# Contents

## Documentation Accessibility

## Documentation Feedback

### 1 About Backup and Recovery

- **Assumed Knowledge**
  - 1-1
- **Repositories, Databases, and File Systems**
  - 1-1
- **Recovery Sequence**
  - 1-3

### 2 Common Backup Tasks

- **Preparing for Backup**
  - 2-1
- **Database Backup**
  - 2-1
- **File System Backup**
  - 2-2
- **Backing Up OS Settings**
  - 2-3

### 3 Foundation Services

- **EPM Workspace and Shared Services**
  - 3-1
- **Calculation Manager**
  - 3-3
- **Smart View**
  - 3-3

### 4 Essbase Components

- **Essbase**
  - 4-1
- **Essbase Studio**
  - 4-26
- **Provider Services**
  - 4-26

### 5 Financial Performance Management Applications

- **Planning**
  - 5-1
Financial Management 5-2
Tax Provision 5-3
Financial Close Management 5-3
Tax Governance 5-3
Profitability and Cost Management 5-3

6 Financial Reporting

Financial Reporting 6-1

7 Data Management

Data Relationship Management 7-1
FDMEE 7-1
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.
Documentation Feedback

To provide feedback on this documentation, send email to epmdoc_ww@oracle.com, or, in an Oracle Help Center topic, click the Feedback button located beneath the Table of Contents (you may need to scroll down to see the button).

Follow EPM Information Development on these social media sites:
LinkedIn - http://www.linkedin.com/groups?gid=3127051&goback=gmp_3127051
Twitter - http://twitter.com/hyperionepminfo
Facebook - http://www.facebook.com/pages/Hyperion-EPM-Info/102682103112642
YouTube - https://www.youtube.com/oracleepminthecloud
1

About Backup and Recovery

Related Topics

• Assumed Knowledge
• Repositories, Databases, and File Systems
• Recovery Sequence

Assumed Knowledge

This guide is for administrators who install, configure, and manage Oracle Enterprise Performance Management System products. It assumes that you have the following skills and knowledge:

• Security and server administration skills
• Administration skills for your operating system (OS)
• Web application server administration skills
• A strong understanding of your organization's security infrastructure, including authentication providers such as Oracle Internet Directory, Lightweight Directory Access Protocol (LDAP), Microsoft Active Directory, and use of Secure Sockets Layer (SSL)
• Strong relational database management system (RDBMS) administration skills
• A strong understanding of your organization's database and server environments, including file systems
• A strong understanding of your organization's network environment and port usage

Repositories, Databases, and File Systems

Many Oracle Enterprise Performance Management System products use repositories, which contain items that the products require. Repository contents differ by product. Some product repositories use RDBMS, some use file systems, and some use both RDBMS and file systems.

Database Backup Types

You can use several types of database backup, depending on your computing environment.

See the Oracle Database Backup and Recovery User's Guide for more information on backup and recovery for Oracle databases.

Physical Backup

Physical backups are copies of physical database files. For example, a physical backup might copy database content from a local disk drive to another secure location.
A physical backup can be hot or cold:

- **Hot backup**—Users can modify the database during a hot backup. Log files of changes made during the backup are saved, and the logged changes are applied to synchronize the database and the backup copy. A hot backup is used when a full backup is needed and the service level does not allow system downtime for a cold backup.

- **Cold backup**—Users cannot modify the database during a cold backup, so the database and the backup copy are always synchronized. Cold backup is used only when the service level allows for the required system downtime.

You can perform a full or incremental physical backup:

- **Full**—Creates a copy of data that can include parts of a database such as the control file, transaction files (redo logs), archive files, and data files. This backup type protects data from application error and safeguards against loss by providing a way to restore original data. Perform this backup weekly, or biweekly, depending on how often your data changes. Making full backups **cold**, so that users cannot make changes during the backups, is recommended.

  **Note:**
  
  Regular cold full physical backups are recommended.

- **Incremental**—Captures only changes made after the last full physical backup. The files differ for databases, but the principle is that only transaction log files created since the last backup are archived. Incremental backup can be done **hot**, while the database is in use, but it slows database performance.

  **Note:**
  
  The database must be in archive log mode for a full physical backup.

In addition to backups, consider using clustering or log shipping to secure database content. See the *Oracle Enterprise Performance Management System Installation and Configuration Guide* and the RDBMS documentation.

**Logical Backup**

A logical backup copies data, but not physical files, from one location to another. A logical backup is used to move or archive a database, tables, or schemas and to verify database structures.

A full logical backup enables you to copy these items across environments that use different components, such as operating systems:

- **Entire applications**
- **Data repositories** such as the Oracle Hyperion Shared Services Registry and Oracle Essbase cubes
- **Individual artifacts** such as scripts, data forms, and rule files
A logical export backup generates necessary Structured Query Language (SQL) statements to obtain all table data that is written to a binary file. A logical export backup does not contain database instance-related information, such as the physical disk location, so you can restore the same data on another database machine. Periodic logical export backups (at least weekly) are recommended in case physical backups fail or the database machine becomes unavailable.

**Backup with Lifecycle Management**

You can use Oracle Hyperion Enterprise Performance Management System Lifecycle Management, which is provided with Oracle Hyperion Foundation Services, to perform logical backups. See the *Oracle Enterprise Performance Management System Lifecycle Management Guide*.

**File System Backup Types**

A complete file system backup includes an entire system directory. For example, backing up the EPM Oracle home directory backs up all installed EPM System products. You can also perform file-system backups of these types and frequencies:

- **Post-installation**—Directories created or modified if you reconfigure products
- **Daily incremental**—New directories or files or those modified since the previous day (including repository content and log files)
- **Weekly full**—All files in the directories for which you perform daily incremental backups
- **As needed**—Data that is modified infrequently

**Recovery Sequence**

Restore Oracle Hyperion Shared Services, and the components that you backed up for Shared Services, before restoring other products.

