# Contents

## Preface

<table>
<thead>
<tr>
<th>Audience</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation Accessibility</td>
<td>x</td>
</tr>
<tr>
<td>Related Resources</td>
<td>x</td>
</tr>
<tr>
<td>Conventions</td>
<td>xi</td>
</tr>
</tbody>
</table>

## 1 Getting Started with Oracle Analytics Cloud - Essbase

| Overview of Oracle Analytics Cloud - Essbase | 1-1 |
| Accessing Oracle Analytics Cloud - Essbase | 1-2 |
| Using Service Credentials to Access the Cloud Service | 1-3 |
| Accessing the Modern User Interface | 1-3 |
| Accessing the Cloud Service from Smart View | 1-4 |
| Cloud Service and Smart View Client URLs | 1-4 |
| Managing Client Software | 1-5 |
| Managing Password and Password Challenge Questions | 1-5 |
| Oracle Learning Library | 1-6 |

## 2 Top Tasks for Oracle Analytics Cloud - Essbase

<p>| Top Tasks Prerequisites | 2-1 |
| Creating Users and Assigning Roles | 2-2 |
| Importing and Exporting Bulk Users | 2-2 |
| Understanding Application Workbooks | 2-3 |
| Creating an Application in the Cloud Service and Provisioning a User to Access and Query the Cube | 2-5 |
| Analyzing an Application in Smart View | 2-7 |
| Modifying an Outline in the Cloud Service | 2-12 |
| Continuing Analysis in Smart View | 2-13 |
| Creating an Application and Cube in Cube Designer | 2-16 |
| Analyzing Data and Performing an Incremental Update in Cube Designer | 2-18 |
| Transforming Tabular Data into a Cube | 2-20 |
| Exporting and Modifying Metadata and Data | 2-21 |</p>
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Managing Users and Roles</td>
<td>3-1</td>
</tr>
<tr>
<td></td>
<td>About User and Role Management</td>
<td>3-1</td>
</tr>
<tr>
<td></td>
<td>Predefined Roles</td>
<td>3-1</td>
</tr>
<tr>
<td></td>
<td>Setting Up Security</td>
<td>3-2</td>
</tr>
<tr>
<td></td>
<td>Provisioning Users</td>
<td>3-3</td>
</tr>
<tr>
<td></td>
<td>Provisioning Roles</td>
<td>3-6</td>
</tr>
<tr>
<td>4</td>
<td>Designing and Creating Cubes Using Application Workbooks</td>
<td>4-1</td>
</tr>
<tr>
<td></td>
<td>About Application Workbooks</td>
<td>4-1</td>
</tr>
<tr>
<td></td>
<td>Downloading a Sample Application Workbook</td>
<td>4-2</td>
</tr>
<tr>
<td></td>
<td>Creating a Cube from an Application Workbook</td>
<td>4-2</td>
</tr>
<tr>
<td></td>
<td>Creating a Cube from Tabular Data</td>
<td>4-3</td>
</tr>
<tr>
<td></td>
<td>Connecting to a Cube in Smart View</td>
<td>4-5</td>
</tr>
<tr>
<td>5</td>
<td>Creating and Managing Cube Outlines</td>
<td>5-1</td>
</tr>
<tr>
<td></td>
<td>About Cube Outlines</td>
<td>5-1</td>
</tr>
<tr>
<td></td>
<td>Viewing and Editing Outline Properties</td>
<td>5-1</td>
</tr>
<tr>
<td></td>
<td>Adding Dimensions and Members to Outlines</td>
<td>5-2</td>
</tr>
<tr>
<td></td>
<td>Adding Dimensions to Outlines Manually</td>
<td>5-2</td>
</tr>
<tr>
<td></td>
<td>Adding Members to Outlines Manually</td>
<td>5-3</td>
</tr>
<tr>
<td></td>
<td>Building Dimensions Using a Rules File</td>
<td>5-4</td>
</tr>
<tr>
<td></td>
<td>Viewing Outlines</td>
<td>5-5</td>
</tr>
<tr>
<td></td>
<td>Hiding Outline Columns</td>
<td>5-6</td>
</tr>
<tr>
<td></td>
<td>Reordering Outline Columns</td>
<td>5-6</td>
</tr>
<tr>
<td></td>
<td>About Duplicate Member Names</td>
<td>5-6</td>
</tr>
<tr>
<td></td>
<td>Enabling Duplicate Member Names</td>
<td>5-7</td>
</tr>
<tr>
<td></td>
<td>Editing Member Properties</td>
<td>5-7</td>
</tr>
<tr>
<td></td>
<td>Setting Dimension and Member Properties</td>
<td>5-8</td>
</tr>
<tr>
<td></td>
<td>Setting Information Properties</td>
<td>5-9</td>
</tr>
<tr>
<td></td>
<td>Creating Aliases</td>
<td>5-12</td>
</tr>
<tr>
<td></td>
<td>Setting Attribute Associations</td>
<td>5-13</td>
</tr>
<tr>
<td></td>
<td>Creating User-Defined Attributes</td>
<td>5-13</td>
</tr>
<tr>
<td></td>
<td>Creating Member Formulas</td>
<td>5-14</td>
</tr>
<tr>
<td></td>
<td>Naming Generations and Levels</td>
<td>5-15</td>
</tr>
<tr>
<td></td>
<td>Setting Advanced Cube Properties</td>
<td>5-15</td>
</tr>
<tr>
<td></td>
<td>Unlocking Objects</td>
<td>5-16</td>
</tr>
</tbody>
</table>
# Removing User Data Locks

- 5-16

# About Filters

- 5-17
  - Creating Filters
  - 5-17

# Working with Attributes

- 5-18

# Managing Outlines

- 5-19
  - Searching an Outline
  - 5-20
  - Managing Outline Search Results
  - 5-20

# Loading Data

- 6
  - About Loading Data
  - 6-1
  - Loading Data Using a Rules File
  - 6-2
  - Uploading Files to a Cube
  - 6-3
  - Managing Jobs
  - 6-4
  - Loading Dimensions and Data Using SQL
  - 6-5
    - Loading Dimensions Using SQL
    - 6-7
    - Loading Data Using SQL
    - 6-10
  - Loading Dimensions and Data by Streaming from a Remote Database
  - 6-12
  - Loading Data and Dimensions from Data Sync to Essbase
  - 6-14
    - Loading Dimensions from Data Sync to Essbase
    - 6-15
    - Loading Data from Data Sync to Essbase
    - 6-15

# Calculating Cubes

- 7
  - About Calculating Data
  - 7-1
  - Creating Calculation Scripts
  - 7-1
  - Executing Calculations
  - 7-2
  - Creating and Using Substitution Variables
  - 7-3
  - Setting Two-Pass Calculation Properties
  - 7-4
  - Tracing Calculations
  - 7-4

# Modeling Data in Private Scenarios

- 8
  - Understanding Scenarios
  - 8-1
  - Workflow for Scenarios
  - 8-3
  - Enabling Scenario Modeling
  - 8-4
    - Creating a Scenario-Enabled Cube
    - 8-5
    - Creating a Scenario-Enabled Sample Cube
    - 8-5
    - Enabling an Existing Cube for Scenario Management
    - 8-6
    - Creating Additional Sandbox Members
    - 8-6
  - Using Scenarios
  - 8-7
    - Creating a Scenario
    - 8-7
## Deleting Applications and Cubes in Cube Designer

9-21

### 10 Exporting Cubes to Application Workbooks

- Exporting a Cube to an Application Workbook 10-1
- Exporting On-Premises Cubes to Import to the Cloud 10-2
  - Downloading the Cube Export Utility 10-2
  - Exporting On-Premises Cubes to Application Workbooks 10-2
  - Reviewing Member Names before Importing an Application Workbook Created by the Cube Export Utility 10-4

### 11 Tracking Changes to Data

- Turning on Data Audit Trail and Viewing the Data Audit Trail 11-1
- Linking a Reporting Object to a Cell 11-2
- Exporting Logs to a Sheet 11-2
- Refresh the Audit Log 11-3

### 12 Connecting Cubes

- Understanding Partitions 12-1
- Understanding Location Aliases 12-1
- Defining a Reusable Connection 12-2
- Creating a Location Alias Based on a Defined Connection 12-2
- Creating a Replicated Partition 12-3

### 13 Migrating On-Premises and Cloud Service Applications

- Preparing to Migrate On-Premises Applications to the Cloud Service 13-1
- Application Migration and Use Cases 13-4
- Application Elements - Supported or Unsupported 13-5
- Migrating Cloud Service Applications 13-7

### 14 Configuring Cloud Service Resource Limits and Application Configuration Properties

- Modifying Cloud Service Resource Limits 14-1
- Setting Application-Level Configuration Properties 14-3
## Executing MaxL and MDX Statements

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running MaxL and MDX Jobs in the Cloud Service</td>
<td>15-1</td>
</tr>
<tr>
<td>About MaxL and MDX Jobs</td>
<td>15-1</td>
</tr>
<tr>
<td>Creating MaxL and MDX Scripts</td>
<td>15-2</td>
</tr>
<tr>
<td>Uploading MaxL and MDX Scripts</td>
<td>15-2</td>
</tr>
<tr>
<td>Running MaxL and MDX Scripts</td>
<td>15-2</td>
</tr>
<tr>
<td>Script: Executing MaxL and MDX from CLI</td>
<td>15-3</td>
</tr>
</tbody>
</table>

## Oracle Analytics Cloud - Essbase Command-Line Interface (CLI)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downloading and Using the Command-Line Interface</td>
<td>16-1</td>
</tr>
<tr>
<td>CLI Command Reference</td>
<td>16-1</td>
</tr>
<tr>
<td>Login/Logout: EssCS Connection</td>
<td>16-2</td>
</tr>
<tr>
<td>Calc: Running a Calculation Script</td>
<td>16-3</td>
</tr>
<tr>
<td>Clear: Removing Data from a Cube</td>
<td>16-3</td>
</tr>
<tr>
<td>Createlocalconnection: Saving a JDBC Connection</td>
<td>16-4</td>
</tr>
<tr>
<td>Dataload: Loading Data to a Cube</td>
<td>16-5</td>
</tr>
<tr>
<td>Deletefile: Removing Cube Files</td>
<td>16-6</td>
</tr>
<tr>
<td>Deploy: Creating a Cube from a Workbook</td>
<td>16-7</td>
</tr>
<tr>
<td>Dimbuild: Loading Dimensions to a Cube</td>
<td>16-7</td>
</tr>
<tr>
<td>Download: Getting Cube Files</td>
<td>16-8</td>
</tr>
<tr>
<td>Help: Displaying Command Syntax</td>
<td>16-9</td>
</tr>
<tr>
<td>LcmExport: Backing Up Cube Files</td>
<td>16-9</td>
</tr>
<tr>
<td>LcmImport: Restoring Cube Files</td>
<td>16-10</td>
</tr>
<tr>
<td>Listapp: Displaying Applications</td>
<td>16-10</td>
</tr>
<tr>
<td>Listdb: Displaying Cubes</td>
<td>16-11</td>
</tr>
<tr>
<td>Listfiles: Displaying Files</td>
<td>16-11</td>
</tr>
<tr>
<td>Listfilters: Viewing Security Filters</td>
<td>16-12</td>
</tr>
<tr>
<td>Listlocks: Viewing Locks</td>
<td>16-12</td>
</tr>
<tr>
<td>Listvariables: Displaying Substitution Variables</td>
<td>16-12</td>
</tr>
<tr>
<td>Script: Executing MaxL and MDX from CLI</td>
<td>16-13</td>
</tr>
<tr>
<td>Setpassword: Storing CLI Credentials</td>
<td>16-14</td>
</tr>
<tr>
<td>Start: Starting an Application or Cube</td>
<td>16-14</td>
</tr>
<tr>
<td>Stop: Stopping an Application or Cube</td>
<td>16-14</td>
</tr>
<tr>
<td>Upload: Adding Cube Files</td>
<td>16-15</td>
</tr>
<tr>
<td>Version: Displaying API Version</td>
<td>16-15</td>
</tr>
</tbody>
</table>

## Working with Logs

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downloading Server and Application Logs</td>
<td>17-1</td>
</tr>
</tbody>
</table>
Preface

Learn how to get started with Oracle Analytics Cloud – Essbase.

Topics

• Audience
• Documentation Accessibility
• Related Resources
• Conventions

Audience

Oracle Analytics Cloud – Essbase is intended for business users, analysts, modelers, and decision-makers across all lines of business within an organization who use Oracle Analytics Cloud – Essbase.

Documentation Accessibility

For information about Oracle’s commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Resources

Use these related resources to expand your understanding of Oracle Analytics Cloud - Essbase.

Topics

• Oracle Public Cloud http://cloud.oracle.com
• Technical Reference for Oracle Analytics Cloud - Essbase
• Accessibility Guide for Oracle Analytics Cloud - Essbase
• Designing and Maintaining Essbase Cubes
• Getting Started with Oracle Analytics Cloud
• Administering Oracle Analytics Cloud in a Customer-Managed Environment

Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td><strong>monospace</strong></td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
Getting Started with Oracle Analytics Cloud - Essbase

Oracle Analytics Cloud - Essbase is a cloud-based business analytics solution that uses a proven, flexible, best-in-class architecture for analysis, reporting, and collaboration. This cloud service delivers instant value and greater productivity for your business users, analysts, modelers, and decision-makers, across all lines of business within your organization. You can interact with the cloud service, through a Web 2.0 or Microsoft Office interface, to analyze, model, collaborate, and report. Oracle Analytics Cloud - Essbase, is built to scale and uses the industry-standard Oracle Cloud infrastructure.

- Overview of Oracle Analytics Cloud - Essbase
- Accessing Oracle Analytics Cloud - Essbase
- Cloud Service and Smart View Client URLs
- Setting Up Your Client
- Managing Password and Password Challenge Questions
- Oracle Learning Library

Overview of Oracle Analytics Cloud - Essbase

Proven Platform and Technology

Oracle Analytics Cloud - Essbase helps you to plan your company’s cloud strategy efficiently by avoiding data and business process fragmentation. The cloud service’s functional architecture optimizes Oracle Cloud resources, and is based on the proven Oracle Essbase platform, which is widely used to solve simple to complex business analytics use cases across all industries. It is designed to help you model business performance levels and deliver what-if analyses for varying conditions. In Oracle Cloud, you can maintain enterprise-wide user profiles in one place so that you can reuse them across all Oracle Cloud services to which your organization subscribes.

Best-in-Class Functionality

Oracle Analytics Cloud - Essbase can be accessed on an intuitive Web 2.0 or Microsoft Office interface for all of your analytic and business modeling needs, from simple rack-and-stack multi-dimensional analysis, to complex procedural business logic applied to your data. You can easily create and share on-the-fly transient models or deliver enterprise-wide long-established databases. New agile forecasting functionality facilitates collaborative what-if analysis and modeling.

Scalable and Flexible

Oracle Analytics Cloud - Essbase is built on the Essbase platform. It leverages the best of the on-premises software and adds new functionality for the cloud. You can now increase the available size and complexity of your models, while reducing the complexity for deploying them. The cloud service offers you specially built technology
to facilitate your rapid deployment of on-the-fly analytic models, enabling them to be extended as your business needs evolve or discarded in favor of your new models. By creating and sharing these ad hoc models, you can quickly build and collaborate using Microsoft Excel and Web interfaces.

Enterprise Ready

Oracle Analytics Cloud - Essbase is your one-stop cloud service to build, deploy, and manage analytic and reporting applications for any size organization and deployments. It supports data backup and migration, and can also distribute application templates throughout your organization without compromising ease of use or self-service options. Oracle Analytics Cloud - Essbase includes comprehensive features to raise issues, get support, and seek product enhancements. The cloud service provides flat-file and Excel-based import and export.

Rapid Deployment

With Oracle Analytics Cloud - Essbase, you can get started immediately, because it requires no initial investment. Your subscription includes everything that you need. You do not need to license, install, upgrade, or patch software. You do not have to buy, install, or configure hardware. Using the new application workbook upload, you can also leverage the deep product expertise of the worldwide Oracle Hyperion Partner network, to develop and deploy cloud-based analytic models in hours.

Portability

If you're an existing Essbase customer, you can leverage its built-in migration capabilities to port your on-premises Essbase applications to Oracle Analytics Cloud - Essbase. Migrating enables your organization to extend Oracle Analytics Cloud - Essbase usage across the enterprise to other lines of business, without additional demands on your IT resources, hardware, personnel or budget.

Accessing Oracle Analytics Cloud - Essbase

You can access a cloud service instance using credentials supplied by your Service Administrator.

You can also access the cloud service from Smart View. See Accessing the Cloud Service from Smart View.

Accessibility mode is not enabled by default. If you're using a screen reader, see Enabling Accessibility in Accessibility Guide for Oracle Analytics Cloud - Essbase for information on enabling accessibility mode.

To access the cloud service, you must have the following information:

- URL to the cloud service
- User name
- Password
- Identity domain to which you belong

The first time you log in, you need to:

- Check the email from your service administrator for the URL for accessing the cloud service.
• Check the email from the Oracle Cloud Administrator (oraclecloudadmin_ww@oracle.com) for your user name, temporary password, and the identity domain to use.

After you log in, the Applications home page is displayed.

Using Service Credentials to Access the Cloud Service

1. Go to the Oracle Analytics Cloud - Essbase URL for your organization:

   https://instance_name-id_domain.epm_us2.oraclecloud.com/essbase

   See Cloud Service and Smart View Client URLs.

2. (Optional) Specify your identity domain.
   a. In Enter your Identity Domain, enter the name of the identity domain that services the instance.
   b. If you access only one identity domain, select Save your identity domain selection for future sign-ins to set it as the default domain.

      Because you are not prompted to specify an identity domain during future sign-in attempts, do not select this check box if you use cloud services that access different identity domains.
   c. Click Go.

3. Enter your user name and password.

4. Click Sign In.

   If you have already reset your default password, the cloud service home page is displayed.

   If you are accessing the cloud service for the first time, the Password Management screen is displayed to help you personalize your password.

   a. In Old Password, enter the temporary password that you received in the email from the Oracle Cloud Administrator (oraclecloudadmin_ww@oracle.com).
   b. In New Password and Re-Type Password, enter a new password that conforms to the password policy displayed on the page.
   c. In Register challenge questions for your account, select challenge questions and their answers to retrieve the password if you forget it.
   d. Click Submit.

Accessing the Modern User Interface

In Oracle Analytics Cloud – Essbase, you can select either Classic User Interface or Modern User Interface.

This guide describes how to use Essbase in the Classic User Interface. For description of how to use Essbase in the Modern User Interface, see Using Oracle Analytics Cloud - Essbase Modern User Interface.

To select the modern interface,

1. Log in using the credentials provided by your Service Administrator.
2. Click the drop-down menu associated with your user name.
3. Select Modern User Interface.

Accessing the Cloud Service from Smart View

When you access Smart View, the login screen is displayed. You must enter the credentials for a cloud service instance and identity domain, which you get from your Service Administrator.

1. Open Microsoft Excel.
2. Select Smart View, and then Panel.
3. Click Private Connections.
4. Enter your cloud service user name and password.
5. Enter the name of the identity domain that your cloud service instance uses.
6. Click Sign In.
7. In the Private Connections list, select Oracle Essbase Cloud Service.

See Cloud Service and Smart View Client URLs.

Cloud Service and Smart View Client URLs

Get the URL for the cloud service instance you are using from your Service Administrator.

The cloud service has URLs for production and test instances. In these URLs, the values for instance_name and id_domain are provided by the customer when applying for the cloud service, and the data_center value is determined by the cloud service. In the cloud service test instance URL, the word "test" is inserted between instance_name and id_domain, as in instance_name-test-id_domain.

- Cloud service production instance URL:
  https://instance_name-id_domain.epm.data_center.oraclecloud.com/essbase

  For example:
  https://myEssbase-myDomain.epm.us2.oraclecloud.com/essbase

- Cloud service test instance URL:
  https://instance_name-test-id_domain.epm.data_center.oraclecloud.com/essbase

  For example:
  https://myEssbase-test-myDomain.epm.us2.oraclecloud.com/essbase

Cloud service components, such as the Smart View client, have their own URLs.

Smart View client URL:
cloud_service_url/essbase/smartview

You can access Smart View if you have valid credentials. You can also configure the Smart View URL. See Creating Data Source Connections to the Cloud Service.
Managing Client Software

You can install the following Oracle Analytics Cloud - Essbase components on a local computer:

- **Smart View**
  You must synchronize the version of the Smart View client that you use to access the cloud service with the version of the servers deployed in the cloud service. The readme file available from the cloud service indicates whether you require a new client installation to access the cloud service.
  See Downloading and Running the Smart View Installer and Creating Data Source Connections to the Cloud Service.

- **Cube Designer extension in Smart View**
  See Installing the Smart View Cube Designer Extension.

- **Cube export utility**
  See Downloading the Cube Export Utility.

- **Lifecycle Management (LCM) command-line utility**
  See Application Migration and Use Cases.

- **Oracle Analytics Cloud - Essbase command-line tool (CLI)**
  See Downloading and Using the Command-Line Interface.

- **MaxL Client utility**

Managing Password and Password Challenge Questions

On your first login, you're prompted to personalize your password and set responses to challenge questions, so that you can retrieve your password if you forget it. Later, you can reset the password and change your challenge questions.

1. Enter your current or temporary password that you received in the email from the Oracle Cloud Administrator.
2. Enter and confirm a new password.
3. Register your challenge questions for your account, by selecting challenge questions and their answers.

To obtain a temporary password because you forgot your password:

1. Go to the URL for your cloud service instance.
   https://instance_name-id_domain.epm_us2.oraclecloud.com/essbase
   To view a list of URLs, see Cloud Service and Smart View Client URLs.
2. (Optional) If you did not set a default identity domain during a previous session, enter your identity domain, and then click Go.
3. In Sign In to Oracle Cloud, click the Can’t access your account? link.
4. Enter your user ID and responses to your challenge questions.
Oracle Learning Library

The Oracle Learning Library (OLL) is dedicated to hosting free instructional content developed by Oracle subject-matter experts. Use the Search function to find tutorials, overview videos, and Oracle by Example (OBE) tutorials. For example, to find videos for related products such as Oracle Hyperion Smart View for Office, use the OLL Advanced Search and set the Product search filter to Smart View.
Top Tasks for Oracle Analytics Cloud - Essbase

These topics take you through a series of workflows that cover many of the top tasks that you can do in the cloud service and in Cube Designer, as a Service Administrator or Power User.

Service Administrator:

• Top Tasks Prerequisites
• Creating Users and Assigning Roles

Power User:

• Importing and Exporting Bulk Users
• Understanding Application Workbooks
• Creating an Application in the Cloud Service and Provisioning a User to Access and Query the Cube
• Analyzing an Application in Smart View
• Modifying an Outline in the Cloud Service
• Continuing Analysis in Smart View
• Creating an Application and Cube in Cube Designer
• Analyzing Data and Performing an Incremental Update in Cube Designer
• Transforming Tabular Data into a Cube
• Exporting and Modifying Metadata and Data
• Creating a Hierarchy from a Measures Dimension
• Analyzing Data with Drill Through Reports

Top Tasks Prerequisites

Before you start reviewing the top tasks topics, be sure you have met these prerequisites:

1. Be sure that you can log into the cloud service.
   See Accessing Oracle Analytics Cloud - Essbase.

2. Be sure that Smart View and Cube Designer extension are installed on client computers.
   See Setting up Cube Designer.
Creating Users and Assigning Roles

In this task, you’ll create a user and assign one of the predefined Oracle Analytics Cloud - Essbase roles to that user.

<table>
<thead>
<tr>
<th>Predefined User-Level Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Administrator</td>
<td>Can administer the entire cloud service instance, including backing up, creating and deleting applications, provisioning users, and running jobs (such as calculations and data loads).</td>
</tr>
<tr>
<td>Power User</td>
<td>Can create applications and cubes, and grant access to users to perform actions on those cubes.</td>
</tr>
<tr>
<td>User</td>
<td>Can access and perform actions on cubes for which access has been granted.</td>
</tr>
</tbody>
</table>

1. Log into Oracle Analytics Cloud - Essbase using the default Oracle Analytics Cloud administrative user, **admin**.

2. On the Applications home page, without selecting an application or cube, click **Security**.

3. On the **Users** tab, click **Create** to add and provision Oracle Analytics Cloud - Essbase users.

4. On the **New User** dialog box, provide the following information:
   a. Enter a user ID for a Service Administrator role: **server_admin**.
   b. Enter the user's name.
   c. Enter the user's e-mail address.
   d. Create and confirm a password.
   e. Select the role **Service Administrator**.

5. Click **Create**. The new Service Administrator user is added to the list of users.

In **Importing and Exporting Bulk Users**, you will import and export batches of users.

Importing and Exporting Bulk Users

In **Creating Users and Assigning Roles**, you learned about creating a user and assigning a role.

Now you’ll learn how to run bulk imports and exports of files with multiple users and roles.

**Importing a Bulk File of Users and Roles**

1. Create a comma-separated Excel file (saved as *.csv*) that contains the user ID, first and last names (optional), email address (optional), password (optional) and role type (User, Power User, or Service Administrator).

2. Log into Oracle Analytics Cloud - Essbase as a Service Administrator, **server_admin**.
5. Browse to the local .csv file created in Step 1, and click Import.
6. Click Download Logs if you want to view the log file for results of the import. Otherwise, click Cancel.
7. Click Search to refresh the page and view the users and roles imported to the cloud service from the .csv file.

**Exporting Bulk Users and Roles**

1. On the Security page, on the Users tab, click Export.
2. Save the .csv file to a local directory.
3. You can open the .csv file in Excel to view the exported user data.

**Note:**

Passwords are not exported, so that column does not contain values. You can enter a password in this column, to assign an initial user password for users imported from this file.

In Understanding Application Workbooks, you’ll download a sample application workbook.

**Understanding Application Workbooks**

In Importing and Exporting Bulk Users, you learned about user roles and how the Power User can create applications. The cloud service provides a collection of sample application workbooks that you can modify for your own use to quickly deploy an application and cube.

Now you’ll learn about the structure of an application workbook.

**Downloading the Block Storage Sample (Dynamic) Application Workbook**

In the Block Storage Sample (Dynamic) application workbook, all non-leaf level members in the cube are dynamically calculated when a user request retrieves them. The calculated values are not stored in the cube; the values are recalculated for each retrieval.

1. On the Applications home page, click Templates.
2. On the Templates page, click Download next to Block Storage Sample (Dynamic).
3. Save the application workbook file, Sample_Basic_Dynamic.xlsx, to a local drive.

**Examining the Structure of the Block Storage Sample (Dynamic) Application Workbook**

Application workbooks contain a number of worksheets that define the metadata and, optionally, data, calculations scripts, and substitution variables for the cube.
1. In Microsoft Excel, open *Sample_Basic_Dynamic.xlsx*.

2. On the Essbase.Cube worksheet, the application name (Sample_Dynamic), cube name (Basic), the names of 10 dimensions, and other information about the dimensions, are defined.

3. Each dimension has a separate worksheet, Dim.\textit{dimname}, in which the dimension is further defined with information, such as the build method and incremental mode. Because the build method for each dimension in this sample application workbook is PARENT-CHILD, members are defined in PARENT and CHILD columns.

   On the Dim.Year worksheet, months roll up to quarters, and quarters roll up to years. For example, child members Jan, Feb, Mar roll up to parent member Qtr1. Child member Qtr1 rolls up to parent member Year.
The Dim.Product and Dim.Market worksheets are similarly structured. In Dim.Product, SKUs roll up to product families, and product families roll up to Product. For example, child members 100-10, 100-20, and 100-30 (SKUs) roll up to parent member 100 (product family). Child member 100 rolls up to parent member Product.

4. This sample application workbook includes data. Scroll to the last worksheet, Data.Basic, to review the structure of the columns and the data.

In Creating an Application in the Cloud Service and Provisioning a User to Access and Query the Cube, using the Block Storage Sample (Dynamic) application workbook, you’ll create the cube named Basic.

Creating an Application in the Cloud Service and Provisioning a User to Access and Query the Cube

In Understanding Application Workbooks, you learned about the structure of an application workbook.

Now you’ll learn how to create an application in the cloud service and provision a user to access and query the cube.

Creating a Cube in the Cloud Service

1. In the cloud service, on the Applications home page, click Import.
2. On the Import Cube – Excel File dialog box, click **Browse**, and open the Block Storage Sample (Dynamic) application workbook, Sample_Basic_Dynamic.xlsx, that you saved in the previous task.

   The Block Storage Sample (Dynamic) application workbook defines the Sample_Dynamic application and Basic cube.

3. For **Build Option**, select **Create Cube**, and then select **Load Data**.

   You do not need to select **Execute Scripts** because the cube is dynamic and is populated with data automatically.

4. Click **Deploy and Close**.

   In a few moments, the Sample_Dynamic application and Basic cube are created.

5. On the Applications home page, expand the **Sample_Dynamic** application, and select the cube, **Basic**.

6. Click **Outline**.

   The outline is a representation of the dimensions in the Basic cube as defined in the application workbook.

7. View a cube dimension, and then drill down into the children of that dimension:

   a. Click on the **Year** dimension to view the quarters.

      ![Year Outline]

   b. Click the individual quarters to view months.

   Now all of the information from the application workbook is represented in the new cube.

**Provisioning a User to Access and Query the Cube**

1. Click **Home** to go to the Applications home page.

   You are still logged in as a power user, which lets you provision a user.

2. Select the application for which you want to provision the user; in this example, select **Sample_Dynamic**.
3. Click **Security**.

4. Click **Provisioning**.

5. Click **Search** to refresh the list of users.

6. Select a user with the User role.

7. On the **Provision** menu, select **Roles**.

8. On the Provision Roles dialog box, select **Database Manager** and click **Add** to move your selection to the **Selected Roles** text box.

   The Database Manager has full control of the cube, but no control over the application.

9. Click **OK**.

In **Analyzing an Application in Smart View**, you'll go to Smart View, log in as the user you just provisioned and then query a cube.

---

### Analyzing an Application in Smart View

In **Creating an Application in the Cloud Service and Provisioning a User to Access and Query the Cube**, you created an application, a cube with data, and a user.

Now you'll learn how to connect to the cube from Smart View, and do some analysis of the data.

This task assumes that you installed Smart View. See **Downloading and Running the Smart View Installer**.

#### Connecting to the Cube from Smart View

1. Open Microsoft Excel.
   
   If Smart View is installed, you can see the Smart View ribbon.

2. On the Smart View ribbon, click **Panel**.

3. On the Smart View Home dialog box, click the arrow next to the **Home** button, then select **Private Connections**.

4. Make a private connection using the same cloud service instance URL that you used to connect to Oracle Analytics Cloud - Essbase and append `/essbase/smartview` to the end of that URL. For example, `http://myserver:9000/essbase/smartview`.

5. Expand EssbaseCluster and, when prompted, log in as the user you created in **Creating an Application in the Cloud Service and Provisioning a User to Access and Query the Cube**.

6. Connect to the Basic cube under the Sample_Dynamic application.
Performing an Ad-hoc Analysis

Once you’re connected to the Basic cube, you’re ready to begin analyzing the data.

1. On the EssbaseCluster tree, under Sample_Dynamic, select the Basic cube, then click Ad hoc analysis.
2. In the resulting grid, you can see one aggregated data value for all five dimensions of this dynamic cube.

<table>
<thead>
<tr>
<th>Product</th>
<th>Market</th>
<th>Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td>105522</td>
<td></td>
</tr>
</tbody>
</table>

3. Navigate into the member Scenario and narrow it down to a specific scenario type of Actual data.
   a. Click the cell containing Scenario.
   b. On the Essbase ribbon, click Member Selection.
   c. In the Member Selection dialog box, select the Actual member.
   d. Click Add to move Actual to the right pane.
   e. If Scenario is already included in the right pane, remove it, and click OK.

The grid should now look like this:

<table>
<thead>
<tr>
<th>Product</th>
<th>Market</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure</td>
<td></td>
<td>105367</td>
</tr>
<tr>
<td>Year</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. Navigate into Measures and narrow it down to the Sales member, to look at sales data.
   a. Click the cell containing Measures.
   b. Click Member Selection.
   c. Expand Profit and Margin, and select the Sales member.
   d. Click Add to move Sales to the right pane.
   e. If any other members of the Measures dimension are already included in the right pane, remove them, and click OK.

The grid should now look like this:

5. Zoom in to Year by double-clicking the cell containing Year.
   The grid should now look like this:

6. Zoom in to Product by double-clicking the cell containing Product.
   The grid should now look like this:
7. Change the display of the products from SKU numbers to descriptive aliases.
   
   a. On the Essbase ribbon, click Change Alias.

   b. Select Default.

   c. (Optional) Widen column A to see show the names.

   The grid should now look like this:
8. Enhance your data display to show time periods per product. Pivot Qtr1 of Colas by right-clicking it and dragging it from B3 to C3. The grid should now look like this:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Market</td>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sales</td>
<td></td>
<td>Sales</td>
<td>Sales</td>
<td>Sales</td>
</tr>
<tr>
<td>3</td>
<td>Qtr1</td>
<td>Qtr2</td>
<td>Qtr3</td>
<td>Qtr4</td>
<td>Year</td>
</tr>
<tr>
<td>4</td>
<td>Colas</td>
<td>24704</td>
<td>27187</td>
<td>28544</td>
<td>25355</td>
</tr>
<tr>
<td>5</td>
<td>Root Beer</td>
<td>26627</td>
<td>27401</td>
<td>27942</td>
<td>27116</td>
</tr>
<tr>
<td>6</td>
<td>Cream Soda</td>
<td>23997</td>
<td>25736</td>
<td>26650</td>
<td>25022</td>
</tr>
<tr>
<td>7</td>
<td>Fruit Soda</td>
<td>20148</td>
<td>21355</td>
<td>22079</td>
<td>20648</td>
</tr>
<tr>
<td>8</td>
<td>Diet Drinks</td>
<td>25731</td>
<td>26787</td>
<td>27495</td>
<td>25665</td>
</tr>
<tr>
<td>9</td>
<td>Product</td>
<td>95476</td>
<td>101679</td>
<td>105215</td>
<td>98141</td>
</tr>
</tbody>
</table>

9. Look at each product by region. Double-click Market in B1. The grid should now look like this:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sales</td>
<td>Sales</td>
<td>Sales</td>
<td>Sales</td>
<td>Sales</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Qtr1</td>
<td>Qtr2</td>
<td>Qtr3</td>
<td>Qtr4</td>
<td>Year</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>East</td>
<td>Colas</td>
<td>6292</td>
<td>7230</td>
<td>7770</td>
<td>6448</td>
</tr>
<tr>
<td>5</td>
<td>East</td>
<td>Root Beer</td>
<td>5726</td>
<td>5902</td>
<td>5863</td>
<td>6181</td>
</tr>
<tr>
<td>6</td>
<td>East</td>
<td>Cream Soda</td>
<td>4868</td>
<td>5327</td>
<td>5142</td>
<td>4904</td>
</tr>
<tr>
<td>7</td>
<td>East</td>
<td>Fruit Soda</td>
<td>3735</td>
<td>3990</td>
<td>4201</td>
<td>3819</td>
</tr>
</tbody>
</table>

10. Drill in to a region to view product sales by state. Double-click East in A4. The grid should now look like this:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sales</td>
<td>Sales</td>
<td>Sales</td>
<td>Sales</td>
<td>Sales</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Qtr1</td>
<td>Qtr2</td>
<td>Qtr3</td>
<td>Qtr4</td>
<td>Year</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>New York</td>
<td>Colas</td>
<td>1998</td>
<td>2358</td>
<td>2612</td>
<td>1972</td>
</tr>
<tr>
<td>5</td>
<td>New York</td>
<td>Root Beer</td>
<td>1778</td>
<td>1989</td>
<td>1879</td>
<td>2293</td>
</tr>
<tr>
<td>6</td>
<td>New York</td>
<td>Cream Soda</td>
<td>2033</td>
<td>2543</td>
<td>2421</td>
<td>2308</td>
</tr>
<tr>
<td>7</td>
<td>New York</td>
<td>Fruit Soda</td>
<td>1896</td>
<td>2195</td>
<td>2413</td>
<td>2010</td>
</tr>
<tr>
<td>8</td>
<td>New York</td>
<td>Diet Drinks</td>
<td>#Missing</td>
<td>#Missing</td>
<td>#Missing</td>
<td>#Missing</td>
</tr>
<tr>
<td>9</td>
<td>New York</td>
<td>Product</td>
<td>7705</td>
<td>9085</td>
<td>9325</td>
<td>8583</td>
</tr>
<tr>
<td>10</td>
<td>Massachu Colas</td>
<td>1456</td>
<td>1719</td>
<td>1905</td>
<td>1438</td>
<td>6518</td>
</tr>
<tr>
<td>11</td>
<td>Massachu Root Beer</td>
<td>1385</td>
<td>1263</td>
<td>1206</td>
<td>1326</td>
<td>5180</td>
</tr>
</tbody>
</table>

Because not every product is sold in every state, some cells have the #Missing label instead of a data value.
In this task, you navigated through a data grid easily, zooming in and pivoting by clicking in the grid itself. You can also use the tools on the Essbase ribbon to perform the same actions. For more help on using Smart View, click the Smart View tab, and then click Help.

