

Oracle® Hyperion Enterprise Performance Management Architect

Batch Client User's Guide

リリース 11.1.2.4

著作権情報

Oracle® Hyperion Enterprise Performance Management Architect Batch Client User's Guide, 11.1.2.4

Copyright © 2006, Oracle and/or its affiliates. All rights reserved.

著者: EPM 情報開発チーム

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

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

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

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

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

U.S. GOVERNMENT END USERS:

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

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

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

目次

ドキュメントのアクセシビリティについて	5
ドキュメントのフィードバック	6
第1章. Performance Management Architectバッチ・クライアントの使用法	7
バッチ・クライアントの起動	8
コマンド・ライン・オプション	8
戻りコード	10
ロギング	12
第2章. コマンド・ファイルの構成	15
変数	16
コメント	17
コマンド	17
スクリプト	19
一般的に使用されるコマンド	19
Optionコマンド	19
Login / Logoutコマンド	20
login	20
Logout	22
Quit	22
Exit	23
Copyコマンド	23
Application	23
Dimension	23
Createコマンド	24
Application	24
Dimension	25
Member	25
Create Associationコマンド	26
Delete Associationコマンド	26
Deleteコマンド	27
Application	27
Association	27
Dimension	28
Member	28
Detach Dimensionコマンド	28
Excludeコマンド	29
Member	29
Executeコマンド	29
インポート(アプリケーションまたは共有ライブラリへ)	30
ディメンションの同期(共有ライブラリとの)	31
デプロイメント	31
再デプロイメント	33
データの同期	34
検証	36

Includeコマンド	36
Dimension	36
Insertコマンド	37
Insert Member	37
移動コマンド	37
メンバーの移動	37
Renameコマンド	38
メンバー名の変更	38
Removeコマンド	38
Remove Member (アプリケーションまたは共有ライブラリ)	39
Remove Dimension	39
Share Dimensionコマンド	39
Updateコマンド	40
Application	41
Dimension	41
Member	41
ディメンションの関連付け	42

ドキュメントのアクセシビリティについて

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

Access to Oracle Support

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

ドキュメントのフィードバック

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

次のソーシャル・メディア・サイトでEPM情報開発をフォローできます:

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

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

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

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

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

1

Performance Management Architect バッチ・ク ライアントの使用法

この項の内容:

バッチ・クライアントの起動	8
コマンド・ライン・オプション	8
戻りコード	10
ロギング	12

Oracle Hyperion EPM Architect バッチ・クライアントによって、データのエクスポート、メタデータのロード、データのロード、計算などのプロセスを組み合わせ、通常の夜間または週末のロード・プロセスの処理中に、これらの操作を開始できます。

バッチ・クライアントによる外部スケジューリング・ツールを使用してプロセスを開始できます。バッチ・クライアントを使用すると、次のような多くのタスクを実行できます:

- Performance Management Architect にメタデータをロードする
- デイメンションおよびメジャーのセキュリティに関するプロパティを更新する
- アプリケーションにデータをロードする

バッチ・クライアントは、Performance Management Architect をインストールするときに自動的にインストールされます。バッチ・クライアントは、Windows プラットフォームで実行できます。Performance Management Architect をインストールすると、インストール中に生成されるクラス・パスを設定するためのバッチ・ファイルが自動的に作成されます。

Performance Management Architect バッチ・クライアントは、次の2つのモードで実行できます:

- コマンド・ライン・モード
 - コマンドを対話的に入力できる
 - 各コマンドは、複数の行にまたがることできる
 - 複数のコマンド文をセミコロン (;) で区切ることができる
 - 単一引用符を使用する必要がある
 - コマンドは即座に実行される
- スクリプト・モード
 - 対話なしで一連のコマンドを実行する
 - プログラムの起動時にコマンド・ファイルを指定でき、オプションで結果ログ・ファイルを指定できる
 - サードパーティのスケジューラを使用して、スクリプトの実行をスケジュールできる

バッチ・クライアントの起動

Performance Management Architectバッチ・クライアントは、対話型のコマンド・ライン・モードで実行するか、またはコマンド・ラインで指定するスクリプト・ファイルを実行できます。

バッチ・クライアントを起動するには、「スタート」、「プログラム」、「Oracle EPM System」、「Foundation Services」、「Performance Management Architect」、「EPMAバッチ・クライアントの起動」の順に選択します。

EPM_ORACLE_HOME\products\Foundation\BPMA\EPMABatchClientに移動し、対話型コマンド・ラインまたはスクリプトを使用してepma-batch-client.batファイルを起動することもできます。(LINUXまたはUNIX環境で、EPM_ORACLE_HOME/products/Foundation/BPMA/EPMABatchClientに移動してepma-batch-client.shファイルを起動します。)

たとえば、対話型コマンド・ライン・モードでバッチ・クライアントを起動する場合は、次のようにパラメータなしでバッチ・ファイルを起動します:

```
EPM_ORACLE_HOME\products\Foundation\BPMA\EPMABatchClient\epma-batch-client.bat
```

スクリプトを起動する場合は、スクリプト・ファイル名に-Cオプションを付けて指定する必要があります。すべての追加パラメータは任意です。例: EPM_ORACLE_HOME\products\Foundation\BPMA\EPMABatchClient\epma-batch-client.bat -CMyScript.txt



注:

スクリプト・ファイル名とログ・ファイル名は、相対パスで指定できますが、バッチ・ファイルを起動するフォルダではなく、EPMABatchClientフォルダに対する相対パスにする必要があります。

コマンド・ライン・オプション

バッチ・クライアントを起動するときは、次の引数を指定できます。

表1 バッチ・クライアントのコマンド

コマンド	説明
-H	バッチ・クライアントのヘルプを表示します。
-C	実行するスクリプト・ファイルの名前を指定します。 例: -C'C:\Scripts\LightsOut.txt'
-G	使用する言語を指定します。このパラメータを使用するには、-Gxと入力します。ここで、xは、次の言語コードのいずれかです: <ul style="list-style-type: none">• DA - デンマーク語• DE - ドイツ語• ES - スペイン語• FR - フランス語• IT - イタリア語