⚠️ **Caution:**

It is imperative that backup and restore operations for Oracle Enterprise Performance Management System components be synchronized, because EPM System components continually read and write information to the Shared Services repository. When restoring Shared Services from a backup, for example, you must also restore registered EPM System components from backups that were made at the same time.
Common Backup Tasks

Related Topics
- Preparing for Backup
- Database Backup
- File System Backup
- Backing Up OS Settings

Preparing for Backup

Complete these tasks before starting a cold backup:
- Stop products and ensure that all users have logged off.
- Stop all related services. See "Starting and Stopping EPM System Products" in the Oracle Enterprise Performance Management System Installation and Configuration Guide.
- Back up the Oracle Hyperion Foundation Services database.

Note:
You can also perform hot backups on Oracle Enterprise Performance Management System products, with some limitations for individual products as described later in this document. These backups must be performed at the same time.

Database Backup

Back up these databases:
- Databases that store Oracle Enterprise Performance Management System application data
- The Oracle Hyperion Shared Services repository
  The Shared Services repository contains the Oracle Hyperion Shared Services Registry, which stores most product configuration settings. It also contains Native Directory, provisioning information, and Oracle Hyperion Enterprise Performance Management Workspace preferences. Oracle recommends a physical full backup immediately after installation and configuration.

For instructions on backing up an individual EPM System component, see the section of this guide that is specific to the component. See EPM Workspace and Shared Services for instructions on backing up Oracle Hyperion Foundation Services.
Also back up any database, using the vendor documentation, that you use to store or extract data for use in product applications. Regular backups of database content are recommended for these EPM System components:

- Foundation Services
- Oracle Hyperion Calculation Manager
- Oracle Data Relationship Management
- Oracle Hyperion Financial Management
- Oracle Hyperion Financial Close Management
- Oracle Hyperion Planning
- Oracle Hyperion Profitability and Cost Management
- Oracle Hyperion Financial Data Quality Management, Enterprise Edition

These products do not use repository databases:

- Oracle Smart View for Office
- Oracle Hyperion Provider Services

**File System Backup**

Regular file system backups are recommended for these Oracle Enterprise Performance Management System products:

- Oracle Hyperion Foundation Services
- Oracle Data Relationship Management
- Oracle Hyperion Financial Management
- Oracle Hyperion Planning
- Oracle Hyperion Profitability and Cost Management

Oracle recommends daily backup of these items:

- `EPM_ORACLE_INSTANCE/config` (to back up the configuration and reconfiguration settings written to the Oracle Hyperion Shared Services Registry)
- `MIDDLEWARE_HOME/user_projects/domains/domain name` (EPM System WebLogic domain directory)

**Note:**

This item applies only to products that require a Web application server.

- `EPM_ORACLE_INSTANCE/import_export` (where Oracle Hyperion Enterprise Performance Management System Lifecycle Management content is located)
- Product applications and application data
- In Windows environments:
  - Windows registry: `HKEY_LOCAL_MACHINE` and all of its subkeys
Note:
For some EPM System components, you need only back up specific subkeys, as described in the procedures for backing up those components.

Backing up the Windows registry enables system recovery if Windows is reinstalled. See Backing Up OS Settings.

Not applicable for these products:
* Oracle Smart View for Office
* Oracle Hyperion Provider Services

See Backing Up OS Settings.

- %CommonProgramFiles%\InstallShield\Universal
- %USERPROFILE%\oracle.instance, which enables you to add, remove, reinstall, and upgrade products

Back up the EPM Oracle home directory (to back up all installed products) if you install a new product or apply a patch.

Back up directories and files by copying them to another location. You can also use utilities provided with your OS (such as the Windows 2003 Backup Utility) or other third-party backup utilities. In case of failure, restore these directories and files by returning the copy to the original location.

Backing Up OS Settings

Windows registry settings and specific system variables in Windows environments should be backed up.

Backing Up Windows Registry Settings

Backing up the Windows registry (HKEY_LOCAL_MACHINE and its subkeys) enables system recovery if Windows is reinstalled.

Note:
For some Oracle Enterprise Performance Management System components, you need only back up specific subkeys, as described in the procedures for backing up those components.

You can use the Windows regedit command to create registry files to back up system and product components. If a failure occurs, you can run the registry files to restore the components.
Note:
The following procedure does not apply to Oracle Smart View for Office, or Oracle Hyperion Provider Services.

To back up a component with regedit:

1. Select Start and then Run.
2. Enter regedit, and click OK.
3. Right-click the subkey for the component in the left panel, and select Export.
4. Select a location for saving the registry file, enter a file name with the .reg extension, and click Save.

Example 2-1 System Variable Backup

Oracle recommends backing up the HYPERION_HOME and EPM_ORACLE_HOME system variables.

Note:
This recommendation does not apply to Smart View.
Foundation Services

Related Topics
- EPM Workspace and Shared Services
- Calculation Manager
- Smart View

EPM Workspace and Shared Services

Oracle Hyperion Enterprise Performance Management Workspace and Oracle Hyperion Shared Services share a file system and database.

Note:
You must synchronize Shared Services backups with product backups to preserve provisioning data.

Backing Up the File System

To back up the file system for EPM Workspace and Shared Services:

1. Copy these folders to a storage device or another network location after you install or reconfigure EPM Workspace.
   - \EPM_ORACLE_INSTANCE\
   - MIDDLEWARE_HOME/user_projects/domains/d

   Note:
   This is a one-time backup of all Oracle Enterprise Performance Management System components that installed and configured on the machine.

2. Perform a weekly full or daily incremental backup of these subfolders of \EPM_ORACLE_INSTANCE/config:
   - FoundationServices
   - Foundation
Note:

The 11.1.2.0 subfolder of EPM ORACLE_INSTANCE/config/foundation contains .reg.properties, which is required for recovery.

3. Optional: Perform a monthly or weekly backup of MIDDLEWARE_HOME/user_projects/domains/domain name/servers/managed server name/logs, which contains only historical information.

Restoring EPM Workspace and Shared Services

To restore EPM Workspace and Shared Services after a failure:

1. Recover all components that you backed up, including Oracle Hyperion Shared Services Registry.
   Place the copied directories and files in their original locations.

