はすべてオプションです。変更する情報によって実行するタスクが異なります。

➤ データ・ソースのプロパティを編集するには:

1 **「データ・ソース・ナビゲータ」で、物理ツリーの「データ・ソース」の下の編集するデータ・ソースを右クリックして、「プロパティ」を選択します。**

2 **データ・ソース接続名を変更するには、新しい名前を「接続名」に入力します。**

3 **データ・ソースの説明を変更するには、新しい説明を「接続の説明」に入力します。**

4 **ダイアログ・ボックスの「パラメータ」セクションで、データ・ソース・タイプに応じて次のタスクを実行します。**

リレーショナル・ソース:

注： データベース名は変更できません。

1. 「サーバー名」で、データベースが存在するサーバーの名前を変更します。

   注： サーバーの接続情報を変更する場合、新しいサーバーでは、元のデータベースと同じタイプ、名前およびテーブル構造のデータベースをホストする必要があります。

2. **Oracle ユーザーのみ:** Oracle データベースに接続する場合、「Oracle SID/サービス名」グループで、Oracle インスタンスの「SID」または「サービス名」に次のように入力します。
3. ポート番号を変更するには、「ポート」の横にある「デフォルト」チェック・ボックスの選択が解除されていることを確認し、新しいポート番号を入力します。
4. データベースの「ユーザー名」および「パスワード」を変更します。
5. 「接続のテスト」をクリックします。

   ダイアログ・ボックスで入力した情報が正しければ、接続成功の確認メッセージが表示されます。

   ダイアログ・ボックスで入力した情報に誤りがある場合、無効なログイン情報が入力されたことを示すメッセージが表示されます。エラーを修正して、正常に接続されるまで再テストしてください。
6. 手順 5 に進みます。

Essbase サーバー接続:

1. 「Essbase サーバー」で、Essbase サーバーが存在するコンピュータの名前を変更します。
2. ポート番号を変更するには、「ポート」の横にある「デフォルト」チェック・ボックスの選択が解除されていることを確認し、新しいポート番号を入力します。
3. **オプション:** Essbase サーバー・クラスタに接続している場合は、「クラスタ」チェック・ボックスを選択します。

注： Essbase サーバー・クラスタは、Essbase 接続の作成前に設定しておく必要があります。クラスタの設定の詳細は、Oracle Enterprise Performance Management System インストールおよび構成ガイド、および Oracle Enterprise Performance Management System High Availability and Disaster Recovery Guide を参照してください。

4. この Essbase サーバー・インスタンスの「ユーザー名」および「パスワード」を変更します。
5. 「データ暗号化」方法を選択します:
   - 「暗号化なし」 - Essbase Studio サーバーと Essbase サーバー・インスタンスまたはクラスタ間の通信のデフォルト・モード。
   - 「SSL」 - 接続中の Essbase サーバー・インスタンスまたはクラスタが、TCP/IP 経由の Secure Socket Layer (SSL)プロトコルに対応するよう構成されている場合はこの方法を選択します。

     SSL を使用すると、Essbase Studio サーバーと Essbase 間の通信が保護されます。

     注： Essbase サーバーが SSL をサポートするように構成されていない場合は、接続が確立されず、エラー・メッセージが表示されます。
6. 「接続のテスト」をクリックします。

   ダイアログ・ボックスで入力した情報が正しければ、接続成功の確認メッセージが表示されます。

   ダイアログ・ボックスで入力した情報に誤りがある場合、無効なログイン情報が入力されたことを示すメッセージが表示されます。エラーを修正して、正常に接続されるまで再テストしてください。
7. 手順 5 に進みます。

次元サーバー・ソース:

1. 「サーバー名」で、Performance Management Architect が存在するコンピュータの名前を変更します。
2. ポート番号を変更するには、「ポート」の横にある「デフォルト」チェック・ボックスの選択が解除されていることを確認し、新しいポート番号を入力します。
3. この Performance Management Architect インスタンスの「ユーザー名」および「パスワード」を変更します。
4. 手順 5 に進みます。

テキスト・ファイル・ソース:

1. テキスト・ファイルのデータ・ソースの「場所」を変更するには、「参照」をクリックします。
2. 「テキスト・ファイルの場所」ダイアログ・ボックスで、アクセスするテキスト・ファイルのデータ・ソースを含むディレクトリを選択します。

注： データ・ソースごとに選択できるディレクトリは 1 つのみです。Essbase Studio では、同じディレクトリの異なるサブディレクトリにあるテキスト・ファイルはサポートされていません。

3. 「選択したディレクトリのコンテンツ」の下にリストされたファイルを見直し、それが目的のテキスト・ファイルのデータ・ソースであることを確認します。
4. 手順 5 に進みます。

5 **「適用」をクリックし、「OK」をクリックします。**

6 **データ・ソース名を変更した後、更新した名前を「データ・ソース・ナビゲータ」で表示する場合は、変更したデータ・ソース接続名を右クリックして「リフレッシュ」を選択します。**

物理ツリーに新しい名前が表示されます。

ソース・テーブルおよび列のプロパティの表示

ミニスキーマ・テーブル、リレーショナル・データ・ソース、テキスト・ファイルのデータ・ソースおよびユーザー定義テーブルのテーブル・プロパティや列プロパティを表示できます。詳細は、次のトピックを参照してください:

- 121 ページの「リレーショナル・ソース・テーブルおよびユーザー定義テーブルのプロパティの表示」
- 122 ページの「テキスト・ファイルのソース・ファイルのプロパティの表示」

リレーショナル・ソース・テーブルおよびユーザー定義テーブルのプロパティの表示

➤ リレーショナル・ソース・テーブルまたはユーザー定義テーブルのプロパティを表示するには:

1 **「データ・ソース・ナビゲータ」で、物理ツリーの「データ・ソース」の下の適切なデータ・ソース接続を展開し、プロパティを表示するリレーショナル・ソース・テーブルまたはユーザー定義テーブルに移動します。**

2 **テーブルを右クリックして「プロパティ」を選択し、「全般」プロパティを表示します。**

次のようなプロパティが表示されます:

- 物理ソース・テーブル名
- データ・ソース接続名
- プライマリ・キー(ある場合)
- コメント

テーブルの列のプロパティを表示するには、「列」タブをクリックします。122 ページの「リレーショナル・ソース列およびユーザー定義テーブル列のプロパティの表示」を参照してください。また、122 ページの「ユーザー定義テーブルの基となるステートメントの表示」も参照してください。

3 **オプション**: ミニスキーマ・ダイアグラムからミニスキーマ・テーブルのプロパティを表示している場合、「コメント」を入力します。

注： リレーショナル・テーブルまたはユーザー定義テーブルのプロパティを表示している場合は、「コメント」フィールドを使用できません。

ミニスキーマ内のテーブルおよび列のプロパティを表示するには、133 ページの「ミニスキーマ・テーブルおよび列のプロパティの表示」を参照してください。

リレーショナル・ソース列およびユーザー定義テーブル列のプロパティの表示

➤ リレーショナル・ソースの列またはユーザー定義テーブルの列のプロパティを表示するには:

1 「データ・ソース・ナビゲータ」で、物理ツリーの「データ・ソース」の下の適切なデータ・ソース接続を展開し、プロパティを表示するリレーショナル・ソース列またはユーザー定義テーブル内の列に移動します。

2 列名を右クリックして「プロパティ」を選択し、「列」タブを選択します。

次の列のプロパティが表示されます:

- 列がキー列かどうか
- 列名
- 列タイプ
- NULL が許可されているかどうか

ミニスキーマ内のテーブルおよび列のプロパティを表示するには、133 ページの「ミニスキーマ・テーブルおよび列のプロパティの表示」を参照してください。

ユーザー定義テーブルの基となるステートメントの表示

➤ ユーザー定義テーブルの基となる SQL 文を表示するには:

1 「データ・ソース・ナビゲータ」で、物理ツリーの「データ・ソース」の下のユーザー定義テーブルに移動します。

2 テーブルを右クリックして「プロパティ」を選択し、「ユーザー定義ステートメント」タブを選択します。

ユーザー定義テーブルの基となる SQL 文が表示されます。

テキスト・ファイルのソース・ファイルのプロパティの表示

➤ ソース・テキスト・ファイルのプロパティを表示するには:

1 「データ・ソース・ナビゲータ」で、物理ツリーの「データ・ソース」の下のテキスト・ファイルに移動します。

2 **ファイル名を右クリックし、「プロパティ」を選択して、「全般」プロパティを表示します。**

ファイルの列プロパティを表示する方法は、123 ページの「テキスト・ファイル・ソースの列のプロパティの表示」を参照してください。

次のようなプロパティが表示されます:

- 物理ソース・ファイルの名前
- データ・ソース接続名
- テキスト・ファイル名
- ファイル内の列の名前およびそれらの列のデータ型

テキスト・ファイル・ソースの列のプロパティの表示

➤ テキスト・ファイル・ソースの列のプロパティを表示するには:

1 **「データ・ソース・ナビゲータ」で、物理ツリーの「データ・ソース」の下のテキスト・ファイル・ソース列に移動します。**

2 **列名を右クリックして「プロパティ」を選択し、「列」タブを選択します。**

次の列のプロパティが表示されます:

- 物理列の名前
- データ・ソース接続名
- テキスト・ファイル名
- 列のデータ型

サンプル・データの表示

「データ・ソース・ナビゲータ」で、データ・ソースのテーブルの使用可能なサンプル・データを 200 行まで表示できるようになります。また、テーブル内のソース列からサンプル・データを表示することもできます。

また、「メタデータ・ナビゲータ」でも、次元要素に関連する使用可能なサンプル・データを最大 200 行まで表示できます。

「データ・ソース・ナビゲータ」からのサンプル・データの表示

➤ データ・ソースのサンプル・データを表示するには:

1 **「データ・ソース・ナビゲータ」でデータ・ソースに移動し、データ・ソースを展開します。**

2 **次のいずれかまたは両方のアクションを実行します:**

- データ・ソースのテーブルのデータを表示するには、テーブルを右クリックして、「サンプル・データの表示」を選択します。
- データ・ソースのテーブルにある 1 つの列のデータを表示するには、テーブルを展開し、列を右クリックして、「サンプル・データの表示」を選択します。

または、テーブル内の 1 つ以上の列のデータを表示するには、「[Ctrl]」キーを押したままデータを表示する列をクリックします。次に右クリックして、「サンプル・データの表示」を選択します。

サンプル・データが、コンソールの作業領域の新しいタブに表示されます。

注： 最大 200 行のサンプル・データが戻されます。

3 **サンプル・データのウィンドウを閉じるには、ウィンドウのタブの「X」をクリックします。**

注： サンプル・データの表示ウィンドウが開いている間にソース・データが追加、編集または削除されても、開いているウィンドウ内のデータは自動的にはリフレッシュされません。変更されたデータを表示するには、ウィンドウを閉じて、「サンプル・データの表示」コマンドを再発行する必要があります。

メタデータ・ナビゲータからのサンプル・データの表示

➤ 次元要素のサンプル・データを表示するには:

1 **「メタデータ・ナビゲータ」で、サンプル・データを表示する次元要素に移動します。**

2 **要素を右クリックして、「サンプル・データの表示」を選択します。**

選択した要素のキャプション・バインディング式とキー・バインディング式が同一の場合は、「サンプル・データの表示」のみが表示され、サブメニューは表示されません。

選択した要素のキャプション・バインディング式とキー・バインディング式が異なる場合は、「キャプション・バインディングを使用」オプションと「キー・バインディングを使用」オプションの両方が、「サンプル・データの表示」のサブメニュー項目として表示されます。

要素のキー・バインディング式が「遅延」の場合、「キー・バインディングを使用」オプションは使用不可になります。

要求したサンプル・データが、コンソールの作業領域の新しいタブに表示されます。

注： 最大 200 行のサンプル・データが戻されます。

たとえば、TBC サンプル・データベースを使用して、キャプション・バインディング式が `connection : \'tbcSource'::'tbc.region'.'REGION' . toString`; キー・バインディング式が `connection : \'tbcSource'::'tbc.region'.'REGIONID' . toString` の REGION テーブルの REGION 列に基づいて次元要素を作成するとします。

「キャプション・バインディングを使用」オプションを選択すると、サンプル・データの結果は次のようになります:

```
        REGION
1   East
2   West
3   South
4   Central
```

「キー・バインディングを使用」オプションを選択すると、サンプル・データの結果は次のようになります:

```
        REGIONID
1   1
2   2
3   3
4   4
```

3 **サンプル・データのウィンドウを閉じるには、ウィンドウのタブの「X」をクリックします。**

接続リストのリフレッシュ

➤ 「データ・ソース・ナビゲータ」でデータ・ソース接続のリストをリフレッシュするには、物理ツリーのルートにある「データ・ソース」を右クリックして「リフレッシュ」を選択します。

接続の削除

データ・ソース接続から作成されたメタデータ要素がない場合にのみ、データ・ソース接続を削除できます。

➤ データ・ソース接続を削除するには、「データ・ソース・ナビゲータ」で、削除するデータ・ソースの名前を右クリックして「削除」を選択します。

注 : メタデータ要素の作成に使用された接続は、関連付けられているメタデータ要素が削除された後でのみ削除できます。

わかりやすい名前の表示

注 : この機能は、Microsoft SQL Server データ・ソース接続のみに適用されます。

デフォルトでは、Microsoft SQL Server データ・ソースに接続すると、「データ・ソース・ナビゲータ」にデータベース・テーブル名が次のようなわかりやすいフォーマットで表示されます:

```
schema_name
.
```

```
database_name
.
table_name
```

「わかりやすい名前の表示」オプションを使用不可にし、次のフォーマットでテーブルのフル・ネームを表示できます:

```
database_name
.
table_name
```

➤ Microsoft SQL Server データ・ソース接続のテーブルのフル・ネームを表示するには、「データ・ソース・ナビゲータ」でデータ・ソース接続名を右クリックして「わかりやすい名前の表示」を選択します。

チェック・マークの選択が解除されます。

➤ Microsoft SQL Server データ・ソース接続のテーブルのわかりやすい名前を表示するには、「データ・ソース・ナビゲータ」でデータ・ソース接続名を右クリックして「わかりやすい名前の表示」を選択します。

チェック・マークは、「わかりやすい名前の表示」が選択されていることを示しています。

ユーザー定義テーブルの作成

このダイアログ・ボックスを使用して 1 つ以上のユーザー定義テーブルを定義して、データベースからデータの仮想ビューを作成します。

127 ページの「ユーザー定義テーブルについて」および127 ページの「ユーザー定義テーブルの例」を参照してください。

➤ 「ユーザー定義テーブル」ダイアログ・ボックスにアクセスするには:

1. **「データ・ソース・ナビゲータ」で適切なデータ・ソースの名前をハイライトします。**
2. **「新規」>「ユーザー定義テーブル」を選択します。**
3. **新しいユーザー定義テーブルを定義します:**
   - 「接続」-テーブルの追加先のデータ・ソースを選択します。
   - 「テーブル名」-テーブルの名前を入力します。
   - 「テーブル定義」-データ・ソースにより必要とされる SQL 構文を使用して、ユーザー定義テーブルを作成するための SQL ステートメントを入力します。このステートメントにより仮想ビュー内のテーブル名が定義されます。また、ステートメントには任意の数のソース列名を含めることができます。テーブル定義は、データ・ソース内の論理テーブルの基準となります。

注： Oracle では、データベース・テーブルを指定するときにフル・パスを含めることをお薦めします。

ユーザー定義テーブルについて

ユーザー定義テーブルは、RDBMS ではなく Essbase Studio で作成する論理テーブルです。標準の RDBMS ビューとして機能するこれらの仮想テーブルは、標準の RDBMS テーブルまたはビューが使用される場所であればどこでも使用できます。

ユーザー定義テーブルにより、リレーショナル・スキーマや、Essbase Studio で生成された SQL を変更することなくモデルを作成および編集できます。データベース・スキーマが Essbase Studio の使用に適した構造になっていない場合は、ユーザー定義テーブルを使用して、データベース・スキーマを変更することなくモデルやアウトラインを構築できます。

Essbase Studio によってユーザー定義テーブルの作成に使用する SQL コマンドが確認され、データベース・スキーマの仮想ビューの確立が可能になります。

注意:

- ユーザー定義 SQL は実行時まで検証されません。詳細は、364 ページの「ユーザー定義 SQL の検証」を参照してください。
- Essbase Studio におけるユーザー定義 SQL に対する権限は、データ・ソース・ユーザーに割り当てられている権限から継承されます。詳細は、364 ページの「ユーザー定義 SQL のデータ・ソース権限」を参照してください。
- Oracle では、ユーザー定義テーブルの SQL 文でデータベース・テーブルを指定するときにフル・パスを含めることをお薦めします。
- テキスト・ファイルのデータ・ソースでは、ユーザー定義テーブルを作成できません。

127 ページの「ユーザー定義テーブルの例」も参照してください。

ユーザー定義テーブルの例

Essbase Studio では、「ユーザー定義テーブルの定義」ダイアログ・ボックスの「テーブル定義」テキスト・ボックス内に SQL 構文を入力することにより、RDBMS ビューと同様のユーザー定義テーブルを作成します。このトピックの例では、RDBMS で使用する SQL 文と Essbase Studio で使用する SQL 文間の相違点を示します。

次の例は、RDBMS 内のビューを作成する SQL 文を示しています:

```
   CREATE VIEW

View_Prod_Proddim
(Caffeinated, FamilyID, Ounces, PackageType,
ProductID, SKU, Family) as

   SELECT
```

```
a.caffeinated, a.familyid, a.ounces, a.pkgtype,
a.productid, a.sku, b.family

FROM
   tbc.product a, tbc.productdim b

WHERE
   a.sku=b.sku
```

前述の例を使用して Essbase Studio でユーザー定義テーブルを作成する場合、SQL 構文には前述の SQL 文の CREATE VIEW 部分は含まれません。作成する SQL テーブル定義の構文は、SELECT から始まる文の部分から開始されます。

注： Oracle では、データベース・テーブルを指定するときにフル・パスを含めることをお薦めします。

次のユーザー定義テーブルの SQL 例は、テーブル名が `catalog.table` フォーマットで表されるデータベース用に記述されています:

```
   SELECT

a.caffeinated,
a.familyid, a.ounces, a.pkgtype,
a.productid, a.sku, b.family

   WHERE
    a.sku=b.sku

   FROM
    tbc.product a, tbc.productdim b
```

次は、テーブルが `catalog.schema.table` フォーマットで表されるデータベース用に記述された同じ例です:

```
   SELECT

a.caffeinated,
a.familyid, a.ounces, a.pkgtype,
a.productid, a.sku, b.family

   WHERE
    a.sku=b.sku

   FROM
    tbc.user1.product a, tbc.user1.productdim b
```

5 ミニスキーマ

この章の内容

ミニスキーマについて 129
ミニスキーマの作成または編集 130
ミニスキーマの操作 132

ミニスキーマについて

ミニスキーマは、1 つ以上のデータ・ソース接続内のテーブルまたはテキスト・ファイルのグラフィカル・モデルです。ミニスキーマは、データ・ソース接続の作成プロセスで作成できます。または、後で接続済のデータ・ソースからミニスキーマを作成することもできます。

多くのデータ・ソースのタイプでは、ミニスキーマはエンティティの相関図の論理モデルと同様に表示されます。テーブル間の結合が表示され、ソース・データベース内の値が次元として分類されます。

ミニスキーマによって、今後のモデリングの基盤となる関連領域を作成するために、すべてのテーブルまたはテーブルのサブセットをデータ・ソース接続に使用できます。

ミニスキーマでは次のタスクを実行できます:

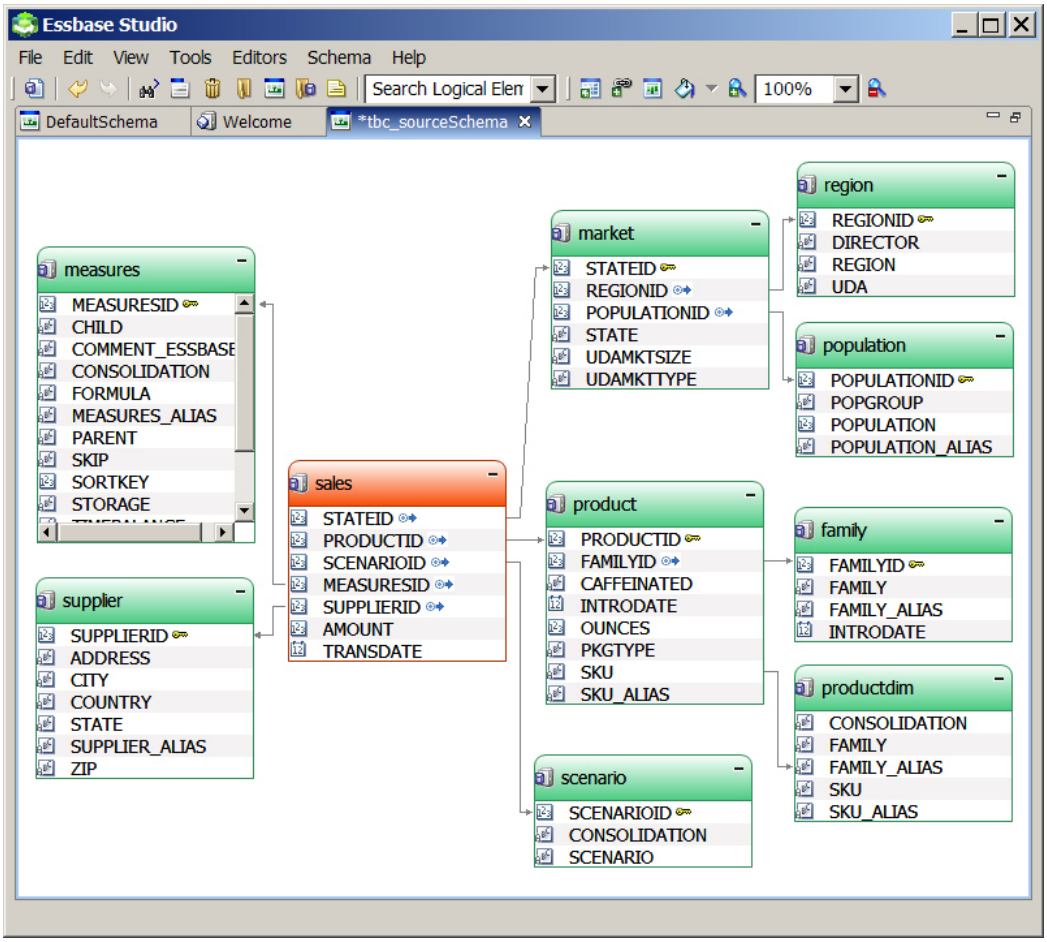
- 手動による、または適用可能な結合のリストから結合を選択する検査を使用したテーブル間の結合の追加または削除

  注： Oracle BI EE またはテキスト・ファイルのデータ・ソースにアクセスしている場合、ミニスキーマを作成できますが、ファイル内でサポートされるのは自己結合のみです。
- テーブルまたは列からのサンプル・データの表示
- メタデータ要素の作成
- 視覚的な手段として役立つ、ミニスキーマ・ダイアグラムのテーブル見出しへの色の追加

ミニスキーマは、95 ページの「リレーショナル・ソースのミニスキーマ・オプションの選択」で、データ・ソース作成プロセスの一部として説明されています。

図 1 はサンプルのミニスキーマです。

図 1　ミニスキーマの例



データ・ソースごとに任意の数のミニスキーマを作成できます。複数のミニスキーマは、データ・ソースの別ビューを作成する方法として便利です。別のミニスキーマには、たとえばデータ・ソース内のテーブルと列のサブセットや、追加の結合を含めることができます。

ミニスキーマの作成または編集

ミニスキーマを作成または編集するには、次のタスクを実行します:

- 130 ページの「ミニスキーマの全般プロパティの設定」の説明に従って、新規のミニスキーマに名前を付けて接続を設定し、コメントを入力または編集します。
- 131 ページの「ミニスキーマでのテーブルの追加または削除」に従って、ミニスキーマのテーブルを追加または削除し、オプションでフィルタを追加します。

ミニスキーマの全般プロパティの設定

➤ ミニスキーマの全般プロパティを設定するには:

1 「データ・ソース・ナビゲータ」でアクションを実行します:

- 適切なデータ・ソース接続に移動して「ミニスキーマ」フォルダを右クリックし、「新規」、「ミニスキーマ」の順に選択し、「ミニスキーマ・ウィザード」を起動します。

  注： また、物理ツリーの最上位にある「データ・ソース」を右クリックして、「新規」、「ミニスキーマ」の順に選択することもできます。

- 適切なデータ・ソース接続に対して、「ミニスキーマ」フォルダを展開し、ミニスキーマを右クリックして「プロパティ」を選択します。

2 **新しいミニスキーマの場合、「ミニスキーマ名」を入力します。**

ミニスキーマを変更する場合、「ミニスキーマ名」フィールドは編集できません。

3 **新しいミニスキーマの場合、ドロップダウン・リストから「接続」を選択してこのミニスキーマと関連付けます。**

「データベース」フィールドには、このデータ・ソース接続に関連付けられているデータベース名が表示されます。このフィールドは編集できません。

既存のミニスキーマの場合、「接続」フィールドは編集できません。

注： 物理ツリーの「データ・ソース」を右クリックしてこの手順を開始した場合は、データ・ソース接続を選択してこのミニスキーマと関連付ける必要があります。データ・ソース接続名を右クリックしてこの手順を開始した場合は、「接続」フィールドにその接続名が移入されます。

4 **オプション: 「説明」の入力または編集を行います。**

5 **次のいずれかのアクションを行います:**

- 新しいミニスキーマを作成する場合、「次へ」をクリックします。

  131 ページの「ミニスキーマでのテーブルの追加または削除」で説明されているように、ミニスキーマ・ウィザードの「テーブルの追加/削除」ページが表示されます。

- ミニスキーマを編集する場合、「テーブルの追加/削除」タブをクリックし、131 ページの「ミニスキーマでのテーブルの追加または削除」の説明に従います。

  また、ドラッグ・アンド・ドロップを使用して、グラフィカルな表示形式の既存のミニスキーマにテーブルを追加することもできます。138 ページの「ミニスキーマでのテーブルの追加」を参照してください。

ミニスキーマでのテーブルの追加または削除

130 ページの「ミニスキーマの全般プロパティの設定」の説明に従ってミニスキーマに名前および接続を設定したら、このトピックの操作を完了します。

注： ミニスキーマ作業領域内で右クリックして「テーブルの追加/削除」を選択することで、既存のミニスキーマから「追加/削除」ダイアログ・ボックスにアクセスすることもできます。

➤ ミニスキーマ・ダイアグラムのテーブルを追加または削除するには:

1 **「テーブルの追加/削除」ページで、「使用可能なテーブル」リストの内容を確認します。このリストには、選択したデータ・ソース接続からのすべてのテーブルが移入されています。**

2 **「使用可能なテーブル」から、ミニスキーマに含めるテーブルを選択し、「選択したテーブルをスキーマに追加」ボタンをクリックして「スキーマのテーブル」リストに移動します。**

すべての使用可能なテーブルを「スキーマのテーブル」リストに移動するには、「スキーマにすべてのテーブルを追加」ボタンをクリックします。

3 **オプション: 「スキーマのテーブル」リストに移動する前にテーブルにフィルタを適用する場合、「フィルタ」テキスト・ボックスにフィルタを入力します。**

4 **オプション: リレーショナル・ソースに基づくミニスキーマの場合、ミニスキーマに関連テーブルを追加するには、「スキーマのテーブル」リストからテーブルを選択し、「関連テーブルの追加」ボタンをクリックします。**

データ・ソース接続を作成したとき、スキーマの移入時にすべての使用可能なテーブルを選択していなかった場合は、関連するテーブルを追加できます(90 ページの「ミニスキーマの移入」を参照)。

5 **ミニスキーマからテーブルを削除するには、「スキーマのテーブル」リストから削除するテーブルを選択し、「選択したテーブルをスキーマから削除」ボタンをクリックします。これにより、テーブルが「使用可能なテーブル」リストに移動されます。**

スキーマのすべてのテーブルを「使用可能なテーブル」リストに移動するには、「スキーマからすべてのテーブルを削除」ボタンをクリックします。

ミニスキーマ・ダイアグラムからテーブルを削除しても、そのミニスキーマの基となるデータ・ソース接続からはテーブルが削除されません。

注: ユーザー定義結合を含むミニスキーマ・テーブルを削除する場合は必ず、その結合を最初に削除してから、ミニスキーマ・テーブルを削除することをお薦めします。

6 **「終了」をクリックします。**

7 **Essbase Studio コンソールのミニスキーマ作業領域で、新しいミニスキーマ・ダイアグラムを確認します。**

8 **オプション: Essbase Studio でミニスキーマのテーブルを自動的に配置するには、ツールバーの「スキーマのレイアウト」ボタンをクリックします。**

注: ミニスキーマ作業領域内を右クリックしてテーブルを自動的に配置することもできます。

ミニスキーマの操作

ミニスキーマを作成したら、次のタスクを実行できます。

- 133 ページの「ミニスキーマの保存」
- 133 ページの「ミニスキーマを開く」

- 134 ページの「ミニスキーマからの要素の削除」
- 134 ページの「ミニスキーマでの結合の追加または削除」
- 138 ページの「ミニスキーマでのテーブルの追加」
- 138 ページの「サンプル・データの表示」
- 139 ページの「ミニスキーマ・テーブルへの色の適用」
- 139 ページの「ミニスキーマ・オブジェクトからのメタデータ要素の作成」
- 140 ページの「ミニスキーマの表示」
- 144 ページの「ミニスキーマのプロパティの編集」
- 145 ページの「ミニスキーマ・リストのリフレッシュ」
- 145 ページの「ミニスキーマの削除」

ミニスキーマの保存

➤ ミニスキーマを保存するには、ミニスキーマ作業領域で右クリックして、「ミニスキーマの保存」を選択します。

または、「ファイル」>「保存」を選択するか、「[Ctrl]+[S]」を押します。

ミニスキーマを開く

➤ 編集するミニスキーマを開くには:

1 **「データ・ソース・ナビゲータ」で、適切なデータ・ソース接続を展開し、「ミニスキーマ」フォルダを展開します。**

2 **ツリーのミニスキーマ名を右クリックして、「編集」を選択します。**

ミニスキーマ・テーブルおよび列のプロパティの表示

➤ ミニスキーマ・テーブルおよび列のプロパティを表示するには:

1 **ミニスキーマがまだ開いていない場合は開きます(133 ページの「ミニスキーマを開く」を参照)。**

2 **ミニスキーマ作業領域で、プロパティを表示するテーブル・オブジェクトを右クリックして、「プロパティ」を選択します。**

テーブルのプロパティが、「プロパティ」ダイアログ・ボックスの「全般」タブに表示されます。

3 **「列」タブをクリックして、次の列プロパティを表示します:**

- 列がキー列かどうか
- 列名
- 列タイプ

- NULL が許可されているかどうか

ミニスキーマからの要素の削除

ミニスキーマからテーブルまたは結合を削除して、ミニスキーマを変更できます。

131 ページの「ミニスキーマでのテーブルの追加または削除」で説明するダイアログ・ボックス・ベースの手順に従って、ミニスキーマからテーブルを削除できます。また、次に説明するプロセスを使用して、テーブルまたは選択した結合を削除することもできます。

注： ユーザー定義結合を含むミニスキーマ・テーブルを削除する場合は必ず、その結合を最初に削除してから、ミニスキーマ・テーブルを削除することをお薦めします。

➤ ミニスキーマからテーブルまたは結合を削除するには:

1. **ミニスキーマがまだ開いていない場合は開きます(133 ページの「ミニスキーマを開く」を参照)。**
2. **オプション: ミニスキーマ作業領域で、削除するテーブル・オブジェクトを右クリックし、「削除」を選択します。**

   削除するすべてのテーブルに対して、この手順を繰り返します。
3. **オプション: 結合を削除するには、ミニスキーマ作業領域で結合を右クリックし、「削除」を選択します。**

   削除するすべての結合に対して、この手順を繰り返します。
4. **ミニスキーマを保存するには、ミニスキーマ作業領域で右クリックし、「ミニスキーマの保存」を選択します。**

ミニスキーマでの結合の追加または削除

ミニスキーマ・テーブルに結合を手動で追加したり、Essbase Studio でデータ・ソースを検査し、可能な論理結合を判断したりできます。

- 結合を手動で追加するには、134 ページの「手動での結合の追加または編集」を参照してください。
- 検査により結合を追加するには、137 ページの「検査による結合の追加」を参照してください。

手動での結合の追加または編集

Essbase Studio では同一のダイアログ・ボックスを使用して、結合を手動で作成および編集します。

「ミニスキーマ結合のプロパティを編集」ダイアログ・ボックスで、次の結合操作を実行できます:

- リレーショナル・テーブル間の結合の追加または編集

- 単一のリレーショナル・テーブルまたはフラット・ファイル内の自己結合の追加または編集

結合ペアで使用されている列を変更したり、結合ペアをさらに追加したり、その他の条件を変更したりして、結合を編集できます。

➤ 結合を追加または編集するには:

1 **ミニスキーマがまだ開いていない場合は開きます(133 ページの「ミニスキーマを開く」を参照)。**

2 **次のタスクを実行します:**

- 結合を追加するには、ミニスキーマ内の結合の追加先テーブルを選択して右クリックし、次に「結合の追加」を選択します。

  選択したテーブルが、「ミニスキーマ結合のプロパティを編集」ダイアログ・ボックスの左の「テーブル」ドロップダウン・リストに表示されます。

  注: Oracle BI EE とテキスト・ファイルのデータ・ソースから作成されたミニスキーマでは自己結合のみを作成できるため、選択したテーブルは左右両方の「テーブル」テキスト・ボックスに表示され、両方とも編集できません。

  または、ミニスキーマの背景を右リックして「結合の追加」を選択できます。ただし、この場合、どちらかの「テーブル」ドロップダウン・リストにはテーブルの名前が表示されません。結合するテーブルを選択します。

  注: Oracle BI EE またはテキスト・ファイルのデータ・ソースから作成されたミニスキーマ内の自己結合の場合は、左の「テーブル」ドロップダウン・リスト・ボックスのみがアクティブになります。左の「テーブル」ドロップダウン・リスト・ボックスでテーブルを選択すると、右の「テーブル」テキスト・ボックスにもそのテーブル名が表示されます。

- 結合を編集するには、ミニスキーマ内の結合線をダブルクリックして、「ミニスキーマ結合のプロパティを編集」ダイアログ・ボックスを表示します。

  注: 自己結合の場合、左右の「テーブル」テキスト・ボックスは選択できません。

3 **右の「テーブル」ドロップダウン・リストで、結合の確立先のテーブルを選択します。**

注: Oracle BI EE またはテキスト・ファイルのデータ・ソースから構築されたミニスキーマの場合は、自己結合のみを実行できます。そのため、「テーブル」ドロップダウン・リスト・ボックスは選択できません。

4 **結合を追加する場合は、「+」ボタンをクリックします。**

注： 結合を編集するか、最初の結合を追加する場合は、この手順をスキップして、手順 5.1 に進みます。

または、「列」グリッドから結合ペアを選択し、赤い「X」ボタンをクリックして、リストから結合を削除できます。

5 **たとえばリレーショナル・データ・ソースからの結合を操作するには:**

1. 「列」グリッドで、最初の空白行の左のセルをマウスでポイントし、クリックしてそのテーブルに関連付けられている列名のリストを表示します。

   または、既存の結合を編集する場合、編集する結合ペアの左のセルをクリックして、列名のリストを表示します。
2. 結合を開始する列を選択します。
3. 「列」グリッドで、グリッドの同じ行の右のセルをマウスでポイントし、クリックしてそのテーブルに関連付けられている列名のリストを表示します。

   または、既存の結合を編集する場合、編集する結合ペアの右のセルをクリックして、列名のリストを表示します。
4. 結合を完了する列を選択します。
5. **オプション:** 外部結合の場合、「外部」チェック・ボックスを選択してから、オプションを選択します:
   - 「左」 - 左側のテーブルからすべての行を戻し、右側のテーブルから一致する行を戻します。
   - 「完全」 - 左右の両方のテーブルから、すべての一致する行および一致しない行を戻します。
   - 「右」 - 右側のテーブルからすべての行を戻し、左側のテーブルから一致する行を戻します。

   「外部」チェック・ボックスを選択しない場合、デフォルトで内部結合が作成されます。内部結合がある場合、両方のテーブルから一致する行のみが戻されます。
6. **オプション:** 作成する結合の組合せごとに、手順 2 から手順 5.5 までを繰り返します。
7. **オプション:** 現在選択されているテーブルのペアに追加する結合ごとに、手順 4 から手順 5.5 までを繰り返します。

6 **たとえば、リレーショナル、Oracle BI EE またはテキスト・ファイルのデータ・ソースからの自己結合を処理するには:**

1. 左の「列」ドロップダウン・リストで、結合を開始する列名を選択します。
2. 右の「列」グリッドで、結合を完了する列名を選択します。
3. **オプション:** 外部結合の場合、「外部」チェック・ボックスを選択して、次のいずれかのオプションを選択します:
   - 左

- 完全
- 右

結合オプションについては、手順 5.5 で説明しています。

4. **オプション:** 作成する新しい自己結合ペアごとに、手順 2 から手順 6.3 までを繰り返します。
5. **オプション:** 現在選択されているテーブルに追加する自己結合ごとに、手順 5.1 から手順 6.3 までを繰り返します。

7 結合の追加および編集を保存するには、ミニスキーマ作業領域で右クリックして、「ミニスキーマの保存」を選択します。

注： マウスを使用して、ミニスキーマの 2 つのテーブルの列間に結合線を描いても結合を追加できます。次に、結合線をダブルクリックして「ミニスキーマ結合のプロパティを編集」ダイアログ・ボックスを開くと、結合のプロパティを編集できます。

検査による結合の追加

Essbase Studio でデータベースを検査し、可能な結合ペアのリストが戻されるようにすることができます。

➤ 検査による結合を追加するには:

1 ミニスキーマがまだ開いていない場合は開きます(133 ページの「ミニスキーマを開く」を参照)。

2 ミニスキーマの背景を右クリックし、「検査により結合を追加」を選択します。

「検査による結合の作成」ダイアログ・ボックスが表示され、Essbase Studio によって検出された可能な結合ペアがリストされます。

3 オプション: テーブルまたは列で結果をフィルタするには、次のタスクを実行します:

- フィルタ対象のテーブルの名前を「テーブル名フィルタ」に入力します。

  テーブル名では大文字と小文字が区別されます。

  入力したテーブル名が結合ペアの「結合ソース」側または「結合ターゲット」側のいずれかに含まれている場合は、そのペアが表示されます。
- フィルタ対象の列の名前を「列名フィルタ」に入力します。

  列名では大文字と小文字が区別されます。

  入力した列名が結合ペアの「結合ソース」側または「結合ターゲット」側のいずれかに含まれている場合は、そのペアが表示されます。

4 ミニスキーマに追加する結合ペアを選択するタスクを実行します:

- 結合ペアの結果をすべて選択するには、「すべてのアイテムを選択します」チェック・ボックスを選択します。
- 結合ペアの結果を選択するには、結果グリッドで結合ペアの横にあるチェック・ボックスを選択します。

5 「OK」をクリックします。

6 選択した結合ペアがミニスキーマ・ダイアグラムに追加されたことを確認します。

7 変更内容を保存するには、ミニスキーマ作業領域内で右クリックし、「ミニスキーマの保存」を選択します。

ミニスキーマでのテーブルの追加

物理ソースからテーブルをミニスキーマに追加することでミニスキーマを変更できます。

131 ページの「ミニスキーマでのテーブルの追加または削除」のダイアログ・ボックス・ベースの手順に従うか、次のプロセスを使用できます。

➤ テーブルをミニスキーマに追加するには:

1 ミニスキーマがまだ開いていない場合は開きます(133 ページの「ミニスキーマを開く」を参照)。

2 「データ・ソース・ナビゲータ」で、データ・ソース接続名の下の追加するテーブルを選択して、ミニスキーマ作業領域にドラッグします。

結合が物理データ・ソースに存在する場合、結合は自動的に作成されます。

注: 同じ物理テーブルを 2 回以上ミニスキーマに追加することはできません。

3 ミニスキーマに追加するすべての物理テーブルについて、手順 2 を繰り返します。

4 変更内容を保存するには、ミニスキーマ作業領域内で右クリックし、「ミニスキーマの保存」を選択します。

サンプル・データの表示

ミニスキーマ内または個別の列から、テーブルの使用可能なサンプル・データを表示できます。

➤ ミニスキーマでサンプル・データを表示するには:

1 ミニスキーマがまだ開いていない場合は開きます(133 ページの「ミニスキーマを開く」を参照)。

2 次のタスクのいずれか、または両方を実行します:

- ミニスキーマ内のテーブルのデータを表示するには、ミニスキーマ・ダイアグラムでそのテーブルを右クリックして、「サンプル・データの表示」を選択します。
- ミニスキーマ内のテーブルから単一列のデータを表示するには、ミニスキーマ・ダイアグラムで列を右クリックして、「サンプル・データの表示」を選択します。

要求したサンプル・データが、コンソールの作業領域の新しいタブに表示されます。

注： サンプル・データの表示ウィンドウが開いている間にソース・データが追加、編集または削除されても、開いているウィンドウ内のデータは自動的にはリフレッシュされません。変更したデータを表示するには、ウィンドウを閉じ、テーブルまたは列を再選択して、「サンプル・データの表示」コマンドを再発行する必要があります。

3 サンプル・データのウィンドウを閉じるには、ウィンドウのタブにある「X」をクリックします。

ミニスキーマ・テーブルへの色の適用

色を使用して、ミニスキーマの様々なテーブルを示すことができます。たとえば、複数のファクト・テーブルを含む大きなミニスキーマがあるとします。これらのファクト・テーブルをミニスキーマ作業領域で見つけやすいように、赤や他の色に設定できます。

さらに、ミニスキーマで事前に設定された色を変更することもできます。

ミニスキーマ・テーブルに色を適用すると、その色はテーブル要素の最上位のタブ部分に適用されます。ミニスキーマの色の例は、図 1 を参照してください。

➤ ミニスキーマ・テーブルに色を適用するには:

1 ミニスキーマがまだ開いていない場合は開きます(133 ページの「ミニスキーマを開く」を参照)。

2 色を追加するテーブル要素を選択します。

複数のテーブルを選択するには、「[Ctrl]」キーを押したまま、色を追加するテーブルをクリックします。

注： まだ色が追加されていなテーブル(薄いグレーで表示)、またはすでに色が適用されているテーブルを選択できます。

3 右クリックし、「色」を選択して、使用できる色を表示します。

4 リストから色を選択します。

5 ミニスキーマ作業領域でミニスキーマ・ダイアグラムに対する効果を表示します。

6 変更を保存するには、ミニスキーマ・ビューを右クリックし、「ミニスキーマの保存」を選択します。

ミニスキーマ・オブジェクトからのメタデータ要素の作成

メタデータ要素をミニスキーマのオブジェクトからすばやく簡単に作成して、データ・ソースまたはソースの特定要素の代替ビューを作成できます。

➤ メタデータ要素をミニスキーマ・オブジェクトから作成するには:

1 ミニスキーマがまだ開いていない場合は開きます(133 ページの「ミニスキーマを開く」を参照)。

2 メタデータ要素の作成元のミニスキーマからオブジェクトを選択します。例:

- メタデータ要素をミニスキーマの 1 つのテーブルから作成するには、テーブルを選択して右クリックし、次に「メタデータ・ナビゲータに追加」を選択します。
- メタデータ要素をミニスキーマの複数のテーブルから作成するには、「[Shift]」キーを押したままテーブルをクリックして選択し、右クリックしてから「メタデータ・ナビゲータに追加」を選択します。
- メタデータ要素をミニスキーマ・テーブルの 1 つの列から作成するには、列を選択して右クリックし、次に「メタデータ・ナビゲータに追加」を選択します。
- メタデータ要素を 1 つ以上のミニスキーマ・テーブルの複数の列から作成するには、「[Ctrl]」キーを押したままミニスキーマ・テーブルの列をクリックして選択し、右クリックしてから「メタデータ・ナビゲータに追加」を選択します。

前述のリストのアクションを実行すると、「フォルダの選択」ダイアログ・ボックスが表示されます。

3 **「フォルダの選択」で、メタデータ要素を保管するフォルダに移動してから「OK」をクリックします。**

注： この時点で、メタデータ要素を保管する新しいフォルダを作成できます。「フォルダの選択」ダイアログ・ボックスで「フォルダの作成」ボタンをクリックし、ポップアップ・ダイアログ・ボックスで「新規フォルダ名」を指定します。「OK」をクリックしたら、「フォルダの選択」ダイアログ・ボックスで新規フォルダに移動してから「OK」をクリックします。

4 **指定したミニスキーマ要素が「メタデータ・ナビゲータ」に追加されていることを確認します。**

5 **作成するすべてのメタデータ要素について、手順 2 と手順 4 を繰り返します。**

注： ドラッグ・アンド・ドロップを使用して、メタデータ要素を作成することもできます。手順 2 の説明に従ってミニスキーマから要素を選択しますが、右クリックはしません。かわりに、選択した要素を「メタデータ・ナビゲータ」の適切なフォルダにドラッグします。この方法を使用する場合は、プロセス中にフォルダを作成するオプションはありません。

ミニスキーマの表示

一部のミニスキーマ・ダイアグラムは、非常に大きくなることがあるため、Essbase Studio コンソールでは、ミニスキーマ・ダイアグラムを移動するツールとコマンドが提供されています。

- サムネイル・サイズのビューで、フォーカスするミニスキーマの場所に移動するには、サムネイル・ビューアを使用します。

141 ページの「ミニスキーマ作業領域の使用」を参照してください。

- 最大化するとミニスキーマ作業領域が画面全体に広がります。サムネイル・ビューアを併用すると、フォーカスするミニスキーマ作業領域の場所を簡単に指定できます。

  143 ページの「ミニスキーマ作業領域の最大化と最小化」を参照してください。
- 作業領域のミニスキーマ要素のサイズを拡大または縮小するには、ズーム・コマンドを使用します。

  143 ページの「ミニスキーマ作業領域のズーム・インおよびズーム・アウト」を参照してください。

ミニスキーマ作業領域の使用

サムネイル・ビューアを使用して、ミニスキーマ作業領域でフォーカスする領域を指定します。

注： サムネイル・ビューアは、作業領域のミニスキーマを最小化している場合でも最大化している場合でも使用できます(143 ページの「ミニスキーマ作業領域の最大化と最小化」を参照)。

➤ サムネイル・ビューアを使用するには:

1 **ミニスキーマがまだ開いていない場合は開きます(133 ページの「ミニスキーマを開く」を参照)。**

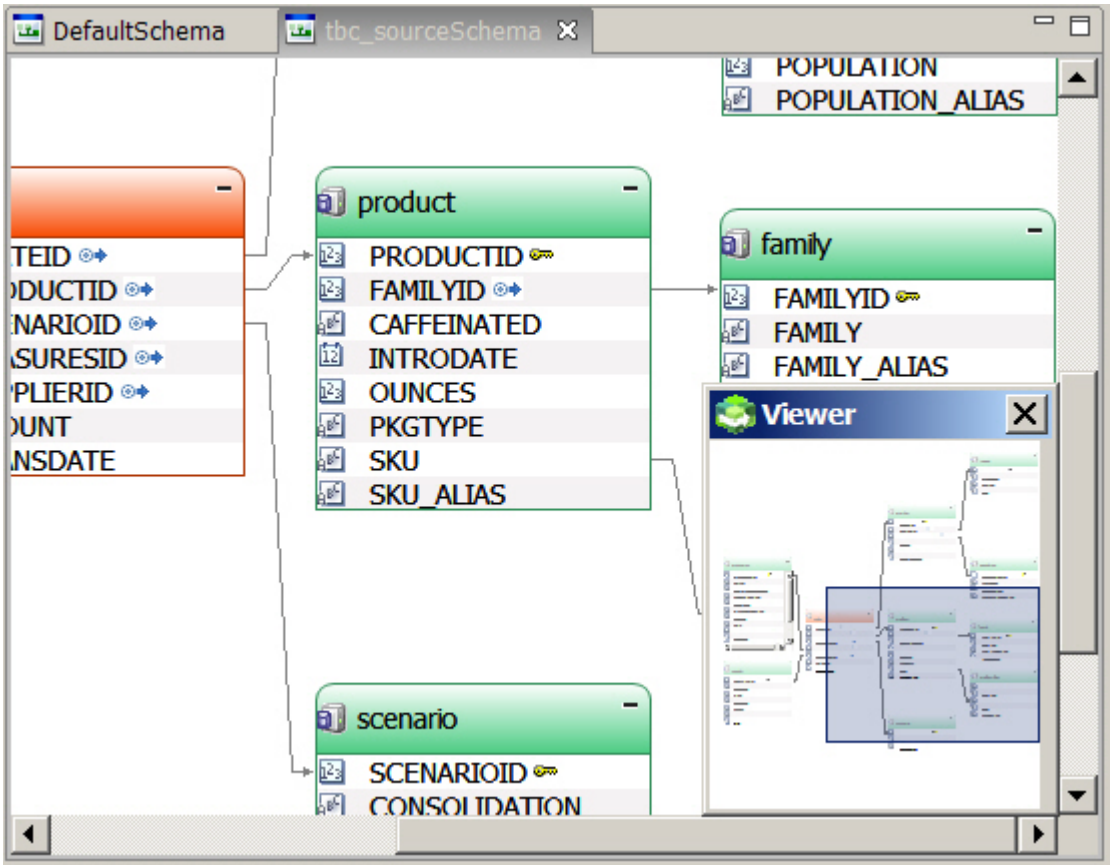
2 **サムネイル・ビューアを起動するには、ミニスキーマ作業領域の右下にあるビューア・アイコン をクリックします。**

サムネイル・ビューアがミニスキーマ作業領域の右下隅にポップアップされます。サムネイル・ビューアには、メイン・ミニスキーマ作業領域に表示されているミニスキーマと同じ部分が縮小フォーマットで表示されます。小さい方の透明な青いポインタがミニスキーマの部分を覆います。

図 2　サムネイル・ビューア

3 **表示するミニスキーマ作業領域の場所の方向にポインタをドラッグします。**

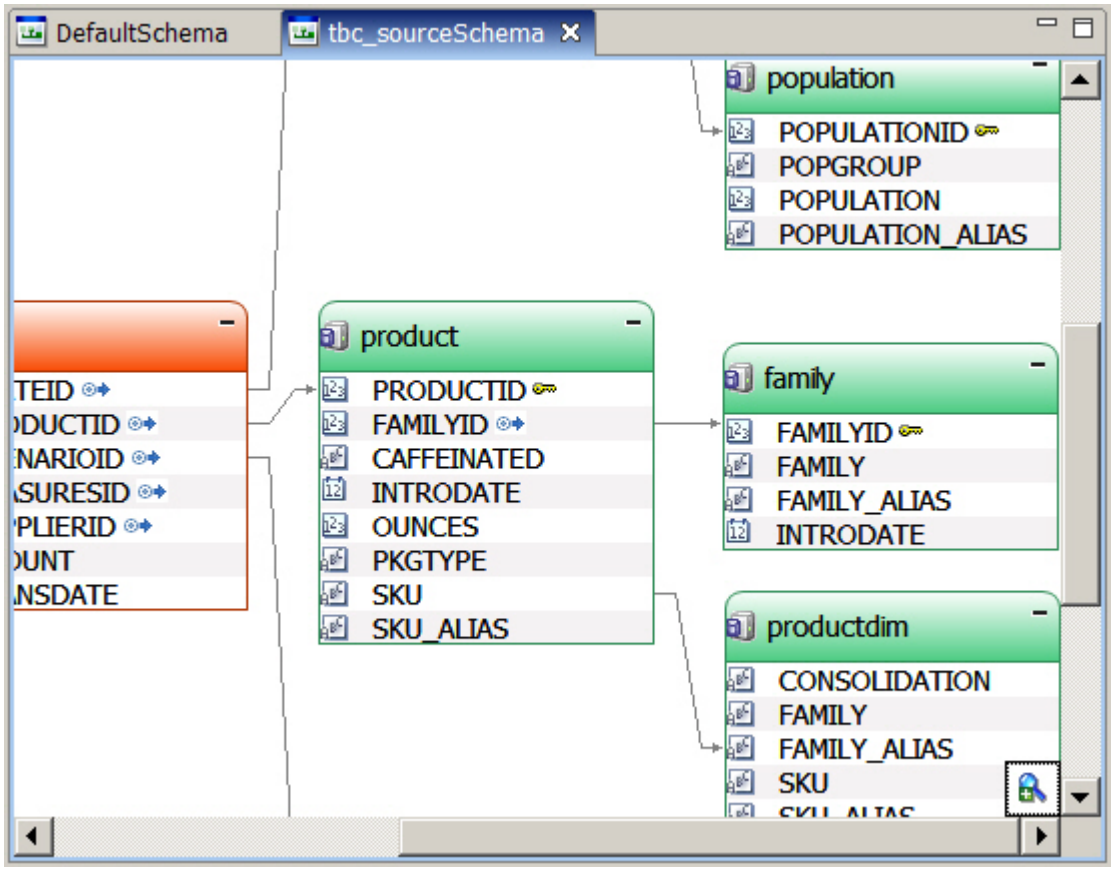
図 3　サムネイル・ビューアおよびミニスキーマ作業領域のフォーカス位置



4　サムネイル・ビューアを閉じるには、サムネイル・ビューアの右上隅にある「X」をクリックします。

ミニスキーマ作業領域のフォーカスは、サムネイル・ビューアで選択した部分に設定されています。

図 4　サムネイル・ビューアを閉じた後のミニスキーマ作業領域のフォーカス位置



ミニスキーマ作業領域の最大化と最小化

Essbase Studio コンソールのデフォルト・レイアウトでは、ミニスキーマ作業領域は、「メタデータ・ナビゲータ」と「データ・ソース・ナビゲータ」の間に最小化された状態で表示されます。ミニスキーマ作業領域を拡大すると、大きなミニスキーマ・ダイアグラムで移動できます。

注：　最小化および最大化されたミニスキーマ作業領域の両方でサムネイル・ビューアを使用できます(141 ページの「ミニスキーマ作業領域の使用」を参照)。

➤ ミニスキーマ作業領域を最大化するには、ミニスキーマ作業領域の右上隅にある「最大化」ボタン をクリックします。

➤ ミニスキーマ作業領域を最小化するには、ミニスキーマ作業領域の右上隅にある「最小化」ボタン をクリックします。

ミニスキーマ作業領域のズーム・インおよびズーム・アウト

ズーム・インおよびズーム・アウトのコマンドを使用してミニスキーマ・オブジェクトの表示サイズを変更できます。オブジェクトのサイズを拡大するには、ズーム・インします。オブジェクトのサイズを縮小するには、ズーム・アウトします。

➤ ミニスキーマ内のオブジェクトを拡大するには、ミニスキーマ作業領域を右クリックして「ズーム・イン」を選択します。

➤ ミニスキーマ内のオブジェクトを縮小するには、ミニスキーマ作業領域を右クリックして「ズーム・アウト」を選択します。

ズーム・インおよびズーム・アウト・コマンドは、必要な表示サイズになるまで何度でも繰り返すことができます。

ミニスキーマでのテーブルの配置

ミニスキーマ作業領域の任意の場所にミニスキーマ・テーブルを手動で配置することも、Essbase Studio によってテーブルを配置することもできます。

➤ 作業領域に表示されているミニスキーマ・テーブルを自動的に配置(レイアウト)するには、「スキーマのレイアウト」ボタン をクリックします。

注： また、ミニスキーマ作業領域を右クリックし、「スキーマのレイアウト」を選択することでも、テーブルを自動的に配置できます。

ミニスキーマのプロパティの編集

ミニスキーマのプロパティを編集する場合、ミニスキーマ・ウィザードで「プログラムの追加と削除」ダイアログ・ボックスの完了時に指定したプロパティを変更することになります。

次の編集タスクを実行できます:

- ミニスキーマでのテーブルの追加と削除
- フィルタの作成
- 関連テーブルの追加

➤ ミニスキーマ・プロパティを編集するには:

1 **オプション: ミニスキーマが開いていない場合は、ミニスキーマを開きます(133 ページの「ミニスキーマを開く」を参照)。**

2 **「データ・ソース・ナビゲータ」で、物理ツリーの「ミニスキーマ」を展開し、編集するミニスキーマを右クリックして「プロパティ」を選択します。**

3 **131 ページの「ミニスキーマでのテーブルの追加または削除」のミニスキーマ・プロパティの編集手順に従います。**

注： ミニスキーマ・プロパティの編集手順は、131 ページの「ミニスキーマでのテーブルの追加または削除」に記載されている手順と同じです。

ミニスキーマ・リストのリフレッシュ

➤ データ・ソース接続の「ミニスキーマ」フォルダ内のミニスキーマのリストをリフレッシュするには:

1 **「データ・ソース・ナビゲータ」で、データ・ソース接続を展開します。**

2 **「ミニスキーマ」フォルダを右クリックして、「リフレッシュ」を選択します。**

ミニスキーマの削除

➤ ミニスキーマを削除するには:

1 **「データ・ソース・ナビゲータ」で、データ・ソース接続を展開し、「ミニスキーマ」フォルダを展開します。**

2 **削除するミニスキーマの名前を右クリックして、「削除」を選択します。**

ミニスキーマを削除しても、そのミニスキーマの基となるデータ・ソース接続は削除されません。

注: ユーザー定義結合が含まれるミニスキーマを削除する際は常に、その結合を最初に削除してから、ミニスキーマを削除することをお薦めします。

6 イントロスペクション

この章の内容

イントロスペクションの概要 147
イントロスペクション・ウィザードでのミニスキーマ・オプションの選択 148
イントロスペクション・ウィザードでのファクト・テーブルの選択..... 148
イントロスペクション・ウィザードでの次元テーブルの選択 149
イントロスペクション・ウィザードでの階層の選択..... 150

イントロスペクションの概要

イントロスペクションは、物理データ・ソースにメタデータ要素があるかどうかを検査する方法です。イントロスペクションを実行するとき、データ・ソースに存在する構造情報が検査され、ファクト・テーブル、次元テーブル、階層、別名および属性が検出されます。イントロスペクションから導出されたメタデータ要素は、キューブ・スキーマの作成と、オプションで Essbase モデルの作成に使用されます。

イントロスペクションは、データ・ソースの作成プロセス中に実行できます。または、既存のデータ・ソースに対してイントロスペクションを実行して、そのデータ・ソースをスクレーピングしてメタデータ要素を取得できます。

注： イントロスペクションは、Oracle BI EE またはテキスト・ファイルのデータ・ソースに対してはサポートされません。

イントロスペクションを実行すると、プロセスはデータ・ソースを分析し、次の候補を検出して提示します:

- ファクト・テーブル
- 次元テーブル
- 階層

イントロスペクション・プロセス中に、ミニスキーマを作成することもできます。

イントロスペクション・ウィザードに従ってプロセスを進めます。ウィザードでは、選択候補を表示し、保持するアイテムを選択できます。次の各トピックではこのプロセスについて説明しています:

- 148 ページの「イントロスペクション・ウィザードでのミニスキーマ・オプションの選択」

- 148 ページの「イントロスペクション・ウィザードでのファクト・テーブルの選択」
- 149 ページの「イントロスペクション・ウィザードでの次元テーブルの選択」
- 150 ページの「イントロスペクション・ウィザードでの階層の選択」

イントロスペクション・ウィザードでのミニスキーマ・オプションの選択

確認するデータ・ソースを選択してから、ミニスキーマ・オプションを選択することで、イントロスペクションを開始します。

➤ イントロスペクションを開始して、ミニスキーマ・オプションを選択するには:

1 **「データ・ソース・ナビゲータ」で、イントロスペクションの実行対象のデータ・ソースを右クリックし、「イントロスペクション」を選択します。**

2 **次のいずれかのオプションを選択します:**

- 「新規スキーマ・ダイアグラムを作成します」-このオプションを選択した場合は、新規ミニスキーマの名前を入力します。
- 「ミニスキーマ・ダイアグラムをスキップ」 - このオプションを選択した場合、ミニスキーマ・ダイアグラムを作成せずに、イントロスペクション・プロセスを続行します。

  注 : テキスト・ファイルのデータ・ソースにアクセスする場合、ミニスキーマを作成することはありますが、そのミニスキーマに結合は含まれません。

3 **必ず、「イントロスペクションを使用した階層の検出」を選択してください。**

4 **「次へ」をクリックします。ミニスキーマを作成しない場合は、「終了」をクリックします。**

「次へ」をクリックすると、96 ページの「リレーショナル・ソースのミニスキーマへの移入」で説明されているように、ウィザードの「ファクト・テーブルの選択」ページに移動します。

イントロスペクション・ウィザードでのファクト・テーブルの選択

イントロスペクション中にファクト・テーブルの候補が表示されます。これらの候補からファクト・テーブルを選択します。

➤ ファクト・テーブルを選択するには:

1 **「ファクト・テーブルの選択」でファクト・テーブルとして使用するオブジェクトを選択し、「次へ」をクリックします。**

「ファクト・テーブル」のリストには、データ・ソースの確認後に Essbase Studio がファクト・テーブルの候補として表示するテーブルが含まれています。

これらの選択肢から 1 つ以上のファクト・テーブルを受け入れるか、または矢印キーを使用してテーブルを削除し、「使用可能なテーブル」からファクト・テーブルとして使用する 1 つ以上のテーブルを選択します。

注： 「使用可能なテーブル」リストには、データ・ソースの作成時に選択された残りのテーブル、ビュー、別名テーブルおよびシノニムがすべて含まれています。

2 **オプション:「使用可能なテーブル」に表示されるテーブルを制限するには、「フィルタ」を入力して「適用」をクリックします。**

たとえば、データベース・テーブルに、業績を示す perf など、特定のビジネス領域を表す接頭辞が付いている場合、次のようにフィルタを入力すれば、その領域に関連するテーブルのみが戻されます:

```
perf*
```

フィルタは、選択したデータ・ソース内のすべてのテーブル、ビュー、別名またはシノニムに適用されます。

3 **選択が完了したら、「次へ」をクリックし、イントロスペクション・ウィザードの「次元テーブルの選択」ページを表示します。**

イントロスペクション・ウィザードでの次元テーブルの選択

イントロスペクション中に、選択する次元テーブルの候補が表示されます。

➤ 次元テーブルを選択するには:

1 **「次元テーブルの選択」で、次元テーブルとして指定するオブジェクトを選択します。**

「選択した次元テーブル」リストには、データ・ソースの確認後に Essbase Studio によって使用可能な次元テーブルとして表示されるオブジェクトがすでに含まれていることがあります。これらの選択を受け入れるか、矢印キーでこれらのオブジェクトを削除した後で、「使用可能なオブジェクト」からオブジェクトを選択し、次元テーブルとして指定できます。

2 **オプション:「使用可能なテーブル」に表示されるテーブルを制限するには、「フィルタ」を入力して「適用」をクリックします。**

たとえば、データベース・テーブルに、業績を示す perf など、特定のビジネス領域を表す接頭辞が付いている場合、次のようにフィルタを入力すれば、その領域に関連するテーブルのみが戻されます:

```
perf*
```

フィルタは、選択したデータ・ソース内のすべてのテーブル、ビュー、別名またはシノニムに適用されます。

3 **「次へ」をクリックしてウィザードの「階層の選択」ページを表示します。**

イントロスペクション・ウィザードでの階層の選択

➤ イントロスペクション中に、メタデータ要素として追加する階層を選択するには:

1 **「階層の選択」で、「階層」リストから使用する階層を選択します。**

データ・ソースの確認後、Essbase Studio が検出した階層が「階層」に表示されます。デフォルトですべての階層が選択されています。次のいずれかを実行できます:

- 検出されたすべての階層を受け入れます。
- メタデータ要素として追加しない階層を選択し、「選択したアイテムを削除」ボタンをクリックします。
- マルチチェーン階層内で階層を選択し、「選択したアイテムを削除」ボタンをクリックします。
- 階層内で個々のメンバーを選択し、「選択したアイテムを削除」ボタンをクリックします。
- 「新規階層を作成」ボタンを選択して、階層を構築します。その後、次の手順を実行します:

  1. 「階層の作成」ダイアログ・ボックスで、階層の「名前」を入力し、オプションで「説明」を入力します。
  2. **オプション:** 現在の階層がメジャー階層の場合、「メジャー階層として作成」チェック・ボックスを選択します。
  3. 「OK」をクリックして「イントロスペクション・ウィザード」の「階層の選択」ページに戻ります。
  4. 「階層」リストから新しい階層を選択し、「使用可能なテーブルおよび列」リストで階層に追加する最初の列に移動し、その列を選択して、「子として列を追加」ボタンをクリックします。

     追加した列が階層のメンバーになります。

     注: 階層に追加する最初の列は、子として追加する必要があります。
  5. 「子として列を追加」または「兄弟として列を追加」ボタンを使用して、階層に他の列を追加します。

     注: 必ず最初に階層名または階層の適切なメンバーを選択してから、子または兄弟メンバーとして追加する列を選択してください。

- 「使用可能なテーブルおよび列」でテーブルを選択し、「テーブルに階層を追加」をクリックして階層を構築します。その後、次の手順を実行します:

  1. 「階層」リストで新しい階層を選択します。

     階層名のフォーマットは次のようになります: `tablenameHierarchy`。

  2. 「使用可能なテーブルおよび列」リストで、階層に追加する最初の列に移動し、その列を選択してから「子として列を追加」ボタンをクリックします。

  3. 「子として列を追加」または「兄弟として列を追加」ボタンを使用して、階層に他の列を追加します。

     注: 必ず最初に階層名または階層の適切なメンバーを選択してから、子または兄弟メンバーとして追加する列を選択してください。

2 「終了」をクリックします。

ミニスキーマを作成した場合は、「ミニスキーマ」作業領域でミニスキーマを確認できます。

また、イントロスペクション・プロセスによって作成されたメタデータ要素も確認します。これらは「メタデータ・ナビゲータ」の適切なフォルダの下にリストされます。

7 メタデータ要素

この章の内容

メタデータ要素について 153
メタデータ要素の作成または編集 154
テキスト・リストの作成または編集 164
日付要素の作成 165
メタデータ・フォルダの作成または編集 165
メタデータ要素の操作 166

メタデータ要素について

メタデータ要素は、データ・ソースの物理オブジェクトから導出される論理オブジェクトです。メタデータ要素は、データ・ソース接続の作成時またはその後に作成できます。メタデータ要素は、Essbase Studio での使用を目的としてすでに保管およびカタログが作成されている他のメタデータから作成することもできます。目的に応じたいくつかのメタデータ要素のタイプがあります。これらのメタデータ要素は次のとおりです:

- 次元要素-次の機能を持つ物理ソース列の論理的表現です。
  - データ・ソース接続プロセス中、またはその後の任意のタイミングで作成されます。
  - テキスト、数値または日付ベースです。
  - 基となる物理列以外のバインディングを指定するために編集できます。
  - フィルタを追加するために編集できます。
  - ソート順を追加するために編集できます。
- 導出テキスト・メジャー-範囲として表される定義済ルールによって値が決まるテキスト・メジャーです。
- テキスト・リスト–事前にマッピングされた値と ID であり、Essbase メンバーに関連付けることができます。
- 日付要素-Essbase Studio によって既存の日付型のメタデータ要素から導出される詳細な日付型のメタデータ要素です。

作成するメタデータ要素は、階層、メジャー階層、カレンダ階層など、他のメタデータ要素を構築するために使用できます(第 9 章「階層」を参照)。

フォルダは、別のタイプのメタデータ要素です。フォルダは、「メタデータ・ナビゲータ」のアイテムの整理に役立ちます。165 ページの「メタデータ・フォルダの作成または編集」を参照してください。

メタデータ要素の作成または編集

メタデータ要素の作成手順は、次の項を参照してください:

- 154 ページの「次元要素および導出テキスト・メジャーの作成または編集」
- 162 ページの「導出テキスト・メジャーの作成または編集」
- 164 ページの「テキスト・リストの作成または編集」
- 165 ページの「日付要素の作成」
- 165 ページの「メタデータ・フォルダの作成または編集」

次元要素および導出テキスト・メジャーの作成または編集

次元要素または導出テキスト・メジャーの作成および編集のワークフローは次のとおりです:

- 次元要素:
  - 次元要素を作成します。

    154 ページの「次元要素の作成」を参照してください。
  - 「プロパティの編集」ダイアログ・ボックスの「メイン」タブで、式、フィルタ、バインディングおよびソート順のプロパティを定義または編集します。

    156 ページの「次元要素の一般プロパティの定義または編集」を参照してください。
- 導出テキスト・メジャーの場合、導出テキスト・メジャーを作成または導出テキスト・メジャーにアクセスし、次に、「プロパティの編集」ダイアログ・ボックスの「メイン」タブで、式および範囲値を定義します。

  162 ページの「導出テキスト・メジャーの作成または編集」を参照してください。
- 次元要素および導出テキスト・メジャーについては、すべての該当する別名セット・バインディングを編集します。

  163 ページの「指定した別名セットに対する別名セット・バインディングの作成または編集」を参照してください。

次元要素の作成

次の手順で説明するいくつかの方法で次元要素を作成できます:

- メタデータ・ナビゲータからの次元要素の作成

- 「データ・ソース・ナビゲータ」からの次元要素の作成
- ミニスキーマからの次元要素の作成

メタデータ・ナビゲータからの次元要素の作成

➤ 「メタデータ・ナビゲータ」から次元要素を作成するには:

1 **「メタデータ・ナビゲータ」で、次元要素を保管するフォルダを右クリックし、「新規」>「次元要素」を選択します。**

これにより、次元要素の「プロパティの編集」ダイアログ・ボックスが開きます。

2 **「名前」およびオプションの「説明」を入力します。**

3 **156 ページの「次元要素の一般プロパティの定義または編集」に進み、バインディング式や、オプションでフィルタおよびソート順を定義します。**

「データ・ソース・ナビゲータ」からの次元要素の作成

➤ 「データ・ソース・ナビゲータ」から次元要素を作成するには:

1 **「データ・ソース・ナビゲータ」で、次元要素の基となる物理要素に移動します。**

2 **要素を選択して「データ・ソース・ナビゲータ」からドラッグし、「メタデータ・ナビゲータ」で保管先のフォルダにドロップします。**

注 : 「データ・ソース・ナビゲータ」では、複数のテーブル要素または列要素を選択できます。

3 **オプション: 156 ページの「次元要素の一般プロパティの定義または編集」に進み、バインディング式の編集や、フィルタおよびソート順の定義を行います。**

ミニスキーマからの次元要素の作成

➤ ミニスキーマから次元要素を作成するには:

1 **「データ・ソース・ナビゲータ」で、データ・ソース接続、「ミニスキーマ」フォルダの順に展開し、作成する次元要素が含まれるミニスキーマを選択します。**

2 **ミニスキーマを右クリックして、「編集」を選択します。**

ミニスキーマをダブルクリックして開くこともできます。

3 **ミニスキーマ作業領域で、ダイアグラムの要素(テーブル全体や個々の列など)をドラッグし、「メタデータ・ナビゲータ」の保管先フォルダにドロップします。**

注 : また、ミニスキーマ・ダイアグラムからテーブルまたは列の要素を選択したり、右クリックして「メタデータ・ナビゲータに追加」を選択したりして、ミニスキーマから次元要素を作成することもできます。

4 **オプション: 156 ページの「次元要素の一般プロパティの定義または編集」に進み、バインディング式の編集や、フィルタおよびソート順の定義を行います。**

次元要素の一般プロパティの定義または編集

次元要素の一般プロパティを定義または編集する前に、154 ページの「メタデータ要素の作成または編集」で説明する方法を使用して、次元要素を作成します。

導出テキスト・メジャーは、162 ページの「導出テキスト・メジャーの作成または編集」で説明する方法を使用して作成および編集する必要があります。

注： キャプション・バインディング、キー・バインディング、フィルタおよびソートの要素は、すべて同じデータ・ソースから導出する必要があります。

➤ 次元要素の一般プロパティを定義するには:

1 **次のいずれかのアクションを行います:**

- 次元要素を作成するには、154 ページの「次元要素の作成」で説明する手順を完了します。
- 次元要素を編集するには、「メタデータ・ナビゲータ」で要素を見つけて右クリックし、「編集」を選択します。

  注： 編集する次元要素をダブルクリックして開くこともできます。

2 **次元要素の基となる式を作成します。**

157 ページの「次元要素の基となる式の作成」の手順に従います。

3 **キー・バインディング・オプションを選択します。**

159 ページの「次元要素のキー・バインディング・オプションの選択」の手順に従います。

4 **オプション: この次元要素のフィルタを作成するには、「式」ボックスの「ソース」、「関数」、「演算子」タブから、適切な要素を「フィルタ」ボックスにドラッグ・アンド・ドロップし、必要に応じてフィルタ基準を入力します。**

注： フィルタの作成手順は、次元の基となる式の作成に使用した手順と同じです。157 ページの「次元要素の基となる式の作成」を参照してください。

たとえば、すべてのダイエット製品で SKU のフォーマットが XXX-20 であることがわかっている場合、次のように substr() 関数を使用してフィルタを設定できます:

```
'substr'( connection : \'tbc_source'::'tbc.product'.'SKU', 5, 2 ) ==  “20”
```

注： フィルタ式の場合、テキスト・ファイルのデータ・ソースでサポートされるのは、部分文字列(substr)関数と>、>=、<、<=、|=、==の演算子のみです。

5 **オプション:** 次元の列のソート順を定義するには、「式」リストの「ソース」タブから、列を「列のソート」にドラッグ・アンド・ドロップし、「ソート順」列で下向きの矢印をクリックして「昇順」または「降順」オプションを選択します。

注： 「ソート順」列に下向きの矢印がない場合、「ソート順」で列の行をクリックし、「昇順」/「降順」オプションをアクティブにします。

6 **オプション:** 163 ページの「指定した別名セットに対する別名セット・バインディングの作成または編集」で説明する方法で、適用可能な別名セットと次元要素を関連付けます。

次元要素の基となる式の作成

➤ 式を作成するには:

1 **ディメンション要素を作成するか、編集用に開きます(まだ行っていない場合)。**

- 次元要素を作成するには、154 ページの「次元要素の作成」で説明する手順を完了します。
- 次元要素を編集するには、「メタデータ・ナビゲータ」で要素を見つけて右クリックし、「編集」を選択します。

注： 編集する次元要素をダブルクリックして開くこともできます。

2 **「プロパティの編集」ダイアログ・ボックスで、次元要素の基となる式を作成します。**

たとえば、次の式により、次元要素の基本が製品テーブルの SKU 列になります:

```
connection : \'tbc_source'::'tbc.product'.'SKU'
```

また、独自の式またはステートメントを入力(手入力)することも可能です。たとえば、Essbase アウトラインまたはこの要素のスプレッドシートで表示するメンバー名として使用する場合、`"Seasons"`という文字列を入力します。

独自の式またはステートメントを入力(手入力)して次元要素を定義または編集する場合、文字列および名前の表示に関するガイドラインに従う必要があります。159 ページの「ユーザー定義式の入力のルール」を参照してください。

注意:

- デフォルトでは、テキストベース、数値ベースまたは日付ベースであるかに関わらず、次元要素には、基となる物理要素と同じデータ型が保持されます。たとえば、TBC データベースの SALES テーブルの AMOUNT に基づいて次元要素を作成する場合、その次元要素のデータ型またはユニットは、数値になります。その要素のキー・バインディング式は、常に数値である必要があります。

  キー・バインディング式では異なるユニット・タイプのオペランドを使用できますが、式の結果は、次元要素のユニットと互換性のあるユニットの値を戻す必要があります。次元要素のユニットが文字列の場合、式の結果

を文字列に変換するために、プロパティ.toString を任意の非文字列式に適用できます。

たとえば、TBC データベースの FAMILY テーブルの FAMILY 列に基づいて次元要素を作成したとします。この次元要素のキャプションおよびキー・バインディング式は次のようになります:

```
connection : \'tbcSource'::'TBC.tbc.FAMILY'.'FAMILY'
```

.toString を次のように追加することで、キー・バインディング式を数値ベースの式に変更できます:

```
connection : \'tbcSource'::'TBC.tbc.FAMILY'.'FAMILYID'.toString
```

- 次元要素の基となる式は、CPL(Common Platform Language)で記述されます。この式は、言語で定義する構文に従ったオペランドと演算子のシーケンスです。各式では値が戻され、その値の型によって式の型が定義されます。付録 C「CPL リファレンス」を参照してください。
- キャプション・バインディングおよびキー・バインディングの式の場合、テキスト・ファイルのデータ・ソースでサポートされるのは、部分文字列(substr)関数と連結演算子(||)のみです。
- Essbase Studio では、メタデータ要素間のサイクル依存関係は確認されません。サイクル依存関係は、自身に戻ってくるメタデータ要素間の関係です。詳細は、365 ページの「サイクル依存関係のガイドライン」を参照してください。
- 独立次元の式の作成に関する制限は、371 ページの「独立次元のバインディングに関する制限」を参照してください。

次の手順をガイドラインとして使用して、式を作成します:

1. ダイアログ・ボックスの左下にある「ソース」タブを選択します。
2. メンバーを表示する次元を展開します。
3. メンバーを選択します。
4. 右向きの矢印を使用して、メンバーの接続文字列を「キャプション・バインディング」テキスト・ボックスに移動します。
5. 「関数」タブを選択します。
6. SQL レベルを展開して関数の型を表示します。
7. 関数の型を展開して関数を表示します。
8. 関数を選択します。
9. 右向きの矢印を使用して、関数文字列を「キャプション・バインディング」テキスト・ボックスに移動します。
10. 「演算子」タブを選択します。
11. 演算子の型を展開して演算子を表示します。
12. 演算子を選択します。