コマンド	説明
	<ul style="list-style-type: none"> • JA - 日本語 • KO - 韓国語 • PT_BR - ブラジル・ポルトガル語 • RU - ロシア語 • SV - スウェーデン語 • TR - トルコ語 • ZH_CN - 簡体字中国語 • ZH_TW - 繁体字中国語 <p>たとえば、フランス語の場合は-Gfr、イタリア語の場合は-Gitとなります。-Gオプションが指定されていない場合、バッチ・クライアントはオペレーティング・システムの現在のデフォルト言語を使用しようとしています。リソース・ファイルが見つからない場合は、デフォルト言語である英語が使用されます。特定の言語のリソース・ファイルで文字列が見つからない場合は、英語版が使用されます。</p>
-R	<p>マシンに結果を書き込むファイルの名前を指定します。</p> <p>例: -R'C:\LogFiles\ScriptResult.log'</p>
-L	<p>コマンドをログ出力するかどうかを指定します。デフォルト値は0です。</p> <p>ログのコマンドは次のとおりです:</p> <p>Off (デフォルト) =コマンドをログ出力しない</p> <p>On =コマンドをログ出力する</p> <p>例: -LOn</p>
-S	<p>コマンドが失敗すると、スクリプトの実行を停止します。デフォルト値は1(TRUE)です。</p> <p>エラーが発生した場合は停止: TRUE</p> <p>エラー時に続行: FALSE</p> <p>例: -SFalse エラーが発生しても続行します。</p>
-U	<p>Performance Management Architectにログインするために使用するユーザー名を指定します。</p> <p>例: -U'Admin'</p>
-P	<p>Performance Management Architectにログインするために使用するパスワードを指定します。</p> <p>例: -Ppassword</p>

コマンド	説明
-V	スクリプト確認のオン/オフを切り換えます。検証をオンにすると、実行前にスクリプトの構文エラーがチェックされます。エラーがあると、スクリプトの実行は停止します。「Off」を指定すると、実行前に検証は行われません。「On」を指定すると、実行前にスクリプトが検証されます。デフォルト値「On」で、実行前にスクリプトが検証されます。たとえば、 <code>-voff</code> を指定すると、実行前にスクリプトは検証されません。
-O	検証のみです。これを指定すると、バッチ・クライアントはスクリプトを検証しますが、結果にかかわらず、スクリプトは実行されません。これは、スクリプトに構文エラーがあるかどうかをテストするために使用できます。
-N	コンソールに表示される出力に使用する文字エンコードを指定します例: <code>-Ncp866</code> Windows環境では、デフォルトはロケールのコード・ページ(<code>chcp</code> コマンドの使用を決定できる)です。UNIX環境では、デフォルトはUTF-8です。 注: 日本語のWindows環境では、コード・ページをMS932に設定する必要があります。例: <code>-NMS932</code> 。



注:

-SなどのOn/Offを使用するコマンド・ライン・パラメータでは、0または1、YまたはN、TrueまたはFalse、OnまたはOffを使用できます。たとえば、`-S0`、`-SY`、`-SNo`、`-STrue`、`-SOff`などは、すべて有効です。

次に、コマンド・ライン・オプションの使い方のコード例を示します。

```
epma-batch-client -H
epma-batch-client -C"C:\EPM_ORACLE_INSTANCE\EPMA\Commands.txt"
-R"C:\EPM_ORACLE_INSTANCE\EPMA\ResultFile.txt" -LOn -SFalse
-Uadmin -Ppassword
```

戻りコード

バッチ・クライアントが終了するときは、スクリプトの終了状態に基づいて呼び出しプログラムに結果コードが戻されます。StopOnErrorの設定に基づいて、次の2つのシナリオがあります。

StopOnError = False (-S0)

StopOnErrorがFALSEの場合の戻りコードは、一般的な成否を表します。

0 =成功(エラーなし)

-1 =失敗(1つ以上のエラーが発生)

StopOnError = TRUE (-S1)

StopOnErrorがTRUEの場合、エラーが発生するとバッチ・クライアントは終了し、次の表に基づくコードを戻します。デフォルトでは、StopOnErrorはTRUEに設定されています。

バッチ・クライアントは、成否に基づいて次のような結果コードを戻します。

コマンド	コマンド・コード	クラス	クラス・コード	戻りコード
正常終了	N/A	N/A	N/A	0
一般エラー	N/A	N/A	N/A	-1
検証エラー	N/A	N/A	N/A	1
解析エラー	N/A	N/A	N/A	100
コマンド・ライン・エラー	N/A	N/A	N/A	4
コピー	15	Application	1	1501
コピー	15	Dimension	2	1502
作成	1	Application	1	101
作成	1	Dimension	2	102
作成	1	Member	3	103
作成	1	Association	10	110
デバッグ	21	N/A	N/A	2100
削除	2	Application	1	201
削除	2	Dimension	2	202
削除	2	Member	3	203
削除	2	Association	10	210
添付解除	16	Dimension	2	1602
除外	3	Member	3	303
実行	4	DataSynchronization	4	404
実行	4	Deploy	5	405
実行	4	DimensionSynchronization	6	406
実行	4	Import	7	407
実行	4	Redeploy	9	409
実行	4	Validate	12	412
終了	5	N/A	0	500
インクルード	6	Dimension	2	602
インクルード	6	Member	3	603
挿入	7	Member	3	703
ログイン	8	N/A	0	800
ログアウト	9	N/A	0	900

コマンド	コマンド・コード	クラス	クラス・コード	戻りコード
移動	19	Member	3	1903
オプション	20	N/A	0	2000
終了	10	N/A	0	1000
削除	11	Dimension	2	1102
削除	11	Member	3	1103
名前変更	18	Member	3	1803
設定	12	N/A	0	1200
共有	17	Dimension	2	1702
更新	13	Application	1	1301
更新	13	DimensionAssociation	8	1308
更新	13	Dimension	2	1302
更新	13	Member	3	1303
変数	14	N/A	0	1400

DOSのバッチ・ファイルまたはWindowsのコマンド・ファイルで、次のようにして、エラーをチェックできます:

```
Call epma-batch-client.bat .\scripts\MyScript.txt
IF ERRORLEVEL 0 goto ON_SUCCESS
IF ERRORLEVEL 100 goto PARSE_ERROR
If ERRORLEVEL 101 goto APP_CREATE_FAILED
```

ロギング

バッチ・クライアントには、結果ファイルとトレース・ファイルを使用して出力できるいくつかのロギングのレベルがあります。結果ファイルには、コマンド・ウィンドウに入力するコマンドの詳細およびその実行ステータス、エラーまたは警告メッセージが含まれます。トレース・ファイルには、デバッグに役立つスタック・トレースの詳細が含まれます。

バッチ・クライアントでは、ロギングにOracle Diagnostics Logging (ODL)を使用します。すべての構成パラメータがconfディレクトリのlogger.xmlに格納されます。たとえば、EPM_ORACLE_INSTANCE\products\Foundation\BPMA\EPMABatchClient\outputなどです。