2. Restart all products and related services. See "Starting and Stopping EPM System Products" in the Oracle Enterprise Performance Management System Installation and Configuration Guide.

Backing Up the Database

To back up the database for EPM Workspace and Shared Services:

1. Navigate to your database system tables.

2. Back up the Shared Services and EPM Workspace schema if you use Oracle RDBMS; otherwise, back up the SQL Server database.

Restoring the Database to a Different Server

If you back up a database from one server and restore it to a second server (for example, because the first database server is under maintenance), you must use the WebLogic Admin Console to update the data source and the reg.properties file:

To restore a database to a different server:

1. Update the data source:
   a. From the WebLogic Admin Console, select Services, then dataSource, and then EPMSystemRegistry.
   b. On the Connection Pool tab, specify the user name, password, and JDBC URL for the second server.
   c. Click Save and Activate Changes.

2. Open reg.properties with a text editor and update the user name, password, and JDBC server URL.

   The reg.properties file is in EPM ORACLE INSTANCE/config/foundation/11.1.2.0.

Calculation Manager

To enable recovery after a failure, you need only back up the database that you use with Oracle Hyperion Calculation Manager, as described in the RDBMS documentation. Additional precautions that you can take:

• Back up the product components in EPM_ORACLE_HOME/products/Foundation/CALC.
• Perform a weekly full or daily incremental backup of MIDDLEWARE_HOME/user_projects/domains/domain name/servers/managed server name/logs.

Smart View

To enable recovery of Oracle Smart View for Office after a failure, copy the Microsoft Office documents containing Smart View data to another location.

Include these file types:

• XLS and XLSX (Excel)
• DOC and DOCX (Word)
• PPT and PPTX (PowerPoint)

To recover from a failure, replace the backed up Microsoft Office documents in their original locations.
Essbase Components

Related Topics

• Essbase
• Essbase Studio
• Provider Services

Essbase

Regular Oracle Essbase backups, which should be integrated into production server maintenance, are key to database maintenance. The volatility of the database and server environment and the need for rapid database restoration if a server interruption occurs should determine backup frequency.

Restore refers to the process of returning a database to the state it was in when a backup was performed. For block storage databases, you can use the automated Essbase backup and restore feature or using manual procedures. For aggregate storage databases, you must use manual backup and restore features.

Recover refers to the process of returning a database to a post-backup state that includes transactions that occurred after the backup was performed. To recover a database, you must use the automated backup and restore and the automated transaction logging and replay features.

For more information about MaxL statements and Administration Services Console options discussed in this section, see the Oracle Essbase Technical Reference and Oracle Essbase Administration Services Online Help, respectively.

Backing Up and Recovering Block Storage Databases

To back up and restore block storage databases, use either method:

• Automated database backup and restore and transaction logging and replay

Backup and restore provides the equivalent functionality of manually backing up and restoring a database. When a backed-up database is restored, transactions that occurred after the backup procedure are not recovered. However, with transaction logging and replay, post-backup transactions are captured and can be replayed. Thus, a backed-up database can be recovered to the most-recent state before the interruption occurred.

Using the database backup and restore and transaction logging and replay features eliminates the need for various manual steps and, therefore, enables administrators to back up and recover databases more efficiently. Oracle recommends incorporating these features in your backup and recovery strategy.

See:

– Sequence for Fully Recovering Block Storage Databases
– Using Backup and Restore for Block Storage Databases
Using Transaction Logging and Replay For Block Storage Databases

- Manual backup and restore

Essbase customers who have designed a backup and restore strategy that uses manual procedures and who do not need the functionality of transaction logging and replay can continue using their manual strategy.


Sequence for Fully Recovering Block Storage Databases

When you restore a database from a backup, the state of the restored database is exactly as it was when the backup was performed. Any transactions that took place after the backup are not recovered. Transaction logging captures ongoing transactions. These transactions can be replayed to recover the database to its most recent state.

A typical sequence for using these two features is to enable transaction logging and then back up a database. After restoring the backed-up database, you can replay the logged transactions that took place after the backup operation.

Transactions that reset the outline and data together (for example, clearing all loaded data and resetting the outline to empty) can be logged and replayed. See Table 5 for a list of transactions that can be logged and replayed.

Outline changes are not logged and, therefore, cannot be replayed. The outline itself, however, is saved as part of a backup. If you change the outline after backing up a database, after restoring the backed-up database, you can replay the logged transactions that took place after the backup operation.

Table 1 is an example of how to use these features in tandem.

Table 4-1 Example Sequence for Using Backup and Restore and Transaction Logging and Replay

<table>
<thead>
<tr>
<th>Time Period</th>
<th>(Sequence ID) Transaction or Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>T₀</td>
<td>Enable transaction logging.</td>
</tr>
<tr>
<td>T₁</td>
<td>(1) Perform a data load.</td>
</tr>
<tr>
<td>T₂</td>
<td>(2) Calculate the database.</td>
</tr>
<tr>
<td>T₃</td>
<td>(3) Back up the database.</td>
</tr>
<tr>
<td>T₄</td>
<td>(4) Perform another data load.</td>
</tr>
<tr>
<td>T₅</td>
<td>(5) Update cells in a spreadsheet, and perform a lock and send to update the database.</td>
</tr>
<tr>
<td>T₆</td>
<td>(6) Calculate the database.</td>
</tr>
<tr>
<td>T₇</td>
<td>System failure results in a corrupted database.</td>
</tr>
<tr>
<td>T₈</td>
<td>Restore the backup of the database taken at T₃. The restored database includes the results of transactions with sequence IDs of 1 and 2 but not those of 4, 5, and 6.</td>
</tr>
</tbody>
</table>
Table 4-1  (Cont.) Example Sequence for Using Backup and Restore and Transaction Logging and Replay

<table>
<thead>
<tr>
<th>Time Period</th>
<th>(Sequence ID) Transaction or Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_9$</td>
<td>Replay transactions with sequence IDs of 4, 5, and 6. These transactions occurred after the backup was performed at $T_3$ and before the database was corrupted at $T_7$, and were not recovered when the database was restored at $T_8$.</td>
</tr>
<tr>
<td>$T_{10}$</td>
<td>(7) Back up the recovered database.</td>
</tr>
<tr>
<td>$T_{11}$</td>
<td>Change the database outline.</td>
</tr>
<tr>
<td>$T_{12}$</td>
<td>(8) Back up the database outline or manually save the changed outline.</td>
</tr>
</tbody>
</table>

Using Backup and Restore for Block Storage Databases

The backup and restore feature applies to block storage databases.