13. 右向きの矢印を使用して、演算子を「キャプション・バインディング」テキスト・ボックスに移動します。

ユーザー定義式の入力のルール

独自の式またはステートメントを入力(手入力)して次元要素を定義または編集する場合、名前と文字列の表記に関するガイドラインに従う必要があります。このトピックでは次の操作を説明します:

- 159 ページの「ユーザー定義式での名前の入力」。
- 159 ページの「ユーザー定義式での文字列の入力」

ユーザー定義式での名前の入力

独自の式またはステートメントを入力(手入力)して次元要素を定義または編集する場合、名前の表現に関する次のガイドラインに従う必要があります:

- 名前の前後に単一引用符を入力します。次に 2 つの例を示します:

  ab という名前の場合、'ab'と入力する必要があります。

  a''b という名前の場合、'a''b'と入力する必要があります。
- 名前の中の単一引用符を繰り返します。例:

  a'b という名前の場合、'a''b'と入力する必要があります。
- 名前の前後に単一引用符が入る場合、単一引用符を 2 個前後に入力します。例:

  'a'b'という名前の場合、'''a''b'''と入力する必要があります。

ユーザー定義式での文字列の入力

独自の式またはステートメントを入力(手入力)して、次元要素を定義または編集する場合、次のガイドライン従って文字を表現する必要があります:

- 文字列は二重引用符で囲む必要があります。次に 2 つの例を示します:

  文字列 ab は"ab"と入力する必要があります。

  文字列 a'b は"a'b"と入力する必要があります。
- 文字列内の二重引用符は繰り返す必要があります。例:

  名前 a"b は"a""b"と入力する必要があります。
- 文字列の前後にある二重引用符は 2 つの二重引用符で囲む必要があります。例:

  名前"a"b"は"""a""b"""と入力する必要があります。

次元要素のキー・バインディング・オプションの選択

➤ キー・バインディング・オプションを選択するには:

1 **ディメンション要素を作成するか、編集用に開きます(まだ行っていない場合)。**

- 次元要素を作成するには、154 ページの「次元要素の作成」で説明する手順を完了します。
- 次元要素を編集するには、「メタデータ・ナビゲータ」で要素を見つけて右クリックし、「編集」を選択します。

  注： 編集する次元要素をダブルクリックして開くこともできます。

2 **まだ実行していない場合は、157 ページの「次元要素の基となる式の作成」の手順を完了します。**

3 **この次元要素を使用するアウトラインのタイプに応じて、この次元要素の「キー・バインディング」オプションを次の中から選択します:**

- 「キャプションと同一」-これがデフォルトです。Essbase Studio では、157 ページの「次元要素の基となる式の作成」で入力する式を使用して、Essbase アウトラインまたはスプレッドシートにこの次元要素のメンバー名をロードする方法を示します。

  ヒント： この次元要素のインスタンスが一意の名前を持つ場合、およびこの特定の次元要素が一意のメンバー名を持つアウトラインに属している場合は、「キャプションと同一」バインディング・オプションを使用します。

  「キャプションと同一」を選択すると、「キャプション・バインディング」テキスト・ボックスの式が「キー・バインディング」テキスト・ボックスにコピーされます。
- 「詳細」-このオプションを選択してから、「キー・バインディング」テキスト・ボックスに式を入力します。この式により、このメンバーの構築方法が Essbase に通知されます。

  キャプション・バインディングに重複するメンバー名が含まれていることや、データ・ソースからの一意のメンバー・キーの構築方法がわかっている場合は、「拡張」バインディング・オプションを使用します。

  ヒント： この特定の次元要素が重複するメンバー名を含むアウトラインに属している場合は、「拡張」バインディング・オプションを使用します。

  注： テキスト(文字列)以外のデータ型を持つ次元要素の場合、「詳細」オプションは自動的に設定されます。

  「キー・バインディング」テキスト・ボックスで、このメタデータ要素の Essbase に渡される値ごとに一意の識別子を生成する式を指定します。ここで指定する式は、データ・ソースから 1 つ以上のメンバー・キー列に関連付けられている必要があります。Essbase Studio では、割り当てられたキーを使用して、この次元要素に関連付けられるメンバーを一意に識別します。

たとえば、データ・ソースの「市場」テーブルに、重複するメンバー名を含む「都市名」という列があることがわかっているとします。また、「都市 ID」という別の列は、「市場」テーブルのメンバー・キー列で、この列では各都市に一意の都市 ID が付いています。そのため、都市 ID はキー・バインディング式での使用に適したID です。または、連結演算子を使用して、「地域」というテーブルにある別の列の地域名に都市名列を接頭辞として追加し、キー・バインディング式を作成できます。

「式」ボックスの「ソース」、「関数」、「演算子」タブの中から、適切な要素を「キー・バインディング」テキスト・ボックスへドラッグ・アンド・ドロップします。

注： キャプション・バインディングおよびキー・バインディングの式の場合、テキスト・ファイルのデータ・ソースでサポートされるのは、部分文字列(substr)関数と連結演算子(||)のみです。

「拡張」を選択すると、「キー・バインディング」テキスト・ボックスの式が「キャプション・バインディング」テキスト・ボックスの式とは別の式になります。

- 「遅延」-このオプションを選択すると、Essbase Studio から自動的にキー・バインディングを生成するか、またはキー・バインディング式を後から手動で作成できます。

ヒント： キー・バインディング式の生成方法が不明な場合は、「遅延」オプションを使用します。Essbase Studio にこの式を生成させることができます。

Essbase Studio にバインディング式を生成させるか、階層の作成または編集時に式を作成します。

また、すでに階層で使用されている次元要素の編集に戻り、キー・バインディング・オプションを「遅延」に変更することもできます。こうすると、Essbase Studio で、次元要素が属するすべての階層でデフォルトのキーが自動的に生成されます。または、階層の編集プロセスで独自にキーを指定できます。

階層では、Essbase Studio によってキー・バインディングのフル・パスが生成されます。たとえば、次の階層の場合、各次元要素のキー・バインディングがデフォルトのキャプション・バインディングになります:

```
        FAMILY - connection : \'tbcSource'::'tbc.family'.'FAMILY'
SKU - connection : \'tbcSource'::'tbc.product'.'SKU'
```

SKU 次元要素のキー・バインディングが「遅延」に変更されると、Essbase Studio で生成される階層のキー・バインディングは、次の階層での位置に対する次元要素のフル・パスになります:

```
class : \'tbcSource'\'productdim'\'FAMILY'.'caption' || "_" || class :
```

```
\'tbcSource'\'product'\'SKU'.'caption'
```

「遅延」オプションは、一意のメンバー名と重複するメンバー名のどちらもサポートする Essbase アウトラインで使用できます。

「遅延」オプションを選択すると、「キー・バインディング」テキスト・ボックスは空白のままになります。

4 次元要素の作成または編集プロセスを完了するには、156 ページの「次元要素の一般プロパティの定義または編集」の 156 ページの手順 4 に戻ります。

導出テキスト・メジャーの作成または編集

導出テキスト・メジャーは、値が定義済のルールによって範囲として表現されるテキスト・メジャーです。たとえば、メジャー「売上高」に基づく「売上高業績インデックス」という導出テキスト・メジャーを、対応する売上高の値が該当する範囲に応じて「高」、「中」、「低」と表示するように定義できます。

導出テキスト・メジャーを作成または編集するには:

- この項の手順に従って、「次元要素のプロパティの編集」ダイアログ・ボックスの「メイン」タブで、式と範囲を定義します。
- 適用可能な別名セットを次元要素に関連付けます。

  163 ページの「指定した別名セットに対する別名セット・バインディングの作成または編集」を参照してください。

注: キューブ・スキーマで導出テキスト・メジャーを使用して Essbase モデルを構築する場合、そのモデルに対して「XOLAP モデル」オプションは使用できません。

➤ 導出テキスト・メジャーを作成するには:

1 **「メタデータ・ナビゲータ」で、導出テキスト・メジャーを保管するフォルダを右クリックし、「新規」>「導出テキスト・メジャー」を選択します。**

2 **「プロパティ」ダイアログ・ボックスで、この導出テキスト・メジャーの「名前」を入力し、オプションで「説明」を入力します。**

3 **導出テキスト・メジャーの基となる式を作成するには、適切な数値ソース列を「式」リストのテーブルと列のリストからドラッグし、「式」ボックスにドロップします。**

「関数」タブおよび「演算子」タブとそれに対応する要素は、導出テキスト・メジャーには使用できません。

注: テキスト・ファイルのデータ・ソース接続内の要素から導出テキスト・メジャーの式を作成することはできません。

Essbase Studio では、メタデータ要素間のサイクル依存関係は確認されません。サイクル依存関係は、自身に戻ってくるメタデータ要素間の関係です。詳細は、365 ページの「サイクル依存関係のガイドライン」を参照してください。

4 **「範囲」で、プラス記号(「+」)をクリックして、範囲および範囲に対応するテキスト値を入力します。**

プラス記号(「+」)をクリックして、次の行にそれぞれの新しい範囲を入力します。

この導出テキスト・メジャーが、後で Essbase キューブとして配置されるキューブ・スキーマで使用される場合は、「ラベル」列に入力したテキスト値がレポートに表示されます。

表 2 で示すように、正確な数値範囲の値を入力できます。または、表 3 で示すように範囲キーワードを使用することもできます。

次に、売上高に関する一連の範囲の例を示します:

表 2　正確な数値を使用した範囲

開始値	終了値	ラベル
0	25000	悪い
25001	75000	良好
75001	100000	優秀

表 3　範囲キーワードを使用した範囲

開始値	終了値	ラベル
below	25000	悪い
25000	75000	良好
75000	above	優秀

範囲キーワードを使用するには:

1. 「開始値」列で、セルをクリックしてドロップダウンをアクティブにして、リストから「<below>」を選択します。
2. 「終了値」列で、セルをクリックしてドロップダウンをアクティブにして、リストから「<above>」を選択します。

5 **オプション: 163 ページの「指定した別名セットに対する別名セット・バインディングの作成または編集」で説明する方法で、適用可能な別名セットと次元要素を関連付けます。**

指定した別名セットに対する別名セット・バインディングの作成または編集

「別名セット」ダイアログ・ボックスを使用して、別名セット・バインディングを作成できます。また、次元要素または導出テキスト・メジャーを作成または編集するときに、別名セット・バインディングを作成または編集することもできます。

注： 別名セット・バインディングの追加または削除は慎重に行ってください。このダイアログ・ボックスで追加または削除したバインディングは、選択した別名セットのバインディングに反映されます。

➤ 指定した別名セットに対する別名セット・バインディングを作成または編集するには:

1. **操作する要素をまだ開いていない場合は、次のアクションを実行します:**
   - 「メタデータ・ナビゲータ」で次元要素を作成するには、次元要素を保管するフォルダを右クリックし、「新規」>「次元要素」を選択します。
   - 次元要素を編集するには、「メタデータ・ナビゲータ」で要素を見つけて右クリックし、「編集」を選択します。

   「プロパティの編集」ダイアログ・ボックスが表示されます。
2. **「別名」タブを選択します。**
3. **別名バインディング式を作成または編集するには、「別名セットの編集」ボタンをクリックします。**
4. **「別名バインディング」ダイアログ・ボックスで、この要素にバインドする別名情報が含まれる列を選択し、「適用」をクリックします。**
5. **オプション:** 別名バインディングを削除するには、「別名セット」グリッドで行を選択し、「削除」をクリックします。
6. **作成または編集する別名バインディングごとに、手順 3 と手順 4 を繰り返します。**
7. **終了したら、「OK」をクリックします。**

テキスト・リストの作成または編集

テキスト・リストは、テキスト文字列を含む列を、それらの文字列の ID を含む列にマップするメタデータ要素です。テキスト・リストで使用する列は、データ・ソースから取得された列であり、通常はテキスト文字列と ID を含む特定のデータ・ソース・テーブルから取得されます。この構造体は、Essbase Studio でテキスト・リストを作成する前に設定します。

テキスト・リストは、テキスト・メジャーとともに使用されます。テキスト・メジャーを使用すると、Essbase で数値データのほかにテキスト・ベースのコンテンツも分析できるようになります。テキスト形式のコンテンツの保管と分析は、通常はセルを所定のリスト内のテキスト値のいずれかにする必要がある場合に役立ちます。

たとえば、ある製品が異なる 5 つの色で販売されている場合があります。色は、値が必ず 5 色のいずれかとなるテキスト・メジャーです。これらの色は、対応する数値 ID にマップされているテキスト文字列のセットです。これらのマッピングは、ユーザーが作成するテキスト・リスト要素に含まれます。

➤ テキスト・リストを作成または編集する前に、次の手順を実行します:

1. **テキスト・リストで使用することを予定している ID と値を含む列があるデータベース・テーブルが存在することを確認します。**

2 **テキスト・リストで使用することを予定しているデータベース列が、「メタデータ・ナビゲータ」に次元要素として存在することを確認します。**

➤ テキスト・リストを作成または編集するには:

1 **次のいずれかのアクションを行います:**

- 新しいテキスト・リストを作成するには、「メタデータ・ナビゲータ」でテキスト・リストを保管するフォルダを右クリックし、「新規」>「テキスト・リスト」を選択します。
- テキスト・リストを編集するには、「メタデータ・ナビゲータ」でテキスト・リストを見つけて右クリックし、「編集」を選択します。

2 **「プロパティ」ダイアログ・ボックスで、このテキスト・リストの「名前」を入力し、オプションで「説明」を入力します。**

3 **「バインディング」で、データ値を含む列をデータ・ソースから選択し、右矢印ボタンをクリックして「値バインディング」に移動します。**

4 **オプション:「バインディング」で、数値を含む列をデータ・ソースから選択し、右矢印ボタンをクリックして「ID バインディング」に移動します。**

5 **「OK」をクリックして、テキスト・リストを保存します。**

注 : これが新規テキスト・リストの場合は、「メタデータ・ナビゲータ」に表示されることを確認します。

日付要素の作成

➤ 日付要素を作成するには:

1 **「メタデータ・ナビゲータ」で、新しい日付要素の基となる論理日付列を指定します。**

選択した論理日付列は、物理データ・ソースの日付列を表している必要があります。そうしないと、「日付要素の作成」機能を使用できません。

2 **論理日付列を右クリックし、「日付要素の作成」を選択します。**

3 **「日付要素の作成」ダイアログ・ボックスで、作成する日付要素の横にあるチェック・ボックスを選択します。**

4 **「OK」をクリックします。**

「メタデータ・ナビゲータ」に、元の日付要素と同じテーブルに属するテキスト列として新しい日付要素が表示されます。

メタデータ・フォルダの作成または編集

データ・ソースの作成時にメタデータ要素を作成することを選択した場合、接続ウィザードには、これらの要素を格納するメタデータ・フォルダを作成するオプションがあります。ただし、メタデータ・フォルダはいつでも作成でき、これらのフォルダを使用して、次元要素や階層などのメタデータ要素だけでなく、キューブ・スキーマやドリルスルー・レポートなどの他のオブジェクトも格納できます。

メタデータ・フォルダは、「メタデータ・ナビゲータ」でアイテムを整理するのに役立つ場合があります。

➤ メタデータ・フォルダを作成するには:

1. 「メタデータ・ナビゲータで」、フォルダを作成する場所に移動し、「ファイル」>「新規」>「フォルダ」を選択します。
2. フォルダの「名前」を入力します。
3. **オプション:**「説明」を入力します。
4. 「適用」をクリックし、「OK」をクリックします。

➤ フォルダの名前または説明を編集するには:

1. フォルダ名を右クリックして、「編集」を選択します。
2. **オプション:**「名前」を更新します。
3. 「オプション:」「説明」を更新します。
4. 「適用」をクリックし、「OK」をクリックします。

メタデータ要素の操作

メタデータ要素に対して、次のタスクを実行できます:

- 166 ページの「メタデータ要素のコピー」
- 166 ページの「メタデータ要素の名前変更」
- 167 ページの「メタデータ要素の削除」
- 167 ページの「メタデータ要素のサンプル・データの表示」
- 167 ページの「系列の表示」

メタデータ要素のコピー

➤ メタデータ要素を別の場所にコピーするには:

1. 「メタデータ・ナビゲータ」で、コピーするメタデータ要素を見つけます。
2. 要素を右クリックして「コピー」を選択します。
3. ルートまたはフォルダなど、要素のコピー先に移動し、右クリックして「貼付け」を選択します。

メタデータ要素の名前変更

➤ メタデータ要素の名前を変更するには、「メタデータ・ナビゲータ」でアクションを実行します:

- 要素をダブルクリックして要素のプロパティのダイアログ・ボックスを開き、名前フィールドに新しい名前を入力します。

- 要素を右クリックして「名前変更」を選択し、新しい名前を入力します。

注： メタデータ要素の名前を変更する場合、付録 B「Essbase Studio の命名規則」」にある次のガイドラインに従う必要があります。

メタデータ要素の削除

➤ メタデータ要素を削除するには、「メタデータ・ナビゲータ」で要素名を右クリックして、「削除」を選択します。

他のメタデータ要素が依存しているメタデータ要素は削除できません。最初に依存しているメタデータ要素を削除する必要があります。たとえば、キューブ・スキーマで使用され、ドリルスルー交差として使用されている階層は、影響を受けるドリルスルー・レポート、Essbase モデル、キューブ・スキーマなどを削除するまで削除できません。

メタデータ要素を削除しようとしたときにエラー・メッセージ・ボックスが表示される場合は、「詳細」ボタンをクリックして、削除しようとしているメタデータ要素に依存しているアイテムを判断します。これは反復的なプロセスです。依存しているメタデータ要素がすべて削除されるまで、メタデータ要素の削除を試行するたびに、エラー・メッセージ・ボックスが表示されます。

注： 系列ビューアは、依存しているメタデータ要素を迅速に確認するのに役立ちます。「メタデータ・ナビゲータ」でメタデータ要素を右クリックし、系列の表示を選択します。

メタデータ要素のサンプル・データの表示

➤ 指定したメタデータ要素のサンプル・データを表示するには、「メタデータ・ナビゲータ」でその要素名を右クリックして「サンプル・データの表示」を選択します。

「メタデータ・ナビゲータ」のメタデータ要素からサンプル・データを表示する方法の詳細は、123 ページの「サンプル・データの表示」を参照してください。

系列の表示

➤ メタデータ要素の系列を表示するには、「メタデータ・ナビゲータ」でその要素名を右クリックして、「系列の表示」を選択します。

系列の詳細は、第 15 章「系列」を参照してください。

8 別名セット

この章の内容

別名セットについて 169
別名セットの操作 170

別名セットについて

別名を使用したオブジェクトの識別は、一般的なビジネス・プラクティスです。Essbase では、別名はメンバーまたは共有メンバーの代替名です。たとえば、Sample Basic データベースの製品次元のメンバーは、100 などの製品コードと、Cola などのより記述的な別名の両方によって識別されます。別名テーブルに保管されている別名によって、アウトラインやレポートを読みやすくすることができます。別名は、言語、地域またはわかりやすい名前でグループ化できます。

別名セットを使用すると、1 つのメンバーに対して複数の別名を指定できます。たとえば、次のように異なる種類のレポートに異なる別名を使用できます-ユーザーは 100-10 を Cola として馴染みがありますが、広告主およびエグゼクティブは The Best Cola の方が馴染みがある場合があります。Sample Basic データベースのうち、2 つの記述的な別名を持つ製品のリストを次に示します:

```
Product

Default

Long Names

100-10    Cola          The Best Cola
100-20    Diet Cola     Diet Cola with Honey
100-30    Caffeine Free   Cola All the Cola, none of the Caffeine
```

別名セットは、特定の名前付き別名セットをメンバー名にマッピングします。Essbase とは異なり、デフォルトの別名テーブルはありません。「メタデータ・ナビゲータ」で、任意のフォルダ(root フォルダを含む)内に別名セットを作成して保存します。ユーザーは、「メタデータ・ナビゲータ」で右クリック・メニューを使用して別名セットの編集、コピー、名前変更、削除およびエクスポートを行うことができます。次いで、キューブ配置プロセスの前に、次の方法で別名セットを Essbase モデルのメンバーに関連付けます:

- Essbase モデル・レベルで、すべてのメンバーに 1 つ以上の別名セットを適用する

- 次元およびメンバー・レベルで、1 つ以上の別名セットを適用する

詳細は、170 ページの「別名セットの操作」を参照してください。

別名セットの操作

「メタデータ・ナビゲータ」から、次のタスクを実行できます:

- 別名セットの作成-170 ページの「別名セットの作成」を参照してください。
- 別名セットでのバインディングの変更、追加または削除-172 ページの「別名セットの編集」を参照してください。
- 別名セットのコピー、名前変更、削除またはエクスポート - 175 ページの「別名セットの管理」を参照してください。

別名セットの概要は、169 ページの「別名セットについて」を参照してください。

別名セットの作成

別名セットは、作成する別名のバインディングを保持するコンテナです。

➤ 別名セットを作成するには:

1 **「メタデータ・ナビゲータ」で、別名セットを保管するフォルダを右クリックして「新規」を選択し、次に「別名セット」を選択します。**

「別名セット」ダイアログ・ボックスが表示されます。

2 **別名セットの「名前」およびオプションで「説明」を入力します。**

3 **次のいずれかのタスクを実行して、この別名セットに 1 つ以上のバインディングを作成します。**

- 170 ページの「手動による別名セットのバインディングの作成 」
- 171 ページの「検査による別名セットのバインディングの作成」

4 **別名セットにバインディングを追加した後、「OK」をクリックし、「メタデータ・ナビゲータ」でリストされた新しい別名セットを表示します。**

手動による別名セットのバインディングの作成

(170 ページの「別名セットの作成」の説明に従って)別名セットを作成すると、バインディングを別名セットに追加できます。バインディングでは、特定の別名セットの別名情報を保持しているデータ・ソースの列を指定します。

この手順では、手動でバインディングを作成します。また、検査により自動でバインディングを作成することもできます。手順は、171 ページの「検査による別名セットのバインディングの作成」を参照してください。

➤ 別名セットのバインディングを手動で作成するには:

1 **「メタデータ・ナビゲータ」で、バインディングを追加する別名セットを右クリックし、「編集」を選択して「別名セット」ダイアログ・ボックスを起動します(まだ起動していない場合)。**

注: バインディングを別名に追加する前に、170 ページの「別名セットの作成」の説明に従って、別名セットを作成しておく必要があります。すでに別名セットを作成している場合は、手順 2 に進みます。

2 **「バインディング」リスト・ボックスの右にある「バインディングを作成」ボタンをクリックして、「バインディングの作成」ダイアログ・ボックスを表示します。**

「バインディングの作成」ダイアログ・ボックスには、次のセクションがあります:

- 「バインディング・ソース」ツリー。関連付けられている別名のバインディングを含むメタデータ・ツリーのすべての項目が表示されます。
- 「バインディング式」テキスト・ボックス。バインディング式を入力します。

3 **「バインディング・ソース」ツリーで、別名のバインディングに関連付けるメタデータ要素を選択します。**

4 **「バインディング式」テキスト・ボックスの下にあるソース・ツリーで、バインディング式の生成に使用する列に移動します。**

5 **列名をダブルクリックし、バインディング式をテキスト列に追加します。**

「バインディング・ソース」ツリーで選択したメタデータ・オブジェクトのバインディング式を生成するには、「バインディング式」テキスト・ボックスの下にあるソース・ツリーを使用します。または、このテキスト・ボックスに式を入力することもできます。

6 **「OK」をクリックします。**

作成した式は、「別名セット」ダイアログ・ボックスの「バインディング」セクションに次のフォーマットで表示されます:

- 「次元要素」の下には、別名に関連付けるメタデータ要素が表示されます。
- 「バインディング」の下には、別名の基となるソース要素が表示されます。

7 **この別名セット内の変更するすべてのバインディングに対して手順 2 から手順 6 を繰り返します。**

8 **別名セットにすべてのバインディングを作成したら、「OK」をクリックして「別名セット」ダイアログ・ボックスを閉じます。**

検査による別名セットのバインディングの作成

(170 ページの「別名セットの作成」の説明に従って)別名セットを作成すると、バインディングを別名セットに追加できます。バインディングでは、別名セットの別名情報を保持しているデータ・ソースの列を指定します。

この手順では、検査によりバインディングを作成します。また、手動でバインディングを作成することもできます。170 ページの「手動による別名セットのバインディングの作成 」を参照してください。

➤ 検査により別名セットのバインディングを作成するには:

1 **「メタデータ・ナビゲータ」で、バインディングを追加する別名セットを右クリックし、「編集」を選択して「別名セット」ダイアログ・ボックスを起動します(まだ起動していない場合)。**

注 : バインディングを別名に追加する前に、170 ページの「別名セットの作成」の説明に従って、別名セットを作成しておく必要があります。すでに別名セットを作成している場合は、手順 2 に進みます。

2 **「バインディング」ボックスの右側で、「検査によるバインディングの作成」をクリックします。**

3 **「別名の作成」で「別名列のパターン」を探し、基本バインディングに使用する列名のすべてまたは一部を表す文字列を入力します。**

たとえば、データベースの各テーブルに<language>_alias という形式の列名が含まれている場合は、次の検索パターンを入力します:

```
*alias
```

4 **別名バインディングに対して検査するデータ・ソース接続を選択します。**

5 **「OK」をクリックします。**

Essbase Studio により、別名バインディングに対して選択したデータ・ソース接続が検査され、「別名セット」ダイアログ・ボックスの「バインディング」の下にリストされます。

6 **「別名セット」ダイアログ・ボックスに示された別名セット・バインディングを確認し、必要に応じて編集または削除します。**

7 **別名セットに作成したバインディングを確認する際、「OK」をクリックして「別名セット」ダイアログ・ボックスを閉じます。**

別名セットの編集

別名セットのバインディングに対して、次の編集機能を実行できます:

- 次元要素名、次元要素パスまたはバインディング文字列による昇順または降順のバインディングのソート-173 ページの「別名セットの次元要素およびバインディングのソート」を参照してください。
- セットの既存のバインディングの変更-173 ページの「別名セットのバインディングの変更」を参照してください。
- 別名セットへのバインディングの追加-174 ページの「別名セットへのバインディングの追加」を参照してください。
- 別名セットからのバインディングの削除-174 ページの「別名セットからのバインディングの削除」を参照してください。

別名セットの次元要素およびバインディングのソート

別名セットのバインディングを、次元要素名、次元要素パスまたはバインディング文字列によって昇順または降順でソートできます。ソートすることで、作業対象のバインディングを簡単に見つけられます。

➤ 次元要素およびバインディングをソートするには:

1 **まだ行っていない場合は、「メタデータ・ナビゲータ」で、ソートを適用する別名セットを右クリックし、「編集」を選択して「別名セット」ダイアログ・ボックスを起動します。**

2 **「バインディング」で、バインディングを選択し、右クリックします。**

3 **ポップアップ・メニューで、オプションを選択します:**

- 「次元要素名でソート」
- 「次元要素パスでソート」

  注: このオプションを使用するには、「次元要素のフルパスを表示」オプションを選択してこれを有効にします。有効になると、チェック・マークがこのオプションの横に表示されます。選択すると、フルパスと次元要素名が「次元要素」列に表示されます。

- 「バインディングでソート」
- 「昇順にソート」
- 「降順にソート」
- 「次元要素のフルパスの表示」

4 **オプション: 作業対象のバインディングを見つけたら、次のいずれかのタスクを実行できます:**

- 173 ページの「別名セットのバインディングの変更」
- 174 ページの「別名セットからのバインディングの削除」

別名セットのバインディングの変更

別名セットのバインディングを変更できます。

➤ バインディングを変更するには:

1 **「メタデータ・ナビゲータ」で、バインディングを変更する別名セットを右クリックし、「編集」を選択して「別名セット」ダイアログ・ボックスを起動します(まだ起動していない場合)。**

2 **「バインディング」で、変更するバインディングを選択します。**

3 **「バインディング」の右にある「選択したバインディングを編集」ボタンをクリックして、「バインディングの編集」ダイアログ・ボックスを表示します。**

一番上のペインに、現在のバインディング構文が表示されます。

4 **次のいずれかのアクションを行います:**

- ダイアログ・ボックスの一番上のペインに、バインディング式を入力します。構文は次のとおりです:

  ```
  connection :
  \'<logical_source_name'::'<physical_data_source_name>.<physical_table_name>'.'<physical_column_name>'
  ```

- テキスト・ボックスの一番下のペインにある物理ツリーで、バインディング式の生成に使用する物理データ・ソース内の列に移動します。列名をダブルクリックして、テキスト列にバインディング式としてこれを追加します。

  ヒント： この方法を使用する場合、始める前にダイアログ・ボックスの一番上のペインからテキストを消去することをお薦めします。

5 バインディングを変更したら、「OK」をクリックして「別名セット」ダイアログ・ボックスに戻ります。

6 この別名セット内の変更するすべてのバインディングに対して手順 2 から手順 5 を繰り返します。

別名セットへのバインディングの追加

別名セットに 1 つ以上のバインディングを追加できます。

➤ バインディングを追加するには:

1 「メタデータ・ナビゲータ」で、バインディングを追加する別名セットを右クリックし、「編集」を選択して「別名セット」ダイアログ・ボックスを起動します(まだ起動していない場合)。

2 選択した別名セットに追加するすべてのバインディングに対して、170 ページの「手動による別名セットのバインディングの作成 」の手順 2 から手順 6 までを実行します。

別名セットからのバインディングの削除

別名セットからバインディングを削除できます。

➤ バインディングを削除するには:

1 「メタデータ・ナビゲータ」で、バインディングを削除する別名セットを右クリックし、「編集」を選択して「別名セット」ダイアログ・ボックスを起動します(まだ起動していない場合)。

2 削除するバインディングを選択してから、「バインディング」リスト・ボックスの右にある「選択したバインディングを削除」ボタンをクリックします。

  注： リストから連続する複数のバインディングを選択するには、[Shift]キーを使用します。リストから連続していない複数のバインディングを選択するには、[Ctrl]キーを使用します。

3 「削除の確認」プロンプトが表示されたら「はい」をクリックします。

削除するために選択したバインディングが「バインディング」リストから消去されます。

注： 削除の確認プロンプトが表示されないようにするには、「今後、このメッセージを表示しません。」チェック・ボックスを選択します。

別名セットの管理

選択した別名セットに対して次の操作を実行できます:

- コピー - 175 ページの「別名セットのコピーまたは移動」を参照してください。
- 名前変更-175 ページの「別名セットの名前変更」を参照してください。
- 削除-176 ページの「別名セットの削除」を参照してください。
- エクスポート - 176 ページの「別名セットのエクスポート」を参照してください。

別名セットのコピーまたは移動

➤ 別名セットをコピーするには:

1 **「メタデータ・ナビゲータ」で、コピーする別名セットを右クリックして「コピー」を選択します。**

2 **「メタデータ・ナビゲータ」で宛先フォルダを右クリックし、ポップアップ・メニューから「貼付け」を選択します。**

「メタデータ・ナビゲータ」では別のフォルダにコピーする場合、別名セット名が保持されます。

同じフォルダにコピーする場合、別名セットの名前が自動的に変更されます。たとえば、skuAS という別名セットを同じフォルダにコピーすると、自動的に skuAS_copy という名前に変更されます。

3 **「メタデータ・ナビゲータ」で、コピーした別名セットを表示します。**

➤ 別名セットを別のフォルダに移動するには:

1 **「メタデータ・ナビゲータ」で、移動する別名セットを選択します。**

2 **別名セットを宛先フォルダにドラッグ・アンド・ドロップします。**

別名セットの名前変更

注： 別名セットの名前を変更する場合、付録 B「Essbase Studio の命名規則」のガイドラインに従う必要があります。

➤ 別名セットの名前を変更するには:

1 **「メタデータ・ナビゲータ」で、名前を変更する別名セットを右クリックして「名前変更」を選択します。**

2 **「メタデータ・ナビゲータ」で別名セット名がハイライトされた状態で、新しい名前を入力します。**

3 **「メタデータ・ナビゲータ」で、名前変更した別名セットを表示します。**

別名セットの削除

➤ 別名セットを削除するには:

1 **「メタデータ・ナビゲータ」で、削除する別名セットを右クリックして「削除」を選択します。**

注: 連続する複数の別名セットをツリーから削除するように選択するには、[Shift]キーを使用します。連続しない複数の別名セットをツリーから削除するように選択するには、[Ctrl]キーを使用します。

2 **「削除の確認」プロンプトが表示されたら「はい」をクリックします。**

削除するように選択した別名セットが、「メタデータ・ナビゲータ」から消去されます。

注: 削除を確認するメッセージが表示されないようにするには、「今後、このメッセージを表示しません。」チェック・ボックスを選択します。

別名セットのエクスポート

74 ページの「選択したカタログ要素のエクスポート」の説明に従って、選択した別名セットを XML ファイルにエクスポートできます。この XML ファイルを使用して、別名セットを別のマシンのカタログにインポートできます。

エクスポートおよびインポートに関するすべてのトピックは、72 ページの「Essbase Studio カタログ・データベースのエクスポートとインポート」を参照してください。

9 階層

この章の内容

階層について 177
メジャー階層について 178
標準階層およびメジャー階層の作成 180
カレンダ階層の作成 189
階層の編集 197

階層について

階層によってデータの集計方法および移動方法が決まります。たとえば、多くのビジネスでは、データを月ごとに要約したり、月ごとのデータをロールアップして四半期ごとの数値を取得したり、四半期ごとのデータをロールアップして、年次データを取得したりします。郵便番号、都市、州および国ごとにデータを要約することもあります。

Essbase Studio では、階層を使用して次元をモデリングします。任意の次元を使用して、レポート用のデータを集計できます。階層のレベルは、次のいずれかの方法で結合されている異なるソース・データベース・テーブルから取得できます。

- リレーショナル・データ・ソースの物理結合単位。
- Essbase Studio のミニスキーマで作成された結合単位。

たとえば、市場次元の階層は次のようになります。

```
  MARKET
REGION
  STATE
```

この階層では、地域は市場階層の子であり、州は地域の子になります。階層の構造は、個々の州の売上レポートを作成したり、州の数値を集計して地域売上レポートを作成したりできる Essbase アウトラインに継承されます。

キューブ・スキーマで使用される階層によって、表示される Essbase モデルの構造が決まります。Essbase モデルでは、次のものを含む名前付き構造が提供されます:

- 階層構造自体-データを集計するためのレベル単位のシーケンスです。

たとえば、州ごとの売上合計をロールアップすると、地域ごとの売上合計になります。階層内で都市が州の子である場合、都市ごとの売上合計を州ごとの売上合計にロールアップできます。

- 階層内の選択したメタデータ要素に配置されるデータ・フィルタ-特定のカテゴリに関する情報を選択する方法です。

  たとえば、REGION 列をフィルタして、USA の売上情報のみを含めるようにできます。

- データ順序の整理-列の順序をソートします。

  たとえば、MONTH 列を降順にソートして、最近の合計を最初に表示できます。

- 変換-Essbase のメンバー名になる列のデータ値やメジャーを制御する方法です。

  たとえば、重複するメンバー名をサポートしていないキューブで、階層内の都市が州の子である場合、STATE 列の適切な値を使用して、都市の各値に接頭辞を付けることでメンバー名を一意にできます。これにより、CA_ALBANY と NY_ALBANY を区別できるようになります。

階層を作成したら、「メタデータ・ナビゲータ」からキューブ・スキーマ・ウィザードに階層を直接ドラッグできます。さらに、階層は任意の数のキューブ・スキーマで再使用できます。異なる階層を作成することで、ユーザー・グループごとにキューブ・スキーマをカスタマイズできます。たとえば、階層で階層フィルタを使用して、特定の企業区分のユーザーを表示していると仮定します。このような階層の場合、ユーザーは、特定の区分に関連するデータのみを含むキューブ・スキーマをより簡単に作成できます。

Essbase Studio では、標準階層、メジャー階層およびカレンダ階層がサポートされています。メジャー階層の詳細は、178 ページの「メジャー階層について」を参照してください

階層の作成の詳細は、次の各項を参照してください:

- 180 ページの「標準階層およびメジャー階層の作成」
- 189 ページの「カレンダ階層の作成」

366 ページの「階層の使用方法のガイドラインと制限」を参照してください。

メジャー階層について

Essbase Studio サーバーはメジャー階層を使用して Essbase モデルに会計次元を作成します。データに複数のメジャー(データ値)が含まれている場合は、メジャー階層を定義してすべてのメジャーをその階層に追加します

Essbase Studio には、メジャー階層について次のガイドラインがあります。

メジャー階層はファクト・テーブルから作成することをお薦めします。メジャー階層にはユーザー定義メンバーが含まれることもあります。ただし、メジャー階層は次元テーブルから作成することもできます。これにもユーザー定義メンバーが含まれることがあります。

- ファクト・テーブルから構築されたメジャー階層から作成される会計次元は、“ファクトから構築された会計”と呼ばれます。

  キューブ・スキーマでは、ファクト・テーブルから構築されたメジャー階層を使用する場合、ファクト・テーブルの列ヘッダーが次の Essbase モデルの会計次元のメンバーになります。

  TBC サンプル・データベースを使用すると、ファクト・テーブル(SALESFACT)から構築されたメジャー階層の例は次のようになります:

```
     Profit    (user-defined member)
 SALES   (from SALESFACT)
 COGS    (from SALESFACT)
```

  ファクト・テーブルの列タイプがテキストの場合、Essbase Studio では、これをテキスト・メジャーとして処理し、自動的にテキスト・リストが作成されます。

- 次元テーブルから構築された階層から作成される会計次元は、1 つ以上の緩いメジャーと組み合せて、“ファクトから構築されていない会計”と呼ばれます。(緩いメジャーは階層に編成されていない個々のメジャーです。)

  この場合、次元テーブルのいずれかで指定された列にはファクトの列見出しを表す行が含まれています。この列は個別の次元テーブルの列に結合されており、これらのファクトの値を含んでいます。モデルが作成されたとき、会計次元(ファクトから構築されない)が作成されるのは、2 つの次元テーブルからです。

  TBC サンプル・データベースを使用すると、非ファクト・テーブル(MEASURES と SALES)から構築されたメジャー階層の例は次のようになります:

```
     PARENT    (from MEASURES)
 CHILD   (from MEASURES)
```

  MEASURES テーブルでは、CHILD 列の値は MEASURESID 列の値に対応しています。MEASURESID 列は SALES テーブルの MEASURESID 列に結合していて、SALES テーブルの AMOUNT 列の実際のデータ値を提供します。

  キューブ・スキーマで、“ファクトから構築されない会計”のメジャー階層を使用する場合、キューブ・スキーマ・ウィザードの「階層」フィールドに階層をドラッグします。「メジャー/メジャー階層」フィールドでは少なくとも 1 つ緩いメジャーもドラッグする必要があります。

  Essbase Studio では、“ファクトから構築されない会計”のメジャー階層のメンバーに緩いメジャーの値をマッピングします。

  ファクト・テーブルの列タイプがテキストの場合、Essbase Studio では、これをテキスト・メジャーとして処理し、自動的にテキスト・リストが作成されます。

つまり、メジャー階層をファクト・テーブルの列から作成することをお薦めしますが、Essbase Studio では、次元テーブルまたはユーザー定義メンバーからメジャー階層を作成できます。列ヘッダー・タイプがテキストの場合、Essbase Studio では、テキスト・リストが構築されます。

注： 1024KB を超えるテキスト・リストのために配置が中止されることがあります。

標準階層およびメジャー階層の作成

階層は、メンバー・レベルのグループまたはメンバーのグループの組織構造を定義します。製品や市場などのビジネス・オブジェクトの連結およびナビゲーションを定義する場合は、標準階層を使用します。

データ値を表す次元要素を持つ階層を作成するには、メジャー階層を使用します。データに複数のメジャー(データ値)が含まれている場合は、メジャー階層を定義してすべてのメジャーをその階層に追加します。

階層の列としてキー・バインディングを使用する手順は、182 ページの「階層での遅延キー・バインディング列の使用」を参照してください。

様々な階層タイプを184 ページの「階層の例」に示します。

カレンダ階層を作成するには、189 ページの「カレンダ階層の作成」を参照してください。

➤ 標準階層またはメジャー階層を作成するには:

1 **「メタデータ・ナビゲータ」で適切なフォルダを右クリックし、「新規」>「階層」または「新規」>「メジャー階層」を選択して階層エディタを起動します。**

2 **階層の「名前」を入力します。**

標準階層のデフォルトの名前は `NewHierarchy` で、メジャー階層のデフォルトの名前は `NewMeasureHierarchy` です。

3 **オプション:「説明」を入力します。**

4 **次の方法のいずれか、または方法を組合せて使用して、階層に要素を追加します。**

注： 同じ階層で、論理要素、物理要素およびユーザー定義の要素を組み合せることができます。「メタデータ・ナビゲータ」の論理要素と「データ・ソース・ナビゲータ」の物理要素を追加できます。また、ユーザー定義の親を階層に追加し、その子として論理要素または物理要素を追加することも可能です。

「メタデータ・ナビゲータ」から次の手順でドラッグ・アンド・ドロップします:

1. 「メタデータ・ナビゲータ」で、階層に追加する最初のメタデータ要素に移動します。
2. 追加するメタデータ要素を選択して「階層の編集」ダイアログ・ボックスにドラッグし、「データ」グループの「階層」列にドロップします。

3. 階層に追加する次のメタデータ要素を選択し、追加した前の要素に直接ドラッグ・アンド・ドロップします。

   メタデータ要素を前の要素の上に直接ドラッグ・アンド・ドロップすると、階層構造が作成されます。

   メタデータ要素を前の要素の下の行へドラッグ・アンド・ドロップすると、階層で新しいチェーンが開始されます。

   注： また、物理要素を「データ・ソース・ナビゲータ」からドラッグし、親要素の上または下にドロップすることも可能です。または、「追加」ボタンをクリックし、子として追加を選択して論理要素を親要素に追加するか、兄弟として追加を選択して階層で新しいチェーンを開始します。

4. **オプション:** 構築する階層に必要な回数だけ手順 4.3 を繰り返します。

「データ・ソース・ナビゲータ」からドラッグ・アンド・ドロップします:

1. 「データ・ソース・ナビゲータ」で、階層に追加する最初の物理要素に移動します。
2. 追加する物理要素を選択して「階層の編集」ダイアログ・ボックスにドラッグし、「データ」グループの「階層」列にドロップします。

   階層に追加する物理要素ごとに、対応するメタデータ要素を、同じ場所にある「メタデータ・ナビゲータ」に追加します。
3. 階層に追加する次の物理要素を選択し、追加した前の要素に直接ドラッグ・アンド・ドロップします。

   物理要素を前の要素の上に直接ドロップすると、階層構造が作成されます。

   物理要素を前の要素の下の行にドロップすると、階層で新しいチェーンが開始されます。

   注： また、論理要素を「メタデータ・ナビゲータ」からドラッグし、親要素の上または下にドロップすることも可能です。または、「追加」ボタンをクリックし、子として追加を選択して論理要素を親要素に追加するか、兄弟として追加を選択して階層で新しいチェーンを開始します。

4. **オプション:** 構築する階層に必要な回数だけ手順 4.3 を繰り返します。

「追加」ボタンを使用して、次の操作を実行します:

1. 「追加」ボタンをクリックし、アクションを実行します:
   - 「子の追加」または「兄弟の追加」を選択します。

     「エンティティの選択」ダイアログ・ボックスが表示されます。子または兄弟として追加する論理要素に移動して選択し、「OK」をクリックします。

     階層の最初の要素では、「子の追加」または「兄弟の追加」を選択できることに注意してください。
   - 「ユーザー定義の子の追加」または「ユーザー定義の兄弟の追加」を選択します。

「階層」で、グリッドの編集可能フィールドにユーザー定義メンバーの名前を入力します。デフォルトの名前は NewMemberN です。

階層の最初の要素では、「ユーザー定義の子の追加」または「ユーザー定義の兄弟の追加」を選択できます。

注： ユーザー定義の要素は、列ベース(リレーショナル)要素の子として追加できません。

注： 「キー・バインディング」列には「未指定」というテキストが表示されます。階層を保存すると、「キー・バインディング」列が更新され、子要素または兄弟要素の名前が表示されます。

2. 「追加」ボタンをもう一度クリックして、手順 4.1 の説明に従って別の選択を行います。

注： 「子の追加」または「兄弟の追加」オプションを選択して階層の最上位レベルを作成すると、「ユーザー定義の子の追加」オプションは使用できません。有効な選択肢は「子の追加」、「兄弟の追加」または「ユーザー定義の兄弟の追加」です。

3. **オプション:** 構築する階層に必要な回数だけ手順 4.2 を繰り返します。

5 「保存」をクリックします。

オプションで201 ページの「階層のプレビュー」の説明に従い、「プレビュー」をクリックして「サンプル・データ」ダイアログ・ボックスを起動すると階層構造を表示できます。階層のプレビューを終了したら「OK」をクリックします。

注： メジャー階層では階層のプレビューを使用できません。階層自体にメンバーが表示されます。

階層での遅延キー・バインディング列の使用

階層でのキー・バインディングでは、メタデータ要素からキー・バインディングが反映されます。階層で遅延キー・バインディングを使用できます。

遅延バインディングを定義するか、Essbase Studio に遅延バインディングを生成させるかを選択できます。遅延要素のキー・バインディングを定義することを選択した場合、キー・バインディングはその階層についてのみ保存されます。キー・バインディングによってオリジナルのメタデータ要素が更新されることはありません。

➤ 階層でキー・バインディング列を使用するには:

1 「遅延」キー・バインディング・オプションを使用して次元要素を作成します(160 ページの手順 3 を参照)。

2 「メタデータ・ナビゲータ」で適切なフォルダを右クリックし、「新規」>「階層」を選択して階層エディタを起動します。

3 階層の「名前」を入力します。

デフォルト名は NewHierarchy です。

4 **オプション-「説明」を入力します。**

5 **「メタデータ・ナビゲータ」から次の手順でドラッグ・アンド・ドロップします:**

1. 「メタデータ・ナビゲータ」で、適用可能な次元要素がある場所に移動します。
2. 次元要素を選択し、「階層の編集」ダイアログ・ボックスにドラッグして、「データ」グループの「階層」列の下にドロップします。

   「キー・バインディング」のテキストが「遅延」になります。

   注： 手順のこの段階で階層を保存すると、「キー・バインディング」列のテキスト「遅延」は、Essbase Studio によって生成されたバインディングで置き換えられます。

6 **「キー・バインディング」列で、次元要素のセルをクリックし、 ボタンをクリックします。**

「エンティティの編集」ダイアログ・ボックスが表示されます。

7 **遅延バインディングを定義するための式を作成します:**

注： 遅延キー・バインディングを定義する式は、Common Platform Language(CPL)で記述します。この式は、言語で定義する構文に従ったオペランドと演算子のシーケンスです。各式では値が戻され、その値の型によって式の型が定義されます。付録 C「CPL リファレンス」を参照してください。

1. ダイアログ・ボックスの左下にある「ソース」タブを選択します。
2. メンバーを表示する次元を展開します。
3. メンバーを選択します。
4. メンバーをドラッグまたはダブルクリックして、メンバーの接続文字列を「式」テキスト・ボックスに移動します。
5. 「関数」タブを選択します。
6. SQL レベルを展開して関数の型を表示します。
7. 関数の型を展開して関数を表示します。
8. 関数を選択します。
9. 関数文字列をドラッグまたはダブルクリックして、「式」テキスト・ボックスに移動します。
10. 「演算子」タブを選択します。
11. 演算子の型を展開して演算子を表示します。
12. 演算子を選択します。
13. 演算子をドラッグまたはダブルクリックして、「式」テキスト・ボックスに移動します。

8 **オプション: 遅延バインディング定義をさらに調整するためのフィルタを追加する式を作成します:**

1. ダイアログ・ボックスの左下にある「ソース」タブを選択します。
2. メンバーを表示する次元を展開します。
3. メンバーを選択します。
4. メンバーをドラッグまたはダブルクリックして、メンバーの接続文字列を「式」テキスト・ボックスに移動します。
5. 「関数」タブを選択します。
6. SQL レベルを展開して関数の型を表示します。
7. 関数の型を展開して関数を表示します。
8. 関数を選択します。
9. 関数文字列をドラッグまたはダブルクリックして、「式」テキスト・ボックスに移動します。
10. 「演算子」タブを選択します。
11. 演算子の型を展開して演算子を表示します。
12. 演算子を選択します。
13. 演算子をドラッグまたはダブルクリックして、「式」テキスト・ボックスに移動します。

9 **「OK」をクリックします。**

「キー・バインディング」列に、作成したバインディング定義が表示されます。

10 **「保存」をクリックします。**

注： 手順のこの段階で階層を保存すると、「キー・バインディング」列が、作成したバインディング定義になります。

階層の例

基本的な標準階層やメジャー階層のタイプから、別の種類の階層を作成できます。例については、次のトピックを参照してください

- 185 ページの「シングルチェーン階層」
- 185 ページの「マルチチェーン階層」
- 185 ページの「共有メンバーを持つマルチチェーン階層(代替階層)」
- 186 ページの「属性次元を持つマルチチェーン階層」
- 187 ページの「再帰階層」
- 189 ページの「DATE 型のメタデータ要素から構築された時間階層」

シングルチェーン階層

シングルチェーン階層は最も単純な階層です。階層のレベルは同じデータ・ソースから導出する必要があります。ただし、階層に物理要素と論理要素の両方のレベルを含めることは可能です。

図 5 に、市場の単純なシングルチェーン階層の例を示します。この階層には 2 つのレベルがあります。親レベルは、TBC データベースの地域テーブルの地域列です。子レベルは、市場テーブルの州列です。

図 5　単純なシングルチェーン階層

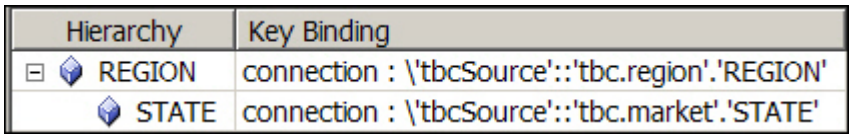
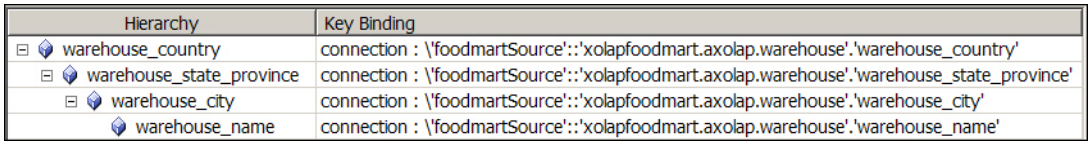


図 6 は、シングルチェーンの多世代階層で、階層に 3 つ以上のレベルが存在することを意味します。FoodMart データベースに基づいて、この階層のレベルがすべて同じテーブルから導出されています。多世代階層のレベルは同じテーブルから導出する必要はありません。

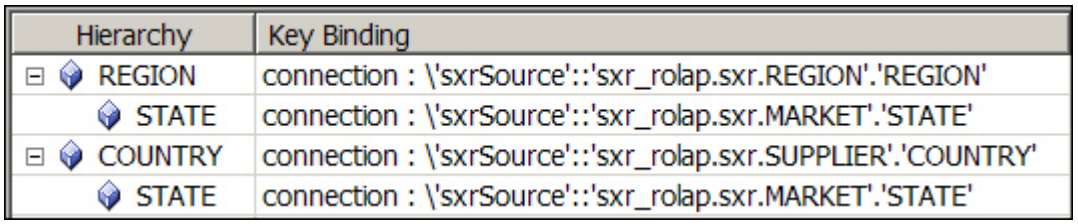
図 6　シングルチェーンの多世代階層



マルチチェーン階層

マルチチェーン階層では、同一のビジネス領域に属している要素の別のロールアップを表すことができます。図 7 には、地域における州のロールアップ、国における州のロールアップが示されています。

図 7　マルチチェーン階層



共有メンバーを持つマルチチェーン階層(代替階層)

共有メンバーを持つ次元を構築するマルチチェーン階層、つまり代替階層を設計できます。

これを実行する方法の 1 つとして、最初に次元要素のコピーを作成し、共有メンバーとして指定する方法があります。コピーを作成することで、この要素のバインディング式は、元の次元要素と同一になります。図 8 では、これは SKU_Diet で

す。次いで、チェーン内に代替階層である 2 つ目の階層を作成するときに、ユーザー定義のメンバーを親として追加し、新しい次元要素(SKU_Diet)を子として追加します。これが、図 8 に示すように共有メンバーになります。

図 8　共有メンバーを持つマルチチェーン階層(代替階層)

Hierarchy	Key Binding
⊟ FAMILY	connection : \'tbcSource'::'tbc.family'.'FAMILY'
SKU	connection : \'tbcSource'::'tbc.product'.'SKU'
⊟ Diet	"Diet"
SKU_Diet	connection : \'tbcSource'::'tbc.product'.'SKU'

属性次元を持つマルチチェーン階層

キューブの属性次元を生成するには、図 9 の説明に従って、マルチチェーン階層を構築し、同じ次元要素または物理列を各チェーンのリーフレベル・メンバーとして追加します。

図 9　属性次元を構築するためのマルチチェーン階層

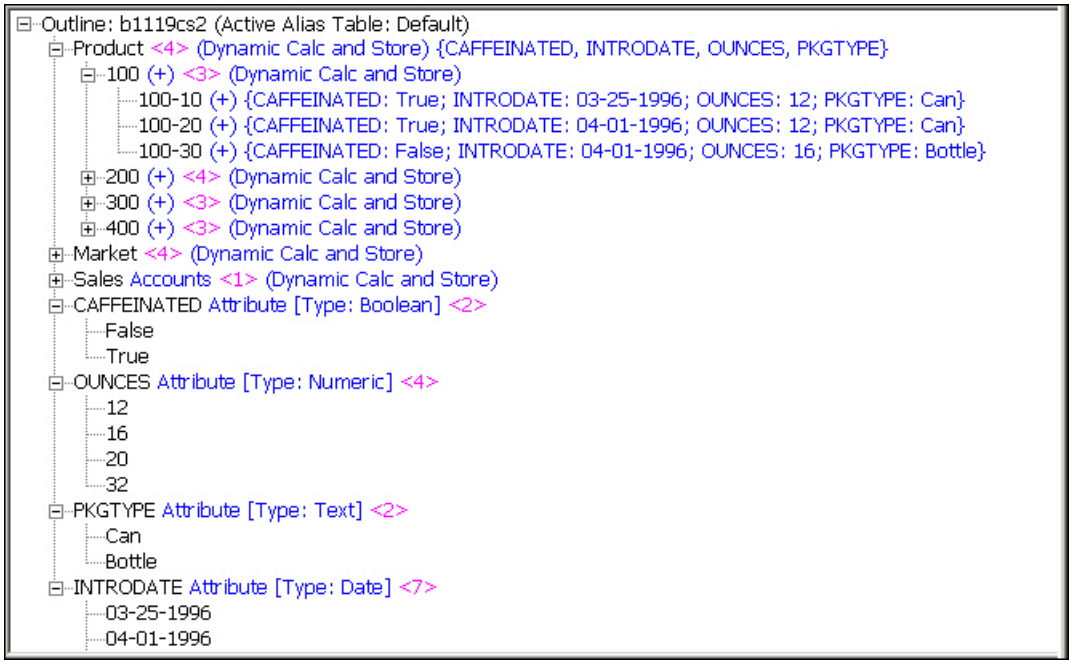
Hierarchy	Key Binding
⊟ FAMILY	connection : \'tbcSource'::'tbc.family'.'FAMILY'
SKU	connection : \'tbcSource'::'tbc.product'.'SKU'
⊟ CAFFEINATED	connection : \'tbcSource'::'tbc.product'.'CAFFEINATED'
SKU	connection : \'tbcSource'::'tbc.product'.'SKU'
⊟ OUNCES	connection : \'tbcSource'::'tbc.product'.'OUNCES'
SKU	connection : \'tbcSource'::'tbc.product'.'SKU'
⊟ PKGTYPE	connection : \'tbcSource'::'tbc.product'.'PKGTYPE'
SKU	connection : \'tbcSource'::'tbc.product'.'SKU'
⊟ INTRODATE	connection : \'tbcSource'::'tbc.product'.'INTRODATE'
SKU	connection : \'tbcSource'::'tbc.product'.'SKU'

「キューブ・スキーマ・ウィザード」で、キューブ・スキーマのこの階層を使用して Essbase モデルを構築します。Essbase モデルの構築後、「Essbase モデル・プロパティ」で、チェーン内に存在する各属性階層の親の属性メンバー・プロパティを設定します。図 9 の例では、Caffeinated、Ounces、PkgType および IntroDate メンバーの属性プロパティを設定しています。詳細は、247 ページの「属性としてのメンバーの選択」を参照してください。

モデルの属性のフォーマット・ルールを設定するには、218 ページの「モデルの属性の定義」を参照してください。

図 10 は、キューブを配置した後に表示される Essbase アウトラインの一部を示しています。

図 10



再帰階層

再帰階層には、親子関係の 2 つの列のコンテンツに基づくメンバーが含まれます。

注： ソース・データベースに再帰テーブルが含まれる場合、関連情報にアクセスし、データを正しく集計するには、親子列間に自己結合を作成する必要があります。この操作は、Essbase Studio で行うことができます。詳細は、134 ページの「ミニスキーマでの結合の追加または削除」を参照してください。

TBC サンプル・データベースで、Measures テーブルには、再帰階層の構築に必要な親子関係を持つ PARENT および CHILD の 2 つの列が含まれます。PARENT 列と CHILD 列の行の関係を次に示します:

```
   PARENT       CHILD
Measures      Profit
Profit       Margin
Margin       Sales
Margin       COGS
Profit       Total Expenses
Total Expenses   Marketing
Total Expenses   Payroll
Total Expenses   Misc
Measures      Inventory
Inventory     Opening Inventory
Inventory     Additions
Inventory     Ending Inventory
Measures      Ratios
Ratios       Margin %
Ratios       Profit %
```

```
Ratios      Profit per Ounce
```

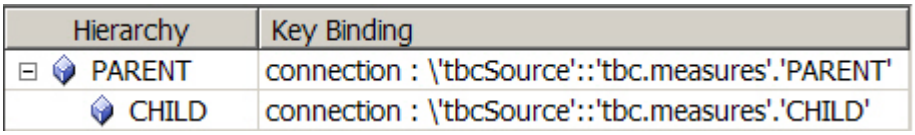
PARENT 列と CHILD 列を使用して、Essbase アウトラインは次のように作成されます:

```
  Measures
Profit
  Margin
    Sales
    COGS
  Total Expenses
    Marketing
    Payroll
    MISC
Inventory
  Opening Inventory
  Additions
  Ending Inventory
Ratios
  Margin %
  Profit %
  Profit per Ounce
```

再帰階層を正しく構築するには、再帰テーブルの親列と子列の間に自己結合を作成する必要があります。TBC サンプルを使用して、Measures テーブルの PARENT 列と CHILD 列の間に自己結合を作成します。

図 11 は、Essbase Studio で TBC サンプル・データベースの Measures テーブルの PARENT 列と CHILD 列を使用して作成された再帰階層の例です:

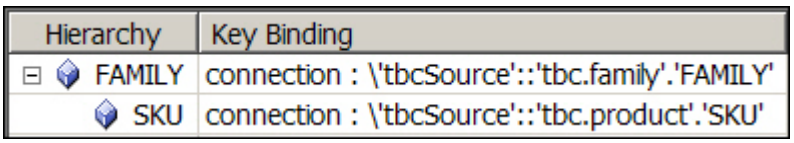
図 11　再帰(親子)階層



物理およびメタデータ要素から構築された階層

「データ・ソース・ナビゲータ」の列と「メタデータ・ナビゲータ」の列で構成された階層を構築できます。図 12 では、親メンバー FAMILY は「データ・ソース・ナビゲータ」から階層にドラッグされた列です; 子メンバー SKU は「メタデータ・ナビゲータ」から階層にドラッグされた次元要素です。

図 12　物理およびメタデータ要素から作成された混在階層



「データ・ソース・ナビゲータ」から階層に列をドラッグすると、「メタデータ・ナビゲータ」では、階層が作成されたのと同じフォルダにその列に対する次元要素が自動的に作成されます。

DATE 型のメタデータ要素から構築された時間階層

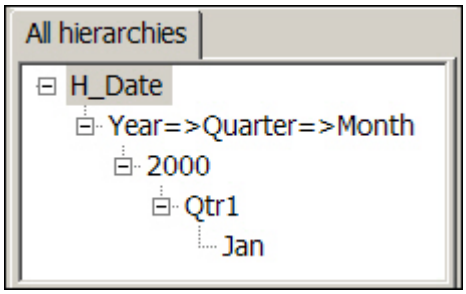
図 13 で示すような時間階層を構築するには日付タイプのメタデータ要素が必要です。開始前に、165 ページの「日付要素の作成」の手順を完了してください。

図 13　年、四半期および月レベルのある時間階層

Hierarchy	Key Binding
Year	'year'(connection : \'tbcSource'::'tbc.sales'.'TRANSDATE')
Quarter	'quarterAsString'(connection : \'tbcSource'::'tbc.sales'.'TRANSDATE')
Month	'monthShortName'(connection : \'tbcSource'::'tbc.sales'.'TRANSDATE')

図 14 に、階層エディタで「プレビュー」ボタンをクリックしたときに表示される階層の例を示します。

図 14　時間階層のプレビュー



カレンダ階層の作成

「カレンダ階層の編集」ダイアログ・ボックスでは、標準企業カレンダをモデルにした時間次元を構築する 1 つの方法が提供されます。標準(グレゴリオ暦)カレンダ、業界固有のカレンダ、またはカスタマイズされたバージョンのカレンダを基に時間次元を構築できます。

注：　カレンダ階層では、可変属性はサポートされません。

すべてのカレンダ・タイプに対して、Essbase Studio では、年、半期、四半期などの選択対象となる期間定義のリストが提供されます。

Essbase Studio では、次のカレンダ・タイプの階層がサポートされます:

- グレゴリオ暦-190 ページの「グレゴリオ暦カレンダ階層」を参照してください。
- 会計-190 ページの「会計カレンダ階層」を参照してください。

- リテール-192 ページの「リテール・カレンダ階層」を参照してください。
- ISO-193 ページの「ISO カレンダ階層」を参照してください。
- 製造-194 ページの「製造カレンダ階層」を参照してください。

注： カレンダ階層を使用してキューブ・スキーマおよび Essbase モデルを構築する場合、キューブ・スキーマの作成時に、カレンダ階層内の最下レベルのデフォルトのデータ・ロード・バインディングを指定する必要があります。詳細は、204 ページの「データ・ロード・マッピングの定義」を参照してください。

グレゴリオ暦カレンダ階層

グレゴリオ暦カレンダは、1 月 1 日から始まり 12 月 31 日で終わる 12 か月の標準的なカレンダです。グレゴリオ暦カレンダには、年、半期、三半期、四半期、月、週、日などの期間のメンバーがあります。

➤ グレゴリオ暦カレンダ階層を作成するには:

1. 「メタデータ・ナビゲータ」で適切なフォルダを右クリックして「新規」>「カレンダ階層」を選択し、「カレンダ階層の編集」ダイアログ・ボックスを起動します。
2. 「階層名」で、グレゴリオ暦カレンダ階層の名前を入力します。
3. 「カレンダ」から、「グレゴリオ暦」を選択します。
4. 「パラメータのモデリング」で、モデリングする期間の開始日および終了日を入力し、「週最初の日」を選択します。
5. 時間レベルを定義する方法は、195 ページの「時間レベルの定義」を参照してください。

   時間レベルはすべてのカレンダ・タイプに適用されます。
6. 日付属性を定義する方法は、196 ページの「日付属性の定義」を参照してください。

   日付属性はすべてのカレンダ・タイプに適用されます。
7. リンク値属性を設定する方法は、196 ページの「リンク値属性」を参照してください。

   リンク値属性はすべてのカレンダ・タイプに適用されます。
8. 「OK」をクリックして、「カレンダ階層の編集」ダイアログ・ボックスを閉じます。

会計カレンダ階層

会計カレンダ定義は、会社のレポート要件に基づいており、任意の日付から開始できます。このカレンダでは、1 週間は 7 日間です。12 か月のレポート期間では、4 週間の月が 2 か月と 5 週間の月が 1 か月の合計 3 か月が、四半期パターンとして繰り返されます。各四半期の 3 か月パターン(週数が 4-4-5、4-5-4 または 5-4-4)

を選択すると、指定したパターンに基づいて月が定義されます。1 年間が 53 週間の場合は、いずれかの月に余剰の週を追加できます。

会計カレンダは、選択した時間レベルをカレンダ次元階層として構築します。次の時間レベルのいずれかを選択します:

- 年
- 半期
- 三半期
- 四半期
- 月
- 週
- 日

➤ 会計カレンダ階層を作成するには:

1 **「メタデータ・ナビゲータ」で適切なフォルダを右クリックして「新規」>「カレンダ階層」を選択し、「カレンダ階層の編集」ダイアログ・ボックスを起動します。**

2 **「階層名」に、会計カレンダ階層の名前を入力します。**

3 **「カレンダ」から「会計」を選択し、「セマンティク・ルール」リンクをクリックします。**

4 **「年セマンティク・ルール」と「月セマンティク・ルール」で、次のオプションを選択して会計年度の構造を調整します。**

「年セマンティク・ルール」で選択した内容によって、「月セマンティク・ルール」のオプションが変わります。

すべてのオプションで、このグループの「月」、「週」または「日」のドロップダウン・ボックスが使用されます。

年セマンティク・ルール

- 「開始週 - 月内の番号」-会計年度が開始する「月」と「週」を選択します。
- 「終了週 - 月内の番号」会計年度が終了する-「月」と「週」を選択します。
- 「年は指定日を含む週に始まります」-「月」および「日」から特定の日付を選択します。会計年度はこの日付を含む週の最初の日に開始します。
- 「年は指定日を含む週に終了します」-「月」および「日」から日付を選択します。会計年度はこの日付を含む週の最終日に終了します。
- 「年は指定日を含む週またはその直後の週から始まる」-「月」および「日」から日付を選択します。会計年度は、この日付の週の最初の日、またはこの日付の翌週の最初の日に開始します。
- 「年は指定日に始まります」-「月」および「日」から日付を選択します。会計年度はこの日付に開始します。
    - 「53 週を適用」-第 1 週または最終週のいずれか選択した方に、完全な週にするための日数が追加されます。日数は隣接する会計年度から取り込まれます。

- 「52 週を適用」-選択された最初または最後の週がカレンダから削除され、直前または直後の会計年度に移動します。

月セマンティク・ルール

これらのオプションの一部は、年セマンティク・ルールの選択内容によっては使用不能になっている場合があります。

- 「指定された週番号で開始」-開始週番号は、前述の「週」の選択で指定されます。
- 「月は常に指定した日を含む週に始まります」-日付は前述の「日」の選択で指定されます。
- 「月は指定した日を含む週またはその直後の週に始まります」-日付は前述の「日」の選択で指定されます。
- 「四半期-月パターンを使用」-四半期ごとの月あたりの週数のパターンをドロップダウン・ボックスで指定します。1 年間を 53 週にする場合は、余剰週を含む月を「余剰週のある月」ドロップダウン・ボックスで指定できます。
- 「月は常に特定の日番号に始まります」-日付は前述の「日」の選択で指定されます。

「OK」をクリックして「カレンダ階層の編集」ダイアログ・ボックスに戻ります。

5 **「パラメータのモデリング」で、モデリングする期間の開始日および終了日を入力し、「週最初の日」を選択します。**

6 **時間レベルを定義する方法は、195 ページの「時間レベルの定義」を参照してください。**

時間レベルはすべてのカレンダ・タイプに適用されます。

7 **日付属性を定義する方法は、196 ページの「日付属性の定義」を参照してください。**

日付属性はすべてのカレンダ・タイプに適用されます。

8 **リンク値属性を設定する方法は、196 ページの「リンク値属性」を参照してください。**

リンク値属性はすべてのカレンダ・タイプに適用されます。

9 **「OK」をクリックして、「カレンダ階層の編集」ダイアログ・ボックスを閉じます。**

リテール・カレンダ階層

このカレンダは National Retail Federation が元になっていて、複数年にわたり前週比のデータを分析するようモデリングされています。四半期のパターンは 4-5-4 で、5 から 6 年ごとに週を移動できます。開始日は毎年異なりますが、常に 2 月初旬に当たります。前年比で比較する場合、53 週の年の 1 週目を省略して余剰週を標準化し、どちらの年の場合も一連の休日を同じにしておくのが標準的な手法です。会計カレンダには、年、半期、四半期、月、週、日といった期間のメンバーが含まれます。

➤ リテール・カレンダ階層を作成するには:

1 「メタデータ・ナビゲータ」で適切なフォルダを右クリックして「新規」>「カレンダ階層」を選択し、「カレンダ階層の編集」ダイアログ・ボックスを起動します。

2 「階層名」に、リテール・カレンダ階層の名前を入力します。

3 「カレンダ」から「小売」を選択し、「セマンティク・ルール」リンクをクリックします。

4 「月セマンティク・ルール」で選択できるのは、「四半期-月パターンを使用」のみで、次のオプションを指定できます:

- 「週パターン」-各四半期の月ごとの週パターン。
- 「余剰週のある月」-53 週の年の場合、余剰週を含む月を選択します。

「OK」をクリックして「カレンダ階層の編集」ダイアログ・ボックスに戻ります。

5 「パラメータのモデリング」で、モデリングする期間の開始日および終了日を入力し、「週最初の日」を選択します。

6 時間レベルを定義する方法は、195 ページの「時間レベルの定義」を参照してください。

時間レベルはすべてのカレンダ・タイプに適用されます。

7 日付属性を定義する方法は、196 ページの「日付属性の定義」を参照してください。

日付属性はすべてのカレンダ・タイプに適用されます。

8 リンク値属性を設定する方法は、196 ページの「リンク値属性」を参照してください。

リンク値属性はすべてのカレンダ・タイプに適用されます。

9 「OK」をクリックして、「カレンダ階層の編集」ダイアログ・ボックスを閉じます。

ISO カレンダ階層

ISO カレンダは、週 7 日で構成されています。年は、グレゴリオ暦の新年(1 月 1 日)の前後に開始できます。年は、ISO カレンダの第 1 週にグレゴリオ暦年の最初の木曜日が含まれているような日で開始するようにモデリングされます。週の初日は月曜日として定義されます。ISO 8601 カレンダ階層には、年、週および日の期間のメンバーを含めることができます。

➤ ISO カレンダ階層を作成するには:

1 「メタデータ・ナビゲータ」で適切なフォルダを右クリックして「新規」>「カレンダ階層」を選択し、「カレンダ階層の編集」ダイアログ・ボックスを起動します。

2 「階層名」で、ISO カレンダ階層の名前を入力します。

3 「カレンダ」から、「ISO」を選択します。

4 「パラメータのモデリング」で、モデリングする期間の開始日および終了日を入力し、「週最初の日」を選択します。

5 時間レベルを定義する方法は、195 ページの「時間レベルの定義」を参照してください。

時間レベルはすべてのカレンダ・タイプに適用されます。

6 日付属性を定義する方法は、196 ページの「日付属性の定義」を参照してください。

日付属性はすべてのカレンダ・タイプに適用されます。

7 リンク値属性を設定する方法は、196 ページの「リンク値属性」を参照してください。

リンク値属性はすべてのカレンダ・タイプに適用されます。

8 「OK」をクリックして、「カレンダ階層の編集」ダイアログ・ボックスを閉じます。

製造カレンダ階層

製造カレンダでは、1 年間が週 7 日の 13 期間で定義されています。13 期間は、3 期間の四半期が 3 つ、4 期間の四半期が 1 つに分かれています。各期間は 4 週ですが、1 年が 53 週になる場合、例外的に余剰週のある期間が 1 つ存在することになります。

13 期間を定義するときに、余剰期間がある四半期を指定します。1 年が 53 週ある場合は、余剰週のある期間を指定する必要があります。年の開始日を指定する場合は、年の週数が 52 週か 53 週かを指定する必要があります。

カレンダの年をどのように週に分割するかは、週の定義によって決まります。週の定義を調整して年 52 週または年 53 週にできます。

➤ 製造カレンダ階層を作成するには:

1 「メタデータ・ナビゲータ」で適切なフォルダを右クリックして「新規」>「カレンダ階層」を選択し、「カレンダ階層の編集」ダイアログ・ボックスを起動します。

2 「階層名」で、製造カレンダ階層の名前を入力します。

3 「カレンダ」から、「製造」を選択し、「セマンティク・ルール」リンクをクリックします。

4 「年セマンティク・ルール」と「月セマンティク・ルール」で、次のオプションを選択して製造年度の構造を調整します。

「年セマンティク・ルール」で選択した内容によって、「月セマンティク・ルール」のオプションが変わります。

すべてのオプションで、このグループの「月」、「週」および「日」ドロップダウン・ボックスが使用されます。

年セマンティク・ルール

- 「開始週 - 月内の番号」-製造年度が開始する「月」と「週」を選択します。
- 「終了週 - 月内の番号」–製造年度が終了する「月」と「週」を選択します。

- 「年は指定日を含む週に始まります」–「月」および「日」から日付を選択します。製造年度は、この日付を含む週の最初の日に開始します。
- 「年は指定日を含む週に終了します」-「月」および「日」から日付を選択します。製造年度はこの日付を含む週の最終日に終了します。
- 「年は指定日を含む週またはその直後の週から始まる」-「月」および「日」から日付を選択します。製造年度は、この日付の週の最初の日、またはこの日付の翌週の最初の日に開始します。
- 「年は指定日に始まります」-「月」および「日」から日付を選択します。製造年度はこの日付に開始します。
  - 「53 週を適用」-第 1 週または最終週のいずれか選択した方に、完全な週にするための日数が追加されます。日数は隣接する製造年度から取り込まれます。
  - 「52 週を適用」-第 1 週または最終週のいずれか選択した方がカレンダから削除され、隣接する製造年度に移されます。

期間セマンティク・ルール

これらのオプションの一部は、年セマンティク・ルールの選択内容によっては使用不能になっている場合があります。

- 「4 期間ある四半期」–四半期を選択します。
- 「5 週ある期間」–期間を選択します。

「OK」をクリックして「カレンダ階層の編集」ダイアログ・ボックスに戻ります。

5 **「パラメータのモデリング」で、モデリングする期間の開始日および終了日を入力し、「週最初の日」を選択します。**

6 **時間レベルを定義する方法は、195 ページの「時間レベルの定義」を参照してください。**

時間レベルはすべてのカレンダ・タイプに適用されます。

7 **日付属性を定義する方法は、196 ページの「日付属性の定義」を参照してください。**

日付属性はすべてのカレンダ・タイプに適用されます。

8 **リンク値属性を設定する方法は、196 ページの「リンク値属性」を参照してください。**

リンク値属性はすべてのカレンダ・タイプに適用されます。

9 **「OK」をクリックして、「カレンダ階層の編集」ダイアログ・ボックスを閉じます。**

時間レベルの定義

時間レベルは、カレンダ階層に設定するレベルの数、つまり深さです。

➤ 時間レベルを定義するには:

1 **「時間レベル」の下で、含める階層レベルの横にあるチェック・ボックスを選択します。**

次のことに注意してください:

- チェック・ボックスが選択済で選択解除できないレベルは必須であり、このカレンダ・タイプと一緒に使用する必要があります。
- チェック・ボックスの選択が解除されていて選択できないレベルは、このカレンダ・タイプには許可されていません。

2 **階層レベルのラベリング・ルールを変更するには、そのレベルを展開して、「ラベリング・ルール」の横にある「編集」ボタンを選択します。**

「ラベリング・ルールの編集」ダイアログ・ボックスが表示されます。

3 **「ラベリング・ルールの編集」でリストからルールを選択し、「OK」をクリックします。**

日付属性の定義

➤ 日付属性を定義するには:

1 **「カレンダ階層の編集」で「日付属性」をクリックします。**

2 **「予約された日付」で、目的の曜日ごとにチェック・ボックスを選択します。**

予約日を使用可能にすると、選択した曜日ごとに属性次元が追加されます。デフォルトの次元名は指定されていますが、この名前は選択した曜日の「次元名」の下にあるテキスト・ボックスで上書きできます。

選択した曜日に対して複数の次元名を指定できます。複数の次元名を区切るにはカンマを使用します。たとえば、「日曜日」を選択して「週」と「日曜日」の 2 つの次元名を指定する場合、次の構文を使用します:

```
Week,Sunday
```

3 **選択した日付に休日属性を追加するには、「追加」をクリックしてからカレンダ・ツールから休日を選択します。**

休日属性を日付から削除するには、日付をクリックして「削除」を選択します。

4 **日付レベルのメンバーごとに曜日の名前の属性を追加するには、「日付モデリングを実行」を選択し、日付レベルのメンバーごとに曜日名を割り当てます。**

5 **「OK」をクリックして「カレンダ階層の編集」ダイアログ・ボックスに戻ります。**

リンク値属性

リンク値属性(LVA)は、時間次元メンバーの周期性を記述するために使用される属性タイプです。周期性とは、時間次元メンバー間で共有されるパターンで、時間ベースの分析で時間次元メンバーが意味を持つようにするものです。たとえば、

1 月と 4 月はどちらも、グレゴリオ暦カレンダの四半期の開始月という周期性を持ちます。

周期的に反復するメンバーを分析用に集約する場合、このダイアログ・ボックスを使用してメンバーに LVA を定義します。年単位の四半期、年単位の月または四半期単位の週などのパラメータに基づいて集約できます。