次に、logger.xmlファイルのコード例を示します。

```
<logging_configuration>
  <log_handlers>
    <log_handler name="traceFile"
class="oracle.core.ojdl.logging.ODLHandlerFactory">
      <property name="path" value="{user.dir}/output/EPMABatchClientTrace.log"/>
      <property name="maxFileSize" value="1000000"/>
      <property name="maxLogSize" value="5000000"/>
      <property name="useSourceClassAndMethod" value="true"/>
      <property name="encoding" value="UTF-8"/>
    </log_handler>
    <log_handler name="console"
      level="ALL"
```

```
        class="java.util.logging.ConsoleHandler"
        formatter="com.hyperion.bpma.logger.GenericFormatter"/>
    <log_handler name="resultsFile"
        class="com.hyperion.bpma.logger.GenericFileHandler"
        formatter="com.hyperion.bpma.logger.GenericFormatter"
        level="ALL"/>
</log_handlers>
<loggers>
    <logger name="BatchClient.trace" level="ALL" useParentHandlers="false">
        <handler name="traceFile"/>
    </logger>
    <logger name="BatchClient.console" level="INFO" useParentHandlers="false">
        <handler name="console"/>
    </logger>
    <logger name="BatchClient.results" level="INFO" useParentHandlers="false">
        <handler name="resultsFile"/>
    </logger>
</loggers>
</logging_configuration>
```

コマンドが正常終了したかどうかを簡単に判断できるように、次のコマンドについて、関連するジョブIDおよびジョブID URLがログに記録されます。

- Execute Deploy
- Execute ReDeploy
- Execute Validate
- Execute DataSynchronization
- Execute Import
- Copy Application
- Detach Dimension
- Share Dimension



注:

Java 1.6で使用可能な標準フォーマッタを使用するオプションがあります。
たとえば、`com.hyperion.bpma.logger.GenericFormatter`のかわりに
`java.util.logging.SimpleFormatter`を使用できます。

2

コマンド・ファイルの構成

この項の内容:

変数	16
コメント	17
コマンド	17
スクリプト	19
一般的に使用されるコマンド	19

コマンド・ファイルは、バッチ・クライアントの入力ファイルです。これには、次のうちの1つ以上が含まれている可能性があります:

- コマンド
- 変数、宣言および割当て
- コメント

次に、コマンド・ファイルのコード例を示します。

```
// Test Script
set bpmaserverurl=http://localhost/hyperion-bpma-server;
set workspaceurl=http://localhost:19000/workspace;

login admin,password;

set ApplicationName = 'Sample';

// Delete some members
Delete Member
Properties(MemberName, DimensionName, ParentName, DeleteAllDescendants)
Values('M1-1-1', 'A1', 'M1-1', true);

Delete Member
Properties(MemberName, DimensionName, ParentName, DeleteAllDescendants)
Values('M1', 'A1', '#root', false);

Delete Member
Properties(MemberName, DimensionName, ParentName, DeleteAllDescendants)
Values('M1', 'A1', '#root', true);

Delete Dimension
Properties(DimensionName)
Values('A1');

Delete Dimension
Properties(DimensionName)
Values('E1');
```

```

Delete Application
  Properties(ApplicationName, WaitForCompletion)
  Values('TestApp1');

set ApplicationName = '';

// Delete shared dims
Delete Dimension
  Properties(DimensionName)
  Values('S1');

quit;

```

コマンド・ファイルのほとんどのコマンドは即座に実行されます。ただし、EXECUTEコマンドは実行に時間がかかり、WaitForCompletionパラメータをサポートします。DIMSYNCRONIZATIONを除くすべて実行コマンドはこのパラメータをサポートします。たとえば、WaitForCompletionパラメータを使用して、バッチ・クライアントにコマンドが実行されるまで待機させることもできます。次のコマンドにWaitForCompletionパラメータの例を示します。この場合、管理者は実行時間が比較的長いデータ同期コマンドを実行しています。実行に長い時間を要するその他のコマンドとしては、インポートおよびアプリケーションのデプロイメントがあります。

```

execute datasynchronization
  parameters(DataSynchronizationName, DataTransformationOperator,
  DataTransformationValue, FileName, ValidateOnly, WaitForCompletion)
  values('CommaSync3', '*', '1.2345', '', 'false', 'true');

```

変数

変数は、定義してスクリプトで呼び出すことができます。変数の特性は次のとおりです：

- 変数には、名前と単一の値のタイプがある
- 変数の値は、任意のデータ型にできる
- 一度定義した変数を複数の場所で使用できる
- コマンド間で変数の値を変更できる
- 変数名には大文字と小文字の区別がある
- varキーワードを使用して変数を定義し、\$を使用して変数を参照できる

次に、変数を使用して共有ディメンションを作成するコード例を示します。

```

// Create Shared Dimension Script
set bpmserverurl=http://localhost/hyperion-bpma-server;
set workspaceurl=http://localhost:19000/workspace;

login admin,password;
var DimType='Scenario';
// Create a shared dimension
create Dimension
  Properties(DimensionName, DimensionDescription, DimensionType)
  Values('S1', 'New Scenario', '$DimType');

```


次の3つのタイプの変数があります:

- ユーザー

ユーザー変数は、`var`キーワードを使用して割り当てます。例:

```
var variable1 = 'abc';
```

- システム

システム変数は、`set`コマンドを使用して割り当てます。例:

```
set bpmaserverurl='http://localhost/hyperion-bpma-server';  
set workspaceurl='http://localhost:19000/workspace';
```

- オブジェクト

オブジェクト変数は、`set`コマンドを使用して割り当てます。例:

```
set ApplicationName = 'Comma';  
  
set dimension=Account;
```

コメント

行の先頭に2つのスラッシュ(`//`)を記述すると、スクリプト内の任意の行をコメントにできます。複数の行にわたってコメントにする場合は、各行をコメントにする必要があります。例:

```
//execute datasynchronization  
// parameters(DataSynchronizationName, DataTransformationOperator,  
DataTransformationValue, FileName,  
// ValidateOnly, WaitForCompletion)  
// values('CommaSync3', '*', '1.2345', '', 'false', 'true');
```

コマンド

コマンドの構成要素は次のとおりです:

- コマンド動詞
- コマンド・クラス
- パラメータ値の集合またはプロパティ値の集合

パラメータおよびその値は、カンマによって区切ります。サポートされているコマンド動詞は、次のとおりです:

- CREATE
- COPY
- DEBUG
- DELETE
- DETACH
- EXCLUDE
- EXECUTE
- EXIT
- INCLUDE
- INSERT
- LOGIN
- LOGOUT
- MOVE
- OPTION
- QUIT
- REMOVE
- RENAME
- SHARE
- UPDATE
- SET
- VARIABLE

サポートされているコマンド・クラスは、次のとおりです:

- APPLICATION
- ASSOCIATION
- DIMENSION
- DIMENSIONASSOCIATION
- DIMSYNCHRONIZATION
- MEMBER
- IMPORT
- DEPLOY
- DATASYNCHRONIZATION
- REDEPLOY
- VALIDATE



ヒント:

ディメンションのrootメンバーには、#root定数を使用します。共有ライブラリを宛先にするコマンドのアプリケーション名には、#shared定数を使用します。

スクリプト

スクリプトは、連続して実行できるコマンドの集合です。スクリプトには、任意の順序でコマンドを記述できますが、loginのような特定の初期化コマンドは、他のコマンドより前に実行する必要があります。スクリプト内の各コマンドはセミコロン(;)によって区切られます。コマンドには空白文字を含め、複数の行にまたがれます。