In Modifying an Outline in the Cloud Service, you'll go back to the cloud service and modify an outline.

## Modifying an Outline in the Cloud Service

In Analyzing an Application in Smart View, you analyzed an application in Smart View. Now you'll modify a cube outline in the cloud service.

### Creating a New Member

You start by creating a new member.

1. In the cloud service, on the Applications home page, select the Basic cube in the Sample_Dynamic application.
2. Click Outline.
3. Click Lock and Edit.
   - Locking the outline allows one person at a time to make changes.
4. Enter the Scenario dimension by clicking Scenario.
5. Insert a member:
   - a. Click Budget.
   - b. Click Add sibling.
6. Enter the member name, Forecast, and press Enter.
7. Press the tilde (~) consolidation operator.
   - The Forecast member does not aggregate with the other members in its dimension.
8. Leave the data storage type as Store Data because we want users to be able to input forecast data.
9. Click Save and Unlock.

### Seeding the Forecast Member with Data

To seed the Forecast member with data, we'll create a calculation script and calculate forecast data.

1. On the Applications home page, with the Basic cube still selected, click Scripts.
2. In the Name field, enter salesfcst for the calculation script.
3. In the Script box, enter a simple formula:

   ```plaintext
   Forecast(Sales=Sales->Actual*1.03);
   ```
Forecast for sales is equal to actual sales multiplied by 1.03, which seeds the Forecast member for Sales with a value 3% higher than the actual sales.

4. Click **Save**.

**Executing the Script**

Calculation scripts are executed as jobs.

1. On the Applications home page, with the **Basic** cube still selected, click **Jobs**.
2. Click **New Job**.
3. On the **Execute Job** dialog box, from the **Job Type** menu, select **Calc Execution**.
4. On the **Scripts** menu, select the **salesfcst** calculation script that you created.
5. Click **Execute**.
6. Click **Refresh** to see that the job completes.

In **Continuing Analysis in Smart View**, you'll analyze data in Excel.

**Continuing Analysis in Smart View**

In **Analyzing an Application in Smart View**, you learned to analyze data in Smart View. In **Modifying an Outline in the Cloud Service**, you added a Forecast member to the outline, and seeded it with data.

Now you'll reconnect to the cube in Smart View, and do further analysis of the data.

1. Open Excel and create a worksheet like the following one, by typing the member names in these cells: A3=Market, B3=Product, C1=Year, C2=Actual, D1=Sales, D2=Forecast.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Year</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Market</td>
<td>Product</td>
<td>Actual</td>
<td>Forecast</td>
</tr>
</tbody>
</table>

2. On the Smart View ribbon, reconnect to Basic cube in the Sample_Dynamic application, as you did in **Analyzing an Application in Smart View**.

Your previous connection URL should be shown in the list of Private Connections.
3. When prompted to log in, connect as the same user you created in Creating an Application in the Cloud Service and Provisioning a User to Access and Query the Cube.

4. To populate cells with data values, click **Ad hoc analysis**. In the resulting grid, you should be able to see the results of your calculation in Modifying an Outline in the Cloud Service. The yearly sales data refreshes for both Actual and Forecast, and the forecast appears to be about 3% higher than the actual:

```
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Year</td>
<td>Sales</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Actual</td>
<td>Forecast</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Market</td>
<td>Product</td>
<td>400511</td>
<td>412526.3</td>
</tr>
</tbody>
</table>
```

5. To test that the calculation is correct, create this Excel formula, =D3/C3, in cell E3, which divides the forecast data by the actual data, to ensure that D3 is 3% higher than C3.

```
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Year</td>
<td>Sales</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Actual</td>
<td>Forecast</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Market</td>
<td>Product</td>
<td>400511</td>
<td>412526.3</td>
<td>=D3/C3</td>
</tr>
</tbody>
</table>
```

The test result should confirm the 3% increase, in which Actual is 400511, Forecast is 412526.3, and E3 is 1.0.

```
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Year</td>
<td>Sales</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Actual</td>
<td>Forecast</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Market</td>
<td>Product</td>
<td>400511</td>
<td>412526.3</td>
<td>1.03</td>
</tr>
</tbody>
</table>
```

6. Zoom in on Product and Market. You can see that for all products and all markets, the forecast data is present and is 3% higher than the actual.
7. Now, build a worksheet that you will use to do a data analysis on the forecast, and make some changes.
   a. Click the cell containing Forecast, then click **Keep Only**.
   b. Select cells A3-B3 containing East and Colas, then click **Keep Only**.
      The grid should now look like this:

      ![Worksheet Grid]

      |   |   |   |
      |---|---|---|
      | A | B | C |
      | 1 | Year | Sales |
      | 2 | Forecast |
      | 3 | East | Colas | 28572.2 |
   c. With cells A3-B3 still selected, click **Zoom In** to view per-state information for detailed product SKUs.
      The grid should now look like this:

      ![Worksheet Grid]

      |   |   |   |
      |---|---|---|
      | A | B | C |
      | 1 | Year | Sales |
      | 2 | Forecast |
      | 3 | East | Colas | 28572.2 |
      | 4 | New Yo | Cola | 9208.2 |
      | 5 | New Yo | Diet Cola | #Missing |
      | 6 | New Yo | Caffeine Free Cola | #Missing |
      | 7 | New Yo | Colas | 9208.2 |
      | 8 | Massac | Cola | 6713.54 |
      | 9 | Diet Cola | #Missing |
   d. Pivot the Year dimension down into the columns, and then zoom in all the way to the bottom level of the months.
The grid should now look like this:

```
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>New Yo</td>
<td>Cola</td>
<td>698.34</td>
<td>664.35</td>
<td>695.25</td>
<td>733.36</td>
<td>778.68</td>
<td>916.7</td>
</tr>
<tr>
<td>5</td>
<td>New Yo</td>
<td>Diet Col</td>
<td>#Missing</td>
<td>#Missing</td>
<td>#Missing</td>
<td>#Missing</td>
<td>#Missing</td>
<td>#Missing</td>
</tr>
<tr>
<td>6</td>
<td>New Yo</td>
<td>Caffeine</td>
<td>#Missing</td>
<td>#Missing</td>
<td>#Missing</td>
<td>#Missing</td>
<td>#Missing</td>
<td>#Missing</td>
</tr>
<tr>
<td>7</td>
<td>New Yo</td>
<td>Colas</td>
<td>698.34</td>
<td>664.35</td>
<td>695.25</td>
<td>733.36</td>
<td>778.68</td>
<td>916.7</td>
</tr>
<tr>
<td>8</td>
<td>Massac</td>
<td>Cola</td>
<td>508.82</td>
<td>484.1</td>
<td>506.76</td>
<td>534.57</td>
<td>567.53</td>
<td>668.47</td>
</tr>
</tbody>
</table>
```

e. Enter some monthly values to create a Diet Cola forecast. For example, enter 500 in each of the cells in the range C5:H5.

```
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>New Yo</td>
<td>Cola</td>
<td>698.34</td>
<td>664.35</td>
<td>695.25</td>
<td>733.36</td>
<td>778.68</td>
<td>916.7</td>
</tr>
<tr>
<td>5</td>
<td>New Yo</td>
<td>Diet Col</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
</tr>
</tbody>
</table>
```
f. Click Submit Data, and notice that the full year forecast in cell O5 changes to 3000, which is the sum of 500 in each of 6 months.

In this task, you learned how easy it is to analyze and edit the cube in Smart View, as long as you have the correct provisioning.

In Creating an Application and Cube in Cube Designer, you’ll get familiar with Cube Designer.

Creating an Application and Cube in Cube Designer

In Continuing Analysis in Smart View, you analyzed data in Excel.

Now you’ll use Cube Designer to create an application and cube, similar to what we did in the cloud service in a previous task.

Opening the Application Workbook in Cube Designer

1. In Excel, on the Cube Designer ribbon, click Cloud.

   If you are prompted to log in, then log in as a Power User.

2. In the General category, select Sample_Basic.xlsx.

3. Click Open.

The Sample Basic application workbook is different from the Sample Basic Dynamic application workbook in that the Product and Market dimensions do not have dynamically calculated members.
For example, go to the Dim.Market worksheet in Sample_Basic.xlsx. Look at the Storage column. There are no X characters, which indicates that the members are stored.

**Note:**

X characters in the Storage column indicate dynamically calculated members.

Therefore, after creating the dimensions and loading the data, you also need to calculate the cube.

Creating, loading, and calculating the cube can all be done in one step in the Build Cube dialog box.

**Creating, Loading, and Calculating the Cube**

1. On the Cube Designer ribbon, with the Sample Basic application workbook (Sample_Basic.xlsx) still open, click Build Cube.
2. On the Build Option menu, select Create Cube.
3. Click Run.
   - If there is an existing application with the same name, you are prompted to overwrite the application and cube. Click Yes to delete the original application and build this new application.
4. Click Yes to confirm your selection.
   - The View Jobs icon displays an hourglass while the job is in progress. Cube Designer notifies you when the job is completed.
5. Click Yes to launch the Job Viewer and see the status of the job.

**Viewing the Application in the Cloud Service**

1. Log into the cloud service.
2. On the Applications home page, expand the Sample application and select the Basic cube.
3. Click Outline.
   - View the outline, and see that the expected dimensions are present.
4. Click Settings.
5. Under Statistics, click Basic
   - Under Basic Statistics, you see that both level 0 and upper-level blocks exist, showing that the cube is fully calculated.

In Analyzing Data and Performing an Incremental Update in Cube Designer, you'll analyze data in this cube and perform incremental updates from Excel.
Analyzing Data and Performing an Incremental Update in Cube Designer

In Creating an Application and Cube in Cube Designer, you executed a cube build, loaded data, and ran the calculation script defined in the workbook.

Now you'll analyze data, and then perform an incremental cube update.

Analyzing Data in the Sample Basic Cube

1. In Excel, on the Cube Designer ribbon, click Analyze.
2. On the Analyze menu, select Connect Query Sheets.
   If you are prompted to log in, then enter your cloud service user name and password.
3. Select Reuse sheet contents only to reuse the contents of the worksheet.
   You're connected to the Basic cube in the Sample application.
4. You can now analyze the data.
   a. Zoom in on Cream Soda to see all of the low-level products that are part of the Cream Soda family.
   b. Zoom out on New York to see all of the East region, and zoom out again to see all Markets.

That was a quick validation that the cube build was successful and a quick look at how to analyze data.

Performing an Incremental Update on the Sample Basic Cube

1. Go to the product definition dimension on the Dim.Product worksheet, where you'll update the product dimension with some extra products.
2. Insert new members into the workbook, following the 400 product family.
   a. Create a new child under Product called 500 and give it the name Cane Soda.
   b. Create three new SKUs under 500: 500-10, 500-20, and 500-30.
   c. Create three children under Cane Soda to go with the new SKUs. Call them Cane Cola, Cane Diet Cola, and Cane Root Beer.
3. Save the updated workbook.

4. Click **Build Cube**.

5. Click **Run**.

6. Click **Yes** to confirm your selection.

7. Click **Yes** to launch the **Job Viewer**.

8. Wait a few moments and then click **Refresh** to view the status of the job.
   
   You should see **Success**. If the job returns **Error**, then you can double-click the job for more information.

9. Close the **Job Viewer**.

10. Click **View Hierarchy**.

11. On the Dimension Hierarchy dialog box, see that the Cane Soda product group was created.

<table>
<thead>
<tr>
<th>Product</th>
<th>400</th>
<th>Fruit Soda</th>
</tr>
</thead>
<tbody>
<tr>
<td>400-10</td>
<td></td>
<td>Grape</td>
</tr>
<tr>
<td>400-20</td>
<td></td>
<td>Orange</td>
</tr>
<tr>
<td>400-30</td>
<td></td>
<td>Strawberry</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product</th>
<th>500</th>
<th>Cane Soda</th>
</tr>
</thead>
<tbody>
<tr>
<td>500-10</td>
<td></td>
<td>Cane Cola</td>
</tr>
<tr>
<td>500-20</td>
<td></td>
<td>Cane Diet Cola</td>
</tr>
<tr>
<td>500-30</td>
<td></td>
<td>Cane Root Beer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product</th>
<th>Diet</th>
<th>Diet Drinks</th>
</tr>
</thead>
<tbody>
<tr>
<td>100-20</td>
<td></td>
<td>Shared Diet Cola</td>
</tr>
<tr>
<td>200-20</td>
<td></td>
<td>Diet Root Beer</td>
</tr>
<tr>
<td>300-30</td>
<td></td>
<td>Diet Cream</td>
</tr>
</tbody>
</table>
12. Go to the query worksheet, Query.Sample.
13. Collapse the Product dimension by selecting Product in the worksheet and then clicking Keep Only.
14. Select Product again and click Zoom In.
15. Select Cane Soda and click Keep Only.
16. Select Cane Soda and Zoom In to see the children members.

In Transforming Tabular Data into a Cube, you will create an application using an unstructured Excel worksheet.

Transforming Tabular Data into a Cube

You’ve learned to create cubes from application workbooks. You can also create cubes from tabular data. The tabular data can be from any source system (such as an ERP) or data warehouse, as long as the data contains facts and dimension information, and is contained in one worksheet in an Excel file.

In this task, you’ll use the Cube Designer extension to create a cube from a Sales report and analyze the outline.

1. In Excel, select the Cube Designer ribbon, then click Cloud
2. On the Cloud Files dialog box, select Gallery, double click Cubes, double click Table Format, then select Sample_Table.xlsx.
3. Click Open to view the Excel file.

The Sample_Table.xlsx file contains one worksheet, Sales, which represents a common, simple sales report that you might receive from someone in your organization. The column headings indicate that there are measures (such as Units and Discounts), time representations (such as Time.Month and Time.Quarter), geographic regions (such as Regions.Region and Regions.Areas), and products (such as Product.Brand and Product.LOB).

From this report, you can create an application and cube by using introspection, which is a method of inspecting a physical data source (in this case, the Sample_Table.xlsx file) for metadata elements.

4. On the Cube Designer ribbon, click Transform Data.

5. On the Transform Data dialog box, you can accept the default names for the application (Sample_Table) and cube (Basic) or you can change them.

6. Cube Designer inspects the tabular data to detect relationships that determine appropriate dimensionality.

7. Click Run and, when prompted to create the cube, click Yes.

8. When the job is completed, you'll see the status in the Job Viewer dialog box.

   Click Refresh until the status updates to Success.

9. Log into the cloud service.

10. On the Applications home page, expand the Sample_Table application, select the Sales cube, then click Settings.


   Under Dimensions, you can see that the Sales cube has the following dimensions: Measures, Time, Regions, Product, Years_Dim, and Channel_Dim.

   Under Basic Statistics, the number 4928 for Existing level 0 blocks indicates that data has been loaded into the cube.

12. Click Outline.

13. Glance at the outline, then click Measures to zoom in on the members in that dimension.

   You'll notice that Units, Discounts, Fixed Costs, Variable Costs, and Revenue are in a flat hierarchy. In Exporting and Modifying Metadata and Data, you'll create a hierarchy for these Measures so that you can see Revenue net of Discounts, and total costs (fixed and variable).

### Exporting and Modifying Metadata and Data

In Transforming Tabular Data into a Cube, you created an application and cube from tabular data.

In this task, you'll export the newly created application and cube to an application workbook.

1. In the cloud service, on the Applications home page, expand the Sample_Table application, and select the Sales cube.
2. Click Actions and then select Export.
3. On the Export Cube dialog box, click Data.
4. Click Export.
   - If the data size is less than 400 MB, this exports the metadata and data to an Excel file called an application workbook. Save the application workbook, Sample_table.xlsx to your Downloads area. The application workbook defines the cube that you exported.
   - If the data size exceeds 400 MB, the data file is saved in a compressed file and is not included in the exported Excel file. The ZIP file containing the data and the application workbook can be downloaded from Files.
5. Open Sample_Table.xlsx. If your data is larger than 400 MB, Sample_Table.xlsx is in the ZIP file created Sales.zip in Files. Download Sales.zip and extract the contents to access Sample_Table.xlsx.
   For an overview of the structure of application workbooks, see Understanding Application Workbooks.
6. Scroll to the Data.Sales worksheet, which is the data worksheet for the cube.
   The worksheet has the required dimensions, and the expected Measures names, and the data.
   In Creating a Hierarchy from a Measures Dimension, you’ll create a new hierarchy for the Measures dimension.

Creating a Hierarchy from a Measures Dimension

In Exporting and Modifying Metadata and Data, you exported to an application workbook and opened the Measures dimension.

In this task, you’ll create a new hierarchy for the Measures dimension and analyze the results.

1. In Excel, open the Sample_Table.xlsx application workbook, then go to the Dim.Measures worksheet.
   You’ll now create a member to be a new parent and create a new hierarchy.
3. On the inserted row, enter Net Revenue in the CHILD column, and Measures in the PARENT and PARENTID columns.
4. Enter Measures.Net Revenue in the MEMBERID column to define the member ID in the Parent.Child format.
5. Move the Discounts row to the bottom row.
   The parent of Revenue and Discounts is now Net Revenue. Therefore, change their MEMBERID values to Measures.Net Revenue.
7. Enter X for Storage for Net Revenue.Revenue and tilde (~) for Consolidation for Net Revenue.
Net Revenue will be dynamically calculated and will not consolidate with the other members in the dimension.

8. Enter the **Consolidation** value of + for Revenue and - for Discounts. Net Revenue is defined as the difference between Revenue and Discount.

9. Change the **Incremental Mode** to **Remove Unspecified**. Any existing relationships will no longer be kept.

10. Click **Save** to save the application workbook.

11. On the Cube Designer ribbon, click the **Build Cube** icon and select **Update Cube - Retain All Data**.

12. Click **Run** to run the update and then **Yes**.

13. When the job is completed, you’ll see the status in the Job Viewer dialog box. Click **Close**.

14. Open a blank Excel workbook, then click **Analyze**, then **Create Smart View Ad Hoc Query**.

15. On the Smart View pane, select the **Sales** workbook in the hierarchy and then **Ad Hoc analysis**.

16. Select the **Measures** dimension.

17. On the outline, click **Measures** to zoom in on the members in that dimension and to view the created hierarchy of Revenue and Discounts.

**Analyzing Data with Drill Through Reports**

When you create an Essbase cube, you do not use all of the data available from the Oracle database, external database, or file (delimited or Excel). Instead, you choose and summarize the data in the cube that you will use for further analysis.

In this task, you’ll see how to define and execute Drill Through Reports so that you can analyze additional detailed data retrieved from external data sources.

---

**Note:**

Power Users, who can import an application, and create a connection and datasource at the application level, or use the global datasource created by the service admin user, can create drill through reports in the web interface. They can also generate drill through reports using Smart View.

---

**Defining Drill Through Connections to External Databases or Files**

1. On the Applications home page, without selecting an application, click **Refresh**, click **Admin**, and then click the **Connections**.

2. From the **Create** drop-down menu, select **Database, File** or the relevant Oracle database, as the data source connection.

3. On the Create Connection dialog box, enter the details that are relevant to the type of connection:
• Database (external): Enter database name, host, user, password, description, database type, port, and server ID or service name.
• File (delimited or Excel): Enter file name, catalog path and description.
• Essbase: Enter database name, URL, host, port, user, password, and description.
• BI: Enter database name, host, port, user, password, and description.

4. Click Test to check that the connection to the relevant database or file is successful.
5. Click Save to save the connection definition.

Defining Drill Through Data sources
1. After you define the drill through connection, click the Datasources tab.
2. Click Create.
3. On the General tab of the Create Datasource dialog box:
   a. Enter a name for the data source.
   b. From the Connection drop-down menu, select the connection that you just defined.
   c. Under Datasource, enter the relevant file, or database query details.
   d. Click Next.
4. On the Columns tab, you can view the mapped columns and adjust the data types if necessary. The Name will appear as column header, or you can enter an alias for column names, and it will be displayed as the column header on the Drill Through Report. Click Next.
5. On the Parameters tab, you can view any parameters and their default values. Click Next.
6. On the Preview page, you can view the results of the query. Click Save.

You can now see the names of the defined data sources and connection names.

Defining the Drill Through Report
1. On the Applications home page, select the relevant application and Essbase cube.
2. Click Scripts for
3. Click the Drill Through Reports tab.
4. From the Create drop-down menu, select Datasource.
5. On the Drill Through Report dialog box, enter the report name and select your defined data source.
6. For each column to include in the report, select its corresponding Report Columns check box and the relevant Dimension/Generation (Filter condition) mapping.
7. Under Drillable Regions, enter the drillable regions for the report to display in Smart View. For example, @children("Market"). This drillable region would display all of the children of the member Market.
8. Click **Save** when done.

**Executing the Drill Through Report**

You can now execute the Drill Through Report based on the data loaded in the cube and in the source database or file.

1. Open Smart View, zoom in and click the relevant cell in the drillable region upon which Drill Through Report is based.

2. To execute the report, go to the Essbase ribbon, and click **Drill-through**. You can now view and analyze the results of the Drill Through Report execution in Smart View.
Managing Users and Roles

Your administrator can add user accounts, for everyone you expect to use the service, and assign them to suitable roles. Oracle Analytics Cloud - Essbase implements several security layers, which are managed by Oracle, to create a highly secure environment for the cloud service.

- About User and Role Management
- Predefined Roles
- Setting Up Security
- Provisioning Users
- Provisioning Roles

About User and Role Management

Your access to cloud service instances is restricted by cloud service security, as controlled by Oracle Identity Cloud Service (IDCS) or WebLogic LDAP. If you’re an authorized user, you’re granted access according to your assigned role.

The identity management option is defined in the Oracle Analytics Cloud service management interface. If IDCS is enabled, users are created in the IDCS administration interface. Users with a Service Administrator role in IDCS can directly log into Essbase, and are automatically provisioned with an Essbase Service Administrator role. IDCS users with a Service User role require further user provisioning on the cloud service, by a Service Administrator.

Users must have an account in the identity domain associated with the cloud service. Users are created and assigned to predefined roles on the Security page in Oracle Analytics Cloud - Essbase.

According to your assigned role, you can view and work with applications, cubes, and databases. Your role links you to the business activities that you’re permitted to perform within a cloud service instance, and the application data that you can access.

Users and roles are described in Setting Up Security and in Predefined Roles.

Predefined Roles

You can access Oracle Analytics Cloud - Essbase features and functionality when you’re assigned a role at user and application levels. Database roles (Manager, Update, and Access) are assigned at the application level.

If Oracle Identity Cloud Service (IDCS) is enabled, users and groups are created in the IDCS administration interface. In LDAP identity management mode, users and groups are created on the Security page of the cloud service.

You can also assign Essbase application roles to users and groups, on the Security page.
Roles are hierarchical; access granted to lower-level roles is inherited by higher-level roles. For example, Service Administrators, in addition to the access that only they have, inherit the access granted to Power User and User roles.

### Table 3-1  User-Level Roles

<table>
<thead>
<tr>
<th>User-Level Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Administrator</td>
<td>Full access to administer users, applications, and databases.</td>
</tr>
<tr>
<td>Power User</td>
<td>Ability to create and delete applications and databases within applications, that were created by this user.</td>
</tr>
<tr>
<td>User</td>
<td>Ability to access any assigned application, or a database that has a minimum access permission.</td>
</tr>
</tbody>
</table>

### Table 3-2  Application-Level Roles

<table>
<thead>
<tr>
<th>Application-Level Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Manager</td>
<td>Ability to create, delete, and modify databases and application settings within the assigned application; assign users to an application; create and delete scenarios.</td>
</tr>
<tr>
<td>Database Manager</td>
<td>Ability to manage databases, database elements, locks, and sessions within the assigned application; create and delete scenarios.</td>
</tr>
<tr>
<td>Database Update</td>
<td>Ability to read and update data values based on assigned scope, using assigned calculations and filters; create and delete scenarios.</td>
</tr>
<tr>
<td>Database Access</td>
<td>Ability to access scenarios; read data values in all cells; access specific data and metadata, unless further restricted by filters.</td>
</tr>
</tbody>
</table>

## Setting Up Security

### Use-Case Overview

The following are common use cases for assigning (provisioning) users.

- Users can view and access cubes for which they were assigned access to the related applications. Power Users can create ad-hoc cubes for personal analysis of data.
- Power Users can create enterprise-level cubes and grant other users access to applications that the Power User created.
- Service Administrators can assign users at all levels and manage all aspects of the relevant applications, databases, and users.
- Service Administrators can assign a Database Update role for users who need to update data to a sandbox dimension, during Scenario Management.

### Provisioning Options

The following security provisioning options are available.

- Create, edit, and delete users and groups, or change the default user password policy (for LDAP identity mode).
See Provisioning Users.

- Import and export users and groups, on the Users and Groups pages on the cloud service interface (for LDAP identity mode).
  See Importing and Exporting Bulk Users.
- Provisioning and deprovisioning roles, and related reports.
  See Provisioning Roles.

Provisioning Users

If you're a Service Administrator or Power User, you can provision users and groups for applications that you created.

Note:

- If Oracle Identity Cloud Service (IDCS) is enabled, users and groups are created and managed in the IDCS administrator interface. Otherwise, in LDAP mode, users and groups are created and managed on the Security page of the cloud service, as described here.
- You can easily import or export user and group data files using the corresponding buttons. See Importing and Exporting Bulk Users.
- You can't create users and groups with the same name.

Creating a User

1. On the Applications home page, without selecting an application or cube, click Security.
2. On the Users tab, click Create to add a user.
3. On the New User dialog box, enter the ID, Name, Email, and Password for the user, and then select a Role.

Note:

Roles are described in Predefined Roles.

4. Click Create.

Editing a User

1. On the Applications home page, without selecting an application or cube, click Security.
2. On the Users tab, select a user row, click Actions, and then select Edit.
3. On the Edit User dialog box, modify the user details and click Submit.
Deleting a User

1. On the Applications home page, without selecting an application or cube, click Security.

2. On the Users tab, select a user row, click Actions, and then select Delete.

3. On the Delete User dialog box, click Yes if you want to confirm the user deletion.

Creating a Group

1. On the Applications home page, without selecting an application or cube, click Security.

2. On the Groups tab, click Create.

   - **Note:**
     Groups, whose security is handled by IDCS, have access defined as None in the cloud service, and must be manually provisioned.

3. On the New Group dialog box, enter the Name and Description for the group, and then select a Role.

   - **Note:**
     Roles are described in Predefined Roles.

4. Click Create.

Editing a Group

1. On the Applications home page, without selecting an application or cube, click Security.

2. On the Groups tab, select a group row, click Actions, and then select Edit.

3. On the Edit Group dialog box, modify the group details and click Submit.
   The group's details are modified.
Deleting a Group

1. On the Applications home page, without selecting an application or cube, click Security.

2. On the Groups tab, select a row for a group, click Actions, and then select Delete.

3. On the Delete Group dialog box, click Yes if you want to confirm the group deletion.

Modifying Password Policy Details

You can customize the password policy applied to new users created in the service or when resetting passwords. This is only applicable for LDAP mode.

1. Connect to the service instance using Secure Shell (SSH) client software.

2. Switch to the Oracle user:
   
   ```
   sudo su - oracle
   ```

3. Edit the policy file, in the PSM service instance:
   
   ```
   vi /u01/data/domains/esscs/config/fmwconfig/essconfig/essbase/essbase-password-validation-rules.xml
   ```