➤ LVA を定義するには、定義する集約ごとにチェック・ボックスを選択してから、必要に応じて次元名と別名接頭辞を変更します。

階層の編集

階層を編集する場合、階層はキューブ・スキーマと Essbase モデルで使用されることに注意してください。階層を変更すると、階層から構築されたキューブ・スキーマとモデルがその階層との同期を取れなくなります。基本階層を変更する場合は、キューブ・スキーマと Essbase モデルを再作成してください。

➤ 階層を編集するには:

1 **「メタデータ・ナビゲータ」で階層を選択して右クリックし、「編集」を選択します。**

2 **次のタスクを実行します:**

- 標準階層またはメジャー階層を編集する場合、180 ページの「標準階層およびメジャー階層の作成」の手順 2 から始まる手順に従います。
- カレンダ階層を編集する場合、189 ページの「カレンダ階層の作成」を参照して適切なカレンダ・タイプを選択し、そのカレンダ・タイプに対して手順 2 以降の手順に従います。

10 キューブ・スキーマ

この章の内容

キューブ・スキーマの概要..... 199
キューブ・スキーマの作成または編集..... 199
既存のキューブ・スキーマからの Essbase モデルの作成..... 206

キューブ・スキーマの概要

キューブ・スキーマは、Essbase キューブの基盤構造です。キューブ・スキーマでは、階層、メジャーおよびメジャー階層を選択し、階層にある個々のメンバーのデータ・ロードの上書きを指定します。また、キューブ・スキーマから Essbase モデルを作成するためのオプションもあります。

キューブ・スキーマの作成または編集

キューブ・スキーマ・ウィザードを使用して Essbase キューブを作成し、オプションで、Essbase モデルを作成します。キューブ・スキーマの編集にも同じインタフェースを使用します。

➤ 次のいずれかのアクションを行います:

- 新しいキューブ・スキーマを作成するには、「メタデータ・ナビゲータ」で、「新規」>「キューブ・スキーマ」を選択して「キューブ・スキーマ・ウィザード」を起動します。
- キューブ・スキーマを編集するには、「メタデータ・ナビゲータ」でその名前を右クリックし、「編集」を選択します。

キューブ・スキーマ・ウィザードは 2 つのデフォルト画面で構成され、そこから次のタスクを実行できます:

- メジャーおよび階層の選択-200 ページの「キューブ・スキーマのメジャーおよび階層の選択」を参照してください。
- キューブ・スキーマ・オプションの設定-202 ページの「キューブ・スキーマ・オプションの設定」を参照してください。

オプションの 3 番目の画面では、選択した階層、メジャーおよびメジャー階層のデータ・ロード・バインディングを定義できます。詳細は、204 ページの「データ・ロード・マッピングの定義」を参照してください。

キューブ・スキーマのメジャーおよび階層の選択

キューブ・スキーマの作成時や編集時にはメジャーおよび階層を選択します。

➤ 「キューブ・スキーマ・ウィザード」でメジャーおよび階層を選択するには:

1 **「メジャーおよび階層の選択」で、「キューブ・スキーマ名」を入力します。**

2 **オプション:「コメント」を入力します。**

3 **「使用可能な次元要素」リストから、このキューブ・スキーマに含める階層をドラッグし、右の「階層」ボックスにドロップします。**

同じ階層は 2 回以上追加できません。

また、「使用可能な次元要素」リストで階層を選択して「選択した階層を追加」ボタンをクリックすることでも階層を追加できます。

注: 「階層」リストから階層を削除するには、「階層」リストの階層を選択して「選択した項目を削除」ボタンをクリックします。

4 **「使用可能な次元要素」リストから 1 つ以上のメジャーまたは 1 つのメジャー階層をドラッグし、右の「メジャー/メジャー階層」ボックスにドロップします。**

同じメジャーまたはメジャー階層は 2 回以上追加できません。

また、「使用可能な次元要素」リストでメジャーまたはメジャー階層を選択して「選択したメジャーまたはメジャー階層を追加」ボタンをクリックすることでもメジャーまたはメジャー階層を追加できます。

注: キューブ・スキーマに追加できるメジャー階層は 1 つのみですが、メジャーは 1 つ以上追加できます。階層に編成されていない個々のメジャーを 1 つ以上追加する場合、これらは「緩い」メジャーと呼ばれます。1 つのメジャー階層か、1 つ以上の緩いメジャーのいずれかを追加する必要があります。

注: 「メジャー/メジャー階層」リストからメジャーまたはメジャー階層を削除するには、「メジャー/メジャー階層」リストのメジャーまたはメジャー階層を選択して「選択したメジャーまたはメジャー階層を削除」ボタンをクリックします。

5 **オプション: 201 ページの「階層のプレビュー」で説明されているように、このキューブ・スキーマとして選択した階層の構造を表示するには、「階層のプレビュー」をクリックします。**

階層の表示を終えたら、「OK」をクリックします。

6 **「次へ」をクリックすると、202 ページの「キューブ・スキーマ・オプションの設定」で説明されているように、ウィザードの次ページ、「キューブ・スキーマ・オプション」が開きます。**

階層のプレビュー

階層の作成時または編集時、あるいはキューブ・スキーマの作成時または編集時に、階層の構造をプレビューできます。

また、階層は「メタデータ・ナビゲータ」からいつでもプレビューできます。

➤ 階層をプレビューするには:

1 **次のアクションを実行して、「サンプル・データ」ウィンドウにアクセスします:**

- 階層エディタで、階層の作成時または編集時に「プレビュー」をクリックします。
- 「キューブ・スキーマ・ウィザード」の「メジャーおよび階層の選択」ページで、「階層のプレビュー」をクリックします。
- 「メタデータ・ナビゲータ」の階層を右クリックして、「階層のプレビュー」を選択します。
- 「メタデータ・ナビゲータ」の階層をダブルクリックして、プレビューする階層を選択します。「階層」エディタ(Essbase Studio コンソールの作業領域のタブ)で、「プレビュー」をクリックします。

2 **表示オプションを選択します:**

- 「キャプション・バインディングを使用」 - 階層の各次元要素のキャプション・バインディング式から導出されたデータを使用して階層を表示します。
- 「キー・バインディングを使用」 - 階層の各次元要素のキー・バインディング式から導出されたデータを使用して階層を表示します。

「サンプル・データ」ウィンドウが表示されます。

たとえば、TBC サンプル・データベースを使用した場合、単純な Market 階層は次のメンバーで構成されています:

- REGION。キャプション・バインディング式は `connection : \'tbcSource'::'tbc.region'.'REGION' . toString`;キー・バインディング式は `connection : \'tbcSource'::'tbc.region'.'REGIONID' . toString`
- STATE。キャプション・バインディング式は `connection : \'tbcSource'::'tbc.market'.'STATE' . toString`;キー・バインディング式は `connection : \'tbcSource'::'tbc.market'.'STATEID' . toString`

「キャプション・バインディングを使用」オプションを選択すると、プレビュー結果は次のようになります:

```
     Market
REGION=>STATE
 East
  New York
  Massachusetts
  Florida
```

```
  Connecticut
  New Hampshire
 West
  California
  Oregon
  etc.
```

「キー・バインディングを使用」オプションを選択すると、プレビュー結果は次のようになります:

```
     Market
REGION=>STATE
 1
  1
  2
  3
  3
  5
 2
  6
  7
  etc.
```

注： プレビューする Performance Management Architect の階層を選択すると、個々のメンバーのプロパティが表示される「EPMA 次元のプレビュー」・ウィンドウが表示されます(手順 4 を参照)。

3 **「キューブ・スキーマ・ウィザード」の「メジャーおよび階層の選択」ページから階層をプレビューする場合は、次の手順を実行します:**

1. 「サンプル・データ」ウィンドウの上部にあるタブをクリックして、キューブ・スキーマの 1 つの階層またはすべての階層を表示します。
2. 階層サンプルの表示が終了したら、「OK」をクリックして「キューブ・スキーマ・ウィザード」に戻ります。

4 **Performance Management Architect 次元をプレビューする場合、「EPMA 次元のプレビュー」・ウィンドウで、次の手順を実行します:**

1. 左のフレームにある階層を展開します。
2. 階層のメンバーを選択し、右の「プロパティ名」および「プロパティ値」列にリストされるプロパティを表示します。
3. プロパティを表示する各メンバーに対して、手順 4.2 を繰り返します。

キューブ・スキーマ・オプションの設定

キューブ・スキーマ・ウィザードで、キューブ・スキーマの作成または編集時に「キューブ・スキーマ・オプションの設定」ページを使用します。デフォルトのデータ・ロード・マッピングを上書きするかどうか、およびキューブ・スキーマから Essbase モデルを作成するかどうかのオプションを指定します。

➤ キューブ・スキーマ・ウィザードでオプションを設定するには:

1 **オプション: デフォルトのロード・メンバーおよびデータ・ロード・バインディングを定義するには、「キューブ・スキーマ・オプション」ページで、「デフォルト・データ・ロード・バインディングを上書き」チェック・ボックスを選択します。**

注 : キューブ・スキーマにカレンダ階層が含まれる場合、「デフォルト・データ・ロード・バインディングを上書き」チェック・ボックスを選択する必要があります。

階層、メジャーまたはメジャー階層がそれぞれ異なるデータ・ソースに由来するものである場合、デフォルトのデータ・ロード・バインディングを上書きして、キューブの配置中に発生するエラーを最小限に抑えることができます。

このチェック・ボックスを選択した場合、「次へ」ボタンが使用可能になります。このウィザードの「データ・ロード・マッピングの定義」ページを完了する方法の詳細は、204 ページの「データ・ロード・マッピングの定義」を参照してください。

2 **オプション: 「Essbase モデルの作成」を選択して、このキューブ・スキーマから Essbase モデルを作成し、次の手順を完了します:**

1. 「モデル名」に入力します。
2. キューブ・スキーマのメジャーとしてメジャーが 1 つのみが指定されている場合、「会計次元」で、このモデルの勘定科目の生成元の階層を指定します。

   キューブ・スキーマのメジャーとしてメジャーが複数指定されている場合、会計次元はメンバーとしてのメジャーを使用して作成されます。メジャー階層が指定されている場合、その階層が会計次元としてタグ付けされます。

   Essbase Studio で会計次元を生成できるようにするには、「システムにより生成」を選択します。

3 **Performance Management Architect ベースのモデルの場合のみ: 「モデル・タイプ」で、このモデルのデータ・ストレージ・タイプを「集約ストレージ」と「ブロック・ストレージ」のいずれにするかを選択します。**

注 : キューブ・スキーマの作成プロセス中に、データ・ストレージ・タイプを選択する必要があります。Essbase モデルでは変更できません。

4 **キューブ・スキーマの構築を開始するには、「終了」をクリックします。指定した場合は、Essbase モデルが構築されます。**

または、手順 1 で「デフォルト・データ・ロード・バインディングを上書き」チェック・ボックスを選択した場合は、「次へ」をクリックして204 ページの「データ・ロード・マッピングの定義」の指示に従ってください。

注 : キューブ・スキーマの処理中にメモリー不足エラーが発生した場合は、78 ページの「仮想メモリーの構成」の説明に従って、Essbase Studio コンソールの仮想メモリー設定を増やします。

5 **オプション:** このプロセスが完了したら、Essbase Studio コンソールの作業領域で、Essbase モデルのグラフィカル表示を確認します。

データ・ロード・マッピングの定義

ウィザードのこのページは、ウィザードの「キューブ・スキーマ・オプション」ページで「デフォルト・データ・ロード・バインディングを上書き」チェック・ボックスを選択した場合に表示されます。データのロードに使用されるデフォルトのロード・バインディングを上書きするには、このダイアログ・ボックスを使用します。

次のガイドラインがデータ・ロード・マッピングに適用されます:

- データは、再帰階層を含む階層のリーフ・レベルでロードされます。したがって、データ・ロード・マッピングは階層の最下位レベルで定義します。
- 純粋なユーザー定義のマルチチェーン階層の場合は、階層全体のリーフ・レベルで 1 つのデフォルト・ロード・バインディングのみを指定できます。チェック・ボックスは「データ・ロード・バインディング」列で有効にされます。純粋なユーザー定義階層ごとに 1 つのチェック・ボックスしか選択できません。
- 混合のマルチチェーン階層、つまりリレーショナル・チェーンおよびユーザー定義のチェーンを含む階層の場合は、次のようになります:
  - 一意のリーフ・レベルごとに 1 つのデータ・ロード・バインディングを指定できます。
  - チェック・ボックスは、ユーザー定義のすべての階層チェーンのリーフ・レベルで選択できます。
- メジャー階層の場合、指定できるバインディングの数に制限はありません。唯一の条件は、リーフ・レベルに対して指定することです。
- カレンダ階層の場合は、このトピックの手順を完了する必要があります。その階層の最下位レベルに対してデフォルトのデータ・ロード・バインディングを指定する必要があります。デフォルトのデータ・ロード・バインディングを指定しないと、Essbase モデルは構築できません。
- 既存のキューブ・スキーマからすでに Essbase モデルが作成されている場合、そのキューブ・スキーマにデータ・ロード・バインディングを追加すると、キューブ・スキーマとモデルは非同期になります。既存の Essbase モデルから配置する場合、そのモデルは新しいデータ・ロード・バインディングをピック・アップしないため、配置結果は無効になります。新しいデータ・ロード・バインディングを使用してキューブを配置するには、新しい Essbase モデルを作成し、その新しいモデルから配置する必要があります。

➤ キューブ・スキーマ・ウィザードでデフォルトのデータ・ロード・バインディングを指定するには:

1 **「データ・ロード・マッピングの定義」ウィンドウで、「キューブ・スキーマ要素」列に表示された階層を確認し、どの階層にデフォルトのデータ・ロード・バインディングを指定するかを決定します。**

2 **「キー・バインディング」列のエントリを記録します。**

「キー・バインディング」列には次のものがあります:

- 変更可能なデータ・ロード・バインディングを含む要素のキー・バインディング。
- ユーザー定義メンバーの定義。
- 「遅延」キー・バインディングを含む次元要素に対する「遅延」ステータス。

この列のキー・バインディング、定義およびステータスは読取り専用で、参照用として表示されます。

3 **「データ・ロード・バインディング」列で、選択した階層の最下位レベルの行をクリックし、次の該当する処理を実行します:**

- 変更するキー・バインディングを含むリレーショナル階層、再帰階層およびカレンダ階層の場合、省略記号のボタン をクリックし、「デフォルト・ロード・バインディング」ダイアログ・ボックスを起動して、手順 4 に進みます。
- 純粋なユーザー定義のマルチチェーン階層の場合、1 つのチェック・ボックスを選択します。
- 混合のマルチチェーン階層の場合、データ・ロード・バインディング式を構築(手順 4)するか、1 つ以上のユーザー定義階層チェーンのリーフ・レベルでチェック・ボックスを選択します。
- シングルまたはマルチチェーンのメジャー階層の場合、1 つ以上のチェーンに対するデータ・ロード・バインディングを指定します(手順 4)。

4 **バインディング式を構築するには、「デフォルト・ロード・バインディング」の「式」の下で、「ソース」タブをクリックしてバインディング式の構築対象となる物理要素に移動してから、次の手順を実行します:**

1. 物理要素を「ソース」から「式」ボックスにドラッグします。
2. **オプション:** 「関数」タブをクリックして、式で使用する関数に移動し、次にその関数を「式」ボックスにドラッグします。
3. **オプション:** 「演算子」タブをクリックして、式で使用する演算子に移動し、次にその演算子を「式」ボックスにドラッグします。

   注: 関数と演算子は、式の正しい場所に正確に配置する必要があります。Essbase Studio では、これらのアイテムは正しい場所に自動的にはドロップされません。
4. **オプション:** 必要に応じて、手動で式を編集します。

5 **オプション: デフォルト・ロード・バインディングの「フィルタ」を入力します**

6 **「OK」をクリックして「デフォルト・ロード・バインディング」ダイアログ・ボックスを閉じ、「キューブ・スキーマ・ウィザード」の「データ・ロード・マッピングの定義」ページに戻ります。**

7 **ユーザー定義メンバーが含まれた階層の場合、デフォルト・ロード・メンバーを次のように示します:**

- 純粋なユーザー定義のシングルチェーン階層の場合は、階層の最下位レベルが自動的にデータ・ロード対象として選択され、選択されたことが「データ・ロード・バインディング」列のチェック・マークによって示されます。このチェック・マークは選択解除できません。
- 純粋なユーザー定義のマルチチェーン階層の場合、いずれか 1 つのチェーンのみから最下位レベルをデフォルト・ロード・メンバーとして選択します(「データ・ロード・バインディング」列のチェック・マークによって示されます)。
- 混合のシングルチェーン階層で、最下位レベルがリレーショナルまたは列ベースの場合、 をクリックし、手順 4 の詳細手順を完了して、デフォルト・ロード・メンバーのバインディング式を構築します。
- 混合のマルチチェーン階層で、最下位レベルがリレーショナルまたは列ベースの場合、いずれか 1 つのチェーンのみに対して をクリックし、手順 4 の詳細手順を完了して、デフォルト・ロード・メンバーのバインディング式を構築します。デフォルト・ロード・メンバーは、マルチチェーン混合階層の 1 つのチェーンに対してのみ指定できます。

8 **「終了」をクリックしてキューブ・スキーマの構築を開始します。指定した場合は、Essbase モデルを構築します。**

9 **オプション: このプロセスが完了したら、Essbase Studio コンソールの作業領域で、Essbase モデルのグラフィカル表示を確認します。**

既存のキューブ・スキーマからの Essbase モデルの作成

キューブ・スキーマの作成時に Essbase モデルを作成していなかった場合でも、既存のキューブ・スキーマからいつでも作成できます。

➤ キューブ・スキーマから Essbase モデルを作成するには:

1 **「メタデータ・ナビゲータ」で、Essbase モデルの作成元とするキューブ・スキーマに移動します。**

2 **キューブ・スキーマを右クリックして、「Essbase モデルの作成」を選択します。**

3 **「Essbase モデル」ダイアログ・ボックスで、「モデル名」を入力します。**

4 **オプション: 「説明」を入力します。**

5 **「会計次元」で、このモデルの会計次元の生成元となる階層を指定します。**

Essbase Studio で会計次元を生成できるようにするには、「システムにより生成」を選択します。

11 Essbaseプロパティ

この章の内容

モデル・プロパティ 207
次元のプロパティ 224
メンバーのプロパティ 245
Essbase モデル・プロパティの操作 280

モデル・プロパティ

次のトピックでは、「Essbase モデル・プロパティ」ダイアログ・ボックスの各タブで編集する Essbase モデルのプロパティを説明します:

- 「全般」タブ - 209 ページの「モデルの全般プロパティの設定」
- 「別名」タブ - 217 ページの「Essbase モデルの別名セットの選択」
- 「属性」タブ - 218 ページの「モデルの属性の定義」

モデルの詳細は、208 ページの「Essbase モデルの概要」を参照してください。

「Essbase モデル・プロパティ」ダイアログ・ボックスにアクセスするには、209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」を参照してください。

Essbase モデルについて

Essbase モデルは、リレーショナル・データベースのテーブルおよび列から作成される論理モデル(スター・スキーマ)です。Essbase モデルは、多次元データベースの構造を生成するために使用されます。

キューブ・スキーマを構築するときに、キューブに含める階層、メジャーおよびメジャー階層を指定します。Essbase モデルでは、キューブ・スキーマを構成するオブジェクトと結合がグラフィカルに表示されます。

Essbase モデルの詳細は、208 ページの「Essbase モデルの概要」を参照してください。

「メタデータ・ナビゲータ」から「Essbase モデル・プロパティ」ダイアログにアクセスできます。209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」を参照してください。

注： Essbase モデルの使用中にメモリー不足エラーが発生した場合は、78 ページの「仮想メモリーの構成」の説明に従って、Essbase Studio コンソールの仮想メモリー設定を増やします。

Essbase モデルの概要

Essbase モデルは、リレーショナル・データベースのテーブルおよび列から作成される論理モデル(スター・スキーマ)です。Essbase モデルは、多次元データベースの構造を生成するために使用されます。

キューブ・スキーマを構築するときに、キューブに含める階層、メジャーおよびメジャー階層を指定します。Essbase モデルでは、キューブ・スキーマを構成するオブジェクトと結合がグラフィカルに表示されます。

Essbase モデルは、ソース・データベースの値がファクトまたはファクトの次元のいずれかに分類できるという概念に基づいています。ファクトは、売上高や売上ユニット数など、データベースにある数値の可変値です。

次元は、値の取得または保存用にデータを整理するためのデータ・カテゴリです。次元には通常、内部でグループ化された関連メンバーの階層が含まれています。たとえば、多くの場合、年次元には四半期や月などの期間ごとのメンバーが含まれます。通常、次元はビジネスの機能に関連付けられます。製品、地域、年は代表的な次元です。次元の詳細は、224 ページの「次元の概要」を参照してください。

関連付けられたファクトは、店舗の場所や売上ユニットの製品 ID などの追加情報を提供する関連データ値です。Essbase モデルには、ファクト・テーブル、次元テーブル、次元分岐およびオプションで時間次元と会計次元が含まれています。階層の列にタグを付けて属性を使用可能にすることもできます。これらの列は、色やサイズなど、アウトラインの属性になり、Essbase でレポートを作成するための追加レイヤーを提供します。

Essbase Studio を使用して作成する Essbase モデルは、物理スター・スキーマではなく論理モデルです。このモデルは、ソース・データベース・テーブルから選択するデータ値の論理的な表現であり、Essbase でレポートを作成するために使用されます。Essbase にデータをロードするために必要な基本構造が含まれるアウトラインを作成するには、モデルを使用します。

同じスキーマから複数のモデルを作成して各モデルのプロパティを編集し、1 つ以上の Essbase アウトラインに各モデルを配置できます。

Essbase モデルには次の機能があります:

- 再使用できます。同じモデルを複数のアウトラインの基準として使用できます。
- データベース・アウトラインがソース・データベースの変更の影響を受けないようにする抽象概念のレイヤーを提供します。
- 階層を作成して、ソース・データベースのデータを構造化および要約できます。これらの階層は複数のアウトラインで使用できます。
- OLAP および XOLAP 環境をサポートしています。

213 ページの「OLAP の概要」および214 ページの「XOLAP の概要」を参照してください。

キューブ・スキーマの作成プロセス時に Essbase モデルを構築するか、既存のキューブ・スキーマから後で構築するかを指定します。1 つのキューブ・スキーマには、そのキューブ・スキーマに関連付けられている複数の Essbase モデルを含められます。

Essbase モデルを配置する前にそのプロパティを変更するには、「Essbase モデル・プロパティ」ダイアログ・ボックスを使用します。また、別のレベルでプロパティを表示および編集することもできます。224 ページの「次元の概要」および245 ページの「メンバーのプロパティ」を参照してください。

「Essbase モデル・プロパティ」ダイアログ・ボックスには、モデル・プロパティを編集するための次の 3 つのタブがあります:

- 「全般」タブ - 209 ページの「モデルの全般プロパティの設定」
- 「別名」タブ - 217 ページの「Essbase モデルの別名セットの選択」
- 「属性」タブ - 218 ページの「モデルの属性の定義」

209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」を参照してください。

「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス

➤ 「Essbase モデル・プロパティ」ダイアログ・ボックスにアクセスするには:

1 **「メタデータ・ナビゲータ」で、モデル名を右クリックします。**

2 **「Essbase プロパティ」を選択します。**

デフォルトでモデルの「全般」タブが表示されます。

モデルの全般プロパティの設定

「Essbase モデル・プロパティ」ダイアログ・ボックスの「全般」タブで、Essbase モデルの全般プロパティを設定します。

モデルの詳細は、208 ページの「Essbase モデルの概要」を参照してください。

➤ Essbase モデルの全般プロパティを設定するには:

1 **209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」の手順を使用して、「Essbase モデル・プロパティ」ダイアログ・ボックスにアクセスします。**

モデル名とキューブ・スキーマ名が、読取り専用の「名前」フィールドと「キューブ・スキーマ」フィールドに表示されます。

2 **オプション:「コメント」テキスト・ボックスに説明を入力します。たとえば、「4月の分析とあわせて使用」のように入力します。**

3 **オプション: モデルの完全な配置履歴を表示するには、「最後に配置」フィールドの右側のボタンをクリックします。**

「最後に配置」フィールドには、最近の配置に関する情報が表示されます。

注： 現在のモデルがこれまで配置されたことがない場合、「最後に配置」フィールドは空白です。

4 **オプション: 次のオプションを少なくとも 1 つ選択します:**

- 「ASO ストレージ・モデル」-モデルが集約ストレージをサポートできるようにします。

  データを集約ストレージ・フォーマットで保管する場合は、このオプションを選択します。Essbase アウトラインに大量の疎次元と 1 つか 0 の密次元が含まれる場合、このオプションを使用します。

  「ASO ストレージ・モデル」オプションを選択すると、次のメッセージが表示されます:

  「モデルを集約ストレージに設定すると、設定の一部に調節が必要となる場合があります。Essbase Studio がモデルを検査し、モデルへの調節を行う場合は「はい」をクリックし、そうでなければ「いいえ」をクリックします。

  これらの調整は、検証で警告やエラーが発生する可能性を取り除くことを目的としており、「はい」を選択することをお薦めします。「いいえ」を選択してもモデルに対してすべての操作を続行できますが、検証での警告やエラーを回避できない場合があります。

  注： 「ビジネス・モデル」オプションを使用して、Oracle BI EE データ・ソースに基づく Essbase モデルを操作する際、データ・ソース接続の作成時に「キューブ・スキーマと Essbase モデルの作成」オプションを選択した場合は、「ASO ストレージ・モデル」オプションがデフォルトで有効になります。必要に応じて、この選択はクリアできます。

- 「重複するメンバー名のサポート」-重複するメンバー名をモデルがサポートできるようにします。

  重複メンバー名をサポートする Essbase アウトラインでは、次元と親レベルが異なる場合に重複が許可されます。これにより、メンバー名が重複する可能性がある場合(顧客次元やサプライヤ次元の都市など)の分析アプリケーションの作成が大幅に容易になります。

  注： 「ビジネス・モデル」オプションを使用して、Oracle BI EE データ・ソースに基づく Essbase モデルを操作する際、データ・ソース接続の作成時に「キューブ・スキーマと Essbase モデルの作成」オプションを選択した場合は、「重複するメンバー名のサポート」オプションがデフォルトで有効になります。必要に応じて、この選択はクリアできます。

- 「Unicode 文字セットを使用」-モデルが Unicode 文字セットをサポートできるようにします。

  Essbase アプリケーションが文字テキストを UTF-8 でエンコードし、異なる言語で設定されたコンピュータを使用するユーザー間でアプリケーション・データを共有できるようにする場合、このオプションを使用します。

- 「XOLAP モデル」-モデルが XOLAP モデルとして機能できるようにします。

  214 ページの「XOLAP の概要」を参照してください。

5 **オプション:「カスタム・データ・ロード設定」ボタンをクリックして、「データ・ロード設定の定義」ダイアログ・ボックスにアクセスすると、SQL をデータ・ロード用にカスタマイズできます。**

  注: このダイアログ・ボックスは、モデル内のすべての要素がリレーショナル・ソースからのものである場合のみ使用可能になります。SQL をデータ・ロード用にカスタマイズしない場合は、「カスタム・データ・ロード設定」ボタンの右に「カスタム SQL 使用可能: FALSE」というメッセージが表示されます。

  詳細は、211 ページの「標準データ・ロード SQL の上書き」を参照してください。

6 **「型付きメジャーのサポート」で、「日付フォーマット」を選択します。**

  デフォルトの日付フォーマットは、`yyyy-mm-dd` です。

  詳細は、263 ページの「型付きメジャーの指定」を参照してください。

7 **「適用」をクリックします。**

  注: このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

標準データ・ロード SQL の上書き

このダイアログ・ボックスを使用して、データ・ロード・プロセス中に使用できるように、Essbase Studio で生成される標準的な SQL 文を編集します。

ヒント: 標準データ・ロード SQL の編集に関するヒントは、212 ページの「データ・ロード SQL の上書き編集のガイドライン」を確認してください。

➤ 標準データ・ロード SQL 文を上書きするには:

1 **209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」の手順を使用して、「Essbase モデル・プロパティ」ダイアログ・ボックスにアクセスします。**

2 **「カスタム・データ・ロード設定」ボタンをクリックし、「データ・ロード設定の定義」ダイアログ・ボックスにアクセスします。**

3 「データ・ロードにカスタム SQL を使用」を選択します。

4 **オプション:** フォルダとして SQL 文を表示するには、「ステートメントをフォルダとして表示」オプションを選択します。

デフォルトで、左のテキスト・ボックスの SQL 文はリストとして表示されます。

5 左のテキスト・ボックスで、変更する SQL 文をハイライトします。

注: SQL 文全体をハイライトする必要はありません。右のテキスト・ボックスにコピーして貼り付ける SQL 文の部分をハイライトすると(次の手順を参照)、ステートメント全体がコピーされ貼り付けられます。

6 右のテキスト・ボックスにある SQL 文のコピーを貼り付けるには、方向ボタンを使用します。

7 SQL 文を編集します。

注: SQL 文を変更すると、変更された文の総数が「変更済ステートメント」メッセージに表示されます。

8 「OK」をクリックします。

注: データ・ロード SQL を編集した後に標準データ・ロード SQL を使用する場合は、「データ・ロードにカスタム SQL を使用」チェック・ボックスを選択解除します。カスタマイズした SQL は保持されますが、使用されません。

データ・ロード SQL の上書き編集のガイドライン

Essbase Studio サーバーで生成された標準 SQL を編集して、データのロード時に使用できます。編集した SQL を選択して、Essbase データベースにデータをロードするときのパフォーマンスを向上できます。

Essbase Studio サーバーで生成された標準 SQL を編集する場合は、次のガイドラインに使用してください:

- 編集した SQL の SELECT 句内の列の順序は、標準 SQL の SELECT 句の列の順序と一致している必要があります。
- 勘定科目のタグが付けられた次元のユーザー定義メンバーは、ユーザー定義 SQL の SELECT 句にリストする必要があります。
- 編集した SQL 文の数は、標準 SQL 文の数に一致している必要はありません。
- 「データ・ロードにカスタム SQL を使用」チェック・ボックスを選択していて、1 つ以上の SQL 文に対してコピーまたは編集を行っていない場合、それらの SQL 文のデータ・ロードは無視されます。これにより、ステートメントをデータ・ロードから除外できます。

  カスタム・データ・ロードで特定のステートメントの SQL を編集しないで保持するには、「データ・ロードにカスタム SQL を使用」チェック・ボックスを選択解除した状態で、編集ペインにその SQL をコピーします。

- データ・ロードの結果作成されるデフォルトの列数は、次元当たり 1 列とすべての追加データ列の合計です。メンバーに、次元の前のメンバー(親やすべての祖先など)の接頭辞が付加されている場合、より多くの列が戻されます。
- NULL のデータ・ロード SQL 文の列がいくつかある場合、SQL 文を追加して、アウトラインの次のレベルでデータをロードできます。これは、NULL 移行処理として知られています。
- データ・ロード SQL 文はフィルタ・セットに結び付けられているため、フィルタ・セットごとに 1 つずつ、SQL 文の複数のセットを作成できます。
- データ・ロード SQL 文はアウトラインに関連けられます。
- SQL テーブルの別名を生成するプロセスでは、SQL の予約済キーワードがスキップされます。ユーザー定義 SQL で予約済キーワードが使用されていないことを確認してください。
- データ・ソース列がレベル 0 のメンバー・セットにバインドされており、基本メンバーと共有メンバーの両方を含んでいる場合、(同じ次元内に共有メンバーを持つ)特定の基本メンバーに対して Essbase Studio で生成されたデータ・ロード SQL 内に集約されるデータは、共有メンバーの数に応じて増加する可能性があります。この場合、211 ページの「標準データ・ロード SQL の上書き」の説明に従って、ユーザー独自のカスタム・データ・ロード SQL を定義することをお薦めします。この動作は、Oracle Essbase Integration Services からインポートしたメタアウトラインから作成された Essbase モデルにも、最初から Essbase Studio で作成された Essbase モデルにも当てはまります。

  たとえば、TBC サンプルを使用すると、100-10 で Diet の下に共有メンバーがあるため、100-10 に対して標準データ・ロード SQL 内に集約されるデータは本来の量の 2 倍になります。正しい集約データを取得するには、「データ・ロードにカスタム SQL を使用」オプションを選択し、手動でカスタム・データ・ロード SQL に group by 列をもう 1 つ追加します

注意:

- ユーザー定義 SQL は実行時まで検証されません。詳細は、364 ページの「ユーザー定義 SQL の検証」を参照してください。
- Essbase Studio におけるユーザー定義 SQL に対する権限は、データ・ソース・ユーザーに割り当てられている権限から継承されます。詳細は、364 ページの「ユーザー定義 SQL のデータ・ソース権限」を参照してください。

OLAP の概要

OLAP(オンライン分析プロセス)は、多次元で複数のユーザーが関わるクライアント・サーバー型の計算環境です。これは、集計された企業データをリアル・タイムで分析するユーザーを対象にしています。OLAP システムには、ドリルダウン、データのピボット、複雑な計算、トレンド分析およびモデリングの機能があります。

OLAP は、プランニングした戦略のテスト・シナリオを作成して、複雑な前提条件を含む課題を解決する必要があるビジネス・マネージャを対象としています。ユーザーは、次のようなデータ・カテゴリ間の関係を分析できます:

- 製品 A の過去 1 か月間の売上はどうだったのか?過去 5 年間の同月の売上と比較してどうなのか?
- 歩合と価格は販売員の製品 A の売上成績に影響したのか?
- 製品 B の翌月の売上はどのようになるのか?
- 製品 B の売上は特定の地域ではよかったのか?
- 過去 1 年間製品 C は顧客から返品されたか?欠陥が原因で返品されたのか?特定の工場で欠陥製品が製造されたのか?

Essbase Studio を使用し、Essbase の多次元データベースを構築することで、このような質問にすばやく答えられます。

多次元データベース

多次元データベース(MDDB)には、そのメンバーと次元の交差にある集計データを保管します。たとえば、ある会社が第 1 四半期に東部地域で製品 20 個を販売した場合、Essbase は、製品、東部、第 1 四半期、ユニット売上の交差に 20 を保管します。

多次元データベースでは、次元はビジネス・プランのコア・コンポーネントを表すデータ・カテゴリであり、多くの場合、ビジネス機能と関連しています。製品、地域、年は代表的な次元です。ほとんどのデータベースで、アプリケーションの存続期間中に次元が変更されることはほとんどありません。

多次元データベースでは、メンバーは次元の個別のコンポーネントです。たとえば、製品 A と製品 B は製品次元のメンバーです。各メンバーには一意の名前があります。1 つの次元には複数のメンバーを入れることができます。一部の次元では、アプリケーションの有効期間内にメンバーが頻繁に変更されます。

メンバーは、複数のメンバーの親および他のメンバーの子になることができます。Essbase のアウトラインでは、下にあるメンバーを互いにインデントして連結関係を示します。

XOLAP の概要

Extended Online Analytic Processing(XOLAP)は、ビジネス・インテリジェンスにおける OLAP の役割のバリエーションです。具体的には、XOLAP は、アウトライン・メタデータのみを保管し、クエリー時にリレーショナル・データベースからデータを取得する、Essbase 多次元データベースです。このように、XOLAP はソース・リレーショナル・データベースを Essbase データベースと統合し、リレーショナル・データベースの拡張容易性とともに、多次元データベースのより高度な分析機能を活用します。OLAP と XOLAP のどちらが環境に適しているかは、ビジネス・ニーズによって決まります。

OLAP の詳細は、213 ページの「OLAP の概要」を参照してください。

OLAP と XOLAP では、メタデータ・アウトラインとその基底データの保管場所が異なります:

- OLAP では、メタデータと基底データは、Essbase データベースにあります。

- XOLAP では、メタデータは Essbase データベースにあり、基底データはソース・リレーショナル・データベースにとどまります。

メタデータとデータの場所の違いは OLAP と XOLAP の機能に影響するため、Essbase Studio を使用する際に XOLAP がどのように役立つかを理解するために重要です。

OLAP は従来のリレーショナル・データ・ストレージとデータ分析に役立ちます。XOLAP は、(Integration Services のユーザーによく知られている)ハイブリッド分析や拡張リレーショナル・アクセスなどの混合(ハイブリッド)環境でサポートされる操作に適しています。ハイブリッド分析と拡張リレーショナル・アクセスの基本概念の多くが、Essbase Studio の XOLAP キューブの機能に組み込まれています。

XOLAP ワークフロー

XOLAP 環境でのデータ取得のワークフローは、非 XOLAP 環境の場合と次の点で類似しています:

1. モデルは、XOLAP 対応として指定されます。

   216 ページの「XOLAP へのモデルの指定」を参照してください。

2. キューブは配置されますが、その時点でデータはロードされません。
3. Essbase データベースは、Smart View、Oracle Essbase Visual Explorer、または Essbase データベースにアクセス可能なその他のレポート作成ツールを使用してクエリーされます。
4. Essbase では、必須 SQL が動的に生成されて、ソース・リレーショナル・データベースからデータが取得されます。

XOLAP を使用するためのガイドライン

369 ページの「XOLAP の機能のガイドライン」も参照してください。

XOLAP には、いくつかの制限とサポートされていない用途があります。

XOLAP の制限

XOLAP には、次のような制限があります:

- 管理サービス・コンソールの「アウトライン・エディタ」では XOLAP アウトラインを変更できないわけではありませんが、許可はされていません。XOLAP の操作では、アウトラインの作成後、構造と次元テーブルのコンテンツに自動的に変更が組み込まれることはありません。アウトラインを変更するには、基礎となる Essbase モデルを変更し、新しい Essbase データベースに再配置する必要があります。
- XOLAP 対応モデルの増分構築がサポートされています。

  次の操作のいずれかを(単独または組合せて)Essbase モデルで実行する場合、XOLAP キューブで増分ロードを実行できます。:

  - 階層にメンバーを追加する
  - 階層からメンバーを削除する

- 階層内のメンバーを再度親にする(再編成する)

- 導出テキスト・メジャーが Essbase モデルを構築するためにキューブ・スキーマで使用されている場合、そのモデルに XOLAP は対応できません。
- XOLAP を使用できるのは、ストレージの集約時のみです。データベースでは、自動的に重複メンバーが使用可能になります。
- Essbase Studio は XOLAP 対応の Essbase モデルに属性次元を作成できます; ただし、属性次元の子レベルは 1 つのみであることが必要です。
- XOLAP では、カタログに対応するスキーマ・マッピングを持たない次元がサポートされていますが、このような次元の場合、保管されるメンバーとなるメンバーは 1 つだけです。
- XOLAP モデルのユーザー定義メンバーには、親または子メンバーを保管できません。階層は動的階層として指定する必要があります。
- XOLAP で値を計算するには、ユーザー定義メンバーに有効な式が関連付けられている必要があります。
- 「Essbase モデル・プロパティ」ダイアログ・ボックスの「全般」タブでは、次元サーバー(Performance Management Architect)ソースから作成されたモデルに対して、「XOLAP モデル」オプションが使用不可になっています。

XOLAP でサポートされていない用途

XOLAP では、次の用途はサポートされていません:

- フラット・ファイル
- 不完全な階層
- 再帰階層
- カレンダ階層
- フィルタ
- テキスト・メジャー
- 複数のリレーショナル・データ・ソース
- 次元サーバー(Performance Management Architect)データ・ソース

XOLAP へのモデルの指定

モデルを指定して、XOLAP で使用可能にできます。

モデルの詳細は、208 ページの「Essbase モデルの概要」を参照してください。

XOLAP の詳細は、214 ページの「XOLAP の概要」を参照してください。

➤ XOLAP にモデルを指定するには:

1 **209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」の手順を使用して、「Essbase モデル・プロパティ」ダイアログ・ボックスにアクセスします。**

モデル名とキューブ・スキーマ名が、読取り専用の「名前」フィールドと「キューブ・スキーマ」フィールドに表示されます。

2 「XOLAP モデル」チェック・ボックスを選択します。

3 「適用」をクリックします。

Essbase モデルの別名セットの選択

「Essbase モデル・プロパティ」ダイアログ・ボックスの「別名」タブで、モデルで使用する別名セットを選択します。

別名の詳細は、266 ページの「別名の概要」を参照してください。

モデルの詳細は、208 ページの「Essbase モデルの概要」を参照してください。

注： データ・ロードを最適化する場合(243 ページの「データ・ロードの最適化」を参照)、Essbase がデータ・ロードの別名最適化に使用する別名を保持するために、別名テーブルが 1 つ必要になります。合計 32 個のうち、その他の 31 個のテーブルは、別名セットとして選択できます。31 個を超えるテーブルを選択すると、一部のテーブルがデータ・ロード中にドロップされます。

➤ Essbase モデルで使用する別名セットを選択するには:

1 209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」の手順を使用して、「Essbase モデル・プロパティ」ダイアログ・ボックスにアクセスします。

2 Essbase モデル名をフォーカスした状態で、「別名」タブを選択します。

3 テーブルを「使用可能なテーブル」リストから「選択したテーブル」リストに移動するために、アクションを実行します:

- 1 つ以上の別名セットを選択し、「選択した別名セットをモデルに追加」( )をクリックします。

  注： テーブルをダブルクリックして移動することもできます。

- 「すべての使用可能な別名セットをモデルに追加」( )をクリックします。

4 「キューブでの名前」列で重複名のある別名セットを選択した場合、「メタデータ・ナビゲータ」で、その別名セットの名前変更を求めるメッセージが表示されます。

異なるフォルダ内にある別名セットは同じ名前にすることが可能であるため、別名セットに新しい(別の)名前を入力すると、すべての別名テーブル名がキューブ内で一意になります。

5 オプション: 「選択したテーブル」ボックスでテーブルを選択して、次のいずれかのボタンをクリックすることにより、別名セットの順序を並べ替えます:

- 「選択した別名をリストで上へ移動」( )

- 「選択した別名をリストで下へ移動」( )

6 **テーブルを「選択したテーブル」リストから「使用可能なテーブル」リストに移動するために、アクションを実行します:**

- 1 つ以上の別名セットを選択し、「選択した別名セットをモデルから削除」( )をクリックします。

  注： テーブルをダブルクリックして移動することもできます。

- 「すべての別名セットをモデルから削除」( )をクリックします。

7 **「適用」をクリックします。**

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

モデルの属性の定義

「Essbase モデル・プロパティ」ダイアログ・ボックスの「属性」タブで、モデルの属性次元の作成方法を定義します。

属性を使用して、次元の特性、つまり属性に関するデータの取得および分析を行います。たとえば、サイズやパッケージに基づいて製品の収益性を分析できます。また、各市場の地域における人口規模などの市場の属性を分析に組み込むことで、より有効な結論が出せます。

属性の詳細は、218 ページの「属性の概要」を参照してください。

可変属性の詳細は、219 ページの「可変属性の概要」を参照してください。

次のトピックでは、「属性」タブで編集する Essbase モデル・プロパティについて説明します:

- 221 ページの「属性メンバー名のフォーマットの設定」
- 221 ページの「属性計算メンバー名のフォーマットの設定」
- 223 ページの「属性のブール、日付および数値の範囲の指定」

属性の概要

属性は、製品のサイズや色などのデータ特性を記述します。属性を使用して、次元のメンバーをその特性に基づいてグループ化したり分析したりできます。

属性の分析により、たとえば、小規模市場で缶で販売されるカフェインレス飲料の売上が予想よりも低いなどの傾向がわかります。さらに、同様の市場区分での様々な製品の最低売上または最高売上など、特定の属性条件による分析をフィルタできます。

一般的な特徴に基づいてデータを選択、集約およびレポートできます。また、次のようないくつかの集計メソッドから選択できます:

- 合計
- カウント
- 平均
- 最小
- 最大

属性のタイプは、次のとおりです:

- テキスト
- 数値
- ブール
- 日付

次の例で示すように、属性による分析で深い見通しが可能になり、より確実な決定を行うことができます:

- 共通の特徴(属性)に基づいてデータを選択、集約およびレポートできます。
- 属性をテキスト、数値、ブールまたは日付型として定義することで、AND、OR、NOT、<、>、および=の比較などの型関連の関数を使用してデータをフィルタ(選択)できます。
- 数値属性タイプを使用して、属性範囲別に統計値をグループ化できます。たとえば、 <500,000、500,000-1,000,000 および>1,000,000 などの人口グループがあります。
- 属性データの合計、カウント、最小値または最大値、および平均値を表示できます。
- 計算スクリプトおよびメンバー式で、数値属性値を使用して計算を実行できます。
- データをドリル・ダウンして詳細情報を調べたり、ドリル・アップしてデータの要約を参照したりできます。

可変属性の概要

製品には通常、その製品を規定または定義する属性があります。たとえば、ある製品がサイズをオンスで示す属性と味を示す属性を持つとします。その場合、製品が基本次元になり、オンスと味が属性次元になります。

注： 基本次元と属性次元の詳細な説明は、Oracle Essbase Database Administrator's Guide を参照してください。

可変属性を使用すると、独立次元と呼ばれる第 3 の次元に関連して 2 つの値を追跡できます。たとえば、8 オンスの製品を 1 年間追跡できます。このシナリオでは、時間が独自次元です。この第 3 の係数は変わることがあります(そのため可変という名前がついています)。たとえば、製品を 1 年間、1 四半期、または 1 か月にわたって追跡できます。

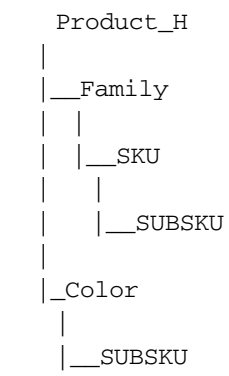
もう 1 つの例として、次のシナリオを考えてください: ある顧客の営業担当員が 1 年の途中で変わります。顧客の合計売上と営業担当員の 6 か月間の担当は次のようになります:

表 4　可変属性の例: 営業担当員の途中変更

3 月	4 月	5 月	6 月	7 月	8 月
4000	6000	2000	1000	1000	7000
ジョーンズ	ジョーンズ	ジョーンズ	スミス	スミス	スミス

この例では、営業担当員が可変属性です。データ取得によって、営業担当員ジョーンズが 3 月から 5 月まで顧客に合計$12,000 相当の製品を販売し、営業担当員スミスが 6 月から 8 月まで顧客に合計$9,000 相当の製品を販売したことがわかっています。可変属性を使用しないと、認識できるのは現在の営業担当員スミスのみであり、総販売額($21,000)がスミスの成績になります。

可変属性は、メンバーをグループ化する代替手段になります。たとえば、色を使用して SKU をグループ化できます。このシナリオでは、属性次元「色」が SUBSKU に関連付けられます:



Essbase Studio で色が可変属性として設定されると、次の表に似た結果が取得されます:

表 5　SUBSKU に関連付けられた色の可変属性の例

SUBSKU	SKU
赤	100
白	400
白	600
黒	200
黒	300
銀	500

可変属性には複数のチェーンが必要で、リーフ・レベルは一致している必要があります。

属性メンバー名のフォーマットの設定

接頭辞または接尾辞を割り当てて、属性名を一意に設定できます。接頭辞または接尾辞の識別子を割り当てると、アウトラインに属性メンバーのフル・ネームが表示されないので注意してください。情報を取得すると属性のフル・ネームを表示できます。たとえば、スプレッドシートを表示すると属性のフル・ネームが表示されます。

➤ Essbase モデルで属性のメンバー名のフォーマットを設定するには:

1 209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」の手順を使用して、「Essbase モデル・プロパティ」ダイアログ・ボックスにアクセスします。

2 「属性」タブを選択します。

3 「メンバー名フォーマット」グループを検索します。

4 接頭辞または接尾辞として使用する「値」を次の中から選択します:

注： メンバー名に接頭辞と接尾辞の両方を追加できます。

- 「なし」
- 「親」
- 「親の親」
- 「祖先」
- 「次元」

5 接頭辞または接尾辞で使用する「区切り記号」を次の中から選択します:

- 「アンダースコア」
- 「パイプ」
- 「キャレット」

6 「接頭辞」または「接尾辞」を選択して、選択した値をどちらに使用するかを指定します。

選択した接頭辞または接尾辞に基づいて、「サンプル」フィールドに属性名の例が表示されます。

7 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

属性計算メンバー名のフォーマットの設定

属性計算メンバー名のフォーマットを設定できますが、属性計算メンバー名はメンバー名と同じにはできません。

属性計算の詳細は、223 ページの「属性計算の概要」を参照してください

➤ Essbase モデルの属性計算メンバー名のフォーマットを設定するには:

1 209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」の手順を使用して、「Essbase モデル・プロパティ」ダイアログ・ボックスにアクセスします。

2 「属性」タブを選択します。

3 「属性計算メンバー名」グループを探します。

4 **オプション:**「次元」テキスト・ボックスで、属性計算次元の名前を変更します。

5 **オプション:**「合計メンバー」テキスト・ボックスで、合計メンバーの名前を変更します。

   たとえば、Sum を Total に変更できます。

   この属性では、メンバーの集計プロパティまたは式に基づいてメンバーが合計されます。合計では、合計が計算されますが、常に単純合計を計算するわけではありません。合計では、メンバーの集計プロパティまたは式に基づいてメンバーが合計されます。たとえば、12 オンスの飲料の収益(%)を計算できます。合計は、Profit%式に基づいており、収益が売上で除算され、100 で乗算されます。

6 **オプション:**「カウント・メンバー」テキスト・ボックスで、カウント・メンバーの名前を変更します。

   この属性では、指定した属性または属性の組合せでメンバーの数が計算されます。カウントには、存在するデータ・ブロックを持つメンバーのみが含まれます。

7 **オプション:**「最小メンバー」テキスト・ボックスで、最小メンバーの名前を変更します。

   この属性では、指定した属性または属性の組合せの最小データ値が計算されます。

8 **オプション:**「最大メンバー」テキスト・ボックスで、最大メンバーの名前を変更します。

   この属性では、指定した属性または属性の組合せの最大データ値が計算されます。

9 **オプション:**「平均メンバー」テキスト・ボックスで、平均メンバーの名前を変更します。

   この属性では、指定した属性またはその組合せの力学的平均値(平均値)が計算されます。平均(合計を個数で除算)には、存在するデータ・ブロックを持つメンバーのみが含まれます。

10 「適用」をクリックします。

注: このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

属性計算の概要

属性計算には、次のプロパティがあります:

- システム定義: アウトラインに属性次元を作成すると、生成されるレポートおよびスプレッドシートには、すべての属性メンバーで使用できる定義済の計算が 5 つ含まれます。
- 動的計算: ユーザーが計算を要求すると、すべての属性計算が実行され、その後計算が破棄されます。計算済データはデータベースに保管できません。
- レポートでの表示: 属性計算の結果はアウトラインに表示されません。計算結果は、スプレッドシートおよびレポートでのみ使用できます。

属性のブール、日付および数値の範囲の指定

メンバーに対して、ブール、日付および数値の範囲を指定できます。

➤ メンバーに対して属性のブール、日付および数値の範囲を指定するには:

1 209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」の手順を使用して、「Essbase モデル・プロパティ」ダイアログ・ボックスにアクセスします。

2 「属性」タブを選択します。

3 「ブール、日付、および数値範囲」グループを探します。

4 「TRUE メンバー」テキスト・ボックスに、属性メンバー名を入力します。

メンバー名 TRUE がアウトラインの別の場所で使用されている場合、「TRUE メンバー名」テキスト・ボックスに `Yes`、`Allowed` または別の適切な名前を入力します。デフォルトのメンバー名は、「TRUE」です。

5 「日付」ドロップダウン・リストから、日付フォーマットを選択します。

- 「mm-dd-yyyy」-たとえば、2000 年 10 月 19 日は、10-19-2000 と表示されます。

  注: 日付型の属性メンバーに対しては、別の値を指定しない限り、デフォルトの mm-dd-yyyy フォーマットが割り当てられます。

- 「dd-mm-yyyy」-たとえば、2000 年 10 月 19 日は、19-10-2000 と表示されます。

  注: 日付メンバー名フォーマットを変更すると、既存メンバーの名前が無効になる場合があります。たとえば、10-19-2000 というメンバーがあり、フォーマットを dd-mm-yyyy に変更すると、このメンバーは無効になる場合があります。

6 「FALSE メンバー」テキスト・ボックスに、属性メンバー名を入力します。

メンバー名「FALSE」がアウトラインの別の場所で使用されている場合、「FALSE メンバー名」テキスト・ボックスに **Yes**、**Allowed** または別の適切な名前を入力します。デフォルトのメンバー名は、「FALSE」です。

7 「数値」ドロップダウン・リストで、数値範囲を指定します:

- 「範囲の一番上」 - 数値属性の名前が数値範囲の一番上の値に設定されます。範囲が 3,000,001 から 6,000,000 の場合、このオプションを選択すると名前が 6,000,000 として設定されます。
- 「範囲の一番下」 - 数値属性の名前が数値範囲の一番下の値に設定されます。範囲が 3,000,000 から 5,999,999 の場合、このオプションを選択すると名前が 3,000,000 として設定されます。

8 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

次元のプロパティ

次のトピックでは、次元レベルの「Essbase モデル・プロパティ」ダイアログ・ボックスにある各タブで編集する次元プロパティを説明します:

- 「全般」タブ - 226 ページの「次元の全般プロパティの設定」
- 「情報」タブ - 228 ページの「次元の編集」
- 「勘定科目情報」タブ - 234 ページの「会計次元プロパティの編集」
- 「式」タブ」 - 237 ページの「次元への式の追加」
- 「別名」タブ - 238 ページの「次元の別名の表示および編集」
- 「UDA」タブ - 239 ページの「次元へのユーザー定義属性の割当て」
- 「アウトライン構築」タブ - 240 ページの「アウトライン構築オプションの選択」

次元の詳細は、224 ページの「次元の概要」を参照してください。

次元プロパティのタブにアクセスする方法は、226 ページの「「次元プロパティ」タブへのアクセス」を参照してください。

次元の概要

次元は、値の取得または保存用にデータを整理するためのデータ・カテゴリです。次元には通常、内部でグループ化された関連メンバーの階層が含まれています。たとえば、多くの場合、年次元には四半期や月などの期間ごとのメンバーが含まれます。

次元は、ビジネス・プランのコア・コンポーネントを表し、多くの場合はビジネス機能に関連しています。製品、地域、年は代表的な次元です。多くのデータベースでは、次元は静的でアプリケーションの有効期間中に変更されることはほとんどありません。

メンバーは、次元の個別のコンポーネントです。たとえば、製品 A、製品 B および製品 C は、製品次元のメンバーである場合があります。各メンバーには一意の名前があります。次元には、無制限の数のメンバーを含めることができます。

この概要では、次のトピックを説明します:

- 225 ページの「次元タイプ」
- 225 ページの「時間次元の使用ルール」
- 226 ページの「会計次元の使用ルール」

次元プロパティのタブにアクセスする方法は、226 ページの「「次元プロパティ」タブへのアクセス」を参照してください。

次元タイプ

次元を特定のタイプとしてタグ付けすると、次元からそのタイプのためにデザインされた組込み機能にアクセスできます。たとえば、次元を勘定科目タイプとして定義すると、その次元のメンバー・レベルに対して会計操作を指定できます。

次元には次の 3 つのタイプがあります:

- 標準-特別なタイプまたは機能はありません。

  注： 次元を作成するときにタイプを指定しない場合、次元は標準次元として作成されます。

- 時間-データの収集と更新の頻度を記述します。時間としてタグ付けされた次元では、期首のタイム・バランスや期末のタイム・バランスなど、複数の会計次元機能が使用可能になります。

  時間次元の使用のガイドラインは、225 ページの「時間次元の使用ルール」を参照してください。

- 勘定科目-利益や在庫など、メジャーを実行する項目が含まれます。アウトライン内でメジャーとしてタグ付けされた次元は、関連する Essbase データベース内で勘定科目としてタグ付けされた次元に対応します。

  注： 会計次元は、Essbase モデルの作成時に選択されます。モデル内の次元で会計次元にできるのは 1 つのみです。

  会計次元の使用のガイドラインは、226 ページの「会計次元の使用ルール」を参照してください。

時間次元の使用ルール

時間次元にタグを付ける場合は、次のルールに従います:

- 時間のタグを付けることができる次元は 1 つのみです。
- 時間として次元にタグを付ける場合は、次元内のすべてのメンバーは時間プロパティを継承します。

- 時間のタグが付けられた次元内に複数の階層を作成できます。たとえば、同じ時間次元に、年、四半期、月の階層と年、季の階層を指定できます。
- 時間のタグ付きの次元を持たないアウトラインを作成できます。
- 時間のタグが付けられていない次元に時間メンバーを追加できます。
- 時間次元と会計次元は、データベースの他の次元より前に計算されます。
- アウトラインの 2 回目のパスで時間次元のメンバーを計算できます。2 パス計算の概要は、232 ページの「2 パス計算の概要」を参照してください。

会計次元の使用ルール

会計次元にタグを付ける場合は、次のルールに従います:

- 勘定科目としてタグ付けできる次元は 1 つのみです。
- 勘定科目として次元にタグを付ける場合、次元内のすべてのメンバーは勘定科目プロパティを継承します。
- 勘定科目としてタグ付けされていない次元を持つアウトラインを作成できます。このシナリオでは、1 つのメジャーを含む会計次元が作成されます。
- 時間次元と会計次元は、データベースの他の次元より前に計算されます。
- アウトラインの 2 回目のパスで会計次元のメンバーを計算できます。2 パス計算の概要は、232 ページの「2 パス計算の概要」を参照してください。

「次元プロパティ」タブへのアクセス

➤ 「Essbase モデル・プロパティ」ダイアログ・ボックスの「次元プロパティ」タブにアクセスするには:

1 **「メタデータ・ナビゲータ」で、モデル名を右クリックします。**

2 **「Essbase プロパティ」を選択します。**

3 **モデルを展開して次元名を表示します。**

4 **次元を選択します。**

デフォルトで次元の「全般」タブが表示されます。

次元の全般プロパティの設定

「全般」タブでは、わかりやすい名前の定義、コメントの追加、および世代およびレベルの命名など、次元の全般プロパティを設定できます。

次元の詳細は、224 ページの「次元の概要」を参照してください。

➤ Essbase モデルの次元の全般プロパティを設定するには:

1 **226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。**

「名前」テキスト・ボックスには、選択した次元の現在の名前が表示されます。

2 **オプション: キューブの次元のわかりやすい名前を「キューブでの名前」テキスト・ボックスに入力します。**

「ソース要素パス」には、現在選択されている次元が存在する階層要素のパスが表示されます。

注： <、{、[、*やその他のワイルドカード文字はキューブ名に使用できません。 378 ページの「メタデータ要素の命名規則」を参照してください。

3 **オプション: コメントを追加または選択します。**

市場次元のコメントは、米国本土、AK および HI などとなります。データベース列にコメントが存在する場合、「コメント」フィールドには、これらのコメントのドロップダウン・リストが表示されます。

注： 階層要素では、ドロップダウン・リスト・ボックスは使用できません。会計次元のメンバーの場合、会計次元がファクト・テーブルから作成されていると、ドロップダウン・リスト・ボックスは使用できません。

4 **オプション: 世代またはレベルの名前を作成または編集するには、「編集」ボタンをクリックします。**

「編集」ボタンの横に表示されるステータス・メッセージにより、世代またはレベルに名前が付けられたかどうかが示されます。

詳細は、227 ページの「世代およびレベルの命名」を参照してください。

5 **「適用」をクリックします。**

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

世代およびレベルの命名

Essbase モデル内の世代およびレベルには、独自の名前を付けられます。名前は、世代またはレベルを説明する単語または語句です。たとえば、アウトラインのすべての市に対して Cities という世代名を作成します。それぞれの世代またはレベルには、1 つの名前のみ定義できます。

計算スクリプトやレポート・スクリプトで、メンバー名のリスト、あるいは世代番号またはレベル番号のリストを指定する必要がある場合は、世代名やレベル名を使用します。たとえば、計算スクリプトで行う計算の範囲を、特定の世代のメンバーに限定できます。

重複メンバー名を許可する次元では、特定の世代やレベルでメンバー名が一意となるように指定できます。

➤ 世代名またはレベル名を作成、変更または削除する手順は、次のとおりです:

1 **226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、適切な次元を選択し、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。**

2 **次元プロパティの「全般」タブで、「名前付き世代/レベル」の下にある「編集」ボタンをクリックします。**

「名前付き世代およびレベルの編集」ダイアログ・ボックスが表示されます。

3 **世代名またはレベル名を作成するには、次のタスクを実行します:**

1. 「名前付き世代の追加」ボタンまたは「名前付きレベルの追加」ボタンをクリックします。

   世代またはレベルのデフォルトの名前および番号が表示されます。

2. デフォルトの名前を変更するには、「世代名」または「レベル名」のセル内をクリックして、新しい名前を入力します。

3. 同じ行の「番号」列でセル内をクリックして、世代番号またはレベル番号を入力します。

   たとえば、Sample Basic データベースの月の世代に名前を付ける場合、3 を選択します。Sample Basic データベースの月のレベルに名前を付ける場合、1 を選択します。

4. **オプション:** 重複メンバー名が許可された Essbase モデルにおいて、重複メンバー名次元の特定の世代やレベルでメンバー名が一意となるように指定するには、「一意」列のチェック・ボックスを選択します。

5. 作成するすべての名前付き世代またはレベルについて、手順 3.1 から手順 3.5 を繰り返します。

4 **オプション: 追加したアイテムを世代番号順またはレベル番号順に表示するには、「値ごとに名前付き世代を編成」または「値ごとに名前付きレベルを編成」をクリックします。**

5 **名前付き世代またはレベルを変更するには、手順 3.2 から手順 3.5 を実行します。**

6 **名前付き世代またはレベルを削除するには、次のアクションを実行します:**

- 削除する名前付き世代を 1 つ以上選択し、「選択した名前付き世代の削除」をクリックします
- 削除する名前付きレベルを 1 つ以上選択し、「選択した名前付きレベルの削除」をクリックします

7 **「OK」をクリックしてダイアログ・ボックスを終了し、次元プロパティの「全般」タブに戻ります。**

次元の編集

「情報」タブで、次元を環境に適した方法で使用されるように編集できます。次元タイプを定義して、その組込み機能を活用できます。また、次元とそのデータに対して最適な保管方法を選択できます。さらに、計算を定義し、ソート順序を選択することもできます。

次元の詳細は、224 ページの「次元の概要」を参照してください

次のトピックでは、「情報」タブで編集する次元プロパティについて説明します:

- 229 ページの「次元タイプの選択」
- 230 ページの「動的時系列の使用」
- 231 ページの「次元の保管方法の選択」
- 231 ページの「2 パス計算の選択」
- 232 ページの「データの保管方法の選択」
- 233 ページの「解決順の選択」

次元タイプの選択

次元を特定のタイプとしてタグ付けすると、次元はそのタイプのためにデザインされた組込み機能を使用できます。

次元タイプの詳細は、224 ページの「次元の概要」を参照してください。

➤ 次元タイプを選択するには:

1. **226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。**
2. **「情報」タブを選択します。**
3. **「次元タイプ」グループを検索します。**
4. **次のいずれかの「次元タイプ」を選択します:**
   - 「標準」-次元を特定のタイプを持たない次元としてタグ付けします。
   - 「勘定科目」-次元を会計次元としてタグ付けします。この次元タイプには、利益や在庫などメジャーを実行する項目が含まれます。

     会計次元のタグ付けのガイドラインは、226 ページの「会計次元の使用ルール」を参照してください。

     注: 一部のモデルでは、会計次元はモデル作成時に決定され、変更できません。その場合、「勘定科目」オプションは使用不可になります。
   - 「時間」-次元を時間次元としてタグ付けします。この次元タイプは、データの収集と更新の頻度を記述します。時間としてタグ付けされた次元では、期首のタイム・バランスや期末のタイム・バランスなど、複数の会計次元機能が使用可能になります。

     時間次元のタグ付けのガイドラインは、225 ページの「時間次元の使用ルール」を参照してください。

     次元を時間次元としてタグ付けすると、「動的系列」ボタンが使用可能になります。このボタンを選択すると、「動的時系列」ダイアログ・ボックスが開き、ブロック・ストレージの期間累計レポートを使用できるようになります。230 ページの「動的時系列の使用」を参照してください。
5. **「適用」をクリックします。**

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

動的時系列の使用

動的時系列メンバーを使用可能および使用不可にし、世代に関連付けるには、「動的時系列」ダイアログ・ボックスを使用します。8 つの定義済の動的時系列メンバーを使用して、期間累計値を動的に計算できます。動的時系列メンバーには別名も指定できます。

➤ 動的時系列の値を設定するには:

1. 226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。
2. 「情報」タブを選択します。
3. 「次元タイプ」グループで、「時間」を選択します
4. 「動的系列」ボタンをクリックします。
5. 次の定義済の動的時系列のいずれかを選択します。該当する系列の左にあるチェック・ボックスを選択してください:
   - 「H-T-D」 - 履歴初めからの累計
   - 「Q-T-D」 - 当四半期初めからの累計
   - 「Y-T-D」 - 当年初めからの累計
   - 「S-T-D」 - 当季初めからの累計
   - 「P-T-D」 - 当期間初めからの累計
   - 「M-T-D」 - 当月初めからの累計
   - 「W-T-D」 - 当週初めからの累計
   - 「D-T-D」 - 当日初めからの累計
6. 「世代」列内でクリックし、ドロップダウン・リストから世代レベルを選択します。
7. 必要な場合、時系列メンバーに対して 1 つ以上の別名を入力します。

   注： 別名セットの列は、別名セットがモデルに追加された場合にのみ表示されます(170 ページの「別名セットの作成」を参照)。

   注： 動的時系列メンバーに対して別名を入力し、後でモデルから別名セットを削除した場合、ここで入力した動的時系列の別名情報は失われます。
8. 「OK」をクリックします。
9. 「情報」タブで、「適用」をクリックします。

次元の保管方法の選択

次元で選択する保管方法は、その次元が密か疎かにより異なります。

➤ 次元の保管方法を選択するには:

1 226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。

2 「情報」タブを選択します。

3 「次元ストレージ」グループを探します。

4 「次元ストレージ」オプションを選択します:

- 「密」-使用可能なデータ・ポジションが入力される割合が大きい場合、密次元の保管方法を使用します。たとえば、すべての市場のほとんどの製品に時間データが多く存在する場合、年は頻繁に密次元となります。
- 「疎」-使用可能なデータ・ポジションが入力される割合が少ない場合、疎次元の保管方法を使用します。たとえば、製品次元が製品の単位を表していて、市場次元が製品が販売される地域を表している場合、製品と市場の次元は、すべての製品が通常すべての市場で販売されるわけではないため、疎次元となります。
- 「既存」-次元に現在設定されている保管方法です。

5 「適用」をクリックします。

注: このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

2 パス計算の選択

「2 パス計算」は、子の値が親の値または別のメンバーの値に依存する場合に必要です。

「2 パス計算」の概要は、232 ページの「2 パス計算の概要」を参照してください。

➤ 「2 パス計算」を選択するには:

1 226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。

2 「情報」タブを選択します。

3 「2 パス計算」グループを探します。

4 「2 パス計算」を選択します。

5 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

2 パス計算の概要

デフォルトでは、Essbase アウトラインはボトム・アップ形式で計算されます。まず子の値が計算されてから親の値が計算されます。ただし、子の値が親の値や別メンバーの値に依存する場合もあります。

依存している次元およびメンバーの正確な値を得るために、まずアウトラインが計算されてから、別の次元およびメンバーの計算値に依存する次元およびメンバーが再計算されます。Essbase アウトラインの 2 回目のパスで計算される次元およびメンバーは 2 パス計算と呼ばれます。

たとえば、売上高と利益率の比率を計算するには、Essbase は、利益率(売上高などの子メンバーを基本とする親メンバー)を計算する必要があります。新しく計算された利益率の数値に基づいて比率を計算するには、利益率%の比率のメンバーを 2 パス計算としてタグ付けします。Essbase は、データベースを計算してから利益率%のメンバーを再計算します。2 番目の計算によって正しい結果が得られます。

注： 2 パス計算は、任意のメンバーに設定できるプロパティですが、会計次元のメンバー、動的計算のメンバー、動的計算および保管のメンバーでのみ機能します。2 パス計算をその他のメンバーに割り当てても、Essbase では無視されます。

データの保管方法の選択

Essbase で次元のデータ値を保管する方法とタイミングを指定できます。たとえば、ユーザーが値の計算を要求した場合のみ Essbase が値を計算して、その値を破棄するように設定できます。

➤ 次元のデータ保管方法を選択するには:

1 226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。

2 「情報」タブを選択します。

3 「データ・ストレージ」グループを探します。

4 次のいずれかの「データ・ストレージ」方法を選択します:

- 「データの保管」-データ値をメンバーとともに保管します。これがデフォルトです。
- 「動的計算および保管」-ユーザーの要求に応じてデータ値を計算し、その後データ値を保管します。
- 「動的計算」-ユーザーの要求に応じてデータ値を計算し、その後データ値を破棄します。

注： 集約ストレージ・アウトラインで「動的計算および保管」または「動的計算」設定を使用すると、アウトラインの検証およびデータ・ロード中に警告が生成される場合があります。通常、これらの警告はステータス・メッセージで、キューブ配置に失敗した場合を除き無視されます。

- 「共有しない」-メンバーを暗黙的に共有できないようにします。
- 「ラベルのみ」-ナビゲーションに使用するメンバーを作成します。ラベルのみのメンバーにはデータ値が含まれません。

注： 集約ストレージ・アウトラインを使用する場合、「ラベルのみ」オプションを選択する際のガイドラインに注意する必要があります。257 ページの「「ラベルのみ」オプションを持つ集約ストレージ」を参照してください。

- 「既存」-既存のデータ保管方法を使用します。

5 **「適用」をクリックします。**

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

注： 集約ストレージ・アウトラインで「動的計算および保管」または「動的計算」設定を使用すると、アウトラインの検証およびデータ・ロード中に警告が生成される場合があります。通常、これらの警告はステータス・メッセージで、キューブ配置に失敗した場合を除き無視されます。

解決順の選択

「解決順」オプションでは、次元またはメンバーの計算順序を表す番号を指定できます。

「解決順」オプションの使用方法の詳細は、255 ページの「解決順の概要」を参照してください。

➤ 解決順を選択するには:

1 **226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。**

2 **「情報」タブを選択します。**

3 **「解決順」グループを探します。**

4 **次のいずれかの「解決順」オプションを選択します:**

- 「次元の解決順」-次元のすべてのメンバーの計算順序を表す 1 から 127 の番号を指定します。

注： デフォルトの次元の解決順は-1 です。これは、Essbase キューブの現在の解決順が使用されることを示します。

- 「メンバーの解決順」-次元レベルの計算順序を表す 1 から 127 の番号を指定します。

  注： デフォルトのメンバーの解決順は-1 です。これは、Essbase キューブの現在の解決順が使用されることを示します。

5 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

会計次元プロパティの編集

「勘定科目情報」タブでは、会計次元のプロパティを編集して最適なタイム・バランスを選択したり、Essbase による欠落した値またはゼロの処理方法を指定したり、予算データと実績データの差異をレポートする方法を選択したりできます。

注： 「次元プロパティ - 勘定科目情報」タブは、会計次元でのみ使用できます。

次元の詳細は、224 ページの「次元の概要」を参照してください。

次のトピックでは、「勘定科目情報」タブで編集する次元プロパティについて説明します:

- 234 ページの「タイム・バランスの選択」
- 235 ページの「差異レポートの作成方法の選択」
- 236 ページの「スキップ・オプションの選択」

タイム・バランスの選択

タイム・バランスのメソッドを選択して、時間次元で親メンバーの計算方法を決定できます。

タイム・バランス機能の詳細は、235 ページの「タイム・バランスの概要」を参照してください。

➤ タイム・バランスを選択するには:

1 226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。

2 会計次元を選択します。

3 「勘定科目情報」タブを選択します。

4 「タイム・バランス」グループを探します。

5 次のいずれかの「タイム・バランス」オプションを選択します:

- 「なし」-デフォルト値です。タイム・バランスのプロパティをなしに設定すると、Essbase では時間次元の親が通常の方法でロール・アップされます。親の値は、その親の子の式および集計プロパティに基づきます。
- 「期首」-親の値が分岐の最初のメンバーの値を表すようにする場合(多くの場合、期間の始め)、タイム・バランスを期首に設定します。
- 「期末」-親の値が分岐の最後のメンバーの値を表すようにする場合(多くの場合、期間の最後)、タイム・バランスを期末に設定します。
- 「平均」-親の値が子の値の平均を表すようにする場合は、タイム・バランスを平均に設定します。
- 「既存」-既存のタイム・バランスを使用します。

注： タイム・バランスを「なし」以外に設定する場合、スキップ・プロパティを設定して、欠落している値や 0 の値を Essbase が検出した場合の対応を指示する必要があります。236 ページの「スキップ・オプションの選択」を参照してください。

6 **「適用」をクリックします。**

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

タイム・バランスの概要

デフォルトでは、Essbase による時間次元の親の計算は、集計プロパティおよび親の子の式に基づいています。たとえば、デフォルトで年次元の第 1 四半期メンバーが、子である 1 月、2 月および 3 月の合計である場合を考えます。合計ではなくその期間の開始値、終了値または平均値が必要な場合、会計次元の適切なメンバーの適切なタイム・バランス・プロパティを選択できます。会計次元のメンバーにタイム・バランス・プロパティを設定すると、プロパティは時間次元の勘定科目メンバーの計算方法に影響します。

たとえば、第 1 四半期の期首在庫を第 1 四半期の在庫開始値に反映させる場合、勘定科目メンバーである期首棚卸高の最初のタイム・バランス・プロパティを設定します。期首棚卸高が年次元で計算されると、第 1 四半期は 1 月(第 1 四半期階層の最初の子)に等しくなります。

差異レポートの作成方法の選択

差異レポート・プロパティによって、Essbase が、メンバー式に@VAR または @VARPER 関数があるメンバーの実績データと予算データの差異を計算する方法が決まります。会社に支出を示すメンバーの場合、支出のプロパティが必要になります。

➤ 差異レポートの作成方法を選択するには:

1 226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。

2 モデルの会計次元を選択します。

3 「勘定科目情報」タブを選択します。

4 「差異レポート」グループを探します。

5 次のいずれかの「差異レポート」の作成方法を選択します:

- 「支出なし」-売上などの支出なし項目の場合、実績は予算よりも大きくなります。実績が予算より小さい場合、差異は負数になります。@VAR 関数は、実績から予算を減算します。たとえば、売上予算が$100 で、売上実績が$110 の場合、差異は 10 になります。デフォルトでは、メンバーは支出なしになっています。
- 「支出」-支出項目の場合、支出実績は支出予算より小さくなります。支出実績が支出予算より大きい場合、差異は負数になります。@VAR 関数は、予算から実績を減算します。たとえば、支出予算が$100 で、支出実績が$110 の場合、差異は-10 になります。

6 「適用」をクリックします。

注: このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

スキップ・オプションの選択

タイム・バランスを期首、期末または平均として設定する場合、スキップ・プロパティを設定して、欠落している値や 0 の値が検出された場合の対応を Essbase に指示する必要があります。

➤ スキップ・オプションを選択するには:

1 226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。

2 会計次元を選択します。

3 「勘定科目情報」タブを選択します。

4 「スキップ」グループを探します。

5 次のいずれかの「スキップ」オプションを選択します:

- 「なし」-親の値の計算時にデータをスキップしません。このプロパティはデフォルト値です。ただし、Essbase で平均の計算時に#MISSING データが検出された場合、メンバーの合計数では除算されません。実績値を持つメンバーの数で除算が行われます。したがって、スキップ・プロパティを「なし」に設定しないと、#MISSING が平均値に影響しなくなります(最初と最後は影響する)。

- 「欠落」-親の値の計算時に#MISSING データをスキップします。
- 「ゼロ」-親の値の計算時にゼロに相当するデータをスキップします。
- 「欠落およびゼロ」-#MISSING データとゼロに相当するデータを両方とも親の値の計算時にスキップします。

6 **「適用」をクリックします。**

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

次元への式の追加

式により、Essbase サーバーが Essbase データベースのメンバー間の関係を計算する方法が決定されます。

式の詳細は、237 ページの「式について」を参照してください。

次元の詳細は、224 ページの「次元の概要」を参照してください。

➤ 次元に式を追加するには:

1 **226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。**

2 **「式」タブを選択します。**

3 **「式」テキスト・ボックスに式を入力します。**

次の式の制限に注意してください:

- 式の長さは最大で 64,000 文字までです。
- ASO(集約ストレージ)モデル内の式は、MDX フォーマットである必要があります。
- BSO(ブロック・ストレージ)モデル内の式は、標準の Essbase 計算スクリプト・フォーマットである必要があります。

注： 式は、モデルが配置されるまで確認されません。

4 **「適用」をクリックします。**

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

式について

式は、演算子、関数、次元名、メンバー名および数値定数の組合せです。式は、次元、メンバー、ユーザー定義メンバーおよびメジャーと関連付けられます。

式の操作に関するガイドラインを次に示します:

- ブロック・ストレージ・モデル内の式は、標準の Essbase 計算スクリプト・フォーマットである必要があります。
- 式では、"Root Beer"や"Cream Soda"のようにメンバー名は引用符で囲まれていて、ソースによって処理が異なります。
  - 「Essbase モデル・プロパティ」の「式」テキスト・ボックスに式を直接入力する場合は、次の書式を使用します:

    ```
    ""Root Beer"+"Cream Soda";"
    ```

  - 「Essbase モデル・プロパティ」の「外部ソースから」フィールドの外部ソース列を参照する場合は、必ずリレーショナル・ソースでそれぞれの引用符の前に円記号(バックスラッシュ)(\)を入力します; 次に例を示します:

    ```
    \"Root Beer\"+\"Cream Soda\"
    ```

  - Performance Management Architect で構築された BSO キューブを使用している場合は、Performance Management Architect の「メンバー式 BSO」フィールドで、式全体を引用符で囲みます; 次に例を示します:

    ```
    ""Root Beer"+"Cream Soda""
    ```

  書式に関するこれらのガイドラインに従わない場合は、Root Beer"+"Cream Soda のように、Essbase によって開始と閉じの引用符が削除され式が無効になります。
- 集約ストレージ・モデルを操作する場合:
  - 式は MDX フォーマットで表す必要があります。たとえば、(PROFIT*100)/SALES のように適切な MDX 構文を使用します。MDX 式では、%演算子はサポートされていません。
  - 動的階層内のメンバーの式のみを指定できます。

式の操作の詳細は、Oracle Essbase Database Administrator's Guide を参照してください。

次元の別名の表示および編集

「別名」 タブで、次元の現在の別名のリストを表示できます。

別名および別名セットに関するその他の情報は、次のトピックを参照してください:

- 266 ページの「別名の概要」。
- 170 ページの「別名セットの作成」。
- 224 ページの「次元の概要」。

➤ 次元に割り当てられた別名セットを表示して編集するには:

1 **226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスの次元プロパティのタブにアクセスします。**

2 **「別名」タブを選択します。**

次元に割り当てられている別名のリストがあれば表示されます。「別名セット」列に表示されている名前が、Essbase モデルの「別名」タブで指定されている「キューブでの名前」にマップされます。

3 **オプション:「ソース・マッピング」列内のエントリを編集して、選択した次元の次元別名を更新します。**

たとえば、Product という名前の次元にフランス語の別名を使用する場合、「ソース・マッピング」列に Produit と入力します。

4 **別名ソース・マッピングを変更する各次元に対して、手順 3 を繰り返します。**

注: 別名の割当てを変更するには、217 ページの「Essbase モデルの別名セットの選択」を参照してください。

次元別名は、接頭辞または接尾辞を追加することによって編集できます。これを行う手順は、メンバー別名に接頭辞または接尾辞を追加する手順と同じです。

➤ 次元別名に接頭辞を追加するには、271 ページの「別名への接頭辞の追加」の手順に従ってください。

➤ 次元別名に接尾辞を追加するには、272 ページの「別名への接尾辞の追加」の手順に従ってください。

次元へのユーザー定義属性の割当て

ユーザー定義属性(UDA)を次元に割り当てて次元の特徴を説明できます。たとえば、大規模市場という UDA を作成して、売上が一定額を超える市場を識別できるようにします:

240 ページの「UDA の概要」を参照してください。

224 ページの「次元の概要」を参照してください。

➤ UDA を次元に割り当てるには:

1 **226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。**

2 **「UDA」タブを選択します。**

3 **「UDA 値」テキスト・ボックスで、次元に使用する新しい UDA を入力します。**

現在次元に割り当てられている UDA が「既存 UDA」テキスト・ボックスに表示されます。

4 「リストに追加」をクリックします。

UDA が「既存 UDA」ボックスに表示されます。

注： 作成済の UDA も「既存 UDA」ボックスに表示されます。

5 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

UDA の概要

次元とメンバーに対して独自のユーザー定義属性(UDA)を作成できます。UDA は、語または句です。たとえば、大規模市場という UDA を作成して、一定額以上が売れる市場を識別できます。UDA は、計算スクリプトとレポート・スクリプトで使用されます。

メンバーまたは次元に対して UDA を作成するときには、次のガイドラインを考慮してください:

- 1 つのメンバーまたは次元に複数の UDA を定義できます。
- 同じメンバーまたは次元に同じ UDA を 2 回定義することはできません。
- 異なるメンバーおよび次元に同じ UDA を定義できます。
- UDA では、同じ名前をメンバー、別名、レベルまたは世代として使用できます。UDA に名前を付けるときには、付録 B「Essbase Studio の命名規則」の命名ルールに従ってください。
- Essbase 共有メンバーに対して UDA を作成することはできません。
- UDA は指定したメンバーにのみ適用されます。メンバーの子孫や祖先が自動的に UDA を受け取ることはありません。

アウトライン構築オプションの選択

Essbase アウトラインを構築するときに、「アウトライン構築」タブを選択して、代替階層のストレージ・プロパティを指定します。また、「アウトライン構築」タブを使用して、Essbase アウトライン内の重複するメンバー設定を移動することもできます。

階層の詳細は、177 ページの「階層について」を参照してください。

次元の詳細は、224 ページの「次元の概要」を参照してください。

次のトピックでは、「アウトライン構築」タブで編集する次元プロパティを説明します:

- 241 ページの「階層ストレージ設定の選択」
- 242 ページの「重複するメンバー設定の移動」
- 242 ページの「共有メンバーの前に実メンバーを配置」

- 243 ページの「データ・ロードの最適化」

階層ストレージ設定の選択

Essbase アウトラインの構築で、階層と代替階層のストレージ・プロパティを指定できます。

➤ 階層のストレージ・オプションを指定するには:

1 **226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。**

2 **「アウトライン構築」タブを選択します。**

3 **「階層設定」グループで、オプションを選択します:**

- 「次元レベルで保管済」-次元を保管階層として設定します。これがデフォルトです。
- 「次元レベルで動的」–次元を動的階層として設定します。これが選択されている場合は、「圧縮次元として作成」チェック・ボックスが有効になります。

  オプション:

  「圧縮次元として作成」チェック・ボックス - 集約ストレージ・データベースを最適化します。

  これが選択されている場合、データは次元内で (キー，複数値) としてグループ化されます。

  このチェック・ボックスが選択されていない場合、データは (キー，単一値)，(キー，単一値) として格納されます。

  ヒント: アカウント次元を圧縮することにより、集約ストレージ・データベースを最適化することをお薦めします。

  圧縮次元の詳細は、Oracle Essbase Database Administrator's Guide の集約ストレージ・データベースの圧縮次元に関する項を参照してください。
- 「複合階層使用可能」-次元で保管階層と動的階層の両方を使用します。

  注: データの集計とナビゲーションの方法は階層によって決定されるため、世代 2 のメンバーの集約ストレージ設定がある場合は、「メンバー情報」タブへのリンクをクリックして、集約ストレージ設定を確認する必要があります。

4 **「適用」をクリックします。**

注: このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

重複するメンバー設定の移動

Essbase アウトラインの重複するメンバー設定を移動できます。

➤ 重複するメンバー設定を移動するには:

1 226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。

2 「アウトライン構築」タブを選択します。

3 「重複するメンバー設定を移動」グループで、オプションを選択します:

- 「重複するメンバーを移動しない」 - 重複メンバーは Essbase アウトラインに移動されず、共有メンバーが追加されます。
- 「重複するメンバー(およびその子孫)を移動」 - 重複するメンバーとその子孫が Essbase アウトラインに移動されます。
- 「重複するメンバーを無視し、移動しない」 - 重複メンバーは Essbase アウトラインに移動されず、共有メンバーも追加されません。
- 「Gen2 の重複のみ移動」 - 世代 2 のメンバーとその子孫のみが Essbase アウトラインに移動されます
- 「非 Gen2 の重複のみ移動」 - 世代 2 のメンバーは無視され、Essbase アウトラインに移動されません。

4 「適用」をクリックします。

注: このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

共有メンバーの前に実メンバーを配置

集約ストレージ・アウトラインの場合、Essbase では、実メンバーが常に共有メンバーの前に配置されている必要があります。Essbase モデルを作成する際に、再帰的な次元で実メンバーの前に共有メンバーを配置している可能性があります。

「Essbase モデル・プロパティ」の「アウトライン構築」タブにあるチェック・ボックスを選択すると、実メンバーの前にある共有メンバーのすべてのインスタンスの位置が逆になります。

➤ Essbase モデルに、実メンバーの前に配置されている共有メンバーが含まれる場合は、次のチェック・ボックスを選択します:

「共有メンバーが実メンバーの前にある場合、共有メンバーと実メンバーの位置を逆にする」

次の例は、Essbase により共有メンバーと実メンバーの順序が修正される前および修正された後の再帰階層のメンバーを示しています。親の順序は同じままで、実メンバーの位置が変更されていることに注意してください。

ケース 1、修正前

```
 Employees (dimension)
Engineer
  John Smith (shared)
Manager
  John Smith (shared)
Director
  John Smith (actual)
```

ケース 1、修正後

```
 Employees (dimension)
Engineer
  John Smith (actual)
Manager
  John Smith (shared)
Director
  John Smith (shared)
```

ケース 2、修正前

```
 Employees (dimension)
Engineer
  John Smith (shared)
  Paul Williams (shared)
Manager
  John Smith (actual)
  Paul Williams (actual)
```

ケース 2、修正後

```
 Employees (dimension)
Engineer
  John Smith (actual)
  Paul Williams (actual)
Manager
  John Smith (shared)
  Paul Williams (shared)
```

データ・ロードの最適化

データ・ロードを最適化できます。最適化すると、結合が無視され、かわりにファクト・テーブルの列がデータ・ソースのバインディングとして使用されます。

注： データ・ロードを最適化する場合、別名テーブルとして選択したテーブルへの影響を考慮する必要があります。217 ページの「Essbase モデルの別名セットの選択」を参照してください。

➤ データ・ロードを最適化するには:

1 226 ページの「「次元プロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにある次元プロパティのタブにアクセスします。

2 「アウトライン構築」タブを選択します。

3 「最適化」データ・ロード・グループを探します。

4 「データ・ロードを最適化」を選択します。

注: キューブ・スキーマ・ウィザードでデータ・ロードのバインディングが上書きされている場合、このオプションは無視されます。

次元の順序付け

次元のソート順によって、次元が Essbase モデルや後続の Essbase アウトラインに表示される順序が決定され、次元の計算順序にも影響します。集約ストレージ・データベースとブロック・ストレージ・データベースの計算順序は、データベースの設計時にあらかじめ決定しておく必要があります。ブロック・ストレージ・データベースでは特に、次元の順序がパフォーマンスに影響することがあります。集約ストレージ・データベースおよびブロック・ストレージ・データベースの計算については、Oracle Essbase Database Administrator's Guide で説明されています。ただし、次の点に注意してください:

- 集約ストレージ・データベース - 次元の解決順序は、集約ストレージ・データベースによって使用され、次元のソート順を設定する際に考慮する必要があります。次元の解決順序またはソート順が設定されていない場合は、アウトラインの順序で集約が実行されます。属性は最後に計算されます。
- ブロック・ストレージ・データベース - 次元のソート順は、Essbase に固有の計算順序に基づきます。密次元(たとえば、時間次元やメジャー次元)が最初に計算され、疎次元が次に計算され、属性次元が最後に計算されます。

次元ソート順には、Essbase モデル・ツールバーのボタンを使用します。

図 15　次元ソート順のツールバー・ボタン

➤ Essbase モデル内で次元をソートするには:

1 **移動する次元を選択し、上矢印または下矢印をクリックして、モデル内の次元の位置を移動します。**

- モデル内で次元を上に移動するには、「選択した次元をリストで上へ移動([Alt] + [Shift] + [↑])」をクリックします。

  または、キーボード・ショートカット([Alt]+[Shift]+[↑]キー)を使用します。

- モデル内で次元を下に移動するには、「選択した次元をリストで下へ移動([Alt] + [Shift] + [↓])」をクリックします。

または、キーボード・ショートカット([Alt]+[Shift]+[↓]キー)を使用します。

2 **次元順序の変更を保存するには、「次元順序の変更をモデルに適用します」をクリックします。**

3 **オプション: 現在のセッション中に行った次元順序の変更を取り消すには、「次元順序の変更を取り消します」をクリックします。**

メンバーのプロパティ

次のトピックでは、メンバー・レベルの「Essbase モデル・プロパティ」ダイアログ・ボックスの各タブで編集するメンバーのプロパティを説明します:

- 「全般」タブ - 246 ページの「メンバーの全般プロパティの設定」
- 「情報」タブ - 253 ページの「メンバー情報の編集」
- 「勘定科目情報」タブ - 260 ページの「会計次元のメンバーの編集」
- 「式」タブ」 - 264 ページの「メンバーへの式の追加」
- 「別名」タブ - 265 ページの「メンバーの別名の編集」
- 「UDA」タブ - 273 ページの「メンバーへのユーザー定義属性の割当て」
- 「変換」タブ - 274 ページの「メンバーの変換」

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

「メンバーのプロパティ」タブにアクセスするには、246 ページの「「メンバーのプロパティ」タブへのアクセス」を参照してください。

メンバーの概要

次元は、ビジネス・プランのコア・コンポーネントを表し、多くの場合はビジネス機能に関連しています。製品、地域、年は代表的な次元です。多くのデータベースでは、次元は静的でアプリケーションの有効期間中に変更されることはほとんどありません。

メンバーは、次元の個別のコンポーネントです。たとえば、製品 A、製品 B および製品 C は、製品次元のメンバーである場合があります。各メンバーには一意の名前があります。次元には、無制限の数のメンバーを含めることができます。

一部の次元では、アプリケーションの有効期間内にメンバーが頻繁に変更されます。また、メンバーが、あるメンバーの親であると同時に別のメンバーの子となる場合もあります。Essbase アウトラインでは、メンバーをお互いの下にインデントして連結関係を示します。たとえば、製品次元の総売上を製品の記述から合計し、製品コードで分類し、さらに製品 ID で分類します。

「メンバーのプロパティ」タブにアクセスするには、246 ページの「「メンバーのプロパティ」タブへのアクセス」を参照してください。

「メンバーのプロパティ」タブへのアクセス

➤ 「Essbase モデル・プロパティ」ダイアログ・ボックスの「メンバーのプロパティ」タブにアクセスするには:

1 **「メタデータ・ナビゲータ」で、モデル名を右クリックします。**

2 **「Essbase プロパティ」を選択します。**

3 **メンバーの名前を表示するモデルを展開します。**

4 **メンバーを選択します。**

デフォルトでは、メンバーの「全般」タブが表示されます。

メンバーの全般プロパティの設定

「全般」タブで、メンバーの全般プロパティを設定できます。さらに、可変属性を含む、メンバーの属性も設定できます。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

次のトピックでは、「全般」タブで編集するメンバーのプロパティを説明します:

- 246 ページの「メンバーの全般プロパティの指定」
- 247 ページの「属性としてのメンバーの選択」
- 249 ページの「メンバーの可変属性の設定」

メンバーの全般プロパティの指定

➤ Essbase モデルのメンバーの全般プロパティを指定するには:

1 **246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。**

「名前」テキスト・ボックスに、選択したメンバーの名前が表示されます。

「ソース要素パス」テキスト・ボックスに、現在選択されているメンバーのパスが表示されます。

「データ・ソース・バインディング」テキスト・ボックスに、論理メンバーが関連付けられている物理要素が表示されます。

2 **オプション:「コメント」グループからオプションを選択します:**

- 「コメント」 - 選択したメンバーに関するコメントを入力します。
- 「外部ソースから」 - ドロップダウン・リストから、コメントを含む当該データベース列を選択します。これらはキューブ配置時にメンバーとともにロードされます。

注： 会計次元のメンバーの場合、会計次元がファクト・テーブルから作成されていると、ドロップダウン・リスト・ボックスは使用できません。

メンバーごとに前述のオプションのいずれかを選択できます。

3 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

属性としてのメンバーの選択

メンバーを属性として選択したり、属性のタイプを指定したりできます。

属性とその使用方法の詳細は、218 ページの「属性の概要」を参照してください。

➤ メンバーを属性として選択するには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。

2 「属性の設定」で、「(メンバー名)の Essbase 属性」チェック・ボックスを選択します。

メンバーが置かれている次元の名前が、「属性次元名」テキスト・ボックスに表示されます。

3 ドロップダウン・リストから、次のいずれかの属性タイプを選択します:

- 「数値」
- 「ブール」
- 「文字列」
- 「日付/時間」

4 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

可変属性の履歴テーブルの設定

メンバーの可変属性を設定できます。可変属性を使用する際には、履歴テーブルと呼ばれるテーブルを最初に用意する必要があります。履歴テーブルの設定が終わると、249 ページの「メンバーの可変属性の設定」の手順を実行する用意ができます。

履歴テーブルには、可変属性メンバー用の 1 つの列(重複メンバーをサポートするアウトライン用には複数の列)と、独立次元のメンバーを特定するために Essbase で使用できる値を示す複数の列が含まれます。

例:

Product Table

```
| ProductID | SKU |
Other product columns
 |
```

Fact Table
(can also be used to build the Time Dimension)

```
| ProductID | Transdate |
Other fact table columns
 |
```

Product Package Table
(history table for varying attributes [VA])

```
  | ProductID | Pkg Type (VA) | From Date | To Date |

  | 1     | Plastic Bag   | Jan    | Jul   |

  | 1     | Paper Bag    | Aug    | Dec   |
```

上の例では、"Transdate"列が含まれるファクト・テーブルが示されており、この列は Essbase Studio で時間次元の構築に使用できます。

ただし、ほとんどの Essbase Studio ユーザーはファクト・テーブルから時間次元を構築しません。実際のところ、ユーザーが直接ファクト・テーブルから次元を構築することは推奨されていません。したがって、テーブル・スキーマは次のようになります:

Employee Table

```
| EmpID | EmpName | SS# |
```

```
Time Table
 (to be used to build the Time dimension


|
Columns for Date Time...
 | TimeID |


Fact Table
 (can also be used to build Time dimension)


| EmpID | Hire Date (TimeID) |
Other fact table columns
 |


Position Table
 (history table for varying attributes)


  | EmpID | Position (VA) | From Date | To Date |
```

"Position"テーブルには Employee テーブルおよび Time テーブルへの結合があります。

"Position"次元の次元構築クエリーには Time テーブルが関係します。このテーブルはファクト・テーブルよりはるかに小さいため、構築上のパフォーマンスに与える影響は少なくなります。

"To"列が空の場合、"From Date"列の値は TimeID 列になる場合があります。

```
Position Table
 (the history table for varying attributes)


  | EmpID | Position (VA) | From Date (TimeID) |
```

メンバーの可変属性の設定

メンバーの可変属性を設定できます。可変属性を設定する前に、247 ページの「可変属性の履歴テーブルの設定」の説明に従って履歴テーブルを設定することをお薦めします。

可変属性とその使用方法の詳細は、219 ページの「可変属性の概要」を参照してください。

注： カレンダ階層では、可変属性はサポートされません。

➤ 可変属性としてメンバーを設定するには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。

2 「属性の設定」で、「(メンバー名)の Essbase 属性」チェック・ボックスを選択します。

メンバーが置かれている次元の名前が、「属性次元名」テキスト・ボックスに表示されます。

デフォルトでは、メンバーの可変属性は使用できません。

3 ドロップダウン・リストから、次の「属性タイプ」を選択します:

- 数値
- ブール
- 文字列
- 日付/時間

注： デフォルトの属性タイプは文字列です。

4 「可変属性設定」グループで、「編集」ボタンをクリックします。

「可変属性の編集」ダイアログ・ボックスが表示されます。

5 「可変属性の作成」チェック・ボックスを選択します。

6 次のいずれかの「関連付けモード」を選択します:

- 「既存のものを維持」-属性で重複する範囲がある場合、元の範囲を使用して可変属性の値を決定します。
- 「上書き」-属性で重複する範囲がある場合、「上書き」オプションでは、元の範囲が後の範囲で上書きされます。

7 「独立次元設定」グループを探します。

8 「次元」列で、次元名の左にあるチェック・ボックスを選択して、次元を選択します。

注： 赤い円で囲まれた小さな白い X が「状態」列に表示されます。これは、次元を可変属性の機能で使用できるようにするには、いくつかの手順を実行する必要があることを示しています。カーソルを X の上に置くと、ポップアップ・メッセージで必要な手順が表示されます。必要のない手順を実行すると、X は小さな黄色い三角形で囲まれた感嘆符に置き換わります。

9 「リーフ・レベル」列で、次元のセルをハイライトして、ドロップダウン・リストから、可変属性の値が依存する範囲を含むメンバーを選択します。

注： 1 つの可変属性に対して 2 つの独立次元が同じリーフ・メンバー名を使用している場合、配置は失敗します。

たとえば、可変属性次元 VAYRPER と 2 つの独立次元 Period と Year があるとします。Period と Year のリーフ・メンバー名が同じ場合、配置は失敗します。

このエラーを回避するには、独立次元を表す少なくとも 1 つの階層で、リーフ・メンバーとして使用される次元要素の名前を変更します。その後、配置を再試行します。

10 **「タイプ」列で、次元のセルをハイライトして、ドロップダウン・リストから、可変属性の値が「範囲」か「個別」かを選択します:**

- 「範囲」 - 可変属性の値は、値の範囲に影響します; たとえば、クライアントの営業担当者が 1 年の中ごろで変わる場合、可変属性の値は 1 月から 12 月をカバーします。
- 「個別」 - 可変属性の値は、個別の値に影響します; たとえば、カンザス州のクライアントの営業担当者と、オクラホマ州のクライアントの営業担当者が異なる場合、可変属性はオクラホマ州のみをカバーします。

注： 「範囲」タイプは、時間範囲に使用します。

11 **「開始」列で、次元のセルをクリックして、 ボタンをクリックします。**

注： 独立次元のバインディングを定義する式は共通プラットフォーム言語 (Common Platform Language: CPL)で作成されます。この式は、言語で定義する構文に従ったオペランドと演算子のシーケンスです。各式では値が戻され、その値の型によって式の型が定義されます。付録 C「CPL リファレンス」を参照してください。

「独立次元バインディング」ダイアログ・ボックスが表示されます。ここで、独立次元のバインディングを定義したり、フィルタを追加して定義を絞り込んだりします。

注： 使用しやすいように、「独立次元バインディング」ダイアログ・ボックスをサイズ変更できます。

12 **「独立次元バインディング」ダイアログ・ボックスで、次元のバインディングを定義する式を作成します:**

注： 式の作成に関する制限は、371 ページの「独立次元のバインディングに関する制限」を参照してください。

1. ダイアログ・ボックスの左下にある「ソース」タブを選択します。
2. メンバーを表示する次元を展開します。
3. メンバーを選択します。
4. 右矢印を使用してメンバーの接続文字列を「式」テキスト・ボックスに移動します。

5. 「関数」タブを選択します。
6. SQL レベルを展開して関数の型を表示します。
7. 関数の型を展開して関数を表示します。
8. 関数を選択します。
9. 右矢印を使用して関数文字列を「式」テキスト・ボックスに移動します。
10. 「演算子」タブを選択します。
11. 演算子の型を展開して演算子を表示します。
12. 演算子を選択します。
13. 右矢印を使用して演算子を「式」テキスト・ボックスに移動します。

13 オプション: フィルタを追加する式を作成して、バインディング定義を絞り込みます:

1. ダイアログ・ボックスの左下にある「ソース」タブを選択します。
2. メンバーを表示する次元を展開します。
3. メンバーを選択します。
4. 右矢印を使用してメンバーの接続文字列を「フィルタ」テキスト・ボックスに移動します。
5. 「関数」タブを選択します。
6. SQL レベルを展開して関数の型を表示します。
7. 関数の型を展開して関数を表示します。
8. 関数を選択します。
9. 右矢印を使用して関数文字列を「フィルタ」テキスト・ボックスに移動します。
10. 「演算子」タブを選択します。
11. 演算子の型を展開して演算子を表示します。
12. 演算子を選択します。
13. 右矢印を使用して演算子を「フィルタ」テキスト・ボックスに移動します。

14 「OK」をクリックします。

15 オプション: 「終了」列で、次元のセルをクリックして、表示されるボタンをクリックします。

「独立次元バインディング」ダイアログ・ボックスが表示されます。ここで、引き続き独立次元のバインディングを定義したり、フィルタを追加してバインディングを絞り込んだりします。

注: 使いやすいように、「独立次元バインディング」ダイアログ・ボックスを拡大できます。

16 必要に応じて、手順 12 から手順 15 までを繰り返します。

17 「可変属性の編集」ダイアログ・ボックスで、「OK」をクリックします。

メンバー情報の編集

「情報」タブで、環境に適した方法でデータが計算されるようにメンバーを編集できます。子から親にメンバーのデータをロール・アップする方法を指定したり、計算の解決順を選択したりできます。また、2 パス計算を選択し、親メンバーの値によって子メンバーの値を決定することもできます。

「情報」タブで、データの保管方法を選択して、Essbase によってメンバーのデータ値が保管される方法やタイミングを決定できます。また、集約ストレージ・オプションを選択することもできます。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

次のトピックでは、「情報」タブで編集するメンバー・プロパティについて説明します:

- 253 ページの「子メンバーの集計メソッドの選択」
- 255 ページの「メンバーを計算するときの解決順の選択」。
- 256 ページの「2 パス計算オプションの選択」
- 256 ページの「メンバー・データの保管方法の選択」
- 258 ページの「集約ストレージ・オプションの選択」

子メンバーの集計メソッドの選択

計算時の子メンバーから親へのロール・アップ方法を決定するには、集計メソッドを選択します。

集計オプションの詳細は、254 ページの「子メンバーの集計の概要」を参照してください。

➤ 集計メソッドを選択するには:

1. **246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。**
2. **「情報」タブを選択します。**
3. **次のいずれかの「集計」方法を選択します:**
   - 「+ [加算]」-他のメンバーで実行された計算結果にメンバーを追加します。有効な値が他にない場合、この演算子がデフォルトの演算子になります。
   - 「- [減算]」-メンバーを-1 で乗算してから、その結果を他のメンバーで実行された計算の合計に追加します。
   - 「* [乗算]」-他のメンバーで実行された計算結果でメンバーを乗算します。
   - 「/ [除算]」-以前に他のメンバーで実行された計算結果でメンバーを除算します。
   - 「% [パーセント]」-以前に他のメンバーで実行された計算の合計でメンバーを除算し、その結果に 100 を乗算してパーセント値にします。
   - 「~ [無視]」-メンバーは、その親の集計には使用されません。

- 「^ [なし]」 -メンバーは、どの集計にも使用されません。
- 「外部ソース」 -ドロップダウン・リストで、集計演算子が保管されているデータベースの列を選択します。データ・ソースの選択した列の集計値(+、-、*、/、%、~)が集計に使用されます。

注： 外部データ・ソースで使用するメンバー・コードについては、Oracle Essbase Database Administrator's Guide のメンバー・プロパティを操作するためのデータ・ソースの使用、および集約ストレージの次元構築に対するデータ・ソースの違いの章を参照してください。

4 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

子メンバーの集計の概要

メンバーの集計演算子は、親への子のロール・アップ方法を決定します。デフォルトで、新しいメンバーには、加算(+)演算子が与えられます。これは、Essbase がメンバーを加算するという意味です。たとえば、Essbase は、1 月、2 月および 3 月の数値を加算して、結果をその親である第 1 四半期に保管します。

重要: 集約ストレージ・アウトラインを処理する場合、次の集計プロパティのガイドラインが適用されます:

- 勘定科目のタグが付いた次元には、すべての集計演算子を使用できます。
- その他の次元には、(+)集計演算子のみを使用できます。

  また、次元内のメンバーに「ラベルのみ」プロパティを使用している場合、最下位の「ラベルのみ」メンバーの直接の子にはどの集計演算子も使用できます; ただし、最下位の「ラベルのみ」メンバーの間接の子にはすべて、(+)集計プロパティでタグを付ける必要があります。

注意:

- 集約ストレージ・アウトラインを処理する場合、「外部ソース」集計オプションからデータベースの列を慎重に選択してください。ロードの実行時、Essbase Studio サーバーは、データベース列に保管されている集計演算子を認識しない場合があります。検証中にエラーの可能性があることはサーバーから通知されません。

  注： 外部データ・ソースで使用するメンバー・コードについては、Oracle Essbase Database Administrator's Guide のメンバー・プロパティを操作するためのデータ・ソースの使用、および集約ストレージの次元構築に対するデータ・ソースの違いの章を参照してください。
- 会計次元のメンバーの場合、その会計次元がファクト・テーブルから作成されたものであれば、「元のデータベース列」ドロップダウン・リスト・ボックスは表示されません。

メンバーを計算するときの解決順の選択

解決順によって、次元のメンバーが評価される順序が決まります。

「解決順」オプションの使用方法の詳細は、255 ページの「解決順の概要」を参照してください。

➤ 解決順を選択するには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。

2 「情報」タブを選択します。

3 「解決順」グループを探します。

4 次のいずれかの「解決順」オプションを選択します:

- 「メンバーの解決順」-テキスト・ボックスで、動的階層の評価順序を表す 1 から 127 の番号を指定します。デフォルトは 0 です。
- 「外部ソース」-ドロップダウン・リストで、メンバーの評価順序を含むデータベースの列を選択します。

  注： 「元のデータベース列」ドロップダウン・リスト・ボックスは、ファクト・テーブルに基づくユーザー定義次元または会計次元のメンバーには使用できません。

5 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

解決順の概要

解決順番号が同じメンバーは、データベース・アウトラインに表示される次元の順序で評価されます。解決順番号がないメンバーは、解決順番号の付いたメンバーがすべて評価された後に評価されます。解決順の番号が一番大きいメンバーが最初に評価されます(たとえば、解決順番号が 20 の式は、解決順番号が 5 の式よりも先に評価されます)。

解決順番号が同じメンバーは、データベース・アウトラインに表示される次元の順序で評価されます。解決順番号がないメンバーは、解決順番号の付いたメンバーがすべて評価された後に評価されます。

注： デフォルトの値は 0 です。解決順が 0 のメンバーは、そのメンバーの次元の解決順を継承します。

「次元の解決順」テキスト・ボックスで、次元で評価されるメンバーのデフォルトの順序を示す番号を指定します。指定できる解決順番号は 1 から 127 です。デフォルトの値は 0 です。「メンバーの解決順」テキスト・ボックスで、次元メンバーの

解決順序を指定します。たとえば、市場次元では、市場が次元メンバーになります。

2 パス計算オプションの選択

「2 パス計算」は、子の値が親の値または別のメンバーの値に依存する場合に必要です。

2 パス計算の使用の詳細は、232 ページの「2 パス計算の概要」を参照してください。

➤ 2 パス計算を選択するには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。

2 「情報」タブを選択します。

3 ダイアログ・ボックスの右上の、「2 パス計算」グループを見つけます。

4 次のいずれかの「2 パス計算」オプションを選択します:

- 「なし」-現在選択しているメンバーで、2 パス計算を使用しません。
- 「2 パス計算」-現在選択しているメンバーで、2 パス計算を使用します。
- 「外部ソース」-ドロップダウン・リストで、データベース内の 2 パス計算を含む列を選択します。

5 「適用」をクリックします。

注: このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

メンバー・データの保管方法の選択

データの保管方法を選択して、Essbase によってメンバーのデータ値が保管される方法やタイミングを決定できます。たとえば、ユーザーが要求した場合にのみメンバーの値を計算し、計算後はその値を破棄するように Essbase を設定できます。

➤ メンバー・データの保管方法を選択するには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。

2 「情報」タブを選択します。

3 「データ・ストレージ」メソッド・グループを探します。

4 次のいずれかの「データ・ストレージ」オプションを選択します:

- 「データの保管」-データ値をメンバーとともに保管します。これがデフォルトです。

- 「動的計算および保管」-ユーザーの要求に応じてデータ値を計算し、その後データ値を保管します。
- 「動的計算」-ユーザーの要求に応じてデータ値を計算し、その後データ値を破棄します。
- 「共有しない」-メンバーは共有できません。
- 「ラベルのみ」-ナビゲーションに使用するメンバーを作成します。ラベルのみのメンバーにはデータ値が含まれません。

  注： 集約ストレージ・アウトラインを使用している場合、「ラベルのみ」オプションを選択するためのガイドラインを考慮してください。257ページの「「ラベルのみ」オプションを持つ集約ストレージ」を参照してください。
- 「既存」-既存の保管方法を使用します。
- 「外部ソース」-ドロップダウン・リストで、保管方法を含むデータベースの列を選択します。

  注： 外部データ・ソースで使用するメンバー・コードについては、Oracle Essbase Database Administrator's Guide のメンバー・プロパティを操作するためのデータ・ソースの使用、および集約ストレージの次元構築に対するデータ・ソースの違いの章を参照してください。

5 **「適用」をクリックします。**

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

「ラベルのみ」オプションを持つ集約ストレージ

集約ストレージ・メンバーで「ラベルのみ」オプションを選択する場合、次のガイドラインを考慮する必要があります:

- 「会計」のタグが付いた次元には、レベル 0 のメンバーを除き、次元レベルを含めたすべてのメンバー・レベルに、メンバーの「ラベルのみ」プロパティを使用できます。
- 「会計」以外のタグが付いた次元には、次元レベルを含め、「ラベルのみ」のタグが付いたメンバー以上のすべてのレベルにも「ラベルのみ」タグを付ける必要があります。さらに間接の子には、レベル 0 のメンバーを除き、すべて(+)集計プロパティのタグをつける必要があります。

  注： レベル 0 のメンバーには、「ラベルのみ」のタグを付けることはできません。
- 次元内に代替階層が含まれ、そのいずれかの階層に「ラベルのみ」プロパティが使用されている場合は、次元内のすべての階層で同じメンバー・レベルを「ラベルのみ」に設定する必要があります。

集約ストレージ・オプションの選択

集約ストレージを使用している場合、データ値を保管するためのオプションを選択できます。258 ページの「集約ストレージのガイドライン」を参照してください。

➤ 集約ストレージ・オプションを選択するには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。

2 「情報」タブを選択します。

3 「ASO ストレージ・オプション」グループを探します。

4 次の「ASO ストレージ・オプション」を選択します:

- 「保管」-データ値はメンバーとともに保管されます。新しいメンバーは、デフォルトでこの設定になります。
- 「動的」-メンバーに関連付けられているデータ値は、ユーザーが要求するまで計算されません。計算が完了すると、データ値は保管されずに破棄されます。
- 「外部ソース」-ドロップダウン・リストで、集約ストレージ・オプションを含むデータベースの列を選択します。

  注: 外部データ・ソースで使用するメンバー・コードについては、Oracle Essbase Database Administrator's Guide のメンバー・プロパティを操作するためのデータ・ソースの使用、および集約ストレージの次元構築に対するデータ・ソースの違いの章を参照してください。

5 「適用」をクリックします。

注: このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

集約ストレージのガイドライン

集約ストレージ・オプションを選択する場合は、次のガイドラインを考慮してください。

注: 集約ストレージとその使用の詳細は、Oracle Essbase Database Administrator's Guide の集約ストレージ・データベースの作成、計算および管理に関する説明を参照してください。

階層タイプの集約ストレージのガイドライン

- 式は、動的階層または比較次元のメンバーに対してのみ指定できます。

- 次元内に代替階層が含まれ、そのいずれかの階層に「ラベルのみ」オプションが使用されている場合は、次元内のすべての階層で同じメンバー・レベルを「ラベルのみ」に設定する必要があります。
- 動的階層で属性の関連付けを持つことができるのは、保管済メンバーのみです。
- 保管階層のすべてのメンバーは、+ [加算]集計メソッドでタグ付けする必要があります。「ラベルのみ」としてタグ付けされたメンバーと「ラベルのみ」メンバーの直接の子の場合は、この限りではありません。
- 共有メンバーが保管階層で使用されるのは、1 回のみである可能性があります。
- 複数の階層を持っている次元の最上位のメンバーには、「ラベル」のタグを付ける必要があります。
- 圧縮次元には、動的階層が 1 つのみ含まれている必要があります。
- 1 つの動的階層には、式を持たないレベル 0 のメンバーが少なくとも 1 つ必要です。

ストレージ・タイプの集約ストレージのガイドライン

- 集約ストレージ・アウトラインでは、保管済次元レベルの番号の製品は、2^32 未満である必要があります。
- 通貨の設定は指定できません。

次元タイプの集約ストレージ・ガイドライン

- 属性次元は、会計次元に関連付けられません
- 会計次元では、レベル 0 のメンバーを除き、すべてのメンバー・レベルに、メンバーの「ラベルのみ」プロパティを使用できます。
- 会計次元以外のすべての次元では、「ラベルのみ」タグが付けられたメンバーの上にあるすべてのレベルでも「ラベルのみ」タグが付けられている必要があります。「ラベルのみ」タグが付けられたメンバーのすべての子は、「+ [加算]」集計メソッドを使用してタグ付けする必要があります。ただし、メンバーの直接の子の場合はこの限りではありません。

  注： レベル 0 のメンバーには「ラベルのみ」タグを付けられません。
- 会計次元では、すべての集計メソッドを使用できます。
- 会計次元以外のすべての次元では、「+ [加算]」集計メソッドのみを使用できます。「ラベルのみ」タグが付けられたメンバーのすべての子は、「+ [加算]」集計メソッドを使用してタグ付けする必要があります。ただし、メンバーの直接の子の場合はこの限りではありません。
- 式は、動的階層または比較次元のメンバーに対してのみ指定できます。
- 「動的時系列」設定は、時間次元でのみ使用できます。
- 圧縮次元には、動的階層が 1 つのみ含まれている必要があります。
- 圧縮次元には、式のないレベル 0 のメンバーが少なくとも 1 つは必要です。

会計次元のメンバーの編集

「勘定科目情報」タブで、勘定科目メンバーのプロパティを編集して最適なタイム・バランスを選択し、Essbase に欠落した値または 0 を処理する方法を指示し、さらに実績データと予算データの差異をレポートする方法を選択できます。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

次のトピックでは、「勘定科目情報」タブで編集するメンバーのプロパティを説明します:

- 260 ページの「タイム・バランスの選択」
- 261 ページの「スキップ・オプションの選択」
- 262 ページの「差異レポートの作成方法の選択」
- 263 ページの「データ・ロード・スケーリングの指定」

タイム・バランスの選択

会計次元では、タイム・バランス・メソッドを選択して、親メンバーの計算方法を決定できます。

タイム・バランス機能の詳細は、235 ページの「タイム・バランスの概要」を参照してください。

➤ タイム・バランスを選択するには:

1. **246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。**
2. **会計次元を選択します。**
3. **メンバーを選択します。**
4. **「勘定科目情報」タブを選択します。**
5. **「タイム・バランス」グループを探します。**
6. **次のいずれかの「タイム・バランス」オプションを選択します:**
   - 「なし」-デフォルト値です。Essbase では、時間次元の親が通常の方法でロール・アップされます。親の値は、その親の子の式および集計プロパティに基づきます。
   - 「期首」-親の値は、分岐の最初のメンバーの値を表します(多くの場合は、期間の始め)。
   - 「期末」-親の値は、分岐の最後のメンバーの値を表します(多くの場合は、期間の最後)。
   - 「平均」-親の値は、子の値の平均を表します。
   - 「外部ソース」-ドロップダウン・リストで、データベース内の必要なタイム・バランス・オプションを含む列を選択します。

注： タイム・バランスを期首、期末または平均として設定する場合、スキップ・プロパティを設定して、欠落している値や 0 の値が検出された場合の対応を Essbase に指示する必要があります。236 ページの「スキップ・オプションの選択」を参照してください。

注： 外部データ・ソースで使用するメンバー・コードについては、Oracle Essbase Database Administrator's Guide のメンバー・プロパティを操作するためのデータ・ソースの使用、および集約ストレージの次元構築に対するデータ・ソースの違いの章を参照してください。

7 **「適用」をクリックします。**

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

スキップ・オプションの選択

タイム・バランスを期首、期末または平均として設定する場合、スキップ・プロパティを設定して、欠落している値や 0 の値が検出された場合の対応を Essbase に指示する必要があります。

➤ スキップ・オプションを選択するには:

1 **246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。**

2 **会計次元を選択します。**

3 **メンバーを選択します。**

4 **「勘定科目情報」タブを選択します。**

5 **「スキップ」グループを探します。**

6 **次のいずれかの「スキップ」オプションを選択します:**

- 「なし」-デフォルト値です。親の値を計算するときにデータをスキップしません。ただし、平均値の計算時に Essbase によって#MISSING データが検出されると、メンバーの合計数ではなく、実績値を持つメンバーの数で除算が行われます。したがって、スキップ・プロパティを「なし」に設定しないと、#MISSING が平均値に影響しなくなります(最初と最後は影響する)。
- 「欠落」-親の値の計算時に#MISSING データをスキップします。
- 「ゼロ」-親の値の計算時にゼロに相当するデータをスキップします。
- 「欠落およびゼロ」-#MISSING データとゼロに相当するデータを両方とも親の値の計算時にスキップします。
- 「外部ソース」-ドロップダウン・リストで、目的のスキップ・オプションを含むデータベースの列を選択します。

注： 外部データ・ソースで使用するメンバー・コードについては、Oracle Essbase Database Administrator's Guide のメンバー・プロパティを操作するためのデータ・ソースの使用、および集約ストレージの次元構築に対するデータ・ソースの違いの章を参照してください。

7 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

差異レポートの作成方法の選択

差異レポート・プロパティによって、Essbase が、メンバー式に@VAR または @VARPER 関数があるメンバーの実績データと予算データの差異を計算する方法が決まります。会社に支出を示すメンバーの場合、支出のプロパティが必要になります。

➤ 差異レポートの作成方法を選択するには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。

2 会計次元を選択します。

3 メンバーを選択します。

4 「勘定科目情報」タブを選択します。

5 「差異レポート」グループを探します。

6 次のいずれかの「差異レポート」の作成方法を選択します:

- 「支出なし」-売上などの支出なしの項目の場合、実績が予算よりも大きくなっている必要があります。実績が予算より小さい場合、差異は負数になります。@VAR 関数は、実績から予算を減算します。たとえば、売上予算が$100 で、売上実績が$110 の場合、差異は 10 になります。デフォルトでは、メンバーは支出なしになっています。
- 「支出」-支出項目の場合、支出実績は支出予算より小さくなります。支出実績が支出予算より大きい場合、差異は負数になります。@VAR 関数は、予算から実績を減算します。たとえば、支出予算が$100 で、支出実績が $110 の場合、差異は-10 になります。
- 「外部ソース」-ドロップダウン・リストから、目的の差異のオプションを含むデータベースの列を選択します。

注： 外部データ・ソースで使用するメンバー・コードについては、Oracle Essbase Database Administrator's Guide のメンバー・プロパティを操作するためのデータ・ソースの使用、および集約ストレージの次元構築に対するデータ・ソースの違いの章を参照してください。

7 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

データ・ロード・スケーリングの指定

Essbase Studio が Essbase データベースにデータをロードするときに、そのデータをスケーリングできます。データ・ソースの値とデータベースの値に異なるタイプの測定方法が使用されている場合、データのスケーリングが必要になることがあります。たとえば、データ・ソースが売上高を 100 単位で追跡しているのに対し、Essbase データベースは売上高を 1 単位で追跡している場合などです。この場合、入力値に 100 を乗算します。

➤ メンバーのデータ・ロード・スケーリングを選択するには:

1. 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。
2. 会計次元を選択します。
3. メンバーを選択します。
4. 「勘定科目情報」タブを選択します。
5. 「データ・ロード・スケーリング」グループを探します。
6. 「スケーリングを使用可能にする」チェック・ボックスを選択します。
7. 「スケーリング係数」テキスト・ボックスに、使用するスケーリング値を入力します; たとえば、入力値を 10 倍する場合は `10` を入力します。
8. 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

型付きメジャーの指定

型付きメジャーは Essbase の分析能力を拡張します。メジャーは、数値のほかに、テキストや日付タイプの値にも関連付けられます。

- メジャーが表示される次元内で、テキスト・メジャーには"テキスト"というタグが付けられます。テキスト・メジャーでは、セルの値にテキスト・ラベルを列挙したリストを 1 つ含められます。これらのラベルは、テキスト・リスト・オブジェクトと呼ばれるマッピング・アーティファクトを使用して、アウトライン・レベルで定義します。
- メジャーが表示される次元内で、日付メジャーには"日付"というタグが付けられます。日付メジャーでは、セルの値を特定のフォーマットの日付形式にできます。

  Essbase モデルに日付メジャーが含まれる場合は、配置の前に必ず、295 ページの「キューブの配置」の情報を確認してください。

次の一般的なガイドラインは、テキスト・メジャーと日付メジャーの両方に適用されます:

- 既存のメジャーの次元(たとえば、勘定科目)に追加します。
- これらのメジャーは集約しないでください。テキストおよび日付メジャーには、デフォルトで、集約を行わないことを示す記号(^)が割り当てられます。
- クエリーは、データがロードされたレベルと同じレベルで行う必要があります。

➤ 型付きメジャーを指定するには:

1 **使用する型付きメジャーが「メタデータ・ナビゲータ」に存在することを確認します。**

2 **246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。**

3 **「型付きメジャー」タブを選択します。**

4 **オプション:「メンバー・フォーマット文字列」テキスト・ボックスに MDX 式を入力して、小数点付き、カンマ付きまたはテキストなどの目的のフォーマットで値が表示されるようにセルの値のフォーマットを変更します。**

   注: メンバー文字列フォーマットと、導出されたテキスト・メジャーを混同しないでください。154 ページの「次元要素および導出テキスト・メジャーの作成または編集」を参照してください。

   注: MDX 式は 256 文字を超えてはならない。

5 **「テキスト・リストの関連付け」グループで、テキスト・リストに移動します。**

6 **テキスト・リストを選択します。**

   テキスト・リスト名が「現在割当て済のテキスト・リスト」フィールドに表示されます。

7 **「適用」をクリックします。**

メンバーへの式の追加

式により、Essbase が Essbase データベースのメンバー間の関係を計算する方法が決定されます。

式の詳細は、237 ページの「式について」を参照してください。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

➤ 次元に式を追加するには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。

2 「式」タブを選択します。

3 式を入力するためのオプションを選択します:

- 「式」テキスト・ボックスに式を入力します。
- 「外部ソースから」を選択し、ドロップダウン・リストから、使用する式を含む外部データ・ソース列を選択します。

注: 式の長さは最大で 64KB までです。

注: 外部データ・ソースで使用するメンバー・コードについては、Oracle Essbase Database Administrator's Guide のメンバー・プロパティを操作するためのデータ・ソースの使用、および集約ストレージの次元構築に対するデータ・ソースの違いの章を参照してください。

4 「適用」をクリックします。

注: このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

メンバーの別名の編集

「別名」タブでは、メンバーに対して作成した別名セットのリストを表示できます。また、別名の変更またはリフォーマットも可能です。

別名に関するガイドラインおよび詳細は、266 ページの「別名の概要」を参照してください。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

次のトピックでは、「別名」タブで編集するメンバーのプロパティについて説明します:

- 267 ページの「別名セットの表示」
- 267 ページの「別名の変更およびリフォーマット」
- 268 ページの「検索ルールを作成して別名を変更およびリフォーマットする方法」
- 269 ページの「検索ルールを変更して別名を変更およびリフォーマットする方法」
- 270 ページの「別名の大文字/小文字の変更」
- 270 ページの「別名のスペースのリフォーマット」
- 271 ページの「別名への接頭辞の追加」
- 272 ページの「別名への接尾辞の追加」

別名の概要

別名は、次元またはメンバーのわかりやすい代替名です。たとえば、SKU 製品コード 100 で識別されるメンバーに、Kool Cola という、よりわかりやすい名前を付けられます。この簡単に識別できる別名をメンバー名のかわりに表示できます。したがって、別名を使用することで、Essbase アウトラインおよびレポートが読みやすくなります。

この概要では、次のトピックを説明します:

- 266 ページの「別名の使用」
- 266 ページの「複数の別名の使用」

別名の使用

別名を使用するときの重要事項は次のとおりです:

- 別名は、別名テーブルに保管されます。
- 別名は、言語、地域またはわかりやすい名前でグループ化できます。
- リレーショナル・ソースを基本とする要素の場合、事前に外部データ・ソースの別名列を作成しておく必要があります。
- 別名は 80 文字に制限されています。80 文字を超える別名を持つデータ・ソースの列が存在する場合、次の 2 つの対処法があります:
  - SUBSTRING 関数を使用して 80 文字以下の文字を抽出し、別名で使用します。
  - 別名を置き換えます。
- 別名セットを使用して、1 つ以上の別名をメタデータ要素に割り当てられます (Essbase では、メンバー・レベル)。170 ページの「別名セットの操作」を参照してください。

複数の別名の使用

別名セットを使用すると、1 つのメンバー・レベルで複数の別名を指定できます。たとえば、次のように異なる種類のレポートに異なる別名を使用できます-ユーザーは 100-10 を Cola として馴染みがありますが、広告主およびエグゼクティブは The Best Cola の方が馴染みがある場合があります。このリストは、データベース内の一部の製品がどのように 2 つの記述的な別名を持つかを表しています:

```
Product

Default

Long Names

100-10    Cola        The Best Cola
100-20    Diet Cola     Diet Cola with Honey
100-30    Caffeine Free   Cola All the Cola, none of the Caffeine
```

別名セットは、特定の名前付き別名セットをメンバー名にマッピングします。論理ツリー内の各オブジェクトについて、カタログで定義された各別名セットに 1 つの別名を割り当てます。

別名セットの詳細は、169 ページの「別名セットについて」を参照してください。

「別名セット」ダイアログ・ボックスを使用して、別名セットを作成します。170 ページの「別名セットの作成」を参照してください。

別名セットの表示

「別名」タブに、メンバーに割り当てられている別名セットのリストが表示されます。

別名の詳細は、266 ページの「別名の概要」を参照してください。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

➤ メンバーに割り当てられた別名を表示するには:

1 **246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。**

2 **「別名」タブを選択します。**

ダイアログ・ボックスの一番上にあるグリッドで、「別名セット」列に、モデルに割り当てられている別名セットが表示されます。

左ペインで選択したメンバーに別名セットがマップされている場合、「ソース・マッピング」列に、別名マッピングのデータ・ソース接続、テーブルおよび列が表示されます。

注： 別名セクションは読取り専用です。

別名の変更およびリフォーマット

別名の名前変更またはリフォーマットを行って、アウトラインおよびレポートを読みやすくできます。これを行うには、検索ルールを作成または変更します:

- 268 ページの「検索ルールを作成して別名を変更およびリフォーマットする方法」を参照してください。
- 269 ページの「検索ルールを変更して別名を変更およびリフォーマットする方法」を参照してください。

注： 別名の名前を変更する場合、付録 B「Essbase Studio の命名規則」のガイドラインに従う必要があります。

別名の詳細は、266 ページの「別名の概要」を参照してください。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

検索ルールを作成して別名を変更およびリフォーマットする方法

➤ 検索ルールを作成して別名を変更およびリフォーマットするには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。

2 「別名」タブを選択します。

3 「ルールの作成」ボタンをクリックします。

「検索ルール」ダイアログ・ボックスが表示されます。

4 「検索対象」テキスト・ボックスに、変換する別名を入力します。

5 「置換後の文字列」テキスト・ボックスに、現在の別名を置き換える新しい別名を入力します。

6 変換を調整するには、次の検索オプションを選択します:

- 「大文字と小文字を区別」-検索条件に完全一致した別名のみが変換されます。
- 「完全に一致する単語」-検索条件に一致し、かつすべての単語が一致する別名が変換されます。
- 「すべて置換」-検索条件に一致する別名すべてが変換されます。

注: 検索オプションは、任意の数またはすべて選択することも、選択しないこともできます。

7 「OK」」をクリックして検索ルールを保存します。

作成した検索ルールが「別名」タブに表示されます。アイコンには、検索オプションのステータスが次のように反映されます:

-

  大文字と小文字を区別した検索-検索を絞り込むために「大文字と小文字を区別」を選択した場合、「TRUE」が表示されます。

-

  完全に一致する単語のみ-検索を絞り込むために「完全に一致する単語を検索」を選択した場合、「TRUE」が表示されます。

-

  すべて置換-検索を絞り込むために「すべて置換」を選択した場合、「TRUE」が表示されます。

8 「適用」をクリックします。

注: このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

検索ルールを変更して別名を変更およびリフォーマットする方法

➤ 別名の変更およびリフォーマットに使用する検索ルールを変更するには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。

2 「別名」タブを選択します。

3 「テキスト置換ルール」セクションで別名を右クリックします。

4 「詳細の編集」ボタンをクリックします。

「検索ルール」ダイアログ・ボックスが表示されます。

5 **オプション:**「検索」テキスト・ボックスに、変換する別名を入力するか、またはテキスト・ボックスに表示された別名を使用します。

6 「置換後の文字列」テキスト・ボックスに、現在の別名を置き換える新しい別名を入力します。

7 変換を調整するには、次の検索オプションを選択します:

- 「大文字と小文字を区別」-検索条件に完全一致した別名のみが変換されます。
- 「完全に一致する単語」-検索条件に一致し、かつすべての単語が一致する別名が変換されます。
- 「すべて置換」-検索条件に一致する別名すべてが変換されます。

> **注:** 検索オプションは、任意の数またはすべて選択することも、選択しないこともできます。

8 「OK」」をクリックして検索ルールを保存します。

作成した検索ルールが「別名」タブに表示されます。検索オプションのステータスを反映したアイコンに注意してください:

-

  大文字と小文字を区別した検索-検索を絞り込むために「大文字と小文字を区別」を選択した場合、「TRUE」が表示されます。

-

  完全に一致する単語のみ-検索を絞り込むために「完全に一致する単語を検索」を選択した場合、「TRUE」が表示されます。

-

  すべて置換-検索を絞り込むために「すべて置換」を選択した場合、「TRUE」が表示されます。

9 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

別名の大文字/小文字の変更

すべて大文字、すべて小文字またはタイトル文字になるように別名の大文字/小文字を変更できます。また、大文字と小文字を元のフォーマットにリセットすることもできます。

別名の詳細は、266 ページの「別名の概要」を参照してください。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

➤ 別名の大文字/小文字を変更するには:

1. **246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。**
2. **「別名」タブを選択します。**
3. **「大文字/小文字を変更」ドロップダウン・リストで、大文字/小文字のオプションを選択します。**
4. **「適用」をクリックします。**

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

別名のスペースのリフォーマット

別名に含まれるスペースをアンダースコアに変換してリフォーマットできます。また、先頭および末尾のスペースを削除することもできます。

別名の詳細は、266 ページの「別名の概要」を参照してください。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

➤ 別名に含まれるスペースをリフォーマットするには:

1. **246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。**
2. **「別名」タブを選択します。**
3. **次のいずれかのスペース・オプションを選択します:**
   - 「先頭/末尾のスペースを削除」-先頭および末尾のスペースがすべて別名から削除されます。
   - 「スペースをアンダースコアに変換」-別名に含まれるすべてのスペースがアンダースコアに変更されます。

4 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

別名への接頭辞の追加

接頭辞または接尾辞、あるいはその両方を別名に追加できます。

272 ページの「別名への接尾辞の追加」を参照してください。

266 ページの「別名の概要」を参照してください。

245 ページの「メンバーの概要」を参照してください。

➤ 別名に接頭辞を追加するには:

1 **246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。**

2 **「別名」タブを選択します。**

3 **ドロップダウン・リストから、次のいずれかの接頭辞オプションを選択します:**

- 「なし」-別名に接頭辞は追加されません。

  注： 後続の手順 5 に記載されているようにカスタム接頭辞を使用する場合は、「なし」を選択します。

- 「親の名前」-メンバーの親の名前が接頭辞として別名に追加されます。
- 「親の親の名前」-メンバーの親の親の名前が接頭辞として別名に追加されます。
- 「すべての祖先の名前」-メンバーのすべての祖先のすべての名前が接頭辞として別名に追加されます。

  注： 接頭辞に「すべての祖先の名前」を選択すると、長い接頭辞が生成される場合があります。別名は 80 文字に制限されています。266 ページの「別名の使用」を参照してください。

- 「次元名」-メンバーが存在する次元の名前が接頭辞として別名に追加されます。

接頭辞追加後の別名のプレビューが、「カスタム接頭辞」テキスト・ボックスのすぐ下にあるテキスト・ボックスに表示されます。

4 **オプション: 接頭辞を別名と区切るには、「接頭辞区切り文字」を選択します。**

5 **オプション: 「カスタム接頭辞」テキスト・ボックスにカスタム接頭辞を入力します。**

注： カスタム接頭辞を入力する場合は、前述の手順 3 に記載されているように、「接頭辞」ドロップダウン・リストに「なし」が選択されていることを確認してください。

注： 一部の特殊文字は接頭辞に使用できません。付録 B「Essbase Studio の命名規則」を参照してください。

6 **「適用」をクリックします。**

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

別名への接尾辞の追加

接頭辞、接尾辞または両方を別名に追加できます。

271 ページの「別名への接頭辞の追加」を参照してください。

266 ページの「別名の概要」を参照してください。

245 ページの「メンバーの概要」を参照してください。

➤ 別名に接尾辞を追加するには:

1 **246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。**

2 **「別名」タブを選択します。**

3 **接尾辞のオプションをプルダウン・リストから選択します:**

- 「なし」-接尾辞が別名に追加されません。

  注： 次の手順 5 で説明するカスタム接尾辞を使用する場合、「なし」を選択します。

- 「親の名前」-メンバーの親の名前が別名に接尾辞として追加されます。
- 「親の親の名前」-メンバーの親の親の名前が接尾辞として別名に追加されます。
- 「すべての祖先の名前」-メンバーのすべての祖先の名前が接尾辞として別名に追加されます。

  注： 接尾辞に「すべての祖先の名前」を選択すると、長い接尾辞が生成される場合があります。別名は 80 文字に制限されています。266 ページの「別名の使用」を参照してください。

- 「次元名」-メンバーが存在する次元の名前が接尾辞として別名に追加されます。

接尾辞が追加された状態の別名のプレビューが「カスタム接尾辞」テキスト・ボックスのすぐ下にあるテキスト・ボックスに表示されます。

4 **オプション:**「接尾辞区切り文字」を選択すると、別名と接尾辞が区切られます。

5 **オプション:**「カスタム接尾辞」テキスト・ボックスにカスタム接尾辞を入力します。

注: カスタム接尾辞を入力する場合は、前述の手順 3 に記載されているように、「接尾辞」ドロップダウン・リストに「なし」が選択されていることを確認してください。

注: 接尾辞で使用できない特殊文字がいくつかあります。付録 B「Essbase Studio の命名規則」を参照してください。

6 「適用」をクリックします。

注: このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

メンバーへのユーザー定義属性の割当て

ユーザー定義属性(UDA)をメンバーに割り当てて、メンバーの特性を表せます。たとえば、Ounces という UDA を作成できます。

240 ページの「UDA の概要」を参照してください。

245 ページの「メンバーの概要」を参照してください。

➤ UDA をメンバーに割り当てるには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。

2 「UDA」タブを選択します。

3 「UDA 値」テキスト・ボックスで、メンバーに使用する新しい UDA を入力します。

4 **オプション:** Essbase Studio にデータ・ソースの列から UDA を取得させる場合は、「外部ソース」を選択し、ドロップダウン・リストで該当の列を選択します。

メンバーに割り当てられている UDA が「既存 UDA」テキスト・ボックスに表示されます。

注: 外部データ・ソースで使用するメンバー・コードについては、Oracle Essbase Database Administrator's Guide のメンバー・プロパティを操作するためのデータ・ソースの使用、および集約ストレージの次元構築に対するデータ・ソースの違いの章を参照してください。

5 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

メンバーの変換

「変換」タブで、メンバー名の変更やリフォーマットを行い、Essbase アウトラインおよびレポートを読みやすくできます。

別名に関するガイドラインおよび詳細は、266 ページの「別名の概要」を参照してください。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

Essbase によるメンバー名の変換またはリフォーマットの順序が原因で、変換操作を実行できないことがあります。274 ページの「メンバー名変換シーケンス」を参照してください。

次のトピックでは、「変換」タブで編集するメンバーのプロパティを説明します:

- 275 ページの「メンバーへの接頭辞の追加」
- 276 ページの「メンバーへの接尾辞の追加」
- 277 ページの「メンバー名の変更およびリフォーマット」
- 277 ページの「検索ルールを作成してメンバー名を変更およびリフォーマットする方法」
- 278 ページの「検索ルールを変更してメンバー名を変更およびリフォーマットする方法」
- 279 ページの「メンバーの大文字/小文字の変更」
- 280 ページの「メンバーのスペースのリフォーマット」

メンバー名変換シーケンス

Essbase では、次のシーケンスでメンバー名が変換およびリフォーマットされます:

1. 置換文字列が処理されます。
2. 先頭および末尾のスペースが削除されます。
3. スペースがアンダースコアに変換されます。
4. 接頭辞が追加されます。
5. 接尾辞が追加されます。
6. 大文字と小文字の変更が行われます。

注： Essbase はこのシーケンスに従うため、変換の実行が許可されない場合があります。たとえば、カスタム接頭語を追加してその接頭語の置換文字列を選択した場合、接頭辞の追加前に置換文字列が処理されるため、操作は機能しません。

メンバーへの接頭辞の追加

接頭辞または接尾辞、またはその両方をメンバーに追加できます。

注： メンバーがテキスト・ファイル・ソースの場合、接頭辞および接尾辞はサポートされません。

接尾辞をメンバーに追加するには、276 ページの「メンバーへの接尾辞の追加」を参照してください。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

➤ メンバー名に接頭辞を追加するには:

1 **246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。**

2 **「変換」タブをクリックします。**

3 **ドロップダウン・リストから「接頭辞」オプションを選択します:**

- 「なし」-接頭辞がメンバー名に追加されません。

  注： 後続の手順 5 に記載されているようにカスタム接頭辞を使用する場合は、「なし」を選択します。

- 「親の名前」-メンバーの親の名前が接頭辞としてメンバー名に追加されます。
- 「親の親の名前」-メンバーの親の親の名前が接頭辞としてメンバー名に追加されます。
- 「すべての祖先の名前」-メンバーのすべての祖先の名前が接頭辞としてメンバー名に追加されます。

  注： 接頭辞に「すべての祖先の名前」を選択すると、長い接頭辞が生成される場合があります。メンバー名は最大 80 文字です。

- 「次元名」-メンバーが存在する次元の名前が接頭辞としてメンバー名に追加されます。

接頭辞を追加した状態のメンバー名のプレビューは、「カスタム接頭辞」テキスト・ボックスのすぐ下のテキスト・ボックスに表示されます。

4 **オプション:「接頭辞区切り文字」を選択すると、メンバー名と接頭辞が区切られます。**

5 **オプション:「カスタム接頭辞」テキスト・ボックスにカスタム接頭辞を入力します。**

注： カスタム接頭辞を入力する場合は、前述の手順 3 に記載されているように、「接頭辞」ドロップダウン・リストに「なし」が選択されていることを確認してください。

注： 一部の特殊文字は接頭辞に使用できません。付録 B「Essbase Studio の命名規則」を参照してください。

6 **「適用」をクリックします。**

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

メンバーへの接尾辞の追加

接頭辞、接尾辞または両方をメンバーに追加できます。

注： メンバーがテキスト・ファイル・ソースの場合、接頭辞および接尾辞はサポートされません。

接頭辞をメンバー名に追加するには、275 ページの「メンバーへの接頭辞の追加」を参照してください。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

➤ メンバー名に接尾辞を追加するには:

1 **246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。**

2 **「変換」タブをクリックします。**

3 **ドロップダウン・リストから「接尾辞」オプションを選択します:**

- 「なし」-接尾辞がメンバー名に追加されません。

  注： 後続の手順 5 に記載されているようにカスタム接尾辞を使用する場合は、「なし」を選択します。

- 「親の名前」-メンバーの親の名前が接尾辞としてメンバー名に追加されます。
- 「親の親の名前」-メンバーの親の親の名前が接尾辞としてメンバー名に追加されます。
- 「すべての祖先の名前」-メンバーのすべての祖先の名前が接尾辞としてメンバー名に追加されます。

  注： 接尾辞に「すべての祖先の名前」を選択すると、長い接尾辞が生成される場合があります。メンバー名は最大 80 文字です。

- 「次元名」-メンバーが存在する次元の名前が接尾辞として別名に追加されます。

接尾辞が追加された状態のメンバー名のプレビューが「カスタム接尾辞」テキスト・ボックスのすぐ下にあるテキスト・ボックスに表示されます。

4 **オプション:**「接尾辞区切り文字」を選択すると、別名と接尾辞が区切られます。

5 **オプション:**「カスタム接尾辞」テキスト・ボックスにカスタム接尾辞を入力します。

注： カスタム接尾辞を入力する場合は、前述の手順 3 に記載されているように、「接尾辞」ドロップダウン・リストに「なし」が選択されていることを確認してください。

注： 接尾辞で使用できない特殊文字がいくつかあります。付録 B「Essbase Studio の命名規則」を参照してください。

6 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

メンバー名の変更およびリフォーマット

メンバー名を変更またはリフォーマットして、アウトラインとレポートを読みやすくすることができます。これを行うには、検索ルールを作成または変更します:

- 277 ページの「検索ルールを作成してメンバー名を変更およびリフォーマットする方法」を参照してください。
- 278 ページの「検索ルールを変更してメンバー名を変更およびリフォーマットする方法」を参照してください。

注： メンバーの名前を変更するときには、付録 B「Essbase Studio の命名規則」のガイドラインに従う必要があります。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

Essbase によるメンバー名の変換またはリフォーマットの順序が原因で、変換操作を実行できないことがあります。274 ページの「メンバー名変換シーケンス」を参照してください。

検索ルールを作成してメンバー名を変更およびリフォーマットする方法

➤ メンバー名を変更およびリフォーマットするための検索ルールを作成するには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。

2 「変換」タブをクリックします。

3 「ルールの作成」ボタンをクリックします。

「検索ルール」ダイアログ・ボックスが表示されます。

4 「検索対象」テキスト・ボックスに、変換するメンバー名を入力します。

5 「置換後の文字列」テキスト・ボックスに、現在のメンバー名を置換する新しいメンバー名を入力します。

6 変換を絞り込む検索オプションを選択します:

- 「大文字と小文字を区別」-正確に検索条件と一致するメンバーのみが変換されます。
- 「完全に一致する単語」-検索条件に一致し、かつ単語が完全に一致するメンバーのみが変換されます。
- 「すべて置換」-検索条件に一致するすべてのメンバーが変換されます。

注： 検索オプションは、任意の数またはすべて選択することも、選択しないこともできます。

7 「OK」」をクリックして検索ルールを保存します。

作成した検索ルールは「変換」タブに表示されます。検索オプションのステータスを反映したアイコンに注意してください:

-

  「大文字と小文字を区別した検索」 - 検索を絞り込むために「大文字と小文字を区別」を選択した場合、「TRUE」が表示されます。

-

  「完全に一致する単語のみ」 - 検索を絞り込むために「完全に一致する単語を検索」を選択した場合、「TRUE」が表示されます。

-

  「すべて置換」 - 検索を絞り込むために「すべて置換」を選択した場合、「TRUE」が表示されます。

8 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

検索ルールを変更してメンバー名を変更およびリフォーマットする方法

➤ メンバー名の変更およびリフォーマットに使用する検索ルールを変更するには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。

2 「変換」タブをクリックします。

3 「テキスト置換ルール」セクションで、メンバーを右クリックします。

4 「詳細の編集」ボタンをクリックします。

「検索ルール」ダイアログ・ボックスが表示されます。

5 **オプション:**「検索」テキスト・ボックスで、変換するメンバー名を入力するか、テキスト・ボックスに表示されるメンバー名を使用します。

6 「置換後の文字列」テキスト・ボックスに、現在のメンバー名を置換する新しいメンバー名を入力します。

7 変換を調整するには、次の検索オプションを選択します:

- 「大文字と小文字を区別」-正確に検索条件と一致するメンバーのみが変換されます。
- 「完全に一致する単語」-検索条件に一致し、かつ単語が完全に一致するメンバーのみが変換されます。
- 「すべて置換」-検索条件に一致するすべてのメンバーが変換されます。

注： 検索オプションは、任意の数またはすべて選択することも、選択しないこともできます。

8 「OK」」をクリックして検索ルールを保存します。

作成した検索ルールは「変換」タブに表示されます。検索オプションのステータスを反映したアイコンに注意してください:

-

「大文字と小文字を区別した検索」 - 検索を絞り込むために「大文字と小文字を区別」を選択した場合、「TRUE」が表示されます。

-

「完全に一致する単語のみ」 - 検索を絞り込むために「完全に一致する単語を検索」を選択した場合、「TRUE」が表示されます。

-

「すべて置換」 - 検索を絞り込むために「すべて置換」を選択した場合、「TRUE」が表示されます。

9 「適用」をクリックします。

注： このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

メンバーの大文字/小文字の変更

メンバーの大文字と小文字を、すべて大文字、すべて小文字、または先頭のみ大文字に変更できます。また、大文字と小文字を元のフォーマットにリセットすることもできます。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

➤ メンバーの大文字と小文字を変更するには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。
2 「変換」タブをクリックします。
3 「大文字/小文字を変更」ドロップダウン・リストで、大文字/小文字のオプションを選択します。
4 「適用」をクリックします。

注: このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

メンバーのスペースのリフォーマット

メンバーに含まれるスペースをアンダースコアに変換してリフォーマットできます。また、先頭および末尾のスペースを削除することもできます。

メンバーの詳細は、245 ページの「メンバーの概要」を参照してください。

➤ メンバー名に含まれるスペースをリフォーマットするには:

1 246 ページの「「メンバーのプロパティ」タブへのアクセス」の手順に従って、「Essbase モデル・プロパティ」ダイアログ・ボックスにあるメンバー・プロパティのタブにアクセスします。
2 「変換」タブをクリックします。
3 次のいずれかのスペース・オプションを選択します:
   - 「先頭/末尾のスペースを削除」-先頭および末尾のスペースがすべてメンバー名から削除されます。
   - 「スペースをアンダースコアに変換」-メンバー名に含まれるすべてのスペースがアンダースコアに変更されます。
4 「適用」をクリックします。

注: このタブの設定を前回保存した値に復元するには、「復元」をクリックします。設定をオリジナルのシステム・デフォルト値に復元するには、「デフォルト」をクリックします。

Essbase モデル・プロパティの操作

Essbase モデル・プロパティを編集した後で、次のタスクを実行できます:

- 281 ページの「モデルの表示」
- 282 ページの「モデル・プロパティの検証」

- 283 ページの「プロパティへの変更の確認」
- 283 ページの「モデルの参照」
- 284 ページの「ツール・チップの使用」

208 ページの「Essbase モデルの概要」を参照してください。

モデルの表示

モデル作業領域を使用して、モデルのグラフィカル表現を表示できます。この機能は、モデルの階層構造を確認するときに役立ちます。

さらに、モデル作業領域を使用して「キューブ配置ウィザード」を起動したり、モデルのプロパティを編集したりできます。

モデルの詳細は、208 ページの「Essbase モデルの概要」を参照してください。

次のトピックでは、モデル作業領域で実行できるタスクを説明します:

- 281 ページの「モデル作業領域を開く」
- 281 ページの「モデル作業領域の使用」

モデル作業領域を開く

モデル作業領域を使用して、次元およびメンバーをグラフィック表示できます。

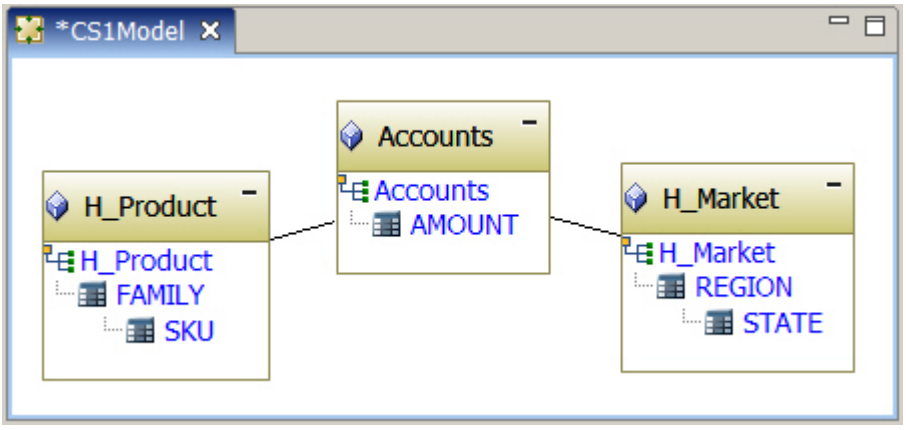
➤ モデル作業領域を開くには:

1 **「メタデータ・ナビゲータ」で、モデル名を右クリックします。**

2 **「編集」を選択します。**

モデルがメンバーおよび次元とともに、モデル作業領域に次のようにグラフィック表示されます:

図 16　モデル作業領域



モデル作業領域の使用

モデル作業領域を使用すると、次のタスクを実行できます:

- **次元の移動** - モデル作業領域内で次元を移動するには、次元をクリックして、移動先にドラッグします。
- **次元の展開と縮小** - 次元を展開または縮小してメンバーを表示または非表示にするには、次元の右上隅にある最大化または最小化アイコンを左クリックします。
- 「元に戻す」 - 次元の移動を元に戻すには、モデル作業領域を右クリックし、「元に戻す」を選択します。
- 「やり直し」 - 次元の移動をやり直すには、モデル作業領域を右クリックし、「やり直し」を選択します。
- 「ズーム・イン」 - 次元を拡大表示するには、モデル作業領域を右クリックし、「ズーム・イン」を選択します。
- 「ズーム・アウト」 - 次元を縮小表示するには、モデル作業領域を右クリックし、「ズーム・アウト」を選択します。
- 「キューブ配置ウィザードの起動」 - モデル作業領域を右クリックし、「キューブ配置ウィザード」を選択します。

  295 ページの「キューブの配置」を参照してください。
- 「プロパティの編集」 - モデル、次元およびメンバーのプロパティを編集するには、モデル作業領域を右クリックし、「Essbase プロパティ」を選択します。

  または、次元かメンバー名を右クリックします

モデル・プロパティの検証

「プロパティの検証」ダイアログ・ボックスを使用して、必要な次元の欠落や無効な名前など、モデルのエラーをチェックできます。

モデルの詳細は、208 ページの「Essbase モデルの概要」を参照してください。

➤ モデルを検証するには:

1 209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」の手順を使用して、「Essbase モデル・プロパティ」ダイアログ・ボックスにアクセスします。

2 「検証を実行」ボタンをクリックし、 「プロパティの検証」ダイアログ・ボックスにアクセスします。

3 表示された要素の 1 つ(モデル、次元、またはメンバー)をダブル・クリックし、要素に関連するエラーまたは無効なプロパティを確認します。

  注 : 「プロパティの検証」ダイアログ・ボックスに要素が表示されていない場合、モデルにエラー条件がありません。

4 エラー・メッセージを選択して右クリックし、オプション・メニューを表示すると、考えられるエラーの修正候補が表示されます。

5 オプションを選択します。

  エラー修正のための「プロパティ」ダイアログ・ボックスが表示されます。

6 「プロパティ」ダイアログ・ボックスでエラーを修正し、「適用」をクリックします。

7 「再検証」ボタンをクリックし、 エラーが修正されたことを確認します。

注： エラーが修正されていれば、ダイアログ・ボックスに要素が表示されなくなります。

8 エラーがすべて修正されたら、「閉じる」をクリックします。

プロパティへの変更の確認

「適用プロパティ」・ダイアログ・ボックスを使用して、このセッションでモデルのプロパティに適用した変更のリストを表示できます。

注： 表示されるのは、このセッションで加えた変更のみです。セッションとは、「Essbase プロパティ」タブにアクセスしていた時間のことで、タブを閉じるまで続きます。

モデルの詳細は、208 ページの「Essbase モデルの概要」を参照してください。

➤ モデル・プロパティへの変更のリストを表示するには:

1 209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」の手順を使用して、「Essbase モデル・プロパティ」ダイアログ・ボックスにアクセスします。

2 「変更の表示」ボタン をクリックして、「適用プロパティ」ダイアログ・ボックスにアクセスします。

このセッション中にモデルに加えた変更が表示されます。

3 要素への変更を表示するには、表示された要素(モデル、次元またはメンバー)のいずれかを展開します。

4 変更内容を変更する場合は、要素を展開します。

プロパティを変更できるようにプロパティ・ダイアログ・ボックスが開きます。

5 プロパティ・ダイアログ・ボックスでプロパティを編集し、「適用」をクリックします。

6 「変更の表示」ボタン をクリックして、「適用プロパティ」ダイアログ・ボックスにアクセスします。

このセッション中に新しくモデルに加えた変更が表示されます。

7 変更の確認が終了したら、「取消し」をクリックします。

モデルの参照

「戻る」および「進む」ボタンを使用して、以前に表示または編集した次元またはメンバーに戻れます。

注： このセッションで表示または編集した次元およびメンバーのみが表示されます。セッションとは、「Essbase プロパティ」タブにアクセスしていた時間のことで、タブを閉じるまで続きます。

モデルの詳細は、208 ページの「Essbase モデルの概要」を参照してください。

➤ 表示また編集した次元およびメンバーを参照するには:

1. 209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」の手順を使用して、「Essbase モデル・プロパティ」ダイアログ・ボックスにアクセスします。
2. モデルを展開して、次元およびメンバーの名前を表示します。
3. 次元またはメンバーの名前を選択します。

   次元またはメンバーのプロパティ・タブが表示されます。
4. **オプション:** 次元またはメンバーのプロパティを編集し、「適用」をクリックします。
5. **オプション:** いくつかの次元およびメンバーのプロパティの編集または表示後に、以前に表示または編集した次元やメンバーに戻るには、「戻る」ボタン をクリックします。

   直前の次元またはメンバーが、該当するプロパティ・タブに表示されます。
6. **オプション:** もう 1 つ前に表示または編集した次元やメンバーに戻るには、「戻る」ボタン を再度クリックします。

   次元またはメンバーが該当するプロパティ・タブに表示されます。
7. **オプション:** 表示または編集した次元やメンバーに進むには、「進む」ボタン をクリックします。