一般的に使用されるコマンド

サブトピック

- Optionコマンド
- Login / Logoutコマンド
- Copyコマンド
- Createコマンド
- Create Associationコマンド
- Delete Associationコマンド
- Deleteコマンド
- Detach Dimensionコマンド
- Excludeコマンド
- Executeコマンド
- Includeコマンド
- Insertコマンド
- 移動コマンド
- Renameコマンド
- Removeコマンド
- Share Dimensionコマンド
- Updateコマンド

次の項では、一般的に使用されるコマンドの例を示します。

Optionコマンド

Optionコマンドを使用すると、スクリプトの実行中にコマンド・ライン・オプションを動的に変更できます。変更可能なオプションは次のとおりです:

```
StopOnError
option StopOnError = true;
```

```
EchoComments
```

```
option EchoComments = true;
```

```
LogCommands  
option LogCommands = true;
```

Login / Logoutコマンド

サブトピック

- [login](#)
- [Logout](#)
- [Quit](#)
- [Exit](#)

login

Performance Management Architectにログインします。

スクリプトを使用してログインする場合は、次のように入力します:

```
Login admin,password;
```

コマンド・ラインを使用してログインする場合は、次のように入力します:

```
Login;
```

暗号化されたファイルにパスワードを保存して、この暗号化されたファイルを使用してログインできます。パスワードを暗号化するには、次の2つの方法のいずれかを使用できます:

- コマンド・ライン引数を使用してパスワードを暗号化するには:

```
..\EPMABatchClient>epma-batch-client.bat -Ac:/some.txt -Xpassword
```

ここで:

○-Xは暗号化する必要があるパスワードを指定します。

○-A (オプション)は、暗号化パスワードを保存するパス/ファイル名を指定します。このパラメータを使用しない場合、password.txtという名前のファイルがバッチ・クライアント・ユーティリティが存在する場所に作成されます。このパラメータを使用する場合、これらのパラメータが渡される順序は事前に定義されているため、-Xの前に指定する必要があります。

- スクリプト・ファイルを使用してパスワードを暗号化するには:

```
Encrypt <password>,<path>;
```

ここで:

○<password>は暗号化する必要があるパスワードを示します。

○<path> (オプション)は、暗号化パスワードが保存されるパス/ファイルを示します。パラメータが渡される順序は事前に定義されているため、パラメータの順序は次の例に示すようにする必要があります。

暗号化されたパスワード・ファイルを使用してログインするには、次の2つのオプションがあります:

- コマンド・ライン引数を使用してログインするには:

```
..\EPMABatchClient>epma-batch-client.bat -CplanDeploy.txt -Uadmin-Fc:\pass.txt
```

ここで:

○-Fは、暗号化パスワードを含むファイルのパスを指定します。パラメータが渡される順序は事前に定義されているため、パラメータの順序は次の例に示すようにする必要があります。

○-Pを使用してパスワードを指定する必要はありません。

- スクリプト・ファイルを使用してログインするには:

```
login <username>, F<path of file containing encrypted password>;
```

パスワードを直接指定するかわりに、“-F<path>”を使用して、パスワードは暗号化されたファイルから取得される必要があることを指定できます。

プロキシ経由でのログイン

SETコマンドを使用し、次の変数を定義して、プロキシ経由でログインできます。

```
set ProxyHost='http://localhost/myProxy';  
set ProxyPort='8080';  
set ProxyUsername='myProxyUser';  
set ProxyPassword='myProxyPassword';
```

ProxyHost - プロキシ・ホスト名。

ProxyPort - プロキシ・ポート番号。

ProxyUsername - 認証するユーザー名。

ProxyPassword - ユーザー・パスワード。

Oracle Access Managerなどのシングル・サインオン(SSO)を使用したログイン

シングル・サインオンを使用したログインには2つのオプションがあります:

- 基本/ダイジェスト認証

基本/ダイジェスト認証の場合は、次のようにSSOTypeをBasicに設定します:

```
set SSOType='Basic';
```

- フォーム・ベース認証

フォーム・ベース認証の場合は、次のようにSSOTypeをFormに設定します:

```
set SSOType='Form';
```

フォーム・ベース認証を使用している場合は、ユーザー名とパスワードをポストするフィールド名を次のように定義する必要があります:

```
set SSOFormUsernameField='userid';  
set SSOFormPasswordField='password';
```



注:

デフォルトのフィールド名は'userid'および'password'です。

Logout

Performance Management Architectからログアウトします。

```
Logout;
```

Quit

バッチ・クライアントを閉じます。

```
Quit;
```

Exit

バッチ・クライアントを閉じます。

```
Exit;
```

Copyコマンド

サブトピック

- [Application](#)
- [Dimension](#)

Application

Copy Applicationコマンドは、アプリケーション・ライブラリで使用可能な"Duplicate As New"コマンドと同じです。

```
Copy Application
Properties ( ApplicationName, CopyApplicationToName, ApplicationDescription,
ApplicationType)
Values( 'Comma', 'CommaCopy', 'Copied App Desc', 'Consolidation');
```

ApplicationName - 既存のアプリケーションの名前です。

CopyApplicationToName - 新規複製されたアプリケーションの名前です。

ApplicationDescription - アプリケーションの説明です。

ApplicationType - アプリケーションのタイプです。有効な値は次のとおりです: 汎用、連結、Planning、Profitability-標準、Profitability-詳細、Essbase ASOまたはEssbase BSO。

Dimension

共有ライブラリ内、アプリケーション内または共有ライブラリとアプリケーションの間でディメンションをコピーする方法を提供します。あるアプリケーションから別のアプリケーションにディメンションを直接コピーできません。

```
Copy Dimension
Properties(ApplicationName, DimensionName, TargetDimensionName,
TargetDimensionDescription,
destApplicationName)
```

```
Values('#Shared', 'Scenario', 'CopyScenario', 'Copy of Scenario Dim', Comma');
```

ApplicationName - 既存のアプリケーションの名前です。

DimensionName - 既存のディメンションの名前です。

TargetDimensionName - ターゲット・ディメンションの名前です。

TargetDimensionDescription - ターゲット・ディメンションの説明です。

destApplicationName - 宛先アプリケーションの名前です。

Createコマンド

サブトピック

- [Application](#)
- [Dimension](#)
- [Member](#)

Application

指定された名前で空のアプリケーションを新規作成します。

```
Create Application  
Properties(ApplicationName, ApplicationDescription, ApplicationType)  
Values('Comma', 'Description for Comma', 'Consolidation');
```

ApplicationName - アプリケーションの有効な名前を含む文字列です。

ApplicationDescription - アプリケーションの有効な名前を含む文字列です。

ApplicationType - 次の値をサポートします:

- 汎用
- 連結
- Planning
- Enterprise Analytics

Essbase ASOアプリケーションの場合

- Essbase Analytics

Essbase BSOアプリケーションの場合

- Profitability
- Profitability-詳細

Dimension

指定されたアプリケーションまたは共有ライブラリに空のディメンションを新規作成します。

```
Create Dimension
Pproperties(ApplicationName, DimensionName, DimensionDescription,
DimensionType)
Values('Comma', 'Test_Account', 'Test Account', 'Account');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリにディメンションを作成する場合は、#Sharedを使用します。

DimensionName - ディメンションの有効な名前です。

DimensionType - ディメンション・タイプは、次のいずれかを使用できます:

- 勘定科目
- 別名
- AllocationType
- 属性
- ConsolidationMethod
- 国
- 通貨
- エンティティ
- 汎用
- ICP
- メジャー
- MeasuresDetailed
- シナリオ
- SecurityClass
- スマート・リスト
- 時間
- UDA
- 値
- バージョン
- 表示
- 年

Member

指定されたディメンションに新規メンバーを作成します。

```
Create Member
Properties(ApplicationName, DimensionName, ParentName, MemberName, MemberDescription)
Values('Comma', 'Member_Dim', '#root', 'TestMember1', 'Description for TestMember1');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリにディメンションを作成する場合は、#Sharedを使用します。

DimensionName - 既存のディメンションの名前です。

ParentName - 新たに作成したメンバーを挿入する親の名前です。ツリーのトップ・レベルに新規メンバーを追加する場合は、#Rootを使用します。

MemberName - 新規メンバーの有効な名前です。

MemberDescription - 新規メンバーの説明です。

Create Associationコマンド

2つのディメンションの間に関連付けを作成します。基本ディメンションを共有する場合は、関連ディメンションを共有ディメンションにする必要があります。

```
Create Association
Properties(ApplicationName, DimensionName, AssociatedDimensionName,
Propertyname,PropertyDescription)
Values('Comma', 'Scenario', 'AttribDim', 'AttribProp', 'Attrib Prop Desc');
```

ApplicationName - 既存のアプリケーションの名前です。

DimensionName - 既存のディメンションの名前です。

AssociatedDimensionName - 関連付けるディメンションの名前です。

Propertyname - 更新するプロパティの名前です。必要なプロパティをいくつでも記載できますが、プロパティには有効な値を指定する必要があります。

PropertyDescription - コメントまたはプロパティの説明を入力できるオプションのパラメータ。

Delete Associationコマンド

既存のディメンションの関連付けを削除します。

```
Delete Association
Properties(ApplicationName, DimensionName, PropertyName)
Values('Comma', 'Scenario', 'SecurityClass');
```

ApplicationName - 既存のアプリケーションの名前です。

DimensionName - 既存のディメンションの名前です。

PropertyName - 更新するプロパティの名前です。必要なプロパティをいくつでも記載できますが、プロパティには有効な値を指定する必要があります。

Deleteコマンド

サブトピック

- [Application](#)
- [Association](#)
- [Dimension](#)
- [Member](#)

Application

指定されたアプリケーションを削除します。

```
Delete Application
Properties(ApplicationName, WaitForCompletion)
Values('Comma');
```

ApplicationName - 既存のアプリケーションの名前です。

WaitForCompletion - TRUEに設定すると、バッチ・クライアントはジョブが終了するまで待機します。FALSEに設定すると、バッチ・クライアントはジョブを送信して続行します。デフォルト値はFALSEです。使用可能な値:

- TRUE
- FALSE

Association

既存のディメンションの関連付けを削除します。

```
Delete Association
Properties(ApplicationName, DimensionName, PropertyName)
Values('Comma', 'Scenario', 'SecurityClass');
```

ApplicationName - 既存のアプリケーションの名前です。

DimensionName - 既存のディメンションの名前です。

PropertyName - 更新するプロパティの名前です。必要なプロパティをいくつでも記載できますが、プロパティには有効な値を指定する必要があります。

Dimension

指定されたディメンションを削除します。

```
Delete Dimension
Properties(ApplicationName, DimensionName)
Values('Comma', 'C_Scenario');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリのディメンションを削除する場合は、#Sharedを使用します。

DimensionName - 既存のディメンションの名前です。

Member

指定されたメンバーを削除します。メンバーのすべての子孫を削除することもできます。

```
Delete Member
Properties(ApplicationName, DimensionName, ParentName, MemberName,
DeleteAllDescendants)
Values('Comma', 'C_Account', '#root', 'TestMember1', 'false');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリのディメンション・メンバーを削除する場合は、#Sharedを使用します。

DimensionName - 既存のディメンションの名前です。

ParentName - メンバーを削除する親の名前です。ツリーのトップ・レベルのメンバーを削除する場合は、#Rootを使用します。

MemberName - 削除するメンバーの名前です。

DeleteAllDescendants - 削除されるメンバーの下の子孫も削除するかどうかを指定します。使用可能な値:

- TRUE
- FALSE

Detach Dimensionコマンド

アプリケーション内の共有ディメンションをローカル・ディメンションに変換します。

```
Detach Dimension
Properties(ApplicationName, DimensionName, RetainFilteredStructure,
RetainPropertyOverrides,waitForCompletion)
Values('Comma', 'Period', 'true', 'true', 'true');
```

ApplicationName - 既存のアプリケーションの名前です。

DimensionName - 既存のディメンションの名前です。

RetainFilteredStructure - TRUEに設定すると、ディメンションの現在のビューが維持され、除外されたメンバーまたは他のフィルタ・メンバーはディメンションのローカル・コピーに表示されません。FALSEに設定すると、ディメンションのすべてのメンバーがディメンションのローカル・コピーに表示されます。使用可能な値:

- TRUE
- FALSE

RetainPropertyOverrides - TRUEに設定すると、すべてのプロパティの上書きが保持されます。それ以外の場合は、ディメンションの共有バージョンからの値が使用されます。使用可能な値:

- TRUE
- FALSE

WaitForCompletion - TRUEに設定すると、バッチ・クライアントはジョブが終了するまで待機します。FALSEに設定すると、バッチ・クライアントはジョブを送信して続行します。使用可能な値:

- TRUE
- FALSE

Excludeコマンド

Member

共有ディメンションからメンバーを除外します。

```
Exclude Member
Properties(ApplicationName, DimensionName, ParentName, MemberName)
Values('Comma', 'Period', '#root', 'P1');
```

ApplicationName - 既存のアプリケーションの名前です。ディメンションを除外するための目標として#Sharedを使用できません。

DimensionName - 共有ライブラリ内の既存のディメンションの名前で、除外するディメンションです。

ParentName - 除外するメンバーの親の名前です。

MemberName - 除外するメンバーの名前です。

Executeコマンド

サブトピック

- [インポート\(アプリケーションまたは共有ライブラリへ\)](#)