You must have the Administrator role to back up and restore a database.

In backing up a database, Essbase performs the following tasks:

1. Places the database in read-only and exclusive-lock mode, which protects the database from updates during the archive process.
2. Writes a copy of the database files listed in Table 2 to an archive file that resides on the Essbase Server computer.

   See Configuring and Specifying the Database Archive File.
3. Returns the database to read-write mode.

Table 2 lists and describes the Essbase database files that are backed up automatically.

Table 4-2  Essbase Database Files That Are Automatically Backed Up

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>essxxxxx.pag</td>
<td>Essbase data files</td>
</tr>
<tr>
<td></td>
<td>For example, ess00001.pag</td>
</tr>
<tr>
<td>essxxxxx.ind</td>
<td>Essbase index files</td>
</tr>
<tr>
<td></td>
<td>For example, ess00001.ind</td>
</tr>
<tr>
<td>dbname.esm</td>
<td>Essbase Kernel file that contains control and disk volume information used for database recovery</td>
</tr>
<tr>
<td>dbname.tct</td>
<td>Transaction control table</td>
</tr>
<tr>
<td>dbname.ind</td>
<td>Free fragment file for data and index-free fragments</td>
</tr>
<tr>
<td>dbname.otl</td>
<td>Outline file, which stores all metadata for a database and defines how data is stored, but does not store data itself</td>
</tr>
</tbody>
</table>
### Table 4-2 (Cont.) Essbase Database Files That Are Automatically Backed Up

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbname.otl.keep</td>
<td>Temporary backup of dbname.otl (created by operations that modify the outline and write it to a new file)</td>
</tr>
<tr>
<td>dbname.otn</td>
<td>Temporary outline file created during a dimension build or outline editing</td>
</tr>
<tr>
<td>dbname.db</td>
<td>Database file containing database settings</td>
</tr>
<tr>
<td>dbname.ddb</td>
<td>Partition definition file</td>
</tr>
<tr>
<td>dbname.ocl</td>
<td>Outline change log created during incremental dimension build</td>
</tr>
<tr>
<td>essxxxx.chg</td>
<td>Outline synchronization change log</td>
</tr>
<tr>
<td>dbname.alg</td>
<td>Spreadsheet update log that stores spreadsheet update transactions as a unit that can be used as the input source for data loads</td>
</tr>
<tr>
<td>dbname.atx</td>
<td>Spreadsheet update log that contains historical information for each transaction, such as user name, date, and timestamp, and the number of transaction rows from the .atx file</td>
</tr>
</tbody>
</table>

In addition to the files that Essbase backs up, you must manually back up the files listed in Table 3. You should back up these files at a later date than when the database backup is performed (for example, a few days after). You should also back up rules files and ESSCMD or MaxL scripts when you create or modify them.

Table 3 lists and describes the Essbase database files that must be backed up manually.

### Table 4-3 Essbase Files That You Must Manually Back Up

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>essbase.sec</td>
<td>Essbase security file</td>
<td>ARBORPATH/bin/</td>
</tr>
<tr>
<td>essbase_timestamp.bak</td>
<td>Backup of the Essbase security file</td>
<td>ARBORPATH/bin/</td>
</tr>
<tr>
<td>essbase.cfg</td>
<td>Essbase Server configuration file</td>
<td>ARBORPATH/bin/</td>
</tr>
<tr>
<td>dbname.app</td>
<td>Application file containing application settings</td>
<td>ARBORPATH/app/</td>
</tr>
</tbody>
</table>
Table 4-3  (Cont.) Essbase Files That You Must Manually Back Up

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>.otl</td>
<td>Database artifact files</td>
<td>ARBORPATH/app/</td>
</tr>
<tr>
<td>.csc</td>
<td></td>
<td>appname/dbname/</td>
</tr>
<tr>
<td>.rul</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.rep</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.eqd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.sel</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ESSCMD or MaxL scripts</td>
<td>No defined storage location</td>
</tr>
</tbody>
</table>

1 Shut down the Agent before backing up essbase.sec.

Configuring and Specifying the Database Archive File

In backing up a database, Essbase copies the database files to an archive file, which must reside on the Essbase Server computer. When specifying the archive file name and location, follow these guidelines:

- Provide the full path to a directory on Essbase Server and the name of the archive file.
  
  If only the archive file name is provided, Essbase writes the archive file to ARBORPATH/app.

- Oracle recommends writing the archive file to a disk other than the one where ARBORPATH is located.

- Oracle recommends naming the archive file with an .arc extension.

- The archive file size corresponds to the size of the database you back up and is limited only by disk space, unless you split the archive file into multiple files. (See Splitting the Archive File Into Multiple Files.) With the single-file configuration, Oracle recommends saving archive files to a file system that supports large files. For Windows, the file system must be formatted as NTFS.

- When backing up a database to an archive file, use force option to overwrite the archive file; otherwise, the operation fails.

⚠️ Caution:

Before using the force option, ensure that you no longer need the current archive file contents.

Splitting the Archive File Into Multiple Files

By default, Essbase creates one large archive file. If you cannot use large files in your environment, or if your file-transfer tools cannot handle large files, you can configure
Essbase to split the archive into multiple files of 2 GB or less: In the `essbase.cfg` file, set the `SPLITARCHIVEFILE` configuration to `TRUE`.