   Note:
   This is the current default policy file:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <essbase-password-validation-rules>
   <cannot-contain-spaces>true</cannot-contain-spaces>
   <cannot-contain-username>true</cannot-contain-username>
   <maximum-password-length>20</maximum-password-length>
   <minimum-alphabetic-chars>0</minimum-alphabetic-chars>
   <minimum-password-length>8</minimum-password-length>
   <minimum-lowercase-chars>0</minimum-lowercase-chars>
   <minimum-numeric-chars>0</minimum-numeric-chars>
   <minimum-special-chars>0</minimum-special-chars>
   <minimum-uppercase-chars>0</minimum-uppercase-chars>
   </essbase-password-validation-rules>
   ```

4. Exit from the editor as follows:
   
   ```
   :wq
   ```

   If you modify the policy file, it is dynamically updated and you do not need to restart the service.
Provisioning Roles

If you're a Service Administrator or Power User, you can provision roles for users assigned to applications.

Provisioning a User Role

1. In the cloud service, on the Applications home page, select an application row, then click Security.
2. If no users are listed, then click Search to display the user list.
3. Select user row(s), then click Provision and Roles.
4. On the Provision Roles dialog box, select the appropriate role, and remove using the arrow button.

Note:
Roles are hierarchical. Upper-level roles include the privileges of lower-level roles. Roles are described in Predefined Roles.

5. Click OK.

Deprovisioning Roles

1. On the Applications home page, select an application row, then click Security.
2. If no users are listed, then click Search to display the user list.
3. Click Deprovision All. Keep Roles option selected (default setting). Unselect other options, as necessary.
4. Click OK.

Note:
Deprovisioning removes the selected permissions.

Viewing the Provision Report

1. On the Applications home page, select an application row and, click Security.
2. If no users are listed, then click Search to display the user list.
3. Select one or more user rows, and select Provision Report.
   In the displayed report, you can view users, roles, and inherited permissions.
Note:

The View menu options include: column display, and collapse and expand hierarchies. The Export button allows you to export the report.
Designing and Creating Cubes Using Application Workbooks

You can design, create, and modify fully functional cubes using Excel-based application workbooks. You can design the cube within the application workbook, quickly import the workbook to the cloud service to create a cube, load data into the cube, and calculate the cube. You can also work with application workbooks in Cube Designer, which is a Smart View extension.

- About Application Workbooks
- Downloading a Sample Application Workbook
- Creating a Cube from an Application Workbook
- Creating a Cube from Tabular Data
- Connecting to a Cube in Smart View

About Application Workbooks

Application workbooks comprise a series of worksheets, which can appear in any order, and define a cube, including cube settings and dimensional hierarchies. Optionally, you can define data worksheets to be loaded automatically when you create the cube, and calculation worksheets to be executed after you load the data. There are strict layout and syntax requirements for application workbooks, and there are many validations to ensure that workbook contents are complete and formatted correctly. If the application workbook contents are not correct, then the cube building process is not successful.

You can modify the worksheets directly in Microsoft Excel or by using the Cube Designer wizard.

The cloud service provides application workbook templates for creating block storage and aggregate storage applications and cubes.

- Block Storage Sample (Stored): Block storage application workbook. File name: Sample_Basic.xlsx.
- Block Storage Sample (Dynamic): Block storage application workbook. All non-leaf level members are dynamic. File name: Sample_Basic_Dynamic.xlsx.
- Block Storage Sample (Scenario): Block storage application workbook with scenarios enabled. All non-leaf level members are dynamic. File name: Sample_Basic_Scenario.xlsx.
- Aggregate Storage Sample: Aggregate storage application workbook. File name: ASO_Sample.xlsx.
- Aggregate Storage Sample Data: Data for the aggregate storage application workbook. File name: ASO_DATA.txt.
- Tabular Data Sample: Tabular data Excel file. File name: Sample_Table.xlsx.
Oracle recommends that you download a sample application workbook and examine the worksheets. See Application Workbooks Reference.

**Downloading a Sample Application Workbook**

Using a sample application workbook provided in the cloud service, you can quickly create sample applications and cubes. The cubes are highly portable, because they are quickly and easily imported and exported.

1. On the Applications home page, click **Templates**.
2. On the Templates page, click **Download** next to the workbook that you want to work with.
3. Save the Excel .xlsx file to a local drive.
4. Open the file and examine the worksheets to understand how you can use the workbook to create an application and cube.

**Creating a Cube from an Application Workbook**

1. In the cloud service, on the Applications home page, click **Import**.
2. On the Import Cube - Excel File dialog box, browse to a sample application workbook that you previously downloaded on the Templates page or to an application workbook that you have modified.
   
   You cannot import Excel files that contain spaces in the filename.
3. Your application and cube names are populated based on the names you specified in the application workbook on the Essbase.Cube worksheet.
   
   • (Optional) You can change the application and cube names on this screen.
   
   • (Required) If an existing application in the cloud service matches the name of the application you are importing, then you must ensure that the cube name is unique. For example, if the name of the application and cube in the Excel workbook is Sample Basic and the cloud service already has a Sample Basic cube, then you're prompted to rename the cube.
4. (Optional) Select **Show Advanced**, which allows you to view the mapping of workbook columns to the dimensions to be created.
5. (Optional) If the application workbook contains data, then select **Load Data**. If you do not select **Load Data**, then only metadata is imported into the cube.

6. Click **Deploy and Close**.
   The application and cube are listed on the Applications home page.

7. To view the outline, expand the application row to show the cube row; then click **Outline**.
   See **Viewing Outlines**.

**Note:**
When you import an application workbook that was created using the command-line Export Utility, some member names might be rejected. See **Reviewing Member Names before Importing an Application Workbook Created by the Cube Export Utility**.

---

### Creating a Cube from Tabular Data

In this workflow, you're using the sample tabular data Excel file named `Sample_Table.xlsx`, which uses intrinsic column headers. See **About Using Tabular Data to Create Cubes**.

1. In the cloud service, on the Applications home page, click **Templates**.
2. On the Templates page, click **Download** next to **Tabular Data Sample**.
3. Save the Sample_Table.xlsx Excel file to a local drive.

4. On the Applications home page, click **Import**.

5. On the Import Cube - Excel File dialog box, browse to Sample_Table.xlsx. The application and cube names are pre-populated. The application name is based on the source file name without the extension (in this example, Sample_Table) and the cube name is based on the worksheet name (in this example, Sales).

   The option to load data is selected by default and cannot be changed. The option to execute scripts is not available.
   - (Optional) You can change the application and cube names on this dialog box.
   - (Required) If an existing application in the cloud service matches the name of the application that you’re importing, then you must ensure that the cube name is unique. For example, if there is already an application named Sample_Table with a cube named Sales, then you’re prompted to rename the cube.

6. (Optional) Click **Show Advanced** to modify the cube type and the type of dimensions to be created.

   ![Import Cube - Excel File dialog box](image)

   You can perform these actions:
   - Change the cube type. By default, cubes are set to **BSO** (block storage) with the **Hybrid BSO** option. You can keep the block storage type but remove the hybrid block storage option, or you can select the **ASO** (aggregate storage) type.
• Select **Enable Sandboxing**, if applicable.
• Click **Dimensions** and, on the Change Dimension Names dialog box, enter names for the dimensions you want to rename.
• Change the dimension types.

If you make any changes, then click **Apply Changes** before proceeding.

7. Click **Deploy and Close**.
   The application and cube are listed on the Applications home page.

8. (Optional) To view the cube outline, expand the application row to show the cube row, then click **Outline**.

   See **Viewing Outlines**.

### Connecting to a Cube in Smart View

In Smart View, you can create a private connection using the quick connection method if you know the URL. The private connection URL is a concatenation of the cloud service instance URL that you use to log into Oracle Analytics Cloud - Essbase and the string `/essbase/smartview` appended to it. For example, `http://example:9000/essbase/smartview`.

1. From the Smart View ribbon, click **Panel**.
2. From the Smart View panel, click **Home** and then select **Private Connections**.
3. In the text box, enter the URL with the `/essbase/smartview` string appended to it, for example, `http://example:9000/essbase/smartview`.
4. Click the connect arrow.
5. On the Login dialog box, enter your Oracle Analytics Cloud - Essbase user name and password, then click **Sign In**.
Creating and Managing Cube Outlines

An outline defines the structure of the cube through dimensions, members, and attributes. The outline structure, along with consolidation operators and formulas, determines how data is stored and calculated.

- About Cube Outlines
- Viewing and Editing Outline Properties
- Adding Dimensions and Members to Outlines
- Viewing Outlines
- About Duplicate Member Names
- Enabling Duplicate Member Names
- Editing Member Properties
- Setting Dimension and Member Properties
- Naming Generations and Levels
- Setting Advanced Cube Properties
- Unlocking Objects
- Removing User Data Locks
- About Filters
- Working with Attributes
- Managing Outlines

About Cube Outlines

Dimensions and members represent data hierarchies. In an outline, each dimension consists of one or more members. The members, in turn, may have child members. This hierarchy determines the drill-path and, with unary operators, a child member's consolidation to its ancestry.

Viewing and Editing Outline Properties

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click Settings.
2. Select the Properties tab.
3. Select Outline.
4. Many of the entries are read-only. However, you can select the following:
   - Date format.
• Attribute Settings Prefix/Suffix Format Value.

• Boolean, Date and Numeric
  – True Member Name
  – False Member Name
  – Date Member Names
  – Numeric Ranges Represent

See Building Attribute Dimensions and Associating Attributes in Designing and Maintaining Essbase Cubes

• Calculation Dimension Names
  – Name
  – Sum Member
  – Count Member
  – Minimum Member
  – Maximum Member
  – Average Member

See Naming Conventions for Attribute Calculations Dimension Member Names and Understanding the Attribute Calculations Dimension in Designing and Maintaining Essbase Cubes.

After you migrate an Essbase on-premises application with a unique member outline to a cloud service instance, you cannot change the outline to allow duplicate members. To allow duplicate member names in your cloud service instance, convert the on-premises unique member outline to a duplicate member outline before migrating the application.

Adding Dimensions and Members to Outlines

You can add dimensions and members to a cube using any of the following methods:

• Add dimensions and members manually using the Outline page.
  See Adding Dimensions to Outlines Manually and Adding Members to Outlines Manually.

• Create a new application and cube, and add dimensions and members, by importing an Excel file.
  See Creating a Cube from an Application Workbook.

• Build dimensions using a flat file and a rules file.
  See Building Dimensions Using a Rules File.

Adding Dimensions to Outlines Manually

Outlines are composed of members. The top level members of an outline are called dimensions. There are two types of dimensions: standard dimensions and attribute dimensions.
Dimension names must always be unique in the outline, even if the outline allows duplicate member names.

In block storage or partial hybrid aggregation cubes (which have one or more stored dimensions), if you add, delete, or change dimensions and then save the outline, then the cube is restructured. After restructuring is complete, recalculate the data. Aggregate storage and fully hybrid aggregation cubes are not restructured because they are dynamic (upper level data is not stored).

If you add a dimension that is virtual (dynamic calc, or dynamic calc and store) or label only, and do not provide a data member (when you add a dimension with child members), then the first level-0 stored member is set as the data member name. If you provide a data member (a member associated with a dimension and used for storing data), it must be of stored type.

1. On the Applications home page, expand the application to show the cube row for the outline you want to modify, then click Outline.
2. On the Outline page, click Lock and Edit.
3. Select a dimension.
4. Click manage, and then select Sibling.
5. Enter a name for the dimension, and press Enter.

When you press Enter, a new row is created and you have the option to create an additional dimension. If you do not want to create another dimension, then press the Escape key.

Use no more than 1024 characters when naming dimensions, members, or aliases.

6. Select the new dimension row and select from the options on the Outline page to define the dimension.

Use the scroll bar at the bottom of the page to see all available options.

Some options cannot be edited in the Outline page. Select properties to edit those options in the Edit Member Properties dialog box.

See Setting Dimension and Member Properties.

Adding Members to Outlines Manually

Members are the individual components of the outline. Unless the cube is enabled for duplicate member names, each member has a unique name.

1. On the Applications home page, expand the application to show the cube row for the outline you want to modify, then click Outline.
2. On the Outline page, click Lock and Edit.
3. To view and select lower-level members in a dimension, drill down in the dimension by clicking the dimension name.
4. When you reach the member to which you want to add a child or sibling member, select the member row, click manage, and then select Child or Sibling.
5. Enter a name for the member.
Use no more than 1024 characters when naming dimensions, members, or aliases.

After adding a sibling member, you can easily add subsequent sibling members by pressing the Enter key.

6. Select from the options on the Outline page to define the member.

Use the scroll bar at the bottom of the page to see all available options.

Some options cannot be edited in the Outline page. Select properties to edit those options in the Edit Member Properties dialog box.

See Setting Dimension and Member Properties.

Building Dimensions Using a Rules File

When you build a cube, data files and data load rules files are created in the cloud service. You can also use data files and data load rules files from a supported on-premises release of Essbase. See Uploading Files to a Cube.

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click Jobs.

2. Select New Job.

3. On the Execute Job dialog box, from the Select Job Type menu, select Dimension Build.

4. From the Database menu, select the cube that you are building.

5. From the Scripts menu, select the dimension build rules file.

6. For Load Type, choose SQL or File:
   - If you are loading SQL data, enter your user name and password.
   - If you are loading a flat data file, choose the file from the Data Files list.

7. From the Restructure Options menu, select an option:
   - Preserve All Data: All data values are preserved.
   - Preserve Input Data: Only the blocks that contain the data that is being loaded are preserved. However, all blocks (upper- and lower-level) that contain loaded data are preserved. This option does not apply to aggregate storage databases.
   - Preserve Leaf Level Data: Only level-0 (leaf level) values are preserved. If all data required for calculation resides in level-0 members, then you should select this option. All upper-level blocks are deleted before the database is restructured. When the database is recalculated, the upper-level blocks are recreated. This option does not apply to aggregate storage databases.
   - Preserve No Data: All data values are cleared.

8. Click Execute.
9. On the Jobs page, click **Refresh** to view the status of the dimension build.

**Note:**

You can use a dimension build to add or modify dimensions, but you cannot use a dimension build to delete an existing dimension.

### Viewing Outlines

You can decide whether to see all or only some of the outline columns and you can change the order of the columns.

1. On the **Applications** home page, expand the application to show the cube row for the outline you want to modify, then click **Outline**.
2. (Optional) Use the **View** menu to choose the columns to show or hide (see **Hiding Outline Columns**) and to reorder columns (see **Reordering Outline Columns**).
Hiding Outline Columns

1. On the Outline page, select **View**, then **Columns**.
2. On the **Columns** dialog box, deselect the columns that you want to hide.

Reordering Outline Columns

1. On the Outline page, select **View**, and then **Reorder Columns**.
2. Select the columns that you want to move.
3. Use the arrows to adjust the column order.

4. Click **OK**.

About Duplicate Member Names

You can specify that duplicate (non-unique) member names and aliases are allowed in a cube outline, with some restrictions. For example, a cube may require two members named New York in the outline: one as a child member of the State member and one as a child of the City member. In the outline, the member names are displayed as New York. The qualified member names are:
Dimension names, generation names, and level names must always be unique, and sibling members under a parent member must always be unique.

**Note:**
- In a duplicate member name outline, each dimension allows duplicate member names, by default.
- Setting an outline for duplicate member names applies to the entire outline and cannot be assigned only to a dimension, for example.
- After you migrate an on-premises Essbase cube with a unique member outline to a cloud service instance, you cannot change the outline to allow duplicate members. If you want your on-premises cube to allow duplicate members in your cloud service instance, you must convert the unique member outline to a duplicate member outline in your on-premises installation before migrating the application to a cloud service instance.

---

### Enabling Duplicate Member Names

To add a duplicate member name, enter the duplicate member in the outline; see Adding Dimensions and Members to Outlines. There are no additional requirements for adding a duplicate member. If the outline is not enabled for duplicate members, then an error is returned when a duplicate member name is entered.

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, and then click **Settings**.
2. On the Properties dialog box, select **Edit-Outline Properties**.
3. Select **Allow duplicate member names**.

### Editing Member Properties

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click **Outline**.
2. On the Outline page, select a dimension row. To view and select lower-level members, click the dimension or member name.
   Continue this process until you reach the member that you want to manage, then select the table row.
3. If the outline is unlocked (it cannot be edited), then click **Lock and Edit**.
4. Use the scroll bar at the bottom to view and change member properties. You can edit some member properties on the Edit Member Properties dialog box, and others, directly on the Outline page. Some fields are read-only, depending on whether the property is relevant to the specific member.
Note:

Some member properties cannot be viewed or edited on the Outline page. Instead, you must edit them on the Edit Member Properties dialog box. See Setting Dimension and Member Properties.

- The **Statistics** column displays information about each dimension and member in the following format:
  
  
  \[ \text{C} . n \text{ D} . n \text{ G} . n \text{ L} . n \]
  
  - \( C . n \) is the number of child dimension members.
  - \( D . n \) is the number of descendants.
  - \( G . n \) is the generation number.
  - \( L . n \) is the level number.

- Consolidation operators (such as \(+\), \(-\), and \(*\)) are displayed on the Outline page, when you drill down on dimension names to view members. There is no column heading for consolidation operators.

Setting Dimension and Member Properties

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click **Outline**.

2. On the Outline page, select a dimension row. To view and select lower-level members, click the dimension name.

   Continue this process until you reach the member that you want to manage, then select the table row.

3. If the outline is unlocked, click **Lock and Edit**.

4. Click **manage**  

   and then select **Properties**.

5. On the Edit Member Properties dialog box, set member properties on these tabs:

   - Information: See Setting Information Properties
   - Aliases: See Creating Aliases
   - Attributes: See Setting Attribute Associations
   - UDAs: See Creating User-Defined Attributes
   - Formulas: See Creating Member Formulas
Setting Information Properties

On the Information tab, you can view or modify basic dimension or member information (such as consolidation properties, storage properties, and comments).

The options available on the tab vary, depending on the outline type, and the dimension and member type. For example, the items available vary depending on whether the cube is block storage or aggregate storage, or whether you selected a dimension or a member. Selecting a different dimension or member, even within the same outline type, displays different options.

All fields provide information about the selected dimension or member. Some fields are read-only; some fields accept entries; some fields apply to all situations; and some fields apply to specific situations. To modify a value, select it and either enter or select a value.

Table 5-1 Dimension and Member Information Member Properties

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Applies to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a dimension or member name. Use no more than 1024 bytes when naming dimensions, members, or aliases.</td>
<td>• Aggregate storage dimensions and members • Block storage dimensions and members</td>
</tr>
<tr>
<td>Comment</td>
<td>Enter a comment. Comments can contain up to 255 characters.</td>
<td>• Aggregate storage dimensions and members • Block storage dimensions and members</td>
</tr>
<tr>
<td>Dimension type</td>
<td>For a dimension within an aggregate storage outline, select:</td>
<td>• Aggregate storage dimensions • Block storage dimensions</td>
</tr>
<tr>
<td></td>
<td>• None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Accounts</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Attribute</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For a dimension within a block storage outline, select:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Accounts</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Attribute</td>
<td></td>
</tr>
<tr>
<td>Attribute type</td>
<td>For an attribute dimension select:</td>
<td>• Aggregate storage dimensions and members • Block storage dimensions and members</td>
</tr>
<tr>
<td></td>
<td>• Text</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Numeric</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Boolean</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Date</td>
<td></td>
</tr>
<tr>
<td>Dimension solve order</td>
<td>Specify a number (0 to 127) to represent the priority with which formulas assigned to the dimension are calculated.</td>
<td>• Aggregate storage dimensions • Hybrid aggregation dimensions</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
<td>Applies to...</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Consolidation</td>
<td>For a member that is not a dimension or an attribute, select a consolidation operator: • + (addition) • – (subtraction) • * (multiplication) • / (division) • % (percentage) • ~ (ignore) • ^ (nonconsolidating) Addition (+) is the default. The ^ (nonconsolidating) operator applies only to block storage cubes.</td>
<td>• Aggregate storage members • Block storage members</td>
</tr>
<tr>
<td>Two-Pass</td>
<td>Select the <strong>Two-Pass calc</strong> check box to calculate the member during a second pass through the outline.</td>
<td>Block storage members</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Select an option to determine how data values for the current dimension or member are stored: • Store data • Dynamic calc (This option does not apply to aggregate storage cubes.) • Never share • Label only • Shared member</td>
<td>• Aggregate storage dimensions and members • Block storage dimensions and members</td>
</tr>
<tr>
<td>Member solve order</td>
<td>Specify a solve order between 0 and 127 to indicate the priority in which the member is calculated.</td>
<td>• Aggregate storage members • Hybrid aggregation members</td>
</tr>
<tr>
<td>Hierarchy</td>
<td>Specify <strong>Stored</strong> (the default) or <strong>Dynamic</strong> or, for a dimension within an aggregate storage outline, select the <strong>Multiple hierarchy enabled</strong> option (which equates to selecting both <strong>Stored</strong> and <strong>Dynamic</strong>). The storage option that you select is applied to the hierarchy headed by the dimension or generation 2 member.</td>
<td>• Aggregate storage dimensions • Generation 2 aggregate storage members</td>
</tr>
</tbody>
</table>
Table 5-1  (Cont.) Dimension and Member Information Member Properties

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Applies to...</th>
</tr>
</thead>
</table>
| Level Usage for Aggregation       | Select one of these options to provide a way for an administrator to influence both default and query-based view selection:  
  • Default: Internal mechanisms decide how to create aggregations.  
  • No aggregation: Aggregation is not performed along this hierarchy. All views selected are at the input level.  
  • Top level only: (Applies to primary hierarchies.) Queries are answered directly from input data.  
  • No intermediate levels: (Applies to primary hierarchies.) This selects top and bottom levels only. | Aggregate storage dimensions                             |
| Variance reporting expense        | For an Accounts member that is assigned a formula that includes the @VAR or @VARPER function and that is within a block storage outline, you can specify how the difference between actual and budget data are calculated. You select the false option to tag the member as nonexpense (the default) or true to tag the member as expense. | Block storage dimensions and members                    |
Creating Aliases

On the Aliases tab, you can assign alternate names, or aliases, to a dimension, member, or shared member. For example, in the Sample Basic cube outline, members of the Product dimension are identified by product codes, such as 100, and by descriptive aliases, such as Cola.

1. On the Edit Member Properties dialog box, click Aliases.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
<th>Applies to…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account information</td>
<td>Time Balance: To use time balance properties, you must have a dimension tagged as Accounts and a dimension tagged as Time.</td>
<td>Block storage Accounts dimension only</td>
</tr>
<tr>
<td></td>
<td>• None: Apply no time balance property. Member values are calculated in the default manner.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Average: A parent value represents the average value of a time period.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• First: A parent value represents the value at the beginning of a time period.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Last: A parent value represents the value at the end of a time period.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Skip option: Select an option (None or Missing) to determine what values are ignored during time balance calculations. If you select None, then no values are ignored, and, if you select Missing, then #MISSING values are ignored. You can specify skip settings only if the time balance property is set as first, last, or average.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Missing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You can set these properties for any members except Label Only members.</td>
<td></td>
</tr>
</tbody>
</table>
2. In the **Default** field, enter the value of the alias.

3. Click **Apply and Close**.

### Setting Attribute Associations

On the Associations tab, you can assign attribute values to the member that you are modifying.

1. On the Edit Member Properties dialog box, click **Associations**.

![Edit Member Properties](image)

The attribute dimensions that are associated with the member you are modifying are listed under **Attribute Dimension**.

2. To change an association, select a value from the **Value** list.

For example, you can change the value of the **Caffeinated** attribute from False to True.

See [Working with Attributes](#).

### Creating User-Defined Attributes

On the UDAs tab, you can create, assign, and unassign user-defined attributes (UDAs). A UDA is a word or phrase that describes the member. For example, you might create a UDA called Major Market and assign it to all members in the outline that are part of a major market, as defined by your organization.

Like attributes, UDAs are used to retrieve data by categories. Unlike attributes, UDAs have no built-in functionality. However, UDAs can be assigned to dense and sparse dimensions, whereas attributes can be assigned to only sparse dimensions.
1. On the Edit Member Properties dialog box, click **UDAs**.

![Edit Member Properties dialog box]

2. In the **New UDA** field, enter a UDA name.

3. Click **Create and Assign** to assign the new UDA to the selected members.

You can use the **Move**, **Move All**, **Remove**, and **Remove All** arrows to select or remove UDAs from the member that you are editing.

### Creating Member Formulas

On the Formula tab of the Edit Member Properties dialog box, you can create and edit member formulas for both block storage and aggregate storage cubes. These formulas are calculated through default cube calculations and calculation-script calculations.

![Edit Member Properties dialog box]

You can construct block storage member formulas from operators, functions, dimension names, member names, substitution variables, and numeric constants. The formula tab provides a pane in which you can enter a formula. See Developing Formulas for Block Storage Databases in *Designing and Maintaining Essbase Cubes*.

To write formulas for block storage outlines, a set of calculation functions and operators, known as the Calculator, or Calc, language, is provided. For descriptions of calculation commands and functions, see *Technical Reference for Oracle Analytics Cloud - Essbase*.

Aggregate storage member formulas cannot be created using Calculator language. Instead, create them using Multidimensional Expression Language (MDX). See Aggregate Storage and MDX Outline Formulas in *Technical Reference for Oracle Analytics Cloud - Essbase* and Developing Formulas on Aggregate Storage Outlines in *Designing and Maintaining Essbase Cubes*.
To create member formulas:

1. Write the formula in a text editor or on the Edit Member Properties dialog box, on the Formula tab.
2. If working in a text editor, copy the formula onto the Formula tab.
3. Click Validate Formula.
4. Fix validation errors and click Validate Formula again, until the formula passes validation.
5. Click Apply and Close.

See Creating and Validating Member Formulas in Cube Designer.

Naming Generations and Levels

You can create your own names for generations and levels in an outline, using a word or phrase that describes the generation or level. For example, you might create a generation name called Cities for all cities in the outline. You can define only one name for each generation or level.

Use generation and level names in calculation scripts wherever you need to specify either a list of member names or a list of generation or level numbers. For example, you can limit a calculation in a calculation script to the members of a specific generation.

An outline cannot contain the same Excel header name for a generation and a dimension. Data Visualization displays generation names, while in Smart View, you use dimension names for browsing.

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click Settings.
2. Select Structure.
3. On the Dimensions page, click a generation or level name to edit that field. For example, in the Levels field, click Lev0.
4. Enter a generation or level name.
5. Click Save.

Setting Advanced Cube Properties

If the current cube is a block storage cube, then you can select whether to enable the following options:
• **Aggregate missing values**: If you never load data at parent levels, selecting this option may improve calculation performance. If this option is selected and you load data at the parent level, then the parent-level values are replaced by the results of the cube consolidation, even if the results are #MISSING values.

• **Create blocks on equations**: If this option is selected, then when you assign a non-constant value to a member combination for which no data block exists, a data block is created. Selecting this option can produce a very large cube.

• **Two-Pass calculation**: If this option is selected, then after a default calculation, members that are tagged as two-pass are recalculated. See [Setting Two-Pass Calculation Properties](#).

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click **Settings**.

2. On the **Properties** tab, click **Advanced**.

3. Select the options that you want.

4. Click **Save**.

### Unlocking Objects

The cloud service uses a checkout facility for cube objects (such as calculation scripts and rules files). Objects are locked automatically when they are in use and the locks are deleted when they are no longer in use.

You can view and unlock objects, according to your security role. Users with the Service Admin role can unlock any object. Users without the Service Admin role can unlock only those objects that they locked.

> **Note:**

Object locks and user locks on data are different.

See [Removing User Data Locks](#).

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click **Admin**.

2. Select the **Locks** tab, and click **Objects**.

3. Select the object and click **Unlock**.

### Removing User Data Locks

Data locks apply to block storage cubes only.

Occasionally, you may need to release a lock before the allotted time expires. For example, if you’re calculating a cube that has active locks on data, and the calculation
encounters a lock, then the calculation must wait. If you release the lock, the
calculation can resume.

You can always unlock data that you locked. To remove another user's data locks, you
must have the Application Manager or Database Manager role.

1. On the Applications home page, expand the application to show the cube row for
   the outline that you want to modify, then click Admin.

2. On the (Locks) tab, click Blocks.

3. Select the lock and click Unlock.

About Filters

Filters control security access to data values in a cube. Filters are the most granular
form of security available.

When you create a filter, you designate a set of restrictions on particular cube cells or
on a range of cells. You can then assign the filter to users or groups.

Your own security role determines if you can create, assign, edit, copy, rename, or
delete filters:

• If you have the Application Manager role, then you can manage any filter for any
  user or group. Filters do not affect you.
• If you have the Database Update role, then you can manage filters for the
  applications that you created.
• If you have the Database Manager role, then you can manage filters within your
  applications or cubes.
• If you have the Database Access role (default), then you have read access to data
  values in all cells, unless your access is further restricted by filters.

Creating Filters

You can create multiple filters for a cube. If you edit a filter, modifications made to its
definition are inherited by all users of that filter.

See Controlling Access to Database Cells Using Security Filters in Designing and
Maintaining Essbase Cubes.

1. On the Applications home page, expand the application to show the cube row for
   the outline that you want to modify, then click Security.

2. On the Filters tab, under New Filters, enter the name of the filter in Name.

3. Click Add Row.

4. For Access, select an access level:
   • None: No data can be retrieved or updated
   • Read: Data can be retrieved but not updated
   • Write: Data can be retrieved and updated
• MetaRead: Metadata (dimension and member names) can be retrieved and updated

Note:

The MetaRead access level overrides all other access levels. Additional data filters are enforced within existing MetaRead filters. Filtering on member combinations (using AND relationships) does not apply to MetaRead. MetaRead filters each member separately (using an OR relationship).

5. For **Member Specification**, specify member names.

   You can filter members separately, or you can filter member combinations. Specify dimension or member names, alias names, member combinations, member sets that are defined by functions, or substitution variable names, which are preceded by an ampersand (&). Separate multiple entries with commas.

6. Create additional rows for the filter as needed.

7. Click **Verify** to ensure that the filter is valid.

   If verification fails, fix the problem. For example, you might have misspelled a member or dimension name. For more information, download the application log. See **Downloading Server and Application Logs**.

8. Click **Save**.

   The filter is added to the list of filters in the left pane.

Note:

If filter rows overlap or conflict, more detailed cube area specifications apply over less detailed, and more permissive access rights apply over less permissive. For example, if you give a user Read access to Actual and Write access to Jan, then the user would have Write access to Jan Actual.

---

**Working with Attributes**

Attributes describe characteristics of data, such as the size and color of products. You can use attributes to group and analyze members of dimensions based on their characteristics. For example, you can analyze product profitability based on size or packaging, and you can make more effective conclusions by incorporating market attributes, such as the population size of each market region, into your analysis.

An attribute dimension such as Pkg Type contains members that describe the packaging characteristics of individual members of a nonattribute dimension. Nonattribute dimensions are called standard dimensions. When you associate an attribute dimension to a standard dimension, the standard dimension becomes the base dimension for the attribute dimension. For example, associating Pkg Type with the standard dimension Product makes Product the base dimension for the Pkg Type attribute dimension. An attribute dimension can have only one base dimension.
You must associate the individual members of the attribute dimension to members of its base dimension. For example, the Pkg Type dimension has two members, Bottle and Can. To enable analysis of products by their packaging, you must associate Bottle or Can to each member of the base dimension.

When using a rules file to build an outline, you can automatically define and build an attribute dimension and associate the members of the attribute dimension with members of the base dimension. For information about building attribute dimensions and associating attributes, see Working with Attributes in *Designing and Maintaining Essbase Cubes*.

When manually working with attributes, use the Outline page and the Edit Member Properties dialog box.

**Workflow for manually building attribute dimensions:**

1. **Create attribute dimensions.**
   
   See [Adding Dimensions to Outlines Manually](#).

   **Note:** In the outline, position attribute dimensions at the bottom.

2. **Tag the dimensions as attribute dimensions and set the attribute dimension type (text, numeric, Boolean, or date).**
   
   • Use the menu in the **Dimension Type** column to set the dimension as an attribute dimension.
   
   • Use the menu in the **Attribute Type** column to set the attribute dimension type.

3. **Add members to attribute dimensions.**
   
   See [Adding Members to Outlines Manually](#).

4. **Associate a standard dimension with an attribute dimension, thereby defining the base dimension of the attribute dimension. Use the menu in the **Base Dimension** column to associate an attribute dimension to a base dimension.**
   
   See [Setting Attribute Associations](#).

**Managing Outlines**

To navigate large outlines efficiently, you can search for members in the Outline page.

This image shows results of a search on the number 1, with the default search options, on a sample database. The results come from four different dimensions: Product, Ounces, Population, and Intro Date.
Searching an Outline

The default is to search the entire outline by member name, but you can select nondefault search options, such as searching a specific dimension, a nondefault alias table, or a UDA by clicking the Search drop-down menu.

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click Outline (Outline).
2. (Optional) In the Search field, click the drop-down menu and select from the available search options.

3. In the Search field, enter a search string and click Search or press the Enter key.

Notes:
- Search results are limited to one hundred rows.
- There is no wildcard search available.
- The string that you type in the search field must match the beginning of the member name (or alias or UDA name). For example, if you type -10, the member 100-10 will not be returned.
- The option to search for UDAs is available only if you are searching within a specified dimension.

Managing Outline Search Results

After searching an outline, you can manage and work with search results in the following ways:

- To view a member in context, with other members at the same level, click locate.
- When viewing search results, in the Outline Editor toolbar, click Outline to go back to the outline.
- To go back to the search results, in the Outline Editor toolbar, click Search Results.
- To edit member properties, from within search results:
  1. Select the member.
  2. Click Lock and Edit.
3. Click **Properties** to edit member properties.

**Note:**

Edits that you can make directly within search results are limited. To make additional changes, such as deleting or moving members, or adding a child or sibling member, click **locate**.
Loading Data

Loading data is the process of adding data values to a cube from a data source. A data source can contain data values, information about members (such as member names, member aliases, and formulas), generation and level names, data storage properties, attributes, and user-defined attributes (UDAs).

- About Loading Data
- Loading Data Using a Rules File
- Uploading Files to a Cube
- Managing Jobs
- Loading Dimensions and Data Using SQL
- Loading Dimensions and Data by Streaming from a Remote Database
- Loading Data and Dimensions from Data Sync to Essbase

About Loading Data

You must have the Database Update role to load data into a cube.

Note:

For all files that you upload and import to Essbase cloud, using Essbase or command line interfaces, their file name lengths are limited to 30 characters, including file extensions. You must rename file names accordingly before performing these operations. In addition, when you are building or changing cubes using application workbooks, the name of the workbooks must be no more than 30 characters.

Load data to a cube using one of the following methods:

- Load data using a flat data file and, if needed, a rules file.
  See Loading Data Using a Rules File.
- Load data values from an application workbook that you are using to build a cube.
  See Creating a Cube from an Application Workbook.
- Submit data values in Smart View.
  See About the Submit Data Options in the Oracle Smart View for Office User’s Guide.
Loading Data Using a Rules File

When you build a cube, data files and data load rules files are created in the cloud service. You can also use data files and data load rules files from a supported on-premises release of Essbase. See Uploading Files to a Cube.

1. Click Home.
2. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click Jobs.
3. Select New Job.
4. On the Execute Job dialog box, from the Job Type menu, select Data Load.
5. On the Database menu, select the cube that you are working with, if it is not already selected.
6. On the Scripts menu, select the data load rules file. If you are loading data, rather than dimensions, choose the rules file named Data.
7. For Load Type, choose SQL or File:
   • If you are loading SQL data, enter your user name and password. For more information, see Loading Dimensions and Data Using SQL.
   • If you are loading a flat data file, choose the file from the Data Files list. If the cube was created from an application workbook containing a data sheet, then the data file you should choose is named with the format Data_Cubename
8. Select the Abort on error check box if you want the data load to stop if an error occurs.
9. Click Execute.
10. Click Refresh to watch the status, and when it completes, scroll to the right of the Status column, and click Job Details.

11. Optional: If the data load succeeded, verify that input blocks were loaded.
   a. Close the Job Details page.
   b. Click Settings.
   c. Under Statistics, click Basic, and verify that level 0 blocks exist.
      To populate upper level blocks, run a Calc Execution job. See Executing Calculations.

12. Optional: Log in to the cube from Smart View and do an ad-hoc analysis to verify that the data was loaded.

### Uploading Files to a Cube

To perform some tasks, you may need to upload required files to a cube. For example, for data loads and dimension builds, you may need to upload text files and rules files.

1. On the Applications home page, expand the application to show the cube row for the outline you want to modify, then click Files.

2. On the Files page, click Upload and browse to the file that you want to upload.
   - calculation script (.csc)
   - text (.txt)
   - rules (.rul)
Managing Jobs

Jobs are operations such as loading data, building dimensions, exporting cubes, running MaxL scripts, running calculations, and clearing data. Jobs are asynchronous, meaning they are run in the background as a unique thread. Each job has a unique id.

Viewing Job Status

Because jobs are run in the background, you must refresh the Jobs page to view their status. Users have access to job listings based on the role they are assigned in the cloud service. For example, a user with the Service Administrator role can see all jobs; a user with the User role can see only those jobs that the user ran.

1. On the Applications home page select the application or cube for which you want to view job status and then click Jobs.

\[\text{Note:}\]

Service administrators can view the job status for all applications and cubes in the service instance by clicking Refresh to deselect applications and cubes and then clicking Jobs.

2. Click Refresh to refresh once, or click Auto Refresh to refresh every few seconds.

\[\text{Note:}\]

In Cube Designer, job status is refreshed automatically. See Viewing, Monitoring, and Troubleshooting Jobs in Cube Designer.
Changing the Default Number of Jobs that can Run Simultaneously

The default number of jobs that can run simultaneously is 10. You can adjust this number higher or lower.

1. On the Applications home page, click Jobs.
2. Click Settings.
3. In the Set maximum parallel jobs dialog box, use the arrows to adjust the number up or down.
4. Click Set maximum jobs.

Viewing Job History

You can also view details for an individual job. To view job details, click Job Details.

Job details include information such as script names, data file names, user names, number of records processed and rejected, and completion status.

Loading Dimensions and Data Using SQL

This task flow demonstrates how to import a table to an RDBMS server, create dimension build and data load rules, connect to the RDBMS, and load dimensions and data using SQL to an Essbase cube.

What you will need:

- Access to an RDBMS server
- Oracle SQL Developer, or another SQL integrated development environment
- A tabular data file. For example, in this exercise, consider a sample_basic_table.csv file that contains columns of data, as in the following abbreviated representation (many rows are omitted):
Access to an Essbase cloud service instance

A valid OCI or DSN-less connection string. For both OCI and DSN-less connections, you do not have to edit odbc.ini. Essbase will still make the connection using the ODBC driver. For examples of connection strings, see the table:

### Table 6-1 Connection Strings

<table>
<thead>
<tr>
<th>Connection Type</th>
<th>Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Call Interface (OCI)</td>
<td>$Keyword$DatabaseServerName:PortNumber/SID $OCI$mydsn01:1521/ORCL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>With ServiceName: ORACLESERVICE:oracle://HostName:PortNumber/ServiceName</td>
<td>With ServiceName: ORACLESERVICE:oracle://somedb99:1234/esscs.host1.oraclecloud.com</td>
</tr>
<tr>
<td>Microsoft SQL Server (DSN-</td>
<td>sqlserver://HostName:1433:DBName sqlserver://myMSSQLHost:1433:myDbName</td>
<td></td>
</tr>
<tr>
<td>less)</td>
<td>db2://HostName:PortNumber:DBName db2://myDB2Host:1234:myDbName</td>
<td></td>
</tr>
<tr>
<td>DB2 (DSN-less)</td>
<td>mysql://HostName:3306:DBName mysql://someHostName:3306:myDbName</td>
<td></td>
</tr>
</tbody>
</table>

Using the above resources, you can complete the following task flows:

- Loading Dimensions Using SQL
- Loading Data Using SQL
Notes

No members from a CellProperties dimension should be included in the tabular data or in the headers of the SQL-based load rules files.

Loading Dimensions Using SQL

This task flow demonstrates how to import a table to a RDBMS server, create dimension build load rules, connect to the RDBMS, and load dimensions using SQL to a cube.

1. In Oracle SQL Developer (or your choice of SQL tool), import a table, from a flat file (for example, sample_basic_table.csv), to your SQL database server connection.

   In Oracle SQL Developer, the imported table, SAMPLE_BASIC_TABLE, might look similar to the following:

   ![SAMPLE_BASIC_TABLE Table](image)

   Next, you will delete some members from Sample Basic, and then create a load rule to rebuild the Market dimension from the SQL table.

2. In the cloud service, in the Applications home page, expand the Sample application, and select the cube, Basic.

3. Click Outline.

4. Click the Market dimension, and then click member East.

5. Select Lock and Edit, and delete some of the states from the East market. For example, delete Connecticut, New Hampshire, and Massachusetts.
6. Click **Save and Unlock**, and then verify that East now only contains the states Florida and New York.

Next, you will create load rules and repopulate the Market dimension, from the SQL table, with the states you have removed.

7. Click **Scripts**, then choose the Rules Editor tab.

8. Click **Create** to begin defining new load rules.

9. On the drop-down menu next to **Verify**, select **Dimension Build**.

10. In the **Name** field, change the name of the rules file to MarketSQLDimbuild.

11. Click the **Dimensions** tab.

12. Click the arrow next to **Add**, and choose **From Outline**.

13. Select Market and click **OK**.

14. Click the **Fields** tab, and click **Create** (do not click the arrow next to **Create**).

15. In field #1, add a generation named Region. Associate it with the Market dimension, specify the generation level as 2, and select **Generation** as the Type.

   The Market dimension is generation 1, and you are adding a child named Region.

16. Click **Create** again and add a second load rule field named State, associated with Market, at generation 3.

17. Click the **Data Source** tab.
18. Enter the valid connection string.
   a. For OCI connections: In the **Name** field of the **Properties** group, enter the
      valid OCI connection string.

   b. For DSN-less connections, such as Oracle DB, Microsoft SQL Server, and
      DB2: You must leave the **Name** field of the **Properties** group empty. Instead,
      enter the connection string in the **Server** field of the **SQL Properties** group.

19. In Oracle SQL Developer (or your alternate SQL tool of choice), write and test a
    SELECT statement selecting some columns from the table

    ```
    SAMPLE_BASIC_TABLE: Select distinct market, statename from SAMPLE_BASIC_TABLE
    ```

20. If the SQL query is valid and returns a result in your SQL tool, copy all of the
    statement, except for the SELECT keyword, to your clipboard. The results of this
    query are the dimensions you will load into the Sample Basic cube.

21. Back in the **Data Source** tab for your dimension build load rule, paste the SQL
    statement, except for the SELECT keyword, into the **Select** field of the **SQL
    Properties** group.

22. Click **Save** to save the MarketSQLDimbuild rule, and then click **Close**.

    Next, you will load the dimensions from Jobs.

23. Click **Jobs**, and click **New Job**.

24. Select **Dimension Build** as the job type, and **SQL** as the load type.

25. From the **Scripts** drop-down list, select the name of the dimension build rules file
    you created, MarketSQLDimbuild.

26. Enter the user name and password of one of your SQL database schema users.

27. From the **Restructure Options** drop-down list, select **Preserve All Data**.

28. Click **Execute**.

    The dimension build begins. Click **Refresh** to watch the status, and when it
    completes, scroll to the right of the Status column, and click **Job Details**.
29. Click **Outline**, and verify that your dimensions were built (verify that Connecticut, New Hampshire, and Massachusetts exist as children under East).

Loading Data Using SQL

This task flow demonstrates how to clear data from a cube, create data load rules, load data (using SQL) from an RDBMS server, and verify in Smart View that the data was loaded.

1. Complete step 1 from **Loading Dimensions Using SQL**, if you haven't already. Next, you will execute a calculation script that clears sales data from the cube, and then load the data again from a table.

2. In the cloud service, in the Applications home page, expand the Sample application, and select the cube, Basic.

3. Click **Jobs**, and click **New Job**.

4. Select **Clear Data** as the job type, and click **Execute**.

5. Click **Yes**. The clear data job begins. Click **Refresh** to watch the status, and when it completes, log in to the database from Smart View.

6. In Smart View, zoom in and keep only on all the dimensions, to see that level zero sales data was cleared. For example:

<table>
<thead>
<tr>
<th>Actual</th>
<th>Connecticut</th>
<th>Cola</th>
<th>Jan</th>
<th>#Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Budget</td>
<td>Connecticut</td>
<td>Cola</td>
<td>Jan</td>
<td>#Missing</td>
</tr>
<tr>
<td>Varianc</td>
<td>Connecticut</td>
<td>Cola</td>
<td>Jan</td>
<td>#Missing</td>
</tr>
<tr>
<td>Varianc</td>
<td>Connecticut</td>
<td>Cola</td>
<td>Jan</td>
<td>#Missing</td>
</tr>
<tr>
<td>Scenario</td>
<td>Connecticut</td>
<td>Cola</td>
<td>Jan</td>
<td>#Missing</td>
</tr>
</tbody>
</table>

Keep the worksheet open. Next, you will create load rules and repopulate the Sales data from the SQL table.

7. Click **Scripts**, then choose the Rules Editor tab.

8. Click **Create** to begin defining new load rules.

9. On the drop-down menu next to **Verify**, ensure that Data Load is selected.

10. In the **Name** field, change the name of the rules file to SalesSQLDataload.

11. In the **Load Option** field, select **Add**.

12. Skip the **Dimensions** tab, and click the **Fields** tab.

13. Click **Create** (do not click the arrow next to **Create**).

14. In Oracle SQL Developer (or your alternate SQL tool of choice), write and test a SELECT statement selecting some columns from the table `SAMPLE_BASIC_TABLE`: Select Product, Year, Scenario, Statename, Sales from `SAMPLE_BASIC_TABLE`

15. Ensure that the SQL query is valid and returns a result in your SQL tool. The results of this query are the data you will load into the Sample Basic cube.

16. Note the order of dimensions as listed in your SQL query. You must follow the same order of dimensions as you define the load rule fields.
17. In the cloud service, back in the **Fields** tab for your data load rule, enter Product as the name of field #1 (because Product is listed first in the SQL query).

18. Click **Create**.

19. Enter Year as the name of field #2 (because Year is listed second in the SQL query).

20. Click **Create**.

21. Enter Scenario as the name of field #3, and click **Create**.

22. Enter Market as the name of field #4, and click **Create**.

23. Enter Sales as the name of field #5.

24. Click the **Data Source** tab.

25. Enter the valid connection string.

   a. For OCI connections: In the **Name** field of the **Properties** group, enter the valid OCI connection string.

   ![Properties](image1)

   b. For DSN-less connections, such as Oracle DB, Microsoft SQL Server, and DB2: You must leave the **Name** field of the **Properties** group empty. Instead, enter the connection string in the **Server** field of the **SQL Properties** group.

26. Copy all of the SQL statement except for the SELECT keyword, and paste it into the **Select** field of the **SQL Properties** group.

   ![Select](image2)

27. Click **Save** to save the SalesSQLDataload rule, and then click **Close**.

   Next, you will load the data from Jobs.

28. Click **Jobs**, and click **New Job**.

29. Select **Data Load** as the job type, and **SQL** as the load type.

30. From the **Scripts** drop-down list, select the name of the data load rules file you created, SalesSQLDataload.

31. Enter the user name and password of one of your SQL database schema users.

32. Click **Execute**.

   The data load begins. Click **Refresh** to watch the status, and when it completes, scroll to the right of the **Status** column, and click **Job Details**.

33. Go back to the worksheet in Smart View, and refresh it to verify that the data was loaded from the SQL table.
If the data or dimensions you want to load to a cube are in a remote database, you can use the stream option in the CLI to push the data or members to your cube, using a rules file.

When you use the `stream` option for the CLI `dataloader` or `dimbuild` command, you must also reference a saved JDBC connection that reflects your driver and connection strings.

**Prerequisites**

1. The rules file must exist in the Files section for the relevant database.
2. The database query used to load data or build dimensions must have the same dimensionality as the columns in the rules file. (For example, see Loading Dimensions Using SQL, where the order of dimensions in the rules file must match the order of dimensions in the SQL query).

**Limits**

- Substitution variables are not supported in SQL statements used in load rules.
- Only use SQL functions that are supported by JDBC. ODBC scalar functions are not supported in CLI.

**Steps for Streaming Option**

To stream data load or dimension build,

1. Create a saved JDBC connection string that reflects your data source’s driver and connection strings, using the CLI `createlocalconnection` command.
2. (Not required for Oracle database) Set an environment variable `EXTERNAL_CLASSPATH` to point to the .jar file for your database driver. See `EXTERNAL_CLASSPATH` Examples.
3. Run the CLI `dataloader` or `dimbuild` command with the streaming option, providing the saved connection name.

You can optionally specify the database query in the `dataloader` or `dimbuild` command. Otherwise, you can specify it in the load rules, in the `Select` section of the `Data Source` tab. For examples, see Loading Dimensions Using SQL and Loading Data Using SQL.
EXTERNAL_CLASSPATH Examples

Setting the EXTERNAL_CLASSPATH environment variable is required before you can stream from any data source other than Oracle database. Set the variable to point to the location of your database driver .jar file.

**DB2**

Set the external classpath variable to point to the location of the DB2 driver jar file.

**C Shell Example**