   次の次元またはメンバーが、該当するプロパティ・タブに表示されます。

ツール・チップの使用

カーソルを使用してツール・チップを表示できます。このツール・チップにより、モデル、次元およびメンバーのプロパティをすばやく表示できます。

➤ ツール・チップを使用してプロパティをすばやく表示するには:

1. 209 ページの「「Essbase モデル・プロパティ」ダイアログ・ボックスへのアクセス」の手順を使用して、「Essbase モデル・プロパティ」ダイアログ・ボックスにアクセスします。
2. モデルを展開して、次元およびメンバーの名前を表示します。
3. カーソルをモデル、次元またはメンバーの名前に合わせます。

次元またはメンバーの主なプロパティの要約が表示されます。

12 モデルの再同期

この章の内容

モデルの再同期について 287
モデルの再同期の使用 288

モデルの再同期について

これらのメタデータ要素、**キューブ・スキーマ**または**階層**のいずれかを変更した場合、モデルの再同期を使用して、そのメタデータ要素を使用する Essbase モデルに変更を伝播できます。モデルの再同期を行う場合、モデルを再作成したり、設定を再構成する必要はありません。

メタデータ要素を変更すると、Essbase Studio により、その要素を使用するモデルに、要素の最新バージョンと同期していないというタグが付けられます。これにより、モデルを再同期するように選択できます。

モデルの再同期の必要性をトリガーするイベントには、次のものがあります:

- キューブ・スキーマの変更:
  - キューブ・スキーマへの階層の追加
  - 階層の削除または置換
  - 階層の移動
  - 階層の変更
- 階層の変更:
  - 階層へのチェーンの追加
  - チェーンの削除または置換
  - チェーンの移動
  - チェーンの変更

メタデータ要素がいずれかの Essbase モデルで使用されている場合、その要素の変更を開始すると、警告が表示されます。また、その要素を使用するすべてのモデルのリストも表示されます。更新の続行を決定すると、「メタデータ・ナビゲータ」で同期していないすべてのモデルの名前の横にこのアイコンが付きます:

。

注： モデルを再同期した場合、もう一方の変更されていない要素の設定は現行値を保持します。階層内の変更されていないチェーンもすべて、その現行設定を保持します。いずれかの要素の名前が変更された場合、その他の要素はすべて現行名を保持します。

詳細は、372 ページの「Essbase モデルの再同期のガイドラインと制限」を参照してください。

モデルの再同期の使用

メタデータ要素を更新した後は、モデルの再同期を使用して、そのメタデータ要素を使用する Essbase モデルに変更を伝播します。

モデルの再同期には各種のタイプがあります:

- 特定のメタデータ要素の変更による影響を受けるモデルを再同期できます。288 ページの「メタデータ要素 - モデルの再同期」を参照してください。
- 複数の要素に基づいて、1 つのモデルを再同期できます。288 ページの「モデルの再同期」を参照してください。

メタデータ要素 - モデルの再同期

➤ 特定のメタデータ要素の変更による影響を受けるモデルを再同期するには:

1 **「メタデータ・ナビゲータ」で、更新したメタデータ要素をハイライトします。**

2 **右クリックして、「非同期モデルの更新」を選択します**

「モデルの同期」ダイアログ・ボックスが表示されます。要素の変更によって再同期する必要が生じたすべてのモデルが、「非同期モデル」にリストされます。

3 **更新するモデルを選択して、「同期するモデル」に移動します。**

注： この方法を使用してモデルを再同期すると、モデルはこの特定の要素の変更のみを反映するように更新されます。モデル内の他の要素は更新されません。

4 **「OK」をクリックして、モデルを更新済のメタデータ要素と同期します。**

注： 他にも変更された要素がある場合、モデルはまだ同期されていない可能性があります。また、再同期操作後もモデルが無効になっていることもあります。再同期されたすべてのモデルを再検証する必要があります。

モデルの再同期

➤ 複数の要素に基づいて 1 つのモデルを再同期するには:

1 **「メタデータ・ナビゲータ」で、非同期モデルをハイライトします。**

「メタデータ・ナビゲータ」で、非同期モデルの名前の横にこのアイコンが表示されます:

2 **右クリックして、「非同期モデルの更新」を選択します**

「モデルの同期」ダイアログ・ボックスが表示されます。更新されたメタデータ要素はすべて、「変更された階層とキューブ・スキーマ」にリストされます。

3 **モデルと同期する更新済要素を選択して、「モデルと同期するオブジェクト」に移動します。**

注： この方法を使用してモデルを再同期すると、モデルは、指定した要素の変更のみを反映するように更新されます。モデル内の他の要素は更新されません。

4 **「OK」をクリックして、部分的な再同期を実行します。**

注： 他にも変更された要素がある場合、モデルはまだ同期されていない可能性があります。また、再同期操作後もモデルが無効になっていることもあります。再同期されたすべてのモデルを再検証する必要があります。

13 キューブ配置

この章の内容

キューブ配置について 291
キューブの配置 295
Essbase サーバー接続の作成 295
キューブ配置のための接続情報の提供 297
配置オプションの設定 299
キューブ配置の増分ロードの設定 304
配置の進捗状況および結果の表示 305

キューブ配置について

キューブ・スキーマ・ウィザードを使用して構築した Essbase モデルは、Essbase サーバーへのキューブの配置に使用されます。キューブ配置は、アウトラインを構築して、アプリケーションおよびデータベースにデータをロードするためのロード・オプションを設定するプロセスです。構築するアウトラインは、管理サービス・コンソールで編集できます。作成したアプリケーションおよびデータベースには、Smart View からアクセスできます。

「キューブ配置ウィザード」を使用して、キューブを配置する場合に次のいくつかのロード・オプションから選択できます:

- アウトラインのみを構築
- データのロードのみ
- アウトラインの構築およびデータのロード
- 増加的にメンバーをロード
- ロード前にキューブからメンバーを削除
- キューブの削除およびキューブへのデータの復元
- ストリーミング・モードまたは非ストリーミング・モードでの配置
- Essbase ルール・ファイルを作成して保存します。このファイルは、メンバーまたはデータ(あるいはその両方)を後でロードする際に使用できます
- 選択したすべてのキューブ配置オプションを含む MaxL ロード・スクリプトを作成して保存します。このファイルは、メンバーまたはデータ(あるいはその両方)を後でロードする際に使用できます

次の要因に応じて、キューブ配置中にデータが取得される方法を調整できます:

- `server.essbase.streamingCubeBuilding` サーバー・プロパティの設定内容

- 「キューブ配置ウィザード」の「データ・ソースの設定」グループで選択したオプション
- ストリーミング・モード配置オプションを選択したかどうか。

292 ページの「配置シナリオとストリーミング・キューブ・ビルディング・プロパティの注意点」および297 ページの「キューブ配置のための接続情報の提供」を参照してください。

日付メジャーをキューブ・スキーマで使用した場合、データを正しく集計するためのキューブ配置の実行については、294 ページの「日付メジャーの配置」を参照してください。

キューブ・スキーマ作成中に、Essbase モデルの作成を選択しなかった場合は、いつでも作成できます。また、206 ページの「既存のキューブ・スキーマからの Essbase モデルの作成」で説明しているように、同一のキューブ・スキーマに対して複数の Essbase モデルを作成することもできます。Essbase モデルの作成後に「キューブ配置ウィザード」を起動して、297 ページの「キューブ配置のための接続情報の提供」に説明されている最初のページを完了します。ウィザードから、Essbase プロパティで説明されているモデルの第 11 章「Essbase プロパティ」も変更できます。

注： この章では、「キューブ配置ウィザード」を使用したキューブの配置について説明します。MaxL 配置コマンドを使用してもキューブを配置できます。「キューブ配置ウィザード」のオプションにより、配置スクリプトが自動的に作成されます。「キューブ配置ウィザード」で使用可能なすべてのオプションは、配置コマンドでも使用でき、MaxL スクリプトに記録されます。詳細は、『Oracle Essbase テクニカル・リファレンス』の MaxL についての説明を参照してください。

配置シナリオとストリーミング・キューブ・ビルディング・プロパティの注意点

キューブ配置について、Essbase Studio は 3 つのデータ取得モード・オプションを提供します:

- Essbase Studio Server によって導出された接続文字列を使用した、非ストリーミング・モード(デフォルト)
- ユーザーが提供した ODBC DSN を使用した、非ストリーミング・モード
- Essbase Studio Server によって導出された接続文字列を使用したストリーミング・モード

このトピックに記述されているとおり、これらのモードの使用はストリーミング・キューブ・ビルディング・プロパティ `server.essbase.streamingCubeBuilding` の設定内容によって異なります。

`server.essbase.streamingCubeBuilding` プロパティの設定の詳細は、42 ページの「server.essbase.streamingCubeBuilding」を参照してください。

Essbase Studio Server によって導出された接続文字列を使用した、非ストリーミング・モード(デフォルト)

```
server.essbase.streamingCubeBuilding=false
```

「キューブ配置ウィザード」で、「ODBC (Essbase は ODBC 接続文字列を動的に作成)」を選択します。

Essbase はこのオプションによって、配置中にデータ・ソースへの接続を簡単に確立でき、すべての接続属性にデフォルト値が想定されます。この場合、Essbase Studio は配置中に ODBC ドライバを使用します。

ユーザーが提供した ODBC DSN を使用した、非ストリーミング・モード

```
server.essbase.streamingCubeBuilding=false
```

「キューブ配置ウィザード」で、「ODBC (ODBC DSN 名を入力)」を選択し、すでに設定済の ODBC DSN を入力します。

Essbase はこのオプションによって、ODBC DSN で指定されている属性を利用できます。たとえば、配置中の Essbase モデルが Unicode データ・ソースを基にしている場合、N-CHAR サポートを有効にした ODBC DSN を作成してから、この DSN を配置時に指定できます。

Essbase Studio Server によって導出された接続文字列を使用したストリーミング・モード

```
server.essbase.streamingCubeBuilding=true
```

「キューブ配置ウィザード」で、「ODBC (Essbase は ODBC 接続文字列を動的に作成)」を選択します。

Essbase はこのオプションによって、配置中にデータ・ソースへの接続を簡単に確立でき、すべての接続属性にデフォルト値が想定されます。この場合、Essbase Studio は配置中に JDBC ドライバを使用します。

次の表に、配置モードのガイドラインと想定される配置シナリオを示します。

表 6　配置モードのガイドラインと想定される配置シナリオ

配置モード	配置シナリオの例
接続文字列を使用した非ストリーミング(デフォルト) `server.essbase.streamingCubeBuilding=false`	データ・ソースがリレーショナルで、デフォルトの ODBC DSN 接続属性値の調整に特別な条件がない場合

配置モード	配置シナリオの例
ODBC DSN を使用した非ストリーミング `server.essbase.streamingCubeBuilding=false`	接続属性のデフォルト値を変更する場合。例: ● Essbase モデルが Unicode ソース・データベースを基にしている場合、N-CHAR サポートを有効にする必要があります ● データ転送パフォーマンスを向上するためデータ圧縮を有効にする場合、ワイヤ・プロトコル・モードを 2 に設定できます ● IBM DB2 データ・ソース接続の認証メソッドとして暗号化オプションが選択されている場合 ● Windows で、Oracle BI EE データ・ソースから構築されたキューブに対して、Essbase が OPMN を使用して起動された場合
ストリーミング `server.essbase.streamingCubeBuilding=true`	例: ● Essbase Studio コンソールを使用して Performance Management Architect データ・ソースから Essbase モデルが構築されます ● Performance Management Architect を使用して Essbase アプリケーションが構築され、Essbase に配置されます ● Essbase モデルが Unicode ソース・データベースを基にしています(N-CHAR サポート) ● IBM DB2 データ・ソース接続の認証メソッドが暗号化オプションです(「パスワードの暗号化」、「ユーザー ID とパスワードの暗号化」または「クライアント」) ● Windows で、Oracle BI EE データ・ソースから構築されたキューブに対して、Essbase が OPMN を使用して起動された場合 ● 配置中の Essbase モデルにカレンダ階層が含まれている場合

日付メジャーの配置

日付メジャーが含まれるキューブを配置する場合は、Essbase Studio では結果セットが最終トランザクション日に生成されるようにするため、データ・ロード SQL の生成中に MAX 集約関数が日付メジャー列に適用されることに注意してください。

次の例では、Date 列が日付メジャーとして使用されています:

```
   Product   State   Sales   Date
-------   -----   ------   ----------
100-10   CA    200.00   2009-01-31
100-10   CA    500.00   2009-12-31
```

MAX 集約関数を使用して、データが 2009-12-31 の日付で集約され、Sales の合計は 700.00 になります。

日付メジャーが含まれるキューブを配置する際は、次のタスクを実行することをお薦めします:

- 「Essbase モデル・プロパティ」の「全般」タブで、日付フォーマットを選択します。

  デフォルトの日付フォーマットは、`yyyy-mm-dd` です。
- 「キューブ配置ウィザード」の「キューブ配置オプション」ページで、必ず「データ・ロード・オプション」グループから「既存データに上書き」を選択します。

キューブの配置

「キューブ配置ウィザード」は、次の画面で構成されています:

- 「Essbase サーバー接続オプション」 - Essbase サーバーの接続に関する情報と、データ・ソースのデータを接続および取得する手法を示します。297 ページの「キューブ配置のための接続情報の提供」を参照してください。

  注： キューブを初めて Essbase サーバーのインスタンスに配置すると、295 ページの「Essbase サーバー接続の作成」で説明するとおり、Essbase サーバーの接続情報を入力するように要求されます。この情報を入力すると、「キューブ配置ウィザード」が起動します。
- 「増分ロード」 - メンバーの増分ロードを選択すると、更新するメンバーを選択できます。304 ページの「キューブ配置の増分ロードの設定」を参照してください。
- 「キューブ配置オプション」 - ロード・オプションを設定し、拒否されたレコードの処理を指定して、迅速な配置をスケジューリングするか、入力した配置情報を MaxL スクリプトとして保存することを選択します。299 ページの「配置オプションの設定」を参照してください。

また、307 ページの「配置履歴の表示」で説明されているように、Essbase モデルごとにキューブの配置履歴を確認することもできます。

Essbase サーバー接続の作成

キューブを Essbase サーバーのインスタンスに初めて配置する場合、Essbase の接続情報が求められます。

➤ Essbase サーバーへの接続を作成するには:

1 **「メタデータ・ナビゲータ」で、キューブの配置元となる Essbase モデルを選択します。**

2 **モデル名を右クリックして、「キューブ配置ウィザード」を選択します。**

再利用可能な Essbase への接続を作成するための「Essbase ログイン」ダイアログ・ボックスが表示されます。

注： すでに既存の Essbase 接続が存在する場合、「Essbase ログイン」ダイアログ・ボックスではなく「キューブ配置ウィザード」が表示されます。

3 **「Essbase ログイン」で、次の情報を入力し、「ログイン」をクリックします:**

- この接続の「名前」
- オプションの「説明」。
- Essbase インスタンスまたはクラスタがある「サーバー」の名前、および「ポート」の番号。

  「サーバー」および「ポート」のデフォルトは、Essbase Studio と同じマシンで実行されている Essbase のインスタンスまたはクラスタです。

  注： クラスタ名では大文字と小文字が区別されます。デフォルトのクラスタ名は `EssbaseCluster-1` です。
- この Essbase サーバーへのアクセスに必要な「ユーザー」および「パスワード」の名前。
- 「データ暗号化」方法:
  - 「暗号化なし」 - Essbase Studio サーバーと Essbase サーバー・インスタンスまたはクラスタ間の通信のデフォルト・モード。
  - 「SSL」 - 接続中の Essbase サーバー・インスタンスまたはクラスタが、TCP/IP 経由の Secure Socket Layer (SSL)プロトコルに対応するよう構成されている場合はこの方法を選択します。

    SSL を使用すると、Essbase Studio サーバーと Essbase 間の通信が保護されます。

    注： Essbase サーバーが SSL をサポートするように構成されていない場合は、接続が確立されず、エラー・メッセージが表示されます。
- Essbase サーバー・クラスタに接続している場合は、「クラスタ」を選択します。

  注： Essbase サーバー・クラスタは、Essbase 接続の作成前に設定しておく必要があります。クラスタ設定の詳細は、『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 高可用性ガイド』を参照してください。

接続名は、「キューブ配置ウィザード」の「Essbase サーバー接続」フィールドに今すぐ入力したり、後で配置するときに入力したりできます。

キューブ配置のための接続情報の提供

「キューブ配置ウィザード」の「Essbase サーバー接続オプション」ページを使用して、Essbase およびデータ・ソース接続情報を指定します。また、配置前に Essbase モデル・プロパティも変更できます。

➤ キューブ配置のサーバー、データ・ソース、およびモデル・プロパティ情報を設定するには:

1 **「メタデータ・ナビゲータ」で、キューブの配置元となる Essbase モデルを選択します。**

2 **モデル名を右クリックして、「キューブ配置ウィザード」を選択します。**

「Essbase サーバー接続オプション」ページが起動されます。

3 **「Essbase サーバー接続」で、配置先とする Essbase サーバーを表す接続を選択します。**

リストにない Essbase サーバー接続に配置するには、「新規接続」をクリックし、295 ページの「Essbase サーバー接続の作成」の手順を終了してから、手順 4 に進みます。

4 **配置先とする「アプリケーション」および「データベース」の名前を入力します。**

注 : XOLAP 対応の Essbase モデルから配置する場合、新しいアプリケーションおよびデータベース名を使用するか、「最初にすべてのメンバーを削除」オプションを使用して既存の XOLAP アプリケーションを上書きして配置することを強くお薦めします。

5 **オプション: 「キューブ配置でストリーミング・モードを有効化」チェック・ボックスを選択して、このキューブ配置をストリーミング・モードで実行します。**

注 : 「キューブ配置でストリーミング・モードを有効化」チェック・ボックスの設定は、異なる配置にわたって保持されません。ストリーミング・モード用のチェック・ボックスを使用するには、キューブ配置を実行するたびにそれを設定する必要があります。

ストリーミング・モードとは、キューブ配置時に Essbase Studio サーバーが (ODBC 接続を使用して外部データ・ソースをクエリーするのではなく)外部データ・ソースを直接クエリーすることを意味します。

配置対象の Essbase モデルに 1 つ以上のリレーショナル・データ・ソースが含まれている場合、「キューブ配置でストリーミング・モードを有効化」チェック・ボックスが有効になります。

Essbase モデルで使用されているデータ・ソースが 1 つ以上のテキスト・ファイル・ソース、1 つ以上の次元サーバー(Performance Management Architect)ソー

ス、またはテキスト・ファイルとリレーショナル・ソースの混合である場合、このチェック・ボックスは無効になります。

Essbase Studio サーバーが非ストリーミング・モードで実行されている場合(`server.essbase.streamingCubeBuilding` プロパティが FALSE に設定されている場合)、このオプションを選択すると、その設定が上書きされます。

6 **「データ・ソースの設定」グループで、オプションを選択します:**

- 「ODBC (Essbase は ODBC 接続文字列を動的に作成)」 - Essbase によって動的に作成されたデータ・ソース名が使用されます。これがデフォルトです。
- 「ODBC (ODBC DSN 名を入力)」 - 使用する ODBC DSN 名を入力します。

  独自のカスタム ODBC DSN パラメータ設定を利用するために ODBC DSN 名を使用して配置する場合は、次のガイドラインに従います:

  - 配置を開始する前に、Essbase がインストールされているサーバー・マシンで、ODBC DSN を設定します。
  - ODBC DSN のユーザー名とパスワードは、この配置で使用されるデータ・ソース接続と同一であることが必要です。

  注: 「server.essbase.streamingCubeBuilding」プロパティとともに ODBC 設定を使用する方法の詳細は、292 ページの「配置シナリオとストリーミング・キューブ・ビルディング・プロパティの注意点」を参照してください。
- 「OCI (OCI 接続識別子を入力)」 - 使用する Oracle OCI 接続識別子を入力します。

  Oracle OCI 接続識別子の構文は、次のとおりです:

  ```
  host:port/SID
  ```

  次に、ホスト・サーバー名が `myserver`、ポート番号が `1521`、および Oracle SID (サービス識別子)が `orcl` の OCI 接続識別子の例を示します:

  ```
  myserver:1521/orcl
  ```

  注: その他の接続識別子フォーマットを使用することも可能です。OCI の詳細は、Oracle のドキュメントを参照してください。

ODBC DSN または OCI 接続識別子を呼び出すことで、DSN または OCI で設定した特定のパラメータを利用できます。たとえば、キューブ配置中に、ドライバ・パフォーマンスおよびフェイルオーバー・オプションを利用するために設定されている Oracle Wire Protocol ドライバを使用する場合があります。

次の条件に当てはまる場合は、「データ・ソースの設定」グループ内のすべてのアイテムが無効になります:

- 「キューブ配置でストリーミング・モードを有効化」チェック・ボックスが選択されています(手順 5)。
- Essbase Studio サーバーがストリーミング・モードで実行されています(`server.essbase.streamingCubeBuilding` プロパティが TRUE に設定されています)。
- Essbase モデルに複数のデータ・ソースからの要素が含まれています
- Essbase モデルがテキスト・ファイル・データ・ソースを使用して作成されました
- Essbase モデルが次元サーバー(Performance Management Architect)データ・ソースを使用して作成されました

7 **オプション: モデルの「Essbase」プロパティを変更する場合は、「モデル・プロパティ」をクリックします。**

第 11 章「Essbase プロパティ」を参照してください。

8 **「次へ」をクリックして、ウィザードの「キューブ配置オプション」ページを表示します。**

注 : エラーがモデル内に存在する場合、「次へ」をクリックすると警告が示されます。「Essbase モデル・プロパティ」ダイアログ・ボックスを起動するかどうかをたずねるダイアログが表示されます。「はい」をクリックすると、プロパティ・ダイアログ・ボックスが起動し、エラーを修正することができます。

配置オプションの設定

「キューブ配置ウィザード」の「キューブ配置オプション」ページで、ロード情報とレコード拒否の設定を指定します。また、後で配置を行う場合のために、配置情報を MaxL スクリプトに保存することを選択できます。

➤ 配置オプションを指定するには:

1 **「キューブ配置ウィザード」の「キューブ配置オプション」ページで、「ロード・タスク・タイプ」グループからロード・オプションを選択します:**

- 「アウトラインの構築」-次元 Essbase アウトラインに追加します。
- 「データのロード」-Essbase データベースにデータを移入します。データをロードすると、データベースの構造アウトラインで定義されたセルに実際の値が設定されます。
- 「アウトラインの構築およびデータのロード」-データのない次元およびメンバーを Essbase アウトラインに追加し、Essbase データベースにデータを移入します。

2 **「データ・ロード・オプション」グループで、次のいずれかのオプションを選択します:**

- 「既存データに追加」-データ・ソースの値をキューブの既存の値に加算します。

- 「既存データから削除」-データ・ソースの値をキューブの既存の値から減算します。
- 「既存データに上書き」-キューブの値をデータ・ソースの値で置き換えます。

注： これらのオプションは、手順 1 でデータ・ロード・オプションを選択した場合にのみ使用可能になります。

3 **オプション: 既存の「Essbase」アウトラインに含まれるすべての次元とメンバーを削除する場合は、「最初にすべてのメンバーを削除」を選択します。**

すべてのメンバーを削除する場合、Essbase Studio は Essbase データベース・アウトラインからすべてのメンバーを削除してから、キューブ・スキーマのメンバー・レベルを使用してアウトラインを再作成します。すべてのメンバーを削除すると、すべてのメンバーを削除せずに Essbase アウトラインの作成または更新を行う場合よりも処理時間がかかる可能性があるため、特に理由がないかぎり、このオプションは使用しないでください。すべてのメンバーを削除する必要があるのは、たとえば、一部のメンバーが Essbase モデルから削除されたことがわかっていて、小さくしたメンバー・セットを含むより小さい Essbase アウトラインを構築する場合などです。

注： このチェック・ボックスは、データのロードのみを選択した場合は選択できません。

4 **オプション: メンバーのロードまたはメンバーとデータのロードを実行する前に Essbase データベース内のすべてのメンバーとデータを削除するには、「データベースを削除して復元」を選択します。**

このアクションにより、アウトラインの構築が行われる前に、メンバーとデータの Essbase データベース・アウトラインが消去され、ロードの所要時間が大幅に短縮されます。

注： このチェック・ボックスは、データのロードのみを選択した場合は、選択できません。

5 **オプション: 「Essbase」アウトライン内で特定の次元またはメンバーが更新されるように選択するには、「増分ロード」を選択します。**

このチェック・ボックスは、データのロードのみを選択した場合は選択できません。アウトラインの作成時に、「最初にすべてのメンバーを削除」オプションまたは「データベースを削除して復元」オプションを選択した場合にも選択できません。

注： このオプションを選択したら、「次へ」ボタンをクリックして、304 ページの「キューブ配置の増分ロードの設定」のタスクを実行します。

6 **オプション: 管理サービス・コンソールからメンバーおよびデータをロードするために後で使用するには、「ルール・ファイルのみ作成して保存」を選択します。**

Essbase Studio により生成されるルール・ファイルには、Essbase データベースへのデータのロード中に、Essbase がメンバーとデータ・ソースのデータに対して行う必要がある変更内容を指定する SQL 文が含まれています。

その際、データ・ソースは変更されません。

ルールおよびルール・ファイルについて

ルールは、Essbase がデータ・ソースを処理するときにデータ値、または次元とメンバーに対して実行する操作を定義します。ルールを使用して、データ値を Essbase データベースに、または次元とメンバーを Essbase アウトラインにマッピングします。

ルールはルール・ファイルに保管されます。Essbase Studio によって生成されるルール・ファイルには、使用する構築方法、データ値またはメンバーの順序をソートするかランダムのままにするか、およびロード前に行うデータ値またはメンバーの変換方法を定義する SQL 文が含まれます。Essbase Studio は次元ごとに別々のルール・ファイルを作成します; ただし、メジャー次元のメンバーが 250 を超える場合には、Essbase Studio は 250 のメンバーごとに 1 つのルール・ファイルを作成します。

ルール・ファイルは Essbase のインストール場所の app ディレクトリに保存されます。

詳細は、374 ページの「管理サービス・コンソールでのルール・ファイルの制限」を参照してください。

7 **オプション:「バックグラウンドでアウトラインを構築」を選択します。**

次のアクションを実行した場合にのみ、「バックグラウンドでアウトラインを構築」オプションを選択できます。

- XOLAP キューブの再配置
- アウトラインの構築のみ

次のアクションを実行した場合、このオプションは選択できません:

- XOLAP キューブの最初の配置
- XOLAP 以外のキューブの再配置
- データのロード

「バックグラウンドでアウトラインを構築」を選択した場合、「データ・ロード・オプション」グループからは選択できません。「最初にすべてのメンバーを削除」、「データベースを削除して復元」、「増分ロード」および「ルール・ファイルのみ作成して保存」オプションも選択できません。

8 **「レコード設定が拒否されました」で、次のタスクを実行します:**

1. 「維持するレコード数」で次のオプションを選択します:
   - 「すべて」-拒否されたすべてのレコードをエラー・ファイルに保持
   - 「制限」-右のテキスト・ボックスに指定した数の拒否されたレコードをエラー・ファイルに保持します。デフォルトでは、200 レコードが保持されます。
2. 「エラー・ファイル名」で、次のいずれかのオプションを選択します:
   - 「デフォルト」-エラー・ファイルにデフォルトのファイル名を付けます。デフォルトのファイル名は次のフォーマットになります:

```
app_name
.
db_name
_
timestamp
.err
```

例:

```
myesbapp.myesbdb_05-02-08_11_30-38.err
```

- 「ファイル名」-カスタム・ファイル名を指定します。たとえば、エラー・ファイルに errors.err という名前を付けることができます。

  また、エラー・ファイルを別の場所に置くように指定することもできます。デフォルトでは、エラー・ファイルは次の場所に置かれます:

  ```
  EPM_ORACLE_HOME
  /products/Essbase/EssbaseStudio/server/ess_japihome/data
  ```

  エラー・ファイルをデフォルト以外の場所に置くには、「ファイル名」テキスト・ボックスに相対パスを入力します。たとえば、エラー・ファイルに errortest という名前を付け、ess_japihome/data ディレクトリの下にある errors というフォルダに置くには、次のように入力します:

  ```
  errors/errortest.err
  ```

  エラー・ファイルに testerrors という名前を付け、Essbase Studio server ディレクトリの下にある errorlogs というフォルダに置くには、次のように入力します:

  ```
  ../../errorlogs/testerrors.err
  ```

3. **オプション:** ロード・プロセス中、Essbase で最初のエラーが発生したらデータ・ロードを停止するには、「最初のレコードが拒否された場合はデータ・ロードを停止します」を選択します。

   このオプションは、「データのロード」オプションまたは「アウトラインの構築およびデータのロード」オプションが選択されている場合に使用可能になります。

9 **「スケジュール・オプション」で、次のアクションを実行します:**

- キューブをすぐに配置する場合は、「すぐに配置」が選択されていることを確認します。

  「すぐに配置」オプションは、デフォルトで選択されています。

- 配置情報を含む MaxL ロード・スクリプトを作成するには、「MaxL ロード・スクリプトとして保存」を選択し、「参照」をクリックして MaxL スクリプトを保存する場所に移動し、「保存」をクリックします。

  Essbase Studio で生成される MaxL の操作の説明は、373 ページの「キューブ配置の制限とガイドライン」を参照してください。

注： 「スケジュール・オプション」グループから、少なくとも 1 つのオプションを選択する必要があります。

10 **次のいずれかのアクションを行います:**

- 「増分ロード」オプションを選択した場合、「次へ」をクリックして、304 ページの「キューブ配置の増分ロードの設定」のタスクを実行します。
- 「終了」をクリックして、キューブを配置します。

  配置が成功すると、配置のステータス情報が示された「配置は成功しました」ダイアログ・ボックスが表示されます。

  注： キューブの配置中にメモリー不足エラーが発生した場合は、78 ページの「仮想メモリーの構成」の説明に従って、Essbase Studio コンソールの仮想メモリー設定を増やします。

  「メタデータ・ナビゲータ」に、配置されたキューブが、配置元の Essbase モデルの子として表示されます。

  配置後に、管理サービス・コンソールでアウトラインを表示して編集します。また、Smart View を使用してキューブのクエリーを実行することもできます。

  配置情報を MaxL スクリプトに保存する場合、スクリプトでは、シェル変数の$1 および$2 が、名前とパスワードのパラメータにそれぞれ使用されます。

  例:

```
        deploy all from model 'cs1Model' in cube schema '\CubeSchemas\cs1'
login $1 identified by $2 on host 'poplar-pc1' to application 'cs2' database
'cs2' add values using connection 'Connection1' keep 200 errors on error
ignore dataload write to default;
```

  上の例でこのスクリプトを使用する際は、Essbase ログイン情報のコマンド・ラインで次のように次のように置き換えます:

  - $1 = ユーザー名
  - $2 = パスワード

キューブ配置の増分ロードの設定

「キューブ配置ウィザード」の「増分ロード」ページで、更新する特定の次元またはメンバーを選択します。

➤ メンバー・ロード更新オプションを設定するには:

1 **「キューブ配置ウィザード」の「増分ロード」ページにある「次元構築更新」の下で、オプションを選択します:**

- 「すべての階層を更新」-モデルのすべての階層が更新され、新しいメンバーがあれば追加されます。

  このオプションを選択すると、すべての階層が自動的に更新対象として選択され、「更新」、「再構築」および「消去」ボタンは使用できません。

- 「選択した階層を更新または再構築」-このオプションを選択すると、階層の選択がすべて解除されます。次いで、階層を選択し、階層に対して実行する操作を次の中から 1 つ選択します:

  - 「更新」-新しいメンバーを階層に追加します。既存のメンバーを削除しないでください。

    このオプションは、階層構造を変更せずに新しいメンバーを追加する場合、または共有メンバーを追加する場合に使用します。更新中、既存の階層が、既存のメンバーを削除せずに更新されます。

    「更新」を選択すると、階層の各メンバーの横に、そのメンバーが更新対象としてマークされていることを示すアイコンが表示されます。各メンバーの横にあるチェック・ボックスはまだ選択されていません。これらのチェック・ボックスは手動で選択する必要があります。

  - 「再構築」-階層のすべてのメンバーを消去して、共有メンバーを含むすべてのメンバーを追加し直します。必要に応じて階層を再構築します。

    このオプションは、階層からメンバーを削除した場合に特に便利です。まだ存在しているメンバーと、新しいメンバーがあれば階層に追加し直され、必要に応じて階層が再構築されます。

    「再構築」を選択すると、階層の各メンバーの横に、メンバーが再構築対象としてマークされていることを示すアイコンが表示されます。各メンバーの横にあるチェック・ボックスはまだ選択されていません。これらのチェック・ボックスは手動で選択する必要があります。

  - 「消去」-現在のセッション中に選択した内容を消去するには、「消去」ボタンを使用します。誤って階層をマークし、その階層のチェック・ボックスを選択した場合は、「消去」をクリックして選択内容を消去します。

    注: 「消去」ボタンは、メンバーの横にあるチェック・ボックスが選択されている階層にのみ有効です。

2 **メンバー構築時にデータベースを再構築するためのパラメータを設定するには、「保持」オプションを選択します。**

- 「すべての既存データ」-この設定はデフォルトです。再構築が行われるときに変更されたアウトラインに該当する既存データをすべて保持します。
- 「入力データ」-ロードされるデータを含むブロックのみを保持します。多くのアプリケーションには、親レベルで入力されたデータが含まれています。このオプションを選択すると、レベル 0(リーフ・ノード)ブロックであるかどうかにかかわらず、データ・ロードによって作成されたブロックが削除されるのを防ぐことができます。
- 「レベル 0 のデータ」-レベル 0 のメンバーのデータのみを保持します。これは、ソース・データベースを変更してデータの再計算が必要な場合、および計算に必要なデータすべてがレベル 0 のメンバー内にある場合に最適な再構築オプションです。このオプションを選択すると、上位レベルのブロックはすべて再構築の前に削除されます。これにより再構築のためのディスク・スペースが削減され、データベースを再計算するときの計算時間が短縮されます。上位レベルのブロックはデータベースの計算時に再作成されます。上位レベルのブロックはデータベースの計算時に再作成されます。
- 「データなし」-すべてのデータをデータベースから消去します。

3 **次のいずれかのアクションを行います:**

- 「終了」をクリックして、キューブを配置します。

  配置が成功すると、「配置は成功しました」ダイアログ・ボックスが表示され、配置の詳細が示されます。

  「メタデータ・ナビゲータ」に、配置されたキューブが、配置元の Essbase モデルの子として表示されます。

  これにより、管理サービス・コンソールでアウトラインを表示して編集できるようになりました。また、Smart View を使用してキューブのクエリーを実行することもできます。
- 配置を MaxL ロード・スクリプトとして保存する場合は、「次へ」をクリックし、299 ページの「配置オプションの設定」のタスクを実行します。

配置の進捗状況および結果の表示

キューブの配置を起動すると、ウィンドウが表示され、配置の情報および統計が示されます。

このウィンドウには、配置の進捗状況に関する情報が表示されます。配置が完了すると、ウィンドウに次の情報が表示されます:

- 配置の開始および完了時間
- 配置の経過時間
- 配置のランタイム統計:
  - 処理されたレコード数
  - 拒否されたレコード数(ある場合)
- エラー・ファイルの場合(拒否されたレコードがある場合)
- 拒否されたレコードのリスト(ある場合)

注意

表示される情報および統計は、Essbase Studio サーバーがストリーミング・モードと非ストリーミング・モードのいずれで実行されているかによって異なります。

- Essbase Studio サーバーが非ストリーミング・モードで実行されている場合、処理されたレコードと拒否されたレコード(ある場合)の両方について、配置のランタイム統計が画面上に表示されます。

  たとえば、配置の後、画面には次のように表示されます:

  ```
  Records processed 17. records rejected: 3.
  ```

  非ストリーミング・モードの場合、サーバー・プロパティ `server.essbase.streamingCubeBuilding` が `false` に設定されます。

- Essbase Studio サーバーがストリーミング・モードで実行されている場合、配置のランタイム統計は、処理されたレコードについてのみ画面に表示されます。拒否されたレコードがある場合、その統計は画面に表示されません。配置の終了後、エラー・ファイルに、拒否されたレコードのリストが含まれます。

  たとえば、配置の後、画面には次のように表示されます:

  ```
  Records processed 17.
  ```

  ストリーミング・モードの場合、サーバー・プロパティ `server.essbase.streamingCubeBuilding` が `true` に設定されます。

このプロパティの詳細は、42 ページの「server.essbase.streamingCubeBuilding」を参照してください。

14 配置済アプリケーション

この章の内容

配置履歴の表示 307
キューブ・リンケージの更新 308

配置履歴の表示

Essbase モデルの配置履歴を表示して、Essbase サーバーの様々なインスタンスに配置されたすべての子キューブのリストを表示できます。

➤ Essbase モデルに関連するキューブの配置履歴を表示するには:

1 **「メタデータ・ナビゲータ」で、Essbase モデルに移動します。**

2 **モデル名を右クリックして、「配置履歴の表示」を選択します。**

Essbase Studio コンソールの作業領域に「配置履歴: <Essbase Model Name>」タブが表示され、グリッド・フォーマットで各配置がリストされます。

表示される情報は、次のとおりです:

- 「ステータス」 - 配置が成功したか失敗したかを表示します。
- 「時間」 - 配置の日時です。

  複製のオプションが選択されている再ホストされた Essbase サーバーの場合、配置の最初の日時を含む最後の正常な配置履歴のコピーが、再ホストの日時を含む更新されたコピーとともにリストされます。
- 「Essbase サーバー」 - キューブの配置先となった Essbase サーバー・インスタンスの名前です。

  複製のオプションが選択されている再ホストされた Essbase サーバーの場合、配置の元の host:port 情報を含む最後の正常な配置履歴のコピーが、新しい host:port 情報を含む更新されたコピーとともにリストされます。
- 「アプリケーション」 - キューブの Essbase アプリケーション名です。
- 「データベース」 - キューブの Essbase データベース名です。
- 「タイプ」 - 配置がメンバー・ロード、データ・ロードまたはその両方であったかどうかを示します。
- 「ユーザー」 - キューブを配置したユーザーの名前です。

3 **グリッド内の配置を表す行をダブルクリックして、「配置エラーおよび警告」ダイアログ・ボックスを表示します。**

配置時に記録されたエラーまたは警告がリストされます。

4 「OK」または「取消し」をクリックして、「配置エラーおよび警告」を閉じます。

5 終了したら、「配置履歴」ウィンドウのタブにある「X」をクリックして閉じます。

キューブ・リンケージの更新

このトピックには、次のサブトピックが含まれています:

- 308 ページの「キューブ・リンケージについて」
- 309 ページの「個々のキューブのキューブ・リンケージを更新する前に」
- 309 ページの「個々のキューブのキューブ・リンケージの更新」
- 311 ページの「Essbase Studio の再ホスト後のキューブ・リンケージの更新」

キューブ・リンケージについて

キューブ・リンケージは、Essbase キューブに格納される次の情報を意味します:

- キューブの配置に使用される Essbase Studio サーバーとポート
- キューブのベースとなったモデル
- キューブ構築の基になったデータ・ソースのログイン情報

Essbase Studio を使用して構築されたキューブでは、キューブ・リンケージ情報を更新して、Essbase Studio サーバー、ポートおよびモデル・コンテキストを変更できます。

キューブ・リンケージの使用例:

- ドリルスルー・レポートを新しい Essbase Studio サーバーにリダイレクトする場合や、ドリルスルー・レポートが別の Essbase モデルに基づいている場合。これは、便利なロード・バランシング・ツールとなります。
- データ・ソースのログイン情報が変更される場合、キューブ・リンケージを更新できることは XOLAP 対応キューブには非常に便利です。キューブ・リンケージを更新すると、データ・ソースのログイン情報が更新されます。モデルの再作成やキューブの再配置の必要はありません。

  このような場合にキューブ・リンケージを更新するには、309 ページの「個々のキューブのキューブ・リンケージを更新する前に」および309 ページの「個々のキューブのキューブ・リンケージの更新」の手順を実行してください。
- Essbase Studio サーバーが(アップグレード後などに)異なるマシンで再ホストされると、再ホストされた Essbase Studio サーバーへの参照の更新プロセスはキューブ・リンケージを使用して処理されます。

  Essbase Studio サーバーを再ホストした場合は、311 ページの「Essbase Studio の再ホスト後のキューブ・リンケージの更新」の手順に従います。

注： 311 ページの「Essbase Studio の再ホスト後のキューブ・リンケージの更新」の手順のステップを実行する前に、Essbase の再ホスト手順を必ず実行してください。詳細は、81 ページの「再ホストされた Essbase サーバーへの参照の更新」を参照してください。

EPM System 製品の再ホストに関する完全な情報については、Oracle Enterprise Performance Management System インストールおよび構成ガイドを参照してください。

個々のキューブのキューブ・リンケージを更新する前に

➤ キューブ・リンケージの更新手順を完了する前に、次のアクションを実行します:

1 **Essbase Studio サーバーのマシン名およびポート番号の情報を確認します。**

2 **入力する Essbase Studio サーバー・インスタンスの関連するカタログに、リンク先となる Essbase モデルの名前が含まれていることを確認します。**

3 **キューブの作成に使用するデータ・ソースのログイン情報が変更されている場合、118 ページの「データ・ソース接続プロパティの編集」の説明に従って、適切なデータ・ソース接続の「プロパティ」ダイアログ・ボックスでログイン情報を更新します。**

個々のキューブのキューブ・リンケージの更新

この手順を開始する前に、309 ページの「個々のキューブのキューブ・リンケージを更新する前に」の手順を実行してください。

➤ Essbase Studio を使用して構築されたすべてのキューブのキューブ・リンケージを更新するには:

1 **Essbase Studio コンソールで、「ツール」、「キューブ・リンケージの更新」の順に選択します。**

2 **「キューブ・リンケージの更新」で、「選択した Essbase アプリケーションおよびデータベースの更新」が選択されていることを確認します。**

このデフォルトの選択内容は、(この手順の後続のステップで)指定する新しい Essbase Studio サーバーを、選択した Essbase キューブにのみ適用します。

3 **「Essbase 接続」で該当する Essbase 接続とアプリケーションを展開し、リンケージを変更する Essbase キューブを選択します。**

リンケージを更新できるのは、Essbase Studio によって構築されたキューブのみです。「キューブ・リンケージ Essbase Studio サーバー」および「キューブ・リンケージ Essbase モデル」の下にエントリのあるキューブは Essbase Studio によって構築されており、更新が可能です。これらの列にエントリがないキューブは更新できません。

4 「更新」をクリックして、「キューブ・リンケージ」ダイアログ・ボックスを起動します。

5 新しい「Essbase Studio サーバー」の名前およびポート番号を次のフォーマットで入力します:

```
essbase_studio_server_name
:
port_number
```

例:

```
aspen3:5300
```

ヒント： 以前に入力したサーバー名とポート番号の組合せは Essbase Studio によって記録されています。必要な組合せがドロップダウン・リストに表示されている場合は、選択できます。

注： Essbase Studio では、入力するサーバー名とポート番号の組合せは検証されません。この情報を間違って入力してもエラー・メッセージが表示されず、キューブ配置を実行するまでエラーは表示されません。

注： データ・ソースのログイン情報の変更のみがキューブ・リンケージを更新する理由である場合、新しい Essbase Studio サーバーの名前またはポートを入力する必要はありません。新しいサーバーまたはポートを指定する場合にのみ新しいサーバー名またはポートを入力します。

6 リストから「Essbase モデル名」を選択します。

Essbase モデル名は手動でも入力できます。入力する Essbase モデル名が、手順 5 で入力した Essbase Studio サーバーに関連付けられているカタログに存在することを確認します。

注： Essbase Studio では、入力したモデル名は検証されません。この情報を間違って入力してもエラー・メッセージが表示されず、キューブ配置を実行するまでエラーは表示されません。

注： データ・ソースのログイン情報の変更のみがキューブ・リンケージを更新する理由である場合、新しいモデル名を入力する必要はありません。新しいモデルを指定する場合にのみ新しいモデル名を入力します。

7 「OK」をクリックして、「キューブ・リンケージの更新」に戻ります。

8 選択したキューブで、「キューブ・リンケージ Essbase Studio サーバー」および「キューブ・リンケージ Essbase モデル」のエントリが正しいことを確認します。

9 更新が必要なすべてのキューブについて、手順 3 から手順 8 を繰り返します。

10 終了するには、「閉じる」をクリックします。

Essbase Studio の再ホスト後のキューブ・リンケージの更新

Essbase Studio サーバーを(アップグレード後などに)再ホストした場合は、このトピックの手順に従います。

Oracle Enterprise Performance Management System 製品の再ホストに関する完全な情報については、Oracle Enterprise Performance Management System インストールおよび構成ガイドを参照してください。

注： この手順のステップを実行する前に、Essbase の再ホスト手順を必ず実行してください。詳細は、81 ページの「再ホストされた Essbase サーバーへの参照の更新」を参照してください。

➤ Essbase Studio の再ホスト後に、Essbase Studio を使用して構築されたキューブのキューブ・リンケージを更新するには、次の手順に従います:

1. **Essbase Studio コンソールで、「ツール」、「キューブ・リンケージの更新」の順に選択します。**
2. **「すべての Essbase アプリケーションおよびデータベースを更新して現在の Essbase Studio サーバーにリンク(すべての Essbase インスタンスの起動が必要)」を選択します。**

   すべての Essbase アプリケーションおよびデータベースが、現在接続している Essbase Studio サーバーにリンクされます。

   注： キューブ・リンケージの更新を反映するには、すべての Essbase インスタンスまたはクラスタが実行されている必要があります。
3. **オプション: 更新後もすべての Essbase アプリケーションを実行しておく場合には、「更新後にすべての Essbase アプリケーションを停止」チェック・ボックスの選択を解除します。**

   デフォルトでは、更新後にすべての Essbase アプリケーションが停止されます。
4. **「更新」をクリックします。**

   キューブ・リンケージが正常に更新されたことを示すメッセージが表示されます。

   失敗した場合は、エラー・メッセージに詳細が示されます。
5. **終了するには、「閉じる」をクリックします。**

15 系列

この章の内容

系列の概要 313
系列作業領域を開く 315
系列作業領域の使用 315

系列の概要

系列作業領域は、メタデータ要素の系列のグラフィカル表示です。これにより、メタデータ要素間の関係をカタログで確認できます。

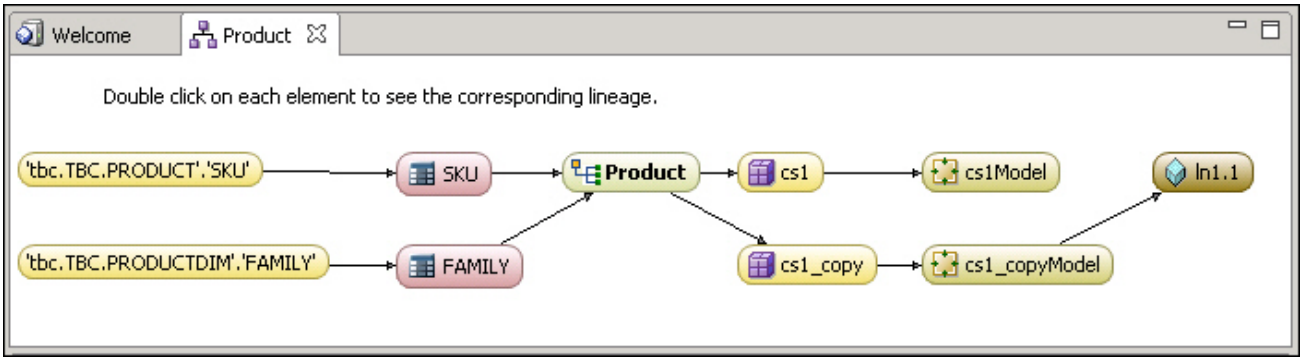
系列作業領域により、要素の導出元となっているデータ・ソースのテーブルおよび列から、要素を使用している配置済キューブまで、幅広い情報を表示できます。系列作業領域は、要素の変更による潜在的な影響を簡単に把握するのに役立ちます; たとえば、階層をキューブ・スキーマから削除する場合です。また、要素ごとに履歴を確認することもできます。

系列作業領域には、選択した要素に関連する要素のチェーンが表示されます。たとえば、標準的な階層の場合、次のような関連オブジェクトが表示されます:

- 階層が構築された各列のキャプション・バインディングおよびキー・バインディングを含む階層列
- ユーザー定義のメンバーを含む階層内のメンバー
- 階層を使用するキューブ・スキーマ
- 階層を使用するキューブ・スキーマから作成される Essbase モデル
- Essbase モデル(階層を使用するキューブ・スキーマから作成)から配置された Essbase キューブ

図 17 に、サンプルの系列ダイアグラムを示します。

図 17　Product という名前の階層の系列



系列は、次元要素のキャプション・バインディングとキー・バインディングが異なることも示します。図 18 に、キャプション・バインディングとキー・バインディングが異なる次元要素を示します。

図 18　キー・バインディングとキャプション・バインディングが異なる系列

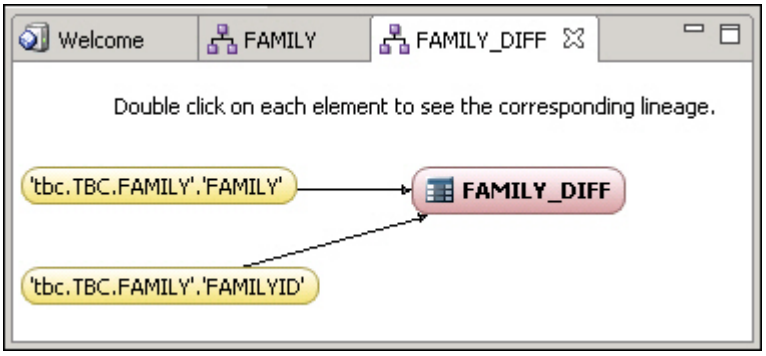


図 19 に、キャプション・バインディングとキー・バインディングが同一であるか、要素のキー・バインディング式が「遅延」に設定されている次元要素を示します。

図 19　キー・バインディングとキャプション・バインディングが同一の系列またはキー・バインディングが「遅延」に設定されている系列

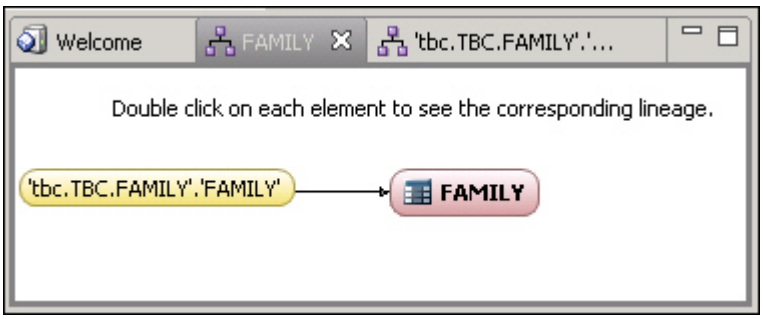


図 19 の説明は、要素のキー・バインディングが「遅延」であることも意味しています。FAMILY 列を右クリックしてポップアップ・メニューに「キャプション・バインディングを使用してサンプルを表示」オプションのみが表示される場合、要素のキー・バインディング式は「遅延」に設定されています。右クリックして「サンプル・データの表示」オプションのみが表示される場合は、キャプション・バインディング式とキー・バインディング式は同一です。

要素の系列を表示するには、315 ページの「系列作業領域を開く」を参照してください。

系列作業領域で実行できるオプションのタスクの詳細は、315 ページの「系列作業領域の使用」を参照してください。

系列作業領域を開く

系列作業領域は、メタデータ要素の系列のグラフィカル表示です。これにより、メタデータ要素間の関係をカタログで確認できます。

➤ メタデータ要素の系列を表示するには、「メタデータ・ナビゲータ」でメタデータ要素を右クリックし、「系列の表示」を選択します。

要素の系列が、Essbase Studio コンソールの中央ペインである系列作業領域に表示されます。

系列作業領域で実行できるオプションのタスクの詳細は、315 ページの「系列作業領域の使用」を参照してください。

系列作業領域の使用

系列作業領域では、次のタスクを実行できます:

- **系列内の移動**-系列作業領域の右下にあるサムネイル・ビューアのアイコン をクリックします。系列作業領域の右下角の「サムネイル・ビューア」には、メイン系列作業領域内の系列と同じ部分の系列が縮小フォーマットで表示されます。小さい方の透明な青いポインタがミニスキーマの部分に該当します。サムネイル・ビューアを使用して、フォーカスするポイントまで系列のサムネイル・ビュー内を移動します。
- **要素の履歴表示**-要素の上にカーソルを置くと、ツール・チップに要素の簡単な履歴が表示されます。この履歴には、要素の作成日、作成者および変更日が含まれます。

  注： 要素の上にカーソルを置いた状態にすると、連結名が完全に表示されます。
- **1 つの要素にフォーカス**-1 つのみの要素の系列を確認するには、対象の要素をダブルクリックします。
- **要素の移動** - 要素を移動するには、対象の要素を左クリックして、目的の場所までドラッグします。
- **複数の要素の同時移動**-一度に複数の要素を移動するには、該当する複数の要素を含むように左クリックしてドラッグします。次に、選択された要素のいずれかを左クリックして目的の場所までドラッグします。選択されているすべての要素がその場所まで移動されます。
- **元に戻す**-要素の移動を元に戻すには、系列作業領域を右クリックして「元に戻す」を選択します。

- **やり直し**-要素の移動をやり直すには、系列作業領域を右クリックして「やり直し」を選択します。
- **ズーム・イン**-拡大された系列のビューを取得するには、系列作業領域を右クリックして「ズーム・イン」を選択します。
- **ズーム・アウト**-縮小された系列のビューを取得するには、系列作業領域を右クリックして「ズーム・アウト」を選択します。
- **プロパティの編集**-作業領域を右クリックして、「プロパティ」を選択します。

  要素を編集するダイアログ・ボックスが表示されます。
- **Essbase プロパティの編集**-Essbase モデルのプロパティを編集するには、モデルを右クリックして「Essbase プロパティ」を選択します。

  要素を編集するダイアログ・ボックスが表示されます。
- **配置履歴の表示**-Essbase モデルの配置履歴を確認するには、モデルを右クリックして「配置履歴の表示」を選択します。
- 「サンプル・データの表示」-選択した要素がどのように構築されているかに応じて、使用可能なサンプル表示オプションが異なります。要素のサンプル・データを表示するには、要素を右クリックします。

  選択した要素のキャプション・バインディング式とキー・バインディング式が同一の場合は、「サンプル・データの表示」のみが表示されます。

  選択した要素のキャプション・バインディング式とキー・バインディング式が異なる場合は、「キャプション・バインディングを使用してサンプルを表示」と「キー・バインディングを使用してサンプルを表示」の両方のメニュー項目が表示されます。

  要素のキー・バインディング式が「遅延」の場合は、「キャプション・バインディングを使用してサンプルを表示」メニュー項目のみが表示されます。

16 ドリルスルー・レポート

この章の内容

ドリルスルー・レポートの概要 317
ドリルスルー・レポートの操作 318
ドリルスルー・レポートの作成 319
レポートの交差レベルの指定 319
レポート・タイプの定義とレポートのカスタマイズ 321
ドリルスルー・レポートと Essbase モデルの関連付け 340

ドリルスルー・レポートの概要

ドリルスルー・レポートを使用して、外部リレーショナル・データ・ソースから直接取得したデータを表示するスプレッドシート・レポートを作成します。

Essbase データベースを作成する場合、関連付けられている外部データ・ソースのデータをすべて使用するわけではありません。かわりに、スプレッドシート・ユーザーに最も必要だと思われるデータを選択し、要約します。たとえば、州ごとのルート・ビールの売上総数を要約できます。スプレッドシート・ユーザーは、この要約を使用して、州および地域の売上を比較できます。1 月のルート・ビールの売上が最も高かった州や 7 月のルート・ビールの売上が最も高かった州などがわかります。

ルート・ビールの売上を向上させる戦略を立てるために、要約よりも詳細な情報が必要なスプレッドシート・ユーザーもいます。次のような情報を取得する必要があります:

- カリフォルニア州で売上が最も低い 10 店舗は?
- これらの店舗の住所は?
- 店舗マネージャの名前は?

通常このような詳細情報は、Essbase データベースではなく外部データ・ソースに保管されています。

詳細な関連データを表示するには、スプレッドシート・ユーザーは外部データ・ソースにドリルスルーし、Essbase データベースのデータのコンテキストにあるデータを含むレポートを表示します。このレポートがドリルスルー・レポートです。

リリース 11.1.1 までは、Integration Services を使用してドリルスルー・レポートを作成していました。リリース 11.1.1 からは、Essbase Studio でもドリルスルー・レ

ポートを作成できるようになりました。Essbase Studio では、ドリルスルー・レポートのスコープが拡張され、機能が向上しています。

ドリルスルー・レポートは、選択したデータ・ソースで定義されている任意の交差のコンテキストまたはレベル(メンバーの組合せ)に基づくことができます。Smart View では、フォントの色やサイズなどの視覚的な手掛りによって、スプレッドシートの 1 つ以上のセルでドリルスルー・レポートが使用可能であることをユーザーに喚起します。

スプレッドシート・ユーザーには、データ・ソースの関連する詳細情報を取得する定義済ドリルスルー・ターゲットのレポートが表示されます。Essbase Studio で作成されるドリルスルー・レポート・ターゲットは、次のとおりです:

- リレーショナル・データベース
- URL (Oracle Hyperion Financial Data Quality Management および Oracle BI EE の URL を含む)
- カスタマイズされた Java メソッド

Integration Services で作成されたドリルスルー・レポート・ターゲットには、リレーショナル・データベースおよび URL が含まれます。

この章には、Essbase Studio でのドリルスルー・レポートの作成および維持に関する情報が含まれています。詳細は次を参照してください:

- Integration Services でのドリルスルー・レポートの作成および維持については、『Oracle Essbase Integration Services オンライン・ヘルプ』を参照してください。
- Smart View でのドリルスルー・レポートの使用については、Oracle Enterprise Performance Management System インストールおよび構成ガイドを参照してください。

ドリルスルー・レポートの操作

ドリルスルー・レポートでは、関連する物理データ・ソースに保管されているデータへの直接アクセスがスプレッドシート・クライアント・ユーザーに提供されます。ドリルスルー・レポートは交差レベル(メンバー・セット)に基づいています。スプレッドシートでは、ユーザーはこれらの交差レベルを表すセルをダブルクリックして、Essbase キューブ以外のターゲット・データ・ソースに保管されている詳細な情報を表示します。

ドリルスルー・レポートは、次の手順で定義します:

- 319 ページの「ドリルスルー・レポートの作成」
- 319 ページの「レポートの交差レベルの指定」
- 321 ページの「レポート・タイプの定義とレポートのカスタマイズ」
- 340 ページの「ドリルスルー・レポートと Essbase モデルの関連付け」

ドリルスルー・レポートの作成

➤ ドリルスルー・レポートを作成するには:

1 **「メタデータ・ナビゲータ」で、ドリルスルー・レポートを保管するフォルダを選択して右クリックし、「新規」>「ドリルスルー・レポート」を選択します。**

フォルダを選択し、右クリックして「ドリルスルー・レポート」オプションを選択すると、ドリルスルー・レポートはそのフォルダに保管されます。「ルート」を右クリックすると、ルート・レベルに保管されます。

2 **「名前」を入力します。**

ドリルスルー・レポート名は最大 50 文字までです。

375 ページの「ドリルスルー・レポートの制限とガイドライン」を参照してください。

3 **「オプション:」「説明」を入力します。**

4 **次のタスクを実行します:**

- レポートの交差レベルを指定します。319 ページの「レポートの交差レベルの指定」を参照してください
- レポート・タイプの定義およびレポートのカスタマイズを行います。321 ページの「レポート・タイプの定義とレポートのカスタマイズ」を参照してください
- エクスポートした 1 つ以上のモデルにレポートを関連付けます。340 ページの「ドリルスルー・レポートと Essbase モデルの関連付け」を参照してください

レポートの交差レベルの指定

ドリルスルー・レポート・エディタの「コンテキスト」タブで、ドリルスルー・レポートの交差レベルを定義する場合、このドリルスルー・レポートをその他のドリルスルー・ターゲットに結び付ける共通スレッドを設定します。

320 ページの「交差レベルの設定場所の指定」を参照してください。

➤ レポートの交差レベルを指定するには:

1 **ドリルスルー・レポート・エディタの「コンテキスト」タブで、「追加」をクリックします。**

「階層の選択」ダイアログ・ボックスが表示されます。

2 **「階層の選択」で、ドリルスルー・レポートに含める階層、メジャー階層または再帰階層に移動してその階層を選択し、「OK」をクリックします。**

ドリルスルー・レポート・エディタの「交差」に、階層が表示されます。

また、「メタデータ・ナビゲータ」から、ドリルスルー・レポート・エディタの「コンテキスト」タブにある「交差」の下の領域に階層を直接ドラッグして追加することもできます。この方法を使用する場合、「追加」ボタンをクリックする必要はありません。

注： すべての階層は、同じリレーショナル・ソースから選択する必要があります。このレポートに定義するドリルスルー交差の階層を 1 つ以上選択できます。

注： Performance Management Architect ソースから作成された階層は、ドリルスルー交差としてサポートされていません。

3 **「交差」の下に展開されている階層で、ドリルスルー交差として指定するメンバー名の横にあるチェック・ボックスを選択します。**

注： 最上位レベルなどの階層のレベルをドリルスルー交差として指定できます。

4 **手順 2 から手順 3 を繰り返して、定義するすべてのドリルスルー交差の階層を追加します。**

注： Smart View のみ: 1 つの階層のみを選択する場合、対応するメンバー・セルおよびデータ・セルからドリルスルー・レポートを使用できます。これは、メンバー・ドリルスルーとして知られています。

5 **このドリルスルー・レポートに再帰階層を選択し、レベルが 0 のデフォルトの交差を変更する場合は、「拡張設定」をクリックして「再帰階層設定」ダイアログ・ボックスを開きます。**

注： 再帰階層がメジャー階層でもある場合は、ドリルスルー・レポート・エディタで通常の階層として扱われ、「拡張設定」ボタンは無効になります。

詳細は、321 ページの「再帰階層の交差レベルの指定」を参照してください。

6 **「レポート・コンテンツ」タブを選択し、321 ページの「レポート・タイプの定義とレポートのカスタマイズ」の説明に従ってレポート・タイプの定義および付随するパラメータの指定を行います。**

交差レベルの設定場所の指定

ドリルスルー・レポートに交差またはコンテキストを指定する場合、Essbase セルを決定して、そこからスプレッドシートのユーザーがドリルスルーできるようにします。

階層の最上位レベルを含むようにドリルスルー・コンテキストを指定する場合、スプレッドシートのユーザーは、関連する Essbase 次元が含まれるどの交差からもドリルスルーできます。たとえば、製品階層全体に対して 1 つの交差レベルを設定すると、スプレッドシートのユーザーは、その製品階層から構築された製品次元のどのレベルからでもドリルスルーできます。スプレッドシートでは、ユーザーが Colas や 100-10 などの製品次元のメンバーを含むデータ値であれば、どれでもダブルクリックできます。

レベルをドリルスルー・コンテキストとして指定すると、階層のそのレベルに基づいたフィルタ(または SQL SELECT 文の WHERE 句)がドリルスルー要求に確実に適用されます。

再帰階層の交差レベルの指定

再帰階層にドリルスルー交差を作成して、レポートに含まれる世代またはレベルを指定できます。ドリルスルー・レポートは、Smart View で表示できます。

再帰階層では、デフォルトのドリルスルー交差はレベル 0 です。

再帰階層の詳細は、187 ページの「再帰階層」を参照してください。

➤ 再帰階層のドリルスルー交差の世代またはレベルを指定するには:

1 **319 ページの「レポートの交差レベルの指定」の手順 1 から手順 5 までを完了します。**

2 **「再帰階層設定」の「世代/レベル」で、適切な再帰階層行のセルをクリックし、このドリルスルー・レポートの交差が世代とレベルのどちらに基づくかを指定します。**

   デフォルトの設定は「レベル」です。

3 **「番号」で、ドリルスルー交差として使用する世代またはレベル番号を指定します。**

   デフォルト設定は 0(ゼロ)です。

4 **ドリルスルー・レポートに含まれているすべての再帰階層について、手順 2 から手順 3 までを繰り返します。**

5 **「OK」をクリックして指定した再帰階層設定を保存し、ドリルスルー・レポート・エディタの「コンテキスト」タブに戻ります。**

6 **「レポート・コンテンツ」タブを選択し、321 ページの「レポート・タイプの定義とレポートのカスタマイズ」の説明に従ってレポート・タイプの定義および付随するパラメータの指定を行います。**

レポート・タイプの定義とレポートのカスタマイズ

「レポート・コンテンツ」タブで、作成するレポートのタイプを定義します。実行するタスクは、定義するレポートのタイプによって異なります。

選択できるレポート・タイプは、次のとおりです:

- 「リレーショナル」-レポートでリレーショナル・クエリーを使用する場合に選択します。リレーショナル・クエリーは、Essbase Studio によって生成される標準 SQL か、ユーザー定義テンプレート SQL のいずれかになります。

  322 ページの「リレーショナル・ソースのレポートの定義およびカスタマイズ」を参照してください。

- 「URL」-レポートで、ブラウザによって起動される URL を指定する場合に選択します。

  332 ページの「URL へのレポートの定義およびカスタマイズ」を参照してください。
- 「Java メソッド」-レポートでユーザー定義の Java メソッドを指定する場合に選択します。

  337 ページの「Java メソッドへのレポートの定義およびカスタマイズ」を参照してください。

リレーショナル・ソースのレポートの定義およびカスタマイズ

次のタスクを実行するには、「レポート・コンテンツ」タブを使用します:

- レポート・タイプの定義とドリルスルー・レポートの列の指定

  322 ページの「リレーショナル・レポート・タイプの定義とドリルスルー・レポートの列の指定」を参照してください。
- ドリルスルー・レポート列のソート順の定義

  324 ページの「ドリルスルー・レポート列のソート順の定義」を参照してください
- テンプレート SQL の定義

  325 ページの「テンプレート SQL の定義」を参照してください。
- 行ガバナーの指定

  326 ページの「リレーショナル・ソースの行ガバナーの指定」を参照してください。
- フィルタの指定

  327 ページの「リレーショナル・ソースのドリルスルー・レポート・フィルタの指定」を参照してください。
- レポートのテスト

  328 ページの「リレーショナル・ソースのレポートのテスト」と329 ページの「テスト・シナリオの例–キャプションおよびキー・バインディングが異なる場合」を参照してください。

リレーショナル・レポート・タイプの定義とドリルスルー・レポートの列の指定

➤ リレーショナル・ドリルスルー・レポートのタイプを定義し、レポートの列とその順序を指定するには:

1 **ドリルスルー・レポート・エディタの「レポート・コンテンツ」タブを選択します。**

2 **「ドリルスルー・レポートのタイプ」で、「リレーショナル」を選択します。**

3 **ドリルスルー・レポートに表示される列を追加します。**

注： レポートの列は、すべて同じリレーショナル・ソースのものにする必要があります。

1. 「レポート・コンテンツ」タブのレポート・グリッドの横にある「追加」をクリックし、ドリルスルーの列をレポートに追加します。

   ドリルスルー列は、スプレッドシート・ユーザーが交差レベルをダブルクリックすると Essbase Studio が取得する外部データベースの列です。列には Essbase データベースで使用できない詳細情報(店舗マネージャのリストなど)が含まれます。

2. 「列の選択」でドリルスルー・レポートに含める列に移動してその列を選択し、「OK」をクリックします。

   選択した列はレポート・グリッドに表示されます。

3. ドリルスルー・レポートに追加する列ごとに、手順 3.1 と手順 3.2 を繰り返します。

4 **オプション:手順 3 で選択した列の名前とは異なる列名を表示するには:**

1. レポート・グリッドで操作する列を選択し、見出しが「表示名」の列の下にあるセルをクリックし、セルをアクティブにします。
2. ドリルスルー・レポートで表示する列の名前を入力します。
3. ドリルスルー・レポートの表示名を変更する列ごとに、手順 4.1 と手順 4.2 を繰り返します。

5 **オプション: 集約関数をドリルスルー列に適用します。**

集約関数を使用して、ドリルスルー・レポートで戻される列値に対して集計を実行します。

集約関数をドリルスルー列に適用するには:

1. レポート・グリッドで操作する列を選択し、見出しが「集約」の列の下のセルをクリックし、ドロップダウン・リストのコントロールをアクティブにします。
2. ドロップダウン・リストから、次の集約関数のいずれかを選択します:
   - 「平均」-列の平均値を戻します。列タイプが数値の場合のみに適用されます。
   - 「カウント」-選択した行の数値を戻します。すべての列タイプに適用されます。
   - 「最小」-列の最小値を戻します。列タイプが数値の場合のみに適用されます。
   - 「最大」-列の最大値を戻します。列タイプが数値の場合のみに適用されます。
   - 「合計」-列の合計値を戻します。列タイプが数値の場合のみに適用されます。
3. 集約関数を適用する列ごとに、手順 5.1 と手順 5.2 を繰り返します。

6 **オプション:** ドリルスルー列のソート順を定義します。

詳細は、324 ページの「ドリルスルー・レポート列のソート順の定義」を参照してください。

7 **オプション:** ドリルスルー・レポートから列を削除するには、レポート・グリッドで列を選択して「削除」をクリックします。

8 **オプション:** ドリルスルー・レポートで重複するレコードを表示するには、「重複するレコードを表示」チェック・ボックスを選択します。

9 **オプション:** 記述した SQL をこのレポートのベースにするには、「ユーザー定義 SQL を使用」チェック・ボックスを選択して「テンプレート SQL」ボタンをクリックします。

詳細は、325 ページの「テンプレート SQL の定義」を参照してください。

または、「テンプレート SQL」ダイアログ・ボックスでユーザー定義の SQL を保存したが使用しない場合、「ユーザー定義 SQL を使用」チェック・ボックスの選択を解除します。Essbase Studio により、ドリルスルー・レポートの作成に使用される SQL が自動的に作成され、ドリルスルー SQL は後で使用できるように保存されます。

10 ドリルスルー・レポート・エディタの「レポート・コンテンツ」タブで残りのタスクを完了するには、次のトピックを参照してください:

- 324 ページの「ドリルスルー・レポート列のソート順の定義」
- 325 ページの「テンプレート SQL の定義」
- 326 ページの「リレーショナル・ソースの行ガバナーの指定」
- 327 ページの「リレーショナル・ソースのドリルスルー・レポート・フィルタの指定」
- 328 ページの「リレーショナル・ソースのレポートのテスト」と329 ページの「テスト・シナリオの例–キャプションおよびキー・バインディングが異なる場合」

ドリルスルー・レポート列のソート順の定義

スプレッドシート・クライアントで表示される行の順序と、取得される行のコンテンツを決定できます。たとえば、PRODUCTDIM.SKUNAME 列のコンテンツを降順に(高い値から低い値へ)ソートできます。このようにソートすると、Vanilla Cream から Old Fashioned、Caffeine Free Cola のように、製品がアルファベットの逆順に表示されます。

ソート順を指定しない場合、スプレッドシート・ユーザーには、外部データ・ソースによって決められた順序でデータが表示されます。たとえば Microsoft SQL サーバーの場合、デフォルトではメンバーは昇順にソートされます。

注: この手順はオプションです。

➤ ドリルスルー列のソート順を定義するには:

1 **ドリルスルー・レポート・エディタの「レポート・コンテンツ」タブのレポート・グリッドで、操作する列を選択し、見出しが「ソート順」である列の下のセルをクリックし、ドロップダウン・リストのコントロールをアクティブにします。**

2 **「昇順」(昇順でのソート)または「降順」(降順でのソート)を選択します。**

3 **ソート順を適用する列ごとに、手順 1 と手順 2 を繰り返します。**

4 **「上に移動」と「下に移動」ボタンを使用して、ドリルスルー・レポートの列の順序を変更します。**

   列は、レポート・グリッドで上から下にリスト表示されていた順序で左から右に表示されます。レポート・グリッドの列を、ドリルスルー・レポートに表示したい順序に変更します。

テンプレート SQL の定義

Essbase Studio では、ドリルスルー・レポートを作成するために使用する SQL が自動的に生成されます。SQL 文は、ドリルスルー・レポートおよび戻される列の OLAP 交差レベルを識別します。Essbase Studio によって生成された SQL は、独自の SQL で上書きできます。

「テンプレート SQL」ダイアログ・ボックスを使用して、テンプレート SQL と呼ばれるドリルスルー SQL のテンプレートを定義します。テンプレート SQL は、パラメータとしてドリルスルー・レポートに渡す必要があるパラメータを現在のレポート・コンテキストから指定します。テンプレート SQL では、任意の定義済のデータ・ソース接続からのテーブルおよび列を組み入れることができます。その際、テーブルや列はミニスキーマ内で使用または結合されていても構いません。

Essbase Studio テンプレート SQL は、変数の区切り記号として$$構文を使用します。$$構文内に含まれるテキストは、ドリルスルーの実行中に実際の列またはデータ値に置き換えられます。ドリルスルー・テンプレート SQL を使用する場合、次のルールに注意してください:

- 次元の交差レベルが世代 1 で定義され、次元が親または子テーブルから構築されている場合、テンプレート SQL 内で$$代替変数は使用できません。
- 次元交差値に関連付けられている事前定義済の変数と次元テーブルの列のすべてのペアが、ユーザー定義の SQL に含まれている必要があります。たとえば、次元 Product の交差は、標準 SQL テンプレートの次の式で指定されます:

  ```
  $$Product-COLUMN$$ IN ($$Product-VALUE$$)
  ```

  この式は、ユーザー定義の SQL テンプレートにも含める必要があります。
- ユーザー定義の SQL テンプレートでは、Essbase Studio サーバーによって生成された事前定義済の SQL テンプレートの別名と同じ名前を再利用する必要があります。

注: この手順はオプションです。

➤ テンプレート SQL を定義するには:

1 ドリルスルー・レポート・エディタの「レポート・コンテンツ」タブで、「テンプレート SQL」ボタンをクリックします。

2 **オプション:** 標準の SQL を参考として使用するには、「キューブ・スキーマ」ドロップダウン・リストから独自の SQL の基本とするキューブ・スキーマを選択して、「標準 SQL の取得」をクリックします。

このキューブ・スキーマの SQL が、読取り専用の「標準 SQL」テキスト・ボックスと編集可能な「ユーザー定義 SQL」テキスト・ボックスに表示されます。

3 このドリルスルー・レポートで必要な SQL を編集または記述します。

4 「ユーザー定義 SQL の更新」をクリックして、ダイアログ・ボックスを終了せずにユーザー定義の SQL を保存します。

5 「検証」をクリックして、「ユーザー定義 SQL」内の構文を検証します。

6 エラーに留意し、ユーザー定義 SQL を適切に修正します。

注: SQL が正しい場合、SQL が有効であることを知らせるメッセージが表示されます。

7 エラーを修正したら、「OK」をクリックしてドリルスルー・レポート・エディタに戻ります。

8 ドリルスルー・レポート・エディタの「レポート・コンテンツ」タブで残りのタスクを完了するには、次のトピックを参照してください:

- 324 ページの「ドリルスルー・レポート列のソート順の定義」
- 326 ページの「リレーショナル・ソースの行ガバナーの指定」
- 327 ページの「リレーショナル・ソースのドリルスルー・レポート・フィルタの指定」
- 328 ページの「リレーショナル・ソースのレポートのテスト」と329 ページの「テスト・シナリオの例–キャプションおよびキー・バインディングが異なる場合」

注意:

- ユーザー定義 SQL は実行時まで検証されません。詳細は、364 ページの「ユーザー定義 SQL の検証」を参照してください。
- Essbase Studio におけるユーザー定義 SQL に対する権限は、データ・ソース・ユーザーに割り当てられている権限から継承されます。詳細は、364 ページの「ユーザー定義 SQL のデータ・ソース権限」を参照してください。

リレーショナル・ソースの行ガバナーの指定

「ドリルスルー・レポート・プロパティ」ダイアログ・ボックスの「レポート・コンテンツ」タブで行ガバナーを指定します。

注: この手順はオプションです。

➤ 行ガバナーを指定するには、「行ガバナー」テキスト・ボックスに、取得する最大行数を入力します。

たとえば、200 行を超える行数を取得するクエリーを停止するには、200 と入力します。

ドリルスルー・レポート・エディタの「レポート・コンテンツ」タブで残りのタスクを完了するには、次のトピックを参照してください:

- 324 ページの「ドリルスルー・レポート列のソート順の定義」
- 325 ページの「テンプレート SQL の定義」
- 327 ページの「リレーショナル・ソースのドリルスルー・レポート・フィルタの指定」
- 328 ページの「リレーショナル・ソースのレポートのテスト」と329 ページの「テスト・シナリオの例–キャプションおよびキー・バインディングが異なる場合」

リレーショナル・ソースのドリルスルー・レポート・フィルタの指定

フィルタを指定して、戻される特定のターゲット・ドリルスルー列の結果を制限します。

➤ リレーショナル・ソースのドリルスルー・レポート・フィルタを指定するには:

1 **ドリルスルー・レポート・エディタの「レポート・コンテンツ」タブで、「フィルタ」をクリックします。**

2 **「ドリルスルー・フィルタ」ダイアログ・ボックスで、「ソース」、「関数」および「演算子」タブにある要素のリストから適切な要素をドラッグして、「フィルタ」ボックスにドロップします。**

たとえば、ドリルスルー・レポートをフィルタして製品ファミリ 400 のメンバーを得る場合、次のようなフィルタを入力できます:

```
'contains'( connection : \'tbcSource'::'tbc.family'.'FAMILY'=="400")
```