The first (or main) archive file that Essbase creates uses the file name that you specify (for example, `samplebasic.arc`). When the main archive file reaches the 2 GB limit, Essbase creates another file. In naming each successive archive file, Essbase increments the main archive file name with "_n", where n is an integer (starting with 1). For example, three archive files created to back up the Sample.Basic database would be named as follows:

`samplebasic.arc samplebasic_1.arc samplebasic_2.arc`

All split archive files are created in the directory that you specified when specifying the file name and location of the main archive file.

**Note:**

When restoring a database in which the archive is split into multiple files, Essbase looks for multiple archive files even if, after the backup, you set `SPLITARCHIVEFILE` to `FALSE` for that database. Also, Essbase expects all archive files for a database to be in the same directory.

### Backing Up Block Storage Databases

To back up a database, you must have the Administrator role.

To back up a database, use one of these tools:

- **Oracle Essbase Administration Services**
  
  See “Backing Up Block Storage Databases” in *Oracle Essbase Administration Services Online Help*.

  In Administration Services Console, use the **Archive Database** command.

- **MaxL**

  See "Alter Database" in the *Oracle Essbase Technical Reference*.

  In MaxL, use the `alter database` statement with the `archive to file` grammar. The syntax for the statement:

  ```sql
  alter database appname.dbname [force] archive to file BACKUP-FILE;
  ```

  For example, the following statement backs up the Sample.Basic database to the specified archive file on Essbase Server:

  ```sql
  alter database Sample.Basic archive to file '/Oracle/samplebasic.arc';
  ```

  If the `samplebasic.arc` file exists on Essbase Server, use `force` grammar to overwrite the file contents. Example:

  ```sql
  alter database Sample.Basic force archive to file '/Oracle/samplebasic.arc';
  ```
At a different time from when you back up the database, manually back up the files listed in Table 3.

 Retrieving Archive File Information

Essbase can retrieve the following information about the backed-up database from the archive file:

- Overview information:
  - Application name
  - Database name
  - Archive time
- Disk volume names

To retrieve archive file information, use the `query archive_file` MaxL statement.

The syntax for the statement:

```
query archive_file BACKUP-FILE [get overview] [list disk volume];
```

In the MaxL Script Editor in Administration Services Console, you can view the results of the query. Figure 1 shows overview information:

**Figure 4-1  Archive File Overview Information**

<table>
<thead>
<tr>
<th>Application</th>
<th>Database</th>
<th>Backup Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample</td>
<td>Basic</td>
<td>Thu Apr 03 10:54:21 2008</td>
</tr>
</tbody>
</table>

Figure 2 shows disk volume names:

**Figure 4-2  Archive File List of Disk Volumes**

You can also view disk volume information in Administration Services Console when using the Restore Database command.

Note:

On Windows, Essbase adds the default ARBORPATH drive (for example, the C: drive) as a disk volume, even if the database that you backed up does not store data on that disk volume. See Working with Disk Volumes.
Restoring Block Storage Databases

You must have the Administrator role to restore a database.

Before restoring the database, you should terminate active client connections to the database.

In restoring a database, Essbase empties the database and locks it. Before extracting the contents of the archive file, Essbase validates the file. Essbase then copies index and page files to the specified disk volumes and updates dbname.esm with disk volume information. See Working with Disk Volumes.

If you have configured Essbase to split the archive file into multiple files (see Configuring and Specifying the Database Archive File), you must specify only the file name of the main archive file that you want to restore (for example, samplebasic.arc).

Note:

When restoring a database in which the archive file is split into multiple files, Essbase looks for multiple archive files, even if, after the backup, you set SPLITARCHIVEFILE to FALSE for that database. Also, Essbase expects all of a database's archive files (main and split) to be in the same directory.

You can restore a database using Administration Services or MaxL.

In Administration Services Console, use the Restore Database command. For more information, see "Restoring Block Storage Databases" in Oracle Essbase Administration Services Online Help.

In MaxL, use the alter database statement with the restore from file grammar. The syntax for the statement:

```
alter database appname.dbname [force] restore from file BACKUP-FILE [replace disk volume VOL];
```

Typically, you restore a database to the application and database from which the backup was taken. Therefore, the names in the archive file of the backed-up database and its associated application, which are to be restored, are the same. However, if the names of the backed-up database and application differ from the application and database to which you are restoring data, you must use force grammar. Statement syntax:

```
alter database appname.dbname force restore from file BACKUP-FILE;
```

For example, you can use the backup for Sample.Basic to restore to Sample.New (the database name is different), MyCompany.Basic (the application name is different), or MyCompany.New (both names are different).

For more information, see Retrieving Archive File Information in this guide, and the "alter database" topic in the Oracle Essbase Database Administrator’s Guide.

Restoring Manually Backed-Up Files

After restoring a backed-up database, you typically need not restore the files that you manually backed up (see Table 3).
Restore manually backed up files in these situations:

- If a server interruption corrupted any files, such as security files
- If you encounter problems (such as missing rules files or scripts) when replaying logged transactions

**Working with Disk Volumes**

You can install Essbase on one disk volume and store index files (`essn.ind`), and data files (`essn.pag`) on one or more other disk volumes. Disk volume information is saved in the `dbname.esm` file and is used when restoring a database to ensure that the index and data files are copied to the correct disk volumes.

Even if you change the disk volumes that a database uses, all currently and previously used disk volume information remains in the database files. When retrieving disk volume information from the database archive file or viewing disk volume information in Administration Services Console or in the MaxL Shell, Essbase lists all the disk volumes that have been associated with the database. Additionally, Essbase lists the default `ARBORPATH` drive (for example, on Windows, the `C:` drive, if `ARBORPATH` resides on `C:`) as a disk volume. See Retrieving Archive File Information.

When restoring a database, you can replace disk volume names; however, the number of disk volumes must be the same, and the space required must at least the same, as those for the database before it was backed up.

In Administration Services Console, use the Advanced option in the Restore Database dialog box to replace the disk volume names.