- [ディメンションの同期\(共有ライブラリとの\)](#)
- [デプロイメント](#)
- [再デプロイメント](#)
- [データの同期](#)
- [検証](#)

EXECUTEコマンドを使用して、ジョブを実行できます。

インポート(アプリケーションまたは共有ライブラリへ)

既存のインポート・プロファイルを実行します。

フラット・ファイルの場合、構文は次のとおりです:

```
Execute Import
Parameters(importtype, profilename, filename, waitforcompletion)Values('flatfile',
'Comma', '.\AppFiles\CommaApp.ads', 'true');
```

インタフェース・テーブルの場合、構文は次のとおりです:

```
Execute Import
Parameters(importtype, profilename, waitforcompletion)
Values('InterfaceTables', 'intapp', 'true');
```

Data Relationship Managementの場合、構文は次のとおりです:

```
Execute Import
Parameters(importtype, profilename, waitforcompletion)
Values('DRM', 'DRM Profile', 'true');
```

ImportType - 実行するインポートの種類です。使用可能な値:

- FlatFile
- InterfaceTables
- DRM (Data Relationship Management)

ProfileName - 既存のインポート・プロファイルの名前です。

FileName - フラット・ファイルのタイプをインポートする場合のインポートするフラット・ファイルの名前です。



注:

インタフェース・テーブルまたはData Relationship Managementのファイルをインポートする場合は、FileNameパラメータを使用しません。これは、フラット・ファイルをインポートする場合にのみ使用されます。

WaitForCompletion - TRUEに設定すると、バッチ・クライアントはジョブが終了するまで待機します。FALSEに設定すると、バッチ・クライアントはジョブを送信して続行します。使用可能な値:

- TRUE
- FALSE

ディメンションの同期(共有ライブラリとの)

共有ライブラリとディメンションの同期をやり取りします。

```
Execute DimensionSynchronization
Parameters(SourceApplicationName, DestApplicationName, DestDimensionName, ReplaceMode)
Values('appName', 'DestAppName', 'DimName', 'true');
```

ApplicationName - 既存のアプリケーションの名前です。ディメンションの同期のアプリケーション名として#Sharedを使用できません。

DimensionName - アプリケーションの既存のディメンションの名前です。

SharedDimensionName - 共有ライブラリ内の既存のディメンションの名前です。

SyncToApp - TRUE値を指定すると、共有ディメンションはアプリケーションと同期され、FALSE値を指定すると、アプリケーションが共有ライブラリに同期されます。使用可能な値:

- TRUE
- FALSE

ReplaceMode - TRUE値を指定すると、ディメンションの同期は置換モードを使用し、FALSE値を指定すると、マージ・モードを使用します。使用可能な値:

- TRUE
- FALSE

デプロイメント

アプリケーションを指定された製品にデプロイします。

```
Execute Deploy
Parameters(ApplicationName, InstanceName, ApplicationServer, HubProject, ClearAll,
CheckIntegrity, waitForcompletion, deployOption, Notes)
Values('AppName', 'HFM931', 'localhost:1423', 'Default Application Group', 'false',
'false', 'true', 'true', 'AppView', 'deploy comments sample');
```

ApplicationName - 既存のアプリケーションの名前です。デプロイメントのアプリケーション名として#Sharedを使用できません。

InstanceName - デプロイ先のインスタンスの名前です。

ApplicationServer - デプロイ先のアプリケーション・サーバーの名前です。

HubProject - デプロイ済アプリケーションを追加するOracle Hyperion Shared Servicesプロジェクトです。

ClearAll - アプリケーション・データベース内のすべてのディメンション・メンバー、および対応するデータ、仕訳、内部取引を削除します。(連結アプリケーションでのみ使用できます。)使用可能な値:

- TRUE
- FALSE

CheckIntegrity - メタデータとデータをチェックして、整合性を確認します。(連結アプリケーションでのみ使用できません。)使用可能な値:

- TRUE
- FALSE

deployOption - 実行するデプロイメントのタイプを指定します。デフォルト値はAppViewで、アプリケーションのみデプロイします。

WaitForCompletion - TRUEに設定すると、バッチ・クライアントはジョブが終了するまで待機します。FALSEに設定すると、バッチ・クライアントはジョブを送信して続行します。使用可能な値:

- TRUE
- FALSE

Planningアプリケーション向け:

datasourceName - 指定した名前で作成したデータ・ソースを作成します。この値は文字列です。

CreateOutline - 初めてアプリケーションをデプロイする場合は、Essbaseアウトラインを作成します。使用可能な値:

- TRUE
- FALSE

RefreshOutline - アプリケーションの構造を変更した後にアプリケーション・データベースをリフレッシュします。使用可能な値:

- TRUE
- FALSE

CreateSecurityFilters - 暗号化されたデータ・ファイルにアクセス権を保管します(Essbase.sec)。使用可能な値:

- TRUE
- FALSE

SharedMembersSecurityFilters - アクセス権を共有メンバーに適用します。使用可能な値:

- TRUE
- FALSE

ValidateSecurityFilterLimit - 1行当たり64KBのOracle Essbaseセキュリティ・フィルタの限界を超えるセキュリティ・フィルタを識別します。これによって、Oracle Essbaseのセキュリティ・フィルタを構築する前に、フィルタ・サイズを検証してサイズの限界を超えないようにできます。使用可能な値:

- TRUE
- FALSE

注意 - オプションで、デプロイメントに関して追加するコメントや注意事項を保管します。

再デプロイメント

アプリケーションを製品サーバーに再デプロイします。

```
Execute Redeploy
Parameters(ApplicationName, InstanceName, HubProject, ClearAll, CheckIntegrity,
waitforcompletion, deployOption, escapeValidateRules, Notes)
Values('AppName', 'HubProj', 'false', 'false', 'true', 'true', 'AppView', 'true',
'Comments on redeployment');
```

有効値については、[31ページのデプロイメント](#)の例を参照してください。

アプリケーションを再デプロイメントする場合、`deployOption`で指定できる値は、Oracle Hyperion PlanningおよびOracle Hyperion Financial Managementアプリケーションをデプロイメントする場合の値とは異なります。使用可能な値:

- ApplicationName - 既存のアプリケーションの名前です。デプロイメントのアプリケーション名として#Sharedを使用できません。

InstanceName - デプロイ先のインスタンスの名前です。

HubProject - デプロイ済アプリケーションを追加するOracle Hyperion Shared Servicesプロジェクトです。

ClearAll - アプリケーション・データベース内のすべてのディメンション・メンバー、および対応するデータ、仕訳、内部取引を削除します。(連結アプリケーションでのみ使用できます。)使用可能な値:

- TRUE
- FALSE

CheckIntegrity - メタデータとデータをチェックして、整合性を確認します。(連結アプリケーションでのみ使用できます。)使用可能な値:

- TRUE
- FALSE

WaitForCompletion - TRUEに設定すると、バッチ・クライアントはジョブが終了するまで待機します。FALSEに設定すると、バッチ・クライアントはジョブを送信して続行します。使用可能な値:

○TRUE

○FALSE

• deployOption - 実行するデプロイメントのタイプを指定します。デフォルト値はAppViewで、アプリケーションのみデプロイします。

• escapeValidateRules—Trueに設定すると、検証せずにルールをデプロイします。Falseに設定すると、検証した後にルールをデプロイします。使用可能な値:

○TRUE

○FALSE

• 注意 - オプションで、デプロイメントに関して追加するコメントや注意事項を保管します。

• AppView - アプリケーションのみデプロイします。(この設定はデフォルトです。)

• Rules - 計算ルールをデプロイします。

• All - 計算ルールとアプリケーションをデプロイします。

他のすべてのアプリケーション・タイプでは、AppViewのみ指定できます。

データの同期

既存のデータ同期プロファイルを実行します。

```
Execute DataSynchronization
Parameters(DataSynchronizationName, DataTransformationOperator,
DataTransformationValue, FileName, ValidateOnly, WaitForCompletion)
Values('CommaSync3', '*', '1.2345', '', 'false', 'true');
```

DataSynchronizationName - 実行するデータ同期プロファイルの名前です。

DataTransformationOperator - 指定できる値は次のとおりです:

- なし
- '*' (乗算)
- '/' (除算)
- '+' (加算)
- '-' (減算)

DataTransformationValue - データ値を変更するためにDataTransformationOperatorと連携して使用する値です。

FileName - 同期で、同期のソースに外部のソース・ファイルを使用する場合、外部ソース・ファイルの場所です。場所は、Webサーバーがアクセスできる場所をURLで示す必要があります。ファイル・パスは、ファイルがリモート・コンピュータ上にある場合は「file:/// <machine name> / <folder name> / <file name>」、ファイルがデータの同期サービスが実行されているコンピュータ上にある場合は「file:/// C: / CvgCompLd_JobTtl.txt」である必要があります。

ValidateOnly - データの同期を実行せずに検証します。

WaitForCompletion - TRUEに設定すると、バッチ・クライアントはジョブが終了するまで待機します。FALSEに設定すると、バッチ・クライアントはジョブを送信して続行します。使用可能な値:

- TRUE
- FALSE

これらのパラメータの他に、データ・ロード・オプション・パラメータもあります。データ・ロード・オプション・パラメータは次のとおりです:

- 動的
- レジストリから取得
- パラメータ名がアプリケーションのタイプに基づいて異なる
- 他のパラメータと異なり、すべてのパラメータが大文字と小文字を区別する
- すべての値が大文字と小文字を区別する

Planningの場合、パラメータは次の1つです: `dataSyncLoadOptionHpMode`。使用可能な値:

- ADD
- SUBTRACT
- OVERWRITE

Essbase ASOの場合、パラメータは次の1つです: `dataSyncLoadOptionAsoMode`。使用可能な値:

- ADD
- SUBTRACT
- OVERWRITE

Essbase BSOの場合、パラメータは次の1つです: `dataSyncLoadOptionBsoMode`。使用可能な値:

- ADD
- SUBTRACT
- OVERWRITE

Planning、Essbase ASOおよびEssbase BSOアプリケーションの場合、デフォルト値はOVERWRITEです。

連結の場合、次の2つのパラメータがあります:

- `dataSyncLoadOptionHfmMode` - 使用可能な値は次のとおりです:
 - merge
 - replace
 - accumulate

デフォルト値はmergeです。

- `dataSyncLoadOptionHfmAccummulateInFile` - 使用可能な値は次のとおりです:
 - TRUE
 - FALSE

デフォルトはfalseです。



注:

ProfitabilityアプリケーションおよびProfitability-詳細アプリケーションでは、データ・ロード・オプションはサポートされません。

検証

アプリケーションの検証を実行します。StopOnErrorオプションがTRUEに設定されているかぎり、スクリプトは検証が失敗した場合に終了します。

```
Execute Validate
Parameters(ApplicationName, ValidateType)
Values('Comma1', 'All');
```

ApplicationName - 既存のアプリケーションの名前です。

ValidateType - 実行する検証のタイプです。たとえば、アプリケーションのみの検証、ビジネス・ルールの検証またはすべて(アプリケーションおよびルール)の検証があります。使用可能な値:

- AppView
- Rules
- All

Includeコマンド

Dimension

共有ライブラリから指定したアプリケーションに既存のディメンションを追加します。ディメンションは共有ディメンションとして追加することも、ローカル・ディメンションとしてアプリケーションにコピーすることもできます。

```
Include Dimension
Properties(DimensionName, IncludeAsShared)
Values('C_Alias', 'true');
```

ApplicationName - 既存のアプリケーションの名前です。ディメンションを含めるための目標として#Sharedを使用できません。

DimensionName - 共有ライブラリ内の既存のディメンションの名前で、インクルードするディメンションです。

IncludeAsShared - 共有ライブラリのソース・ディメンションとのリンクを維持する共有ディメンションとしてディメンションをインクルードする場合はTRUE値を使用します。共有ライブラリのディメンションとは別にディメンションのコピーをインクルードする場合はFALSE値を使用します。使用可能な値:

- TRUE
- FALSE

Insertコマンド

Insert Member

メンバーのコピーを共有メンバーとして挿入します。Insert Memberコマンドを使用できるのは、ローカル・ディメンションおよび共有ライブラリ・ディメンションのみです。アプリケーション内の共有ディメンションではメンバーを挿入できません。

```
Insert Member Properties(DimensionName, ParentName, InsertMemberName,  
MemberToInsertName)  
Values('Account', 'Par1', 'Mem1', 'Mem2');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリのディメンション・メンバーを操作する場合は、#Sharedを使用します。

DimensionName - 既存のディメンションの名前です。

ParentName - メンバーの挿入先の親の名前です。

InsertMemberName - メンバーの挿入先のメンバーの名前です。

MemberToInsertName - 挿入する共有メンバーの名前です。

移動コマンド

メンバーの移動

ディメンション構造内のある場所から別の場所にメンバーを移動します。Move Memberコマンドを使用できるのは、ローカル・ディメンションおよび共有ライブラリ・ディメンションのみです。アプリケーション内の共有ディメンションのメンバーは移動できません。

```
Move Member  
Properties(ApplicationName, DimensionName, FromParentName,  
MemberName, ToParentName, InsertAfterMemberName)  
Values('SampleApp', 'Period', '#root', 'r1', 'P2', '#none');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリのディメンション・メンバーを操作する場合は、#Sharedを使用します。