```
setenv EXTERNAL_CLASSPATH /scratch/db/jars/db2jcc.jar
```

**Korn or Bash Shell Example**

```
export EXTERNAL_CLASSPATH=/scratch/db/jars/db2jcc.jar
```

**MySQL**

Set the external classpath variable to point to the location of the MySQL driver jar file.

**C Shell Example**

```
setenv EXTERNAL_CLASSPATH /scratch/db/jars/mysql-connector-java-5.1.43-bin.jar
```

**Korn or Bash Shell Example**

```
export EXTERNAL_CLASSPATH=/scratch/db/jars/mysql-connector-java-5.1.43-bin.jar
```

**Microsoft SQL Server**

Set the external classpath variable to point to the location of the SQL Server driver jar file.

**C Shell Example**

```
setenv EXTERNAL_CLASSPATH /scratch/db/jars/sqljdbc4-3.0.jar
```

**Korn or Bash Shell Example**

```
export EXTERNAL_CLASSPATH=/scratch/db/jars/sqljdbc4-3.0.jar
```

**Teradata**

Set the external classpath variable to point to the location of both Teradata driver jar files.

**C Shell Example**

```
setenv EXTERNAL_CLASSPATH /scratch/db/jars/tdgssconfig.jar:/scratch/db/jars/terajdbc4.jar
```

**Korn or Bash Shell Example**

```
export EXTERNAL_CLASSPATH=/scratch/db/jars/tdgssconfig.jar:/scratch/db/jars/terajdbc4.jar
```
Loading Data and Dimensions from Data Sync to Essbase

Data Sync is a data replication tool that you can use to load data into an Essbase cube for analysis in Oracle Analytics Cloud. The following workflows apply when you need to load dimensions and data from Data Sync to Essbase.

Workflow in Essbase

Complete following steps in Essbase:

1. Create the Essbase application and cube.
2. Set up dimension load rules files for each dimension, to dictate how the dimensions should be loaded to the Essbase cube. See Loading Dimensions from Data Sync to Essbase.
3. Set up a data load rules file to dictate how the measures data should be loaded to the Essbase cube. See Loading Data from Data Sync to Essbase.

If you already have appropriate dimension and data load rules you can use, skip the rules file creation steps. Be sure you have uploaded the rules files to Oracle Analytics Cloud - Essbase.

Workflow in Data Sync

Complete following steps in Data Sync:

Prepare the dimensions and data as described in Typical Workflow for Loading Data to Oracle Essbase in Preparing Data in Oracle Analytics Cloud. The following is a summary of the workflow:

1. Using manual entry, select Essbase as a target.
2. Write a SQL query in which the columns match the order of the columns defined in the Essbase load rule.
   
   For example, for the Product dimension, if the Essbase rules file lists the order of fields as PRODUCT_NUMBER, PRODUCT_NAME, PRODUCT_TYPE, then the order of selection of columns from your SQL query should be the same:

   ```sql
   select PRODUCT_NUMBER, PRODUCT_NAME, PRODUCT_TYPE from PRODUCTS
   ```

3. Define incremental load strategy options.
4. Define more information about the Essbase target. Provide Essbase specific properties in the Target Tables tab under Pluggable Attributes. Give the application and cube name that you are loading to in Essbase. Provide the Load Type (Dimension Build or Data Load), the object name, and the rules file name that you created in Essbase.
5. Summarize the load objects into groups, in the Project Summary tab. In Data Sync, you define a single group for loading data into one Essbase cube, and within that group, you define a sequence of load steps. For example, if you have five dimensions to load, your group will have a defined sequence of five steps.

   Create one group per cube loaded. Ensure that the Dimension Build type data flows happen first, followed by the Data Load type. In addition, all the data flows must be run serially; for example, no two data flows within a group should be run in parallel.
6. Run the job. If the job hangs or an error occurs, restart the Essbase application and run the job again.

Loading Dimensions from Data Sync to Essbase

This topic tells you how to create a rules file to load dimensions from Data Sync to Essbase. Create one rules file for each dimension you will load to Essbase.

1. In Essbase, on the Applications home page, expand the application, and select the cube.

2. Click Scripts, and then choose the Rules Editor tab.

3. Click Create to begin defining new load rules.

4. On the drop-down menu next to Verify, select Dimension Build.

5. In the Name field of the Properties tab, give a name to the load rule.

6. On the Dimension tab, add a dimension name corresponding to the dimension you will load into Essbase.

7. On the Fields tab, click Create (do not click the arrow next to Create).

8. For the fields, add the column names in order. List the columns in same order as you define the columns in the SQL query you write from Data Sync.

9. Skip the Data Source tab. You provide all your connection details from Data Sync.

10. Save and close the load rule. Now you are ready to load a dimension from Data Sync to Essbase.

Loading Data from Data Sync to Essbase

This topic tells you how to create a data load rules file to load from Data Sync to Essbase.

1. In Essbase, on the Applications home page, expand the application, and select the cube.

2. Click Scripts, and then choose the Rules Editor tab.

3. Click Create to begin defining new load rules.

4. On the drop-down menu next to Verify, select Data Load.

5. In the Name field of the Properties tab, give a name to the load rule.

6. On the Dimension tab, add a dimension name corresponding to the measures dimension that contains the data you will load into Essbase.

7. On the Fields tab, click Create (do not click the arrow next to Create).

8. For the fields, add the column names. List the columns in the same order as they are presented in the SQL query you write from Data Sync.

9. Skip the Data Source tab. You provide all your connection details from Data Sync.

10. Save and close the load rule. Now you are ready to load data from Data Sync to Essbase.
Calculating Cubes

A cube contains two types of values: values that you enter, called input data, and values that are calculated from input data. A cube can be calculated using one of two methods. Outline calculation, which is the simplest calculation method, bases the calculation of a cube on the relationships between members in the database outline and on any formulas that are associated with members in the outline. Calculation script calculation lets you decide exactly how to calculate a database; for example, you can calculate part of a database or copy data values between members.

- About Calculating Data
- Creating Calculation Scripts
- Executing Calculations
- Creating and Using Substitution Variables
- Setting Two-Pass Calculation Properties
- Tracing Calculations

About Calculating Data

If you have the Database Update role, you have access to run the default calculation on the cube, but no access to run any specific calculation scripts. If you have the Application Manager or Database Manager role, you have Calc privileges and rights to execute all calculations.

Access to execute specific calculations can be granted in the Provisioning page. See Provisioning Roles and Provisioning Users.

Creating Calculation Scripts

Calculation scripts specify how cubes are calculated and, therefore, override outline-defined cube consolidations. For example, you can calculate cube subsets or copy data values between members. See Developing Calculation Scripts for Block Storage Databases in Designing and Maintaining Essbase Cubes.

On the Scripts page, you create calculation scripts from calculation commands, equations, and member formulas. Because calculation scripts are text files, you can create calculation scripts in any text editor and paste them onto the Scripts page.
**Note:**

Using Point of View (POV) for runtime substitution variables prevents a calculation script from being validated, even if it is valid. However, a valid script will run.

Validating a script verifies the script syntax. For example, incorrectly spelled function names and omitted end-of-line semicolons are identified. Validation also verifies dimension names and member names.

Calculation scripts do not apply to aggregate storage applications.

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click **Scripts**.
2. Create your calculation script on the Scripts page, or create it in any text editor and paste it onto the Scripts page.
3. Click **Validate**.
4. Correct any validation errors.
5. Click **Save**.

**Executing Calculations**

After creating and saving calculation scripts, you execute them on cubes to perform the calculations. You can execute calculation scripts in the cloud service or from Smart View.

To execute calculation scripts from Smart View, see Calculating Data in Essbase in Oracle Smart View for Office User’s Guide.

1. Create your calculation script, or upload an existing script.
   - See Creating Calculation Scripts and Uploading Files to a Cube.
2. Click **Home**.
3. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click **Jobs**.
4. On the Jobs page, click **New Job**.
5. On the **Execute Job** dialog box, from the **Job Type** menu, select **Calc Execution**.
6. On the **Database** menu, select the cube that you are calculating.
7. On the **Scripts** menu, select a calculation script.
8. Click **Execute**.
Creating and Using Substitution Variables

Substitution variables act as global placeholders for information that changes regularly. You create the variable and a corresponding string value, and the value can then be changed at any time.

You can use substitution variables in calculation scripts. Typically, you use substitution variables to reference information that changes frequently.

When you execute scripts that include substitution variables, each variable is replaced with the value that is specified in the variable definition.

**Note:**

Runtime substitution variables are different from substitution variables in that every runtime substitution variable used in a calculation script must be declared in the SET RUNTIMESUBVARS calculation command. See Using Runtime Substitution Variables in Calculation Scripts Run in Essbase in Designing and Maintaining Essbase Cubes.

1. On the Applications home page, expand the application to show the cube row for the outline you want to modify, then click **Admin**.
2. On the **Variables** tab, click **Create**.
3. On the Create Variables dialog box, enter a variable name and variable value and click **Create**.

9. Click **Refresh** to see the status of your calculation.
Setting Two-Pass Calculation Properties

The Two-Pass Calculation property indicates which members need to be calculated twice to produce the desired value. To obtain the correct values for two-pass members, the outline is calculated, and then members that are dependent on the calculated values of other members are recalculated.

Even though two-pass calculation is a property that you can give to any non-attribute dimension member, it works only on members of the Accounts dimension and Dynamic Calc members. If two-pass calculation is assigned to any other member, it is ignored.

Two-pass calculations are supported only on block storage cubes. Hybrid aggregation and aggregate storage cubes use member solve order, instead of two-pass calculation, to control when members are calculated.

1. On the Applications home page, expand the application to show the cube row for the outline you want to modify, then click **Settings**.
2. Select **Properties**.
3. Select **Edit Advanced Properties**.
4. Select **Two-Pass Calculation**.

Tracing Calculations

You can use calculation tracing to analyze member formula processing, and refine your calculation scripts.

Calculation tracing enables you to access logged information about a calculation, after the calculation script successfully executes against a cube.

Tracing a calculation does not change anything about calculation behavior. If a calculation is launched in Smart View, and the connected server has calculation tracing enabled by an administrator, Smart View displays a pop-up dialog box containing details, after the calculation runs. The calculation tracing information can be pasted from the pop-up dialog into a text editor. Or, you can find the same information in `calc_trace.txt`, located in the database files directory on the cloud service.

The calculation tracing information can help you debug calculation script execution, in case the results of the calculation are not what you expected.

To enable calculation tracing, the administrator must first turn on the CALCTRACE application configuration parameter. Then, Smart View users can select data cells to trace.

Outside of Smart View, you can also use the SET TRACE calculation command in calculation scripts to select data cells to trace. SET TRACE enables you to trace multiple data cells. Additionally, you can trace sections of calculation scripts by using a combination of `SET TRACE mbrList` and `SET TRACE OFF`. However, to use SET TRACE command, you must execute the calculation script outside of Smart View, using Cube Designer or the Jobs page of the cloud service.

Calculation tracing is not supported on applications with scenario management enabled.
Example

The following calculation script is run on Sample Basic. The script includes a SET TRACE command, requesting detailed information to be logged for the data intersection (cell) that represents budgeted January sales, in the California market, for the product SKU number 100-10.

```plaintext
SET TRACE ("100-10", "California", "Jan", "Sales", "Budget");
FIX("California", "Budget")
  "Sales" {
    "100-10" = @MEMBER(@CONCATENATE(@NAME(@PARENT("Product")), ","-20")) / 10;
  };
ENDFIX;
```

Sample Basic has two sparse dimensions: Product and Market. The member formula is on "Sales," a member of Measures, which is a dense dimension. The FIX statement's member list only contains one sparse member, "California," which belongs to the Market dimension.

The number of existing blocks in the FIX statement determines the number of times the traced cell is calculated. In this example, the calculation cycles through all existing sparse member combinations of "California." Each of these combinations represents a block.

After the calculation completes, the following tracing information is logged and displayed:

```
Tracing cell: [100-10][California][Jan][Sales][Budget]  (Cell update count: 1)
Previous value: 840.00
Dependent values:
  [100-20][California][Jan][Sales][Budget] = 140.00
New value: [100-10][California][Jan][Sales][Budget] = 14.00
Computed in lines: [91 - 93] using:
"Sales"(
"100-10"=@MEMBER(@CONCATENATE(@NAME(@PARENT("Product")),","-20"))/10;
)
```

```
Tracing cell: [100-10][California][Jan][Sales][Budget]  (Cell update count: 2)
Block from FIX scope: [100-30][California]
Actual block used in calculation: [100-10][California]
Previous value: 14.00
Dependent values:
  [100-20][California][Jan][Sales][Budget] = 140.00
New value: [100-10][California][Jan][Sales][Budget] = 14.00
Computed in lines: [91 - 93] using:
"Sales"(
"100-10"=@MEMBER(@CONCATENATE(@NAME(@PARENT("Product")),",-20"))/10;
)
```