When using the `alter database MaxL` statement with the replace disk volume VOL grammar, the valid values for the VOL argument are a comma-separated list of volumes to replace:

- 'VOL1' with 'VOL2'
- 'VOL3' with 'VOL4'
- 'VOL5' with 'VOL6'

For example, the following statement restores the Sample.Basic database using the `samplebasic.arc` archive file and replaces the specified disk volumes:

```
alter database Sample.Basic restore from file '/Oracle/samplebasic.arc'
replace disk volume 'C' with 'F', 'D' with 'G', 'E' with 'H';
```

**Working with Unicode and Non-Unicode Applications**

When working with Unicode and non-Unicode applications, Essbase does not allow a backed-up database from a Unicode application to be restored to a non-Unicode application.

**Table 4-4** shows the supported combinations of restoring backed-up databases between Unicode and non-Unicode applications.

<table>
<thead>
<tr>
<th>Backed Up From</th>
<th>Restored To</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicode</td>
<td>Unicode</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 4-4 (Cont.) Combinations of Backed Up and Restored Databases by Unicode and Non-Unicode Application Modes

<table>
<thead>
<tr>
<th>Backed Up From</th>
<th>Restored To</th>
<th>Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicode</td>
<td>Non-Unicode</td>
<td>No</td>
</tr>
<tr>
<td>Non-Unicode</td>
<td>Non-Unicode</td>
<td>Yes</td>
</tr>
<tr>
<td>Non-Unicode</td>
<td>Unicode</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Backing Up and Restoring Partitioned Databases

When backing up and restoring partitioned databases, follow these guidelines:

- Back up and restore the source and target databases simultaneously.
- If you restore a database to an application or database whose names are not the same as the names of the application or database that you backed up, you must redefine the partition. See the Oracle Essbase Database Administrator’s Guide.

Using Transaction Logging and Replay For Block Storage Databases

To facilitate recovering a block storage database to a later time than a restored database, you must enable transaction logging. As transactions are executed, Essbase writes operational parameters to a log store. Users with the Essbase Administrator role can view a list of logged transactions and then replay them to recover the data that was not captured after a backed-up database is restored.

Table 5 lists the transactions, by category, that Essbase logs when transaction logging is enabled:

Table 4-5 List of Logged Transactions by Category

<table>
<thead>
<tr>
<th>Category</th>
<th>Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculation</td>
<td>• Execute the default or server-side calculation</td>
</tr>
<tr>
<td></td>
<td>• Execute a client-side calculation</td>
</tr>
<tr>
<td>Data load</td>
<td>• Import data from different sources to Essbase Server</td>
</tr>
<tr>
<td></td>
<td>• Clear all loaded data and reset the outline to empty</td>
</tr>
<tr>
<td></td>
<td>• Clear all loaded data</td>
</tr>
<tr>
<td></td>
<td>• Execute a client-side data load</td>
</tr>
<tr>
<td>Lock and send</td>
<td>Update database cells with data from Oracle Smart View for Office</td>
</tr>
</tbody>
</table>

The operational parameters that are logged for each transaction vary. Table 6 shows the parameters that are logged for three representative transactions:
Table 4-6 Examples of Parameters Logged for Specific Transactions

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Logged Parameters</th>
</tr>
</thead>
</table>
| Execute a client-side calculation | • User who submitted the request  
  • Start time  
  • End time  
  • Sequence ID  
  • RequestType  
  • Calculation script contents, including runtime substitution variables and their values¹ |
| Import data from different sources to Essbase Server | • User who submitted the request  
  • Start time  
  • End time  
  • Sequence ID  
  • RequestType  
  • Rule file name and location  
  • Data file name and location  
  • If SQL source, the SQL connection information  
  • If the data and rule files are on the client computer, archive the files to the Replay directory. The directory path is logged for use during the replay operation.  
  Essbase autogenerates a unique file name for archived data and rule files, such as 98761234.txt and 98761234.rul. |
| Update database cells with data from Smart View | • User who submitted the request  
  • Start time  
  • End time  
  • Sequence ID  
  • RequestType  
  • Essbase tracks information about changed cells |

¹ When a calculation script with runtime substitution variables is logged, the values of the runtime substitution variables are logged with the script. When the logged calculation script is replayed, the runtime substitution variables retain the logged values.

**Note:**

“Replay” is a reserved word and cannot be used as the name of a block or aggregate storage database. If you named a database “replay” in the past, you must rename the database before enabling the transaction logging and replay feature.

**Enabling Transaction Logging**

Transaction logging can be enabled at the server, application, or database level.
To enable transaction logging, you must specify a centralized location where Essbase can write transaction log stores. The location must be to an existing directory on Essbase Server.

Add the TRANSACTIONLOGLOCATION configuration setting to the `essbase.cfg` file. The syntax for the statement:

```
TRANSACTIONLOGLOCATION [appname [dbname]] LOGLOCATION NATIVE ENABLE | DISABLE
```

**Note:**

Do not change the NATIVE argument, which is a reserved field. For the log location, Oracle recommends using a physical disk other than the disk on which the ARBORPATH directory or disk volumes reside.

In `LOGLOCATION`, Essbase creates subdirectories corresponding to the applications and databases for which you enable transaction logging, using the following structure:

```
LOGLOCATION/appname/dbname
```

For example, the following statement enables transaction logging for all databases associated with the Sample application:

```
TRANSACTIONLOGLOCATION Sample /Oracle/trlog NATIVE ENABLE
```

As transactions are logged for the Sample.Basic database, the log store is written to:

```
/Oracle/trlog/Sample/Basic
```

Depending on the number of databases for which you enable transaction logging, you might want to create more than one centralized transaction log location; however, Oracle recommends keeping the number of log locations to a minimum.

Additionally, by using multiple TRANSACTIONLOGLOCATION statements, you can enable transaction logging at a more global level and, at the same time, disable logging at a more granular level. In the `essbase.cfg` file, for the override to take effect, the more global enabling statement must precede the more granular disabling statement.