DimensionName - 既存のディメンションの名前です。

FromParentName - 移動するメンバーの親の名前です。

MemberName - 移動するメンバーの名前です。

ToParentName - メンバーの移動先の親の名前です。

InsertAfterMember - メンバーの挿入先となるToParentNameに属する子を示します。メンバーを子として挿入する必要があることを示す特別な値'#none'を設定できます。これにより、挿入されるメンバーとその後のすべてのメンバーのソート順が影響を受けます。

Renameコマンド

メンバー名の変更

メンバーおよびそのメンバーのすべての共有コピーを名前変更します。Rename Memberコマンドを使用できるのは、ローカル・ディメンションおよび共有ライブラリ・ディメンションのみです。アプリケーション内の共有ディメンションのメンバーは名前変更できません。

```
Rename Member
Properties(ApplicationName, DimensionName, ParentName, MemberName, NewMemberName)
Values('Comma', 'Account', '#root', 'M2', 'M2REN');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリのディメンション・メンバーを操作する場合は、#Sharedを使用します。

DimensionName - 既存のディメンションの名前です。

ParentName - 名前変更するメンバーの親の名前です。

MemberName - 名前変更する既存のメンバーの名前です。

NewMemberName - メンバーの新しい名前です。

Removeコマンド

サブトピック

- [Remove Member \(アプリケーションまたは共有ライブラリ\)](#)
- [Remove Dimension](#)

Remove Member (アプリケーションまたは共有ライブラリ)

指定したディメンションからメンバーを除去しますが削除はしません。Remove Memberコマンドを使用できるのは、アプリケーション内の共有ディメンションを除去する場合のみです。

```
Remove Member
Properties(DimensionName, ParentName, MemberName)
Values('Account', 'Mem1', 'Mem2');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリのディメンション・メンバーを操作する場合は、#Sharedを使用します。

DimensionName - 既存のディメンションの名前です。

ParentName - 除去するメンバーの親の名前です。

MemberName - 除去するメンバーの名前です。

Remove Dimension

アプリケーションから共有ディメンションを除去します。

```
Remove Dimension
Properties(ApplicationName, DimensionName, Force)
Values('Comma', 'Period', 'true');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリのディメンション・メンバーを操作する場合は、#Sharedを使用します。

DimensionName - 既存のディメンションの名前です。

Force - TRUEに設定した場合、除去対象のディメンションがアプリケーション内の他のディメンションに関連付けられていても除去されます。FALSEに設定した場合、除去対象のディメンションがアプリケーション内の他のディメンションに関連付けられているとディメンションを除去できません。使用可能な値:

- TRUE
- FALSE

Share Dimensionコマンド

ローカル・ディメンションを新しい共有ディメンションに変換するか、既存のディメンションとマージします。

```
Share Dimension
Properties(ApplicationName, DimensionName, ShareAsNew, SharedDimensionName,
MergeAsShared,
WaitForCompletion)
Values('Comma', 'Entity', 'false', 'ShareEntity', 'true', 'true');
```

ApplicationName - 既存のアプリケーションの名前です。

DimensionName - 既存のディメンションの名前です。

ShareAsNew - TRUEに設定すると、SharedDimensionNameプロパティおよびMergeAsSharedプロパティが無視されます。これは、ディメンションを新規として共有する場合に、これらのプロパティが適用されないためです。
使用可能な値:

- TRUE
- FALSE

SharedDimensionName - FALSEに設定する場合は、共有ライブラリで共有するディメンションの名前を指定する必要があります。使用可能な値:

- TRUE
- FALSE

MergeAsShared - TRUEに設定すると、共有対象のディメンションがターゲット・ディメンションとマージされます。FALSEに設定すると、共有対象のディメンションによってターゲット・ディメンションが置換されます。使用可能な値:

- TRUE
- FALSE

WaitForCompletion - TRUEに設定すると、バッチ・クライアントはジョブが終了するまで待機します。FALSEに設定すると、バッチ・クライアントはジョブを送信して続行します。使用可能な値:

- TRUE
- FALSE

Updateコマンド

サブトピック

- [Application](#)
- [Dimension](#)
- [Member](#)
- [ディメンションの関連付け](#)

UPDATEスクリプト・コマンドを使用して、アプリケーション、ディメンションまたはメンバーのプロパティ値を変更する場合は、Performance Management Architectのプロパティ・グリッドに表示されるプロパティ・ラベルではなくプロパティ名を使用する必要があります。プロパティ・ラベルとプロパティ名については、『Oracle Hyperion Enterprise Performance Management Architect管理者ガイド』の付録に説明があります。次に、スクリプトの例を示します:

```
Update Member
Properties(DimensionName, ParentName, MemberName, AggregationWeight, NumDecimalPlaces)
```



```
Values('ScenarioDim', '#root', 'Member1', '3', '2');
```

この例では、DimensionName、ParentNameおよびMemberNameは、すべて標準のスクリプト・アイテムですが、AggregationWeightとNumDecimalPlacesは、Oracle Hyperion EPM Architectのメンバー・レベルのプロパティです。

Application

指定されたアプリケーションの1つ以上のプロパティを更新します。

```
Update Application
Properties(ApplicationName, ValidationAccount)
Values('Comma', 'Validation');
```

ApplicationName - 既存のアプリケーションの名前です。#Sharedのプロパティ値は更新できません。

PropertyName - 更新するプロパティの名前です。必要なプロパティをいくつでも記載できますが、プロパティには有効な値を指定する必要があります。

Dimension

指定されたディメンションの1つ以上のプロパティを更新します。

```
Update Dimension
Properties(ApplicationName, DimensionName, PropertyName)
Values('Comma', 'C_Entity', 'Validation');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリのディメンションを更新する場合は、#Sharedを使用します。

DimensionName - 既存のディメンションの名前です。

PropertyName - 更新するプロパティの名前です。必要なプロパティをいくつでも記載できますが、プロパティには有効な値を指定する必要があります。

Member

指定されたメンバーの1つ以上のプロパティを更新します。

```
Update Member
Properties(ApplicationName, DimensionName, ParentName, MemberName, ValidationAccount)
Values('Comma', 'C_Entity', 'E1', 'E1-1', 'Validation');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリのディメンション・メンバーを更新する場合は、#Sharedを使用します。

DimensionName - 既存のディメンションの名前です。

ParentName - 更新するメンバーの親の名前です。

MemberName - 更新するメンバーの名前です。

PropertyName - 更新するプロパティの名前です。必要なプロパティをいくつでも記載できますが、プロパティには有効な値を指定する必要があります。

ディメンションの関連付け

アプリケーションのタイプおよびアプリケーションに含まれるディメンションに基づいたすべての標準ディメンションの関連付けをアクティブ化します。

```
Update Dimensionassociation
Properties(activateallforapplication) Values('true');
```

ApplicationName - 既存のアプリケーションの名前です。#Sharedのプロパティ値は更新できません。

ActivateAllForApplication - 指定したアプリケーションに対してすべての関連付けをアクティブ化するかどうかを指定します。使用可能な値:

- TRUE
- FALSE