```
Tracing cell: [100-10][California][Jan][Sales][Budget]  (Cell update count: 3)
Block from FIX scope: [200-10][California]
Actual block used in calculation: [100-10][California]
Previous value: 14.00
Dependent values:
  [200-20][California][Jan][Sales][Budget] = 520.00
New value: [100-10][California][Jan][Sales][Budget] = 52.00
Computed in lines: [91 - 93] using:
"Sales"(
"100-10"=@MEMBER(@CONCATENATE(@NAME(@PARENT("Product")),",-20"))/10;
)
```
The calculation tracing log provides the following insights about how the calculation worked, on the cell that was traced:

- The traced cell was calculated several times (the reported cell update count stops at 8).
- The value of the cell, before calculation, was 840.00.
- For each calculation occurrence, dependent values and new values are shown. Dependent values come from the member formula in the FIX statement.
- The final value of the traced cell, after all calculation completes, is 9.
- Lines 91-93 of the calculation script, containing a member formula on Sales, are responsible for the updated values.

For each of the blocks cycled through, “Sales” is calculated using the formula:

"100-10"/@MEMBER(@CONCATENATE(@NAME(@PARENT("Product")),"-20"))/10

The formula contains a sparse member on the left hand side, which could cause the actual calculation block to be different than the initial FIX block. For example, when the calculation cycles through "California"->"100-20", the calculations are actually done in "California"->"100-10".

The trace log entries entitled “Block from FIX scope” and “Actual block used in calculation” are only printed if there is a discrepancy between the blocks in the FIX statement, and the block that is represented in the member formula. These log entries can provide indications as to why there are duplicate calculations, helping you to debug your calculation scripts.

See Also

CALCTRACE and SET TRACE in Technical Reference for Oracle Analytics Cloud - Essbase

Oracle Smart View for Office User's Guide
Modeling Data in Private Scenarios

Using scenario management, scenario participants can perform what-if analysis to model data in their own private work areas. These scenarios can optionally be subject to an approval workflow which includes a scenario owner and one or more approvers. In the workflow, scenario owners merge scenario data with the final cube data only after it is approved.

- Understanding Scenarios
- Workflow for Scenarios
- Enabling Scenario Modeling
- Using Scenarios

Understanding Scenarios

Scenarios are private work areas in which users can model different assumptions within the data and see the effect on aggregated results, without affecting the existing data. Each scenario is a virtual slice of a cube in which one or more users can model data and then commit or discard the changes.

Scenario-enabled cubes have a special dimension called Sandbox. The sandbox dimension is flat, with one member called Base and up to 1000 other members, commonly referred to as sandbox members. All members in the sandbox dimension are level-0. Sandbox members are named sb0, sb1, and so on. Each sandbox is a separate work area, whereas the Base holds the data currently contained in the cube.

Base data is the starting point before you use the sandbox to model possible changes. Sandbox data (also known as scenario data) is not committed unless you apply it, at which point it overwrites the Base data.

Note:

Security and filters apply to the Sandbox dimension.

Sandbox
  Base
  sb0
  sb1
  sb2

When first created, sandbox member intersections are all virtual and have no physical storage. The physical data from the cube is stored in the Base member slice. Querying any of the other sandbox members dynamically reflects the values stored in the Base. Only after you update any of the values in a sandbox are your changes stored physically in the sandbox. After you update some values in a sandbox member, queries against the sandbox reflect a mixture of stored sandbox values and values inherited dynamically from the Base. Changes made in a sandbox are not committed...
to the Base until you do so explicitly, generally after an approval workflow. See Workflow for Scenarios.

After you’re finished with the sandbox, you can put the sandbox through the approval workflow, or you can skip the workflow and commit the updated values to the Base, or reject and discard the sandbox changes.

**Note:**

- You must enable hybrid aggregation for scenario management to work. For queries, it is enabled by default. Do not disable it. For calculations, you need to enable the HYBRIDBSOINCALCSCRIPT application configuration. See HYBRIDBSOINCALCSCRIPT in *Technical Reference for Oracle Analytics Cloud - Essbase*.
- Scenario enabled cubes have a CellProperties dimension that you should ignore, as it is for internal processes. You do not need to modify it nor account for it in calculations, queries, or load rules, and it shouldn’t be included in any calculations or other operations.

**Understanding Scenario Calculations**

If a calculation script doesn't fix on any sandbox member, it implicitly fixes on the Base member.

This behavior is different from calculations on non-sandbox dimensions excluded from the fix, which would include all dimension members in their scope. Sandbox dimensions are calculated differently, as the intent is usually to calculate either Base or specific sandboxes at a given time. Essbase calculates the Base member values, rather than the working sandbox values, except:

- When the calculation fixes on particular sandbox members.
- When the calculation is executed from a sheet launched from a scenario in the web interface. See “From the Essbase Web Interface” in *Viewing a Scenario in Smart View*. This is called a scenario-launched sheet.

**Note:**

A scenario launched sheet is an Excel sheet launched from a scenario in the web interface. See "From the Essbase Web Interface" in *Viewing a Scenario in Smart View*.

If you execute a calculation script from a scenario-launched sheet, the calculation runs in the sandbox associated with the scenario as long as no sandbox is explicitly mentioned in the script. If you intend to calculate in your scenario's sandbox, you should not fix on a sandbox in your script.

**Understanding Scenario Limitations**

- Scenarios are not supported on aggregate storage cubes.
- The following calculation functions are not supported in scenario-enabled cubes:
Note:
There are functions that are not supported in hybrid aggregation mode, which is used with scenario-enabled cubes. See Functions Supported in Hybrid Aggregation Mode in Technical Reference for Oracle Analytics Cloud - Essbase.

- The DATAEXPORT calculation command isn’t supported on sandbox members. It is only supported on the Base member.
- CALC DIM and calculation expressions that calculate a sub-hierarchy, such as @CHILDREN or @IDESCENDANTS are not supported. However, using @IDESCENDANTS in FIX statement or other formula works.
- When you connect to a scenario from a scenario-launched sheet, MDX queries, MDX inserts, and MDX exports will work with the base instead of working with the sandbox for that scenario.
- Partitions are not supported.
- Runtime substitution variables with the svLaunch parameter are not supported when you launch the scenario in Smart View from the web interface. See Viewing a Scenario in Smart View.
- Runtime substitution variables with the svLaunch parameter work correctly when you connect to the scenario directly from a Smart View private connection. This is because the sandbox member is included in the sheet.

Workflow for Scenarios

You can review a scenario using an optional approval workflow. Alternatively, when working with a scenario, you can change data values in the scenario and commit data changes to the cube (or reject them), without going through an approval process.

Note:
Scenario-enabled cubes must be fully dynamic block storage cubes.
Enabling Scenario Modeling

Enabling scenario modeling as part of the cube creation process is as easy as selecting a check box in the user interface or populating the right fields in the scenario-based application workbook.

You can create or enable a cube for scenario modeling using one of the following methods:

- Creating a Scenario-Enabled Cube
- Creating a Scenario-Enabled Sample Cube
- Enabling an Existing Cube for Scenario Management

Also see:

- Creating Additional Sandbox Members

Note:

Data Audit Trail is not supported on scenario-enabled cubes.
Creating a Scenario-Enabled Cube

1. On the Applications home page, click Create.
2. On the Create application/cube dialog box, enter an application name and a cube name.
3. Ensure that Use aggregate storage is not selected.
4. Select Enable scenario management.
5. For Scenario count, enter a number for the number of scenario-based members that you want. The default is 100.
6. Click Create.

Creating a Scenario-Enabled Sample Cube

You can create a scenario-enabled cube by importing the scenario-enabled sample application workbook.

1. On the Applications home page, click Templates.
2. Download the application workbook labeled Block Storage Sample (Scenarios) to your local drive.
3. Open the workbook, Sample_Basic_Dynamic_with_Scenario.xlsx.
4. On the Applications home page, click Import.
5. On the Import Cube - Excel File dialog box, browse to the location of the application workbook on your local drive.
6. Enter an application name and cube name.
7. Click Deploy and Close.
Enabling an Existing Cube for Scenario Management

If you have the application manager role, you can enable an existing cube to use scenario modeling. It is best to do so on a copy of the original cube. Existing scripts, rules, and queries will work as before, when used on the base member. If you need to run them on a scenario member, you'll need to make changes to accommodate the scenario member or run them from a scenario-launched sheet.

**Note:**

A scenario launched sheet is an Excel sheet launched from a scenario in the web interface. See "From the Essbase Web Interface" in Viewing a Scenario in Smart View.

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click **Scenario**.
2. Click the **Administration** tab.
3. Click **Create Sandbox**.
4. On the Create sandbox dimension dialog box, enter a **Member count** to indicate the sandbox member count. The default is 100. (This number can be changed at a later time.)

5. Click **OK**.

Creating Additional Sandbox Members

By default, a new scenario has 100 sandbox members. You can create additional sandbox members (up to 1000).

1. On the Applications home page, expand the scenario-enabled application to show the cube row for the outline that you want to modify, then click **Scenario**.
2. Click the **Administration** tab.
3. Click **Add sandboxes**.
4. On the Add sandboxes dialog box, change the member count to the total number of sandboxes you want.
Using Scenarios

After you create a scenario-enabled cube, you can create scenarios, model data, and submit the scenario for approval.

See:

- Creating a Scenario
- Modeling Data
- Submitting a Scenario for Approval
- Approving or Rejecting Scenario Changes
- Applying or Discarding Data Changes
- Viewing Base Member Data
- Viewing a Scenario in Smart View
- Comparing Scenario Values to Base Values
- Reverting Scenario Values Back to Base Values
- Setting Scenario Cells to #Missing
- Deleting the Scenario

Creating a Scenario

1. On the Applications home page, expand the scenario-enabled application to show the cube row for the outline that you want to modify, then click Scenario.

2. On the Workflow tab, click Add.

3. On the General Information tab, enter a Name, Due Date, Priority, and (optional) Description. When you run calculation scripts on scenarios, by default calculated values are not merged to base data. If you select Use calculated values, calculated values are merged to base data.
4. Click the **Approvers** tab.

   Approvers monitor the process, and approve or reject scenarios. They must have Database Access or higher role. Scenarios can have multiple approvers, in which case each one must approve the scenario before it can be submitted.

   Approvers are not mandatory. The scenario owner can change data values in the scenario and commit data changes to the cube (or reject them), without going through an approval process.

5. Click **Search** to view available users and then use the arrows to move users from the **Available Users** column to the **Approvers** column.

6. Click the **Participants** tab.

   Participants can participate in a what-if analysis. They must have Database Update or Database Access user role. Adding participants is not mandatory. The scenario owner can change data values in the scenario and commit data changes to the cube (or reject them) without designating participants.

7. Click **Search** to view available users and then use the arrows to move users from the **Available Users** column to the **Participants** column.

8. Click **Save and Close**.

**Modeling Data**

As a scenario user, you can model data slices in your own scenario.

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click **Scenario**.
2. On the Workflow tab, start Smart View by clicking the Excel icon before the scenario name.
   You can also start Smart View from outside of the cloud service.

3. Make data changes and view the aggregations in Smart View.
   If you change and submit values and decide you want to go back to the base values, you can revert to the base by typing #Revert in the changed cells and choosing Submit Data on the Smart View Essbase ribbon.
   If a cell in the base has a value, and you want the corresponding cell in the scenario to be #Missing, you can send #Missing to the scenario or you can delete the value in Smart View and select Submit Data on the Smart View Essbase ribbon.

4. Continue this process until you’re ready to submit data for approval. See Submitting a Scenario for Approval.

Submitting a Scenario for Approval

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click Scenario.
2. On the Workflow tab, click Actions, and then select Submit.
3. (Optional) Enter a comment.
4. Click OK.

Approving or Rejecting Scenario Changes

As an Approver for a scenario, you can approve or reject scenario changes, and the scenario owner is notified of the action. After you approve a scenario, the owner can apply the data to the cube.

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click Scenario.
2. On the Workflow tab, select the scenario you want to approve.
3. Click Actions, then select Approve.
4. Enter a comment on the Approve Scenario dialog box.

Applying or Discarding Data Changes

When you apply data changes, the changes stored within the scenario overwrite the base data.

You can apply or discard changes from the Scenario page. You can also apply changes using the DATAMERGE calculation command.

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click Scenario.
2. On the Workflow tab, select the scenario in which you want to apply or discard changes.
3. Click **Actions**, and then select **Apply** or **Discard**. Apply runs the DATAMERGE calculation command with NOCALC and OVERWRITE keywords. Discard clears the data from the scenario.

4. When prompted, confirm your selection.

You can also apply data changes using the DATAMERGE calculation command. See *Technical Reference for Oracle Analytics Cloud - Essbase*.

After the scenario is submitted for approval, internally created security filters enforce read-only access to the scenario for the scenario owner and user, and no access for other users.

### Viewing Base Member Data

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click **Scenario**.

2. On the **Workflow** tab, click **Actions**, and then select **Show base data**.

### Viewing a Scenario in Smart View

You can look at a scenario in Smart View from the Essbase web interface, or from a Smart View private connection.

**From the Essbase Web Interface**

You can launch Smart View from a scenario in the web interface. When you do this, because you enter from the scenario, you can only work in Smart View in the sandbox member associated with the scenario from which you entered. The sandbox member is implicit. You do not see it in the Smart View grid.

<table>
<thead>
<tr>
<th>Note:</th>
</tr>
</thead>
<tbody>
<tr>
<td>A specific scenario is associated with exactly one sandbox member.</td>
</tr>
</tbody>
</table>

1. In a web browser, log in to Essbase.
2. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click **Scenario**.
3. On the **Workflow** tab, click the Excel icon next to the scenario you want to view.
4. Select to open the file.
5. This launches Excel with a Smart View connection to the scenario.

When you do this, the slice of data for that specific scenario is in the worksheet. You can submit data only to that scenario (When you submit data to a scenario, you are submitting data to one sandbox member).
From a Smart View Private Connection

You can open Excel and make a private connection to your cube, without starting from the web interface:

When you do this, the sandbox dimension will be in the worksheet, so you can submit data to any sandbox member to which you have access. This is helpful when you are a participant in more than one scenario, but you must explicitly know which sandbox you want to work in.

**Note:**

To see which sandbox member is associated with a scenario, on the scenarios page, select the scenario, choose **Edit**, and view the **General Information** tab.

1. Open Excel.
2. Make a private connection to your scenario-enabled cube. See Accessing the Cloud Service from Smart View.
3. Do an ad hoc analysis.
4. Drill into the Sandbox dimension to view the sandbox members.

**Examples**

This is a Smart View grid including the Base member and a sandbox member. Sandbox values have not been updated, so they reflect the Base values. Those values are stored only in the Base, not in the sandbox members:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cola</td>
<td>New York</td>
<td>Actual</td>
<td>Jan</td>
<td>Sales</td>
<td>678</td>
</tr>
<tr>
<td>Cola</td>
<td>New York</td>
<td>Actual</td>
<td>Jan</td>
<td>COGS</td>
<td>271</td>
</tr>
</tbody>
</table>

The changed sandbox value, 500, is stored in a sandbox member. The remaining sandbox value, 271 that was not updated is stored only in the Base:

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cola</td>
<td>New York</td>
<td>Actual</td>
<td>Jan</td>
<td>Sales</td>
<td>678</td>
<td>500</td>
</tr>
<tr>
<td>Cola</td>
<td>New York</td>
<td>Actual</td>
<td>Jan</td>
<td>COGS</td>
<td>271</td>
<td>271</td>
</tr>
</tbody>
</table>

This is a grid with multiple sandbox members. If you have the Database Update user role, you can create or work within multiple scenarios simultaneously:

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>Jan</td>
<td>Sales</td>
<td>New York</td>
<td>Cola</td>
<td>Base</td>
<td>sb0</td>
</tr>
<tr>
<td>Actual</td>
<td>Jan</td>
<td>COGS</td>
<td>New York</td>
<td>Cola</td>
<td>678</td>
<td>500</td>
</tr>
<tr>
<td>Actual</td>
<td>Jan</td>
<td>COGS</td>
<td>New York</td>
<td>Cola</td>
<td>271</td>
<td>271</td>
</tr>
</tbody>
</table>
Comparing Scenario Values to Base Values

If you are the owner, approver or participant for a given scenario, you can view scenario and base values in a spreadsheet to compare models.

To view both scenario and base values in an Excel spreadsheet:

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click Scenario.
2. On the Workflow tab, start Excel by clicking the Excel icon under Show Changes.
3. You can view values for both the scenario and base members in the spreadsheet.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Base</td>
<td>sb10</td>
</tr>
<tr>
<td>2</td>
<td>Cola</td>
<td>New Yo</td>
<td>Actual</td>
<td>Jan</td>
<td>Sales</td>
<td>678</td>
<td>700</td>
</tr>
<tr>
<td>3</td>
<td>Cola</td>
<td>Massac</td>
<td>Actual</td>
<td>Jan</td>
<td>Sales</td>
<td>494</td>
<td>500</td>
</tr>
<tr>
<td>4</td>
<td>Cola</td>
<td>Florida</td>
<td>Actual</td>
<td>Jan</td>
<td>Sales</td>
<td>210</td>
<td>230</td>
</tr>
<tr>
<td>5</td>
<td>Cola</td>
<td>Connect</td>
<td>Actual</td>
<td>Jan</td>
<td>Sales</td>
<td>310</td>
<td>350</td>
</tr>
<tr>
<td>6</td>
<td>Cola</td>
<td>New Hi</td>
<td>Actual</td>
<td>Jan</td>
<td>Sales</td>
<td>120</td>
<td>150</td>
</tr>
<tr>
<td>7</td>
<td>Cola</td>
<td>East</td>
<td>Actual</td>
<td>Jan</td>
<td>Sales</td>
<td>1812</td>
<td>1950</td>
</tr>
</tbody>
</table>

- On row 1, sb10 is the scenario (or sandbox) member.
- Also on row 1, Base shows the base values.
- In the scenario, values for sb10 on rows 2 through 6 have been changed, and you can see the aggregated result in row 7.

4. Compare the scenario to the base in order to determine your next steps. For example, you might choose to change the status of the scenario to approved based on this information.

Reverting Scenario Values Back to Base Values

Initially, scenario values are not stored and they are an exact mirror of the base values. After you change the scenario values in Excel and submit the changes to the cube, the scenario values are stored, and they are different from the base. You can revert the scenario values back to the base by typing #Revert in the changed cells and clicking Submit Data on the Smart View ribbon.

To revert scenario values back to the base:

1. In Excel, type #Revert in the scenario cells you want to revert to the base.
2. Click Submit Data on the Smart View ribbon.

The selected scenario values are updated to the base values.

Example

1. Initially, the value in sb1 is an exact mirror of the value in base.
2. Submit a new value, 100, to sb1.

3. Submit #Revert to sb1.

4. Refresh the sheet. See that sb1 again reflects the base value of 678.

Setting Scenario Cells to #Missing

You can set scenario cells to #Missing even if the corresponding base cells have values.

To set a scenario cell to #Missing:

1. Type #Missing in the cell or delete the cell contents.
2. Select Submit Data on the Smart View ribbon.

Example

1. Initially, the value in sb1 is an exact mirror of the value in the base.

2. Enter #Missing in sb1 (or delete the cell contents) and submit data.

3. Refresh the sheet. See that sb1 is #Missing.
Deleting the Scenario

To reuse a sandbox associated with a scenario, you need to delete the scenario after applying or discarding data changes. Deleting the scenario discards changes made to the sandbox and makes it available for a new scenario.

1. On the Applications home page, expand the application to show the cube row for the outline that you want to modify, then click Scenario.

2. On the Workflow tab, select the scenario that you want to delete.

3. Click Actions, and then select Delete.
You can create or modify application workbooks and then deploy cubes to Oracle Analytics Cloud - Essbase using Cube Designer, a Smart View extension.

- About Cube Designer
- About the Cube Designer Ribbon
- About the Cube Designer Wizard
- Downloading Sample Application Workbooks from the Cloud
- Building a Private Inventory of Application Workbooks
- Working with Application Workbooks in Cube Designer
- Creating a Cube from a Private Application Workbook in Cube Designer
- Creating a Cube from Tabular Data in Cube Designer
- Updating Cubes Incrementally in Cube Designer
- Creating and Validating Member Formulas in Cube Designer
- Loading Data in Cube Designer
- Calculating Data in Cube Designer
- Viewing, Monitoring, and Troubleshooting Jobs in Cube Designer
- Viewing Dimension Hierarchies in Cube Designer
- Exporting Cubes to Application Workbooks in Cube Designer
- Deleting Applications and Cubes in Cube Designer

About Cube Designer

The basic components of Cube Designer are the Cube Designer ribbon and the Cube Designer wizard. See About the Cube Designer Ribbon and About the Cube Designer Wizard.

About the Cube Designer Ribbon

Cube Designer helps you to design, create and modify application workbooks to meet their strict layout and syntax requirements. You can also use options on the Cube Designer ribbon to perform a number of cube management tasks, such as loading data, editing formulas and viewing jobs.
Cube Designer Ribbon Options

- **Connections**: Opens the Connections dialog box, in which you choose the cloud service URL.
- **Cloud**: Opens the Cloud Files dialog box, which contains a selection of prebuilt application workbooks, from which you can build sample applications and cubes.
- **Private**: Provides a drop-down menu with options to open or save an application workbook, or to export a cube to an application workbook.
- **Cube Designer**: Opens the Cube Designer wizard, a series of panels in which you can design and edit application workbooks.
- **Formula Editor**: Opens the Formula Editor, which provides an interface in which to develop member formulas, with assistance for developing correct syntax.
- **View Hierarchy**: Opens the Dimension Hierarchy dialog box, in which you can view the hierarchy for the selected dimension worksheet in an application workbook, and perform tasks, such as renaming members and changing storage settings. See *Working with Dimension Worksheets in Cube Designer*.
- **Build Cube**: Opens the Build Cube dialog box, where you can build a cube from the active application workbook. In this dialog box, cube designer automatically detects existing data and calculation worksheets, and then pre-selects options to load the data and run the worksheets.
- **Load Data**: Opens the Load Data dialog box, which contains options to clear all data and to load data.
- **Calculate**: Opens the Calculate Data dialog box, in which you can select an application, a cube, and a calculation script to execute.
- **Analyze**: Provides a drop-down menu with options to create a Smart View ad hoc grid, or connect application workbook query worksheets (Query.query_name worksheets) to Smart View.
- **View Jobs**: Opens the Job Viewer dialog box, in which you can monitor the status of jobs, such as data loads, calculations, imports, and exports.
- **Transform Data**: Opens the Transform Data dialog box, which lets you build a cube from tabular data.
- **Admin Tasks**: Opens a menu from which you can delete an application or delete a cube. Selecting one of these options opens the Delete Application or Delete Cube dialog box.
- **Server name**: Shows the currently defined connection location. When you click **Server name** and log in (if prompted to do so), the Cube Designer dialog box indicates version information for both Cube Designer and Oracle Analytics Cloud - Essbase.

### About the Cube Designer Wizard

The Cube Designer wizard uses a manual system of reading and writing to the worksheets in an application workbook. The **From Sheet** button at the bottom of the Cube Designer wizard reads the entire application workbook's data and populates the wizard with the data. The **To Sheet** button updates the entire application workbook with the data from the Cube Designer wizard. The **Reset** button clears the data from the Cube Designer wizard.
One common use of wizard is to populate it with information from one application workbook using From Sheet, open a new blank workbook, and then use To Sheet to make a clone of the first application workbook.

You can design and edit application workbooks in the Cube Designer wizard. Each of its five tabs correspond to one of the five types of worksheets in an application workbook. See Designing and Creating Cubes Using Application Workbooks.

To open the wizard, click Cube Designer on the Cube Designer ribbon.

If the Smart View panel displays when you click Cube Designer, then click Switch To, and select Essbase Cloud Cube Designer from the drop down menu.

The Cube Designer wizard contains the following tabs:

- **Cube**: You can design and modify the Essbase.Cube worksheet in an application workbook. See Working with the Essbase.Cube Worksheet in Cube Designer.
- **Settings**: You can design and modify the Cube.Settings worksheet in an application workbook. See:
  - Working with the Cube.Settings Worksheet: Alias Tables in Cube Designer.
  - Working with the Cube.Settings Worksheet: Attribute Settings in Cube Designer.
  - Working with Text Lists Worksheets in Cube Designer.
- **Dimensions**: You can design and modify the Dim.dname worksheets in an application workbook. See Working with Dimension Worksheets in Cube Designer.
- **Data**: You can design and modify the Data.filename worksheet in an application workbook. See Working with Data Worksheets in Cube Designer.
- **Calc**: You can design and modify the Calc.scriptname worksheet in an application workbook. See Working with Calculation Worksheets in Cube Designer.

## Downloading Sample Application Workbooks from the Cloud

Using the sample application workbooks provided in the Cloud Files dialog box, you can quickly create sample applications and cubes. The cubes are highly portable, because they are quickly and easily imported and exported.

1. On the Cube Designer ribbon, click Cloud.
2. If prompted to connect, enter your user name and password.

3. On the Cloud Files dialog box, choose the sample application workbook you want to open.

You can then edit the application workbook to fit your requirements in Cube Designer wizard. See Working with Application Workbooks in Cube Designer.

You can save this modified application workbook to your private inventory. See Building a Private Inventory of Application Workbooks.

Building a Private Inventory of Application Workbooks

Cube Designer allows you to create and store application workbooks on the client computer. This lets you keep a private inventory of completed and in-progress application workbooks.

Using the Private icon menu items on the Cube Designer ribbon, you can manage your private application workbook inventory:

- Open existing application workbooks from your inventory.
- Save new and updated application workbooks to your inventory.
- Export cubes to application workbooks and add them to your inventory.

Opening an Application Workbook

1. On the Cube Designer ribbon, click Private.
2. Select Open Application Workbook.
3. Browse to the application workbook and click Open.

Saving an Application Workbook

1. Open the application workbook.
2. On the Cube Designer ribbon, click Private.
3. Select Save Application Workbook.
4. Browse to your inventory location and click Save.

Exporting an Application Workbook

1. On the Cube Designer ribbon, click Private.
2. Select Export Cube to Application Workbook.
3. If prompted to log in to the cloud service, enter your user name and password.
4. In the Export Cube dialog box, select the application and cube you want to export, indicate if you want to include input level data and calculation scripts, and click Run.
5. To add the application workbook to your private inventory, click Save Application Workbook.
Using Cube Designer wizard, you can modify an application workbook, and then you can use the modified workbook to create an updated cube, reflecting your changes.

- Working with the Essbase.Cube Worksheet in Cube Designer
- Working with the Cube.Settings Worksheet: Alias Tables in Cube Designer
- Working with the Cube.Settings Worksheet: Properties in Cube Designer
- Working with the Cube.Settings Worksheet: Dynamic Time Series in Cube Designer
- Working with the Cube.Settings Worksheet: Attribute Settings in Cube Designer
- Working with the Cube.Settings Worksheet: Substitution Variables in Cube Designer
- Working with Dimension Worksheets in Cube Designer
- Working with Data Worksheets in Cube Designer
- Working with Calculation Worksheets in Cube Designer

Limitations of Application Workbooks

Current limitations for using application workbooks, are listed here.

The following limitations currently exist when working on application workbooks in Excel.

- You cannot set up a dimension worksheet using the generation format. Instead, you must import using the parent-child build method.
- Multiple dimension sheets for the same dimension are not supported. You are limited to one worksheet per dimension.
- Application workbooks do not support aggregate storage cubes.
- Changes to the Cube.Settings worksheet cannot be applied incrementally. Instead, you must rebuild the cube to apply those changes.

Working with the Essbase.Cube Worksheet in Cube Designer

Using the Cube tab in the Cube Designer wizard, you can modify the following fields on the Essbase.Cube worksheet:

- Application Name
- Cube Name
- Dimension Definitions

You can change the application name and cube name, and delete one or more dimensions.

1. On the Cube Designer ribbon, select **Cube Designer**.
2. In the Cube Designer wizard, select the **Cube** tab.
3. Select From Sheet to populate the wizard with the contents of the application workbook.

4. Change the application name or the cube name, if you want to.
   For valid values, see Understanding the Essbase.Cube Worksheet.

5. Add one or more dimensions by typing the name in the text box and pressing the enter key after each one.

6. In the Dimensions list
   • If you want to delete a dimension, select a dimension name and press the delete key.
   • If you want to rename a dimension, right click the dimension name and select Rename Dimension.

7. Select To Sheet to propagate the changes to the application workbook.

8. Examine the updated application workbook to see your changes.

Working with the Cube.Settings Worksheet: Alias Tables in Cube Designer

You can add new alias tables in the Cube.Settings worksheet.

1. In the Cube Designer wizard, select the Settings tab.
2. Select **From Sheet** to populate the wizard with the contents of the application workbook.

3. In the **Alias Tables** field, enter a name for the new alias table.

4. Press **Enter**.

5. Select **To Sheet**.

A new alias table name is added on the Cube.Settings worksheet in the application workbook. To add the alias table to a dimension worksheet, open the Dimensions tab in the Cube Designer wizard, and add the alias table to the selected dimension worksheet. See *Working with Dimension Worksheets in Cube Designer*. After you add the alias table to the dimension worksheet, you must populate the aliases manually, or by copying from a source.

**Working with the Cube.Settings Worksheet: Properties in Cube Designer**

You can add new properties in the Cube.Settings worksheet.

1. In the Cube Designer wizard, select the **Settings** tab.

2. Select **From Sheet** to populate the wizard with the contents of the application workbook.

3. Expand the **Properties** section.

4. Make your selections.

   For valid values, see *Understanding the Cube.Settings Worksheet: Properties*.

5. Select **To Sheet** to propagate the changes to the application workbook.

**Working with the Cube.Settings Worksheet: Dynamic Time Series in Cube Designer**

You can add dynamic time series members in the Cube.Settings worksheet.

1. In the Cube Designer wizard, select the **Settings** tab.

2. Select **From Sheet** to populate the wizard with the contents of the application workbook.

3. Expand the **Dynamic Time Series** section.
4. Make the changes that you want.
   For valid values, see Understanding Dimension Worksheets.

5. Select **To Sheet** to propagate the changes to the application workbook.

### Working with the Cube.Settings Worksheet: Attribute Settings in Cube Designer

You change attribute settings on the Cube.Settings worksheet.

1. In the Cube Designer wizard, select the **Settings** tab.

2. Select **From Sheet** to populate the wizard with the contents of the application workbook.

3. Expand the **Attribute Settings** section.

4. Make the changes that you want.
   For valid values, see Understanding the Cube.Settings Worksheet: Attribute Settings.

5. Select **To Sheet** to propagate the changes to the application workbook.
Working with the Cube.Settings Worksheet: Substitution Variables in Cube Designer

You can add cube-level substitution variables on the Cube.Settings worksheet. Enter the name of the substitution variable in column A. Enter the corresponding value of the substitution variable in column B.

Note:
You must enclose member names in double quotes.

Working with Dimension Worksheets in Cube Designer

1. In the Cube Designer wizard, select the Dimensions tab.

2. Select From Sheet to populate the wizard with the contents of the application workbook.

3. Make the changes that you want.
For descriptions of the options and valid values, see Understanding Dimension Worksheets.

4. (Optional) If you want to update the Cube.Generations worksheet in the application workbook, click the Update Generation Worksheet button. See Understanding the Cube.Generations Worksheet.

The Update Generation Worksheet button creates a section in the Cube.Generations worksheet for the dimension selected in the Dimension drop down list on the Dimensions tab of the wizard.

**Note:**

The Dimension section of the Cube.Generations worksheet changes if you add or delete members on the dimension worksheet (Dim.dimname), causing the number of generations in the dimension to change. If you make changes to the dimension worksheet by adding or deleting members, you should always press the Update Generation Worksheet button as part of the editing process.

5. Select To Sheet to propagate the changes to the application workbook.

**Note:**

- After adding alias tables using Cube Designer wizard, populate the alias table column with alias names manually, or by copying them from a source.
- Use no more than 1024 characters when naming dimensions, members, or aliases.
- The length limit for the dimension worksheet is 30 characters, including 3 characters for the "Dim." at the beginning of the sheet name. So, the name following "Dim." can contain up to 27 characters.

## Working with Data Worksheets in Cube Designer

You can create new data worksheets in the Cube Designer wizard.

1. In the Cube Designer wizard, select the Data tab.

2. Select From Sheet to populate the wizard with the contents of the application workbook.

3. Enter a name for the new data worksheet in the Data Sheets field.

4. Press Enter.

5. Select To Sheet

A new data worksheet is created in the application workbook.
Working with Calculation Worksheets in Cube Designer

You can create new calculation worksheets in the Cube Designer wizard.

**Note:**

Cube Designer calculation worksheets apply only to block storage cubes.

1. In the Cube Designer wizard, select the **Calc** tab.
2. Select **From Sheet** to populate the wizard with the contents of the application workbook.
3. In the **Calculation Sheets** field, enter a name for the new calculation worksheet.
4. Press **Enter**.
5. Select **To Sheet**.

A new calculation worksheet is created in the application workbook.

Creating a Cube from a Private Application Workbook in Cube Designer

Using a sample private application workbook, you can create a cube from Cube Designer.

1. In Excel, on the Cube Designer ribbon, select **Private** and then select **Open Application Workbook**.
2. Select an application workbook, then select **Open**.
3. On the Cube Designer ribbon, select **Build Cube**.
4. On the Build Cube dialog box, verify that you want to use the selected options. Cube Designer detects data worksheets and calculation worksheets in the application workbook, and pre-selects those options for you, however you can deselect those options if you want to:
   - **Load Data Sheets Contained within Workbook** is pre-selected if data worksheets exist in the workbook. You can de-select this option if you do not want to load data.
   - **Run Calculation Sheets Contained within Workbook** is pre-selected if calculation worksheets exist in the workbook. You can de-select this option if you do not want to run the calculations.
5. Click **Run**.
6. Click **Yes** to launch Job Viewer and view the status of the Excel import, or click **No** if you don't want to launch Job Viewer.

See Viewing, Monitoring, and Troubleshooting Jobs in Cube Designer.
Working with Text Lists Worksheets in Cube Designer

You can add text list definitions to application workbooks to work with text measures.

1. Open an application workbook.
2. On the Cube Designer ribbon, click **Cube Designer** to open the Cube Designer wizard.
3. Click the **Settings** tab.
4. Click **From Sheet** to populate the wizard with the contents of the application workbook.
5. In the **Text Lists** field, type the name for the new text list.
6. Press Enter.
   
   The text list name is moved to the text box below the **Text Lists** field.
7. Click **To Sheet**.

   A new text list definition section is added on the Cube.Textlists worksheet in the application workbook. If no Cube.Textlists sheet exists, one is created and the text list definition is added. Multiple text list definitions are supported, and will be added to the same worksheet.

After you add the text list, you must enter the text list information manually. This includes the associated members for the text list, the valid text items in the list and their related numeric values.

See **Understanding the Cube.Textlists Worksheet** and, in *Designing and Maintaining Essbase Cubes*, see:

- Working with Typed Measures
- Performing Database Operations on Text and Date Measures

Creating a Cube from Tabular Data in Cube Designer

This workflow uses two sample tabular data Excel files to demonstrate the concepts of intrinsic and forced-designation headers. See *About Using Tabular Data to Create Cubes*.

1. In Excel, on the Cube Designer ribbon, click **Cloud**.
2. On the Cloud Files dialog box, under **Catalog**, select **Applications**, then select a sample tabular data file:
   - **Sample_Table.xlsx**: Intrinsic headers
   - **Modified_Sample_Table.xlsx**: Forced-designation headers
3. Click **Open**.
4. On the Cube Designer ribbon, select **Transform Data**.
5. On the Transform Data dialog box, enter an application and cube name, if you want to change the default names that are prepopulated.
The application name is based on the source file name without the extension and the cube name is based on the worksheet name.

- Sample_Table.xlsx: Application name is Sample_Table and the cube name is Sales.
- Modified_Sample_Table.xlsx: Application name is Modified_Sample_Table and the cube name is Sales.

6. If you selected Sample_Table.xlsx, do not select Preview Data. Skip to step 8 to create the cube.

7. If you selected Modified_Sample_Table.xlsx, press Preview Data. The workbook is sent to Oracle Analytics Cloud – Essbase for analysis and the relationships are returned for viewing.
   
a. Using the Tree View, you can drag and drop columns to create dimension hierarchies, measure hierarchies, and skipped columns. You can also right click on a column name and designate the property of the column: Generation, Attribute, Alias or UDA. You can also select measures, and hierarchical or flat dimensions, in the Options menu, to set those options for the headers. If you select hierarchical, you get a hierarchy with the dimensions receiving the Excel header names. If you select the flat option, you get a flat display of generations that receive the Excel header names. This setting applies to the entire outline.

b. To save changes to the Excel file, click To Sheet.
c. You can also make changes directly in the open Excel file and have those changes reflected in the grid view by clicking From Sheet.
d. If you do not want to save your changes, select Options and then select Reset to Original Header.

8. When you are ready to create the cube, click Run.
9. When prompted, save the application workbook to your private inventory directory. See Building a Private Inventory of Application Workbooks.

10. When asked if you want to create the cube, click Yes.
11. (Optional) When asked if you want to see the cube job status, click Yes.

12. On the Cube Designer ribbon, select Private, then select Export Cube to Application Workbook.
13. On the Export Cube to Application Workbook dialog box, select the application and cube, and then select Run.

To create a cube in the cloud service, see Creating a Cube from Tabular Data.

### Updating Cubes Incrementally in Cube Designer

Updating a cube is how you load dimensions and members to a cube outline using a data source and a rules file. You can also use the cloud service to add dimensions and members manually.

In an existing cube, you can update a dimension, or add a new one.
Note:
You cannot use Cube Designer to delete dimension in an existing cube.

1. In Excel, on the Cube Designer ribbon, select **Build Cube**.
2. Choose an **Update Cube** option from the **Build Option** menu.

When an outline was changed by a dimension build, the database may be restructured. Each of these options specifies how data values are handled during restructures:

a. **Update Cube - Retain All Data**
   All data values are preserved.

b. **Update Cube - Retain Input Data**
   All blocks (both upper- and lower-level) that contain loaded data are preserved.

   Note:
   This option applies only to block storage cubes.

c. **Update Cube - Retain Leaf Data**
   Only leaf (level 0) values are preserved. If all data is required for calculation resides in leaf members, then you should select this option. If selected, then all upper-level blocks are deleted before the cube is restructured. Therefore, the disk space required for restructuring is reduced, and calculation time is improved. When the cube is recalculated, the upper-level blocks are re-created.

d. **Update Cube - Remove All Data**
   All data values are cleared.

   Note:
   This option applies only to block storage cubes.

Note:
Dimension build definitions are contained within the application workbook and automatically generate the necessary rules files. You do not select a rules file when building dimensions in Cube Designer.
Creating and Validating Member Formulas in Cube Designer

In the Cube Designer Formula Editor, you can write formulas for specific outline members in block storage cubes. You can construct member formulas from operators, functions, dimension names, member names, substitution variables, and numeric constants, and you can validate them to check for correct syntax.

Note:

- The Cube Designer Formula Editor applies only to block storage cubes.
- Validation works against existing cubes in the cloud service. It does not detect application workbook changes that have not been applied to the cube.

Formula Editor provides a formula editing pane in which you can enter a formula. You can use the Tab and arrow keys to move focus within Formula Editor. You can also use a point-and-click approach to select and insert formula components into the formula editing pane.

1. Open the application workbook for the cube that you want to modify.
2. If a dimension worksheet has been defined with the Formula property, select the cell in the Formula column for the member you wish to create a formula.
3. On the Cube Designer ribbon, click Formula Editor.
4. Enter your login credentials for the service, if prompted to do so.
5. In the Formula Editor, create the formula.
   - Use the keyboard to enter formula text. Enclose in quotation marks any member names containing blanks or special characters.
   - Select a cell containing a member name or alias from any dimension worksheet. Place the cursor in the appropriate location of the editor and right-click to paste that name surrounded by quotes into the editor.
   - Double-click on a function to have that function syntax pasted into the editor.
6. Click Validate to check formula syntax.
   - If the validation fails, edit the formula and try again. Be sure to check the error message for guidance.

See these topics in Designing and Maintaining Essbase Cubes:

- Developing Formulas for Block Storage Databases
- Understanding Formula Syntax
- Reviewing Examples of Formulas
Loading Data in Cube Designer

At times, you may need to clear and reload data during cube development. The data and rules files used in the data load process must be stored in the cloud service. If a data worksheet is included in the application workbook, then the data files and rules files get automatically generated during the cube build process. Individual files can also be uploaded. See Uploading Files to a Cube.

1. In Excel, on the Cube Designer ribbon, select **Load Data**.
2. On the Load Data dialog box, select the application and cube in which you want to load data.
3. Under **Select a Job Type**, select an option:
   - **Load Data**: to load data to the cube.
   - **Clear all Data**: to clear all data from the cube.
4. Select the data file and load rule file that you want to use.
5. Select whether to **Abort on Error**. If you select **Abort on Error**, the data load is stopped an error is encountered.
6. Click **Run** to start the data load.
7. Click **Yes** to launch Job Viewer and view the status of the data load, or click **No** if you do not want to start Job Viewer.
8. (Optional) View the status in Job Viewer. See Viewing, Monitoring, and Troubleshooting Jobs in Cube Designer.

See Understanding Data Loading and Dimension Building in *Designing and Maintaining Essbase Cubes*.

Calculating Data in Cube Designer

Calculation scripts specify how cubes are calculated and, therefore, override outline-defined cube consolidations. For example, you can calculate cube subsets or copy data values between members. See Developing Calculation Scripts for Block Storage Databases in *Designing and Maintaining Essbase Cubes*.

During cube development, it is common to recalculate a cube many times when validating the data and formulas. The calculation script files used in the calculation process must be stored in the cloud service. If a Calc worksheet is included in the application workbook, then the calculation script files are automatically generated during the cube build process. Individual calculation script files can also be uploaded in the cloud service. See Uploading Files to a Cube.

1. In Excel, on the Cube Designer ribbon, select **Calculate**.
2. On the Calculate Data dialog box, select an application and a cube, and select the calculation script you want to use.
3. Click **Run** to start the calculation.
4. Click **Yes** to start Job Viewer and view the status of the calculation, or click **No** if you do not want to start Job Viewer.

5. (Optional) View the status in Job Viewer.
   
   See **Viewing, Monitoring, and Troubleshooting Jobs in Cube Designer**.

**Viewing, Monitoring, and Troubleshooting Jobs in Cube Designer**

Use the Cube Designer Job Viewer to view, monitor and troubleshoot jobs that you run from your particular client. Jobs are operations such as data loads, dimension builds, and calculations.

---

**Note:**

A record of all Oracle Analytics Cloud – Essbase jobs is maintained on the cloud service instance. Each job has a unique ID number.

---

**Viewing Jobs in Cube Designer Job Viewer**

In Excel, on the Cube Designer ribbon, click **View Jobs**.

The Job Viewer dialog box opens, showing a list of jobs that have been run from that particular client.

**Removing Jobs from the Cube Designer Job Viewer**

- Press **Clear All** to remove all jobs from the Job Viewer dialog box.
- To selectively remove individual jobs, select one or more jobs and press the **Delete** key.
  - Use the **Shift** key to select multiple contiguous jobs.
  - Use the **Ctrl** key to select multiple non-contiguous jobs.

---

**Note:**

Removing jobs from the Job Viewer dialog box only affects the client. You can still view all jobs, including those deleted from the Job Viewer dialog box, in the Web UI.

---

**Monitoring Jobs in Cube Designer Job Viewer**

- While a job is running, the **Job Viewer** icon on the Cube Designer ribbon displays an hourglass.
- When the job finishes running a Job Viewer status dialog box displays, indicating the status of the job.
Troubleshooting Jobs in Cube Designer Job Viewer

1. In Job Viewer dialog box, select a job and click **Details** to see the job details.
2. In the Job Details dialog box, select a file from the **Server Error Files** drop-down menu and click **Open** to view and troubleshoot errors.

Viewing Dimension Hierarchies in Cube Designer

You can view dimension hierarchies in the Cube Designer Dimension Hierarchy viewer. To learn more about hierarchies, see Outline Hierarchies in *Designing and Maintaining Essbase Cubes*.

1. Open the application workbook containing the hierarchy that you want to view.
2. Select the dimension worksheet for the hierarchy that you want to view.
3. On the Cube Designer ribbon, select **View Hierarchy**.

When you view a hierarchy in Cube Designer, you can perform some actions on the hierarchy. These include:

- To search for a member in the hierarchy, enter a member name in the **Find Next** text box, and click **Find Next**.
- To find a member of the dimension in the application workbook dimension worksheet, select a member in the hierarchy, and click **Go To**. The corresponding member in the application workbook is highlighted.
- To rename a member:
  1. Select the member in the hierarchy and click **Rename**.
  2. Enter the new member name.
  3. Press **Enter**.

The corresponding member is renamed wherever found within the Parent and Child columns of the dimension worksheet.

- To set storage for all parents (except members containing formulas or defined as label only) to dynamic calc or to stored:
  1. Select the member in the hierarchy and click **Edit Parents**.
  2. On the drop-down menu, select **Set Storage To Dynamic Calc** or **Set Storage To Stored**.

See Member Storage Properties in *Designing and Maintaining Essbase Cubes*.

- To expand or collapse a hierarchy:

---

**Note:**

If you close Excel while the job is running, the job continues to run, but you will not see a status dialog when it finishes. The job is a server process, so it runs regardless of whether Excel is open or not.
1. Right-click a member in the hierarchy.
2. Select **Expand All** or **Collapse All**.
   - To show or hide aliases, storage, or operators:
     1. Right-click a member in the hierarchy.
     2. Highlight **Show** to expand the menu.
     3. Click **Alias**, **Storage**, or **Operator**, to show or hide those items.

---

**Exporting Cubes to Application Workbooks in Cube Designer**

In Cube Designer, you can export any cube that exists in the cloud service.

1. In Excel, on the Cube Designer ribbon, select **Private**, then select **Export Cube to Application Workbook**.
2. On the Export Cube dialog box, select the application and cube that you want to export.
   - Select **Include Data** if you want input level data included in the application workbook.
   - In block storage cubes, if the size of the data is 400 MB or less, data is exported to the application workbook, on the Data worksheet. If the data size exceeds 400 MB, data is exported to a flat file named Cubename.txt, which is included in a file named Cubename.zip. The .zip file is created in the specified export directory if the export process is successful.
   - In aggregate storage cubes, regardless of the size of the data, it is always exported to a flat file named Cubename.txt, which is included in a file named Cubename.zip. The .zip file is created in the specified export directory if the export process is successful.
   - Select **Include Calculation Scripts** if you want calculation scripts in your block storage cube included in the application workbook.

---

**Note:**

Aggregate storage cubes do not have calculation scripts.
3. Click Run.
4. When the export is completed, click OK.
   The application workbook is saved to the private folder location: C:\Users\username\AppData\Roaming\Oracle\smartview\DBX. Because it is saved to the private folder location, you can open it using the Private icon on the Cube Designer ribbon.

   The exported application workbook can be imported to the cloud service. See these topics:

   • Creating a Cube from an Application Workbook
   • Creating a Cube from a Private Application Workbook in Cube Designer

Deleting Applications and Cubes in Cube Designer

In Cube Designer, you can delete any application or cube that exists in the cloud service. Deleting an application or cube cannot be undone.

1. In Excel, on the Cube Designer ribbon, select Admin Tasks.
2. From the menu, select Delete Application or Delete Cube.
3. From the Delete Application or Delete Cube dialog box, select the application or cube you want to delete.
Exporting Cubes to Application Workbooks

After you create a cube and make modifications to the cube outline, such as adding dimensions and members, and setting dimension and member properties, you can export the cube to an application workbook that reflects the current state of the cube.

- Exporting a Cube to an Application Workbook
- Exporting On-Premises Cubes to Import to the Cloud

Exporting a Cube to an Application Workbook

1. In the cloud service, on the Applications home page, expand the application that contains the cube that you want to export and select the cube row.

2. Click Actions, and then select Export.

3. On the Export Cube dialog box:
   - Select Data if you want to export the data from the cube. How the data is exported depends on whether the cube is block storage or aggregate storage.
     - In block storage cubes, if the size of the data is 400 MB or less, it is exported to the application workbook, on the Data worksheet. If the data size exceeds 400 MB, data is exported to a flat file named Cubename.txt, which is included in a file named Cubename.zip on the Files page.
     - In aggregate storage cubes, regardless of the size, data is always exported to a flat file named Cubename.txt, which is included in a file named Cubename.zip on the Files page.
   - Select Calculation Scripts if you want to export each of the calculation scripts as a separate worksheet within the application workbook.
   - Select Export to server if you want to export the cube files to the app/appname/dbname folder on your cloud service instance.
   - Select Show Advanced if you want to change the build method, Generation or Parent-Child, for dimensions. You can change the build method for all dimensions using the Set Build Method for all dimensions option, or you can selectively change the build method for individual dimensions.

4. When prompted, save the exported application workbook to your local or network drive or download the exported application workbook and data .zip files from the Files page.

File names do not include spaces because files that are imported to the cloud service cannot contain spaces in the file name.

The exported application workbook can be imported to the cloud service. See:

- Creating a Cube from an Application Workbook
Exporting On-Premises Cubes to Import to the Cloud

If you have applications and cubes that were created in a supported on-premises instance of Essbase, then you can use the cube export utility, which is a command-line tool, to export the metadata and data of a cube into an application workbook. Then you can import the application workbook to create a cube in the cloud service.

You can export applications and cubes created in an on-premises instance of Essbase 11.1.2.3.0nn, 11.1.2.4.0nn, 11.1.2.4.5.nn, 12.2.1 and later.

See:
• Downloading the Cube Export Utility
• Exporting On-Premises Cubes to Application Workbooks
• Reviewing Member Names before Importing an Application Workbook Created by the Cube Export Utility

Downloading the Cube Export Utility

The cube export utility is supported on Windows and UNIX/Linux.

To download the cube export utility from the cloud service:

1. On the Applications home page, click Utilities ➔.
2. On the Utilities page, click Download ➔ next to Export Utility.
3. Save the cube export utility, which is named dbxtool.zip, to a local drive.

Exporting On-Premises Cubes to Application Workbooks

1. On your local drive, uncompress the export utility file (dbxtool.zip) that you previously downloaded.
   See Downloading the Cube Export Utility.

2. Ensure that Java Runtime Environment (JRE) 1.7.n or later is installed, and test that it works by entering java -version at a command prompt.

3. In a command prompt, change directories to the dbxtool/bin directory.

4. Run the cube export utility by entering the export command with the options that you want. Options are describes in the following table.
   • Windows: export.bat
   • UNIX/Linux: export.sh

   You must set export.sh to have execute permission. For example, chmod 744 export.sh.
### Table 10-1  Cube Export Utility Syntax Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-server</td>
<td>Cloud service instance URL.</td>
</tr>
<tr>
<td>-application</td>
<td>Application name (case-sensitive).</td>
</tr>
<tr>
<td>-cube</td>
<td>Cube name (case-sensitive).</td>
</tr>
<tr>
<td>-user</td>
<td>User name.</td>
</tr>
<tr>
<td>-password</td>
<td>Password. If not specified, then you are prompted to enter the password.</td>
</tr>
<tr>
<td>-path</td>
<td>Full path to the export directory. If the file already exists on the specified path, then you are prompted whether to overwrite the file. If you enter yes, then the export proceeds.</td>
</tr>
<tr>
<td>-build</td>
<td>(Optional) Build method to use. Options:</td>
</tr>
<tr>
<td></td>
<td>• PARENT-CHILD</td>
</tr>
<tr>
<td></td>
<td>• GENERATION</td>
</tr>
<tr>
<td>-data</td>
<td>(Optional) Export input data. Only input data can be exported using the cube export utility. Input data means only those blocks of data where the block contains at least one data value that was loaded (imported), rather than created as the result of a calculation. How the data is exported depends on whether the cube is block storage or aggregate storage.</td>
</tr>
<tr>
<td></td>
<td>• In block storage cubes, if the size of the data is 400 MB or less, it is exported to the application workbook, on the Data worksheet. If the data size exceeds 400 MB, data is exported to a flat file named Cubename.txt, which is included in a file named Cubename.zip. The .zip file is created in the specified export directory if the export process is successful.</td>
</tr>
<tr>
<td></td>
<td>• In aggregate storage cubes, regardless of the size of the data, it is always exported to a flat file named Cubename.txt, which is included in a file named Cubename.zip. The .zip file is created in the specified export directory if the export process is successful.</td>
</tr>
<tr>
<td></td>
<td>If the -data option is omitted or the export process is not successful, then data is not exported and the Cubename.txt and Cubename.zip files are not created.</td>
</tr>
<tr>
<td>-calc</td>
<td>(Optional) Export calculation scripts. Separate worksheets are created for each calculation script in the cube.</td>
</tr>
</tbody>
</table>

Windows example:

```
export.bat -server server99.us.example.com:1423 -application Sample -cube Basic -user esscs_user -password password -path C:\export_dir -build PARENT-CHILD -calc -data
```

During the export process, a structured file representing metadata, named Appname_Cubename.xml, is created in the specified export directory. This .xml file is then used to create an application workbook, named Appname_Cubename_timestamp.xlsx. If the export process is successful, then only the application workbook file remains in the specified export directory; the .xml file is removed.

The application workbook can be used to redeploy the same cube to the cloud service. See Creating a Cube from an Application Workbook and Creating a Cube from a Private Application Workbook in Cube Designer.
Note:

When importing an application workbook that was created using the cube export utility, you should carefully review member names in the application workbook, because some member records might be rejected when you import the application workbook. See Reviewing Member Names before Importing an Application Workbook Created by the Cube Export Utility.

The application workbook that is created does not have a space in the file name. Files that are imported to the cloud service cannot contain spaces in the file name.

Limitations of Cube Export

- For setting named generations to dimensions, you must use the GENERATION build method. The PARENT-CHILD build method is not supported.
- Named levels are not supported.
- Formulas cannot exceed 32767 characters.

Reviewing Member Names before Importing an Application Workbook Created by the Cube Export Utility

When importing an application workbook that was created using the cube export utility, you should carefully review member names in the application workbook. Member names are exported to the application workbook as is. If a member name ends with a backslash (for example, mbrname\ or mbr\name\), then the member name is exported to the application workbook as is (mbrname\ or mbr\name\). During the import process, however, the trailing backslash is interpreted as an escape character and the member is rejected (not added to the cube outline).

When the import process is completed, a dialog box provides status details, such as whether a dimension build was successful or if errors were encountered.

For each dimension in which one or more member names are rejected, an error file is created. The error file is named err_DimName.txt or err_Dim_DimName.txt. For example, if the Year dimension has any rejected member names, the error file name is err_Year.txt or err_Dim_Year.txt.

In the dimension error file, each rejected member name is listed, as shown:

```
\Record #98 - Error in association transaction [RB6300] to [Curr_EUR] (3362)
"OTHER","RB6300","N",","","Ballsport L","","","Curr_EUR"
```

The rejected member record text files are available on the Files page. Review the text files and correct the issues in the application workbook.
Tracking Changes to Data

Use an audit trail to track changes to cube data, including changes to Linked Reporting Objects (LROs), adding notes, attaching files, and referencing URLs. Export your log to an Excel spreadsheet, and perform ad hoc queries.

To view data audit trail records, you must be a Service Administrator or Power User who has one of these roles on the Application: Application Manager, Database manager, or Database Update role. You can only view those records where your user name matches the user name registered in the audit records. To delete data audit trail records, you must be a Power User who has Application Manager role on the application.

• Turning on Data Audit Trail and Viewing the Data Audit Trail
• Linking a Reporting Object to a Cell
• Exporting Logs to a Sheet
• Refresh the Audit Log

Limitations of Audit Trail

• Supported only on block storage cubes
• Not supported on scenario-enabled cubes
• Not supported with text measures
• Not supported in ad-hoc analysis

Turning on Data Audit Trail and Viewing the Data Audit Trail

You turn on data audit trail by adding AUDITTRAIL DATA as an application level configuration setting.

See Setting Application-Level Configuration Properties.

1. To turn on Data Audit Trail, add the following to the application configuration parameters: AUDITTRAIL DATA.

2. Perform ad hoc analysis through Smart View, make data changes through Smart View and click on Submit - this results in an audit record being stored.

When doing ad hoc analysis, there are many ways of getting a particular Point of View (POV) on to the grid. One of them is by using the POV toolbar. It's got great flexibility for zeroing in on certain members in one or more dimensions. See these topics in Oracle Smart View for Office User's Guide

• Selecting Members from the POV Toolbar
• Displaying the POV Toolbar
• Selecting Members Using the Cell-Based POV

3. With Data Audit Trail enabled, you can view the audit trail in the connection Panel in Smart View. Under the connection information, click on the menu of operations
under More and you will see a menu option titled Audit Trail. Click on Audit Trail to view the data audit trail records for a cube.

![Audit Trail](image)

4. The audit trail record shows the date and time of the change in the first column, the new value or the linked reporting object in the second column and the POV in the third column. The time corresponds to your time zone. Click on an item in the audit trail and you will see a description of the change at the bottom of the pane.

5. You can display a sheet with the new POV and refreshed data value by clicking Ad hoc below the Audit Trail pane. When you click on subsequent audit records and click this icon, you see a different sheet with the POV for that audit record and refreshed data for that POV. This way, you can do further analysis on targeted data.

Linking a Reporting Object to a Cell

You can link a reporting object to a cell. When you do, this change displays in the data audit trail. You can add a note to a cell, attach a file, or reference a URL. When you make these changes, the cells are highlighted in your cube. See these topics in the Oracle Smart View for Office User's Guide on how to link reporting objects to cells:

- Linked Reporting Objects
- Attaching a Linked Reporting Object to a Data Cell
- Launching a Linked Reporting Object from a Data Cell

Exporting Logs to a Sheet

You can easily export your logs to a new Excel sheet just by clicking an icon.

Export your log onto a new sheet using Export. Click this icon to export the logs with all the details for each entry onto a new sheet that looks like this:
Once exported, you can resort columns or remove them to show the information you want to analyze.

Refresh the Audit Log

You can refresh the audit log to see your latest changes at any time.

When you make more changes to your data, you can refresh the log view any time.

Click Refresh.
Connecting Cubes

You can use either partitions or location aliases to analyze data across cubes.

You may have more than one cube that you use for business analysis. To access data across multiple cubes, you can connect the cubes by implementing partitions, location aliases, or both. Two connected cubes can be thought of as a source and target pair. Your current, logged-in session in the cloud service is the host of the target cube, while the cube that you are accessing is the data source.

- Understanding Partitions
- Understanding Location Aliases
- Defining a Reusable Connection
- Creating a Location Alias Based on a Defined Connection
- Creating a Replicated Partition

Understanding Partitions

A partition is a region of a cube (database) that is shared with another cube. You can create a transparent or replicated partition between a source and a target cube, to share congruent cube regions between them.

Transparent partitions allow access to data from the data source as though it were stored in the data target. The data source can be in another application or cube, or on another cloud service instance.

A replicated partition is a copy of a data source region that is stored in the data target. Some users access the data in the source, while others access it in the target. Changes made to the data in a replicated partition flow from the source to the target. If users change the data at the target, it is overwritten when an administrator refreshes the replicated partition. Users can replicate the data using the actions menu.

If all cubes involved in a partition are hosted on the same instance of Oracle Analytics Cloud - Essbase, then no login credentials are needed as part of the partition setup. You must be signed in on the data target, and also provisioned on the data source.

To create a partition, see Creating a Replicated Partition.

To learn more about partitioning, see Designing Partitioned Applications in Designing and Maintaining Essbase Cubes.

Understanding Location Aliases

A location alias is a descriptor for a cube that you can use to easily reference it in calculations.

A location alias is a descriptor for a cube. When you create a location alias, you provide the host name, application name, and database name of a cube. After you create a location alias, you can use the alias to refer to that cube. You can use
location aliases to perform calculations and write operations from one cube (a source) to another (a target), using the @XREF and @XWRITE calculation functions.

To create a location alias, see Creating a Location Alias Based on a Defined Connection

**Defining a Reusable Connection**

This exercise shows you how to create a reusable connection that you can use for partitions and location aliases.

1. On the Applications home page, click **Admin**.
2. Click the **Connections** tab.
3. Click **Create > Essbase** to create an Essbase connection.
4. In the **Name** field, enter a host nickname; for example, myhost01.
5. To create a connection on the current Essbase cloud service instance,
   a. Clear the **Use URL** check box.
   b. In the **Host** field, enter the fully qualified host name; for example, myhost01.example.com.
   c. In the **Port** field, enter the port number of the service instance; for example, 1423.
6. To create a connection using a remote Essbase cloud service instance,
   a. Click the **Use URL** check box.
   b. In the **URL** field, enter the full URL to the remote instance; for example, http://myhost01.example.com:9000/essbase/agent.
7. Enter a user name and password.
8. Click **Test** to verify that the connection is valid.
9. If it is valid, click **Save** to save the connection.

Now you have an Essbase connection defined in the service.

**Creating a Location Alias Based on a Defined Connection**

This exercise shows you how to create a location alias, which you can later use to incorporate values of a source cube into your current, target cube. You do not have to provide a user name and password when you create a location alias. You can use a saved connection.

1. On the Applications home page, select a cube and click **Admin**.
2. Click the **Location Aliases** tab.
3. Click **Create**.
   The default connection information is already in place for the current session. If you have created any saved connections, you can select **From connections** to choose one of them when you create a location alias.
4. Using the default (current) connection, provide an alias name, the application name, and the database (cube) name, and click **Create**.
Now you have created a location alias defined using an Essbase connection. To use it for read operations from a source cube, to the target, use the @XREF function in a calculation script. To use it to write from the source to the target cube, use @XWRITE. These functions have expanded syntax so that you can either incorporate values from a remote cube, or from another cube on the same Essbase instance.

Creating a Replicated Partition

This exercise shows you how to create a replicated partition, which duplicates an area of a remote source cube onto the current target host. You can use similar steps to create a transparent partition.

1. On the Applications home page, select a cube and click **Partitions**.
2. Click **Create >Replicated**.
3. In the **Source connection** tab, click **Default** to use the currently connected Essbase server instance.
   For this exercise, use the default, though you can also click **From connections** to create a partition that spans across Essbase server instances.
4. Provide the source application and database name, and an optional description.
5. Click the **Areas** tab.
6. Click the + icon to add source and target area definitions.
   For example, add a source area of @DESC("West"), @DESC("200"), @CHILDREN("Qtr2"), and add the same matching target area.
7. Click **Cell count** to see how many cells are in the defined partition area and to ensure that the counts are matching.
8. Optionally provide Area-specific member mapping, and member mapping.
9. Validate and save the partition definition.

Now you have created a replicated partition on your target instance. You can refresh it periodically from the source, by selecting **Replicate data from source** on the **Partitions** page.
Migrating On-Premises and Cloud Service Applications

If you have existing applications from a supported on-premises installation of Essbase, you can migrate them to Oracle Analytics Cloud - Essbase. You can use the Lifecycle Management (LCM) command-line utility, included with the cloud service, to export an on-premises application to a .zip file, and then import the .zip file to the cloud service. You can also migrate applications that are in the cloud service from one cloud service instance to another.

- Preparing to Migrate On-Premises Applications to the Cloud Service
- Application Migration and Use Cases
- Application Elements - Supported or Unsupported
- Migrating Cloud Service Applications

Preparing to Migrate On-Premises Applications to the Cloud Service

If you have an existing Essbase on-premises application and cube to migrate to the cloud service, keep in mind the following design considerations and prerequisites.

Lifecycle Management Utility

With the Lifecycle Management (LCM) command-line utility, you can export on-premises applications and cubes, and then import them to the cloud service. First read these prerequisites. When you are ready to migrate, see Migrating Cloud Service Applications.

Use of the LCM utility requires that you have installed Java Development Kit 8 or higher, and have set the JAVA_HOME environment variable.

To learn about other options for creating applications (besides LCM), see Application Creation Options Other than LCM.

Unicode Mode

You must convert all applications and associated elements to Unicode mode (UTF-8 encoding) before you export and migrate them to the cloud service.

Enable Unicode on the server and then for the Essbase application, or a copy, prior to running LCM export. Use the following the MAXL commands:

```
alter system enable unicode — this enables the Essbase server to support unicode for applications
alter application <appname> set type unicode_mode — this enables the application and application elements for Unicode. This is irreversible transaction — you may want to make a copy of the application and enable it for unicode.
```
Then use ESSUTF8 utility, available in ARBORPATH directory, to convert any disk files to Unicode – otl, rul, csc, txt, etc.

Substitution Variables

If you have server-level substitution variables, then you must convert them to application-level variables before migrating to the cloud service. Server-level substitution variables are not supported.

Hybrid Aggregation Mode

The default calculation and query processor in the cloud service is hybrid mode. Hybrid mode enables block storage cubes to have dynamic, upper-level sparse members, and fully dynamic query and calculation. You can query data immediately after updating it, without running batch calculations. In hybrid mode, there is no impact to your cubes if you choose not to apply Dynamic Calc to upper level sparse members.

For more information about hybrid mode, see Calculation and Query Processor.

Implied Sharing

Implied sharing is not supported in the cloud service. All stored intersections have data, regardless of their child count.

Configuration Settings

Default configuration values are different in the cloud service.

- IGNORECONSTANTS setting is now TRUE by default. Calculations in hybrid mode do not assign constants.
- INDEXCACHESIZE and DATACACHESIZE settings now control cache sizes for all cubes on the cloud service (except aggregate storage cubes). Formerly, these settings only affected newly created or migrated cubes. You cannot change the cache sizes using MaxL. You can only change the cache sizes using these configuration settings.
- GRIDSUPPRESSINVALID is now TRUE by default. Invalid intersections are not displayed in Smart View grids.
- QRYGOVEXECTIME is now set by default to 300 seconds, meaning that queries time out if not completed in that timeframe.

In addition to the above noted configuration changes, other configuration settings are in place with default, pre-set values, that you can modify (if needed) at the application level. See Setting Application-Level Configuration Properties.

Oracle strongly recommends managing all configuration settings at the application level. Application-level configuration is preserved during the LCM utility export and import processes.

Application Files and Artifacts

You must convert all application-level files and artifacts into database-level files and artifacts before you export them from on-premises Essbase instance, and before you migrate to the cloud service. This includes calculation scripts, rule files, and text files. In the cloud service, artifacts are supported at the database level only.

You can import on-premises rules files to the cloud service and execute them in the cloud service.
If you encounter file upload size restrictions between external clients and the cloud service, then you may need to split large files into smaller files and then concatenate them together after uploading them to the cloud service, using an SSH connection to the server.

**Outlines**

Outlines are encrypted on the cloud service deployment servers. If you need to export and import outlines between cloud service servers, then the LCM utility and application workbooks are the only supported methods for doing so.

**Unsupported Application and Database Settings**

The following application and database level settings are no longer supported:

- Enable/disable Commands (enabled by default)
- Enable/disable Connects (enabled by default)
- Enable/disable Updates (enabled by default)
- Data and index cache size controls (defaults are fixed, but can be changed per application using INDEXCACHESIZE and DATACACHESIZE configuration settings)
- Minimum permission levels (create security filters prior to LCM export instead)
- Set lock timeout
- Currency conversion
- Disk volumes

**Partitions**

Only transparent are supported by the LCM utility. Linked and federated partitions are not supported. When you export the target application using the LCM utility, the partition definition is exported as part of the application.

When you perform the LCM import operation, always import the source applications before the target applications. If the source application has not been imported prior to the target application, then the partition definition will not work, and you will have to use MaxL to re-create the partition definition after importing the source application.

**Size Requirements**

Ensure that pre-existing applications you plan to migrate to the cloud service will fit at the resource level you procure. Estimate sizing requirements, and procure the most relevant combination of CPU, memory, and storage.

**Application Creation Options Other than LCM**

In addition to using LCM to migrate exported applications, you can also create applications in the following ways:

- Use Excel application workbooks.
  See About Application Workbooks.
- In Smart View, use the Cube Designer extension.
  See About the Cube Designer Ribbon.
Application Migration and Use Cases

LCM Utility Use Cases

Oracle Analytics Cloud - Essbase provides a Lifecycle Management (LCM) command-line based utility that can be used to migrate Essbase on-premises applications, folders and elements using a .zip file, for import to the cloud service. The following releases are supported for migration: 11.1.2.3.0nn, 11.1.2.4.0nn, 11.1.2.4.5nn, 12.2.1, and later.

Not all application and database elements can be exported. See Application Elements - Supported or Unsupported.

LCM Utility Download and Typical Usage

1. On the Applications home page, click Utilities, and download the LCM utility (EssbaseLCMUtility.zip).
2. In the uncompressed downloaded file, run EssbaseLCM.bat (Windows) or EssbaseLCM.sh (Linux), based on the platform on which you want to run the utility.

Note:

You can execute the file from any location against a remote cloud service instance.

3. Manually convert the on-premises Essbase application to Unicode before exporting it to a .zip file.
4. Run the LCM export command to download the on-premises Essbase application and its elements to the specified .zip file.
5. To import the application into the cloud service, use the Oracle Analytics Cloud - Essbase CLI tool to upload the .zip file to a target cloud service application.
6. Log into the cloud service to see the application and cube on the Applications home page.

The LCM import and export commands are described in Migrating Cloud Service Applications.

Use Case: Migrate from 11.1.2.3.nn On-Premises to the Cloud Service

1. Download the LCM utility from the cloud service and execute it in the system running Essbase 11.1.2.3.nn. This use case is similarly relevant to Essbase 11.1.2.4.nn.
2. Export the required source application using the utility. This includes application-level elements.
3. In the exported .zip file, if a partition exists in the source application, then edit the partition XML to correct any partition settings.
4. Import the .zip file into the cloud service using the CLI tool.
Note:
You must move server-level substitution variables, which are used in on-premises applications, to the application level prior to import.

5. After importing using CLI, perform the following in the Oracle Analytics Cloud - Essbase user interface:
   • Assign calculation scripts to relevant users
   • Assign cloud-based user roles to users

Application Elements - Supported or Unsupported

The following elements are supported or not supported for migration using the LCM utility.

Application Level
Migrated elements that are supported at the application level:
• Application definition metadata (such as application type)
• Calculation scripts
• Excel worksheets
• Macro-enabled Excel files*
• Open XML Excel files*
• Platform connections and datasources
• Report scripts
• Rules files
• Shadow configuration for JAR (Java ARchive) files*
• Substitution variables
• Text files

* The asterisked application elements are supported only for Jagent-based deployments (11.1.2.4.5nn, 12.2.1 and later).

Database Level
Migrated elements that are supported at the database level:
• Calculation scripts
• Cube metadata (cube properties are required to create the cube)
• Data (data is required to be moved to the correct location at the cube level for the cloud service)
• Database transaction ID*
• Drill-through definitions
• Excel files
• Filter definitions
• Location aliases
• Macro-enabled Excel files*
• Open XML Excel files*
• Outline
• Transparent partitions

**Note:**

When partitions exist in the source between a source application and database, and a target application and database, only partitions from the target are exported to the file system.

When partitions exist between cubes being migrated, you must import the data source before the data target. Otherwise, partition definitions may not be restored.

• Replicated partitions
• Report scripts
• Rules files
• Scenario management
• Substitution variables
• Text files

* The asterisked database elements are supported only for Jagent-based deployments (11.1.2.4.5nn, 12.2.1 and later).

**Server Level**

Migrated elements that are supported at the server level:

• Connections and datasource

**Note:**

You can migrate global or server-level connections and datasources by adding the optional argument: `-include-server-level` (or `-isl`) to LCM import or export operations.

**Security**

Users must be exported and imported separately before security elements are imported. See Migrating Cloud Service Applications.

The following security elements are exported and imported as part of the LCM utility.

• User calculation scripts
• User filter definitions
• User roles (supported only for migration across Oracle Analytics Cloud - Essbase deployments and releases)
Unsupported Elements

The following elements are not supported for migration:

- @XREF function formulas
- Application logs
- Catalog server
- Disk volume definitions
- Linked reporting objects
- Linked and federated partitions
- Server-level custom-defined functions and custom-defined macros
- Server-level essbase.cfg file
- Server-level substitution variables
- Tablespaces
- XML files

Migrating Cloud Service Applications

You can use the LCM (Lifecycle Management) utility, for Windows or Linux, and the Command-Line Interface (CLI) tool to migrate your source application and elements across Oracle Analytics Cloud - Essbase deployments and releases.

The process flow involves exporting the application to a zip file and then importing the .zip file to the target deployment or release.

Note:

- If you are migrating across Oracle Analytics Cloud - Essbase deployments and releases, from v17.3.3 (or earlier), use the scripts for migrating to Essbase, as described in the section, Scripts for Administration Tasks in the Administering Oracle Analytics Cloud document. This also applies to export and import of provisioning of application roles and scripts across these deployments.
- In the cloud service, all applications must be in Unicode. Therefore, Essbase on-premises applications must be manually converted to Unicode before exporting them to a .zip file.
- Only transparent partitions are supported by the utility. Federated or linked partitions are not supported.
- During LCM/CLI import, if scenarios are present in the exported .zip file, but you do not provide scenario database details, then import of scenario management data is bypassed.
- Restoring an application or database from a prior backup, after the application or database was re-created using LCM import, is not supported.
For information on downloading the LCM command-line utility and for use cases, see Application Migration and Use Cases.

The required user roles are as follows:

- For exporting: Application manager for the application created. In addition, the following can use the LCM utility and CLI tool: Service administrator for all applications; Power user for all applications created by the Power user.
- For importing: Power user or Service administrator, for creating new applications during import.

Exporting an On-Premises Application from the Source

1. Run the LCM utility.
2. At the LCM utility command prompt, enter the following command syntax to export the application to a .zip file:

   ```
   ```

   **Note:**

   The prior arguments (-server, -user, -password, -application, -zipfile) can be specified in any order. To prompt for a password, do not include the -password password option. To skip the export of cube data during export, specify -nodata, which is an optional argument). By default, all cube data is exported. All of the options in the command (from -scenario onward, which are the database details) are optional; provide them only if you want to migrate scenarios along with other elements in an application.

Exporting a Cloud Service Application from the Source

1. Run the CLI tool to export the cloud service application to a .zip file.
2. Follow the instructions as described in Oracle Analytics Cloud - Essbase Command-Line Interface (CLI)

Importing an On-Premises or Cloud Service Application to the Target

When partitions exist in the source between a source application or database, and a target application or database, only partitions from the target are exported to the file system.

When partitions exist between cubes being migrated, you must import the data source before the data target. Otherwise, partition definitions may not be restored.

1. Run the CLI tool to import the Essbase on-premises application from the .zip file to the target.
2. Follow the instructions as described in Oracle Analytics Cloud - Essbase Command-Line Interface (CLI).

Migrating Applications and Cubes from Financial Consolidation and Close Cloud Service or Planning and Budgeting Cloud Service
1. Export from Planning and Budgeting Cloud Service (PBCS) or Financial Consolidation and Close Cloud Service (FCCS) using the PBCS interface, or the EPM Automate Utility command line tool using `exportsnapshot`. See EPM Automate Utility Commands in Working with EPM Automate for Oracle Enterprise Performance Management Cloud.

2. Run the CLI tool to import the Essbase application and cubes from the PBCS .zip file using CLI command `lcmimport`, as described in LcmImport: Restoring Cube Files.
14

Configuring Cloud Service Resource Limits and Application Configuration Properties

The cloud service is preconfigured with resource-limits that you may never need to modify. If necessary, you can modify resource limits for the cloud service instance and you can set configuration properties at the application level.

- Modifying Cloud Service Resource Limits
- Setting Application-Level Configuration Properties

Modifying Cloud Service Resource Limits

⚠️ This topic does not apply to Oracle Autonomous services.

If you have a Service Administrator role, you can customize your Oracle Analytics Cloud - Essbase instance by modifying the resource limits that were set when the size and shape of the cloud service compute node was selected during setup.

For information on compute shapes, see Before You Create a Service in Designing and Maintaining Essbase Cubes.

Note:

Before changing resource limits, you should know what the current resource limits are in the cloud service instance. The Service Administrator can access the cloud service virtual machine using a Secure Shell (SSH) client. Log in as user oracle and enter the `ulimit -a` command.

Changing the cloud service resource limits could affect performance.

You can also view the system resources for available host .RAM and disk space, and the minimum required. If the available system resources are less than the minimum required, the system will stop responding.

1. On the Applications home page, without selecting an application or cube, click Settings.

If without selecting an application or cube, Settings is not available, put your cursor in the search field, before or after the asterisk (*), and click Search.

2. On the Settings page, under Resource limits, is a list of the predefined settings and their descriptions, which are all set to the default value.
3. To change a resource limit, enter a value from 0 to unlimited. If the value of a resource limit is not already set to unlimited, you can select **Unlimited**.

4. Click **Submit**.

Default value of resource limits for the OC3, OC4, OC5, and OC6 compute shapes.

### Table 14-1  Resource Limit Default Values for Compute Shapes OC3, OC4, OC5, and OC6

<table>
<thead>
<tr>
<th>Resource</th>
<th>Command</th>
<th>OC3</th>
<th>OC4</th>
<th>OC5</th>
<th>OC6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core File Size [Blocks]</td>
<td>ulimit -c</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
</tr>
<tr>
<td>Data Segment Size [KB]</td>
<td>ulimit -d</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
</tr>
<tr>
<td>Maximum Memory [KB]</td>
<td>ulimit -m</td>
<td>7</td>
<td>14.5</td>
<td>29.5</td>
<td>59.5</td>
</tr>
<tr>
<td>Open Files</td>
<td>ulimit -n</td>
<td>262144</td>
<td>262144</td>
<td>262144</td>
<td>262144</td>
</tr>
<tr>
<td>User Max Processes</td>
<td>ulimit -u</td>
<td>16000</td>
<td>32000</td>
<td>32000</td>
<td>32000</td>
</tr>
<tr>
<td>Virtual Memory Size [KB]</td>
<td>ulimit -v</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
</tr>
<tr>
<td>WebLogic Heap Size [MB]</td>
<td>-Xmx</td>
<td>1.5</td>
<td>3</td>
<td>4.5</td>
<td>6</td>
</tr>
</tbody>
</table>

Default value of resource limits for the OC1M, OC2M, OC3M, OC4M, and OC5M compute shapes.

### Table 14-2  Resource Limit Default Values for Compute Shapes OC3, OC4, OC5, and OC6

<table>
<thead>
<tr>
<th>Resource</th>
<th>Command</th>
<th>OC1M</th>
<th>OC2M</th>
<th>OC3M</th>
<th>OC4M</th>
<th>OC5M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core File Size [Blocks]</td>
<td>ulimit -c</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
</tr>
<tr>
<td>Data Segment Size [KB]</td>
<td>ulimit -d</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
</tr>
<tr>
<td>Maximum Memory [KB]</td>
<td>ulimit -m</td>
<td>14.5</td>
<td>29.5</td>
<td>59.5</td>
<td>119.5</td>
<td>249</td>
</tr>
</tbody>
</table>
Table 14-2 (Cont.) Resource Limit Default Values for Compute Shapes OC3, OC4, OC5, and OC6

<table>
<thead>
<tr>
<th>Resource</th>
<th>Command</th>
<th>OC1M</th>
<th>OC2M</th>
<th>OC3M</th>
<th>OC4M</th>
<th>OC5M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Files</td>
<td>ulimit -n</td>
<td>262144</td>
<td>262144</td>
<td>262144</td>
<td>262144</td>
<td>262144</td>
</tr>
<tr>
<td>User Max Processes</td>
<td>ulimit -u</td>
<td>32000</td>
<td>32000</td>
<td>32000</td>
<td>32000</td>
<td>32000</td>
</tr>
<tr>
<td>Virtual Memory Size [KB]</td>
<td>ulimit -v</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
<td>unlimited</td>
</tr>
<tr>
<td>WebLogic Heap Size [MB]</td>
<td>-Xmx</td>
<td>3</td>
<td>4.5</td>
<td>6</td>
<td>12</td>
<td>20</td>
</tr>
</tbody>
</table>

Credentials for the Oracle Database that is used with your cloud service instance are used for administrative tasks related to managing services.

1. Under Database Settings, enter the user name and password for the Oracle Database.
2. Click Submit.

Setting Application-Level Configuration Properties

If you have the Service Administrator role, or the Power User role for applications that you created, you can customize applications with application-level properties. Application-level configuration properties apply to all databases in the application.

You can specify configuration properties in an application workbook on the Cube.Settings worksheet under Application Configuration. For example, in the Block Storage Sample (Stored) application workbook (sample_basic.xlsx), the DATACACHESIZE property is set to 3M and the INDEXCACHESIZE property is set to 1M. Also, you can specify configuration properties for an application that is deployed using the web UI, where you can pick configuration properties from a list.

For syntax and information about the configuration properties that you can use, see Config Settings List in the Technical Reference for Oracle Analytics Cloud - Essbase.

**Note:**

Do not attempt to modify essbase.cfg on the cloud instance file system. This file is configured automatically upon installation.

1. On the Applications home page, select the application whose configuration parameters you want to change, then click Settings.
2. In the Settings page, click the Settings tab.
   You can add or remove properties, and change the value of properties.
3. To add a property, click Add.
4. In the Property field, select a property from the list. Then, in the Value field, enter a valid value.
For example, select IGNORECONSTANTS and set the value to TRUE.

5. When you are finished making changes, click **Apply**.

6. Click **Yes**.

The application restarts.
15

Executing MaxL and MDX Statements

MaxL is a multi-dimensional database access language for Oracle Analytics Cloud - Essbase. MaxL is a practical, expressive interface for administering and querying the cloud service using statements.

MDX is a data extraction and analysis language you can use to query, export, insert, and report against data and metadata in cubes.

To use MaxL and MDX in Oracle Analytics Cloud - Essbase, you can use the Jobs interface, the esscs command-line interface (CLI), or the MaxL client utility. To run MaxL and MDX scripts, you must first upload the scripts, as .msh files, to the cloud service. You can use the CLI to upload files. See Upload: Adding Cube Files.

To run MaxL or MDX statements, you must be a power user or an administrator.

Topics
• Running MaxL and MDX Jobs in the Cloud Service
• Executing MaxL and MDX from CLI

Running MaxL and MDX Jobs in the Cloud Service

This topic does not apply to Oracle Autonomous services.

You can run MDX jobs in Oracle Autonomous Services, but not MaxL jobs.

You can run MaxL and MDX from the Jobs page of Oracle Analytics Cloud - Essbase. MaxL is an expressive interface for administering and querying the cloud service. MDX is a data and metadata extraction and analysis language.

• About MaxL and MDX Jobs
• Creating MaxL and MDX Scripts
• Uploading MaxL and MDX Scripts
• Running MaxL and MDX Scripts

About MaxL and MDX Jobs

Using the Jobs page in the cloud service, you can run MaxL scripts to create, administer, and query on applications and cubes. In the same way, you can run MDX scripts for data extraction and analysis.

To learn the syntax of MaxL statements that you can write to execute operations from MaxL scripts, see the MaxL Statement Reference in Technical Reference for Oracle Analytics Cloud - Essbase.

To learn more about MDX, see the MDX section of Technical Reference for Oracle Analytics Cloud - Essbase.
Creating MaxL and MDX Scripts

Before you can run MaxL jobs, you must write or obtain MaxL scripts, using the MaxL scripting language. For MaxL syntax, see MaxL Statement Reference in *Technical Reference for Oracle Analytics Cloud - Essbase*.

1. Write a MaxL script using a text editor.
   
   Here is an example MaxL script:
   ```
   alter system enable unicode;
   create or replace application "ASOmaxl" type unicode_mode using aggregate_storage;
   create or replace database "ASOmaxl"."Sample";
   alter system unload application ASOmaxl;
   alter system load application ASOmaxl;
   ```
   
   More sample scripts can be downloaded from *Utilities*, in the *Sample Utility Scripts* package. See the Readme in that package for more information.

2. Save the script file locally with a `.msh` extension.

   Now you are ready to upload the script to the cloud service so that you can run it.

   You should also save MDX scripts with a `.msh` extension and upload them to the cloud service. To learn more about writing MDX scripts, see Writing MDX Queries in *Designing and Maintaining Essbase Cubes*.

Uploading MaxL and MDX Scripts

This topic does not apply to Oracle Autonomous services.

You can upload MDX scripts in Oracle Autonomous Services, but not MaxL scripts.

After you write MaxL scripts to execute batch jobs, or MDX scripts for query and extraction, upload the scripts to the cloud service so that you can run them. Before uploading MaxL or MDX scripts, ensure that they are saved with a `.msh` extension.

1. On the Applications home page, select an application or cube, then click *Files*.

2. On the Files page, in *Upload Files*, select *Browse* and browse to the `.msh` script file that you want to upload.

3. Select the script file and click *Upload*.

4. Under *Files*, verify that the script file is listed.

Running MaxL and MDX Scripts

You can run saved MaxL and MDX scripts to execute batch jobs or queries in the cloud service.

1. On the Applications home page, select the application or cube to which you uploaded a MaxL or MDX script, then click *Jobs*.
2. On the Jobs page, click **New Job**.

3. On the Execute Jobs dialog box, select the **MaxL Script** job type, then select the file from the list of scripts.

4. Ignore all other options, and click **Execute**.

5. (Optional) Refresh the job status, or allow it to automatically refresh.

6. When the job completes, click **Job Details** 🕵️‍♂️ to see information about the job.

   The Job Details-MaxL Script dialog box lists the script name, and the names of two output files that you can download:

   - `script\name_Jobid_idnumber_out.txt`: A spool file containing the standard output of the MaxL job, including the statements that were executed, and any informational, error, or warning messages generated.
   - `scriptname_Jobid_idnumber_err.txt`: An error file containing any standard error output, if errors occurred.

---

### Script: Executing MaxL and MDX from CLI

This CLI command runs a MaxL or MDX script. MaxL and MDX scripts must have the `.msh` file extension.

You can run saved MaxL/MDX scripts to execute batch jobs in Essbase.

**Syntax**

```
script [-v] -script scriptfilename -application appname -db cubename
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Optional. Show extended descriptions</td>
</tr>
<tr>
<td>-script</td>
<td>-s</td>
<td>MaxL/MDX script name. Must have <code>.msh</code> file extension.</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Application name</td>
</tr>
<tr>
<td>-db</td>
<td>-d</td>
<td>Database (cube) name</td>
</tr>
</tbody>
</table>

**Notes**

- In MaxL scripts, you do not need to give a full path to any files used in MaxL statements. Files are assumed to be in the relevant application or database directory.

- To run MaxL or MDX scripts, you must first upload the scripts, as `.msh` files, to the cloud service. You can use the CLI to upload files. See [Upload: Adding Cube Files](#).

- If you use any MaxL statements that log out sessions, such as `alter system logout session`, ensure that the session-terminating statement is the last line of the MaxL script. The MaxL session will end without processing any subsequent statements in the script.

**Example**

```bash
esscs script -s abc.msh -a Sample -d Basic
```
Limitations
The following MaxL statement is not supported on the cloud service:
import database data from data_string
Oracle Analytics Cloud - Essbase
Command-Line Interface (CLI)

The command-line interface is a nongraphical interface between you and the cloud service, in which you enter shell commands to perform administrative actions on Oracle Analytics Cloud - Essbase.

• Downloading and Using the Command-Line Interface
• CLI Command Reference

Downloading and Using the Command-Line Interface

1. If it is not already installed, download and install Java SE Development Kit 8 from Oracle Technology Network.

2. Set the JAVA_HOME environment variable on your system to point to the JDK installation folder. For example, set it to `C:\Program Files\Java\jdk1.8.0_45`.

3. In the cloud service, on the Applications home page, click Utilities.

4. On the Utilities page, click Download next to the utility labeled Command Line Tool.

5. Save cli.zip to a local drive.

6. Uncompress cli.zip, and see the extracted files under the cli folder.

7. Start the command-line interface:
   • Windows: Run `esscs.bat`.
   • UNIX: Run `esscs.sh`.

CLI Command Reference

The following commands are available in the command-line interface. Arguments to commands can be issued in any order.

• calc
• clear
• createlocalconnection
• dataload
• deletefile
• deploy
• dimbuild
• download
• help
• lcmexport
• lcmimport
• listapp
• listdb
• listfiles
• listfilters
• listlocks
• listvariables
• login, logout
• script
• setpassword
• start
• stop
• upload
• version

To display help for all commands, enter `esscs -h`. To display help for a specific command, enter `esscs command -h`.

To turn on verbose output for any command, meaning that extended information (if available) is displayed, enter `esscs command -v command arguments`.

For example, `esscs script -v -application Sample -db Basic -script abc.msh`.

Login/Logout: EssCS Connection

Before you can issue CLI commands to the cloud service, you must log in. If a secure connection is required, then the URL must begin with `https`.