Consider the following examples:

- The first statement enables transaction logging for all applications and their associated databases on Essbase Server; the second statement disables transaction logging for all databases associated with a specific application (Sample):

  ```
  TRANSACTIONLOGLOCATION /Oracle/trlog NATIVE ENABLE
  TRANSACTIONLOGLOCATION Sample /Oracle/trlog NATIVE DISABLE
  ```

- The first statement enables transaction logging at the application level (Sample); the second statement disables transaction logging for a specific database (Basic) in the application:

  ```
  TRANSACTIONLOGLOCATION Sample /Oracle/trlog NATIVE ENABLE
  TRANSACTIONLOGLOCATION Sample Basic /Oracle/trlog NATIVE DISABLE
  ```
Note:

Before renaming or copying an application or database for which transaction logging is enabled, you must enable logging for the renamed or copied application or database and specify the same LOGLOCATION that is specified for the source application and database.

Configuring Transaction Replay

To enable the replay of logged transactions, Essbase creates archive copies of data load and rules files in `ARBORPATH/app/appname/dbname/Replay`.

By default, Essbase archives only data load and rules files for client data loads.

To change the type of data to be archived, add the `TRANSACTIONLOGDATALOADARCHIVE` configuration setting to the `essbase.cfg` file. The syntax for the statement:

`TRANSACTIONLOGDATALOADARCHIVE [appname [dbname]] [OPTION]`

Valid values for the OPTION argument:

- **CLIENT**—(Default) Archives data load and rules files for client data loads.
- **SERVER**—Archives data load and rules files on the server and SQL-server data loads.
- **SERVER_CLIENT**—Archives server and client data.
- **NONE**—No data is archived.

If you select NONE and you use client data, Essbase cannot replay the data load. To recover transactions, you must manually load the client data before you replay the remaining transactions.

Caution:

Server data loads are replayed using the data load and rules files that are archived on the server in the Replay directory. Do not rename these files. Also, if the contents of the data load and rules files are changed before the replay operation, the modified data is used during replay. Therefore, the data in the recovered database will not be the same as the original data.

- **SERVER_CLIENT**—Archives server and client data.
- **NONE**—No data is archived.

Caution:

If you use server or SQL data and the data and rules files are not archived in the Replay directory (for example, you did not use the SERVER or SERVER_CLIENT option), Essbase replays the data that is in the data source, which may or may not be the data that was originally loaded.
For example, the following statement archives server and client data for all databases on Essbase Server:

```
TRANSACTIONLOGDATALOADARCHIVE SERVER_CLIENT
```

### Viewing Logged Transactions

An administrator can view the list of logged transactions and the information associated with each transaction. You can specify whether to list transactions that were logged after the following conditions:

- A specified time
- The time when the last replay request was executed or the time of the last restored backup (whichever occurred later)

To view logged transactions, use a tool:

<table>
<thead>
<tr>
<th>Tool</th>
<th>Topic</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration Services</td>
<td>Viewing Logged Transactions for Block Storage Databases</td>
<td>Oracle Essbase Administration Services Online Help</td>
</tr>
<tr>
<td>MaxL</td>
<td>query database</td>
<td>Oracle Essbase Technical Reference</td>
</tr>
</tbody>
</table>

In Administration Services Console, use the **Display Transactions** command to view a list of transactions, as shown in **Figure 3** You can sort the list by the following attributes:

- Sequence ID
- Username
- Start Time
- End Time
- Request Type

**Figure 4-3  List of Logged Transactions**

<table>
<thead>
<tr>
<th>#</th>
<th>User</th>
<th>Start Time</th>
<th>End Time</th>
<th>Request Type</th>
<th>Calcscript</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>admin</td>
<td>Tue Apr 00:00:00</td>
<td>Tue Apr 00:00:00</td>
<td>Default calculation</td>
<td>Default Calc...</td>
</tr>
<tr>
<td>6</td>
<td>admin</td>
<td>Tue Apr 00:00:00</td>
<td>Tue Apr 00:00:00</td>
<td>Spread sheet updates</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>admin</td>
<td>Tue Apr 00:00:00</td>
<td>Tue Apr 00:00:00</td>
<td>Spread sheet updates</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>admin</td>
<td>Tue Apr 00:00:00</td>
<td>Tue Apr 00:00:00</td>
<td>Spread sheet updates</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>admin</td>
<td>Tue Apr 00:00:00</td>
<td>Tue Apr 00:00:00</td>
<td>Spread sheet updates</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>admin</td>
<td>Tue Apr 00:00:00</td>
<td>Tue Apr 00:00:00</td>
<td>Default calculation</td>
<td>Default Calc...</td>
</tr>
<tr>
<td>1</td>
<td>admin</td>
<td>Tue Apr 00:00:00</td>
<td>Tue Apr 00:00:00</td>
<td>Spread sheet updates</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>admin</td>
<td>Tue Apr 00:00:00</td>
<td>Tue Apr 00:00:00</td>
<td>Spread sheet updates</td>
<td></td>
</tr>
</tbody>
</table>
Although you can sort the list of transactions, you must replay transactions in the same order as the sequence ID.

When using the `query database` MaxL statement with the `list transactions` grammar, you can specify whether to display the list output in the MaxL Shell window (the default) or to write the list output to a specified file. The syntax for the statement:

```maxl
query database appname.dbname list transactions [after TIME [[force] write to file FILE-NAME]];
```

For example, the following statement displays, in the MaxL Shell window, the transactions in the Sample.Basic database that were logged after the last replay request was executed or after time of the last restored backup (whichever occurred later):

```maxl
query database Sample.Basic list transactions;
```

Writing the output to a file is useful for processing the log information with other tools. For example, you can import the file contents into a relational database and write a tool to analyze the information.

When using the `write to file` grammar, the list output is written to a comma-separated file on the Essbase Server computer. Provide the full path to an existing directory on Essbase Server and the name of the output file. If only the output file name is provided, Essbase writes the file to `ARBORPATH/app`.

When writing to an existing output file, use `force` grammar to overwrite the file.