ドリルスルー・レポートをフィルタして、製品ファミリ 400 かつニューヨーク州のみのメンバーを得る場合、次のようなフィルタを入力できます;

```
contains(connection : \'tbcSource'::'tbc.market'.'STATE',"New York") and
contains(connection : \'tbcSource'::'tbc.family'.'FAMILY',"400")
```

3 **「OK」をクリックして「ドリルスルー・フィルタ」ダイアログ・ボックスを閉じ、ドリルスルー・レポート・エディタの「レポート・コンテンツ」タブに戻ります。**

4 **ドリルスルー・レポート・エディタの「レポート・コンテンツ」タブで残りのタスクを完了するには、次のトピックを参照してください:**

- 324 ページの「ドリルスルー・レポート列のソート順の定義」

- 325 ページの「テンプレート SQL の定義」
- 326 ページの「リレーショナル・ソースの行ガバナーの指定」
- 330 ページの「「遅延」オプションの例」と329 ページの「テスト・シナリオの例–キャプションおよびキー・バインディングが異なる場合」

リレーショナル・ソースのレポートのテスト

この手順は、リレーショナル・ソースに基づくドリルスルー・レポートのテストに使用します。この手順では、テストの一般的な方法を概説します。その他のテスト・シナリオについては、329 ページの「テスト・シナリオの例–キャプションおよびキー・バインディングが異なる場合」と331 ページの「テスト・シナリオの例–再帰階層」を参照してください。

➤ ドリルスルー・レポートをテストするには:

1 **「レポート・コンテンツ」タブで、「テスト」をクリックします。**

2 **テストに使用する「キューブ・スキーマ」を選択します。**

3 **各交差の「列値」を入力します。**

「コンテキスト」タブで交差として選択した列が「交差」の下に表示されます。また、交差セル内をクリックしてセルの右にある下矢印を選択し、同じ交差で使用可能なすべての列が表示されるドロップダウン・リストから列を選択することもできます。「列値」に、選択した各列に対して有効な値を入力します。

たとえば、TBC サンプル・データベースを使用していて、`Family` が交差として表示される場合、「列値」列に「300」と入力して、300 製品ファミリの結果を表示します。さらに、`Family` の交差セルのドロップダウン・リストを使用して、使用可能なドリルスルー列のリストから別の列を選択します。たとえば、`SKU` の場合、「列値」列に `300-30` と入力します。

再帰階層をテストしている場合、「ドリルスルー・レポートのテスト」ダイアログ・ボックスの「交差」フィールドと「列値」フィールドは、標準階層をテストしている場合とは異なる方法で使用されます。

- 「交差」フィールドに、再帰階層名が表示されます。たとえば、再帰階層名が recur_hier の場合は、`recur_hier` と表示されます。
- 「列値」フィールドに、データ値ではなく有効な親または子列の名前を入力します。

4 **オプション: 整数を入力して「表示最大行」を指定します。**

デフォルトの表示行数は 20 行です。表示行の最大数は 1024 です。

5 **「結果の表示」をクリックします。**

「生成されたレポート」セクションを表示して、ドリルスルー・レポートでの結果の表示方法を確認します。

6 **「閉じる」をクリックして、ドリルスルー・レポート・エディタの「レポート・コンテンツ」タブに戻ります。**

7 **「保存」をクリックしてから「閉じる」をクリックし、「ドリルスルー・レポートのテスト」ダイアログ・ボックスを閉じます。**

ドリルスルー・レポートの定義を続行するには、「関連付け」タブを選択し、340 ページの「ドリルスルー・レポートと Essbase モデルの関連付け」のタスクを実行します。

次の場所にテスト・シナリオの例があります:

- 329 ページの「テスト・シナリオの例–キャプションおよびキー・バインディングが異なる場合」–交差に「拡張」および「遅延」キー・バインディング・オプションを使用する要素が含まれる場合
- 331 ページの「テスト・シナリオの例–再帰階層」–ドリルスルー・レポートに再帰階層が含まれる場合。

テスト・シナリオの例–キャプションおよびキー・バインディングが異なる場合

ドリルスルー・レポートをテストする際、「ドリルスルー・レポートのテスト」ダイアログの「列値」フィールドに入力する値は、交差内の次元要素のキー・バインディング式を表す値にする必要があります。

一般に、次元要素のキー・バインディング式の値は、アウトライン・メンバー名を表すキャプション・バインディング式の値と同じです。ただし、ドリルスルー・レポートの交差の作成に使用される一部の階層は、キャプション・バインディングがキー・バインディングと異なる次元要素を使用して作成されている場合があります。このようなケースに該当するのは、次元要素のキー・バインディングにおいて「拡張」または「遅延」オプションが選択されている場合です。

次に示す手順の例では、交差に「拡張」および「遅延」オプションを使用する要素が含まれている場合のドリルスルー・レポートのテスト用のシナリオが含まれています。次の各トピックを参照してください。

- 329 ページの「「拡張」オプションの例」
- 330 ページの「「遅延」オプションの例」

注: この手順の例では、サンプル TBC データベースへのデータ・ソース接続が作成済であることを前提としています。

「拡張」オプションの例

➤ 次元要素で「拡張」オプションを使用する階層をテストするには:

1 **TBC データベースの Product テーブル(TBC.PRODUCT)からの SKU に基づくキャプション・バインディング式により、SKU_Advanced という次元要素を作成します。例:**

```
connection : \'tbcSource'::'TBC.PRODUCT'.'SKU'
```

2 **「キー・バインディング」のタイプを「拡張」に設定します。**

3 **「キー・バインディング」式を Product テーブル(TBC.PRODUCT)の PRODUCT 列に基づかせます。例:**

```
connection : \'tbcSource'::'tbc.product'.'PRODUCTID'
```

4 FAMILY テーブルの FAMILY 列を親レベルとして持ち、SKU_Advanced 次元要素を子として持つ、product_advanced 階層を作成します。

5 手順 4 で作成した product_advanced 階層を使用してキューブ・スキーマと Essbase モデルを作成します。TBC.SALES の AMOUNT をメジャーとして使用します。

6 次のように指定して、ドリルスルー・レポートを作成(「ファイル」>「新規」>「ドリルスルー・レポート」)します:

1. 「コンテキスト」タブで、product_advanced 階層を追加します。
2. 「レポート・コンテンツ」タブで、TBC.PRODUCT の任意のフィールドを配置します。例: CAFFEINATED または PKGTYPE。

7 「テスト」をクリックして「ドリルスルー・レポートのテスト」ダイアログ・ボックスを開きます。

8 「交差」の下にある最初の行のドロップダウン矢印をクリックして「SKU_Advanced」を選択します。

9 「列値」に、手順 3 のキー・バインディング式に表示された値を表す値を入力します。

このシナリオで作成した階層を使用して、PRODUCTID 列の値を入力します。例:

```
1
```

10 「結果の表示」をクリックします。

「遅延」オプションの例

➤ 次元要素で「遅延」オプションを使用する階層をテストするには:

1 TBC データベースの Product テーブル(TBC.PRODUCT)からの SKU に基づくキャプション・バインディング式により、SKU_Delayed という次元要素を作成します。例:

```
connection : \'tbcSource'::'TBC.PRODUCT'.'SKU'
```

2 「キー・バインディング」のタイプを「遅延」に設定します。

3 FAMILY テーブルの FAMILY 列を親レベルとして持ち、SKU_Delayed 次元要素を子として持つ、product_delayed 階層を作成します。

階層内で、SKU_Delayed 要素のキー・バインディング式は次のように表示される必要があります:

```
class : \'tbcSource'\'family'\'FAMILY'.'caption' || "_" || class :
\'DimElements'\'SKU_Delayed'.'caption'
```

4 手順 3 で作成した product_delayed 階層を使用してキューブ・スキーマと Essbase モデルを作成します。'TBC.SALES'の AMOUNT をメジャーとして使用します。

5 次のように指定して、ドリルスルー・レポートを作成します:

1. 「コンテキスト」タブで、手順 3 で作成した product_delayed 階層を追加します。
2. 「レポート・コンテンツ」タブで、TBC.PRODUCT の任意のフィールドを配置します。例: CAFFEINATED または PKGTYPE。

6 「テスト」をクリックして「ドリルスルー・レポートのテスト」ダイアログ・ボックスを開きます。

7 「交差」の下にある最初の行のドロップダウン矢印をクリックして「SKU_Delayed」を選択します。

8 「列値」に、手順 3 のキー・バインディング式に表示された値を表す値を入力します。

このシナリオで作成した階層を使用して、FAMILY 列の値と SKU 列の値を、アンダースコアで区切って入力します。例:

```
100_100-10
```

9 「結果の表示」をクリックします。

テスト・シナリオの例–再帰階層

再帰階層をテストしている場合は、標準階層のテスト時とは異なる方法で「ドリルスルー・レポートのテスト」ダイアログ・ボックスの「交差」および「列値」フィールドを使用します:

- 「交差」フィールドに、再帰階層名が表示されます。
- 「列値」フィールドに、データ値ではなく親または子列の名前を入力します。

これを次のシナリオ例に示します。

注 : 例の手順では、(1)サンプル TBC データベースにデータ・ソース接続を作成してあることと、(2)MEASURES テーブルの PARENT 列と CHILD 列の間に自己結合を含む TBC ソースのミニスキーマを作成してあることを前提とします。

➤ ドリルスルーに対して選択した再帰階層をテストするには:

1 MEASURES テーブルの PARENT 列を親レベルとして持ち、MEASURES テーブルの CHILD 列を子レベルとして持つ、recursive_hier という再帰階層を作成します。

注 : MEASURES テーブルの PARENT 列と CHILD 列の間の自己結合を含むミニスキーマを作成してあることを確認します。

2 手順 1 で作成した recursive_hier 階層を使用してキューブ・スキーマと Essbase モデルを作成します。'TBC.SALES'の AMOUNT をメジャーとして使用します。

3 次のように指定して、ドリルスルー・レポートを作成します:

1. 「コンテキスト」タブで、手順 1 で作成した recursive_hier 階層を追加します。
2. 「レポート・コンテンツ」タブで、TBC.PRODUCT の任意のフィールドを配置します。例: CAFFEINATED または PKGTYPE。

4 「テスト」をクリックして「ドリルスルー・レポートのテスト」ダイアログ・ボックスを開きます。

5 「交差」フィールドには、再帰階層の名前 recursive_hier が含まれます。

6 「列値」フィールドで、列名を選択します。たとえば COGS を選択します。

注： ここでは、列データではなく列名が使用されます。

7 「結果の表示」をクリックします。

URL へのレポートの定義およびカスタマイズ

Essbase Studio では、ドリルスルー・レポートのオプションが拡張されており、ドリルスルーのターゲットとして URL を指定できます。ドリルスルー・レポート・エディタの「レポート・コンテンツ」タブで URL を指定すると、ユーザーは、スプレッドシートのドリルスルー交差から直接 URL にドリルスルーできます。

URL レポート・タイプを使用してターゲット URL を定義します。ドリルスルーのターゲットとして静的 URL を指定できます。同様に、FDM や Oracle BI EE Web ページなどをターゲットにするように動的 URL を指定できます。

URL の構文は、ターゲット URL の要件に一致している必要があります。ただし、次元、列および値を表す場合、通常すべての URL は、次のような変数構造になります:

```
$$<dimension-name>-VALUE$$
```

ユーザーがドリルスルー・セルをクリックすると、Essbase Studio によって、選択したドリルスルー交差のコンテキストにあるターゲット URL を生成するために必要な置換が行われます。

たとえば、市場次元がドリルスルーのターゲット URL の視点(POV)内に存在する場合、市場の値の変数は次のようになります:

```
$$Market-VALUE$$
```

スプレッドシート・ユーザーが市場次元のメンバーを含む交差からドリルスルー・セルを選択する場合、Essbase Studio によって、適切な市場の値が URL 構文に置換されます。例:

```
East
```

次の例では、製品次元の値の変数は、次のようになります:

```
$$Product-VALUE$$
```

スプレッドシート・ユーザーが製品次元のメンバーを含む交差からドリルスルー・セルを選択する場合、Essbase Studio によって、適切な製品の値が URL 構文に置換されます。例:

```
Cola
```

注： Smart View では、URL の長さに関する制限は使用中のブラウザによって異なります。

➤ URL レポート・タイプの定義およびドリルスルーする URL の指定を行うには:

1 **ドリルスルー・レポート・エディタの「コンテキスト」タブで、「ドリルスルー・レポートのタイプ」から「URL」を選択します。**

2 **「URL」テキスト・ボックスに、ドリルスルー・レポートのターゲット URL を入力します。**

また、次のいずれかのボタンをクリックしたときに Essbase Studio によって生成される構文を編集して URL を作成することもできます:

- 「サンプル URL テンプレート」-「コンテキスト」タブで指定した交差に基づいてターゲット URL の構文が提供されます。

  詳細は、333 ページの「サンプル URL テンプレート」を参照してください。

- 「FDM URL テンプレート」-「コンテキスト」タブで指定した交差に基づいてターゲット FDM URL の構文が提供されます。

  詳細は、334 ページの「サンプル FDM URL テンプレート」を参照してください。

- 「OBI URL テンプレート」-「コンテキスト」タブで指定した交差に基づいてターゲット Oracle BI EE URL の構文が提供されます。

  詳細は、336 ページの「サンプル Oracle BI EE URL テンプレート」を参照してください。

3 **「保存」>「閉じる」をクリックして、ドリルスルー・レポート・エディタを閉じるか、「関連付け」タブを選択して、340 ページの「ドリルスルー・レポートと Essbase モデルの関連付け」のタスクを実行します。**

サンプル URL テンプレート

ドリルスルー・レポート・ターゲットとして URL を定義するには、サンプル URL テンプレートを使用します。静的 HTTP URL や、ドリルスルー・レポート・エ

ディタの「コンテキスト」タブで指定したドリルスルー交差を考慮する動的 URL をドリルスルー・ターゲットとして指定できます。

注： Smart View では、URL の長さに関する制限は使用中のブラウザによって異なります。

構文

静的 HTTP URL を入力できます。例:

```
http://www.oracle.com
```

動的ターゲットの URL のために、Essbase Studio コンソールでは、Google を使用するサンプルが提供されています。検索する値の数は、ドリルスルー・レポート・エディタの「コンテキスト」タブで指定したドリルスルー交差の数によって決まります。ドリルスルー交差が製品および市場という 2 つの階層で指定された場合の構文を次に示します:

```
http://www.google.com/search?hl=en&q=$$Product-VALUE$$+$$Market-VALUE$$
```

スプレッドシート・ユーザーがドリルスルー・レポートを実行すると、Essbase Studio により、ドリルスルー交差のコンテキストに応じて、適切な URL の置換が行われます。

例

この例では、Essbase Studio のドリルスルー・レポート・エディタのサンプル URL テンプレートを使用します。また、ドリルスルー・レポート・エディタの「コンテキスト」タブで製品および市場の階層が選択されたことを前提としています。

製品と市場の交差の「ドリルスルー」セルをユーザーがクリックすると、Essbase Studio によって、変数`$$Product-VALUE$$`および`$$Market-VALUE$$`が置換され、次の URL 文字列が生成されます:

```
http://www.google.com/search?hl=en&q=Cola+East
```

サンプル FDM URL テンプレート

サンプル FDM URL テンプレートにより、ドリルスルー・レポート・エディタの「コンテキスト」タブで指定する交差に基づいて、FDM のターゲット URL の構文を指定します。

構文

この構文は、製品および市場という 2 つの次元のみがキューブ内にあり、製品および市場階層がドリルスルー・レポート・エディタの「コンテキスト」タブで選択されていることを前提としています。

```
http://<server-name>/HyperionFDM/AuthorizedPages/
IntersectionSummaryByLocation.aspx?fdmAppName=<app-name>&fdmTargetAppName=<target-
app-name>&attribute=system.ds.essbase&attribute=server.ds.<server-
ds>&attribute=app.ds.<app-
ds>&attribute=database.ds.<database.ds>&attribute=alias.ds.<alias-ds>&sso_token=$
$CSS-TOKEN$$&attribute=Product.id.$$Product-VALUE$$&attribute=Market.id.$$Market-
VALUE&rcp_version=1.5.0
```

注： Smart View では、URL の長さに関する制限は使用中のブラウザによって異なります。

パラメータ

- <server-name>-接続するサーバーおよびポートの名前または IP アドレスです。
- <app-name>-FDM ターゲット・アプリケーション名です。この値は、FDM 管理者が製品内で特定のデータ・セットまたはリポジトリに割り当てる識別子と一致する必要があります。たとえば、Oracle Hyperion Financial Management に同梱されている Comma サンプル・アプリケーションの場合、この値は comma になります。
- <target-app-name>-ドリルスルー・リクエストが生成される元の Essbase アプリケーションの名前。
- <server-ds>-FDM に対するドリルスルー・リクエストが生成される元の、Essbase アプリケーションおよびデータベースをホスティングしている Essbase Server の名前または IP アドレス。
- <app-ds>-ドリルスルー・リクエストが生成される元の Essbase アプリケーションの名前。
- <database.ds>-ドリルスルー・リクエストが生成される元の Essbase データベースの名前。
- <alias-ds>-ドリルスルー・リクエストで使用する別名テーブルの名前。別名テーブルが不要な場合は Default を使用します。
- Product.id.$$Product-VALUE$$-視点(POV)次元および次元メンバーの値で、この場合、製品次元の値です。Essbase Studio では、指定された次元に関連付けられた変数の要素として、自動的に次元名が生成されます。ドリルスルー・レポートを実行すると、変数の名前の$$Product-VALUE$$は、POV の実際の値(たとえば、Cola)に置き換えられます。

  FDM ソースで次元が別の名前で呼び出される場合、ドリルスルー構文を使用して手動で名前を置き換える必要があります。たとえば、FDM で「製品」次元が「項目」次元と呼ばれる場合、次のように、Essbase Studio で生成される構文の次元名「製品」を「項目」に置き換える必要があります:

```
http://<server-name>/HyperionFDM/AuthorizedPages/
IntersectionSummaryByLocation.aspx?fdmAppName=<app-
name>&fdmTargetAppName=<target-app-
name>&attribute=system.ds.essbase&attribute=server.ds.<server-
```

```
ds>&attribute=app.ds.<app-
ds>&attribute=database.ds.<database.ds>&attribute=alias.ds.<alias-ds>&sso_token=$
$CSS-TOKEN$$&attribute=Items.id.$$Product-VALUE$$&attribute=Market.id.$$Market-
VALUE&rcp_version=1.5.0
```

たとえば、レポートの一部の次元の情報のみが対象の場合でも、FDM のターゲット・レポートで表示される次元をすべて FDM の URL 構文で指定する必要があります。

たとえば、レポートの一部の次元の情報のみが対象の場合でも、FDM のターゲット・レポートで表示される次元をすべて FDM の URL 構文で指定する必要があります。デフォルトでは、サンプル FDM テンプレートの構文により、ドリルスルー・レポート・エディタの「コンテキスト」タブで指定した交差をすべて取得し、それらをテンプレート構文ステートメントに追加します。FDM のターゲット・レポートに次元が残っている場合、それらを構文ステートメントにも含める必要があります。

例

次の URL の例、FDM URL は、シナリオ、年、勘定科目、市場、製品次元を含む FDM のターゲット・レポートにドリルスルーします。

製品次元と市場次元は、ドリルスルー・レポート・エディタの「コンテキスト」タブで交差として指定されています。Essbase Studio では、これらは自動的に FDM テンプレート構文に追加されます。

会計次元と市場次元は、Oracle Hyperion Financial Data Quality Management のターゲット・レポートの一部ですが、ドリルスルー・レポートで交差として指定されていません。そのため、次に示すように、URL 構文で明示的に次元とメンバー名を指定する必要があります:

```
   http://myfdmsvr:19000/HyperionFDM/AuthorizedPages/
IntersectionSummaryByLocation.aspx?
fdmAppName=FDMAPP&fdmTargetAppName=Sample&attribute=system.ds.essbase&attribute=serve
r.ds.myfdmsvr11&attribute=app.ds.Sample&attribute=database.ds.Basic&attribute=alias.d
s.Default&sso_token=$$CSS-TOKEN$$&attribute=Product.id.$$Product-VALUE$
$&attribute=Market.id.$$Market-VALUE$$&attribute=Year.id.$$Year-VALUE$
$&attribute=Scenario.id.$$Scenario-VALUE$$&attribute=Accounts.id.$$Accounts-VALUE$
$&rcp_version=1.5.0
```

サンプル Oracle BI EE URL テンプレート

サンプル Oracle BI EE URL テンプレートは、ドリルスルー・レポート・エディタの「コンテキスト」タブで指定した交差に基づいて、ターゲット Oracle BI EE URL の構文を提供します。

この構文および例は、製品および市場階層がドリルスルー・レポート・エディタの「コンテキスト」タブで選択されていることを前提としています。

構文

ドリルスルー・レポート・エディタの「コンテキスト」タブで製品階層および市場階層を選択した後の構文は、次のとおりです:

```
obi:http://<server-name>:9704/analytics/saw.dll?PortalPages&PortalPath=<portal-path>&Action=Navigate&P0=2&P1=eq&P2=<Product-column>&P3=$$Product-VALUE$$&P4=eq&P5=<Market-column>&P6=$$Market-VALUE$$
```

注： Smart View では、URL の長さに関する制限は使用中のブラウザによって異なります。

パラメータ

- `<server-name>`-接続するサーバーの名前または IP アドレスです。
- `<portal-path>`-Oracle BI EE Presentation Service Report/Dashboard へのパスです。
- `<number-of-parameters>`-レポート・パラメータの数です。
- `<Market-column>`および`<Product-column>`-ターゲット・ドリルスルー・レポートで取得する Oracle BI EE データ・ソースの列です。たとえば、製品次元では SKU 列を、市場次元では地域列を選択できます。

  ```
  obi:http://obiserver:9704/analytics/saw.dll?PortalPages&PortalPath=paint&Action=Navigate&P0=2&P1=eq&P2=SKU&P3=$$Product-VALUE$$&P4=eq&P5=Region&P6=$$Market-VALUE$$
  ```

  複数の列を指定できます。

例

ドリルスルー・レポート・エディタの「コンテキスト」タブで製品階層および市場階層を選択した後に、ドリルスルー要求を実行すると、接頭辞「obi」がターゲット URL 文字列から削除され、かわりに次の URL 文字列が作成されます:

```
http://obiserver:9704/analytics/saw.dll?PortalPages&PortalPath=paint&Action=Navigate&P0=2&P1=eq&P2=SKU&P3=100-10&P4=eq&P5=Region&P6=East
```

Java メソッドへのレポートの定義およびカスタマイズ

作成するカスタム Java メソッドにドリルスルーするためのドリルスルー・レポートを定義するには、Java メソッドのレポート・タイプを使用します。

特定の要求を満たすレポートを生成する Java メソッドを作成できます。たとえば、特定のアプリケーションからデータを抽出する Java メソッドを作成できます。

ドリルスルー・レポートを実行するクライアント・アプリケーションの Smart View ユーザーの場合、リレーショナル・ソースへのドリルスルー・レポートの実行と、ユーザー定義のJava メソッドを使用したドリルスルー・レポートの実行に違いはありません。

➤ Java メソッド・レポート・タイプの定義やドリルスルー時に実行する Java メソッドの指定を行うには:

1 **「ドリルスルー・レポートのタイプ」で、「Java メソッド」を選択します。**

2 **「Java クラス名」で、Java クラスの完全なパッケージ名を入力します。例:**

```
test.com.hyperion.cp.scripts.acceptance.test_cases.TestDTRJavaMethod
```

3 **「Java メソッド名」で、Java メソッド名を入力します。例:**

```
runTest
```

Essbase Studio サーバーによって呼び出されるユーザー定義の Java メソッドには、次のシグネチャを含める必要があります:

```
public static void <method-name>(ArrayList<String>[]> args, ResultSet
result)
```

ユーザー定義の Java メソッドの `ArrayList<String>[]> args` パラメータは、ドリルスルー・レポート引数の配列リストです。配列リストの各項目は、交差のメンバーを表しており、次元名、クラス名およびメンバー名が含まれています。例

```
“ProductH”  “SKU”  “100-10”
“TimeH”  “Time Year”  “2006”
```

メンバーが Essbase の次元の最上位のメンバーである場合、クラス名の値は NULL になります。例:

```
“ProductH”  null  “ProductH”
```

`ResultSet` パラメータは、Java パッケージ com.hyperion.cp.interfaces のインタフェースです。

注 : 結果セットのレコードに値を設定する前に、結果セットのすべてのタグを定義する必要があります。

ユーザー定義の Java メソッドに次のインタフェース・メソッドを使用できます:

```
     /**
 * Adds a new tag to result set signature
 * @param tag Tag name
 * @param clazz Java class that corresponds to the tag
 *       (Boolean, Integer, Long, Double, String)
 */
public void addTag(String tag, Class clazz);

/**
 * Sets value in result set record
 * @param tag Tag name
 * @param value Value
 */
public void setValue(String tag, boolean value);
public void setValue(String tag, int value);
public void setValue(String tag, long value);
public void setValue(String tag, double value);
public void setValue(String tag, String value);

/**
 * Adds record to the result set
 *
 */
public void addRecord();
```

ユーザー定義の Java メソッドである runTest の例を次に示します。このメソッドは、ドリルスルー・レポートのパラメータを階層、クラスおよび値のタグを持つ結果セットに変換します。すべてのタグの値は文字列です。

```
     package test.com.hyperion.cp.scripts.acceptance.test_cases;

import com.hyperion.cp.interfaces.ResultSet;

import java.util.ArrayList;
import java.util.Iterator;

public class TestDTRJavaMethod
{
 /**
  * User defined java method for DTR
  * @param result Result of DTR
  * @param args Actual arguments of the DTR
  */
 public static void runTest(ArrayList<String[]> args, ResultSet result)
 {
  // make signature of the result

  result.addTag("Hierarchy", String.class);
  result.addTag("Class", String.class);
  result.addTag("Value", String.class);

  Iterator<String[]> iterator = args.iterator();

  while (iterator.hasNext())
```

```
    {
     // add records to the result

     String[] tuple = iterator.next();

     result.setValue("Hierarchy", tuple[0]);
     result.setValue("Class", tuple[1]);
     result.setValue("Value", tuple[2]);

     result.addRecord();
    }
   }
}
```

スプレッドシート・ユーザーがドリルスルー・レポートを実行すると、Java メソッドが実行されて、次のようなドリルスルー・レポートが作成されます:

```
     ====================================
Hierarchy   Class     Value
====================================
ProductH    SKU       100-10
TimeH     Time Year   2006
MarketH     REGION    East
MyMeasuresH  Msr_Gen4    Sales
====================================
```

4 「保存」>「閉じる」をクリックして、ドリルスルー・レポート・エディタを閉じるか、「関連付け」タブを選択して、340 ページの「ドリルスルー・レポートと Essbase モデルの関連付け」のタスクを実行します。

ドリルスルー・レポートと Essbase モデルの関連付け

ドリルスルー・レポートは、レポートで定義された同じ交差を含むカタログの任意のモデル、またはすべてのモデルと関連付けることができます。

➤ ドリルスルー・レポートと Essbase モデルを関連付けるには:

1 **ドリルスルー・レポート・プロパティ・エディタの「関連付け」タブで、このドリルスルー・レポートと関連付ける Essbase モデル名を選択します。**

現在のドリルスルー・レポートで定義された交差を含むカタログのすべてのモデルを関連付けるには、「すべての Essbase モデルを選択」チェック・ボックスを選択します。

「関連付け」タブで選択が行われていない場合、Oracle Hyperion Smart View for Office では、スプレッドシートのどのセルもドリルスルー・セルとしてマークされません。

2 **オプション:** 現在のドリルスルー・レポートで定義された交差を含む、Essbase モデルの最新のリストを表示するには、「モデル・リストをリフレッシュ」をクリックします。

3 「保存」をクリックしてから「閉じる」をクリックし、ドリルスルー・レポートを終了します。

17 検索とサーチ

この章の内容

検索について 343
メタデータ要素の検索 343
メタデータ要素のサーチ 344

検索について

Essbase Studio では、検索とサーチの 2 つの方法でメタデータ要素を検索できます。

「検索」では、Essbase Studio コンソールの選択されたユーザー・インタフェース・コンポーネント内でメタデータを検索します。たとえば、「メタデータ・ナビゲータ」、ミニスキーマ作業領域または「データ・ソース・ナビゲータ」でメタデータ要素を検索できます。さらに、「メタデータ・ナビゲータ」または「データ・ソース・ナビゲータ」のフォルダを選択して、そのフォルダ内でのみ検索を実行することもできます。

サーチは、メタデータ・カタログ全体をサーチします。つまり、階層やキューブ・スキーマなど、特定のメタデータ要素のタイプでのみサーチを絞り込めます。

343 ページの「メタデータ要素の検索」および344 ページの「メタデータ要素のサーチ」を参照してください。

メタデータ要素の検索

「メタデータ・ナビゲータ」、「ミニスキーマ」ペインまたは「データ・ソース・ナビゲータ」などの UI コンポーネント内でメタデータ要素を検索するには、「検索」を使用します。

➤ メタデータ要素を検索するには:

1 **検索を実行する Essbase Studio コンソールの UI コンポーネント内にカーソルを合わせます。**

たとえば、「データ・ソース・ナビゲータ」でミニスキーマのグラフィック表示内のメタデータ要素を検索するには、データ・ソース接続、「ミニスキーマ」フォルダの順に展開し、ミニスキーマを右クリックして「編集」を選択します。

2 **「編集」>「検索」を選択して「検索」ダイアログ・ボックスを表示します。**

3 **「検索」テキスト・ボックスに検索文字列を入力します。**

たとえば、product という文字列を含むミニスキーマのテーブルまたは列を検索するには、「検索」テキスト・ボックスに product と入力します。

4 **オプション:「検索」を詳細に設定するには、オプションを選択します:**

- 「大文字/小文字の一致」-検索文字列と完全に一致するメタデータ要素のみを検索します。

  たとえば、Product という検索文字列を入力して「大文字/小文字の一致」を選択すると、Product という名前のメタデータ要素のみが検索され、PRODUCT という名前の要素は検索されません。

- 「単語の一致」-単語全体が検索文字列と一致する要素のみを検索します。

  たとえば、product という検索文字列を入力して「単語の一致」を選択すると、product という名前のメタデータ要素は検索されますが、productdim という名前のメタデータ要素は検索されません。

- 「現在の選択に適用」-画面上の特定の選択範囲内にある要素のみを検索します。

  たとえば、「メタデータ・ナビゲータ」で 1 つ以上のフォルダを選択し、検索文字列を入力してから「現在の選択に適用」オプションを選択すると、選択したフォルダ内でのみ検索が実行されます。

5 **「検索」ボタンをクリックします。**

選択した UI 要素内で、検索文字列との最初の一致で検索が停止し、黄色くハイライトされます。たとえば、「ミニスキーマ」ペインのミニスキーマ全体で検索を実行すると、そのミニスキーマ内で、テーブルまたは列のどちらでも、検索文字列との最初の一致で検索が停止します。

6 **次の検索文字列との一致を特定するには、再度「検索」をクリックします。**

7 **「検索」ダイアログ・ボックスの一番下に「これ以上ありません」というメッセージが表示されるまで、手順 6 を繰り返します。**

メタデータ要素のサーチ

「検索」を使用して、カタログに含まれるメタデータ要素を検索します。これは、メタデータ要素ライブラリの規模が大きい場合に便利です。

➤ メタデータ要素をサーチするには:

1 **「編集」を選択してから「メタデータ要素の検索」を選択します。**

サーチ・インタフェースは、Essbase Studio コンソールの左のペイン内のタブとして起動します。

2 **サーチ対象のメタデータ要素の「名前」を入力します。**

3 **「タイプ」で、「<すべて>」を選択するか、ドロップダウン・リストを使用してメタデータ要素の特定のタイプにサーチ対象を絞り込みます。**

次のメタデータ要素に対してサーチを実行できます:

- 階層
- フォルダ

- Essbase モデル
- メタデータ要素
- ドリルスルー・レポート
- キューブ・スキーマ

4 **オプション: サーチ場所を絞り込むには次の手順を実行します:**

1. 「検索範囲」の「参照」をクリックします。「フォルダの検索」ダイアログ・ボックスでサーチ対象のフォルダに移動します。
2. サーチ対象のフォルダをダブルクリックして、ダイアログ・ボックスを閉じます。

デフォルトでは、「メタデータ・ナビゲータ」のコンテンツ全体がサーチされます。

5 **「検索」をクリックして、「検索結果」を確認します。**

6 **オプション: サーチ条件およびサーチ結果を消去して新しいサーチを開始するには、「消去」をクリックして、手順 2 から手順 5 までを繰り返します。**

7 **「検索」ダイアログ・ボックスを閉じるには、「検索」タブの「X」をクリックします。**

注 : 「検索」ダイアログ・ボックスの「名前」フィールドに入力したメタデータ要素名は保持され、今後のサーチ時にドロップダウン・リストから選択できます。

18 プリファレンス

この章の内容

一般プリファレンスの設定 347
スキーマ・プリファレンスの設定 347
Essbase モデルのプリファレンスの設定 348

一般プリファレンスの設定

Essbase Studio コンソールに表示するメッセージや警告に関する一般プリファレンスを設定します。

➤ 一般プリファレンスを設定するには:

1 「編集」を選択し、「プリファレンス」を選択します。

2 「データ・ソース・ナビゲータ」、「メタデータ・ナビゲータ」または作業領域からアイテムを削除するときに確認メッセージを表示するには、「「削除の確認」ダイアログを表示」チェック・ボックスを選択します。

3 コンソール・メッセージ・ペインに表示するメッセージのタイプを制御するには、「コンソール・メッセージ」グループで次のオプションを選択または選択解除します:

- 「情報メッセージを表示」
- 「警告メッセージを表示」
- 「エラー・メッセージを表示」

4 「適用」をクリックして選択項目を適用し、「OK」をクリックします。

注: デフォルトのプリファレンス設定に戻すには、「デフォルトに戻す」をクリックします。

スキーマ・プリファレンスの設定

Essbase Studio コンソールのミニスキーマの表示を制御するには、スキーマ・プリファレンスを設定します。

➤ スキーマ・プリファレンスを設定するには:

1 「編集」を選択し、「プリファレンス」を選択します。

2 **ミニスキーマのテーブル・オブジェクトにあるデータ・ソース・オブジェクトの表示を制御するには、「リレーショナル・テーブルの表示オプション」の次のオプションを選択または選択解除します**

- 列タイプを表す記号を列名の左側に表示するには、「列タイプ・インディケータ」を選択します
- キー記号をプライマリ・キー列の名前の右側に表示するには、「プライマリ・キー」を選択します
- 外部キー関係を持つ列を表す記号を表示するには、「外部キー」を選択します

3 **結合するテーブル間の結合線の表示方法を制御するには、「結合コネクタ」グループから次のオプションを選択します:**

- 「対角線」-テーブル間の結合線を必要に応じて対角線上に描画します。
- 「長方形」-テーブル間の結合線を必要に応じて左または右の角度でのみ描画します。

4 **テーブル間の結合線の末端の表示を制御するには、「結合の末端」グループから次のオプションを選択します:**

- 「矢印」-結合の末端が矢印で表示されます。
- 「クロウズ・フット」-結合の末端が 3 つに分かれたクロウズ・フット・パターンで表示されます。

5 **「適用」をクリックして選択項目を適用し、「OK」をクリックします。**

注： デフォルトのプリファレンス設定に戻すには、「デフォルトに戻す」をクリックします。

Essbase モデルのプリファレンスの設定

特定のユーザー・プロンプトを制御する Essbase モデルのプリファレンスを設定します。

➤ Essbase モデルのプリファレンスを設定するには、次の手順に従います:

1 **「編集」を選択し、「プリファレンス」を選択します。**

2 **次のチェック・ボックスを選択して、各アクションを有効にします:**

- 「集約ストレージを TRUE に設定している場合、モデルを調整するかどうかをユーザーに確認する」
- 「Essbase モデル・プロパティ」ダイアログ・ボックスを閉じる前に変更されたモデルを検証するかユーザーに確認する」

特定のプロンプトを無効にするには、チェック・ボックスの選択を解除します。

3 **「適用」をクリックして選択項目を適用し、「OK」をクリックします。**

注： デフォルトのプリファレンス設定に戻すには、「デフォルトに戻す」をクリックします。

19 Integration Servicesカタログ移行

この章の内容

カタログ移行の概要 351
「EIS カタログの移行」ダイアログ・ボックスへのアクセス 353
EIS モデルおよびメタアウトラインの移行 354
ガイドラインおよび制限 356

カタログ移行の概要

注： この概要で説明されている手順は、データベース管理者、またはリレーショナル・データベースの概念、Integration Services および作業環境における特定のリレーショナル・データベースに精通した人を対象にしています。

Integration Services では、カタログと呼ばれるリレーショナル・リポジトリにメタデータが保管されます。(Integration Services では OLAP メタデータ・カタログと呼ばれています。)Essbase Studio にも、メタデータを含むカタログが存在します。Essbase Studio では、Integration Services カタログ内のメタデータを Essbase Studio カタログに移行するメカニズムが提供されます。

この概要では、次のトピックを説明します:

- 351 ページの「モデルの移行」
- 352 ページの「メタアウトラインの移行」

モデルの移行

Essbase Studio では、Integration Services カタログから Essbase Studio カタログへのモデルの移行時に、いくつかのタスクが実行されます。

Essbase Studio の移行ユーティリティによって、まず Integration Services のモデルと同じ名前のフォルダが作成されます。その後、移行が行われます。

Integration Services のモデルには、テーブルのサブセットおよびスター・スキーマ結合に関する情報を含むスター・スキーマが含まれています。ユーザーが階層を定義している場合、この情報はモデルにも含まれます。

移行ユーティリティでは、このメタデータを Essbase Studio に移行するために次の手順が実行されます:

- スター・スキーマのテーブルごとに、Essbase Studio のモデル・フォルダの下にサブフォルダが作成されます。
- テーブルの列ごとに、各テーブルのフォルダ内に次元要素が作成されます。テーブルの列が各次元要素のキャプションおよびキー・バインディングに反映されます。
- 時間次元要素では、適切な時間の変換が適用され、新しい次元要素が作成されます。
- Integration Services のモデルで定義された階層ごとに、対応する階層が Essbase Studio のモデル・フォルダに作成されます。
- 再帰階層の自己結合を含むすべての結合が移行されるように、Integration Services のモデルのスター・スキーマに対応して、Essbase Studio にミニスキーマが作成されます。

メタアウトラインの移行

メタアウトラインを Integration Services のカタログから Essbase Studio のカタログに移行するときに、Essbase Studio によっていくつかのタスクが実行されます。

Essbase Studio 移行ユーティリティによって、まず Integration Services のメタアウトラインと同じ名前のサブフォルダが(モデル・フォルダ内に)作成されます。その後、移行ユーティリティにより実際の移行が行われます。

この項では、次のトピックを説明します:

- 352 ページの「階層のマッピング」
- 353 ページの「メジャー列のマッピング」
- 353 ページの「Essbase エクスポート・モデルの作成」
- 353 ページの「別名のマッピング、ユーザー定義メンバーおよびレポート」

階層のマッピング

Integration Services メタアウトラインは、次元と属性次元で構成されています。各次元には、代替階層を含められます。移行ユーティリティでは、次のように Integration Services 次元を Essbase Studio 階層にマッピングします:

- 各 Integration Services 次元では、対応する階層が作成されます。
- 各属性次元は、Essbase Studio の代替階層として作成されます。
- 各 Integration Services 再帰次元では、親および子のメンバーを識別することで階層が作成されます。
- Integration Services メタアウトラインに、ファクト・テーブルに基づいていない会計次元が含まれている場合、会計次元は Essbase Studio のファクト・テーブルから作成されません。

メジャー列のマッピング

階層がマッピングされた後、移行ユーティリティによってキューブ・スキーマが作成されます。次いで、メジャー列が次の 2 つのシナリオのいずれかに従ってマッピングされます:

- Integration Services メタアウトラインにファクト・テーブルに基づいた会計次元が含まれている場合、メジャー階層が作成され、キューブ・スキーマに追加されます。
- Integration Services メタアウトラインにファクト・テーブルに基づいていない会計次元が含まれている場合、メジャー列は緩いメジャーとして追加されます。

Essbase エクスポート・モデルの作成

キューブ・スキーマが作成され、メジャー列がマップされた後に、移行ユーティリティにより Essbase エクスポート・モデルが作成され、各エクスポート・モデル要素のプロパティが割り当てられます。

様々なプロパティのカテゴリが、異なるレベルで適用されます:

- Essbase エクスポート・モデルのプロパティ
- 階層プロパティ
- チェーン・プロパティ
- レベル・プロパティ

別名のマッピング、ユーザー定義メンバーおよびレポート

移行プロセスの最終段階では、移行ユーティリティによって次の要素がマッピングされます:

- 別名テーブル・セットが作成されます。
- 別名の次元要素に応じて別名が定義されます。
- Integration Services メタアウトラインのユーザー定義メンバーは、次元要素として Essbase Studio メタアウトライン・フォルダに作成されます。
- Integration Services メタアウトラインのドリルスルー・レポートは、交差情報、列のリスト、行ガバナー、時間ガバナー、テンプレート SQL およびその他のパラメータとともに移行されます。次にドリルスルー・レポートは、エクスポート・モデルと関連付けられます。

「EIS カタログの移行」ダイアログ・ボックスへのアクセス

➤ 「EIS カタログの移行」ダイアログ・ボックスにアクセスするには:

1 **Essbase Studio コンソールで「ツール」メニューにアクセスします。**

2 「EIS カタログの移行」を左クリックします。

EIS モデルおよびメタアウトラインの移行

Essbase Studio に移行する Integration Services カタログを、Essbase Studio と同じマシンに配置する必要はありません。Integration Services カタログは、Integration Services がサポートするデータベースであれば任意の場所に配置できます。

Integration Services カタログが Essbase Studio と同じマシンにある場合でも、Integration Services カタログに ODBC DSN を作成する必要があります。この DSN では、Integration Services カタログがあるデータベースを指定する必要があります。

注： 次の手順は、データベース管理者およびリレーショナル・データベースの概念、Integration Services、および作業環境のリレーショナル・データベースに精通したユーザーを対象としています。

➤ Integration Services モデルおよびメタアウトラインを Integration Services から Essbase Studio に移行するには:

1 「EIS カタログの移行」ダイアログ・ボックスにまだアクセスしていない場合はアクセスします。353 ページの「「EIS カタログの移行」ダイアログ・ボックスへのアクセス」を参照してください。

2 「EIS カタログ DSN」テキスト・ボックスに、移行する Integration Services カタログを示す ODBC DSN を入力します。

注： Integration Services カタログを示す ODBC DSN が存在しない場合、ODBC DSN を作成する必要があります。Integration Services カタログが Essbase Studio と同じマシンにない場合、Essbase Studio クライアントが稼働するマシンに ODBC DSN を作成する必要があります。この DSN では、Integration Services カタログがあるデータベースを指定する必要があります。

3 「ユーザー」テキスト・ボックスに、EIS カタログのユーザー名を入力します。例: `tbc`。

4 EIS カタログのパスワードを入力します。

5 「モデルのフェッチ」ボタンをクリックします。

Essbase Studio が Integration Services カタログにアクセスし、モデルとメタアウトラインがカタログにリスト表示されます。メタアウトラインは個々のモデルでグループ化されます。

6 「モデルおよびデータ・セット」列で、移行する Integration Services モデルおよびメタアウトラインを選択します。

注： 複数のモデルとメタアウトラインを移行できます。

7 **「データ・ソース接続」列をクリックし、ドロップダウン・リストからターゲット・リレーショナル・データベースを選択します。**

注： ターゲット・リレーショナル・データベースは、Integration Services のターゲット・データ・ソースと一致している必要があります。

オプション:「新規データ・ソース」をクリックしてデータ・ソースの名前を入力すると、ターゲット・データ・ソースを作成できます。

注： 複数のモデルを移行する場合、移行するモデルごとに Essbase Studio のターゲット・リレーショナル・データベースを選択する必要があります。

8 **「カタログ」列をクリックし、ドロップダウン・リストから Integration Services カタログを選択します。**

注： リレーショナル・データベースが Oracle の場合、カタログを選択する必要はありません。

9 **「スキーマ」列をクリックし、ドロップダウン・リストからスキーマを選択します。**

注： 複数のモデルを移行する場合、移行するモデルごとにスキーマを選択する必要があります。

10 **「フォルダの選択」テキスト・ボックスに、Essbase Studio のターゲット・フォルダを入力します。**

オプション: ターゲット・フォルダを簡単に検索するには、「参照」ボタンをクリックします。

オプション:「フォルダの選択」テキスト・ボックスにフォルダ名を入力すると、フォルダを作成できます。

注： ターゲット・フォルダのリストは、Essbase Studio の「メタデータ・ナビゲータ」にも表示されます。

11 **「移行」ボタンをクリックします。**

「進捗情報」ダイアログ・ボックスで、移行の進捗を表示します。

12 **オプション: 移行を取り消すには:**

1. 「取消し」を押します。
2. 手順 10 で指定したターゲット・ディレクトリにある任意のオブジェクトを削除します。

OLAP モデルとメタアウトラインが Integration Services カタログから Essbase Studio カタログに移行します。移行が完了すると、モデルが「メタデータ・ナビゲータ」に表示されます。データ・ソースを作成すると、「データ・ソース・ナビゲータ」にそのデータ・ソースの名前が表示されます。

ガイドラインおよび制限

サブトピック

- メタアウトラインとモデルの移行のガイドライン
- 移行されないプロパティ

メタアウトラインとモデルの移行のガイドライン

- Integration Services カタログの移行では、64 ビット UNIX プラットフォームまたは Windows 2008 64 ビット・プラットフォームではサポートされていません。.
- ハイブリッド分析対応のメタアウトラインは移行されません。
- Unicode 対応のメタアウトラインは移行されません。
- メタアウトラインにリストされるテーブルは、すべてソース・データベースに存在している必要があります。
- Integration Services モデルのデータ型は、ソース・データベースのデータ型と一致する必要があります。Essbase Studio はデータ型を確認し、不一致の場合、移行は失敗します。
- 複数の ODBC DSN を持つメタアウトラインは移行されません。
- 再帰階層に定義されたドリルスルーを持つメタアウトラインはサポートされています。移行後、これらの再帰階層ドリルスルー定義は Essbase Studio で使用できます。
- 代替データ・ソースに対して定義されたドリルスルーを持つメタアウトラインは移行できますが、Essbase Studio では、代替データ・ソース・ドリルスルー機能はサポートされていません。
- Essbase Studio は、次元要素のバインディングを使用して固有または重複するアウトラインを作成します。ただし、デフォルトでは、Migration Studio はすべてのメタアウトラインを固有のものとして移行します。Essbase Studio で重複するアウトラインを作成するには、キー列を指定して次元要素のキー・バインディングを変更する必要があります。キー列のデータ型は、次元要素のデータ型と一致する必要があります。
- データ・ソース列がレベル 0 のメンバー・セットにバインドされており、基本メンバーと共有メンバーの両方を含んでいる場合、(同じ次元内に共有メンバーを持つ)特定の基本メンバーに対して Essbase Studio で生成されたデータ・ロード SQL 内に集約されるデータは、共有メンバーの数に応じて増加する可能性があります。この場合、211 ページの「標準データ・ロード SQL の上書き」の説明に従って、ユーザー独自のカスタム・データ・ロード SQL を定義することをお薦めします。この動作は、Integration Services からインポートしたメタアウトラインから作成された Essbase モデルにも、最初から Essbase Studio で作成された Essbase モデルにも当てはまります。

  たとえば、TBC サンプルを使用すると、100-10 で Diet の下に共有メンバーがあるため、100-10 に対して標準データ・ロード SQL 内に集約されるデータは本来の量の 2 倍になります。正しい集約データを取得するには、「データ・

ロードにカスタム SQL を使用」オプションを選択し、手動でカスタム・データ・ロード SQL に group by 列をもう 1 つ追加します

- Integration Services と Essbase Studio では、属性メンバー処理に違いがあります。

  Integration Server では、対応する基本次元に属性関連付けの関係が存在するかどうかにかかわらず、属性次元のすべての属性メンバーが追加されます。このために、Integration Server では、メンバー・ロード中に 3 タイプのルール・ファイルが生成されます: 1 つは基本次元用、1 つは属性次元用、もう 1 つは属性関連付け用です。

  Essbase Studio では、対応する属性関連付けの関係が基本次元に存在する場合にのみ、属性メンバーが作成されます。キューブの配置中、2 つのルール・ファイルのみが生成されます: 1 つは基本次元用、1 つは属性メンバーおよびその関連付け用です。これによりキューブ配置の時間は短縮されますが、基本メンバー関連付けのない属性メンバーはプロセスで削除されます。

  たとえば、Integration Services で TBC サンプル・メタアウトラインを使用して作成されたアウトラインには、次のメンバーを持つ属性次元 Population が含まれます:

```
     Small: 3000000, 6000000
Medium: 900000, 1200000, 1500000, 1800000
Large: 21000000, 24000000, 27000000, 30000000, 33000000
```

  ただし、TBC メタアウトラインが Essbase Studio カタログに移行され、Essbase Server に配置された場合、新しいアウトラインには次のメンバーを持つ属性次元 Population が含まれます:

```
     Small: 3000000, 6000000
Medium: 900000, 1200000, 1500000
Large: 21000000, 33000000
```

  これは、population の値 18000000、24000000、27000000 または 30000000 に都道府県/州がないためです。

  属性次元が再帰的に(親子テーブルから)構築された場合、Essbase Studio では、Integration Services の場合と同じように、3 つのルール・ファイル(基本次元用、属性次元用および属性関連付け用のルール・ファイル)が生成されます。その結果、属性次元には関連付けに関係なくすべてのメンバーが含まれます。

移行されないプロパティ

前述のアイテムの他にも、Essbase Studio 移行ユーティリティにより移行されない Integration Services プロパティがいくつかあります。次のようなプロパティが含まれます(ただし、これに限定されません):

- 余分な結合
- ユーザー定義のデータ・ロード SQL

- ユーザー定義の属性

  注： この制限は、演算子やパススルー SQL による変換ルールに適用されます。ただし、Q、DD、DM、DW、WM、WY、MMM、MM、MONTH、YY および YYYY など、いくつかの日付関連の変換はサポートされています。

- ガバナー
- サンプルストップ・インディケータの表示
- 一意キーの列
- 再帰階層のメンバー名列
- メタアウトライン・レベルのフィルタ
- 上書き

注： 前述のリストは、Essbase Studio の移行ユーティリティでサポートされていない Oracle Essbase Integration Services プロパティを示していますが、これはすべてではなく、その中で最もよく使用されるプロパティを示しています。

A 制限とガイドライン

この付録の内容

概要 359
カタログおよびデータ・ソースのガイドライン 359
イントロスペクションに関する制限 364
メタデータ要素使用のガイドラインと制限 365
階層の使用方法のガイドラインと制限 366
キューブ・スキーマに関する制限 367
Essbase プロパティの編集および用途に関する制限 368
Essbase モデルの再同期のガイドラインと制限 372
キューブ配置の制限とガイドライン 373
ドリルスルー・レポートの制限とガイドライン 375

概要

この付録では、Essbase Studio の操作時に適用される制限事項を説明します。

カタログおよびデータ・ソースのガイドライン

次の各トピックでは、Essbase Studio カタログ、物理データ・ソース、およびデータ・ソース接続に関する制限について説明します。

- 360 ページの「カタログ・データベースにつき 1 つの Essbase Studio サーバー」
- 360 ページの「カタログ・アクセスのガイドライン」
- 360 ページの「カタログおよびデータ・ソース権限のガイドライン」
- 361 ページの「カタログ・エクスポート・ファイルに含まれないパスワード」
- 361 ページの「XML のインポートに関する制限」
- 361 ページの「Oracle クライアント・ドライバのガイドライン」
- 362 ページの「MySQL の制限」
- 362 ページの「Microsoft SQL Server の制限」
- 362 ページの「データ・ソースとしての Excel ファイルはサポートされていない」
- 362 ページの「Oracle Call Interface (OCI)の制限」

- 362 ページの「列のデータ型のサポート」
- 363 ページの「Essbase Studio カタログのアップグレード」
- 364 ページの「ユーザー定義 SQL の検証」
- 364 ページの「ユーザー定義 SQL のデータ・ソース権限」
- 364 ページの「カタログおよびデータ・ソースの一般的な制限事項とガイドライン」

カタログ・データベースにつき 1 つの Essbase Studio サーバー

Essbase Studio カタログ・データベースは、同時か連続かを問わず、複数の Essbase Studio サーバー・インスタンスで使用しないでください。各 Essbase Studio サーバーが固有のカタログ・データベースをポイントするようにすることを強くお薦めします。

カタログ・アクセスのガイドライン

複数のクライアントが Essbase Studio サーバーの 1 つのインスタンスに対して要求を行う際に、クライアントの 1 つがメタデータ要素のデータ・ソースを取得中である場合、アクセスの問題が発生します。データ・ソースを取得する場合、取得によって導出されたメタデータを追加するには、Essbase Studio でカタログへの排他的アクセスが必要です。ログイン要求、階層または次元要素の表示または編集など、Essbase Studio クライアントからのその他の要求は、取得が完了するまでは Essbase Studio サーバーによって処理されません。

要求を実行できないことを示すプロンプトが表示されるまでクライアントが待機する時間は、構成可能です。詳細は、46 ページの「server.readLockTimeOut」および 46 ページの「server.writeLockTimeOut」を参照してください。

また、94 ページの「リレーショナル・ソースのテーブルの選択」の説明のように、「接続ウィザード」の「テーブルの選択」ページの「参照時にカタログをロック」チェックボックスを選択することで、取得プロセス中にカタログをロック解除することも選択できます。

カタログおよびデータ・ソース権限のガイドライン

- Essbase Studio カタログ・データベース・ユーザーには、少なくとも Essbase Studio カタログ・データベースへの書込み権限が必要です。カタログ・データベース・ユーザーは、`server.properties` ファイルの `catalog.username` および `catalog.password` プロパティで指定したユーザーです。このユーザーには、少なくとも `catalog.db` プロパティで指定したデータベースへの書込み権限が必要です。

- Essbase Studio カタログ・データベースに Oracle を使用する場合は、カタログ・データベースへの接続に使用するユーザーに次の権限を設定することをお薦めします。

```
open_cursors=nnn SCOPE=MEMORY
```

  nnn >= 300 です。
- 接続ウィザードでデータ・ソース接続を設定するときに、Essbase Studio ユーザーは、少なくとも接続先の外部データ・ソース(データベース)への読取り権限があるユーザー名とパスワードを指定する必要があります。
- ドリルスルー・レポートのテンプレート SQL およびカスタム・データ・ロード SQL を使用することで発生する問題を回避するために、外部データ・ソースへの読取り権限のみをユーザーに付与してください。

Oracle クライアント・ドライバのガイドライン

DataDirect Oracle ODBC ドライバでは、Microsoft Oracle ODBC ドライバとは異なる接続文字列のフォーマットを使用します。Essbase Studio で使用する接続文字列のフォーマットは、DataDirect Oracle ODBC ドライバ向けに設計されています。

非 DSN 接続を使用する場合、次のガイドラインに注意してください:

- Oracle への非 DSN 接続は、DataDirect Wire Protocol ODBC ドライバのみでの使用を対象としています。
- IBM DB2 への非 DSN 接続は、DataDirect Wire Protocol ODBC ドライバのみでの使用を対象としています。
- Microsoft SQL Server への非 DSN 接続は、Microsoft SQL Server ODBC ドライバと DataDirect Wire Protocol ODBC ドライバでの使用を対象としています。
- Teradata への非 DSN 接続は、ネイティブの Teradata ODBC ドライバのみでの使用を対象としています。DataDirect Wire Protocol ODBC ドライバでの使用は対象外です。

カタログ・エクスポート・ファイルに含まれないパスワード

データ・ソース接続をエクスポートする際、ソース・データベースに対するパスワードは生成される XML ファイルに含まれません。XML ファイルをインポートした後で、118 ページの「データ・ソース接続プロパティの編集」の説明に従って、接続を編集してパスワードを追加し直します。

XML のインポートに関する制限

インポートする XML ファイルに含まれるデータ・ソース接続がターゲット・カタログにも存在する場合、ミニスキーマおよび他のオブジェクト(表やビュー、シノニムなど)はインポートされません。

回避策: エクスポートされたデータ・ソースと既存のデータ・ソースに相違がある場合は、次のタスクを完了します:

1. データ・ソース接続を右クリックして、「増分更新」を選択します。「増分更新」ダイアログ・ボックスで、データ・ソース接続に追加する表、ビューおよびシノニムを選択します。
2. データ・ソース接続内の個別の表、ビューおよびシノニムを右クリックして「リフレッシュ」を選択し、列を追加または削除してプロパティを更新します。

MySQL の制限

MySQL では、データベース・テーブルに対する変更(列の名前変更など)が、対応する MySQL ビューに正しく反映されません。このため、Essbase Studio コンソールでビューを含む MySQL データ・ソースをリフレッシュすると、予期しない結果が発生します。

Microsoft SQL Server の制限

Essbase Studio では、Microsoft SQL Server の NTEXT データ型はサポートされません。

データ・ソースとしての Excel ファイルはサポートされていない

Essbase Studio では、Microsoft Excel ファイルはデータ・ソースとしてサポートされていません。

Oracle Call Interface (OCI)の制限

Essbase Studio では、Oracle Call Interface (OCI)を使用したデータ・ソース接続の作成はサポートされません。

列のデータ型のサポート

Essbase Studio は JDBC インタフェースを使用して、リレーショナル列のデータ型を抽出します。リレーショナル列と論理メタデータ要素のデータ型の間に次の関連付けが作成されます。

- 整数との関連付け:

  INTEGER

  TINYINT

  SMALLINT

  BIT

- long との関連付け:

  BIGINT
- ブールとの関連付け:

  BOOLEAN
- 日付との関連付け:

  DATE
- 時間との関連付け:

  TIME
- 日時との関連付け:

  TIMESTAMP
- double との関連付け:

  DOUBLE

  FLOAT

  REAL

  DECIMAL

  NUMERIC
- 文字列との関連付け:

  CHAR

  VARCHAR

  NCHAR

  NVARCHAR

  LONGVARCHAR

  CLOB

  VARBINARY
- バイナリとの関連付け:

  BINARY

  BLOB

  LONGVARBINARY

  および、上記のリストに含まれていないその他の SQL データ型

Essbase Studio カタログのアップグレード

Essbase Studio カタログのエクスポート/インポートを使用して、あるリリースのカタログ・データベースを別のリリースのカタログ・データベースに移行することはできません。かわりに、Oracle Enterprise Performance Management System インス

トールおよび構成ガイドに記載されているアップグレードの手順に従う必要があります。

ユーザー定義 SQL の検証

Essbase Studio サーバーでは、データ・ロードやドリルスルー操作およびユーザー定義テーブルに使用される SQL 文に対して、検証が実行されません。Essbase Studio では文が記述どおりに実行されます。ユーザー定義 SQL を検証するには、それを実行する必要があります。364 ページの「ユーザー定義 SQL のデータ・ソース権限」も参照してください。

ユーザー定義 SQL のデータ・ソース権限

Essbase Studio におけるユーザー定義 SQL に対する権限は、データ・ソース・ユーザーに割り当てられている権限から継承されます。データ・ソース・ユーザーの権限の制限は、Essbase Studio 管理者が責任を持って設定する必要があります。364 ページの「ユーザー定義 SQL の検証」も参照してください。

カタログおよびデータ・ソースの一般的な制限事項とガイドライン

- テキスト・ファイルのデータ・ソースのディレクトリ名と、個々のテキスト・ファイル名にスペースを含めることはできません。
- テキスト・ファイルのデータ・ソース・ディレクトリのパスの長さには、次の制限があります:
  - ネイティブ・モード: 121 バイト
  - Unicode モード: 1028 バイト
- 次のデータ・ソース・エンティティでは、単一引用符(')は使用できません:
  - リレーショナル・データ・ソース・スキーマ、カタログ、テーブルまたは列の名前。
  - テキスト・ファイルのデータ・ソース・ディレクトリ、ファイルまたは列の名前。
- データ・ソース名(DNS)では、英語以外の文字はサポートされていません。
- サンプル・データの表示ウィンドウが開いている間にソース・データが追加、編集または削除されても、開いているウィンドウ内のデータは自動的にはリフレッシュされません。変更されたデータを表示するには、ウィンドウを閉じて、「サンプル・データの表示」コマンドを再発行する必要があります。

イントロスペクションに関する制限

- イントロスペクションは、次のデータ・ソースではサポートされません:
  - テキスト・ファイル

- Oracle BI EE

メタデータ要素使用のガイドラインと制限

次の各トピックでは、メタデータ要素の使用制限を説明します。

- 365 ページの「導出テキスト・メジャーに関する制限」
- 365 ページの「サイクル依存関係のガイドライン」
- 365 ページの「Oracle BI EE のビジネス・モデル次元要素のキー・バインディングとキャプション・バインディングの生成ルール」

導出テキスト・メジャーに関する制限

- テキスト・ファイルの接続の要素から導出テキスト・メジャーは作成できません。
- キューブ・スキーマで導出テキスト・メジャーを使用して Essbase モデルを構築する場合、そのモデルに対して「XOLAP モデル」オプションおよび「Oracle OBIEE による XOLAP モデル」・オプションは使用できません。

サイクル依存関係のガイドライン

Essbase Studio では、メタデータ要素間のサイクル依存関係は確認されません。サイクル依存関係とは、自身に戻ってくるメタデータ要素間の関係です。

たとえば、メタデータ要素 A が式の定義にメタデータ要素 B を使用し、メタデータ要素 B が式の定義にメタデータ要素 C を使用し、メタデータ要素 C が式の定義にメタデータ要素 A を使用するとします。この場合、これらの要素を定義する式には、Essbase Studio で確認されないサイクル依存関係があります。

サイクル依存関係を持つメタデータ要素を作成すると、これらの要素間のサイクル依存関係を自己責任において検証し続ける必要があります。

Oracle BI EE のビジネス・モデル次元要素のキー・バインディングとキャプション・バインディングの生成ルール

Oracle BI EE のビジネス・モデル・データ・ソースの参照中に、Essbase Studio は、選択された各 Oracle BI EE 次元に対して階層を作成します。少なくとも 1 つのプライマリ・キー列を含む各 Oracle BI EE 次元レベルに対して、Essbase Studio は Oracle BI EE 次元の指定されたレベルを表す次元要素を作成します。

次に、これらの次元要素用のバインディング式の生成に使用されるルールを示します:

キー・バインディング・ルール

- ルール 1 - Oracle BI EE 次元レベルに単一のプライマリ・キーが含まれる場合、そのプライマリ・キー列が次元要素のキー・バインディングになります。
- ルール 2 - Oracle BI EE 次元レベルに複数のプライマリ・キーが含まれる場合、キー・バインディング式は、区切り記号にアンダースコア(_)を使用してすべてのプライマリ・キー(単位が `string` でない場合には `string` に変換)を連結した結果になります。

注： Oracle BI EE 次元レベルに論理プライマリ・キー列が含まれない場合、Essbase Studio は指定された Oracle BI EE 次元レベルに対して次元要素を作成しません。

キャプション・バインディング・ルール

- ルール 1 - キー・バインディング式の単位が `string` (複数のプライマリ・キーの連結結果ではない)の場合、次元要素のキャプション・バインディングはキー・バインディング式と同一になります。
- ルール 2 - キー・バインディング式の単位が `string` ではない場合は、キー列(プライマリ・キー列ではない)であり、型が CHAR または VARCHAR である最初の Oracle BI EE 論理列レベルが、指定された次元要素のキャプション・バインディング式として使用されます。
- ルール 3 - キー・バインディング式の単位が `string` ではない場合で、型が CHAR または VARCHAR のキー列がない場合は、キー・バインディング変換.`toString` が指定された次元要素のキャプション・バインディング式として使用されます。
- ルール 4 - キー・バインディング式の単位が複数のプライマリ・キーの連結結果として `string` である場合は、キャプション・バインディング式はキー・バインディング式と同一になります。

階層の使用方法のガイドラインと制限

次の各トピックでは、階層の使用に関するガイドラインと制限について説明します:

- 366 ページの「階層のガイドライン」
- 367 ページの「標準階層およびメジャー階層に関する制限」
- 367 ページの「カレンダ階層に関する制限」

階層のガイドライン

標準階層およびメジャー階層には、次のガイドラインが適用されます:

- リレーショナル・データ・ソースを操作する場合:
  - 「メタデータ・ナビゲータ」または「データ・ソース・ナビゲータ」の列から階層の列を追加できます。

- シングルチェーン階層の列は、単一データ・ソース内の 1 つ以上のテーブルから取得する必要があります。
- マルチチェーン階層では、階層の各チェーンは異なるデータ・ソースから取得できますが、マルチチェーン階層のシングルチェーンでは、単一データ・ソース内の 1 つ以上のテーブルから列を取得する必要があります。
- 複数のデータ・ソースから構築されたマルチチェーン階層を使用してキューブ・スキーマを作成する場合は、204 ページの「データ・ロード・マッピングの定義」で説明しているように、階層の各チェーンの最下位レベル(Gen1 またはリーフ・レベル)でデータ・ロード・バインディングをオーバーライドする必要があります。

- テキスト・ファイルのデータ・ソースを操作する場合、階層内の列(シングルチェーンまたはマルチチェーン)は、単一データ・ソース内の同一テキスト・ファイルから取得する必要があります。

標準階層およびメジャー階層に関する制限

標準階層およびメジャー階層には、次の制限が適用されます:

- データ・ソースの列またはメタデータ要素がすでに階層チェーンに存在する場合、列または要素を階層チェーンに追加できません。
- 階層チェーンに複数のデータ・ソースの列またはメタデータ要素をドラッグ・アンド・ドロップする場合には注意が必要です。1 つ以上の列または要素がすでにチェーンに存在する場合、ドロップは拒否されます。複数の列または要素を階層チェーンにドラッグする場合、チェーンに存在していない列または要素を選択してください。

カレンダ階層に関する制限

カレンダ階層には次の制限が適用されます:

- Essbase モデルにカレンダ階層から作成された次元が含まれている場合、次の制限が適用されます:
  - テキスト・ファイルのデータ・ソースはサポートされません。
  - 配置の場合、Essbase Studio サーバーをストリーミング・モードで実行する必要があります(42 ページの「server.essbase.streamingCubeBuilding」を参照)。
  - キューブ配置時、「データ・ロード・オプション」グループで「既存データに上書き」オプションは選択できません。

キューブ・スキーマに関する制限

これらの制限はキューブ・スキーマに適用されます:

- 既存のキューブ・スキーマからすでに Essbase モデルが作成されている場合、そのキューブ・スキーマにデータ・ロード・バインディングを追加すると、キューブ・スキーマとモデルは非同期になります。既存の Essbase モデルから

配置する場合、そのモデルは新しいデータ・ロード・バインディングをピック・アップしないため、配置結果は無効になります。新しいデータ・ロード・バインディングを使用してキューブを配置するには、新しい Essbase モデルを作成し、その新しいモデルから配置する必要があります。

Essbase プロパティの編集および用途に関する制限

次のトピックでは、Essbase モデル・プロパティの編集および用途に関する制限について説明します:

- 368 ページの「Essbase モデルの再構築のガイドライン」
- 369 ページの「カスタム・データ・ロード SQL のガイドライン」
- 369 ページの「重複するメンバー名のサポートの制限事項」
- 369 ページの「XOLAP の機能のガイドライン」
- 371 ページの「独立次元のバインディングに関する制限」
- 371 ページの「可変属性編集のガイドライン」
- 372 ページの「テキスト・ファイルのデータ・ソース・メンバーの変換の制限」

Essbase モデルの再構築のガイドライン

このトピックでは、特定のメタデータ要素やキューブ・スキーマを変更する場合の Essbase モデルの処理を説明します。変更にはモデルの再構築が必要かどうかも記載します。

Essbase モデルの再構築が不要な操作

次の操作をメタデータ・フォルダ、次元要素、導出テキスト・メジャー、テキスト・リスト、階層、メジャー階層またはキューブ・スキーマに対して行った場合、Essbase モデルの再作成または再構築は必要ありません:

- 名前変更
- 移動

さらに、次の操作を実行した場合も Essbase モデルの再作成または再構築は必要ありません:

- 次元要素のバインディング、フィルタ、ソート順または別名セット・バインディングの変更
- 導出テキスト・メジャーのバインディング、範囲または別名セット・バインディングの変更
- テキスト・リストの値バインディングまたは ID バインディングの変更
- キューブ・スキーマでの上書きデータ・ロード・バインディングの変更
- 可変属性を含むモデルの場合:

- 可変属性バインディングを含む階層の名前の変更
- 可変属性次元でリーフ・レベルとして使用される次元要素の名前の変更
- 可変属性の独立次元である階層の名前の変更

Essbase モデルの再構築が必要な操作

次の操作を行った場合、Essbase モデルの再作成または再構築が必要であることに注意してください:

- 階層コンテンツの変更; 階層またはメジャー階層でのメンバーの順序変更、追加または削除など
- キューブ・スキーマでの次元性の変更; 階層の追加や削除など
- キューブ・スキーマでの緩いメジャーの追加または削除
- キューブ・スキーマでのメジャー階層の変更
- キューブ・スキーマでのデフォルトのデータ・ロード・バインディングの上書き

カスタム・データ・ロード SQL のガイドライン

重複メンバー名がサポートされる Essbase モデルでカスタム・データ・ロード SQL を使用しないでください。Essbase Studio では、内部的に特別なタグが生成され、重複メンバーが一意に識別されます。

重複するメンバー名のサポートの制限事項

重複するメンバーのサポートが使用可能になっている Essbase モデルでは、遅延キー・バインディングが含まれる文字列以外の次元要素がモデルに存在する場合、検証エラーが発生します。Essbase Studio サーバーでは、配置時に必要な連結されたキー・バインディングを生成できません。

この問題を回避するには、Essbase モデルの「重複するメンバー名のサポート」オプションの選択を解除します。ただし、これを実行できない場合でも、その他の解決方法があります:

- 遅延キー・バインディングを含む次元要素に、キー・バインディングを設定します。

  159 ページの「次元要素のキー・バインディング・オプションの選択」を参照してください。
- 使用されている場合は、階層の次元要素にキー・バインディングを設定します。

  182 ページの「階層での遅延キー・バインディング列の使用」を参照してください。

XOLAP の機能のガイドライン

215 ページの「XOLAP を使用するためのガイドライン」も参照してください。

次のガイドラインが XOLAP の機能に適用されます:

- すべてのチェーンの結合パスが次元テーブルの同じセットに由来している場合、Essbase Studio では、XOLAP 対応の Essbase モデルの配置時に代替階層がサポートされます。次に示すサンプルの代替階層では、説明のために TBC データベースが使用されています。
  - 次の例では、ファクト・テーブルへの結合パスは同じ次元テーブルに由来しています。

```
        - PKGTYPE       (TBC.PRODUCT)
 - SKU        (TBC.PRODUCT)
- OUNCES        (TBC.PRODUCT)
 - SKU_ALIAS     (TBC.PRODUCT)
```

    この場合、SALES ファクト・テーブルへの結合パスは次のようになります:

    TBC.PRODUCT joins TBC.SALES.

  - 次の例では、ファクト・テーブルへの結合パスは次元テーブルの同じセットに由来しています。

```
        - FAMILY       (TBC.FAMILY)
 - SKU        (TBC.PRODUCT)
- FAMILYID       (TBC.FAMILY)
 - SKU_ALIAS     (TBC.PRODUCT)
```

    この場合、SALES ファクト・テーブルへの結合パスは次のようになります:

    TBC.FAMILY joins TBC.PRODUCT joins TBC.SALES.