Syntax (login)

```
login [-verbose] -essbaseurl http://host:port/essbase -user username [-password password]
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Show extended descriptions</td>
</tr>
<tr>
<td>-essbaseurl</td>
<td>-url</td>
<td>Address of an instance of Oracle Analytics Cloud - Essbase</td>
</tr>
<tr>
<td>-user</td>
<td>-u</td>
<td>User name</td>
</tr>
</tbody>
</table>
| -password    | -p           | Optional. Password for user. Best practice is to set the password using `setpassword`.

Examples (login)

Insecure connection:
esscs login -v -url http://example:9000/essbase -u smith

Secure connection:

esscs login -url https://example:9000/essbase -u smith

Syntax (logout)

logout

Example (logout)

esscs logout

Calc: Running a Calculation Script

Calculates a cube.

To run calculation scripts, you must first upload the scripts, as .csc files, to the cloud service. You can use the CLI to upload files. See Upload: Adding Cube Files.

Syntax

calc [-verbose] -application appname -db cubename -script scriptfilename

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Show extended descriptions</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Application name</td>
</tr>
<tr>
<td>-db</td>
<td>-d</td>
<td>Database (cube) name</td>
</tr>
<tr>
<td>-script</td>
<td>-s</td>
<td>Calculation script name. Must have .csc file extension. You do not need to give a full path. Files are assumed to be in the relevant application or database directory.</td>
</tr>
</tbody>
</table>

Example

esscs calc -v -a Sample -d Basic -s CALCALL.CSC

Clear: Removing Data from a Cube

Clears data from a cube.

Syntax

clear [-verbose] -application appname -db cubename [-option clearOption][-regionspec regionSpec]]

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Optional. Show extended descriptions</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Application name</td>
</tr>
<tr>
<td>-db</td>
<td>-d</td>
<td>Database (cube) name</td>
</tr>
<tr>
<td>Option</td>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-option</td>
<td>-O</td>
<td>Optional. Keyword specifying what to clear. The options are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ALL_DATA—All data, linked objects, and the outline are cleared</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ALL_AGGREGATIONS—All aggregated data is cleared</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• PARTIAL_DATA—Only specified data region is cleared. Use with -regionspec</td>
</tr>
<tr>
<td>-regionspec</td>
<td>-rs</td>
<td>MDX expression specifying the region to clear</td>
</tr>
</tbody>
</table>

**Example**

esscs clear -a ASOSamp -d Basic -O PARTIAL_DATA -rs "((Jan],[Sale],[Cash])"

**Createlocalconnection: Saving a JDBC Connection**

Creates a JDBC connection and stores it locally.

**Description**

You must use this command to create and save the local connection before you can use the CLI `dataload` or `dimbuild` commands with the streaming option. You must also set an environment variable `EXTERNAL_CLASSPATH` to point to the .jar file for your database driver. For examples of setting this variable, see [Loading Dimensions and Data by Streaming from a Remote Database](#).

**Syntax**

```
createlocalconnection [-verbose] -name streamConnection -connectionstring connectionString -user userName [-driver jdbcDriver]
```

**Option**

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Show extended descriptions</td>
</tr>
<tr>
<td>-name</td>
<td>-N</td>
<td>Connection name</td>
</tr>
<tr>
<td>-connectionstring</td>
<td>-cs</td>
<td>JDBC connection string. Format can be with SID, as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>jdbc:oracle:thin:@host:port:SID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or with service name, as follows</td>
</tr>
<tr>
<td></td>
<td></td>
<td>jdbc:oracle:thin:@host:port/service_name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See Examples.</td>
</tr>
<tr>
<td>-user</td>
<td>-u</td>
<td>User name</td>
</tr>
<tr>
<td>-driver</td>
<td>-D</td>
<td>JDBC driver. If not provided, Oracle Database is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>considered the default, as</td>
</tr>
<tr>
<td></td>
<td></td>
<td>oracle.jdbc.driver.OracleDriver</td>
</tr>
</tbody>
</table>

**Examples**

The following examples reflect various data sources.
If the -driver option and jdbcDriver parameter are not provided, Oracle database is the assumed database by default.

**Oracle DB – Example with SID (Service ID)**

```
esscs createLocalConnection -N OracleDBConnection1 -cs
jdbc:oracle:thin:@myhostname01:1521:ORCL -u OracleUser -D
oracle.jdbc.driver.OracleDriver
```

**Oracle DB – Example with Service Name**

```
esscs createLocalConnection -N OracleDBConnection2 -cs
jdbc:oracle:thin:@host1.domain.com:1521/ORCL.esscs.host1.oraclecloud.com -u
OracleUser
```

**DB2**

```
esscs createLocalConnection -N DB2conn -cs jdbc:db2://myhostname02.domain.com:50000/TBC -u myDB2User -D com.ibm.db2.jcc.DB2Driver
```

**MySQL**

```
```

**Microsoft SQL Server**

```
esscs createLocalConnection -N MSSQLConn -cs jdbc:sqlserver://myhostname04.domain.com:1433 -u MSSQLUsr -D
com.microsoft.sqlserver.jdbc.SQLServerDriver
```

**Teradata**

```
esscs createLocalConnection -N TeraDconn -cs jdbc:teradata://myhostname05.domain.com/DBS_PORT=1025 -u MSSQLUsr -D com.teradata.jdbc.TeraDriver
```

### Dataload: Loading Data to a Cube

Loads data to a cube.

This command requires one of the following sets of options:

- Data file and optional rules file
- Rules file with user name and password
- Stream option referencing a saved local connection. For additional information about the stream option, see Loading Dimensions and Data by Streaming from a Remote Database.

To load data, you must first upload the data load and rules files to the cloud service. You can use the CLI to upload files. See Upload: Adding Cube Files.

### Syntax

```
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Show extended descriptions</td>
</tr>
<tr>
<td>Option</td>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Application name</td>
</tr>
<tr>
<td>-db</td>
<td>-d</td>
<td>Database (cube) name</td>
</tr>
<tr>
<td>-file</td>
<td>-f</td>
<td>Data load file name. You do not need to give a full path. Files are assumed to be in the relevant application or database directory.</td>
</tr>
<tr>
<td>-rule</td>
<td>-r</td>
<td>Optional. Rules file name. You do not need to give a full path. Files are assumed to be in the relevant application or database directory.</td>
</tr>
<tr>
<td>-user</td>
<td>-u</td>
<td>Optional. User name. Requires password if used.</td>
</tr>
<tr>
<td>-password</td>
<td>-p</td>
<td>Optional. Password for user. If omitted, user will be prompted for password.</td>
</tr>
<tr>
<td>-stream</td>
<td>-S</td>
<td>Optional. Use streaming data load. Requires -conn option if used. For additional information about the stream option, see Loading Dimensions and Data by Streaming from a Remote Database.</td>
</tr>
<tr>
<td>-connection</td>
<td>-conn</td>
<td>Required if streaming option is used. Name of a saved connection that was created using the CLI command createlocalconnection.</td>
</tr>
<tr>
<td>-query</td>
<td>-q</td>
<td>Optional. Database query to submit along with the streaming data load.</td>
</tr>
<tr>
<td>-rows</td>
<td>-rows</td>
<td>Optional. Number of rows to stream simultaneously. Default is 100.</td>
</tr>
<tr>
<td>-abortOnError</td>
<td>-abort</td>
<td>Abort data load if error is encountered.</td>
</tr>
</tbody>
</table>

**Example**

```
esscs dataload -application Sample -db Basic -file Calcdat.txt -abortOnError true

esscs dataload -application Sample -db Basic -rule Basic.rul -stream -connection oraConn -query "Select * from Data" -rows 50
```

**Deletefile: Removing Cube Files**

Removes cube artifacts from the application, database, or user home directory.

**Syntax**

```
deletefile [-verbose] -file fileName [-application application [-db database]]
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Show extended descriptions</td>
</tr>
<tr>
<td>-file</td>
<td>-f</td>
<td>Name of the file to delete</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Optional. Application name. If not provided, files are assumed to be in the user home directory on the cloud service.</td>
</tr>
<tr>
<td>-database</td>
<td>-db</td>
<td>Optional. Database (cube) name</td>
</tr>
</tbody>
</table>

**Example**

```
esscs deletefile -a Sample -d Basic -f Act1.rul
```
Deploy: Creating a Cube from a Workbook

Creates a cube from an Excel application workbook.

Syntax


<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Show extended descriptions</td>
</tr>
<tr>
<td>-file</td>
<td>-f</td>
<td>Name of the application workbook file</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Optional. Application name. If not provided, application name will be taken from the workbook.</td>
</tr>
<tr>
<td>-database</td>
<td>-db</td>
<td>Optional. Database (cube) name. If not provided, database name will be taken from the workbook.</td>
</tr>
<tr>
<td>-loaddata</td>
<td>-l</td>
<td>Optional. Load data, if the application workbook contains a data worksheet. Otherwise, only metadata is imported into the cube.</td>
</tr>
<tr>
<td>-recreateapplication</td>
<td>-ra</td>
<td>Optional. Re-create the application, if it already exists.</td>
</tr>
<tr>
<td>-createfiles</td>
<td>-cf</td>
<td>Optional. Create cube artifacts in the files directory on the cloud service instance.</td>
</tr>
<tr>
<td>-executescript</td>
<td>-e</td>
<td>Optional. Execute calculation scripts, if the application workbook contains a calculation worksheet.</td>
</tr>
</tbody>
</table>

Example

esscs deploy -a SampleD1 -d BasicD1 -f Sample_Basic.xlsx -l -ra -cf -e

Dimbuild: Loading Dimensions to a Cube

Loads dimensions to a cube.

To load dimensions, you must first upload the dimension-build and rules files to the cloud service. You can use the CLI to upload files. See Upload: Adding Cube Files.

Syntax

dimbuild [-verbose] -application appname -db cubename -file fileName -rule rulesFile [-user userName [-password password]] [-stream] [-connection connectionName] [-query queryString] [-rows n] [-restructureOption restructureOption] [-forcedimbuild]

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Show extended descriptions</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Application name</td>
</tr>
<tr>
<td>-db</td>
<td>-d</td>
<td>Database (cube) name</td>
</tr>
<tr>
<td>-file</td>
<td>-f</td>
<td>Dimension build file name. You do not need to give a full path. Files are assumed to be in the relevant application or database directory</td>
</tr>
<tr>
<td>Option</td>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-rule</td>
<td>-r</td>
<td>Rules file name. You do not need to give a full path. Files are assumed to be in the relevant application or database directory</td>
</tr>
<tr>
<td>-user</td>
<td>-u</td>
<td>Optional. User name. Requires password if used</td>
</tr>
<tr>
<td>-password</td>
<td>-p</td>
<td>Optional. Password for user. If omitted, user will be prompted for password</td>
</tr>
<tr>
<td>-stream</td>
<td>-S</td>
<td>Optional. Use streaming dimension build. Requires -conn option if used. For additional information about the stream option, see Loading Dimensions and Data by Streaming from a Remote Database.</td>
</tr>
<tr>
<td>-connection</td>
<td>-conn</td>
<td>Required if streaming option is used. Name of a saved connection that was created using the CLI command createlocalconnection.</td>
</tr>
<tr>
<td>-query</td>
<td>-q</td>
<td>Optional. Database query to submit along with the streaming dimension build.</td>
</tr>
<tr>
<td>-rows</td>
<td>-rows</td>
<td>Optional. Number of rows to stream simultaneously. Default is 100.</td>
</tr>
<tr>
<td>-restructureOption</td>
<td>-R</td>
<td>Controls your preservation choices for the outline restructure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For block storage, possible options are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ALL_DATA: Preserve all data when loading dimensions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• NO_DATA: Do not preserve data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• LEAFLEVEL_DATA: Preserve only level 0 data values. If all data required for calculation resides in level-0 members, then you should select this option. All upper-level blocks are deleted before the cube is restructured. When the cube is recalculated, the upper-level blocks are re-created.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• INPUT_DATA: Preserve only input data. For aggregate storage, possible options are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ALL_DATA: Preserve all data when loading dimensions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• NO_DATA: Do not preserve data.</td>
</tr>
<tr>
<td>-forcedimbuild</td>
<td>-F</td>
<td>Force the dimension build without verifying the outline.</td>
</tr>
</tbody>
</table>

**Examples**

```
esscs dimbuild -a Sample -d Basic -r Basic.rul -u smith -p password -R NO_DATA -F
```

```
esscs dimbuild -a Sample -d Basic -r Basic.rul -S -conn oraConn -q "Select * from Data" -rows 50 -R NO_DATA
```

**Download: Getting Cube Files**

Downloads cube artifacts from an instance of Essbase to a local directory. You may need to download text files, rules files, or calculation script files from a cube, so you can upload them to another cube.
Syntax

download [-verbose] -file filename [-application appname [-db cubename]] [-localdirectory path] [-overwrite] [-nocompression]

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Show extended descriptions</td>
</tr>
<tr>
<td>-file</td>
<td>-f</td>
<td>Name of file to download</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Optional. Application name. If not provided, artifacts are downloaded from the user home directory on the cloud service.</td>
</tr>
<tr>
<td>-db</td>
<td>-d</td>
<td>Optional. Database (cube) name</td>
</tr>
<tr>
<td>-localdirectory</td>
<td>-ld</td>
<td>Optional. A local directory path</td>
</tr>
<tr>
<td>-overwrite</td>
<td>-o</td>
<td>Optional. Overwrite existing file</td>
</tr>
<tr>
<td>-nocompression</td>
<td>-nc</td>
<td>Optional. Disable compression of data transfer</td>
</tr>
</tbody>
</table>

Example
esscs download -v -f Product003.rul -a Sample -d Basic -ld c:/temp -o

esscs download -file Acli.rul -ld c:/temp -o

Help: Displaying Command Syntax

Displays CLI command level help in the console or terminal.

Syntax
[command] -help | -h

Examples
esscs -help
esscs -h
esscs dataload -help

LcmExport: Backing Up Cube Files

Backs up cube artifacts to a Lifecycle Management (LCM) .zip file.

Syntax
### Option Abbreviation Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-threads</td>
<td>-T</td>
<td>Optional. Number of threads to spawn if using parallel export</td>
</tr>
<tr>
<td>-skipdata</td>
<td>-skip</td>
<td>Optional. Do not include data in the backup</td>
</tr>
<tr>
<td>-overwrite</td>
<td>-o</td>
<td>Optional. Overwrite existing backup file</td>
</tr>
</tbody>
</table>

**Notes**

- This command, like other CLI commands, can be used from outside the cloud service virtual machine, whereas the LCM utility must be run within the cloud service virtual machine.

**Example**

```
esscs lcmExport -v -a Sample -z Sample.zip -ld c:/temp -skip -o
```

### LcmImport: Restoring Cube Files

**Restores cube artifacts from a Lifecycle Management (LCM) .zip file.**

**Syntax**

```
lcmImport [-verbose] -zipfilename filename [-overwrite]
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Optional. Show extended descriptions</td>
</tr>
<tr>
<td>-zipfilename</td>
<td>-z</td>
<td>Name of compressed file containing backup files</td>
</tr>
<tr>
<td>-overwrite</td>
<td>-o</td>
<td>Optional. Recreate the target application.</td>
</tr>
</tbody>
</table>

**Notes**

- This command, like other CLI commands, can be used from outside the cloud service virtual machine, whereas the LCM utility must be run within the cloud service virtual machine.
- When partitions exist between cubes being migrated, you must import the data source before the data target. Otherwise, partition definitions may not be restored.

**Example**

```
esscs lcmImport -z C:/Sample/Sample.zip -o
```

### Listapp: Displaying Applications

**Lists applications that exist on an instance of Oracle Analytics Cloud - Essbase.**

**Syntax**

```
listapp [-verbose] [details]
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Optional. Show extended descriptions</td>
</tr>
</tbody>
</table>
Option | Abbreviation | Description
--- | --- | ---
-details | -dtl | Optional. Display more details in the output.

**Example**

esacs listapp

**Listdb: Displaying Cubes**

Lists databases that exist within an application on Oracle Analytics Cloud - Essbase.

**Syntax**

listdb [-verbose] [details]

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Optional. Show extended descriptions</td>
</tr>
<tr>
<td>-details</td>
<td>-dtl</td>
<td>Optional. Display more details in the output.</td>
</tr>
</tbody>
</table>

**Example**

esacs listdb

**Listfiles: Displaying Files**

Lists cube artifacts that exist on an instance of Oracle Analytics Cloud - Essbase. Cube artifacts may include text files, rules files, calculation script files, or MaxL script files. Cube artifacts are any files that are needed to perform actions on applications and cubes.

**Syntax**

listfiles [-verbose] [-type filetype] [-application appname [-db cubename]]

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Optional. Show extended descriptions</td>
</tr>
</tbody>
</table>
| -type | -t | Optional. File extension/type to display, not including the period. Supported file types are:
  • .csc (calculation scripts)
  • .rul (rules files)
  • .txt (text files)
  • .msh (MaxL scripts)
  • .xls, .xlsx (Excel workbooks)
  • .xlam (macro-enabled Excel workbooks)
  • .xml (XML files)
  • .zip (compressed zip files)
  • .csv (comma-separated files)
| -application | -a | Optional. Application name. If not provided, files from the user home directory on the cloud service are displayed. |
### Listfilters: Viewing Security Filters

View a list of Essbase security filters.

**Syntax**

```
listfilters [-verbose] -application appname -db cubename
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Optional. Show extended descriptions</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Application name</td>
</tr>
<tr>
<td>-db</td>
<td>-d</td>
<td>Database (cube) name</td>
</tr>
</tbody>
</table>

**Example**

```
esscs listfilters -v -a Sample -d Basic
```

### Listlocks: Viewing Locks

View any locked data blocks or cube-related objects.

**Syntax**

```
listlocks [-verbose] -application appname -db cubename [-object]
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Optional. Show extended descriptions</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Application name</td>
</tr>
<tr>
<td>-db</td>
<td>-d</td>
<td>Database (cube) name</td>
</tr>
<tr>
<td>-object</td>
<td>-obj</td>
<td>Optional. Display locked files/artifacts.</td>
</tr>
</tbody>
</table>

**Example**

```
esscs listlocks -v -a Sample -d Basic -obj
```

### Listvariables: Displaying Substitution Variables

Lists substitution variables defined at the application or cube scope.

**Syntax**

```
listvariables [-verbose] -application application [-db database]
```
<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Show extended descriptions</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Optional. Application name.</td>
</tr>
<tr>
<td>-database</td>
<td>-db</td>
<td>Optional. Database (cube) name</td>
</tr>
</tbody>
</table>

**Example**

```
esscs listvariables -a Sample -d Basic
```

**Script: Executing MaxL and MDX from CLI**

This CLI command runs a MaxL or MDX script. MaxL and MDX scripts must have the `.msh` file extension.
You can run saved MaxL/MDX scripts to execute batch jobs in Essbase.

**Syntax**

```
script [-v] -script scriptfilename -application appname -db cubename
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Optional. Show extended descriptions</td>
</tr>
<tr>
<td>-script</td>
<td>-s</td>
<td>MaxL/MDX script name. Must have <code>.msh</code> file extension.</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Application name</td>
</tr>
<tr>
<td>-db</td>
<td>-d</td>
<td>Database (cube) name</td>
</tr>
</tbody>
</table>

**Notes**

- In MaxL scripts, you do not need to give a full path to any files used in MaxL statements. Files are assumed to be in the relevant application or database directory.
- To run MaxL or MDX scripts, you must first upload the scripts, as `.msh` files, to the cloud service. You can use the CLI to upload files. See [Upload: Adding Cube Files](#).
- If you use any MaxL statements that log out sessions, such as `alter system logout session`, ensure that the session-terminating statement is the last line of the MaxL script. The MaxL session will end without processing any subsequent statements in the script.

**Example**

```
esscs script -s abc.msh -a Sample -d Basic
```

**Limitations**

The following MaxL statement is not supported on the cloud service:

```
import database data from data_string
```
Setpassword: Storing CLI Credentials

Sets a password for a CLI user, and stores the user credentials so that the user can log in without entering a password.

Syntax

```
setpassword [-v] -url URL -u userName
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Show extended descriptions</td>
</tr>
<tr>
<td>-url</td>
<td>-url</td>
<td>Address of an instance of Oracle Analytics Cloud - Essbase</td>
</tr>
<tr>
<td>-user</td>
<td>-u</td>
<td>The user whose password to set</td>
</tr>
</tbody>
</table>

Example

```
esscs setpassword -url http://example:9000/essbase -user smith
```

Start: Starting an Application or Cube

Start an Essbase application or cube, loading it into memory.

Syntax

```
start [-verbose] -application appname [-db cubename]
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Optional. Show extended descriptions</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Application name</td>
</tr>
<tr>
<td>-db</td>
<td>-d</td>
<td>Optional. Database (cube) name</td>
</tr>
</tbody>
</table>

Example

```
esscs start -v -a Sample -d Basic
```

Stop: Stopping an Application or Cube

Stop an Essbase application or cube.

Syntax

```
stop [-verbose] -application appname [-db cubename]
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Optional. Show extended descriptions</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Application name</td>
</tr>
<tr>
<td>-db</td>
<td>-d</td>
<td>Optional. Database (cube) name</td>
</tr>
</tbody>
</table>
Example
esscs stop -v -a Sample -d Basic

Upload: Adding Cube Files

Uploads cube artifacts from a local directory to an instance of Oracle Analytics Cloud - Essbase. To perform tasks such as data loads, dimension builds, calculations, or running MaxL scripts, you may need to upload text files, rules files, calculation script files, or MaxL script files to a cube.

Syntax
upload [-verbose] -file filename [-application appname [-db cubename]] [-overwrite] [-nocompression] [-compressionalgorithm]

<table>
<thead>
<tr>
<th>Option</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-verbose</td>
<td>-v</td>
<td>Optional. Show extended descriptions</td>
</tr>
<tr>
<td>-file</td>
<td>-f</td>
<td>Name of file to upload</td>
</tr>
<tr>
<td>-application</td>
<td>-a</td>
<td>Optional. Application name. If not provided, files are uploaded from the user home directory on the cloud service.</td>
</tr>
<tr>
<td>-db</td>
<td>-d</td>
<td>Optional. Database (cube) name</td>
</tr>
<tr>
<td>-overwrite</td>
<td>-o</td>
<td>Optional. Overwrite existing file</td>
</tr>
<tr>
<td>-nocompression</td>
<td>-nc</td>
<td>Optional. Disable compression of data transfer</td>
</tr>
<tr>
<td>-compressionalgorithm</td>
<td>-ca</td>
<td>Optional. Available if -nc is not used. Defines which compression algorithm to use for data transfer. Possible choices: gzip or lz4.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• gzip—Default if compression is used. Provides smaller data transfer with slower calculation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• lz4—Provides faster calculation with a slower data transfer.</td>
</tr>
</tbody>
</table>

Usage examples:
- ca gzip
- ca lz4

Example
esscs upload -v -f c:/temp/Maxl01.msh -a Sample -d Basic -o -ca lz4

Version: Displaying API Version

Gets the version of the REST API that is behind this instance of Oracle Analytics Cloud - Essbase.

Syntax
version

Example
esscs version
Working with Logs

You can download and view logs at the server level and at the applications level.

Downloading Server and Application Logs

As a Service Administrator, you can download all logs (server and application). You can download the latest log as well as rolled over logs.

1. On the Applications home page, click Admin without selecting an application.
2. Expand Server and Applications to see the available logs.
3. Click Download and select Download Latest or Download All.

You can download these kinds of server logs:

- Managed Server: Warnings and error messages regarding Weblogic issues
- Console Output: Managed server run-time exceptions and messages
- Platform: User interface
- Agent
- Application Provider Service

As an Application Manager, you can download applications logs. You can download the latest log, as well as rolled over logs.

1. On the Applications home page, select an application, then click Admin.
2. Click Logs.
3. Select Latest or All, then click Download.
Oracle recommends that you download a sample application workbook and examine the worksheets to familiarize yourself on how to design your own application and cube.

- Understanding the Essbase.Cube Worksheet
- Understanding the Cube.Settings Worksheet
- Understanding the Cube.Generations Worksheet
- Understanding the Cube.Textlists Worksheet
- Understanding Dimension Worksheets
- Understanding Data Worksheets
- Understanding Calculation Worksheets
- About Using Tabular Data to Create Cubes

Also see Downloading a Sample Application Workbook.

### Understanding the Essbase.Cube Worksheet

The Essbase.Cube worksheet defines the application and cube name and dimension information, such as dimension names, types, storage (dense or sparse) and outline order.

The following image shows the Essbase.Cube worksheet in a sample application workbook.

<table>
<thead>
<tr>
<th>Application Name</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name</td>
<td>Basic</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

#### Dimension Definitions

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Type</th>
<th>Storage</th>
<th>Outline</th>
<th>Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>Time</td>
<td>Dense</td>
<td>1</td>
<td>Product</td>
</tr>
<tr>
<td>Measures</td>
<td>Accounts</td>
<td>Dense</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Product</td>
<td>Regular</td>
<td>Sparse</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Market</td>
<td>Regular</td>
<td>Sparse</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Scenario</td>
<td>Regular</td>
<td>Sparse</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Caffeinated</td>
<td>Attribute-Boolean</td>
<td></td>
<td>6</td>
<td>Product</td>
</tr>
<tr>
<td>Ounces</td>
<td>Attribute-Numeric</td>
<td></td>
<td>7</td>
<td>Product</td>
</tr>
<tr>
<td>Pkg Type</td>
<td>Attribute-Text</td>
<td></td>
<td>8</td>
<td>Product</td>
</tr>
<tr>
<td>Population</td>
<td>Attribute-Numenc</td>
<td></td>
<td>9</td>
<td>Market</td>
</tr>
<tr>
<td>Intro Date</td>
<td>Attribute-Date</td>
<td></td>
<td>10</td>
<td>Product</td>
</tr>
<tr>
<td>Property or Field</td>
<td>Valid Values</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Application Name   | • The application name must not exceed 30 characters.  
• Do not use spaces.  
• Application names are not case-sensitive.  
• The following special characters are not allowed: % $ - { } ( ) ! ~ ` # & @ ^ | Enter the name of the application. |
| Database Name      | • The cube name must not exceed 30 characters.  
• Do not use spaces.  
• Cube names are not case-sensitive.  
• The following special characters are not allowed: % $ - { } ( ) ! ~ ` # & @ ^ | Enter the name of the cube. |
| Version            | This must be a positive integer.                                                                                                                                                                            | This is the application workbook version. |
| Dimension Name     | Dimension names cannot be the same as the cube name.                                                                                                                                                        | Enter the name of each dimension. There must be at least two dimensions in a cube. For block storage, one dimension must be a dense dimension. Use no more than 1024 characters when naming dimensions, members, or aliases. The following special characters are not allowed: @, ,, !, {, }, /, *. |
| Dimension Type     | • Time  
• Accounts  
• Regular  
• Attribute-Boolean  
• Attribute-Numeric  
• Attribute-Text  
• Attribute-Date | Describes the type of dimension. Regular is the Default. Per cube, you can only use one Time and one Accounts dimension type. |
| Dimension Storage  | • Dense  
• Sparse | Sparse is the default. There must be at least one dense dimension. |
| Outline Order      | This must be a positive integer.                                                                                                                                                                            | This is the order of the dimension in the outline. Attribute dimensions must be ordered after base dimensions. |
Understanding the Cube.Settings Worksheet

The Cube.Settings worksheet defines the application type (aggregate storage or block storage) and many cube and outline properties such as dynamic time series members and substitution variables.

Each of the five sections in the Cube.Settings worksheet has information about its fields and values, and how to modify those fields and values by using the Cube Designer wizard.

- Understanding the Cube.Settings Worksheet: Alias Tables
- Understanding the Cube.Settings Worksheet: Properties
- Understanding the Cube.Settings Worksheet: Dynamic Time Series
- Understanding the Cube.Settings Worksheet: Attribute Settings
- Understanding the Cube.Settings Worksheet: Substitution Variables

Understanding the Cube.Settings Worksheet: Alias Tables

This section of the Cube Settings worksheet lists alias tables that need to be created for the cube.

It must contain at least the Default row.

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Default</td>
<td>Every cube has a table named Default. You can create additional alias tables in the rows following the Default row.</td>
</tr>
<tr>
<td>Rows following the default row. These new rows can be created manually, or using the Cube Designer wizard.</td>
<td>Naming conventions for member names apply. See Naming Conventions for Dimensions, Members, and Aliases in Designing and Maintaining Essbase Cubes.</td>
<td>You can set multiple aliases for a member using multiple alias tables.</td>
</tr>
</tbody>
</table>

Note:

You define alias table names on the Cube.Settings worksheet. You define the contents of the alias tables on the dimension worksheets.
See Setting Aliases in *Designing and Maintaining Essbase Cubes*.

### Understanding the Cube.Settings Worksheet: Properties

The following table shows the fields, values and descriptions for the Properties section on the Cube.Settings worksheet:

**Table A-2  Properties Section of the Cube.Settings Worksheet**

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application Type</strong></td>
<td>• ASO</td>
<td>This is an application property. Defines whether the cubes in the application use aggregate storage (ASO) or block storage (BSO).</td>
</tr>
<tr>
<td></td>
<td>• BSO</td>
<td></td>
</tr>
<tr>
<td><strong>Outline Type</strong></td>
<td>• Unique</td>
<td>This is a database property. • Unique: member names in the outline must be unique. • Duplicate: Duplicate member names are permitted in the outline.</td>
</tr>
<tr>
<td></td>
<td>• Duplicate</td>
<td></td>
</tr>
<tr>
<td><strong>Aggregate missing values</strong></td>
<td>• Yes</td>
<td>This is a database property. Defines whether missing (#MISSING) values are aggregated during a cube calculation.</td>
</tr>
<tr>
<td></td>
<td>• No</td>
<td></td>
</tr>
<tr>
<td>Property or Field</td>
<td>Valid Values</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Create blocks on equations | • Yes  
• No          | This is a database property. If you enter Yes, then when you assign a nonconstant value to a member combination for which no data block exists, a data block is created. Entering Yes can produce a very large cube. Sometimes, new blocks are not desired; for example, when they contain no other values. In large databases, creation and processing of unneeded blocks can increase processing time and storage requirements. For more specific control, you can use the SET CREATEBLOCKONEQ calculation command within a calculation script to control creation of blocks at the time the command is encountered in the script. See the SET CREATEBLOCKONEQ calculation command in Technical Reference for Oracle Analytics Cloud - Essbase. |
| Two-Pass calculation     | • Yes  
• No          | This is a database property. If you enter Yes, then after a default calculation, members that are tagged as two-pass are recalculated, overwriting the aggregation results from the first calculation pass. The two-pass tag is effective on members of the dimension tagged as Accounts and on Dynamic Calc and Dynamic Calc and Store members of any dimension. |
| Date Format              | There are many valid date formats. These are some examples:  
• mm dd yyyy  
• dd mm yy  
• mm/dd/yy  
• mm-dd-yyyy | This is a database property. You can set the format of member names in date attribute dimensions. If you change the date format, then you must rebuild the date attribute dimensions and reassociate dimension members. |
Table A-2  (Cont.) Properties Section of the Cube.Settings Worksheet

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario Sandboxes</td>
<td>• 0</td>
<td>This value defines whether the cube contains a sandbox dimension for creating scenarios of the data, and the number of sandbox members within the sandbox dimension. A value of 0 indicates no sandbox dimension.</td>
</tr>
<tr>
<td></td>
<td>• A positive integer less than 1000.</td>
<td></td>
</tr>
</tbody>
</table>

You can modify the Properties section on the Cube.Settings worksheet in the Cube Designer wizard. See Working with the Cube.Settings Worksheet: Properties in Cube Designer.

Understanding the Cube.Settings Worksheet: Dynamic Time Series

Table A-3  Dynamic Time Series Section of the Cube.Settings Worksheet

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H-T-D</td>
<td>Integer value representing the generation number</td>
<td>History to date</td>
</tr>
<tr>
<td>Y-T-D</td>
<td>Integer value representing the generation number</td>
<td>Year to date</td>
</tr>
<tr>
<td>S-T-D</td>
<td>Integer value representing the generation number</td>
<td>Season to date</td>
</tr>
<tr>
<td>P-T-D</td>
<td>Integer value representing the generation number</td>
<td>Period to date</td>
</tr>
<tr>
<td>Q-T-D</td>
<td>Integer value representing the generation number</td>
<td>Quarter to date</td>
</tr>
<tr>
<td>M-T-D</td>
<td>Integer value representing the generation number</td>
<td>Month to date</td>
</tr>
<tr>
<td>W-T-D</td>
<td>Integer value representing the generation number</td>
<td>Week to date</td>
</tr>
<tr>
<td>D-T-D</td>
<td>Integer value representing the generation number</td>
<td>Day to date</td>
</tr>
</tbody>
</table>


See Using Dynamic Time Series Members in Designing and Maintaining Essbase Cubes.

Understanding the Cube.Settings Worksheet: Attribute Settings

The following table shows the fields, values and descriptions for the Attribute Settings section on the Cube.Setting worksheet:
Table A-4  Attribute Settings

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension Name</td>
<td>Default: Attributes Calculation</td>
<td>To avoid duplicating names in an outline, you can change the names of members of the attribute calculations dimension. Regardless of the name that you use for a member, the function of the member remains the same. For example, the Sum member always calculates a sum, no matter what you name it. See Changing the Member Names of the Attribute Calculations Dimension in Designing and Maintaining Essbase Cubes.</td>
</tr>
<tr>
<td>Sum Member</td>
<td>Default: Sum</td>
<td>This is a member of the attribute calculations dimension. The name to use when requesting sum data.</td>
</tr>
<tr>
<td>Count Member</td>
<td>Default: Count</td>
<td>This is a member of the attribute calculations dimension. The name to use when requesting count data.</td>
</tr>
<tr>
<td>Minimum Member</td>
<td>Default: Min</td>
<td>This is a member of the attribute calculations dimension. The name to use when requesting minimum data.</td>
</tr>
<tr>
<td>Maximum Member</td>
<td>Default: Max</td>
<td>This is a member of the attribute calculations dimension. The name to use when requesting maximum data.</td>
</tr>
<tr>
<td>Average Member</td>
<td>Default: Avg</td>
<td>This is a member of the attribute calculations dimension. The name to use when requesting average data.</td>
</tr>
<tr>
<td>False Member</td>
<td>Default: False</td>
<td>The initial Boolean member names in a cube are set as True and False. See Setting Boolean Attribute Member Names in Designing and Maintaining Essbase Cubes.</td>
</tr>
</tbody>
</table>
### Table A-4  (Cont.) Attribute Settings

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Member</td>
<td>Default: True</td>
<td>The initial Boolean member names in a cube are set as True and False.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See Setting Boolean Attribute Member Names in <em>Designing and Maintaining Essbase Cubes</em>.</td>
</tr>
<tr>
<td>Prefix/Suffix Value</td>
<td>• None</td>
<td>See Setting Prefix and Suffix Formats for Member Names of Attribute Dimensions in <em>Designing and Maintaining Essbase Cubes</em>.</td>
</tr>
<tr>
<td></td>
<td>• Dimension</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Parent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Grandparent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Ancestors</td>
<td></td>
</tr>
<tr>
<td>Prefix/Suffix Format</td>
<td>• Prefix</td>
<td>You can define unique names by attaching a prefix or suffix to member names in Boolean, date, and numeric attribute dimensions in the outline.</td>
</tr>
<tr>
<td></td>
<td>• Suffix</td>
<td>See Setting Prefix and Suffix Formats for Member Names of Attribute Dimensions in <em>Designing and Maintaining Essbase Cubes</em>.</td>
</tr>
<tr>
<td>Prefix/Suffix Separator</td>
<td>• _ Underscore</td>
<td>You can define unique names by attaching a prefix or suffix to member names in Boolean, date, and numeric attribute dimensions in the outline.</td>
</tr>
<tr>
<td></td>
<td>•</td>
<td>Pipe</td>
</tr>
<tr>
<td></td>
<td>• ^ Carat</td>
<td>Select a separator (to place between the prefix or suffix and the original name): underscore (_), pipe (</td>
</tr>
<tr>
<td>Attribute Numeric Ranges</td>
<td>• Tops of ranges</td>
<td>See Setting Up Member Names Representing Ranges of Values in <em>Designing and Maintaining Essbase Cubes</em>.</td>
</tr>
<tr>
<td></td>
<td>• Bottoms of ranges</td>
<td></td>
</tr>
<tr>
<td>Date Member</td>
<td>• Month First (mm-dd-yyyy)</td>
<td>You can change the format of members of date attribute dimensions. See Changing the Member Names in Date Attribute Dimensions in <em>Designing and Maintaining Essbase Cubes</em>.</td>
</tr>
<tr>
<td></td>
<td>• Day First (dd-mm-yyyy)</td>
<td></td>
</tr>
</tbody>
</table>

You can modify the Attribute Settings section on the Cube.Settings worksheet in the Cube Designer wizard. See *Working with the Cube.Settings Worksheet: Attribute Settings in Cube Designer*. 
Understanding the Cube.Settings Worksheet: Substitution Variables

Substitution variables act as global placeholders for information that changes regularly. You create the variable and a corresponding string value, and the value can then be changed at any time.

A substitution variable can be used in a query or calculation script to represent a member in the outline. By default, there are no substitution variables defined for a cube.

**Note:**

There is not an option to add substitution variables in the Cube Designer wizard, however you can add them directly in the application workbook.

1. On the Cube.Settings worksheet, in the Substitution Variables section, create a new row.
2. Enter the variable name in column A and its value in column B, enclosing the value in quotation marks.
   Example:
   `CurrMonth "Jan`

See Using Substitution Variables in *Designing and Maintaining Essbase Cubes*.

Understanding the Cube.Generations Worksheet

**Cube.Generations Worksheets**

The Cube.Generations worksheet is used for naming generations in an outline.

The term “generation” indicates the distance of a member from the root of the dimension. Using a generation number, you can determine the location of members within the database tree. All members in a database that are the same number of branches from their root have the same generation number. The dimension is generation 1, its children are generation 2, and so on.

You can create names for generations in an outline, such as a word or phrase that describes the generation. For example, you might create a generation name called Cities for all cities in the outline.

You can also use generation names in calculation scripts wherever you need to specify a list of generation numbers. For example, you could limit a calculation in a calculation script to all members in a specific generation.

You can specify only one name per generation. The specified name must be unique; that is, it cannot duplicate a generation, level, or member name or an alias or conventional alias.
Note:
The Dimension section of the Cube.Generations worksheet changes if you change the dimension worksheet (Dim.dimname) by adding or deleting members in such a way that the number of generations in the dimension is changed. If you make changes to the dimension worksheet by adding or deleting members, you should always press the Update Generation Worksheet button on the Dimensions tab of the Cube Designer wizard as part of the editing process.

Cube.Generations Worksheet Format

The following image shows a Cube.Generations worksheet in a sample application workbook.

<table>
<thead>
<tr>
<th>Generation Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimension Name</strong></td>
</tr>
<tr>
<td>Generation Number</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dimension Name</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation Number</td>
<td>Generation Name</td>
</tr>
<tr>
<td>2</td>
<td>Category</td>
</tr>
<tr>
<td>3</td>
<td>Line</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dimension Name</th>
<th>Market</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation Number</td>
<td>Generation Name</td>
</tr>
<tr>
<td>1</td>
<td>Market1</td>
</tr>
<tr>
<td>2</td>
<td>m2</td>
</tr>
<tr>
<td>3</td>
<td>m3</td>
</tr>
</tbody>
</table>

Table A-5 Fields and Valid Values in Generation Worksheets

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension Name</td>
<td>For dimension naming restrictions, see Naming Conventions for Dimensions, Members, and Aliases for naming restrictions.</td>
<td>The dimension name.</td>
</tr>
</tbody>
</table>
Table A-5  (Cont.) Fields and Valid Values in Generation Worksheets

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation Number</td>
<td>A generation number, 1 or greater.</td>
<td>A root branch of the tree is generation 1. Generation numbers increase as you count from the root toward the leaf member.</td>
</tr>
<tr>
<td>Generation Name</td>
<td>You can define only one name for each generation. When you name generations, follow the same naming rules as for members. See Naming Conventions for Dimensions, Members, and Aliases.</td>
<td>The generation name. You can use this field to create or change generation names. Enter the generation name and then build or update the cube using the application workbook. See Updating Cubes Incrementally in Cube Designer.</td>
</tr>
<tr>
<td>Unique</td>
<td>• Yes</td>
<td>For duplicate member name outlines, enter Yes to require unique member names within the associated generation.</td>
</tr>
<tr>
<td></td>
<td>• No</td>
<td></td>
</tr>
</tbody>
</table>

Understanding the Cube.Textlists Worksheet

In application workbooks, the Cube.Textlists worksheet defines text lists. Text lists are used to work with text measures, which extend the analytical capabilities of Essbase Cloud. In addition to numeric values, measures can be associated with text-typed values. Storage and analysis of textual content can be useful when a cell needs to have one of a finite list of textual values; for example, a product may be sold in 5 different colors. The color is a text measure whose value must be one of the 5 colors. The colors are a set of text strings mapped to corresponding numeric IDs. These mappings are contained in tables in the Cube.Textlists worksheet.

You can add multiple text list tables to the same sheet and they can be associated with multiple measures.

The following image shows the Cube.Textlists worksheet in a sample application workbook.
Table A-6  Cube.Textlists Worksheet Fields and Values

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List Name</td>
<td>Must not exceed 80 characters.</td>
<td>A text list must start with a list name followed by its value in the adjacent cell.</td>
</tr>
<tr>
<td>Associated Members</td>
<td>Existing member names.</td>
<td>Member names added in adjacent cells. Multiple members can be added in adjacent cells to the right.</td>
</tr>
<tr>
<td>ID</td>
<td>The first two values under ID are #Missing and #OutOfRange. These two values must exist in every text list table. The other IDs must be integers.</td>
<td>Each ID, including the #Missing, #OutOfRange and numeric values, must map to a text value. The first two IDs, #Missing and #OutOfRange, are for handling cases where the textual data is invalid or empty. For example, if you try to load an unmapped value such as “Average” to a text measure, the cell value would not be updated, and would display as #Missing in a subsequent query. If you load a numeric cell value that is unmapped, the subsequent query would return N/A.</td>
</tr>
<tr>
<td>Text</td>
<td>Up to 80 characters.</td>
<td>The text column contains the text values for each text measure. Each text value must map to an integer in the ID column. Any text value that does not map to an integer in the text list is considered by Essbase Cloud to be invalid.</td>
</tr>
</tbody>
</table>

In *Designing and Maintaining Essbase Cubes*, see:

- Working with Typed Measures
- Performing Database Operations on Text and Date Measures

**Understanding Dimension Worksheets**

Application workbooks contain one dimension worksheet for each of the dimensions listed in the Essbase.Cube worksheet. The name of each dimension worksheet is Dim.dimname; for example, the Year dimension worksheet is called Dim.Year. Dimension names can contain up to 1024 characters, but long dimension names (longer than 31 characters, including "Dim.") are truncated in the dimension sheet name.

Dimension worksheets use load rules syntax. For example, an X in the Storage column means that the data value is not stored.
The following image shows a dimension worksheet in a sample application workbook.

### Dimension Name: Year

**Definitions**

<table>
<thead>
<tr>
<th>Property</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension Name</td>
<td>The name of the dimension. Do not change the dimension name in this field.</td>
<td>Any dimension or attribute dimension in the outline. Defined on the Essbase.Cube worksheet.</td>
</tr>
<tr>
<td>File Name</td>
<td>A valid string. The file name cannot be longer than thirty characters.</td>
<td>The build process creates a data file with a .txt extension in the cloud service for every data worksheet in the application workbook. You can give them meaningful names so that they are easily recognizable if they need to be used again.</td>
</tr>
<tr>
<td>Rule Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Build Method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incremental Mode</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Members**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>X</td>
<td></td>
<td></td>
<td>年</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Q4r1</td>
<td>Qtr1</td>
<td>X</td>
<td></td>
<td>第一季</td>
<td>2</td>
<td>第一四半期</td>
</tr>
<tr>
<td>Q4r2</td>
<td>Jan</td>
<td></td>
<td></td>
<td>一月</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Q4r2</td>
<td>Feb</td>
<td></td>
<td></td>
<td>二月</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Q4r2</td>
<td>Mar</td>
<td></td>
<td></td>
<td>三月</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td>Qtr2</td>
<td>X</td>
<td></td>
<td>第二季</td>
<td>6</td>
<td>第二四半期</td>
</tr>
<tr>
<td>Q4r2</td>
<td>Apr</td>
<td></td>
<td></td>
<td>四月</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Q4r2</td>
<td>May</td>
<td></td>
<td></td>
<td>五月</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Q4r2</td>
<td>Jun</td>
<td></td>
<td></td>
<td>六月</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td>Qtr3</td>
<td>X</td>
<td></td>
<td>第三季</td>
<td>10</td>
<td>第三四半期</td>
</tr>
<tr>
<td>Q4r3</td>
<td>Jul</td>
<td></td>
<td></td>
<td>七月</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Q4r3</td>
<td>Aug</td>
<td></td>
<td></td>
<td>八月</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Q4r3</td>
<td>Sep</td>
<td></td>
<td></td>
<td>九月</td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>

### Table A-7  Fields and Valid Values in Dimension Worksheets

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension Name</td>
<td></td>
<td>The name of the dimension. Do not change the dimension name in this field.</td>
</tr>
<tr>
<td>File Name</td>
<td></td>
<td>A valid string. The file name cannot be longer than thirty characters.</td>
</tr>
</tbody>
</table>
### Table A-7  (Cont.) Fields and Valid Values in Dimension Worksheets

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Name</td>
<td>A valid string. See Name and Related Artifact Limits in <em>Designing and Maintaining Essbase Cubes</em>. The rule name cannot be longer than thirty characters.</td>
<td>The build process creates a rule file with a .rul extension in the cloud service for every dimension worksheet in the workbook. You can give them meaningful names so that they are easily recognizable if they need to be used again.</td>
</tr>
<tr>
<td>Build Method</td>
<td>• PARENT-CHILD • GENERATION</td>
<td>In Cube Designer wizard, you can build a cube with either build method, but you cannot edit a cube built using the Generation build method using the wizard, and you cannot view hierarchies using Cube Designer Dimension Hierarchy viewer.</td>
</tr>
<tr>
<td>Incremental Mode</td>
<td>• Remove Unspecified • Merge</td>
<td>Incremental dimension builds enable you to update existing dimensions with new members.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Merge is the default. This option adds the new members to the dimension while retaining the existing members.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Remove Unspecified removes members that are not specified in the source file.</td>
</tr>
<tr>
<td>Delimiter</td>
<td>The values can be a tab, a space, or any single character except &quot;.&quot;</td>
<td>This value must be updated directly in the Excel sheet. It cannot be updated using the Cube Designer interface.</td>
</tr>
<tr>
<td>Header Rows to Skip</td>
<td>A positive number or zero. Zero is the default.</td>
<td>The number of header rows to skip when performing a data load or dimension build. This value must be updated directly in the Excel sheet. It cannot be updated using the Cube Designer interface.</td>
</tr>
<tr>
<td>Allow Moves</td>
<td>• Yes • No</td>
<td>Within a dimension, moves members and their children to new parents; recognizes primary members and matches them with the data source; not available for duplicate member outlines. This value must be updated directly in the Excel sheet. It cannot be updated using the Cube Designer interface.</td>
</tr>
<tr>
<td>Property or Field</td>
<td>Valid Values</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Member ID</td>
<td>Any unique key</td>
<td>Used to uniquely identify a member in an outline. Required for duplicate outlines.</td>
</tr>
<tr>
<td>Storage Type</td>
<td>N</td>
<td>Never allow data sharing.</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td>Tag as label only (store no data).</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>Set member as stored (non dynamic calc and not label only).</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>Create as dynamic calc.</td>
</tr>
<tr>
<td>Consolidation Operator</td>
<td>+</td>
<td>+ (add)</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>- (subtract)</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>* (multiply)</td>
</tr>
<tr>
<td></td>
<td>/</td>
<td>/ (divide)</td>
</tr>
<tr>
<td></td>
<td>%</td>
<td>% (percent)</td>
</tr>
<tr>
<td></td>
<td>~</td>
<td>~ (no operation)</td>
</tr>
<tr>
<td></td>
<td>^</td>
<td>^ (never consolidate)</td>
</tr>
<tr>
<td>IGNORE</td>
<td>Ignore</td>
<td>Data in a column with the heading, IGNORE is ignored during data loads and dimension builds.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value must be updated directly in the Excel sheet. It cannot be updated using the Cube Designer interface.</td>
</tr>
<tr>
<td>Two-Pass Calculation</td>
<td>Yes</td>
<td>If you enter Yes, after a default calculation, then members that are tagged as two-pass are recalculated.</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>The two-pass tag is effective on members of the dimension tagged as Accounts and on Dynamic Calc and Dynamic Calc and Store members of any dimension.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Two-pass calculation applies only to block storage outlines.</td>
</tr>
</tbody>
</table>
### Table A-7  (Cont.) Fields and Valid Values in Dimension Worksheets

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solve Order</td>
<td>Any number, 0 to 127</td>
<td>Assign a calculation priority (0-127). The formula on the dimension or member that is assigned the highest solve order is calculated first. Values less than 0 or greater then 127 are reset to 0 and 127 respectively. The default value is 0. Members that are not assigned a solve order are assigned the solve order of their dimension.</td>
</tr>
<tr>
<td>Time Balance</td>
<td>A</td>
<td>Treat as an average time balance item (Applies to accounts dimensions only). Uses load rules member property codes. See Using the Data Source to Work with Member Properties in <em>Designing and Maintaining Essbase Cubes</em>.</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>Treat as the first time balance item (Applies to accounts dimensions only). Time balance properties provide instructions about how to calculate data in the Accounts dimension. See Setting Time Balance Properties in <em>Designing and Maintaining Essbase Cubes</em>.</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>Treat as the last time balance item (Applies to accounts dimensions only). Uses load rules member property codes. See Using the Data Source to Work with Member Properties in <em>Designing and Maintaining Essbase Cubes</em>.</td>
</tr>
<tr>
<td>Skip Value</td>
<td>B</td>
<td>Exclude data values of zero or #MISSING in the time balance (applies to accounts dimensions only). Uses load rules member property codes. See Using the Data Source to Work with Member Properties in <em>Designing and Maintaining Essbase Cubes</em>.</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>Exclude data values of #MISSING from the time balance (applies to accounts dimensions only). If you set the time balance as first, last, or average, then set the Skip property to indicate what to do when missing values or values of 0 are encountered. See Setting Skip Properties in <em>Designing and Maintaining Essbase Cubes</em>.</td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>Exclude data values of zero from the time balance (applies to accounts dimensions only).</td>
</tr>
<tr>
<td>Expense Reporting</td>
<td>E</td>
<td>Treat as an expense item (applies to accounts dimensions only)</td>
</tr>
<tr>
<td>Comment</td>
<td>Any string</td>
<td>Enter a comment.</td>
</tr>
<tr>
<td>Formula</td>
<td>Valid calculation syntax</td>
<td>Enter a member formula.</td>
</tr>
</tbody>
</table>
Table A-7  (Cont.) Fields and Valid Values in Dimension Worksheets

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Defined Attribute</td>
<td>Attribute names, such as specific colors or sizes</td>
<td>Defined attribute names used to aid in the analysis of the data.</td>
</tr>
<tr>
<td>Number of UDAs</td>
<td>A numeral</td>
<td>The number of UDAs for this member.</td>
</tr>
<tr>
<td>Available Alias Tables</td>
<td>Naming conventions for member names apply. See Naming Conventions for Dimensions, Members, and Aliases in Designing and Maintaining Essbase Cubes, ALIAS.table_name</td>
<td>After the column heading with ALIAS.table_name, the column is populated with the aliases for the cube.</td>
</tr>
</tbody>
</table>

You can modify dimension worksheets in the Cube Designer wizard. See Working with Dimension Worksheets in Cube Designer.

See Working with Rules Files in Designing and Maintaining Essbase Cubes.

Understanding Data Worksheets

Data Worksheets

You can include one or more data worksheets in an application workbook. The name of each data worksheet is Data.name. For example, for values for the eastern region, the data worksheet might be called Data.East. The name can be anything you choose. You can choose meaningful names so that you can recognize them if you need to use them again.

Note:

Multiple data worksheets are allowed in an application workbook, but they must share the exact same column layout.

Data Worksheet Format

When loading data, a member from every dimension must be defined before a data value. Therefore, the data worksheet places all but one dimension under the column headings titled, Dimension.dimension_name. One dimension is selected as the Measures dimension and members from that dimension must be added manually under the remaining column headings titled Measure.member_name. Only place members that will contain data in the columns titled Measure.member_name.

When scenarios are enabled, cubes have a hidden dimension called sandbox. The sandbox dimension, named Dimension.sandbox, is the first column in the data worksheet. It contains a member called base that you must define when loading data.

The following image shows a data worksheet in a sample application workbook.
The following table describes the settings on the data name worksheets in application workbooks.

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
<td>A valid string. See Name and Related Artifact Limits in <em>Designing and Maintaining Essbase Cubes</em>.</td>
<td>The build process creates a data file with a .txt extension in the cloud service for every data worksheet in the application workbook. You can give them meaningful names so that they are easily recognizable if they need to be used again.</td>
</tr>
<tr>
<td>Rule Name</td>
<td>A valid string. See Name and Related Artifact Limits in <em>Designing and Maintaining Essbase Cubes</em>.</td>
<td>The build process creates a rule file with a .rul extension in the cloud service for every dimension worksheet in the workbook. You can give them meaningful names so that they are easily recognizable if they need to be used again.</td>
</tr>
<tr>
<td>Data Load Option</td>
<td>• Add&lt;br&gt;• Subtract&lt;br&gt;• Replace</td>
<td>If you enter Replace, then the existing values of the database are overwritten with the values of the data source. You can also use incoming data values to add to or subtract from existing database values. For example, if you load weekly values, then you can add them to create monthly values in the database.</td>
</tr>
</tbody>
</table>
### Property or Field

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delimiter</td>
<td>The values can be a tab, a space, or any single character except &quot;.&quot;</td>
<td>This value must be updated directly in the Excel sheet. It cannot be updated using the Cube Designer interface.</td>
</tr>
<tr>
<td></td>
<td>• Tab</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Space</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Any single character except &quot;</td>
<td></td>
</tr>
<tr>
<td>Header Rows to Skip</td>
<td>A positive number or zero.</td>
<td>The number of header rows to skip when performing a data load or dimension build.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value must be updated directly in the Excel sheet. It cannot be updated using the Cube Designer interface.</td>
</tr>
<tr>
<td>Sign Flip Dimension</td>
<td>Dimension name</td>
<td>Reverses the values of data fields by flipping their signs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enter the name of the dimension in the Sign Flip Dimension field, and enter the selected UDA within the specified dimension in the Sign Flip UDA field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value must be updated directly in the Excel sheet. It cannot be updated using the Cube Designer interface.</td>
</tr>
<tr>
<td>Sign Flip UDA</td>
<td>• Flip</td>
<td>Reverses the values of data fields by flipping their signs.</td>
</tr>
<tr>
<td></td>
<td>• Blank</td>
<td>Enter the name of the dimension in the Sign Flip Dimension field, and enter the selected UDA within the specified dimension in the Sign Flip UDA field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value must be updated directly in the Excel sheet. It cannot be updated using the Cube Designer interface.</td>
</tr>
<tr>
<td>Ignore column header</td>
<td>Ignore</td>
<td>Data in a column with the heading, IGNORE is ignored during data loads and dimension builds.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value must be updated directly in the Excel sheet. It cannot be updated using the Cube Designer interface.</td>
</tr>
</tbody>
</table>

### Data Operations

When you load data, values can replace, add to, or subtract from existing data values in the cube. You indicate which of these options to use in the **Data Load Option** field on the data worksheet.
• **Replace**: Overwrites cube values with the data source values. Replace is the default.

• **Add**: Adds data source values to the cube values. For example, if you load weekly data values, you can add them to create cumulative data values in the cube.

• **Subtract**: Subtracts data source values from the database values. For example, to track available budget by week, you can subtract weekly data expenditures from the previous week's budget values.

**Rules Files**

When you build a cube, data files and data load rules files are created in the cloud service. Those files can then be used later if you want to load data to a cube. Data files are named with the file name specified in the definitions area of the data sheet and a .txt extension. For example, `cube_basic.txt`. Rules files are named with the file name specified in the definitions area of the data sheet and a .rul extension. For example, `cube_basic.rul`. You can also use data files and data load rules files from a supported on-premises release of Essbase.

You can modify data worksheets in the Cube Designer wizard. See **Working with Data Worksheets in Cube Designer**.

See Data Sources in *Designing and Maintaining Essbase Cubes*.

---

### Understanding Calculation Worksheets

You can have one or more calculation worksheets in an application workbook.

The following image shows a calculation worksheet in a sample application workbook.

![Calculation Worksheet Example](image)

Within the calculation worksheet, the calculation script begins in cell C6.

The name of each calculation worksheet is `Calc.scriptname`, for example, for the sample `CalcAll` calculation script, the calculation worksheet is called `Calc.calcall`.

The contents of the calculation worksheet are used to create a calculation script in the cloud service. The calculation script uses the file name specified in the definitions area of the calculation sheet and a has a .csc extension. For example, `filename.csc`.

You can execute the calculation script when you build the cube in Cube Designer, if you select **Run Calculation Sheets Contained within Workbook** in the Build Cube dialog box. If you do not want to execute the calculation, do not select this option.
Note:
The calculation scripts are executed in the order they appear in the application workbook.

<table>
<thead>
<tr>
<th>Property or Field</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
<td>See Naming Conventions in Calculation Scripts, Report Scripts, Formulas, Filters, and Substitution and Environment Variable Values in <em>Designing and Maintaining Essbase Cubes</em>.</td>
<td>The File Name defines the calculation script name. The calculation script created on the cloud service when the cube is created is the File Name with a .csc extension.</td>
</tr>
<tr>
<td>Execute Calc</td>
<td>• Yes</td>
<td>If you enter Yes, then the calculation is executed at the time you build the cube. If you enter No, then the calculation is not executed right away. In either case, each calculation worksheet creates a calculation script on the cloud service, using the specified file name with a .csc extension. That way, any of the calculations can be executed at a later time.</td>
</tr>
<tr>
<td></td>
<td>• No</td>
<td></td>
</tr>
</tbody>
</table>

You can modify calculation worksheets in the Cube Designer wizard. See *Working with Calculation Worksheets in Cube Designer*.

## About Using Tabular Data to Create Cubes

You can create a cube from tabular data by extracting fact tables from a relational database into an Excel file and then deploying the cube.

Patterns in the relationships between column headers and data are detected to deploy a multidimensional cube. The process for transforming tabular data into a structure that can be used in a multidimensional cube include these concepts:

- Correlations between columns
- Correlations between column types (such as date, number, and text)
- Header text analysis for common prefixes and business intelligence-related terms (such as cost, price, account)
- Report structure (such as merged cells and empty cells)
- (Optional) Forced-designation headers that are used to explicitly define the shape of a cube and can include formulas to create measures dimensions.
- Measures hierarchies (which can also be generated in Transform Data in Cube Designer).

Sample tabular data Excel files are provided to demonstrate the concepts of intrinsic and forced-designation headers.
When working with tabular data, you should analyze the data before you create a cube from it. Then, after the cube is created, you should determine if the cube outline is the way you want it.

You can create a cube from tabular data in the cloud service or in Cube Designer. See Creating a Cube from Tabular Data.

**Intrinsic Headers**

Intrinsic headers use table.column format, which is demonstrated in the Sample_Table.xlsx file. In this sample file, the column headers have names such as Units, Discounts, Time.Month, Regions.Region, and Product.Brand.

The transformation process creates this hierarchy:

- Units
- Discounts
- Fixed Costs
- Variable Costs
- Revenue
- Time
  - Month
  - Quarter
- Years
- Regions
  - Region
  - Area
  - Country
- Channel
- Product
  - Brand
  ...

**Forced-designation Headers**

With forced-designation headers, you can specify how tabular data should be handled during the transformation process. For example, you can force a column to be treated as a measures or an attributes dimension. Most forced-designation headers require a keyword in brackets [ ]. Forced-designation headers are demonstrated in the Modified_Sample_Table.xlsx file (available in the Cube Designer Gallery only).

Supported forced-designation header formats:

**Table A-8  Forced-designation Header Formats**

<table>
<thead>
<tr>
<th>Designation</th>
<th>Header Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>ReferenceGeneration.Generation[alias]</td>
<td>Year.ShortYearForm[alias]</td>
</tr>
<tr>
<td>Measures</td>
<td>MeasureName[measure]</td>
<td>Price[measure]</td>
</tr>
</tbody>
</table>
Table A-8  (Cont.) Forced-designation Header Formats

<table>
<thead>
<tr>
<th>Designation</th>
<th>Header Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-most parent, if unique, is the account dimension name. If not unique, this member is auto-generated in the account dimension.</td>
<td></td>
</tr>
<tr>
<td>Measures formula</td>
<td>MeasureName[=formula_syntax]</td>
<td>profit[=&quot;price&quot;-&quot;cost&quot;]; profit[=&quot;D1&quot;-&quot;E1&quot;]; price[=IF (&quot;S1&quot; == #MISSING) &quot;R1&quot;; ELSE &quot;S1&quot;; ENDIF:]</td>
</tr>
<tr>
<td>Measures consolidation</td>
<td>MeasureName[+] : add to parent</td>
<td>price.shipment[+]</td>
</tr>
<tr>
<td></td>
<td>MeasureName[-] : subtract from parent</td>
<td>Consolidation can be defined only for measure dim</td>
</tr>
<tr>
<td></td>
<td>MeasureName[-] : no consolidation (equivalent to [measure])</td>
<td>The default is no consolidation.</td>
</tr>
<tr>
<td>Formula consolidation</td>
<td>FormulaName[+=&lt;formula&gt;] : add to parent</td>
<td>profit[+=price-cost] cost.external[+=ExternalWork +ExternalParts]</td>
</tr>
<tr>
<td></td>
<td>FormulaName[=&lt;formula&gt;] : subtract from parent</td>
<td></td>
</tr>
<tr>
<td>UDA</td>
<td>ReferenceGeneration[uda]</td>
<td>Product[uda]</td>
</tr>
<tr>
<td>Skip</td>
<td>ColumnName[skip]</td>
<td>column[skip]</td>
</tr>
<tr>
<td>Recur</td>
<td>ColumnName[recur]</td>
<td>Product[recur]</td>
</tr>
<tr>
<td></td>
<td>The last column cell value is used for empty cells</td>
<td>The last column cell value is used for empty cells</td>
</tr>
<tr>
<td></td>
<td>Recur can be combined with other forced designations; include a comma separated list of forced designations within a bracket, ColumnName[designationA,recur].</td>
<td>Recur can be combined with other forced designations; include a comma separated list of forced designations within a bracket, ColumnName[designationA,recur].</td>
</tr>
</tbody>
</table>

You can specify columns to be measures dimensions and you can use formulas to create measures dimensions with calculated data during the transformation process. The measures and measures formula forced-designation headers are specified with the name for the measures dimension, followed by a keyword or formula that is enclosed in square brackets and appended to the measures dimension name.

You can also consolidate measures and formulas by adding them to, or subtracting from, the parent.

To specify a column to be a measures dimension, in the column header, you enter the name of the measures dimension and then append the keyword [measure]. For
example, you can specify the Units and Fixed Costs columns as measures dimensions by using this syntax: Units[measure] and Fixed Costs[measure].

The transformation process creates this hierarchy, with Units, Discounts, Fixed Costs, Variable Costs, and Revenue as measures:

Time
  Year
    Quarter
    Month
Regions
  Region
    Area
    Country
...
Product
  Brand
...
Units
Discounts
Fixed Costs
Variable Costs
Revenue

You can create a measure generation hierarchy (parent.child[measure] hierarchy), in a similar way that you create regular dimension generations.

For example, to create a measure hierarchy, you enter Measures.profit[measure], profit.cost[measure] and cost.price[measure], which produces the following hierarchy:

Measures
  profit
    cost
      price

To create measures dimensions from formulas, in the column header, you enter the name of the measures dimension and then append the formula syntax in brackets [ ]. Within the brackets, start the formula with an equal sign (=) and end the formula with a semicolon (;). The arguments in the formula correspond to column names or cell coordinates, which must be enclosed in quotes. You can use Essbase calculation functions and commands in the formula.

Assume that you have an Excel file named Spend_Formulas.xlsx with tabular data on the SpendHistory worksheet, which has many columns. For example, there are dimensions named Year (column A) and Quarter (column B), and measures dimensions named Spend (column J) and Addressable Spend (column K). These columns have data. Then there are column headers that use formulas to create a measures dimensions. These columns do not have data. For example, to create the Total Spend dimension, the header in column O uses this Essbase formula: Measure.Total Spend[="Addressable Spend" + "Non-Addressable Spend"]. To create the AddSpendPercent dimension, the header in column P uses this Essbase formula: Measure.AddSpendPercent[="Addressable Spend"/"Total Spend"].

The transformation process creates this hierarchy:
The transformation process can also identify measures dimensions when a dimension name is duplicated. Assume that you have a column header that uses this formula, Meas.profit[="a1"-"b1"], which creates the Meas dimension. If, in another column header, you use the Meas dimension name as the top parent, such as Meas.Sales, the Sales dimension is also considered a measures dimension.
Setting up Cube Designer

You might find it easier to work with application workbooks in Excel using the Cube Designer extension for Smart View.

• Workflow for Setting up Cube Designer
• Downloading and Running the Smart View Installer
• Creating Data Source Connections to the Cloud Service
• Installing the Smart View Cube Designer Extension
• Updating the Smart View Cube Designer Extension
• Deleting Smart View Connection URLs

Workflow for Setting up Cube Designer

This is the workflow for setting the Smart View Cube Designer extension:

1. Install Smart View.
2. Set up a data source connection to the cloud service.
3. Install Cube Designer Smart View extension.
4. Update Cube Designer Smart View extension.

Downloading and Running the Smart View Installer

Smart View Prerequisites

• The latest release of Smart View
  On the Oracle Technology Network Downloads tab, the latest release for Smart View is always certified.
• Microsoft Office 2010, 2013 or 2016
• .NET Framework 4.0

Note:

You must use .NET Framework 4.5 if you are installing Smart View from the cloud service without saving the installer locally.

Installing Smart View

1. Log into Oracle Analytics Cloud - Essbase.

2. On the Applications home page, click Utilities.
3. On the Utilities page, click **Smart View for Essbase**.

![Utilities](image)

4. On the Smart View download page on Oracle Technology Network, click **Accept License Agreement**, and then click **Download Now**.
   
   If the Oracle sign-in page is displayed, then sign in with your Oracle user name (usually your email address) and password.

5. Follow the steps for your browser to download the .zip file, and save it to a folder on your computer.

6. Go to the folder that you used in Step 5, and then double click **smartview.exe** to start the installation wizard.

7. Select a destination folder for Smart View, and then click **OK**. For new installations, Smart View is installed by default in: \C:\Oracle\smartview.
   
   If you are upgrading an installation of Smart View, then the installer defaults to the folder where you previously installed Smart View.

8. When the installation is complete, click **OK**.

   Continue the setup process with **Creating Data Source Connections to the Cloud Service**.

**Creating Data Source Connections to the Cloud Service**

After you install Smart View, you can create connections to Oracle Analytics Cloud - Essbase.

Connections require information about the server and port. Your cloud service administrator should provide you with the information you need to create the connection.

For the private connection to the cloud service, use the quick connection method.

1. In Excel, select the Smart View ribbon, and then click **Panel**.

2. On the **Smart View Panel**, click the arrow next to **Switch to**, and then select **Private Connections** from the list.

3. Still in the **Smart View Panel**, in the text box, enter the URL for the data source to which you want to connect.

   **URL syntax**: `https://server/essbase/smartview`

4. Click **Go** or press **Enter**.

5. On the login window, enter your login credentials, and then select a data source from the drop-down menu.
Continue the setup process with Installing the Smart View Cube Designer Extension.

Installing the Smart View Cube Designer Extension

Before you perform this procedure, you must complete the steps in Creating Data Source Connections to the Cloud Service.

Installing Cube Designer from Smart View
1. On the Smart View ribbon, select Options, and then Extensions.
2. Click the Check for updates link.
   Smart View checks for all extensions that your administrator has made available to you.
3. Locate the extension named Oracle Cube Designer and click Install to start the installer.
4. Follow the prompts to install the extension.

Installing Cube Designer from the Service
1. In the service, from the Application home page, select the application and then click Utilities.
2. In Utilities, find Cube Designer Extension and click Download.

3. In the Opening CubeDesignerInstaller.svext dialog box, select Save File and click OK.
   Save the file to a local directory.
4. Close all Microsoft Office applications and make sure Microsoft Office applications are not running in the background.
5. Double click the CubeDesignerInstaller.svext file.
6. Restart Microsoft Office applications.

Updating the Smart View Cube Designer Extension

If an extension is available for you to update, you can update it from Smart View Excel, on the Extensions tab of the Options dialog box.
To check for Cube Designer Smart View extension updates and install them:

1. From the Smart View ribbon, select **Options** and then **Extensions**.

2. Click the **Check for Updates, New Installs, and Uninstalls** link to check for updates.

   You are prompted to log in.

   If an update is available, the **Update Available** icon is displayed in the **Cube Designer** row.

   **Note:**

   This process uses a server locations list, which was created by previous Smart View connections. If there are connection definitions that are no longer valid, you receive errors when the process tries to connect to those servers. See Deleting Smart View Connection URLs.

3. Click **Remove** to uninstall the extension.


5. Restart Excel.

6. From the Smart View ribbon, select **Options** and then **Extensions**.

7. Click **Check for Updates, New Installs, and Uninstalls**.

   You are prompted to log in.

8. In the Cube Designer row, click **Install**.


10. Open Excel.

11. Ensure that the Cube Designer ribbon is displayed in Excel.

### Deleting Smart View Connection URLs

When you connect to the cloud service from Cube Designer, the list of server locations that are used to connecting is created by previous Smart View connections. If there are connection definitions that are no longer valid, you receive errors.

You can reset the list of connection definitions to remove those that you are unwanted, or are invalid.

To reset the list of server locations:

1. Click the down arrow next to the **Private Connection** drop down list and select **Delete Connection URLs**.
2. In the Delete Connection URLs dialog box, select **Extension Update URLs** from the drop down menu.

3. Select all of the URLs except the one you want to use, and click **Delete**.
The default calculation and query processor lets you perform real-time analytics using procedural calculations and read and write modeling capabilities.

If you have worked with on-premises Essbase, then you likely remember one or more of these cube design variants, tailored for specific purposes:

- Block storage, with large, sparse dimensions, stored and pre-aggregated to achieve good query performance, and a rich set of calculation functions for analysis.
- Aggregate storage, for cubes having a large number of dimensions, and many upper-level aggregations.
- Hybrid aggregation mode, which is block storage enhanced with the benefits of aggregate storage.

The default calculation and query processor is the hybrid aggregation mode. The ASODYNAMICAGGINBSO configuration setting controls whether block storage databases use hybrid aggregation mode.

Hybrid aggregation mode is enabled by the ASODYNAMICAGGINBSO FULL default configuration setting.

Most calculation functions are supported in hybrid aggregation mode. To see a list and syntax for all supported calculation functions, as well as the few exceptions, see the Technical Reference for Oracle Analytics Cloud - Essbase.

See Using Hybrid Aggregation in Designing and Maintaining Essbase Cubes for a more in-depth description of Hybrid aggregation mode.

See ASODYNAMICAGGINBSO in Technical Reference for Oracle Analytics Cloud - Essbase for the syntax to configure Hybrid aggregation mode beyond the default settings.