  - 次の例では、ファクト・テーブルへの結合パスは異なる次元テーブルに由来しています。

```
        - UDAMKTTYPE      (TBC.MARKET)
 - STATE       (TBC.MARKET)
- PKGTYPE       (TBC.PRODUCT)
 - SKU        (TBC.PRODUCT)
```

    この場合、SALES ファクト・テーブルへの異なる結合パスが 2 つあるため、代替階層シナリオはサポートされません:

    1. TBC.MARKET joins TBC.SALES

    2. TBC.PRODUCT joins TBC .SALES

  - 次の例では、ファクト・テーブルへの結合パスは次元テーブルの異なるセットに由来しています。

```
        - FAMILYID       (TBC.FAMILY)
```

```
  - INTRODATE     (TBC.PRODUCT)
- SKU_ALIAS       (TBC.PRODUCTDIM)
  - CAFFEINATED    (TBC.PRODUCT)
```

この場合、SALES ファクト・テーブルへの異なる結合パスが 2 つあるため、代替階層シナリオはサポートされません:

1. TBC.FAMILY joins TBC.PRODUCT joins TBC.SALES
2. TBC.PRODUCTDIM joins TBC.PRODUCT joins TBC .SALES

- 「Essbase モデル・プロパティ」ダイアログ・ボックスの「アウトライン構築」タブで、代替階層が含まれる次元を「複合階層使用可能」としてタグ付けできます。
- Essbase Studio は XOLAP 対応の Essbase モデルに属性次元を作成できます; ただし、属性次元の子レベルは 1 つのみであることが必要です。

独立次元のバインディングに関する制限

これらの制限は、可変属性で使用される独立次元のバインディングの定義に適用されます:

- 独立次元のバインディングを定義する式は、一意、または重複のアウトラインの配置に適用されます。ただし、式は一意と重複のメンバーに同時には適用できません。
- 一意のアウトライン配置の場合、式の結果は独立メンバーのクラスのキャプション・バインディングと一致する必要があります。
- 重複アウトライン配置の場合、式の結果は独立メンバーのクラスのキー・バインディングと一致する必要があります。

可変属性編集のガイドライン

次のシナリオでは、可変属性の再定義または編集が必要です:

- カタログのエクスポート/インポートを実行すると、Essbase モデルの可変属性情報が失われます。カタログのエクスポート/インポート後は、メンバーが含まれる Essbase モデルですべての可変属性メンバーを再定義する必要があります。
- モデル内で使用されるデータ・ソース接続の名前を変更すると、古い接続名を参照するモデルの可変属性メンバーの同期が保たれなくなります。新しいデータ・ソース接続名を参照できるように可変属性メンバーを編集する必要があります。「Essbase モデル・プロパティ」ダイアログ・ボックスで、可変属性メンバーを探して「可変属性の編集」ダイアログ・ボックスにアクセスします。「独立次元設定」グループで、次の処理を行います:
  - 可変属性のタイプが個別の場合は、「開始」列で式を編集します。
  - 可変属性のタイプが範囲の場合は、「開始」列および「終了」列で式を編集します。

可変属性の作成および編集の詳細は、249 ページの「メンバーの可変属性の設定」を参照してください。

テキスト・ファイルのデータ・ソース・メンバーの変換の制限

テキスト・ファイルのデータ・ソースに二重引用符(`""`)が含まれている場合、変換ルールの作成時にはそれらを無視します。キューブ配置プロセスでは、二重引用符は Essbase によって削除されます。たとえば、二重引用符を含むテキスト値の`"2000-01-01"`の場合、2000 という年値を導出するには`substr(1,4)`を使用します。

Essbase ビジネス・モデル・ソースに基づく Essbase モデル

Essbase ビジネス・モデル・データ・ソースに基づく Essbase モデルを使用する場合は、「重複するメンバー名のサポート」オプションを有効化することをお薦めします。「重複するメンバー名のサポート」オプションを選択すると、Essbase ビジネス・モデルに基づいた Essbase キューブで発生する可能性が高いメンバー名の競合を解消できます。

Essbase モデルの再同期のガイドラインと制限

- 基本チェーンと属性チェーンを含むマルチチェーン階層で、基本チェーンを削除すると、Essbase モデルの再同期操作は成功しますが、このモデルのキューブの配置に失敗します。

  Essbase モデルの再同期操作では、モデルの検証は行われません。再同期されたモデルにエラーがあっても、ユーザーには通知されません。したがって、モデルの検証を別途行う必要があります。そのためには、「Essbase モデル・プロパティ」ダイアログ・ボックスでモデルを開き、282 ページの「モデル・プロパティの検証」の手順を実行します。

- 基本チェーンと属性チェーンを含むマルチチェーン階層で、属性チェーンの中間レベルを追加または削除し、モデルの再同期を実行すると、属性次元から属性設定が削除されます。

  属性チェーンのレベルを追加または削除するたびに、モデルの再同期を実行し、影響を受ける Essbase モデルの属性プロパティまたは可変属性プロパティを確認して再設定する必要があります。また、キューブの配置を試行する前にモデルを検証することをお薦めします。

- 可変属性に含まれる階層のリーフ・レベルを削除し、モデルの再同期を実行すると、再同期は成功したように見えますが、可変属性が機能しなくなります。この階層を使用するキューブの配置は失敗します。

モデルの再同期を実行したら、影響を受ける Essbase モデルの属性プロパティまたは可変属性プロパティを確認し、必要に応じて再設定してください。また、キューブの配置を試行する前に必ずモデルを検証してください。

キューブ配置の制限とガイドライン

サブトピック

- キューブ配置に関する一般的な制限
- 管理サービス・コンソールでのルール・ファイルの制限
- XOLAP キューブの Essbase Studio サーバーで生成された MaxL について
- Essbase Studio ルール・ファイルでの MaxL の Import Dimensions ステートメントの使用

キューブ配置に関する一般的な制限

次の一般的な制限がキューブの配置に適用されます:

- カレンダ階層から作成された次元が含まれる Essbase モデルからデータをロードする場合、次の制限が適用されます:
  - テキスト・ファイルのデータ・ソースはサポートされません。
  - Essbase Studio サーバーをストリーミング・モードで実行する必要があります(42 ページの「server.essbase.streamingCubeBuilding」を参照)。
  - 「データ・ロード・オプション」グループでは、「既存データに上書き」オプションを選択することはできません。
- Essbase モデルの作成元のキューブ・スキーマにデータ・ロード・バインディングを追加すると、キューブ・スキーマとモデルは非同期になります。既存の Essbase モデルから配置する場合、そのモデルは新しいデータ・ロード・バインディングをピック・アップしないため、配置結果は無効になります。新しいデータ・ロード・バインディングを使用してキューブを配置するには、新しい Essbase モデルを作成し、その新しいモデルから配置する必要があります。
- XOLAP 対応モデルの増分構築がサポートされています。

  XOLAP 対応の Essbase モデルから再配置する場合、「増分ロード」オプション、新しいアプリケーション名およびデータベース名、または「最初にすべてのメンバーを削除」オプションを使用して既存の XOLAP アプリケーションの上に配置できます。
- Oracle BI EE データ・ソースに基づいた XOLAP 対応の Essbase モデルは配置できません。
- 非ストリーミング・モード(`server.properties` で `server.essbase.streamingCubeBuilding=false`)の場合は、Essbase Studio はリリース 10.1.3.4 以降の Oracle BI EE データ・ソースからのみキューブを配置できます。`server.essbase.streamingCubeBuilding` プロパティがストリーミングに設定(`server.essbase.streamingCubeBuilding=true`)されている場合にのみ、以前のリリースの Oracle Business Intelligence Enterprise Edition (リリース 10.1.3.3)からキューブを配置できます。

このプロパティの詳細は、42 ページの「server.essbase.streamingCubeBuilding」を参照してください。

- **IBM DB2:** SELECT DISTINCT 文での LONG VARCHAR データ型の処理における IBM DB2 の制限により、キューブの配置が失敗します。この失敗を回避するには、`server.essbase.disableDistinct` を true に設定します。

  このプロパティの設定の詳細は、47 ページの「server.essbase.disableDistinct」を参照してください。

- データ・ソース列がレベル 0 のメンバー・セットにバインドされており、基本メンバーと共有メンバーの両方を含んでいる場合、(同じ次元内に共有メンバーを持つ)特定の基本メンバーに対して Essbase Studio で生成されたデータ・ロード SQL 内に集約されるデータは、共有メンバーの数に応じて増加する可能性があります。この場合、211 ページの「標準データ・ロード SQL の上書き」の説明に従って、ユーザー独自のカスタム・データ・ロード SQL を定義することをお薦めします。

  たとえば、TBC サンプルを使用すると、100-10 で Diet の下に共有メンバーがあるため、100-10 に対して標準データ・ロード SQL 内に集約されるデータは本来の量の 2 倍になります。正しい集約データを取得するには、手動でカスタム・データ・ロード SQL に group by 列(Family)をもう 1 つ追加し、「データ・ロードにカスタム SQL を使用」オプションを選択します。

管理サービス・コンソールでのルール・ファイルの制限

管理サービス・コンソールでは、Essbase Studio により生成されたルール・ファイルを使用する際に、次のガイドラインが適用されます:

- データ・ロードのルール・ファイルは読取り専用です; 編集はできません。
- 次元構築(メンバー・ロード)のルール・ファイルは開くことも表示することもできません。
- メジャー階層のメンバーが 250 を超える場合、Essbase Studio は 250 のメンバーごとに 1 つのルール・ファイルを作成します。

XOLAP キューブの Essbase Studio サーバーで生成された MaxL について

「キューブ配置ウィザード」でオプションを選択し、選択内容を MaxL スクリプトとして保存することで、MaxL の deploy ステートメントを生成できます。

XOLAP キューブを再配置する際、Essbase Studio サーバーで生成された MaxL を編集して配置設定を削除または変更すると、データに一貫性がなくなる可能性があります。

XOLAP キューブを再配置するために、Essbase Studio サーバーで生成された MaxL スクリプトを編集して配置設定を変更しないでください。かわりに、「キューブ配

置ウィザード」を起動して必要な選択を行い、その選択内容を新しい MaxL スクリプトに保存することをお薦めします。

Essbase Studio ルール・ファイルでの MaxL の Import Dimensions ステートメントの使用

Essbase Studio で作成したルール・ファイルから、MaxL の Deploy ステートメントを使用して Essbase にメンバーやデータをロードできます。このタスクは Deploy ステートメントを使用すると最も簡単に実行できます。

ただし、MaxL の Import Dimensions ステートメントを使用して Essbase Studio で生成されたルール・ファイルからメンバーをロードすることも可能です。Import Dimensions ステートメントを使用してアウトラインを構築する場合は、ファクト・テーブルまたはユーザー定義メンバーのみが含まれる階層からの会計次元の構築に問題があることに注意してください。この場合は、MaxL の Import Dimensions ステートメントで呼び出せる、空のダミー・テキスト・ファイルを用意します。例:

```
   import database 'tbc.MaxL1' dimension from local text data_file 'C:\dummy.txt'
using server rules_file 'Account' on error append to 'C:\Hyperion\textUn1.log';
```

ドリルスルー・レポートの制限とガイドライン

サブトピック

- ドリルスルー操作の一般的な制限事項
- ドリルスルーにおけるポート番号の制限

ドリルスルー操作の一般的な制限事項

- ドリルスルーのテンプレート SQL を使用するドリルスルー・レポートと重複メンバー名をサポートする Essbase モデルを関連付けないでください。Essbase Studio では、内部的に特別なタグが生成され、重複メンバーが一意に識別されます。
- ドリルスルーのターゲット URL の長さは、最大で 256 文字までです。
- Oracle Hyperion EPM Architect のソースから作成された階層は、ドリルスルー・レポートでサポートされません。
- Essbase Studio で構築されたキューブでは、セル範囲に対するドリルスルーは使用できません。

ドリルスルーにおけるポート番号の制限

Oracle Hyperion Enterprise Performance Management System コンフィグレータを使用した初期構成後に、`server.properties` ファイルの `transport.port` プロパティを手動で変更した場合、ポートの変更前に構築されたキューブに対するドリルスルー接続は正しく機能しません。308 ページの「キューブ・リンケージの更新」の手順に従って、キューブに保管されているポート番号をリフレッシュする必要があります。

注： ポート変更後に構築されたキューブは正常に機能します。

B Essbase Studioの命名規則

この付録の内容

データベースとアプリケーションの命名規則 377
メタデータ要素の命名規則..... 378
ドリルスルー・レポートの命名ガイドライン 380
Essbase 命名規則の例外 381

データベースとアプリケーションの命名規則

次の Essbase 命名ルールがデータベース名とアプリケーション名に適用されます:

- 非 Unicode モードのアプリケーションとデータベースの命名では 8 バイト以下の文字列を使用します。
- Unicode モードのアプリケーションとデータベースの命名では 30 文字以下の文字列を使用します。
- スペースは使用しません。
- 表 7 に記載されている文字は使用しません。

表 7　アプリケーション名とデータベース名で使用できない文字のリスト

文字	説明
*	アスタリスク
[]	大かっこ
:	コロン
;	セミコロン
,	カンマ
=	等号
>	大なり記号
<	小なり記号
.	ピリオド
+	プラス符号
?	疑問符

文字	説明
"	二重引用符
'	単一引用符
/	スラッシュ
\	円記号(バックスラッシュ)
|	垂直棒
	タブ

- データベース名には、次の語を使用しないでください:
  - 文字列 drxxxxxx(大文字と小文字は区別しない)
  - Replay(予約語)
- 集約ストレージ・データベースの場合は、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
REPLAY
TEMP
```

アプリケーション名とデータベース名では、大文字と小文字が区別されます。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)。

メタデータ要素の命名規則

次のトピックでは、メタデータ要素に適用する Essbase の命名ルールを説明します:

- 378 ページの「Essbase 命名ルールの制限を受けるメタデータ要素」
- 379 ページの「メタデータ要素の命名規則」

Essbase 命名ルールの制限を受けるメタデータ要素

Essbase 命名ルールは、次のメタデータ要素に適用されます:

- すべての階層タイプ
- 次元要素

注: 列に基づいていない次元要素; つまり、文字列定数のバインディングは Essbase 命名ルールの制限を受けます。

- 導出テキスト・メジャー
- 別名セット名
- ドリルスルー・レポート

メタデータ要素の命名規則

次のトピックでは、メタデータ要素に適用される命名規則をリストします:

- 379 ページの「メタデータ要素の命名の一般的なガイドライン」
- 380 ページの「使用できない文字」
- 380 ページの「予約語」

メタデータ要素の命名の一般的なガイドライン

メタデータ要素の名前付けを行うときは、次のガイドラインに従います:

注： 通常、Essbase Studio のメタデータ要素の名前に疑問がある場合は、Essbase 命名ルールに従えば安全ですが、381 ページの「Essbase 命名規則の例外」を参照してください。

- メタデータ要素の名前付けを行うときは、80 バイト以下の文字列を使用します。
- 名前では、大文字と小文字の区別を使用可能にしていない場合には、大文字と小文字は区別されません。

  Oracle Essbase Administration Services Online Help のアウトライン・プロパティの設定を参照してください。
- 集約ストレージ・アウトラインで重複メンバー名が使用可能になっている場合、大文字と小文字の区別が使用可能になっていても、同じ名前(大文字と小文字のみが異なる)をメタデータ要素の名前として使用しないでください。たとえば、次元に Product という名前を付けた場合、別の次元に product という名前を付けないでください。
- 名前の中に、二重引用符(" ")、大かっこ([])またはタブを使用しないでください。
- 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、そのようなスペースは無視されます。
- 計算スクリプトのコマンド、演算子およびキーワード
- レポート・ライターのコマンド
- 関数名および関数の引数
- その他のメタデータ要素の名前(メンバーが共有でない場合)

注： 動的時系列メンバーを使用可能にしている場合は、関連する世代名の履歴、年、季、期間、四半期、月、週および日は使用しないでください。

使用できない文字

次元名やメンバー名の先頭では、表 8 の文字は使わないでください:

表 8　メタデータ要素名で制限されている文字のリスト

文字	説明
@	アット・マーク
\	円記号(バックスラッシュ)
{ }	中かっこ
,	カンマ
-	ダッシュ、ハイフンまたはマイナス
=	等号
<	小なり記号
()	丸かっこ
.	ピリオド
+	プラス符号
'	単一引用符
_	アンダースコア
|	垂直棒

予約語

次の名前の変更時には、予約語を使用できます:

- Essbase キューブ
- キューブ・スキーマ
- Essbase モデル
- メタデータ要素

次の語は、階層またはユーザー定義メンバーの名前に使用できません。

ALL AND ASSIGN AVERAGE CALC CALCMBR COPYFORWARD CROSSDIM CURMBRNAME DIM DIMNAME DIV DY

ドリルスルー・レポートの命名ガイドライン

- ドリルスルー・レポートの名前を付けるときは、次のガイドラインに従います:
  - 末尾に円記号(バックスラッシュ)(\)は使用できません
  - パイプ文字(|)は使用できません

- 単一引用符(')は使用できません

Essbase 命名規則の例外

Oracle Essbase の大部分の命名規則は Essbase Studio にも適用されますが、特定の条件下では、次のような一部の Essbase Studio エンティティに異なる命名ルールが存在します:

- フォルダ
- データ・ソース
- ミニスキーマ
- キューブ・スキーマ

C

CPLリファレンス

この付録の内容

CPL 式の概要 383
オペランド 383
演算子 388
CPL SQL 関数 393

CPL 式の概要

次元要素の「ドリルスルー・フィルタ」や「プロパティの編集」などの Oracle Essbase Studio ツールで、共通プラットフォーム言語(Common Platform Language: CPL)式を作成してデータ・ソースの情報にアクセスできます。式は、言語定義の構文に従ったオペランドと演算子のシーケンスです。各式では値が戻され、その値の型によって式の型が定義されます。

注： CPL 式のすべての部分式は、同じ接続から取得する必要があります。

CPL 式は次のものから構成されます:

- 383 ページの「オペランド」
- 388 ページの「演算子」

オペランド

オペランドとは、式の操作の対象となるオブジェクトです。また、データを表し、データにアクセスするために使用される名前です。

構文の要素

オペランドには、次の要素が含まれます。構文は、値のタイプとデータ・ソースによって異なります:

- `<KEYWORD>`

  キーワードは、値のタイプを表します。たとえば、キーワードは、式の値のタイプが `connection` なのか `class` なのかを示します。

- :

コロン(:)は、キーワードの値のタイプを式に割り当てます。

- \

  円記号(バックスラッシュ)("\")は、フォルダを表します。フォルダは、その他のオブジェクトを結合したり、その他のフォルダを含めたり、カタログのオブジェクトの場所を定義したりするオブジェクトです。フォルダには、同じ名前で同じタイプの 2 つのオブジェクトは含められませんが、同じ名前で異なるタイプの 2 つのオブジェクトを含めることはできます。

  フォルダは次のように表されます:

  ```
  <folder>
  :: =
  \
  |
   <folder> <folder_name> \
  ```

  ルート・フォルダへのアクセス:

  ```
  \
  ```

  フォルダへのアクセス:

  ```
  \'folder'\
  ```

  サブフォルダへのアクセス:

  ```
  \'folder'\'subfolder'\
  ```

- '<NAMEx>'

  要素名は単一引用符で囲む必要があります。名前はデータ・ソースに応じて、接続、クラス、カタログ、スキーマ、テーブル、フォルダまたはサブフォルダになります。

- ::

  2 つのコロン(::)は、データベース接続またはテキスト・ファイルにアクセスします。たとえば、tbc というデータベース接続の情報にアクセスする場合は、次のようになります:

  ```
  connection : \'tbc'
  ::
  'tbc.family'.'FAMILY'
  ```

- .

  ドット(.)は、列など、テーブルやファイルに含まれている属性または値にアクセスします。

接続

接続では、リレーショナル・テーブル、多次元キューブまたはフラット・ファイルなど、データを取得する物理データ・ソースにアクセスします。connection キーワードは接続を示しており、ソースによって従う構文は異なります。

注： 接続はルート・フォルダにのみ存在します。

2 つのレベルを持つデータベース

データベース接続には、次の 2 つのレベルがあります:

```
connection : \ '<CONNECTION_NAME>'::'<CATALOG/SCHEMA>.<TABLE>'.'<COLUMN>'
```

例:

```
connection : \'tbc'::'tbc.family'.'FAMILY'
```

3 つのレベルを持つデータベース

データベース接続に 3 つのレベルがある場合があります:

```
connection : \ '<CONNECTION_NAME>'::'<CATALOG>.<SCHEMA>.<TABLE>'.'<COLUMN>'
```

例:

```
connection : \'tbc'::'tbc.tbc.family'.'FAMILY'
```

フラット・ファイル(1 レベル)

フラット・ファイルには、次のように 1 つのレベルのみが含まれます:

```
connection : \ '<DATA_SOURCE_NAME>'::'<FILENAME>'.'<COLUMN>'
```

例:

```
connection : \'DataFile'::'dmdemo.txt'.'Actual'
```

ユーザー定義テーブル

ユーザー定義テーブルには、任意の名前を付けられますが、単一引用符は使用できません:

```
connection : \ '<DATA_SOURCE_NAME>'::'<USER_DEFINED_FILE_NAME>'.'<COLUMN>'
```

例:

```
connection : \'tbc'::'myTable'.'FAMILY'
```

クラス

クラスは、ビジネス・エンティティを表す論理要素で、物理データ・ソースにバインドできます。単純なクラスは、より複雑なクラスの基準になります。論理要素には、整数、数値または文字列などの単位があります。各クラスには、そのデータを表す外部データ・ソースへのバインディングがあります。

クラスを示し、次の一般パターンに従う `class` キーワード:

```
class :
 <folder> '<CLASS_NAME>'
```

例:

```
class : \'folder1'\'folder2'\'My Object'
```

注： クラスはどのフォルダにも配置できます。

定数

定数および定数リテラルは、一時的な値を保管するオブジェクトです。定数は基本値型を表し、定数リテラルは複合値型の値を表します。定数は、階層またはクラスで参照されます。

- **数値定数**

  数値定数の単位は数値です。

  符号なし正数:

  ```
  31.27
  ```

  符号付き負数:

  ```
  -10.4
  ```

  指数:

  ```
  5.4e+4
  ```

- **正数定数**

  整数定数の単位は整数です。

  符号なし正数:

  ```
  1
  ```

  符号付き負数:

  ```
  -45
  ```

- **long 型定数**

  long 型定数の単位は long 型です。

  例:

  ```
  1L
  ```

  例:

  ```
  100000000000001L
  ```

- **ブール定数**

  ブール定数の単位はブールです。

  ブールは、評価条件に応じて TRUE または FALSE のいずれかを戻します。

- **文字列定数**

  文字列定数の単位は文字列です。文字列定数は次の方法で表記できます:

  - 引用符を使用する方法:

    ```
    "Example of string value"
    ```

    二重引用符を文字列内で使用する場合、文字列内の二重引用符を 2 つ続けて入力する必要があります。例:

    ```
    "Example of string value
    ""
    with quote inside"
    ```

  - 大かっこを使用する方法:

    ```
    $[[Example of string value " with quote inside]]$
    ```

  - かっこを使用する方法:

```
$((Example of string value " with quote inside))$
```

大かっこおよびかっこ内の文字列定数には、引用符などの任意の記号を文字列として含めることができます。パーサーからは特殊文字ではなく文字列として解釈されます。

演算子

演算子は、オペランドに対して実行され値が戻されるコマンド式です。各演算子には、名前、優先度およびシグネチャがあります。シグネチャでは、複数のオペランドの型の値とその値のタイプが定義されます。

計算の順序

表 9 に示す順序で複数の演算子を評価します。優先度 1 が最高で、優先度 8 が最低です。

表 9　演算子の優先度

優先度	演算子
1	( )
2	not + - (Unary)
3	* / mod
4	+ - (Binary)
5	||
6	< <= > >= == !=
7	and
8	or

同じ優先度のオペランドは、左から右の順に実行されます。

グループ化

部分式を結合するには、グループ化演算子を使用します。

() グループ化-かっこ

カッコで囲まれたグループは、演算子の標準の優先度よりも優先度が高くなります。また、オペランドを囲んで読みやすくするなど、式の要素を明確にするためにも使用できます。

例:

```
(7-2)*6
```

例

判読を容易にするためにオペランドを囲む

```
(connection :
 \'DB_conn'::'TBC.tbc.SALESFACT'.'OPENINGINVENTORY')
```

接続

```
   (
 (connection :
  \'DB_conn'::'TBC.tbc.SALESFACT'.'OPENINGINVENTORY')
-
 (connection :
  \'DB_conn'::'TBC.tbc.SALESFACT'.'SALES')
)
*
connection : \'DB_conn'::'TBC.tbc.SALESFACT'.'COGS'
```

論理演算子

条件を評価するには、論理演算子を使用します。

not ブール-単項 not。

戻り値:

- TRUE-条件が FALSE の場合
- FALSE-条件が TRUE の場合

例:

```
not true
```

`and` ブール-2 項 and。可換。

戻り値:

- TRUE-両方の条件が TRUE の場合
- FALSE-いずれかの条件が FALSE の場合

例:

```
true and false
```

`or` ブール-2 項 or。可換。

戻り値:

- TRUE-いずれかの条件が TRUE の場合
- FALSE-両方の条件が FALSE の場合

例:

```
true or false
```

`==` 条件-等しい

例:

```
2 == 1
```

`!=` 条件-等しくない

例:

```
"Denver" != "Maine"
```

`<` 序数条件-より小さい

例:

```
0 < -0.1
```

`>` 序数条件-より大きい

例:

```
10 > 0.1
```

`<=` 序数条件-以下

例:

```
0 <= 50
```

>= 序数条件-以上

例:

```
10 >= 0
```

例

クラスの使用

```
class : \'folder1'\'FAMILY' != "100"
```

接続

```
connection :
\'DB_conn'::'TBC.tbc.SALESFACT'.'SALES' < 25000
```

算術演算子

算術演算子は算術演算を行います。

+ 加算または正数-可換です。

- **単項:** 正の整数

  例:

  ```
  +1
  ```

- **2 項:** 加算

  例:

  ```
  2 + 3
  ```

- 減算または負数-非可換です。
  - **単項:** 負の整数

    例:

    ```
    -1
    ```

  - **2 項:** 減算

    例:

    ```
    3 - 2
    ```

* 乗算-可換です。

  例:

  ```
  5 * 10
  ```

/ Division–非可換です。

  例:

  ```
  100 / 10
  ```

mod 絶対値(整数)-除数と非除数の余りを戻します。

  例:

  ```
  18 mod 5
  ```

  3 を戻します。

例

接続

```
   (connection :
 \'DB_conn'::'TBC.tbc.SALESFACT'.'SALES')
-
(connection :
 \'DB_conn'::'TBC.tbc.SALESFACT'.'COGS')
```

グループ化演算子

```
   (
 (connection :
  \'DB_conn'::'TBC.tbc.SALESFACT'.'OPENINGINVENTORY')
```

```
-
 (connection :
  \'DB_conn'::'TBC.tbc.SALESFACT'.'SALES')
)
*
connection : \'DB_conn'::'TBC.tbc.SALESFACT'.'COGS'
```

文字列演算子

文字列では、文字列演算子を使用します。

`||` 連結

連結は、文字列式または文字セットのみに適用できます。複数の文字列を 1 つに組み合せます。非可換的です。

例

接続を使用した文字列の構築

```
   connection : \'tbc'::'tbc.region'.'UDA'
|| "_"
|| connection : \'tbc'::'tbc.region'.'REGION'
```

変換された数値を持つ文字列の構築

```
   "Sales " ||
 (connection : \'tbc'::'tbc.salesfact'.'SALES' . toString)
```

CPL SQL 関数

SQL 関数は標準 SQL ルールをサポートしています。

日付

SQL の日時関数です。

dayOfMonth

与えられた日付の日付番号を戻します。

構文

```
   'dayOfMonth' ( <datetime> )
```

パラメータ

- <datetime>は、日時オペランドです。

戻り値

整数

例

```
'dayOfMonth' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```

month

与えられた日付の月番号を戻します。

構文

```
'month' ( <datetime> )
```

パラメータ

- <datetime>は、日時オペランドです。

戻り値

整数

例

```
'month' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```

monthName

与えられた日付の月名を戻します。

構文

```
'monthName' ( <datetime> )
```

パラメータ

- <datetime>は、日時オペランドです。

戻り値

文字列

例

```
'monthName' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```

monthShortName

与えられた日付の月の省略名を戻します。

構文

```
'monthShortName' ( <datetime> )
```

パラメータ

- <datetime>は、日時オペランドです。

戻り値

文字列

例

```
'monthShortName' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```

quarter

与えられた日付の四半期番号を戻します。

構文

```
'quarter' ( <datetime> )
```

パラメータ

- <datetime>は、日時オペランドです。

戻り値

整数

例

```
'quarter' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```

quarterAsString

与えられた日付の四半期の名前を戻します。

構文

```
'quarterAsString' ( <datetime> )
```

パラメータ

- <datetime>は、日時オペランドです。

戻り値

文字列

例

```
'quarterAsString' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```

weekday

与えられた日付の曜日番号を戻します。

構文

```
'weekday' ( <datetime> )
```

パラメータ

- <datetime>は、日時オペランドです。

戻り値

整数

例

```
'weekday' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```

weekdayName

与えられた日付の曜日の名前を戻します。

構文

```
'weekdayName' ( <datetime> )
```

パラメータ

- <datetime>は、日時オペランドです。

戻り値

文字列

例

```
'weekdayName' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```

weekdayShortName

与えられた日付の曜日のショート名を戻します。

構文

```
'weekdayShortName' ( <datetime> )
```

パラメータ

- <datetime>は、日時オペランドです。

戻り値

文字列

例

```
'weekdayShortName' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```

年

与えられた日付の年を戻します。

構文

```
'year' ( <datetime> )
```

パラメータ

- <datetime>は、日時オペランドです。

戻り値

文字列

例

```
'year' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```

yearShort

与えられた日付の年の省略名を戻します。

構文

```
'yearShort' ( <datetime> )
```

パラメータ

- <datetime>は、日時オペランドです。

戻り値

文字列

例

```
'yearShort' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```

文字列

文字列は、SQL の文字列ルールに従います。したがって、文字列の最初の位置は 1 になります。たとえば、次の文字列では位置 1 に「A」があります。また、例の下に各位置が示されています:

```
“A  s t r i n g”

| | | | | | | |

1 2 3 4 5 6 7 8
```

contains

文字列全体に該当の部分文字列が存在しているかどうかを判断します。

構文

```
contains ( <fullStr> , <substr> )
```

パラメータ

- <fullStr>は、文字列全体です。
- <substr>は、文字列全体に含まれる検索対象となる部分文字列です。

戻り値

ブール

- TRUE-部分文字列が文字列内に存在する場合
- false–文字列に部分文字列がない場合

例

```
contains ( connection : \'DB'::'TBC.tbc.MARKET'.'STATE' , "e" )
```

index

文字列全体にある部分文字列の開始位置を示す数値を戻します。

構文

```
index ( <fullStr> , <subStr> )
```

パラメータ

- <fullStr>は、文字列全体です。
- <substr>は、文字列全体に含まれる検索対象となる部分文字列です。

戻り値

整数

例

```
index ( connection : \'DB'::'TBC.tbc.MARKET'.'STATE' , "e" )
```

leftStr

入力したインデックス番号の左にある入力文字列の部分文字列を戻します。

構文

```
leftStr ( <string> , <index> )
```

パラメータ

- <string>は、文字列です。
- <index>は、インデックス番号です。

戻り値

文字列

例

```
leftStr ( connection : \'DB'::'TBC.tbc.MARKET'.'STATE' , 2 )
```

length

入力文字列のスペースを含む文字数を戻します。

構文

```
length ( <string> )
```

パラメータ

- <string>は、文字列です。

戻り値

整数

例

```
length ( connection : \'DB'::'TBC.tbc.MARKET'.'STATE' )
```

lower

入力文字列のアルファベット文字をすべて小文字に変換します。

構文

```
lower ( <string> )
```

パラメータ

- <string>は、文字列です。

戻り値

文字列

例

```
lower ( connection : \'DB'::'TBC.tbc.MARKET'.'STATE' )
```

lTrim

入力文字列の前のスペースを削除します。

構文

```
lTrim ( <string> )
```

パラメータ

- <string>は、文字列です。

戻り値

文字列

例

```
lTrim ( connection : \'DB'::'TBC.tbc.MARKET.STATE' )
```

rightStr

入力番号インデックスの右に入力文字列の部分文字列を戻します。

構文

```
rightStr ( <string> <index> )
```

パラメータ

- <string>は、文字列です。
- <index>は、インデックス番号です。

戻り値

文字列

例

```
rightStr ( connection : \'DB'::'TBC.tbc.MARKET'.'STATE' , 2 )
```

rTrim

入力文字列の末尾のスペースを削除します。

構文

```
rTrim ( <string> )
```

パラメータ

- <string>は、文字列です。

戻り値

文字列

例

```
rTrim ( connection : \'DB'::'TBC.tbc.MARKET'.'STATE' )
```

soundex

入力文字列の音声記号を戻します。

構文

```
soundex ( <string> )
```

パラメータ

- <string>は、文字列です。

戻り値

文字列

例

```
soundex ( connection : \'DB'::'TBC.tbc.MARKET'.'STATE' )
```

substr

入力文字列の部分文字列です。

構文

```
substr ( <string> , <startNumber> , <numberOfCharacters> )
```

パラメータ

- <string>は、文字列です。
- <startNumber>は、部分文字列の開始位置を表す番号です。
- <numberOfCharacters>は、部分文字列の長さです。

戻り値

文字列

例

```
substr ( connection : \'DB'::'TBC.tbc.MARKET'.'STATE' , 1 , 5 )
```

trim

入力文字列の前後のスペースを削除します。

構文

```
trim ( <string> )
```

パラメータ

- <string>は、文字列です。

戻り値

文字列

例

```
trim ( connection : \'DB'::'TBC.tbc.MARKET'.'STATE' )
```

upper

入力文字列のアルファベット文字をすべて大文字に変換します。

構文

```
upper ( <string> )
```

パラメータ

- <string>は、文字列です。

戻り値

文字列

例

```
upper ( connection : \'DB'::'TBC.tbc.MARKET'.'STATE' )
```

数値

abs

入力数値の絶対値を戻します。

構文

```
abs ( <number> )
```

パラメータ

- <number>は数値です。

戻り値

数値

例

```
abs ( connection : \'tbc'::'tbc.salesfact'.'SALES' )
```

exp

e(2.7182818)を arg 乗した値を戻します。

構文

```
exp ( <number> )
```

パラメータ

- <number>は数値です。

戻り値

数値

例

```
exp ( 3 )
```

ln

入力数値の自然対数を戻します。

構文

```
ln ( <number> )
```

パラメータ

- <number>は数値です。

戻り値

数値

例

```
ln ( 46 )
```

log10

入力した数値の 10 を底とする対数を戻します。

構文

```
log10 ( <number> )
```

パラメータ

- <number>は数値です。

戻り値

数値

例

```
log10 ( 136 )
```

pow

y の累乗値 x を戻します。

構文

```
pow ( <number_x> , <number_y> )
```

パラメータ

- <number_x>は数値です。

- <number_y>は数値です。

戻り値

数値

例

```
pow ( 25 , 2 )
```

sqrt

入力数値の平方根を戻します。

構文

```
sqrt ( <number> )
```

パラメータ

- <number>は数値です。

戻り値

数値

例

```
sqrt ( 16 )
```

用語集

!「感嘆符」を参照してください。

#MISSING「欠落データ」を参照してください。

2 パス 他のメンバーの計算済の値に依存するメンバーを再計算するために使用される、Essbase のプロパティです。2 パスのメンバーは、2 番目のパスの間にアウトラインから計算されます。

Calculation Manager Planning および Financial Management のユーザーがグラフィカルな環境でビジネス・ルールを設計、検証、管理するために使用できる、Enterprise Performance Management Architecture(EPMA)のモジュールの 1 つです。

CDF「カスタム定義関数」を参照してください。

CDM「カスタム定義マクロ」を参照してください。

Cookie Web サイトによってコンピュータ上に配置されたデータのセグメントです。

EPM Oracle インスタンス EPM System 製品のアクティブで動的なコンポーネント(実行時に変更できるコンポーネント)が含まれているディレクトリです。EPM Oracle インスタンス・ディレクトリの場所は構成時に EPM System コンフィグレータを使用して定義します。

EPM Oracle ホーム EPM System 製品で必要なファイルを含むミドルウェア・ホームのサブディレクトリです。EPM Oracle ホームの場所は、EPM System インストーラでのインストール中に指定されます。

essbase.cfg Essbase のオプションの構成ファイルです。管理者は、このファイルを編集して Essbase サーバー機能をカスタマイズできます。一部の構成は、Essbase クライアントで使用することにより、Essbase サーバーの設定を上書きできます。

ESSCMD Essbase の操作を対話的に実行したり、バッチ・スクリプト・ファイルから実行したりするための、コマンドライン・インタフェースです。

ESSLANG テキスト文字を解釈するために使用されるエンコード方式を定義する Essbase 環境変数です。「エンコード方式」も参照してください。

ESSMSH「MaxL シェル」を参照してください。

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

Extract-Transform-Load (ETL) データを抽出してアプリケーションに移行するための、データ・ソース固有のプログラムです。

GUI グラフィカル・ユーザー・インタフェース

ID 外部認証におけるユーザーまたはグループの固有の ID です。

Interactive Reporting 接続ファイル(.oce) データベース API(ODBC、SQL*Net、など)、データベース・ソフトウェア、データベース・サーバーのネットワーク・アドレス、データベース・ユーザー名などのデータベース接続情報を含むファイルです。管理者は、Interactive Reporting 接続(.oce)ファイルを作成して発行します。

Java アプリケーション・サーバー・クラスタ Java 仮想マシン(JVM)のアクティブ-アクティブ・アプリケーション・サーバー・クラスタです。

Java データベース接続(JDBC) Java ベース・クライアントとリレーショナル・データベースにより使用されるクライアントとサーバー間の通信プロトコルです。JDBC インタフェースにより、SQL ベースのデータベース・アクセスのための呼出しレベル API が提供されます。

JSP Java Server Page です。

KeyContacts ガジェット Smart Space ユーザーのグループを含み、Smart Space Collaborator へのアクセスを提供するガジェットです。たとえば、マーケティング・チームおよび開発チーム向けに個別の KeyContacts ガジェットを使用できます。「ガジェット」も参照してください。

LRO「リンク・レポート・オブジェクト」を参照してください。

MaxL Essbase で使用される多次元データベース向けアクセス言語です。データ定義の言語(MaxL DDL)とデータ操作の言語(MaxL DML)から構成されます。「MaxL DDL」、「MaxL DML」および「MaxL シェル」も参照してください。

MaxL DDL Essbase で、バッチまたは対話的なシステム管理のタスクに使用されるデータ定義の言語です。

MaxL DML Essbase で、データのクエリーと抽出に使用されるデータ操作の言語です。

MaxL DML の計算済メンバー 分析を目的として設計されたメンバーです。MaxL DML クエリーのオプションの WITH セクションで定義されます。

MaxL Perl モジュール Essbase MaxL DDL の一部である Perl モジュール(essbase.pm)です。このモジュールを Perl パッケージに追加すると、Perl プログラムから Essbase データベースにアクセスできます。

MaxL シェル MaxL ステートメントを Essbase サーバーに渡すためのインタフェースです。MaxL シェルの実行可能ファイル(UNIX は essmsh、Windows は essmsh.exe)は、Essbase の bin ディレクトリに格納されています。

MaxL スクリプト・エディタ 管理サービス・コンソールのスクリプト開発環境です。MaxL スクリプトで Essbase を管理するとき、テキスト・エディタと MaxL シェルの代替として MaxL スクリプト・エディタを使用できます。

MDX (多次元式) 多次元準拠データベースでクエリーと計算に使用される言語です。

MIME タイプ オブジェクトを開くアプリケーションをシステムが判断するための、アイテムのデータ・フォーマットを示す属性。ファイルの MIME(Multipurpose Internet Mail Extension)タイプはファイル拡張子または HTTP ヘッダーにより決定されます。プラグインはブラウザに対して、サポートされる MIME タイプ、および各 MIME タイプに対応するファイル拡張子を通知します。

NULL 値 データのない値です。NULL 値はゼロに等しくありません。

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

OLAP メタデータ・カタログ Essbase 統合サービスにおける、リレーショナル・データ・ソースから引き出されたデータの性質、ソース、場所および種別について説明するメタデータを含むリレーショナル・データベースです。

OLAP モデル Essbase Integration Services における、リレーショナル・データベースのテーブルおよび列から作成された論理モデル(スター・スキーマ)です。OLAP モデルは多次元データベースの構造を生成するために使用されます。「オンライン分析プロセス(OLAP)」も参照してください。

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

Oracle ホーム 特定の製品に必要なインストール済ファイルを含むディレクトリで、ミドルウェア・ホームのディレクトリ構造内にあります。「ミドルウェア・ホーム」も参照してください。

PL 勘定(P&L) 損益勘定です。P&L は通常、会社の損益計算書を構成する費用勘定と収益勘定の集合を指します。

POV (視点) 行、列、またはページ軸に割り当てられていないメンバーを選択することにより、データ・フォーカスを設定する機能です。たとえば、FDM での POV の選択項目には、場所、期間、カテゴリ、およびターゲット・カテゴリが含まれる可能性があります。また、Smart View で POV をフィルタとして使用すると、Currency 次元を POV に割り当て、Euro メンバーを選択できます。データ・フォームで POV を選択すると、ユーロ建でデータが表示されます。

Production Reporting「SQR Production Reporting」を参照してください。

PVA「期別価額メソッド」を参照してください。

root メンバー 次元分岐における最上位のメンバーです。

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

SPF ファイル SQR Production Reporting Server で作成される、プリンタに依存しないファイルです。フォント、間隔、ヘッダー、フッターなど、フォーマットされた実際のレポート出力を表します。

SQL スプレッドシート SQL クエリーの結果セットを表示するデータ・オブジェクトです。

SQR Production Reporting データ・アクセス、データ操作、および SQR Production Reporting ドキュメント作成のための専用プログラミング言語です。

Structured Query Language (SQL) リレーショナル・データベースに対する指示を処理するために使用される言語です。

TCP/IP「Transmission Control Protocol/Internet Protocol」を参照してください。

Transmission Control Protocol/Internet Protocol (TCP/IP) 異なるオペレーティング・システムおよび内部アーキテクチャを持つコンピュータをリンクする標準的な通信プロトコルのセットです。TCP/IP を使用すると、LAN および WAN に接続する多様なコンピュータとの間でのファイルの交換、メールの送信、およびデータの格納が可能です。

Unicode モードのアプリケーション 文字テキストが UTF-8 でエンコードされている Essbase アプリケーションです。様々な言語に設定されているコンピュータを使用するユーザーが、アプリケーション・データを共有できます。

WebLogic Server ホーム WebLogic Server インスタンスで必要なインストール済ファイルを含むミドルウェア・ホームのサブディレクトリです。WebLogic Server ホームは、Oracle ホームのピアです。

WITH セクション MaxL DML で、オプションで使用できるクエリーのセクションです。セットまたはメンバーを定義する再利用可能な論理を作成するために使用されます。WITH セクションでセットまたはカスタムのメンバーを一度定義すると、クエリー中に複数回参照できます。

Workspace ページ 複数のソース(ドキュメント、URL、その他のコンテンツ・タイプ)からのコンテンツを使用して作成されるページです。Oracle およびその他のソースからのコンテンツを集約するために使用できます。

ws.conf Windows プラットフォーム用の構成ファイルです。

wsconf_platform UNIX プラットフォーム用の構成ファイルです。

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

XOLAP アウトラインのメンバーのみを格納し、クエリー時にリレーショナル・データベースからすべてデータを取得する、Essbase の多次元データベースです。XOLAP は、集約ストレージ・データベースおよび重複メンバー名を含むアプリケーションをサポートします。

Y 軸スケール「調査」セクションに表示される、チャートの Y 軸上の値の範囲です。たとえば、各チャートについて一意の Y 軸スケールを使用したり、すべての詳細チャートに同一の Y 軸スケールを使用したり、または列内のすべてのチャートに同一の Y 軸スケールを使用することが可能です。多くの場合、共通の Y 軸スケールを使用すると、一目でチャートを比較できるようになります。

Zero Administration サーバー上のプラグインの最新バージョン番号を識別するソフトウェア・ツールです。

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

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

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

「変更済」ステータス エンティティのデータが変更されたことを示す集計ステータスです。

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

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

お気に入りガジェット Reporting and Analysis ドキュメントと URL へのリンクを含むガジェットです。「ガジェット」も参照してください。

アウトライン 多次元データベースのデータベース構造です。すべての次元、メンバー、タグ、タイプ、集計、および算術的関係を含みます。データは、アウトラインに定義された構造に応じてデータベースに保管されます。

アウトライン同期 パーティション・データベースで、データベースのアウトラインの変更を他のデータベースに伝播するプロセスです。

アクセス権 リソースに対してユーザーが実行できる一連の操作です。

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

アクティブ-アクティブ高可用性システム すべての使用可能メンバーが要求に対応でき、アイドル状態のメンバーがないシステムです。通常、アクティブ-アクティブ・システムは、アクティブ-パッシブ・システムより豊富な拡張性オプションを備えています。「アクティブ-パッシブ高可用性システム」と対比してください。

アクティブ-パッシブ高可用性システム 常に要求に対応するアクティブ・メンバーと、アクティブ・メンバーに障害が発生した場合にのみアクティブ化されるパッシブ・メンバーが存在するシステムです。「アクティブ-アクティブ高可用性システム」と対比してください。

アクティブ・サービス 実行タイプが保留ではなく開始に設定されているサービスです。

アセンブリ EPM System 製品またはコンポーネントのインストール・ファイルです。

アダプタ ターゲットおよびソース・システムのデータやメタデータにプログラムを統合するためのソフトウェアです。

アップグレード 新しいソフトウェアのリリースを配置、アプリケーションやデータを移動、前の配置から新しい配置へ情報をプロビジョニングするプロセスです。

アドホック・レポート エンド・ユーザーが動的に作成するオンライン分析クエリーです。

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

アプリケーションの通貨 アプリケーションのデフォルトのレポート用通貨です。

アプリケーション・サーバー・クラスタ 同時に稼働し、信頼性と拡張性を実現するために連携している複数のアプリケーション・サーバーが緩やかに結合したグループです。ユーザーには 1 つのアプリケーション・サーバー・インスタンスのように見えます。「垂直アプリケーション・クラスタ」および「水平アプリケーション・クラスタ」も参照してください。

アプリケーション移行ユーティリティ アプリケーションとアーティファクトの移行に使用されるコマンド・ライン・ユーティリティです。

アプリケーション管理者 アプリケーションの設定、構成、維持、および制御の担当者です。アプリケーションのすべての権限およびデータ・アクセス権を持ちます。

アペンダ 宛先を意味する Log4j の用語です。

アーティファクト 個別のアプリケーションまたはリポジトリ・アイテムです(スクリプト、フォーム、ルール・ファイル、Interactive Reporting ドキュメント、財務レポートなど)。オブジェクトとも呼ばれます。

イメージ・ブックマーク Web ページやリポジトリ・アイテムへのグラフィック・リンクです。

インストール・アセンブリ EPM System インストーラにプラグインする製品インストール・ファイルです。

インデックス 1)疎データの組合せによりブロック・ストレージ・データベースでデータを取得する方法です。2)インデックス・ファイルを指します。

インデックス・キャッシュ インデックス・ページを含むバッファです。

インデックス・ファイル ブロック・ストレージのデータ取得情報を格納する Essbase ファイルです。ディスクに常駐し、インデックス・ページを含みます。

インデックス・ページ インデックス・ファイルの下位区分です。インデックス・ページには、データ・ブロックを指すポインタが含まれます。

インデックス項目 疎次元の交差へのポインタです。インデックス・エントリはディスク上のデータ・ブロックをポイントし、オフセットを使用してセルを検索します。

イントロスペクション データベース固有の関係に基づいて階層を判断するために、データ・ソースを詳細に検査することを指します。「スクレーピング」と対比してください。

インポート・フォーマット FDM における、ソース・ファイルの構造の定義です。これにより、ソース・データ・ファイルを FDM データのロード位置にロードできます。

エクスプローラ クライアント/サーバー・ベースのツール。データ・ソースに直接アクセスしたり、リポジトリに格納された構築済データ・モデルに編成された情報を参照したりする必要がある上級ユーザーにクエリー、分析およびレポート作成機能を提供します。

エンコード方式 テキストの作成、保管、表示のためにビット組合せを文字にマッピングする方法の 1 つです。各エンコード方式には UTF-8 などの名前が付けられています。各エンコード方式では、それぞれの値は特定のビット組合せにマッピングされています。たとえば、UTF-8 では大文字の A は HEX41 にマッピングされています。「コード・ページ」、「ロケール」も参照してください。

エンタープライズ・ビュー Administration Services の機能の 1 つです。グラフィカルなツリー・ビューを使用して Essbase 環境を管理できます。エンタープライズ・ビューを使用すると、Essbase アーティファクトを直接操作できます。

エンティティ 部門、子会社、工場、地域、製品など、組織での財務報告単位例: 部門、子会社、工場、地域、製品や財務報告単位。

エージェント アプリケーションやデータベースの開始と停止、ユーザー接続の管理、ユーザー・アクセスのセキュリティ対策を行う Essbase サーバー・プロセスです。このエージェントは、ESSBASE.EXE と呼ばれます。

オンライン分析プロセス(OLAP) 複数ユーザーによりクライアントとサーバー間の計算を行える多次元の環境です。集計された企業データをリアル・タイムで分析するユーザーが使用します。OLAP システムには、ドリルダウン、データのピボット、複雑な計算、トレンド分析およびモデリングの機能があります。

カスケード メンバー値のサブセットに対して複数のレポートを作成するプロセスです。

カスタム・カレンダ システム管理者が作成したカレンダです。

カスタム・プロパティ ユーザーが作成した次元または次元メンバーのプロパティです。

カスタム・レポート 設計レポート・モジュールで作成する複雑なレポートです。コンポーネントの様々な組合せによって構成されます。

カスタム定義マクロ(CDM) Essbase のマクロです。Essbase 計算機の関数および専用マクロ関数を使用して記述されます。カスタム定義マクロが使用する Essbase の内部マクロ言語により、計算関数を組み合せることが可能となり、複数の入力パラメータが処理されます。「カスタム定義関数」も参照してください。

カスタム定義関数(CDF) Java で開発され、MaxL により Essbase の標準計算スクリプト言語に追加された、Essbase 計算関数です。「カスタム定義マクロ」も参照してください。

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

カタログ・ペイン アクティブ・セクションで利用可能な要素のリストを表示する領域です。クエリーがアクティブ・セクションの場合はデータベース・テーブルのリストが表示されます。ピボットがアクティブ・セクションの場合は結果列のリストが表示されます。ダッシュボードの場合は埋込み可能セクション、グラフィック・ツール、およびコントロール・ツールが表示されます。

カテゴリ データ編成に使用するグループです(月など)。

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

ガジェット EPM のコンテンツを容易に表示し、Reporting and Analysis のコア機能にアクセスできる、軽量で単純な専用アプリケーションです。

キャッシュ データを一時的に保持する、メモリー内のバッファです。

キューブ 3 つ以上の次元を含むデータのブロックです。Essbase データベースはキューブです。

キューブ・スキーマ Essbase Studio におけるメジャーおよび階層などのメタデータ要素です。キューブの論理モデルを指します。

キューブ配置 Essbase Studio で、アウトラインを構築してデータを Essbase アプリケーションおよびデータベースにロードするために、モデルのロード・オプションを設定するプロセスです。

クエリー・ガバナー Essbase Integration Server のパラメータまたは Essbase サーバーの構成設定です。データ・ソースに対して実行されるクエリーの時間とサイズを制御します。

クラスタ 単一リソースとして動作して、タスクの負荷を共有し、フェイルオーバーのサポートを提供する一連のサーバーまたはデータベースです。システムにおける単一障害点となるサーバーやデータベースを排除します。

クラスタ・サービス システムとしてクラスタ・メンバーの操作を管理するソフトウェアです。クラスタ・サービスを使用すると、一連のリソースやサービスを定義して、クラスタ・メンバー間でのハートビート・メカニズムを監視し、これらのリソースやサービスをできるだけ効率よくかつ透過的に別のクラスタ・メンバーに移動できます。

クラスタ内部接続 ノード障害を検出するためにハートビート情報についてハードウェア・クラスタで使用されるプライベート・リンクです。

クラスタ棒グラフ カテゴリを横に並べたグラフです。垂直棒グラフでのみ使用されます。

クリーン・ブロック 計算スクリプトによってすべての次元が一度に計算された場合、または計算スクリプトで SET CLEARUPDATESTATUS コマンドが使用された場合の、データベース全体の計算が完了しているデータ・ブロックを指します。

クロス集計レポート テーブル・フォーマットでデータの分類および集計を行うレポートです。テーブルのセルには、交差する分類に合致するデータの集計結果が保管されています。たとえば、製品販売情報のクロス集計レポートに、列見出しとして Small や Large などのサイズ属性、行見出しとして Blue や Yellow などの色属性を表示できます。テーブルの中で Large と Blue が交差するセルには、サイズが Large のすべての Blue 製品の総売上げが表示されます。

グリッドの POV 行、列、またはページの交差に次元を配置せずに、グリッド上で次元メンバーを指定する手段です。レポート設計者はグリッド・レベルで POV の値を設定し、ユーザーの POV がグリッドに影響を与えないように防ぐことができます。次元に含まれるグリッドの値が 1 つのとき、その次元は行、列、またはページではなくグリッドの POV に配置します。

グループ 複数のユーザーに同様のアクセス権を割り当てるためのコンテナです。

グローバル・レポート・コマンド 別のグローバル・コマンドに置き換えられるか、またはファイルが終了するまで実行し続けるレポート・スクリプトのコマンドです。

コミット・アクセス Essbase の取引の処理方法に影響する Essbase カーネルの分離レベルです。コミット・アクセスでは、同時取引は書込みロックを長期間保持し、予測可能な結果を生成します。

コンテキスト変数 タスクフロー・インスタンスのコンテキストを特定するために、特定のタスクフローに定義される変数です。

コンテンツ リポジトリに格納されたあらゆるタイプのファイルの情報です。

コンテンツ・ブラウザ コンテンツを参照して選択し、Workspace ページに配置するために使用できるコンポーネントです。

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

コード・ページ 一連のテキスト文字へのビット組合せのマッピングです。コード・ページは、それぞれ異なる文字セットをサポートします。各コンピュータには、ユーザーが必要とする言語の文字セットについてのコード・ページ設定が含まれます。このドキュメントでは、コード・ページは非 Unicode のエンコードのビット組合せに文字をマッピングします。「エンコード方式」も参照してください。

サイクル・スルー データベースを計算しながら、そのデータベース内で複数パスを実行します。

サイレント応答ファイル インストール管理者が提供する必要があるデータをかわりに提供するファイルです。応答ファイルによって、ユーザーが操作または入力しなくても EPM System インストーラや EPM System コンフィグレータは実行されます。

サブスクライブ 項目またはフォルダが更新されるときに自動的に通知を受け取るように、項目またはフォルダにフラグを付けることを指します。

サブ勘定科目のナンバリング 不連続の整数を使用してサブ勘定科目のナンバリングを行うためのシステムです。

サポート詳細 セルの値を算出した計算および仮定です。

サンプリング エンティティの特性を判別するためにエンティティの代表的な部分を選択するプロセスです。「メタデータのサンプリング」も参照してください。

サービス ビジネス項目を取得、変更、追加、および削除するためのリソースです(権限付与、認証など)。

サーブレット Web サーバーが実行可能なコンパイルされたコードです。

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

シナリオ データを分類するための次元です(Actuals、Budget、Forecast1、Forecast2 など)。

シリアル計算 デフォルトの計算設定です。1 つの計算を複数のタスクに分割して、一度に 1 つのタスクを計算します。

シングル・サインオン(SSO) 一度ログオンすれば、認証を再度求められることなく複数のアプリケーションにアクセスできる機能です。

シングル・ポイント障害 障害が発生した場合にユーザーが通常の機能にアクセスできなくなる、システムのコンポーネントです。

ジョブ 出力生成のために起動できる特殊なプロパティを持つドキュメントです。ジョブには Interactive Reporting、SQR Production Reporting、または汎用ドキュメントを含めることができます。

ジョブの出力 ジョブの実行によって生成されたファイルやレポートです。

スキーマ リレーショナル・データベースにおける、データおよびデータ間の関連を表す論理モデルです。

スクレーピング データ・ソースを検査して、最も基本的なメタデータ要素を得ることを指します。「イントロスペクション」と対比してください。

スケール スケールによって値の表示方法を決定します(整数、十単位、百単位、千単位、百万単位など)。

スコア ターゲットを達成するレベルです。通常はターゲットのパーセンテージとして表されます。

スコアカード 目標を達成する上での、従業員、戦略要素、または責任要素の進行状況を示すビジネス・オブジェクトです。スコアカードに追加された各メジャーおよび子スコアカードについて収集されるデータに基づいて、進行状況が確認されます。

スコープ Essbase の操作または設定により包含されるデータ領域です(セキュリティ設定の影響を受けるデータ領域など)。通常、スコープには 3 レベルの粒度があり、上位レベルが下位レベルを包含します。これらのレベルは上位から下位の順で、システム全体(Essbase サーバー)、Essbase サーバー上のアプリケーション、Essbase サーバー・アプリケーション内のデータベースとなります。「持続性」も参照してください。

ステージ 1)通常は個別のユーザーにより実行される、タスクフロー内の 1 つの論理ステップを形成するタスクの説明です。ステージには手動と自動の 2 つのタイプがあります。2)Profitability では、組織内での割当てプロセスのステップを表す、モデル内の論理区分です。

ステージング・テーブル 特定のアプリケーションの必要性に対応するために作成するデータベースです。ステージング領域は、1 つ以上の RDBMS のスナップショット(再構築されたバージョン)です。

ステージング領域 特定のアプリケーションの必要性に対応するために作成するデータベースです。ステージング領域は、1 つ以上の RDBMS のスナップショット(再構築されたバージョン)です。

ステージ・アクション 自動ステージで、ステージを実行するために呼び出されたアクションです。

ステージ内割当て 同じステージ内のオブジェクトに対する財務フローの割当てです。

ステージ後割当て 割当てモデルにおける割当てです。後に続くモデル・ステージの場所に割り当てられます。

ステータス・バー コマンド、勘定科目、およびデータ・ファイルの現在のステータスに関する有用な情報が表示される画面下部のバーです。

スナップショット 特定の時点の読取り専用データです。

スポットライタ 選択された条件に基づくカラー・コーディングを可能にするツールです。

スマート・カット URL フォームのリポジトリ項目へのリンクです。

スマート・タグ Microsoft Office アプリケーションでのキーワードです。スマート・タグのメニューから使用可能な定義済アクションに関連付けられています。Oracle EPM System 製品でも、スマート・タグを使用して Reporting and Analysis のコンテンツのインポートや Financial Management および Essbase の機能へのアクセスが可能です。

スーパーバイザ サーバーのすべてのアプリケーション、データベース、関連ファイル、セキュリティ機構にフル・アクセスできるユーザーです。

ズームチャート チャートを拡大することにより詳細情報を表示するためツールです。ズームチャートは、チャートに表示されるメトリックについて数値情報を詳細に表示できます。

セカンダリ・メジャー プライマリ・メジャーよりも優先度の低いメジャーです。セカンダリ・メジャーには業績レポートがありませんが、スコアカードで使用したり、次元メジャーのテンプレートを作成するために使用できます。

セキュリティ・エージェント Web アクセス管理プロバイダ(Oracle Access Manager、Oracle Single Sign-On または CA SiteMinder など)です。企業の Web リソースを保護します。

セキュリティ・プラットフォーム Oracle EPM System 製品で外部認証とシングル・サインオン機能を使用するためのフレームワークです。

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

セル・ノート Essbase データベースでセルに付けられるテキスト注釈です。セル・ノートは LRO の一種です。

タイトル・バー Strategic Finance 名、ファイル名、およびシナリオ名のバージョン・ボックスを表示するバーです。

タイムライン・ビューア 特定の場所について、完了したプロセス・フロー・ステップの日時を表示するための FDM の機能です。

タイム・イベント ジョブの実行をトリガーします。

タイム・スケール 指定された期間別(毎月、四半期ごとなど)にメトリックを表示するスケールです。

タスクフロー ビジネス・プロセスの自動化を指します。手続きのルールに従って、あるタスクフロー参加者から別の参加者にタスクが渡されます。

タスクフロー・インスタンス タスクフローの状態と関連データが含まれる、タスクフローの単一のインスタンスです。

タスクフロー参加者 手動ステージおよび自動ステージの両方について、タスクフローのステージのインスタンスに関連付けられているタスクを実行するリソースです。

タスクフロー定義 ステージとステージ間の関係のネットワーク、タスクフローの開始と終了を示す基準、および個別のステージに関する情報(参加者、関連アプリケーション、関連アクティビティなど)から構成される、タスクフロー管理システムのビジネス・プロセスです。

タスクフロー管理システム タスクフローを定義および作成し、その実行を管理するシステムです。定義付け、ユーザーまたはアプリケーションのやりとり、およびアプリケーションの実行可能ファイルが含まれます。

タスク・リスト 特定のユーザーについて、タスクの詳細ステータスを示すリストです。

タブ Strategic Finance で、勘定科目とレポートのナビゲーションを行うことができるビューです。

タプル MDX 構文の要素です。セルは、各次元からのメンバーの交差として参照されます。次元が削除されている場合、最上位のメンバーが示されます。次に例を示します: (Jan); (Jan, Sales); ([Jan], [Sales], [Cola], [Texas], [Actual])。

ダッシュボード 業務の要約を対話的に示すメトリックと指標の集まりです。ダッシュボードにより分析アプリケーションを構築して配置できます。

チャート・テンプレート ワークスペース・チャートに表示するメトリックを定義するテンプレートです。

テキスト・メジャー Essbase では、メジャーが示されている次元で Text としてタグ付けされたメンバーを指します。セルの値は定義済テキストとして表示されます。たとえば、Satisfaction Index というテキスト・メジャーについては、Low、Medium、および High という値を含めることがあります。「型付きメジャー」、「テキスト・リスト」および「導出テキスト・メジャー」も参照してください。

テキスト・リスト Essbase で、数値の識別子にマッピングされたテキストの値を保管するオブジェクトです。テキスト・リストではテキスト・メジャーを使用できます。

デフォルト通貨単位 データの単位スケールです。たとえば、千単位で分析を定義するように選択して 10 を入力すると、10,000 と解釈されます。

データベース接続 データ・ソースへの接続に使用する定義とプロパティを保管し、データベース参照を移動可能にして広く使用できるようにするファイルです。

データ・キャッシュ 非圧縮データ・ブロックを保持するメモリー内のバッファです。

データ・セル「セル」を参照してください。

データ・ファイル・キャッシュ 圧縮データ(PAG)ファイルを保持する、メモリー内のバッファです。

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

データ・モデル データベース・テーブルのサブセットを示します。

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

データ・ロードのルール テキストベース・ファイル、スプレッドシート、またはリレーショナル・データ・セットからのデータをデータベースにロードする方法を決定する一連の基準です。

データ・ロード位置 FDM で、ソース・データをターゲット・システムに送信する報告単位です。通常、ターゲット・システムにロードされる各ソース・ファイルに対して FDM のデータ・ロード位置が 1 つあります。

データ値「セル」を参照してください。

データ関数 データを集約する関数です。データの平均、最大値、カウントを求めたり、他の統計値によりデータのグループを集約します。

トップ・ラベルとサイド・ラベル 列と行の見出しです。ピボット・レポートの上部とサイドにそれぞれ表示されます。

トラフィック・ライト 2 つの次元メンバーの比較、または一定の制限値に基づいて、レポートのセルまたはピンを色分けする機能です。

トリガー ユーザーが指定した基準に従ってデータを監視するための、Essbase の機能の 1 つです。基準に一致すると、Essbase はユーザーまたはシステム管理者にアラートを送信します。

トレーサビリティ メタデータ要素を物理ソースまで追跡する機能です。たとえば Essbase Studio の場合、キューブ・スキーマをそれ自体の階層およびメジャー階層から次元要素、日付/時間要素、メジャー、そして最終的には物理ソース要素まで追跡できます。「系列」も参照してください。

トレース・レベル ログ・ファイルに取り込まれた詳細のレベルです。

トレース割当て 財務データのフローに対する視覚的な追跡を可能にする Profitability の機能です。この追跡は単一の交差から、モデル内の前方または後方に実行できます。

トークン 外部認証システム上の 1 つの有効なユーザーまたはグループの暗号化された ID です。

トースト・メッセージ 画面右下の隅で消えていくメッセージです。

ドライバ Profitability and Cost Management で、ドライバを使用する複数ソース間の数学的関係、およびこれらのソースがコストや収益を割り当てる宛先を示す、割当て方法の一種です。Business Modeling については、「コスト・ドライバ」および「アクティビティ・ドライバ」も参照してください。

ドリルスルー あるデータ・ソースの値から別のソースの対応するデータに移動することです。

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

ネイティブ認証 サーバーまたはアプリケーション内で、ユーザー名とパスワードを認証するプロセスです。

ネスト列見出し 複数の次元からのデータを表示するレポート列の列見出しのフォーマットです。たとえば、Year と Scenario のメンバーが含まれる列見出しはネスト列です。ネスト列見出しでは、見出しの一番上の行の Q1(Year 次元)が、見出しの一番下の行の Actual および Budget(Scenario 次元)で修飾されます。

ハイパーテキスト・マークアップ言語(HTML) Web ブラウザでのデータ表示を指定するプログラミング言語です。

ハイパーリンク ファイル、Web ページまたはイントラネット HTML ページへのリンクです。

ハイブリッド分析 リレーショナル・データベースに格納された下位のデータを、Essbase に格納された要約レベルのデータにマッピングする分析です。リレーショナル・システムの大規模スケーラビリティと多次元データを組み合せます。

ハードウェア・クラスタ ネットワーク・サービス(たとえば IP アドレス)やアプリケーション・サービス(データベースや Web サーバーなど)のシングル・ビューを、これらのサービスのクライアントに提供するコンピュータの集合です。ハードウェア・クラスタの各ノードは、独自のプロセスを実行するスタンドアロン・サーバーです。これらのプロセスは互いに通信して、連携してアプリケーション、システム・リソースおよびデータをユーザーに提供する 1 つのシステムのようなものを形成します。

バックアップ アプリケーション・インスタンスの複製コピーです。

バッチ POV ユーザーの POV において、バッチに含まれる各レポートおよびブックのすべての次元の集合です。バッチのスケジュールを立てる際は、バッチ POV で選択されたメンバーを設定できます。

バッチ・ファイル 複数の ESSCMD スクリプトを呼び出して複数の ESSCMD セッションを実行できるオペレーティング・システム・ファイルです。Windows システムの場合、バッチ・ファイルには BAT というファイル拡張子が付けられます。UNIX の場合、バッチ・ファイルはシェル・スクリプトとして記述されます。

バッチ・ローダー 複数ファイルの処理を可能にする FDM コンポーネントです。

バッチ処理モード サーバー管理や診断のルーチン・タスクを自動化するために使用できるバッチやスクリプト・ファイルを記述するために、ESSCMD を使用する方法です。ESSCMD スクリプト・ファイルは複数のコマンドを実行でき、オペレーティング・システムのコマンドラインから実行したり、オペレーティング・システムのバッチ・ファイルから実行したりすることが可能です。バッチ・ファイルを使用すると、複数の ESSCMD スクリプトを呼び出したり、ESSCMD の複数インスタンスを実行したりできます。

バッチ計算 データベースにおいてバッチで実行される計算です(計算スクリプト、すべてのデータベース計算など)。動的計算はバッチ計算とはみなされません。

バージョン データのシナリオのコンテキスト内で使用される、起こりうる結果です。たとえば、Budget - Best Case と Budget - Worst Case では、Budget がシナリオであり、Best Case と Worst Case がバージョンです。

パターン照合 条件として入力されるアイテムの一部またはすべての文字と値を照合する機能です。欠落文字は、疑問符(?)またはアスタリスク(*)などのワイルド・カード値で表せます。たとえば、「Find all instances of apple」では apple が戻されるのに対して、「Find all instances of apple*」では apple、applesauce、applecranberry などが戻されます。

パフォーマンス・インディケータ ユーザーが指定した範囲に基づくメジャーおよびスコアカード・パフォーマンスを示すために使用されるイメージ・ファイルです。ステータス記号とも呼ばれます。デフォルトのパフォーマンス・インディケータを使用することもできますが、無制限に独自のパフォーマンス・インディケータを作成することも可能です。

パブリック・ジョブ・パラメータ 管理者が作成する再利用可能な名前付きジョブ・パラメータです。必要なアクセス権を持つユーザーがアクセスできます。

パブリック反復タイム・イベント 管理者が作成する再利用可能なタイム・イベントです。アクセス・コントロール・システムからアクセスできます。

パレット JASC に準拠し、.PAL 拡張子を持つファイルです。各パレットには相互に補完し合う 16 色が含まれ、ダッシュボードの色要素の設定に使用することが可能です。

パースペクティブ スコアボードのメジャーやアプリケーションにおける戦略目標をグループ化するために使用されるカテゴリです。パースペクティブにより、主要な利害関係者(顧客、従業員、株主、金融関係者など)またはキー・コンピテンシ領域(時間、コスト、品質など)を示すことができます。

パーティション化 データ・モデルの間で共有またはリンクされるデータの領域を定義するプロセスです。パーティション化は Essbase アプリケーションのパフォーマンスとスケーラビリティに影響することがあります。

パーティション領域 データベース内のサブ・キューブです。パーティションは、データベースの一部からの 1 つ以上のセル領域から構成されます。複製パーティションおよび透過パーティションの場合、2 つのパーティションが同じ形状となるために、領域内のセルの数がデータ・ソースとターゲットで同一となる必要があります。データ・ソース領域に 18 個のセルが含まれる場合、データ・ターゲット領域にも対応する 18 個のセルが含まれている必要があります。

ビジネス・プロセス 集合的にビジネス上の目標を達成するための一連のアクティビティです。

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

ビジュアル・キュー 特定のタイプのデータ値をハイライトする、フォントや色などのフォーマットが設定されたスタイルです。データ値は、次元メンバー、親メンバー、子メンバー、共有メンバー、動的計算、式を含むメンバー、読取り専用データ・セル、読取りおよび書込みデータ・セル、またはリンク・オブジェクトのいずれかになります。

ビュー 年次累計または期別のデータ表示です。

ピボット 取得したデータのパースペクティブを変更します。Essbase では、まず次元が取得され、データが行に展開されます。その後、データのピボット(並替え)を行うことにより、異なる視点を得ることができます。

ピン ピンボードと呼ばれるグラフィック・レポート上に配置される対話型アイコンです。ピンは動的です。ピンは、基盤となるデータ値や分析ツールの基準に基づいて、イメージやトラフィック・ライトの色を変更できます。

ピンボード 3 種類のデータ・オブジェクトの表示タイプの 1 つです。ピンボードは、背景およびピンと呼ばれる対話型アイコンから構成されるグラフィックです。ピンボードを使用するにはトラフィック・ライトを定義する必要があります。

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

ファクト・テーブル スター結合スキーマの中心のテーブルです。外部キー、および次元テーブルから取得した要素により特徴付けられます。通常、このテーブルにはスキーマの他のすべてのテーブルに関連する数値データが含まれます。

フィルタ データ・セットで、特定の基準に従って値を制限する制約です。たとえば、特定のテーブル、メタデータ、または値を除外したり、アクセスを制御したりする場合に使用されます。

フェイルオーバー プライマリ・データベース、サーバーまたはネットワークに障害が発生したり、これらがシャットダウンしたりする場合に、冗長性のあるスタンバイ・データベース、サーバーまたはネットワークに自動的に切り替える機能です。フェイルオーバー用にクラスタリングされているシステムは、高可用性、サーバーの冗長性を利用したフォルト・トレランス、および共有ディスクなどのフォルト・トレラント・ハードウェアを提供します。

フォーマット文字列 1) Essbase では、セル値の表示方法を変換する方法です。2) Data Relationship Management では、Format または Formatted Date 派生プロパティのパラメータで、返すプロパティ値のフォーマットを指定します。

フッター レポート・ページ下部に表示されるテキストまたはイメージです。ページ番号、日付、ロゴ、タイトル、ファイル名、作成者名など、動的な関数や静的なテキストが含まれます。

フリーフォーム・グリッド 動的計算のために、複数のソースからのデータを提示、入力、および統合するためのオブジェクトです。

フリーフォーム・レポート作成 ワークシートに次元メンバーまたはレポート・スクリプト・コマンドを入力することにより、レポートを作成することを指します。

フレーム デスクトップ上の領域です。ナビゲーション・フレームとワークスペース・フレームが 2 つの主要な領域となります。

フロー勘定科目 期別と年次累計の符号なしの値を格納する勘定科目です。

ブック 1) Financial Reporting では、類似したドキュメントのグループを保持するコンテナです。ブックは、次元セクションまたは次元の変更を指定する場合があります。2) Data Relationship Management では、グループとして同時に実行できるエクスポートの集合です。エクスポート結果は、結合することも、個別に出力することもできます。

ブックの POV ブックが実行される次元のメンバーです。

ブックマーク ユーザー個人のページに表示されるレポート・ドキュメントまたは Web サイトへのリンクです。ブックマークのタイプには、マイ・ブックマークとイメージ・ブックマークがあります。

ブロック プライマリ・ストレージ・ユニットです。多次元配列であり、すべての密次元のセルを表します。

ブロックされた勘定科目 手動で入力する必要があるために連結ファイルの計算に含めない勘定科目です。

ブロック・ストレージ・データベース 疎次元に定義されたデータ値の密度に基づいてデータを分類および格納する、Essbase のデータベース・ストレージ・モデルです。データ値はブロック単位で格納され、ブロックは値を含む疎次元メンバーについてのみ存在します。

ブロードキャスト・メッセージ Planning アプリケーションにログオンしているユーザーに対して管理者が送信する単純なテキスト・メッセージです。メッセージには、システムの可用性、アプリケーション・リフレッシュの通知、アプリケーションのバックアップなどの情報が詳細に表示されます。

プライマリ・メジャー 企業および事業のニーズにとって重要な、優先度の高いメジャーです。コンテンツ・フレームに表示されます。

プランナ データの入力と送信、他のプランナが作成したレポートの使用、ビジネス・ルールの実行、タスク・リストの使用、電子メール通知の使用、および Smart View の使用が可能なユーザーです。プランナは、大多数のユーザーから構成されます。

プランニング・ユニット シナリオ、バージョン、およびエンティティの交差におけるデータ・スライスです。プラン・データの準備、確認、注釈付け、および承認のための基本単位です。

プレゼンテーション Web Analysis ドキュメントのプレイリストです。レポートの分類、整理、並べ替え、配布、および確認を行うことができます。リポジトリ内のレポートを参照するポインタを含みます。

プロキシ・サーバー セキュリティを保証するために、ワークステーション・ユーザーとインターネットの間で仲介を行うサーバーです。

プロジェクト 実装でグループ化された Oracle Hyperion 製品のインスタンスです。たとえば、Planning プロジェクトには Planning アプリケーション、Essbase キューブ、Financial Reporting サーバー・インスタンスが含まれることがあります。

プロセス監視レポート FDM データ変換プロセスにおける、場所とその位置のリストです。プロセス監視レポートを使用して、決算手続のステータスを監視できます。レポートにはタイム・スタンプが付けられるので、時間データがロードされた場所を判断するために使用できます。

プロット領域 X 軸、Y 軸、および Z 軸で囲まれている領域です。円グラフの場合は、その周りに表示される長方形の領域です。

プロビジョニング ユーザーおよびグループに対して、リソースへのアクセス権限を付与するプロセスです。

ページ グリッドまたはテーブルでの情報表示の一種です。しばしば Z 軸により示されます。ページには、1 つのフィールドからのデータ、計算により得られるデータ、またはテキストを含めることができます。

ページ・ファイル Essbase のデータ・ファイルです。

ページ・メンバー ページ軸を決定するメンバーです。

ページ見出し レポートの現在のページで表示されているメンバーをリストした、レポート見出しの一種です。ページ上のすべてのデータ値には、ページ見出し内のメンバーが共通属性として適用されています。

ホスト アプリケーションとサービスがインストールされているサーバーです。

ホスト・プロパティ ホストに関係するプロパティです。ホストに複数の Oracle EPM ホームが含まれる場合は、いずれかの Oracle EPM ホームに関係するプロパティとなります。

マイ Workspace ページ ユーザーが作成するカスタマイズ可能な Workspace ページです。リポジトリを参照することなく 1 つの場所から簡単にアクセスできるように特別なマークが付いています。

マスター・データ・モデル 複数のクエリーによりソースとして参照される独立するデータ・モデルです。このモデルが使用される場合は、「クエリー」セクションのコンテンツ・ペインに「ロック済データ・モデル」と表示されます。データ・モデルは「データ・モデル」セクションに表示されるマスター・データ・モデルにリンクされています(管理者によって非表示になっていることがあります)。

マップ・ナビゲータ 戦略、責任、および因果関係の各マップに現在の位置を示す機能です。赤色のアウトラインで示されます。

マップ・ファイル 外部データベースとの間でのデータの送信や取得に使用される定義を格納するファイルです。マップ・ファイルの拡張子は、データ送信用が.mps、データ取得用が.mpr です。

マルチロード 複数の期間、カテゴリ、および場所を同時にロードすることを可能にする FDM の機能です。

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

ミドルウェア・ホーム Oracle WebLogic Server ホームを含み、EPM Oracle ホームおよびその他の Oracle ホームも含むことができるディレクトリです。ミドルウェア・ホームは、ローカル・ファイル・システム、または NFS を介してアクセス可能なリモート共有ディスク上に配置できます。

ミニスキーマ データ・ソースからのテーブルのサブセットをグラフィカルに示したものです。データ・モデリングのコンテキストを表します。

ミニレポート レイアウト、コンテンツ、ハイパーリンク、およびレポートのロード用の 1 つまたは複数のクエリーを含みます。各レポートには、1 つ以上のミニレポートを含めることができます。

メジャー OLAP データベースのキューブに含まれる数値で、分析に使用されます。メジャーには、利益幅、売上原価、売上数量、予算などがあります。「ファクト・テーブル」も参照してください。

メタアウトライン Essbase Integration Services における、OLAP モデルから Essbase アウトラインを作成するための構造とルールを含んでいるテンプレートです。

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

メタデータのサンプリング ドリルダウン操作で次元に含まれるメンバーのサンプルを取得するプロセスです。

メタデータ・セキュリティ ユーザーにより特定のアウトライン・メンバーへのアクセスを制限するための、メンバー・レベルのセキュリティ・セットです。

メタデータ要素 データ・ソースから算出されるメタデータ、および Essbase Studio で使用するために格納され、カタログが作成されるおよびその他のメタデータです。

メンバー 次元内の個別のコンポーネントです。メンバーにより、類似する単位の集まりが個別に特定および区別されます。たとえば、時間次元には Jan、Feb、および Qtr1 などのメンバーが含まれることがあります。

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

メンバー・ロード Essbase 統合サービスにおける、次元およびメンバーを(データなしに)Essbase アウトラインに追加するプロセスです。

メンバー専用レポート・コマンド レポート・ライターのフォーマット・コマンドの 1 つです。レポート・スクリプトで現れると実行されます。このコマンドは関連するメンバーにのみ影響し、メンバーを処理する前にフォーマット・コマンドを実行します。

メンバー選択レポート・コマンド 兄弟、世代、レベルなどのアウトラインの関係に基づいて、メンバーの範囲を選択するレポート・ライター・コマンドの一種です。

モデル 1)アプリケーション固有のデータ表現を含むファイルまたはコンテンツの文字列です。モデルは Shared Services により管理される基本データであり、次に示す 2 つの主要なタイプがあります: 次元および非次元のアプリケーション・オブジェクト; 2)ビジネス・モデリングで、検査対象の領域からの業務および財務上のフローを示し、計算するために接続されたマシン・ネットワークです。

ユーザー・ディレクトリ ユーザーおよびグループの情報を集中管理する場所です。リポジトリまたはプロバイダとも呼ばれます。一般的に使用されているユーザー・ディレクトリには、Oracle Internet Directory (OID)、Microsoft Active Directory (MSAD) および Sun Java System Directory Server があります。

ユーザー変数 ユーザーのメンバー選択に基づいてデータ・フォームを動的に配置し、指定されたエンティティのみを表示する変数です。たとえば、Department というユーザー変数を使用すると、特定の部署および従業員を表示できます。

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

ユーザー定義属性(UDA) アウトラインのメンバーに関連付けられ、メンバーの特性を説明する属性です。これを使用すると、指定された UDA が関連付けられているメンバーのリストが戻されます。

ライトバック 取得を行うスプレッドシートなどのクライアントが、データベースの値を更新する機能です。

ライフサイクル管理 製品環境間でアプリケーション、リポジトリまたは個別のアーティファクトを移行するプロセスです。

ライン・アイテムの詳細 勘定科目で最も下位の詳細レベルです。

リソース システムにより管理されるオブジェクトまたはサービスです(役割、ユーザー、グループ、ファイル、ジョブなど)。

リポジトリ ビューおよびクエリーに使用するためのメタデータ、フォーマットおよび注釈の情報を格納します。

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

リンク 1)リポジトリ・オブジェクトへの参照です。リンクは、フォルダ、ファイル、ショートカットおよび他のリンクを参照できます。2)タスクフローで、あるステージのアクティビティが終了して次のアクティビティが開始するポイントです。

リンク・データ・モデル リポジトリのマスター・コピーにリンクされたドキュメントです。

リンク・パーティション データ・セルを使用して 2 つのデータベースをリンクするための共有パーティションです。ワークシートのリンク・セルをクリックすると、リンク・データベースの次元を示す新しいシートが開きます。これにより、表示される次元をドリルダウンできます。

リンク・レポート・オブジェクト(LRO) セル・ノート、URL、またはテキスト、オーディオ、映像、画像を含むファイルなどの外部ファイルへのセルベースのリンクです。Financial Reporting では、Essbase LRO 向けにサポートされるのはセル・ノートのみです。「ローカル・レポート・オブジェクト」と対比してください。

リンク条件 タスクフローのステージを順序付けるためにタスクフロー・エンジンにより評価される論理式です。

レイアウト領域 コンテンツを配置できる Workspace ページの領域です。

レイヤー 1)階層構造内で横並びにメンバーを含む場所です。世代(上から下へ)またはレベル(下から上へ)により指定されます。2)他のオブジェクトに対して相対的なオブジェクトの場所です。たとえば、Sample Basic データベースでは Qtr1 と Qtr4 は同じ年に含まれるので、世代が同一であることになります。しかし、不均衡階層を含むデータベースの場合、Qtr1 と Qtr4 は同一世代であっても同じレイヤーに位置しないことがあります。

レコード データベースで、1 つの完全な入力項目を形成するフィールドのグループです。たとえば、顧客レコードには、名前、住所、電話番号、および販売データのフィールドが含まれることがあります。

レベル 階層ツリー構造において、データベース・メンバーの関係を定義するレイヤーです。レベルは一番下の次元メンバー(レベル 0)から上位の親メンバーへと並べられます。

レベル 0 のブロック 疎のレベル 0 メンバーの組合せに使用されるデータ・ブロックです。

レベル 0 のメンバー 子の存在しないメンバーです。

レポートの設計 コンポーネント・ライブラリを使用してカスタム・レポートを作成するための Web Analysis Studio のインタフェースです。

レポートの通貨 財務諸表を準備するために使用される通貨です。現地通貨からレポートの通貨に変換されます。

レポート・エクストラクタ スクリプトの実行時に、Essbase データベースからのレポート・データを取得する Essbase コンポーネントです。

レポート・オブジェクト レポートの設計において、テキスト・ボックス、グリッド、イメージ、チャートなどの動作や外観を定義するプロパティを持つ基本要素です。

レポート・スクリプト 1 つまたは複数の運用レポートを生成する、Essbase レポート・ライター・コマンドを格納したテキスト・ファイルです。

レポート・ビューア レポート・スクリプトの実行後に完全なレポートを表示する Essbase コンポーネントです。

ログ・アナライザ Essbase ログのフィルタ、検索、および分析を行うための Administration Services の機能です。

ロケーション別名 データ・ソースを特定する記述子です。ロケーション別名により、サーバー、アプリケーション、データベース、ユーザー名、およびパスワードが指定されます。ロケーション別名は、DBA のデータベース・レベルで管理サービス・コンソール、ESSCMD、または API を使用して設定されます。

ロケール コンピュータで使用される言語、通貨および日付のフォーマット、データのソート順、および文字セットのエンコード方式を指定するコンピュータ設定です。Essbase ではエンコード方式のみが使用されます。「エンコード方式」、「ESSLANG」も参照してください。

ロケール・ヘッダー・レコード スクリプトなど、一部の非 Unicode でエンコードされたテキスト・ファイルの先頭で、エンコード・ロケールを特定するテキスト・レコードです。

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

ロック済データ・モデル ユーザーが変更できないデータ・モデルです。

ローカル・レポート・オブジェクト Explorer で Financial Reporting レポート・オブジェクトにリンクされていないレポート・オブジェクトです。「リンク・レポート・オブジェクト」と対比してください。

ローカル結果 データ・モデルのクエリー結果です。ローカルの結合で結果を使用する場合は、結果をデータ・モデルにドラッグして挿入できます。ローカルの結果を要求すると、カタログに表示されます。

ロード・バランサ 要求をクラスタの個々のアプリケーション・サーバーに分散するハードウェアおよびソフトウェアであり、システムへの唯一のエントリ・ポイントです。

ロード・バランシング サーバー・グループ全体で要求を分散することです。これによって、エンド・ユーザーのパフォーマンスが最適化されます。

ロールアップ「連結」を参照してください。

ワイルド・カード 検索文字列で単一の文字(?)または文字グループ(*)を示す文字です。

ワークフロー FDM でデータを最初から最後まで処理するために必要なステップです。ワークフローは、インポート(GL ファイルからのデータ・ロード)、検証(すべてのメンバーが有効なアカウントにマッピングされていることの確認)、エクスポート(マッピングされたメンバーのターゲット・アプリケーションへのロード)、およびチェック(ユーザー定義の検証ルールを使用してデータを処理することにより、データの精度を確認)から構成されます。

ワークブック 多数のワークシートを含むスプレッドシート・ファイル全体です。

一意でないメンバー名「重複メンバー名」を参照してください。

一意のメンバー名 データベース・アウトライン内に一度のみ存在する、共有されないメンバーの名前です。

一意メンバーのアウトライン 重複メンバー名を使用できないデータベース・アウトラインです。

三角換算法 残高をある通貨から別の通貨へ第三の共通通貨を介して変換する方法です。たとえば、デンマーク・クローネから英国ポンドへ残高を変換するには、残高をクローネからユーロへ、ユーロからポンドへ変換できます。

上位レベル・ブロック 少なくとも 1 つの疎メンバーが親レベルのメンバーになっているデータ・ブロックです。

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

並列エクスポート Essbase データを複数のファイルにエクスポートする機能です。並列エクスポートは、1 つのファイルにエクスポートした場合に比べて時間を短縮できます。また、1 つのデータ・ファイルでサイズが大きくなりすぎた場合の、オペレーティング・システムでの操作上の問題を解決できます。

並列データ・ロード Essbase で、複数プロセスのスレッドによりデータ・ロードのステージを同時に実行することを指します。

並列計算 計算オプションの 1 つです。Essbase では計算がタスクに分割され、一部のタスクは同時に計算されます。

事前計算 ユーザーが取得する前にデータベースの計算を実行することです。

交差 多次元データベース内の次元の交差を表すデータの単位。ワークシートのセル。

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

代替階層 共有メンバーの階層です。代替階層はデータベース・アウトラインの既存の階層に基づきますが、次元に代替レベルを持ちます。代替階層により、異なる視点(POV)から同一データを表示できます。

会社間消去「消去」を参照してください。

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

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

使用済ブロック 最後に計算された後に変更されたセルを含むデータ・ブロックです。子ブロックが使用済である場合(つまり更新されている場合)、上位のブロックにも使用済のマークが付けられます。

例外 事前定義済の条件を満たす値です。フォーマット・インディケータを定義したり、例外が生成されたときに登録ユーザーに通知したりできます。

依存エンティティ 組織内の他のエンティティに属するエンティティです。

保存された仮定 ビジネス上の主要な計算を推進するための、Planning でのユーザー定義の仮定です(事業所の床面積 1平方フィート当たりのコストなど)。

保管階層 集約ストレージ・データベースのアウトラインのみで、アウトラインの構造に従ってメンバーが集約される階層を指します。保管階層のメンバーには、式を含められないなどの一定の制限があります。

信頼できるユーザー 認証されたユーザーです。

修飾名 定まったフォーマットのメンバー名です。重複メンバーのアウトラインにおいて、重複メンバー名を区別します([Market].[East].[State].[New York]、[Market].[East].[City].[New York]など)。

個人の反復タイム・イベント 再利用可能なタイム・イベントです。作成したユーザーのみがアクセスできます。

個人用ページ リポジトリ情報を参照するための個人用ウィンドウです。表示する情報、およびレイアウトと色を選択します。

個人用変数 複雑なメンバー選択の特定の選択ステートメントです。

値次元 入力値、換算値および連結の詳細を定義するために使用される次元です。

兄弟 他の子メンバーと同じ世代で、すぐ上に同じ親を持つ子メンバーです。たとえば、メンバー Florida とメンバー New York はメンバー East の子であり、互いの兄弟です。

入力データ 計算されるのではなくソースからロードされるデータです。

共有 Workspace ページ 専用のシステム・フォルダに格納され、組織全体で共有する Workspace ページです。権限を持つユーザーが共有 Workspace ページの「ナビゲート」メニューからアクセスできます。

共有ストレージ フェイルオーバー・クラスタのすべてのノードに対して使用できる必要のあるデータを含むディスク・セットです。共有ディスクとも呼ばれます。

共有ディスク「共有ストレージ」を参照してください。

共有メンバー ストレージ・スペースを別の同名メンバーとの間で共有するメンバーです。Essbase アウトラインに複数回現れるメンバーが重複して計算されることを防ぎます。

再構成 URL ユーザーが Workspace にログオンしているときに、サーブレット構成設定を動的に再ロードするための URL です。

再構築 データベース・インデックス、また場合によってはデータ・ファイルの再生成もしくは再構築を行う操作です。

冗長データ 重複データ・ブロックです。Essbase で更新されたブロックがコミットされるまで、取引の間保持されます。

凡例ボックス 次元のデータ・カテゴリを特定するためのラベルを含むボックスです。

出資比率 エンティティが親によって所有される程度です。

分離レベル データベース操作のロックとコミットの動作(コミット・アクセスまたはアンコミット・アクセス)を決定する Essbase Kernel の設定です。

列 Data Relationship Management における、インポート・ソース、またはクエリー、比較、検証またはエクスポートの結果に関連付けられたデータのフィールドです。

別名テーブル メンバーの代替名を含むテーブルです。

割当て 割当てモデルでのソースと宛先の関連付けです。割り当てられたコストや収益のフローの方向を制御します。

加重 スコアカードのアイテムに割り当てられた値です。スコアカード全体のスコアの計算において、そのアイテムの相対的な重要性を示します。スコアカードのすべてのアイテムの加重を総計すると 100% になります。たとえば、ある製品について新機能を開発する重要性を認識する場合、開発者のスコアカード上の New Features Coded のメジャーに Number of Minor Defect Fixes のメジャーよりも大きな加重が割り当てられます。

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

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

動的レポート レポートを実行するときに更新されるデータを含むレポートです。

動的参照 データ・ソース内のヘッダー・レコードを指すルール・ファイル内のポインタです。

動的時系列 ブロック・ストレージ・データベースで、期間累計のレポート作成を実行するプロセスです。

動的計算 Essbase で、動的計算メンバー、または動的計算および保管メンバーとしてタグ付けされているメンバーについてデータを取得する場合にのみ実行される計算です。メンバーの値は、バッチ計算で事前に計算されるのではなく、取得時に計算されます。

動的計算および保管メンバー ブロック・ストレージ・アウトラインで、値を最初に取得したときにのみ Essbase によって計算されるメンバーです。計算された値はデータベースに保管され、2 回目以降の取得では計算を実行する必要がありません。

動的計算メンバー ブロック・ストレージ・アウトラインで、Essbase により取得時にのみ計算が行われるメンバーです。取得要求の処理が完了すると、計算された値は破棄されます。

動的階層 集約ストレージ・データベースのアウトラインに限定して、取得時に値が計算されるメンバーの階層です。

勘定科目のブロック 連結ファイルで勘定科目が入力データを受け入れるプロセスです。ブロックされた勘定科目は加算連結プロセスで値を受け取りません。

勘定科目の消去 連結時に連結ファイル内で値がゼロに設定された勘定科目です。

勘定科目タイプ 時間の経過に伴う勘定科目の値のフロー、およびその符号の振舞いを決定するプロパティです。勘定科目の種別のオプションには、支出、収益、資産、負債、および資本が含まれます。

単項演算子 アウトラインのメンバーに関連付けられている算術インディケータ(+、-、*、/、%)です。単項演算子では、データベースのロールアップ中のメンバーの計算方法が指定されます。

原点 2 つの軸の交差です。

反復タイム・イベント ジョブの実行開始点と実行頻度を指定するイベントです。

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

収益勘定科目 期別の値と年次累計値を格納する勘定科目です。値が正の場合は純利益が増えます。

同期 Shared Services とアプリケーション・モデルの同期です。

同期済 モデルの最新バージョンがアプリケーションと Shared Services の両方に存在する状態を指します。「モデル」も参照してください。

名前付きセット MaxL DML で、MaxL DML クエリーのオプションの WITH セクションに定義された論理を使用するセットです。名前付きセットはクエリー内で複数回参照することが可能です。

因果関係マップ 企業戦略を形成する要素の関連、およびこれらの要素が組織の戦略目標を達成するためにどのように連動するかを示すマップ。因果関係マップのタブは、各戦略マップについて自動的に作成されます。

垂直アプリケーション・クラスタ 複数のアプリケーション・サーバー・インスタンスが同じマシン上にあるクラスタです。

型付きメジャー Essbase で、メジャーが示されている次元で Text または Date というタグが付けられたメンバーです。セルの値は事前定義されたテキストまたは日付として表示されます。

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

基本次元 1 つまたは複数の属性次元に関連付けられている標準次元です。たとえば、製品に香りがあるとすると、Product 次元が Flavors 属性次元の基本次元となります。

基本通貨 日常の業務取引が行われる通貨です。

変換 アプリケーションの移行後に移行先環境で正しく機能するように、アーティファクトを変換するプロセス。

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

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

外部でトリガーされるイベント ジョブの実行をスケジュールするための、時間ベースでないイベントです。

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

多次元データベース 3 つ以上の次元でデータを整理、保管、および参照する方法です。次元のセットが交差するポイントが個別の値となります。「リレーショナル・データベース」と対比してください。

子 データベース・アウトライン内で親を持つメンバーです。

子孫 データベース・アウトラインで親の下に位置するメンバーです。たとえば、年、四半期および月を含む次元では、メンバー第 2 四半期およびメンバー 4 月がメンバー年の子孫となります。

安全率 より安全な投資から期待される利回りです (米国の長期国債など)。

定義済ドリル・パス データ・モデルでの定義に従って次の詳細レベルにドリルするために使用されるパスです。

宛先 1) Business Rules では、計算済の値が格納されるデータベースのブロックです。2) Profitability and Cost Management では、割当てモデルでのソースと宛先の関連付けです。割り当てられたコストや収益のフローの方向を制御します。

実績の頻度 日付のセットを作成して結果を収集および表示するために使用されるアルゴリズムです。

実行時プロンプト ビジネス・ルールが実行される前にユーザーが入力または選択する変数です。

密次元 ブロック・ストレージ・データベースでは、次元メンバーのすべての組合せについてデータを含んでいる可能性があります。たとえば、時間次元はしばしば密ですが、これは時間次元がすべてのメンバーのあらゆる組合せを含んでいる可能性があるからです。「疎次元」と対比してください。

対称トポロジ 運用サイトとスタンバイ・サイトの層全体で同一である Oracle Fusion Middleware 障害回復構成です。対称トポロジでは、運用サイトとスタンバイ・サイトにあるホスト、ロード・バランサ、インスタンス、アプリケーションの数は同じです。両方のサイトで同じポートが使用されます。システムは同一に構成され、アプリケーションは同じデータにアクセスします。

対称型マルチプロセッシング(SMP) マルチプロセッシングとマルチスレッディングを使用可能にするサーバーのアーキテクチャです。多数のユーザーが単一のインスタンスに同時に接続した場合でも、パフォーマンスが大きく低下することはありません。

導出テキスト・メジャー Essbase Studio で、範囲として表現された定義済ルールによって管理される値を持つテキスト・メジャーです。たとえば、販売高メジャーに基づく導出テキスト・メジャー「販売実績インデックス」は、「高」、「中」、および「低」の値で構成できます。この導出テキスト・メジャーは、個々の販売高の値を「高」、「中」、「低」の範囲に分類して表示するように定義されます。「テキスト・メジャー」も参照してください。

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

属性の関連付け データベース・アウトラインでの関係です。これにより、属性次元のメンバーが基本次元のメンバーの特性を表します。たとえば、製品 100-10 がグレープ味である場合、製品 100-10 は Flavor 属性の関連付けがグレープになります。したがって、Product 次元のメンバー 100-10 は Flavor 属性次元のメンバー Grape に関連付けられることになります。

属性タイプ 様々な関数(データのグループ化、選択、または計算)を使用可能にするためのテキスト、数値、ブール値、日付、またはリンク属性タイプです。たとえば、Ounces 属性次元は数値タイプを持つので、各製品の属性として指定されるオンス数を使用して当該製品のオンス当たりの収益を計算できます。

属性レポート 基本次元メンバーの属性に基づくレポート作成プロセスです。「基本次元」も参照してください。

属性次元 次元の一種です。次元のメンバーの属性や特質に基づいて分析できます。

属性計算次元 メンバーのグループに対して、合計、カウント、平均、最小、および最大を計算するシステム定義の次元です。この次元は動的に計算され、データベース・アウトラインでは表示されません。たとえば、メンバー Avg を使用すると、製品 Red についてニューヨークでの 1 月の平均売上の値を計算できます。

履歴平均 多数の履歴期間にわたる勘定科目の平均です。

差異 プラン値と実績値などの 2 つの値の差です。

市場リスク割増額 国債よりもリスクの高い投資を投資家に呼びかけるための、安全率に追加して支払われる利回りです。予測される市場利回りから安全率を差し引いて計算されます。この数字が示すモデルは将来の市況に近いものとなる必要があります。

式 Data Relationship Management で、プロパティ値を動的に計算するために派生プロパティによって使用されるビジネス・ロジックです。

式の保存 データ取得中にワークシート内に保持される、ユーザーが作成した式です。

役割 リソースへのアクセス権をユーザーおよびグループに付与する際に使用される手段です。

従業員 特定のビジネス・オブジェクトに対して責任を負う(または関与する)ユーザーです。従業員は組織に勤めている必要はありません(コンサルタントなど)。従業員は、認可のためにユーザー・アカウントに関連付けられている必要があります。

復元 データベースが破損または破壊された場合にデータおよび構造の情報を再ロードする操作です。通常、データベースをシャット・ダウンおよび再起動した後で実行されます。

感嘆符(!) 一連のレポート・コマンドを終了して、データベースからの情報を要求する文字です。レポート・スクリプトは感嘆符を使用して終了する必要があります。レポート・スクリプト内では複数の感嘆符を使用できます。

戦略マップ 上位レベルのミッションおよびビジョンのステートメントを、構成要素である下位レベルの戦略的達成目標に組入れる方法を示します。

戦略目標(SO) 測定可能な結果によって定義された長期目標です。各戦略目標は、アプリケーション内の 1 つのパースペクティブに関連付けられ、1 つの親(エンティティ)を持ち、重要成功要因または他の戦略目的の親になります。

手動ステージ ユーザーの操作が必要なステージです。

抽出コマンド Essbase レポート作成コマンドの 1 つです。データベースから抽出される RAW データの選択、向き、グループ分け、および配列を処理します。小なり記号(<)から始まるコマンドです。

拡張リレーショナル・アクセス リレーショナル・データベースと Essbase 多次元データベースの統合を指します。これにより、すべてのデータがリレーショナル・データベースに保持され、Essbase データベース内の要約レベルのデータにマッピングされます。

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

持続性 Essbase の操作や設定に対する継続的または長期的な影響です。たとえば、ユーザー名やパスワードの有効性について、Essbase 管理者がその持続性を制限することがあります。

接続ファイル「Interactive Reporting 接続ファイル (.oce)」を参照してください。

換算「通貨換算」を参照してください。

換算レート「為替レート」を参照してください。

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

支配比率 所属するグループのコンテキスト内でエンティティが受ける支配の程度です。

数値属性範囲 基本次元メンバーを関連付けるために使用される機能です。メンバーは個別の数値を含み、値の範囲を示す属性を持ちます。たとえば、顧客を年齢別に分類する場合、Age Group 属性次元に 0-20、21-40、41-60、および 61-80 という年齢範囲に該当するメンバーを含めることができます。各 Customer 次元メンバーは Age Group 範囲に関連付けられます。データを取得する際は、個別の年齢の値ではなく年齢範囲に基づいて処理されます。

日付メジャー Essbase で、メジャーが示されている次元で「日付」のタグが付けられているメンバーです。セルの値はフォーマット済の日付として表示されます。メジャーとしての日付は時間次元を使用して示すことが困難なタイプの分析に役立つことがあります。たとえば、一連の固定資産の取得日をアプリケーションで追跡する必要がある場合、取得日の範囲が実現可能な時間次元モデリングの範囲を超えて長期にわたってしまうことがあります。「型付きメジャー」も参照してください。

時系列レポート作成 カレンダの日付(年、四半期、月、週など)に基づくデータのレポート作成プロセスです。

時間次元 データが示す期間です(会計期間、暦時間など)。

暗黙の共有 メンバーが 1 つ以上の子を持ち、連結されている子は 1 つのみである場合、親と子が値を共有します。

最上位メンバー 次元のアウトラインで、階層ツリーの一番上に位置する次元メンバーです。次元メンバー間に階層の関係がない場合は、ソート順で最初のメンバーを指します。階層の関係がある場合、最上位メンバーの名前が次元名と同一となるのが一般的です。

最新 最新の期間として定義されたメンバーからデータ値を抽出するために使用される、スプレッドシートのキー・ワードです。

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

期末 チャートの日付範囲を調整できる期間です。たとえば、月の期末の場合、当月末までの情報がチャートに表示されます。

株式ベータ 株のリスクを指します。その株の収益と市場利益率の差異により測定され、ベータと呼ばれるインデックスで示されます。たとえば、市場利益率が 1%変動するのに伴って株の収益が通常 1.2%変動するのであれば、その株のベータ値は 1.2 です。

検証 アウトラインに対してビジネス・ルール、レポート・スクリプトまたはパーティション定義をチェックして、チェック対象のオブジェクトが有効であることを確認するプロセスです。

検証ルール データの整合性を強化するために FDM で使用されるルールです。たとえば、FDM では、検証ルールによって、FDM からターゲット・アプリケーションにデータがロードされた後に、特定の条件が満たされていることが保証されます。

構築方法 データベース・アウトラインを変更するために使用するメソッドの一種です。データ・ソース・ファイルのデータ・フォーマットに基づいて構築メソッドを選択します。

構造ビュー トピックをコンポーネントのデータ項目の単純なリストとして表示します。

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

標準次元 属性次元以外の次元です。

権限 データまたは他のユーザーとグループを管理するために、ユーザーおよびグループに付与されるアクセス・レベルです。

欠落データ(#MISSING) ラベル付けされた場所のデータが存在しないか、値が含まれていないか、データが入力されていないかまたはロードされていないことを示すマーカーです。たとえば、勘定科目に当期ではなく過去または将来の期間のデータが含まれている場合は、欠落データが存在します。

次元 ビジネス・データを整理して値の抽出や保持のために使用されるデータ・カテゴリです。通常、次元には関連するメンバーをグループ化した階層が含まれます。たとえば、多くの場合、年次元には四半期や月などの期間ごとのメンバーが含まれます。

次元タイプ 定義済の機能を使用可能にする次元のプロパティです。時間のタグが付けられた次元は、定義済のカレンダ機能を持ちます。

次元タブ「ピボット」セクションで、行と列の間でデータのピボットを実行するためのタブです。

次元テーブル 1)特定のビジネス・プロセスに関する多数の属性を含むテーブルです。2)Essbase Integration Services では、Essbase の潜在的な次元を定義する 1 つ以上のリレーショナル・テーブルのための、OLAP モデルのコンテナを指します。

次元性 MaxL DML において、セットで示された次元およびその順序です。たとえば、{ (West, Feb), (East, Mar) }というセットの場合は、含まれている 2 つのタプルはいずれも次元(Region, Year)を反映しているため、同一の次元性であることになります。

次元構築 Essbase アウトラインに次元およびメンバーを追加するプロセスです。

次元構築のルール データ・ロードのルールに似た仕様です。Essbase でアウトラインを変更するために使用されます。変更は外部データ・ソース・ファイルのデータに基づきます。

次元間の無関係性 次元が他の次元と交差しない状況を指します。次元に含まれるデータは、交差しない次元からはアクセスできないため、交差しない次元はその次元とは無関係となります。

残高勘定科目 特定の時点に関連する符号なしの値を格納する勘定科目の種別です。

水平アプリケーション・サーバー・クラスタ 複数のアプリケーション・サーバー・インスタンスがそれぞれ異なるマシン上にあるクラスタです。

汎用ジョブ SQR Production Reporting または Interactive Reporting 以外のジョブを指します。

消去 組織内のエンティティ間での取引をゼロに設定(消去)するプロセスです。

消去済勘定科目 連結ファイルに表示されない勘定科目です。

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

現地通貨 入力通貨タイプです。入力通貨タイプが指定されていない場合は、現地通貨がエンティティの基本通貨に一致します。

疎次元 ブロック・ストレージ・データベースで、他の次元と比較した場合に、すべてのメンバーの組合せのデータを含んでいる可能性の低い次元です。「密次元」と対比してください。たとえば、すべての顧客がすべての製品のデータを持っているわけではありません。

目標 指定された期間(日、四半期など)についてメジャーに期待される結果です。

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

相互割当て 財務フローの割当ての一種です。宛先の 1 つとしてソースが含まれます。

相関サブクエリー 親クエリーの各行で一度評価されるサブクエリーです。サブクエリーのトピック・アイテムを親クエリーのトピックに結合することにより作成されます。

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

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

移行スナップショット アプリケーションの移行のスナップショットです。移行ログに取り込まれます。

移行ログ アプリケーションの移行のすべてのアクションとメッセージを取込むログ・ファイルです。

移行定義ファイル(.mdf) アプリケーションの移行に使用される移行パラメータを含むファイルです。これによりバッチ・スクリプトを処理できます。

移行監査レポート 移行ログから生成されるレポートです。アプリケーションの移行に関する追跡情報を提供します。

税金の初期残高 Strategic Finance では、損失の初期残高、収益の初期残高、および納税の初期残高のエントリは、Strategic Finance の最初の期間に先立つ期間に発生していることを前提とします。

算出ステータス 一部の値または式の計算が変更されたことを示す集計ステータスです。影響を受けるエンティティについて正しい値を取得するには、再集計する必要があります。

算術データ・ロード データベース内の値に対して演算(たとえば各値に 10 を加算するなど)を実行するデータ・ロードです。

算術演算子 式およびアウトラインでのデータの計算方法を定義する記号です。標準的な算術演算子またはブール演算子が使用されます(+、-、*、/、%など)。

管理対象サーバー 内蔵された Java 仮想マシン(Java Virtual Machine: JVM)で実行されるアプリケーション・サーバー・プロセスです。

精度 数値に表示される小数点以下の桁数です。

系列 異なるメタデータ要素間の関係です。メタデータ要素が他のメタデータ要素からどのように導き出されるかを示し、メタデータ要素を物理ソースまでトレースします。Essbase Studio では、この関係を系列ビューアでグラフィカルに表示できます。「トレーサビリティ」も参照してください。

系統データ 割当ての計算後にオプションで生成される追加データです。このデータにより、すべての割当てステップにわたるコストまたは収益のフローについてレポートを作成できます。

組織 各エンティティ、およびその関係を定義するエンティティの階層です。

結合 特定の列または行の共通のコンテンツに基づく 2 つのリレーショナル・データベース・テーブルまたはトピックの間のリンクです。通常、異なるテーブルまたはトピック内の同一または類似するアイテムの間で結合が起きます。たとえば、Customer テーブルと Orders テーブルで Customer ID の値が同一である場合、Customer テーブル内のレコードが Orders 内のレコードに結合します。

統制グループ 証明書および評価の情報を維持および整理するために FDM で使用されるグループです。サーベンス・オクスリ(Sarbanes-Oxley)法の規定に準拠する上で特に役立ちます。

統合 Shared Services を使用して Oracle Hyperion アプリケーションでデータを移動するために実行されるプロセスです。データ統合の定義によりソース・アプリケーションと宛先アプリケーションの間でのデータの移動が指定され、データの動きのグループ化、順序付けおよびスケジュールが決定されます。

繰返し 同じバージョンのデータを修正して移行する予算またはプランニング・サイクルのパスです。

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

耳折れ 折り曲げられたページの角です。チャートのヘッダー領域の右上の隅に表示されます。

自動ステージ ユーザーの操作を必要としないステージです(データ・ロードなど)。

自動逆仕訳 次期に逆仕訳する調整を入力するための仕訳です。

行の抑制 欠落値を含む行を除外し、スプレッドシート・レポートからの文字にアンダースコアを付ける設定です。

表示タイプ リポジトリに保存された 3 種類の Web Analysis フォーマット(スプレッドシート、チャート、ピンボード)のいずれかを指します。

製品 Shared Services における、Planning や Performance Scorecard などのアプリケーション・タイプです。

複製パーティション パーティション・マネージャにより定義されるデータベースの一部。あるサイトで管理されるデータの更新を別のサイトに伝播するために使用されます。ユーザーは、ローカルのデータベースと同じようにデータにアクセスできます。

要約チャート「調査」セクションで、同じ列内で下に表示される詳細チャートをロール・アップするチャートです。各チャート列最上位の要約レベルにメトリックを描画します。

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

親の調整 親に関連して子に転記される仕訳エントリです。

計算スクリプト データベースの集計方法や集約方法を定義する一連のコマンドです。集計プロセスとは別に、割当てや他の計算ルールを指定するコマンドが計算スクリプトに含まれることもあります。

計算済勘定科目 計算式を変更することができない勘定科目です。計算式は、構築するモデルの勘定科目の整合性を保つために固定されています。たとえば、当期純利益、計算済勘定科目の計算式は戦略的財務に組み込まれており、履歴期間または予測期間で変更はできません。

計算結果アイテム データベースやキューブに物理的に格納される列に対して、仮想の列を指します。クエリー実行時にデータベースにより、または Interactive Reporting Studio の「結果」セクションで計算されます。計算結果アイテムは、関数、データ項目、およびダイアログ・ボックスで提供される演算子に基づくデータ計算であり、レポートに含まれたり他のデータの計算に再利用されることがあります。

詳細チャート 要約チャートで、詳細な情報を提供するチャートです。詳細チャートは要約チャートの下にある「調査」セクションに列で表示されます。要約チャートに円グラフが表示される場合、その下の詳細チャートには円の各区分が示されます。

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

認証サービス 単一の認証システムを管理するコア・サービスです。

調整「仕訳」を参照してください。

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

調査「ドリルスルー」を参照してください。

論理 Web アプリケーション Web アプリケーションの内部ホスト名、ポートおよびコンテキストの識別に使用される、別名が付けられた参照です。クラスタ環境または高可用性環境では、分散コンポーネントに対する単一の内部参照を確立する別名です。EPM System では、クラスタ化されていない論理 Web アプリケーションは Web アプリケーションを実行する物理ホストにデフォルトで設定されています。

論理グループ FDM で、ソース・ファイルが FDM にロードされた後に生成される 1 つ以上の論理勘定です。論理勘定はソース・データから導き出される計算済勘定です。

負債勘定科目 一定時点における会社の負債残高を格納する勘定科目タイプです(未払費用、買掛金勘定、長期借入金など)。

責任マップ 組織内の責任チーム(重要事業領域とも呼ばれます)の責任、報告、および依存関係の構造を視覚的、階層的に示します。

貸借一致の仕訳 借方の合計と貸方の合計が等しい仕訳です。

資産勘定科目 勘定科目タイプの 1 つです。会社の資産の値を保管します。

軸 1)測定と分類に使用されるグラフィックを貫通する直線です。2)多次元のデータを整理および関連付けるために使用されるレポートのアスペクトです(フィルタ、ページ、行、列など)。たとえば、Simple Basic でデータ・クエリーを実行する場合、軸では Qtr1、Qtr2、Qtr3、および Qtr4 の値の列を定義できます。Market と Product の階層による合計が行データとして取得されます。

透過パーティション ローカルのデータベースの一部であるかのように、リモート・データベースのデータにアクセスして変更できるようにする共有パーティションです。

透過ログイン ログイン画面を起動せずに認証されたユーザーをログインさせるプロセスです。

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

通貨の上書き 任意の入力期間について選択した入力メソッドを上書きして、デフォルトの通貨/アイテムとして該当期間の値の入力を使用可能にする機能です。入力メソッドを上書きするには、数値の前または後にシャープ(#)を入力します。

通貨パーティション アプリケーションでの定義に従って、基本通貨から現地通貨メンバーを隔離する次元タイプです。通貨タイプ(実績、予算、予測など)を特定します。

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

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

連結ファイル(*.cns) 集計のプロセスでチャートまたはツリー・ビューを使用して Strategic Finance ファイルを追加、削除または移動するためのグラフィカル・インタフェースです。集計ファイルを使用して、集計を定義したり変更したりすることも可能です。

連結ファイル(親) 事業部門のすべてのファイルが連結されたファイルです。連結の定義を含みます。

連結比率 親に連結された子の値の割合です。

適応状態 Interactive Reporting Web Client の権限レベルです。

選択リスト レポート設計者がレポートの視点(POV)を定義する際に各次元に指定するメンバーのリストです。定義されたメンバー・リストに指定されたメンバーを選択するか、または動的リストの関数に定義された条件に一致するメンバーを選択するだけで、選択リストを使用する次元の POV を変更できます。

重複する別名 別名テーブルに複数存在し、データベース・アウトラインの複数メンバーに関連付けられている可能性のある名前です。重複する別名は、重複メンバーのアウトラインでのみ使用できます。

重複メンバーのアウトライン 重複メンバー名を格納しているデータベース・アウトラインです。

重複メンバー名 データベース内に複数存在する、それぞれ異なるメンバーを表す同一のメンバー名です。たとえば、ニューヨーク州を示すメンバーとニューヨーク市を示すメンバーが存在する場合、データベースに New York という名前のメンバーが 2 つ含まれることがあります。データベースに New York という名前のメンバーが 2 つ含まれることがあります。

重要事業領域(CBA) 部門、地域、工場、コスト・センター、プロフィット・センター、プロジェクト・チーム、またはプロセスに編成された個人またはグループです。責任チームまたはビジネス領域とも呼ばれます。

重要成功要因(CSF) 戦略目標を達成するために確立および維持する必要のある能力です。戦略目標または重要プロセスにより所有され、1 つ以上のアクションに対する親となります。

関数 Data Relationship Management で、パラメータを受け入れ動的値を返す、派生プロパティ式の構文要素です。

関連勘定科目 メイン勘定科目に関連する、同一のメイン勘定科目番号でグループ化された勘定科目です。勘定科目の構造体では、すべてのメイン勘定科目および関連勘定科目は同一のメイン勘定科目番号にグループ化されます。メイン勘定科目と関連勘定科目は、勘定科目番号の最初の接尾辞により区別されます。

限界税率 税引き後の負債コストを計算するために使用されるレートです。最近計上された所得に適用される税率(所得額に適用される最高の税率区分の税率)を示し、連邦税、州税、および地方税を含みます。課税対象所得と税率区分の現在のレベルに基づいて、限界税率を予測できます。

障害回復 地理的に離れたスタンバイ・サイトへのアプリケーションおよびデータに対する回復戦略を備えることで、運用サイトでの自然災害や予定外の停止から守る機能です。

隣接する四角形 Interactive Reporting ドキュメントのセクションを個人用ページに埋め込む場合に、Interactive Reporting ドキュメントのコンテンツをカプセル化する必須のパラメータです。高さと幅を表すピクセル、または 1 ページ当たりの行数により指定されます。

集約 集約ストレージ・データベースの値をロール・アップおよび格納するプロセスです。または集約プロセスによって格納された結果を指します。

集約スクリプト 集約を構築するための集約ビューの選択を定義するファイルです。集約ストレージ・データベースのみで使用されます。

集約ストレージ・データベース 潜在的に大きな多数の次元に分類される疎に分散した大規模なデータをサポートするように設計されたデータベースのストレージ・モデルです。上位のメンバーと式は動的に計算され、選択されたデータ値は集約、格納されます。通常、集約の合計所要時間が改善されます。

集約セル 複数のセルから構成されるセルです。たとえば、Children(Year)を使用するデータ・セルは、Quarter 1、Quarter 2、Quarter 3、および Quarter 4 のデータを含む 4 つのセルに展開されます。

集約ビュー 各次元内のメンバーのレベルに基づく集約セルの集合です。計算時間を短縮するため、値は事前に集約されて集約ビューとして格納されています。取得は集約ビューの合計から開始され、合計に追加されます。

集約制約 集約要求ライン・アイテムや集約メタトピック・アイテムに設定する制約です。

集約関数 関数の一種です。合計、平均の計算など、データの要約や分析を実行します。

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

非アクティブ・グループ 管理者によりシステムへのアクセスが非アクティブにされているグループです。

非アクティブ・ユーザー 管理者によりアカウントが非アクティブ化されているユーザーです。

非対称トポロジ 運用サイトとスタンバイ・サイトの層全体で異なる Oracle Fusion Middleware 障害回復構成です。たとえば、非対称トポロジでは、運用サイトよりも少ないホストとインスタンスがあるスタンバイ・サイトが含まれることがあります。

非次元モデル Shared Services のモデル・タイプの 1 つです。セキュリティ・ファイル、メンバー・リスト、計算スクリプト、Web フォームなどのアプリケーション・オブジェクトが含まれます。

領域 メンバーおよび値の定義済のセットであり、パーティションを構成します。

高可用性 障害が発生した場合でもアプリケーションが継続してサービスを提供できるようにするシステム属性です。これは、シングル・ポイント障害の除去、フォルト・トレラント・ハードウェアおよびサーバー・クラスタにより実現されます。1 つのサーバーで障害が発生すると、処理要求は別のサーバーにルーティングされます。

高機能計算 最後に実行された計算以降に更新されたデータ・ブロックを追跡する計算方法です。

索引

記号

2 パス計算
メンバー内, 256
概要, 232
次元, 231
次元に選択, 231
2 レベル・データベース(CPL 式), 385
3 レベル・データベース(CPL 式), 385
<data-source-type>.cache.size, 51
<data-source-type>.pool.maxsize, 52
「Essbase プロパティ」ダイアログ・ボックス, 209
「ドリルスルー・レポート・プロパティ」ダイアログ・ボックス
「メイン」タブ, 319
「レポート・コンテンツ」タブ, 321
「関連付け」タブ, 340
「ラベルのみ」オプション, 257

A - Z

abs(SQL 関数), 404
catalog.db, 39
catalog.password, 40
catalog.url, 36
catalog.username, 40
classes(CPL 式), 386
contains(SQL 関数), 398
CPL
SQL 関数
abs, 404
contains, 398
dayOfMonth, 393
exp, 404
In, 404
index, 399
ITrim, 400
leftStr, 399
length, 400
log10, 405
lower, 400
month, 394
monthName, 394
monthShortName, 395
pow, 405
quarter, 395
quarterAsString, 395
rightStr, 401
rTrim, 401
soundex, 402
sqrt, 406
substr, 402
trim, 403
upper, 403
weekday, 396
weekdayName, 396
weekdayShortName, 397
yearShort, 398
年, 397
数値, 404
文字列, 398
日付, 393
式
1 レベルのフラット・ファイル, 385
2 つのレベルを持つデータベース, 385
2 レベル・データベース, 385
2 項、使用, 391
3 つのレベルを持つデータベース, 385
3 レベル・データベース, 385
classes, 386
long 型定数, 387
オペランド, 383
グループ化, 389
データ・ソース接続, 385
フラット・ファイル(1 レベル), 385
ブール定数, 387
ユーザー定義テーブル, 385

判読, 389
単項、使用, 391
定数, 386
定数リテラル, 386
定義および使用, 383
数値定数, 386
整数定数, 387
文字列定数, 387
文字列演算子, 393
構文の要素, 383
演算子, 388
演算子、文字列, 393
演算子、算術, 391
演算子、論理, 389
算術演算子, 391
複数の演算子, 388
要素、構文, 383
計算の順序, 388
計算順序, 388
論理演算子, 389
概要, 383
Date(SQL 関数), 393
dayOfMonth(SQL 関数), 393
Essbase Studio
カタログ・データベース, 29
カタログ・データベースの設定, 29
サーバー。「Essbase Studio サーバー」を参照
サーバーおよびコンソールの停止, 67
サーバーおよびコンソールの開始, 67
データ・ソース・ナビゲータ, 24
ドリルスルー, 21
メタデータ・カタログ, 21
メタデータ・ナビゲータ, 25
ユーザー・インタフェース, 23
ログイン, 26
導入, 21
概要, 21
管理の概要, 29
系列の追跡, 21
Essbase Studio へのログイン, 26
Essbase Studio コンソール
Essbase Studio サーバーへの再接続, 26
停止, 67
開始, 67
Essbase Studio サーバー
Essbase Studio コンソールからの再接続, 26
アクセスの制限, 359
アクセス制限, 359
コマンド
dumps, 70
exit, 72
pconf, 70
squeue, 71
sres, 72
sthd, 71
version, 69
概要, 69
停止, 67
開始, 67
Essbase Studio サーバーの起動ファイル
タイム・ゾーン構成の問題, 58
Essbase Studio サーバーの起動ファイル、構成, 57
Essbase Studio サーバーへの再接続, 26
Essbase、接続の作成, 101
Essbase サーバー
再ホスト後の参照の更新, 81
再ホスト後の参照の更新について, 80
Essbase モデル
2 パス計算, 231
Unicode、使用可能にする, 211
XOLAP、使用可能にする, 211
XOLAP で使用可能にする, 216
アウトライン構築オプション, 240
カスタム SQL、使用, 211
カスタム・データ・ロード設定, 211
キューブ・リンケージ, 308
コンポーネント, 208
スキップ・オプション
メンバー内, 261
次元, 236
タイム・バランス
メンバー内, 260
次元, 234
ダイアログ・ボックスのタブ, 209
ツール・チップ, 284
データ・ロード、最適化, 244
データ・ロードの最適化, 244
ドリルスルー・レポートの関連付け, 340
ブール範囲, 223
プロパティの検証, 282
プロパティの設定, 209
メンバー
プロパティ, 245

別名, 265
ユーザー定義属性, 239
会計次元プロパティ, 234
作業領域, 281
作業領域を使用, 281
作業領域を開く, 281
保存, 211
再使用, 208
再構築、ガイドライン, 368
再構築のガイドライン, 368
列
タグ付け, 208
属性, 208
別名プロパティ、設定, 217
動的時系列, 230
参照, 284
変更、確認, 283
変更の確認, 283
定義済, 208
属性
メンバー名のフォーマット, 221
定義, 218
計算メンバー名フォーマット, 221
差異レポートの作成方法
メンバー内, 262
次元, 235
式
メンバーへの追加, 264
次元への追加, 237
情報, 208
操作, 280
数値範囲, 223
既存のキューブ・スキーマからの作成, 206
日付範囲, 223
概要, 208
次元
タイプ, 229
プロパティ, 224
モデルでの使用, 208
保管方法, 231
別名, 238
特徴, 208
表示, 281
重複するメンバー設定、移動, 242
重複メンバー名、使用可能にする, 210
階層ストレージ設定, 241
Essbase モデルのプリファレンス、設定, 348
Essbase モデルの再構築、ガイドライン, 368
Essbase モデルの再構築のガイドライン, 368
Essbase モデル・プロパティ
ダイアログ・ボックスのタブ, 207
ダイアログ・ボックスへのアクセス, 209
制限, 368
設定, 209
exp(SQL 関数), 404
FDM URL ドリルスルー・レポート・ターゲット, 333, 334
In(SQL 関数), 404
index(SQL 関数), 399
Integration Services カタログの移行
Essbase エクスポート・モデル, 353
Integration Services モデル, 354
ダイアログ・ボックス、アクセス, 353
メジャー列、マッピング, 353
メタアウトライン, 352, 354
メタデータ, 351
モデル, 351
ユーザー定義メンバー、マッピング, 353
レポート、マッピング, 353
会計次元, 353
共有メンバー処理, 356
別名、マッピング, 353
制約, 356
制限, 356
属性処理, 357
手順, 354
概要, 351
次元, 352
移行されないプロパティ, 357
階層、マッピング, 352
ISO カレンダ階層, 193
ITrim(SQL 関数), 400
Java メソッド、ドリルスルー・レポート・タイプ, 337
JDBC ドライバ
MySQL, 66
Netezza, 66
Teradata, 67
構成, 66
leftStr(SQL 関数), 399
length(SQL 関数), 400
log10(SQL 関数), 405
long 型定数(CPL 式), 387
lower(SQL 関数), 400

MaxL スクリプト、キューブ配置オプションに名前を付けて保存, 302
month(SQL 関数), 394
monthName(SQL 関数), 394
monthShortName(SQL 関数), 395
MySQL JDBC ドライバ, 66
MySQL の制限, 362
  ビューを含むデータ・ソースのリフレッシュ, 362
Netezza JDBC ドライバ, 66
OBI EE, 336。「Oracle BI EE」も参照
OBI EE URL ドリルスルー・レポート・ターゲット, 333, 336
OCI の制限, 362
OLAP
  多次元データベースでの使用, 214
  概要, 213
Oracle BI EE
  接続ウィザードでの Essbase モデルの作成, 100
  接続ウィザードでのキューブ・スキーマの作成, 100
  接続ウィザードでの次元の作成, 98
Oracle Diagnostic Logging (ODL), 57
oracle.jdbc.ReadTimeout, 50
Oracle ユーザー
  タイム・ゾーン構成の問題, 58
Performance Management Architect ソース
  メタデータ要素の作成, 103
  接続の作成, 102
  接続パラメータの定義, 102
pow(SQL 関数), 405
quarter(SQL 関数), 395
quarterAsString(SQL 関数), 395
reinit コマンド
  アップグレード, 32
rightStr(SQL 関数), 401
rTim(SQL 関数), 401
server.charset, 45
server.css.URL, 38
server.datafile.dir, 41
server.essbase.blindShare, 50
server.essbase.disableDistinct, 47
server.essbase.disableDistinct、IBM DB2 の制限, 48, 374
server.essbase.streamingCubeBuilding, 42
server.essbase.TPTapi, 47
server.essbase.uniqueMemberFromCaptionBinding, 53
server.hss.bpmApplication, 40
server.httpPort, 48
server.properties
  <data-source-type>.cache.size, 51
  <data-source-type>.pool.maxsize, 52
  catalog.db, 39
  catalog.password, 40
  catalog.url, 36
  catalog.username, 40
  oracle.jdbc.ReadTimeout, 50
  server.charset, 45
  server.css.URL, 38
  server.datafile.dir, 41
  server.essbase.blindShare, 50
  server.essbase.disableDistinct, 47
  server.essbase.streamingCubeBuilding, 42
  server.essbase.TPTapi, 47
  server.essbase.uniqueMemberFromCaptionBinding, 53
  server.hss.bpmApplication, 40
  server.httpPort, 48
  server.queueSize, 43
  server.readLockTimeOut, 46
  server.resourceCount, 44
  server.runInBackground, 49
  server.sql.fetchSize, 44
  server.tempDir, 45
  server.threadCount, 43
  server.timeoutPeriod, 43
  server.writeLockTimeOut, 46
  transport.port, 48
  カタログ・データベース、使用, 30
  パラメータ, 33
  プロパティ・リスト, 35
  例
    IBM DB2, 55
    Oracle, 54
    SQL Server, 56
  保存, 33
  構成, 33
  構成可能なプロパティのリスト, 35
  編集, 33
server.properties ファイル。「server.properties」を参照
server.queueSize, 43

server.readLockTimeOut, 46
server.resourceCount, 44
server.runInBackground, 49
server.sql.fetchSize, 44
server.tempDir, 45
server.threadCount, 43
server.timeoutPeriod, 43
server.writeLockTimeOut, 46
soundex(SQL 関数), 402
SQL 関数。「CPL SQL 関数」を参照
sqrt(SQL 関数), 406
startServer.bat, 57
substr(SQL 関数), 402
Teradata JDBC ドライバ, 67
transport.port, 48
trim(SQL 関数), 403
UMC。「キューブ・リンケージの更新」を参照
Unicode、Essbase モデル, 211
upper(SQL 関数), 403
URL ドリルスルー・レポート・タイプ, 332
URL ドリルスルー・レポート・ターゲット, 333
weekday(SQL 関数), 396
weekdayName(SQL 関数), 396
weekdayShortName(SQL 関数), 397
XML のインポートに関する制限, 361
XOLAP
  Essbase モデル内で使用可能にする, 211
  モデルの制限, 365
  モデルの指定, 216
  ワークフロー, 215
  使用方法のガイドライン, 215
  概要, 214
year(SQL 関数), 397
yearShort(SQL 関数), 398

あ行

アウトライン構築オプション, 240
アップグレード
  Essbase 接続プロパティ、移行後, 31
  reinit コマンド, 32
  カタログの制限, 363
  カタログ・データベース, 31
  キューブ・リンケージ、移行後, 31
  データ・ソース接続プロパティ、移行後, 31
アプリケーション
  命名ルール, 377
イントロスペクション
  制限, 364
  既存のデータ・ソースでの実行, 118
  概要, 147
  選択
    ファクト・テーブル, 148
    ミニスキーマ・オプション, 148
    次元テーブル, 149
    階層, 150
インポート
  カタログ XML ファイル, 76
エクスポート
  カタログ・データベース全体, 74
  別名セット, 176
  選択したカタログ要素, 74
オペランド(CPL 式), 383

か行

カウント・メンバー, 222
カスタマイズ
  ドリルスルー・レポート, 321
  リレーショナル・ソースのドリルスルー・レポート, 322
カスタム・データ・ロード設定
  SQL に関する制限, 369
  モデル・プロパティ, 211
カタログ。「カタログ・データベース」を参照
カタログのインポート
  カタログ XML ファイルのインポート, 76
  情報, 72
カタログのエクスポート
  カタログ・データベース全体のエクスポート, 74
  情報, 72
  選択したカタログ要素のエクスポート, 74
カタログの移行。「Integration Services カタログの移行」を参照
カタログ・データベース
  server.properties ファイル、使用, 30
  アップグレード, 31
  制限, 359
  定義済, 30
  構成, 30
  権限, 360
  移行, 31
  設定, 29

カレンダ階層
ISO, 193
グレゴリオ暦, 190
リテール, 192
会計, 190
制限, 367
製造, 194
キューブの配置履歴の消去, 79
キューブ・スキーマ
Essbase モデル, 208
キューブ・スキーマ・ウィザードでの Essbase モデルの作成, 202
データ・ロード・バインディング、定義, 204
データ・ロード・マッピング、定義 , 204
メジャー、指定, 200
会計次元の指定, 202
作成, 199
Oracle BI EE ソース, 100
既存のキューブ・スキーマからの Essbase モデルの作成, 206
概要, 199
編集, 199
設定オプション, 202
階層
プレビュー, 201
指定, 200
キューブ・スキーマでのデータ・ロードのバインディング, 204
キューブ・スキーマでのデータ・ロードのマッピング, 204
キューブ・リンケージ
個々のキューブ, 309
個々のキューブの事前の手順, 309
再ホスト後の更新, 311
定義済, 308
情報, 308
更新, 308
キューブ・リンケージの更新, 308
キューブ配置
Essbase サーバーへの接続, 295
Essbase サーバー接続, 295
MaxL スクリプトとして保存, 302
ウィザード, 295
キューブ配置ウィザードのストリーミング・モードオプション, 297
サーバー情報, 297
シナリオ, 292
スケジュール・オプション, 299, 302
ストリーミング・キューブ・ビルディング, 292
データベースの復元, 300
データ・ソース設定情報, 297
データ・ロード・オプション, 299
メンバー、削除, 300
モデル・プロパティ
変更, 299
情報の提供, 297, 299
ルールおよびルール・ファイル, 301
ルール・ファイル、作成, 300
レコードの拒否、キューブ配置中の設定, 301
レコード拒否の設定, 299
ロード・オプション, 291, 299
制限, 373
削除
データベース, 300
メンバー, 300
増分ロード
次元およびメンバーの選択, 300
設定, 304
履歴, 307
履歴、消去, 79
履歴の消去, 79
情報, 291
接続情報, 297
概要, 291
表示履歴, 307
非ストリーミング・キューブ・ビルディング, 292
キューブ配置のシナリオ, 292
キューブ配置のスケジュール・オプション, 302
キューブ配置のデータ・ロード・オプション, 299
キューブ配置のロード・オプション, 299
キューブ配置中のメンバーの削除, 300
キューブ配置中のレコードの拒否、設定, 301
グループ化(CPL 式)
例, 389
判読の支援, 389
接続, 389
部分式を関連付ける方法, 389
グレゴリオ暦カレンダ階層, 190

コピー
メタデータ要素, 166
別名セット, 175

さ行

サイクル依存関係の制限, 365
サポート
列のデータ型, 362
サンプル・データ
データ・ソースの表示, 123
ミニスキーマでの表示, 138
次元要素の表示, 124
サーチ, 344。「検索」も参照
オプションの検索, 345
タイプ, 344
メタデータ要素, 344
メタデータ要素、概要, 343
シングルチェーン階層, 185
スキップ・オプション
メンバー内, 261
次元, 236
スキーマ。「ミニスキーマ」を参照
スキーマ・プリファレンス、設定, 347
スクレーピング。「イントロスペクション」を参照
スケーリング、データ・ロード, 263
ストリーミング・モード
server.properties での設定, 42
キューブ配置ウィザードでの設定, 297
ソース・テーブルおよび列、プロパティの表示, 121
ソート
別名セットのバインディング, 173
別名セットの次元要素, 173

た行

タイム・ゾーン構成の問題
Essbase Studio サーバーの起動ファイル, 58
Oracle ユーザー, 58
タイム・バランス
メンバー内, 260
概要, 235
次元, 234
ツール・チップ, 284
テキスト・ファイルのソース・ファイル、プロパティの表示, 122
テキスト・ファイルのデータ・ソースのモデリング, 107
テキスト・ファイル・ソース
ソース・ファイル、プロパティの表示, 122
ミニスキーマへの移入, 109
ミニスキーマ・オプションの選択, 109
ミニスキーマ作成のスキップ, 109
メタデータ要素の作成, 110
モデリング, 107
列、プロパティの表示, 123
接続の作成, 104
接続パラメータの定義, 105
テキスト・ファイル・ソースの列、プロパティの表示, 123
テスト
ドリルスルー・レポート、シナリオ例, 329
ドリルスルー・レポート、拡張オプションの例, 329
ドリルスルー・レポート、遅延オプションの例, 330
リレーショナル・ソースのドリルスルー・レポート, 328
テンプレート SQL、ドリルスルー・レポート用に定義する, 325
テーブル
サンプル・データ, 138
サンプル・データの表示, 138
ミニスキーマから削除, 134, 144
ミニスキーマでの色の適用, 139
ミニスキーマでの配置, 144
ミニスキーマへの追加, 138
ミニスキーマ関連の追加, 144
メタデータ要素の作成, 139
ユーザー定義、ステートメントの表示, 122
ユーザー定義、プロパティの表示, 121
接続ウィザードでのリレーショナル・テーブルの選択, 94
接続ウィザードでサブセットを追加, 89
結合のフィルタ処理の結果, 137
結合の追加, 134
データの保管方法
メンバー, 256
次元, 232
データベース
2 レベル(CPL 式), 385
3 レベル(CPL 式), 385
命名ルール, 377
データ・ソース

制限, 359
権限, 360
データ・ソースとしての Excel ファイル
制限, 362
データ・ソース・ナビゲータ
タブ, 24
作業領域, 25
データ・ソース接続
CPL 式, 385
わかりやすい名前, 125
イントロスペクション, 118
テキスト・ファイル・ソースのモデリング, 107
テーブルの削除, 115
パラメータの定義, 88
Essbase, 101
Performance Management Architect ソース, 102
テキスト・ファイル・ソース, 105
リレーショナル・ソース, 91
ミニスキーマのスキップ
テキスト・ファイル・ソース, 109
リレーショナル・ソース, 96
ミニスキーマへの移入
テキスト・ファイル・ソース, 109
リレーショナル・ソース, 96
ミニスキーマ・オプション, 89
テキスト・ファイル・ソース, 109
リレーショナル・ソース, 95
ユーザー定義テーブル
例, 127
情報, 127
リストのリフレッシュ, 125
作成
Oracle BI EE ソースの Essbase モデル, 100
Oracle BI EE ソースのキューブ・スキーマ, 100
Oracle BI EE ソースの次元, 98
Performance Management Architect ソース, 102
Performance Management Architect ソース用のメタデータ要素, 103
テキスト・ファイル・ソース, 104
テキスト・ファイル・ソースのメタデータ要素, 110
メタデータ要素, 90
ユーザー定義テーブル, 126
リレーショナル・ソース, 90
リレーショナル・ソースのミニスキーマ, 96
リレーショナル・ソースのメタデータ要素, 98
ワークフローの説明, 88
削除, 125
増分更新, 111
テキスト・ファイルのデータ・ソース, 113
リレーショナル・データ・ソース, 111
接続プロパティの編集, 118
操作, 118
概要, 87
表示
サンプル・データ, 123
ソース・テーブルおよび列のプロパティ, 121
テキスト・ファイルのソース・ファイル, 122
テキスト・ファイル・ソースの列, 123
テーブルのプロパティ, 121
ユーザー定義テーブルのステートメント, 122
列のプロパティ, 122
選択
テキスト・ファイル・ソースのミニスキーマ・オプション, 108
テーブルのサブセット, 89
リレーショナル・ソースのテーブル, 94
データ・ソース接続のわかりやすい名前、表示, 125
データ・ソース権限, 364
データ・ロード SQL
上書き, 211
概要, 212
編集ガイドライン, 212
データ・ロードの最適化, 244
データ・ロード・スケーリング, 263
ドリルスルー・テスト・シナリオの例, 329
ドリルスルー・レポート
Essbase モデルとの関連付け, 340
FDM URL ターゲット, 333, 334
Java メソッド・タイプ、定義, 337
OBI EE URL ターゲット, 333, 336
URL タイプ、定義, 332
URL ターゲット, 333

「ドリルスルー・レポート・プロパティ」ダイアログ・ボックス
「メイン」タブ, 319
「レポート・コンテンツ」タブ, 321
「関連付け」タブ, 340
カスタマイズ, 321
サンプル FDM URL テンプレート, 333, 334
サンプル OBI EE URL テンプレート, 333, 336
サンプル URL テンプレート, 333
テスト・シナリオの例, 329
テンプレート SQL、定義, 325
リレーショナル・ソースのカスタマイズ, 322
リレーショナル・ソースのテスト, 328
リレーショナル・ソースのフィルタの指定, 327
レポート・タイプ、リレーショナル・ソースの定義, 322
レポート・タイプの定義, 321
交差レベル、指定, 319
作成, 319
例、拡張オプションのテスト, 329
例、遅延オプションのテスト, 330
列の順序, 322
制限, 375, 380
定義済, 318
行ガバナー、リレーショナル・ソースの定義, 326
ドリルスルー・レポートと Essbase モデルの関連付け, 340
ドリルスルー・レポートのサンプル FDM URL テンプレート, 333, 334
ドリルスルー・レポートのサンプル OBI EE URL テンプレート, 333, 336
ドリルスルー・レポートのサンプル URL テンプレート, 333
ドリルスルー・レポートの列の順序, 322

な行

ナビゲータ
メタデータ, 25

は行

バインディング
別名セットからの削除, 174
別名セットでのソート, 173
別名セットでの手動による作成, 170
別名セットへの追加, 174
検査による別名セットでの作成, 171
階層での遅延, 182
ファクト
テーブル, 208
モデル, 208
ファクト・テーブル
作成
Oracle BI EE ソース, 100
フィルタ
イントロスペクション
ファクト・テーブル, 149
次元テーブル, 149
リレーショナル・ソースのドリルスルー・レポートでの指定, 327
次元要素, 156
フラット・ファイル(CPL 式), 385
フラット・ファイル・ソース
「テキスト・ファイル・ソース」を参照, 104
ブール定数(CPL 式), 387
ブール範囲、指定, 223
プリファレンス
Essbase モデルの設定, 348
スキーマの設定, 347
設定、一般, 347
プレビュー
データ・ソースのサンプル・データ, 123
ミニスキーマのサンプル・データ, 138
次元要素のサンプル・データ, 124
プロパティ
Essbase モデルの設定, 209
データ・ソース接続の編集, 118
ミニスキーマでの編集, 144
ミニスキーマ作成時の設定, 130

ま行

マルチチェーン階層, 185
ミニスキーマ, 129
オプション
テキスト・ファイル・ソース、接続ウィザード, 109
接続ウィザードのリレーショナル・ソース, 95
サンプル・データの表示, 138
スキップ

テキスト・ファイル・ソース、接続ウィザード, 109
接続ウィザードのリレーショナル・ソース, 96
ズーム・アウト, 143
ズーム・イン, 143
テキスト・ファイル・ソースのオプションの選択, 108
テーブル、プロパティの表示, 133
テーブルの追加, 138
テーブルを削除, 134
プロパティの編集, 144
ミニスキーマ作業領域でのテーブルのレイアウト, 144
ミニスキーマ作業領域でのテーブルの配置, 144
メタデータ要素、ミニスキーマ・オブジェクトから作成, 139
リストのリフレッシュ, 145
作成, 130
テーブルの削除, 131
テーブルの追加, 131
フィルタ、指定, 131
全般プロパティの設定, 130
関連テーブルの追加, 131
作業領域、使用, 141
作業領域の最大化, 143
作業領域の最小化, 143
例, 130
保存, 133
列、プロパティの表示, 133
削除, 145
情報, 129
接続ウィザードでのオプションの選択, 89
接続ウィザードでのテキスト・ファイル・ソースのモデリング, 107
接続ウィザードでのリレーショナル・ソースの作成, 96
操作, 132
検査による結合の追加, 137
概要, 129
移入
テキスト・ファイル・ソース、接続ウィザード, 109
接続ウィザード, 90
接続ウィザードのリレーショナル・ソース, 96
結合の編集, 134
結合の追加, 134
編集
テーブルの削除, 131
テーブルの追加, 131
フィルタ、指定, 131
関連テーブルの追加, 131
色、適用, 139
表示, 140
開く, 133
ミニスキーマのズーム・アウト, 143
ミニスキーマのズーム・イン, 143
ミニスキーマの例, 130
ミニスキーマ作業領域。「作業領域」を参照
ミニスキーマ作業領域でのテーブルのレイアウト, 144
ミニスキーマ作業領域でのテーブルの配置, 144
ミニスキーマ作業領域の最大化, 143
ミニスキーマ作業領域の最小化, 143
メジャー
キューブ・スキーマの指定, 200
メジャー階層
作成, 180
情報, 178
メタデータ・カタログ, 21
メタデータ・ナビゲータ
作成された要素のタイプ, 25
概要, 25
要素のソース, 25
要素の表示, 25
メタデータ・フォルダ
作成, 166
編集, 166
メタデータ要素
コピー, 166
サンプル・データ, 167
サーチ, 344
サーチ、概要, 343
タイプ, 153
作成 , 154
Performance Management Architect ソース, 103
テキスト・ファイル・ソース , 110
ミニスキーマ・オブジェクト, 139
リレーショナル・ソース, 98
接続ウィザード, 90

制限, 365
削除, 167
名前変更, 166
変更後のモデルの再同期, 287
情報, 153
操作, 166
検索, 343
検索、概要, 343
概要, 153
系列、表示, 167, 313
編集, 154
階層、追加, 180
メンバー
2 パス計算, 256
「ラベルのみ」オプションを持つ集約ストレージ, 257
スキップ・オプション, 261
タイム・バランス, 260
タブ、アクセス, 246
データの保管方法, 256
プロパティ, 245
ユーザー定義属性, 273
リフォーマット, 280
全般プロパティの設定, 246
共有メンバーの前に実メンバーを配置, 242
別名の編集, 265
別名セット、表示, 267
勘定科目、プロパティの編集, 260
可変属性, 249
可変属性と履歴テーブル, 247
可変属性の履歴テーブル, 247
名前、変更とリフォーマット, 277
名前のフォーマットの設定, 221
名前変更, 277
名前変更シーケンス, 274
命名シーケンス, 274
命名ルール, 378
変換, 277
大文字/小文字の変更, 279
子の集計メソッド, 253
属性の設定, 247
属性計算メンバー名のフォーマットの設定, 221
差異レポートの作成方法, 262
平均, 222
式、追加, 264
接尾辞, 276
接頭辞, 275
検索ルール, 277, 278
概要, 245
編集, 253
複数の別名, 266
計算の解決順, 255
集約ストレージ・オプション, 258
メンバーの変換, 277
モデルの再同期, 287
1 つのメタデータ要素, 288
1 つのモデル, 288
タイプ, 288
変更のトリガー, 287
概要, 287

や行

ユーザー定義 SQL, 364
検証, 364
ユーザー定義 SQL の検証, 364
ユーザー定義ソースの列、プロパティの表示, 122
ユーザー定義テーブル
CPL 式, 385
ソース・データの表示, 126
テーブルのステートメントの表示, 122
プロパティの表示, 121
作成, 126
例, 127
情報, 127
ユーザー定義属性
メンバーへの割当て, 273
使用上のガイドライン, 240
概要, 240
次元への割当て, 239

ら行

リテール・カレンダ階層, 192
リフレッシュ
データ・ソース接続リスト, 125
ミニスキーマ・リスト, 145
リレーショナル・ソース
ミニスキーマへの移入, 96
ミニスキーマ・オプションの選択, 95
ミニスキーマ作成のスキップ, 96
作成
ミニスキーマ, 96

メタデータ要素, 98
接続, 90
接続パラメータの定義, 91
リレーショナル・ソースの列、プロパティの表示, 122
リレーショナル・ソース・テーブル、プロパティの表示, 121
ルール・ファイル、キューブ配置中の作成, 300
レコードの拒否、キューブ配置中の設定, 301
ロギング
ODL の構成, 57
非推奨のサーバー・プロパティ, 57

わ行

一般プリファレンス、設定, 347
交差レベル
ドリルスルー・レポートでの指定, 319
代替階層, 185
仮想メモリー
増加, 78
構成, 78
会計カレンダ階層, 190
会計次元
キューブ・スキーマでの指定, 202
メンバーの編集, 260
使用するためのルール, 226
次元タイプとして選択, 229
作成
Essbaes モデル
Oracle BI EE ソース、接続ウィザード, 100
Essbase への接続, 101
Essbase モデル
キューブ・スキーマ・ウィザード, 202
既存のキューブ・スキーマ, 206
キューブ・スキーマ, 199
Oracle BI EE ソース、接続ウィザード, 100
キューブ配置中のルール・ファイル, 300
データ・ソース接続
Performance Management Architect ソース, 102
テキスト・ファイル・ソース, 104
リレーショナル・ソース, 90
ワークフロー, 88
ドリルスルー・レポート, 319
ミニスキーマ, 130
テーブルの削除, 131
テーブルの追加, 131
フィルタの指定, 131
メタデータ要素、オブジェクト, 139
全般プロパティの設定, 130
関連テーブルの追加, 131
メタデータ要素
テキスト・ファイル・ソース、接続ウィザード, 110
接続ウィザード, 90
接続ウィザードの Performance Management Architect ソース, 103
接続ウィザードのリレーショナル・ソース, 98
別名セット, 170
手動での別名セットのバインディング, 170
検査による別名セットのバインディング, 171
次元
Oracle BI EE ソース、接続ウィザード, 98
作業領域, 25
オブジェクトの表示, 25
ミニスキーマでの使用, 141
使用, 281
開く, 281
保存
Essbase モデル, 211
server.properties ファイル, 33
スキップ・オプション, 237
タイム・バランス, 235
ミニスキーマ, 133
差異レポートの作成方法, 236
次元
タイプ, 229
プロパティ, 227
保管方法, 231
次元のユーザー定義属性, 240
次元内の式, 237
重複するメンバー設定, 242
階層ストレージ設定, 241
共有メンバー、前に実メンバーを配置, 242
共有メンバーと実メンバーの位置を逆にする, 242
共有メンバーを持つマルチチェーン階層。「代替階層」を参照
共有メンバー処理, 213, 356
共通プラットフォーム言語。「CPL」を参照
再ホスト

EPM System 製品への参照の更新, 80
Essbase サーバーへの参照の更新, 81
Essbase サーバーへの参照の更新について, 80
キューブ・リンケージの更新, 311
再帰階層, 187
列
「データ・ソース・ナビゲータ」での表示, 24
サンプル・データ、表示, 123
タイプの上書き, 108
タイプの選択, 108
プロパティの表示
テキスト・ファイルのソース・ファイル, 123
ミニスキーマ, 133, 138
ユーザー定義テーブル, 122
リレーショナル・ソース, 122
ミニスキーマでの表示, 24
メタデータ要素で表現, 90
メタデータ要素のリレーショナル・ソース, 98
先頭行の名前
テキスト・ファイルのデータ・ソース, 106
テキスト・ファイルのモデリング, 107
増分更新, 114
名前変更, 108
属性, 208
結合のフィルタ処理の結果, 137
結合の追加, 134
非表示, 108
列のデータ型
サポート, 362
列の先頭行の名前
テキスト・ファイルのデータ・ソース, 106
テキスト・ファイルのモデリング, 107
増分更新, 114
別名
Integration Services カタログの移行, 353
スペースのリフォーマット, 270
プロパティ・タブへのアクセス, 265
メンバーの編集, 265
使用, 266
名前変更, 267
命名ルール, 378
変更およびリフォーマット, 267
大文字/小文字の変更, 270
接尾辞, 272
接頭辞, 271
検索ルール, 268, 269
概要, 266
複数, 266
設定、表示, 267
別名セット
Essbase モデル内で使用可能にする, 217
エクスポート, 176
コピー, 175
テーブルを選択, 217
バインディング
ソート, 173
削除, 174
変更, 173
手動作成, 170
検査による作成, 171
追加, 174
作成, 170
削除, 176
名前変更, 175
操作, 170
概要, 169
次元要素のソート, 173
管理, 175
編集, 172
表示, 267
制限
Essbase Studio サーバーへのアクセス, 359
XOLAP モデル, 365
イントロスペクション, 364
カタログ, 359
カレンダ階層, 367
キューブ配置, 373
サイクル依存関係, 365
テキスト・ファイル・ソースのメンバーの変換, 372
データ・ソース, 359
データ・ソースとしての Excel ファイル, 362
ドリルスルー・レポート, 375, 380
ミニスキーマへの物理テーブルの追加, 138
メジャー階層, 367
メタデータ要素, 365
導出テキスト・メジャー, 365
標準階層, 367

独立次元のバインディング, 371
階層, 366, 367
削除
データ・ソース接続, 24, 125
データ・ソース接続、テーブル , 115
データ・ソース接続からのテーブル, 115
ミニスキーマ, 145
ミニスキーマでの結合, 136
ミニスキーマのテーブル, 144
別名セット, 176
別名セットからのバインディング, 174
動的時系列, 230
動的計算
メンバー, 257
属性メンバー名, 221
次元のデータ保管, 232
可変属性
エクスポート/インポート後の再定義, 371
シナリオ例, 220
メンバーの設定, 249
履歴テーブル、設定, 247
概要, 219
独立次元のバインディング、定義, 251
独立次元バインディング、制限, 371
編集ガイドライン, 371
可変属性の履歴テーブル, 247
合計メンバー, 222
名前で制限されている文字, 380
名前の予約語, 380
名前変更
メンバー, 277
別名セット, 175
変換シーケンス, 274
命名ルール
アプリケーション, 377
データベース, 377
ドリルスルー・レポート, 380
メンバー, 378
予約語, 380
使用できない文字, 380
別名, 378
導出テキスト・メジャー, 378
次元, 378
次元要素, 378
増分更新
既存のテキスト・ファイルのデータ・ソース, 113
既存のリレーショナル・データ・ソース, 111
既存データ・ソース, 111
変更。「編集」を参照
大文字と小文字を区別する名前, 378
子メンバーの集計
メソッド, 253
概要, 254
定数(CPL 式), 386
定義
Essbase の接続パラメータ, 101
キューブ・スキーマでのデータ・ロードのバインディング, 204
キューブ・スキーマでのデータ・ロードのマッピング, 204
データ・ソース接続パラメータ
Performance Management Architect ソース, 102
テキスト・ファイル・ソース, 105
リレーショナル・ソース, 91
接続ウィザード, 88
ドリルスルー・レポート・タイプ, 321
ドリルスルー・レポート用のテンプレート SQL, 325
リレーショナル・ソースのドリルスルー・レポート・タイプ, 322
密次元, 231
導出テキスト・メジャー
作成, 154, 162
制限, 365
命名ルール, 378
編集, 154, 162
属性
Essbase Studio と Integration Services の処理の違い, 357
Essbase モデルでの定義, 218
タイプ, 219
メンバー名のフォーマットの設定, 221
使用方法のガイドライン, 219
属性計算メンバー名のフォーマットの設定, 221
概要, 218
次元のユーザー定義, 239
集計メソッド, 218
属性次元を持つマルチチェーン階層, 186
属性計算の概要, 223
属性計算メンバー名のフォーマット

設定, 221
属性計算次元, 222
差異レポートの作成方法
メンバー内, 262
次元, 235
平均メンバー, 222
式
メンバーへの追加, 264
情報, 237
概要, 237
次元への追加, 237
拡張オプション要素のドリルスルー・テストの例, 329
指定
Essbase モデルのメンバー・プロパティ, 246
キューブ・スキーマのメジャー, 200
キューブ・スキーマの階層, 200
ドリルスルー・レポートの交差レベル, 319
ドリルスルー・レポートの列の順序, 322
リレーショナル・ソースのドリルスルー・レポート・フィルタ, 327
リレーショナル・ソースの行ガバナー, 326
接尾辞
メンバー、追加, 276
別名、追加, 272
接続。「データ・ソース接続」を参照
接続ウィザード
ワークフロー, 88
概要, 87
接頭辞
メンバー、追加, 275
別名、追加, 271
数値(SQL 関数), 404
数値定数(CPL 式), 386
数値範囲、指定, 223
整数定数(CPL 式), 387
文字列(SQL 関数), 398
文字列定数(CPL 式), 387
日付範囲、指定, 223
日付要素、作成, 165
日付要素から構築された時間階層, 189
時間次元, 225, 229
最大メンバー, 222
最小メンバー, 222
最後に配置したフィールド, 210
検索, 343。「サーチ」も参照
メタデータ要素, 343
メタデータ要素、概要, 343
単語の一致, 344
大文字/小文字の一致, 344
現在の選択項目への適用, 344
検索とサーチ
概要, 343
機能の違い, 343
検索ルール
メンバー, 277, 278
別名, 268, 269
構成
Essbase Studio サーバーの起動ファイル, 57
JDBC ドライバ, 66
ODL, 57
server.properties, 33
ロギング, 57
仮想メモリー, 78
構文の要素(CPL 式), 383
標準次元, 229
標準階層, 180
権限
カタログ・データベース, 360
データ・ソース, 360
ユーザー定義 SQL データ・ソース, 364
次元
「プロパティ」ダイアログ・ボックスのタブ, 224
アウトライン構築オプション, 240
スキップ・オプション, 236
ソート順, 244
タイプ, 225, 229
タイム・バランス, 234
タブ、アクセス, 226
データの保管方法, 232
データ・ロード、最適化, 244
ファクト, 208
プロパティ, 224
ユーザー定義属性, 240
ユーザー定義属性、割当て, 239
会計次元の使用, 226
作成
Oracle BI EE ソース, 98
保管方法, 231
別名, 238
勘定科目, 229
勘定科目、プロパティの編集, 234
命名ルール, 378

定義, 208
密, 231
差異レポートの作成方法, 235
式、追加, 237
時間, 229
時間次元の使用, 225
概要, 224
標準, 229
疎, 231
編集, 228
計算の解決順, 233
重複するメンバー設定, 242
階層ストレージ設定, 241
次元のソート, 244
次元の順序付け, 244
次元要素
キー・バインディング, 160
サンプル・データの表示, 124
ソート順, 157
フィルタ, 156
一般プロパティの定義, 156
作成
「データ・ソース・ナビゲータ」から, 155
メタデータ・ナビゲータ, 155
別名セットでのソート, 173
命名ルール, 378
概要, 153
編集
一般プロパティ, 156
演算子
CPL 式, 388
グループ化, 389
シーケンス, 388
文字列, 393
算術, 391
複数, 388
論理, 389
関数, 388
順序, 388
独立次元のバインディング
制限, 371
定義, 251
疎次元, 231
移行。「Integration Services カタログの移行」を参照
カタログ・データベース, 31
系列
作業領域
使用, 315
移動, 315
要素の表示, 315
開く, 315
概要, 313
要素の表示, 313
結合
ミニスキーマでの検査による追加, 137
ミニスキーマでの編集, 134
ミニスキーマでの追加, 134
編集
キューブ・スキーマ, 199
データ・ソース接続プロパティ, 118
ミニスキーマ
テーブルの削除, 131
テーブルの追加, 131
フィルタの指定, 131
プロパティ, 144
結合, 134
関連テーブルの追加, 131
メンバー, 253
別名セット, 172
別名セットのバインディング, 173
勘定科目、次元プロパティ, 234
次元, 228
階層, 197
色、ミニスキーマへの適用, 139
行ガバナー、リレーショナル・ソースの定義, 326
表示
キューブ配置の履歴, 307
ソース・テーブルのプロパティ, 121
データ・ソースのサンプル・データ, 123
ミニスキーマ
ズーム・アウト, 143
ズーム・イン, 143
作業領域, 140
ミニスキーマのサンプル・データ, 138
ミニスキーマ作業領域の最大化, 143
ミニスキーマ作業領域の最小化, 143
次元要素のサンプル・データ, 124
製造カレンダ階層, 194
親子階層。「再帰階層」を参照
解決順、メンバー, 255
計算の解決順
メンバー, 255

次元, 233
設定
Essbase モデルのプリファレンス, 348
キューブ・スキーマ・オプション, 202
キューブ配置中のレコードの拒否, 301
スキーマ・プリファレンス, 347
一般プリファレンス, 347
追加
ミニスキーマでの検査による結合, 137
ミニスキーマでの結合, 134
別名セットへのバインディング, 174
遅延オプション要素のドリルスルー・テストの例, 330
重複するメンバー設定、移動, 242
重複メンバー名
サポート制約事項, 369
使用可能にする, 210
開く
ミニスキーマ, 133
階層
Essbase モデル構造の決定, 177
ISO カレンダ、作成, 193
アウトライン構築の代替, 240
カレンダ
ISO, 193
グレゴリオ暦, 190
リテール, 192
会計, 190
作成, 189
製造, 194
キューブ・スキーマでのプレビュー, 201
キューブ・スキーマの指定, 200, 208
キューブ配置、使用, 304
キューブ配置での更新, 304
グレゴリオ暦カレンダ、作成, 190
シングルチェーン, 185
シングルチェーン、多世代, 185
ストレージ設定、選択, 241
タイプ, 184
マルチチェーン, 185
メジャー、作成, 180
メタデータ要素、追加, 180
リテール・カレンダ、作成, 192
リンク値属性, 196
代替, 185
会計カレンダ、作成, 190
例
シングルチェーン, 185
シングルチェーン、多世代, 185
マルチチェーン, 185
代替, 185
再帰, 187
属性次元を持つマルチチェーン, 186
日付要素から構築された時間, 189
再帰, 187
制限, 366, 367
子および兄弟、追加, 181
属性次元を持つマルチチェーン, 186
日付属性, 196
日付要素から構築された時間, 189
概要, 177
標準、作成, 180
物理要素、追加, 181
編集, 197
製造カレンダ、作成, 194
要素、追加, 180
階層での遅延キー・バインディング列, 182
階層の日付属性, 196
階層内のリンク値属性, 196
集約ストレージ
Essbase モデル内で使用可能にする, 210
メンバーのオプション, 258