For example, the following statement writes Sample.Basic database transactions that were logged after November 20, 2009, at 12:20:00 to a CSV file in the Sample.Basic database directory:

```maxl
query database Sample.Basic list transactions after '11_20_2009:12:20:00' write to file EPM_ORACLE_HOME/products/Essbase/EssbaseServer/app/Sample/Basic/listoutput.csv;`
```

### Configuring User Security for Replaying Logged Transactions

In replaying logged transactions, by default, Essbase uses the security settings of the user who originally performed the transaction. If that user no longer exists or if the user's user name is changed, the replay operation fails. To use the security settings of the administrator who performs the replay operation (either explicitly or if the original user's settings cannot be used), use the `REPLAYSECURITYOPTION n` configuration setting in `essbase.cfg`. Values for `n` and whose security settings they specify:

- **1**—(default) The user who originally performed the transaction
- **2**—The administrator performing the replay operation
- **3**—The user who originally performed the transaction

If that user no longer exists or that user's user name was changed, the security settings of the administrator performing the replay operation are used.

### Replaying Logged Transactions

Only an administrator can replay transactions. You can replay logged transactions based on these criteria:

- Transactions logged after a specified time
Transactions logged after the last replay request was executed or after the time of the last restored backup (whichever occurred later)

Selectively, transactions based on a range of sequence IDs

Each logged transaction is assigned a sequence ID, indicating the order in which the transaction was performed. To ensure the integrity of the restored data after a replay, Essbase enforces the replay of transactions in the same order in which they were originally performed. For example, if a data load was followed by a calculation, replaying the calculation before the data load would produce incorrect results; the data load transaction must be replayed before the calculation transaction.

The order of sequence IDs is tracked across multiple replay commands. After a transaction is replayed, you can replay only transactions with a greater sequence ID. For example, after replaying transactions with sequence IDs of 1, 2, and 3, you can replay only transactions with a sequence ID of 4 or greater.

Transactions that are executed and logged after the restore operation are not replayed, unless you replay those transactions using their sequence IDs. After restoring a database, Oracle recommends that you finish replaying the transactions that were logged after the backup and before the restore and that are needed to fully recover the database; then you can continue executing new transactions.

### Note:

You can skip replaying a transaction if you are sure that the transaction results are not required to recover the database. For example, if you performed two full data loads with identical data, you need only replay one of the data loads.

To replay logged transactions, use a tool:

<table>
<thead>
<tr>
<th>Tool</th>
<th>Topic</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration Services</td>
<td>Replaying Logged Transactions for Block Storage Databases</td>
<td>Oracle Essbase Administration Services Online Help</td>
</tr>
<tr>
<td>MaxL</td>
<td>alter database</td>
<td>Oracle Essbase Technical Reference</td>
</tr>
</tbody>
</table>

The replay functionality is the same whether you use Administration Services Console or MaxL.

To selectively replay transactions, enter a comma-separated string of sequence ID ranges. A range can consist of one or more transactions:

- One transaction: $n$ to $n$; for example, 1 to 1
- Multiple transactions: $x$ to $y$; for example, 20 to 100

In Administration Services Console, use the **Replay Transactions** command.

In MaxL, use the **alter database** statement with the **replay transactions** grammar.

Syntax:

```
alter database appname.dbname replay transactions [after TIME] [using sequence_id_range COMMA-SEPARATED_RANGE];
```
For example, the following statement replays the transactions in the Sample.Basic database with sequence IDs 1–10 and 20–100:

```
alter database Sample.Basic replay transactions using sequence_id_range 1 to 10, 20 to 100;
```

### Removing Transaction Log Store and Archived Replay Data Files

Periodically, you might want to remove the transaction log store and the files in the Replay directory to increase available disk space on Essbase Server.

**Transaction log store:** Oracle recommends removing the transaction log store for one database at a time. The log store is in a subdirectory under the log location specified by the TRANSACTIONLOGLOCATION configuration setting. For example, if the log location for the Sample.Basic database is `/Oracle/trlog`, delete the contents of `/Oracle/trlog/Sample/Basic`

See [Enabling Transaction Logging](#).

**Replay directory:** After you have replayed transactions, the data and rules files associated with the replayed transactions can be removed from the `ARBORPATH/app/appname/dbname/Replay` directory (see [Configuring Transaction Replay](#)). You can delete all of the files in the Replay directory, or follow these guidelines for selectively removing files:

- Remove the data and rules files in chronological order, from earliest to latest.
- Do not remove data and rules files with a timestamp that is later than the timestamp of the most recent archive file.

**Note:**

Oracle recommends waiting until after several subsequent database backups before deleting files associated with transaction logging and replay.

### Using Transaction Logging and Replay with Partitioned Databases

When logging and replaying transactions from partitioned databases, follow these guidelines:

- Partition commands (for example, synchronization commands) are not logged and, therefore, cannot be replayed. When recovering a database, you must replay logged transactions and manually make the same partition changes in the correct chronological order.
- When using partitioned databases or using the `@XREF` function in calculation scripts, you must selectively replay logged transactions in the correct chronological order between the source and target databases.

**Note:**

Back up and restore the source and target databases simultaneously. See [Backing Up and Restoring Partitioned Databases](#).
Using Manual Backup and Restore Procedures

If you have designed a backup and recovery strategy using manual procedures, you can continue using them. Oracle, however, recommends that you incorporate backup and restore (see Using Backup and Restore for Block Storage Databases) and transaction logging and replay (see Using Transaction Logging and Replay For Block Storage Databases) into your backup and recovery strategy.

See these topics for manually backing up and restoring block storage databases:

- Backing Up and Restoring Database Files
- Backing Up Files During Runtime
- Backing Up and Restoring Data

Backing Up and Restoring Database Files

Regularly back up the server, application, and database files listed in Table 7. Typically, you should perform a file backup after Essbase applications and databases, and the Agent, are shut down.

Table 4-7  Essbase Files to Back Up

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>essxxxxx.ind1</td>
<td>Essbase index file</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td>essxxxxx.pag2</td>
<td>Essbase data file</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td>dbname.esm</td>
<td>Essbase Kernel file that contains control information used for database recovery</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td>dbname.tct</td>
<td>Transaction control table</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td>dbname.ind</td>
<td>Free fragment file for data and index free fragments</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td>dbname.otl</td>
<td>Outline file, which stores all metadata for a database and defines how data is stored, but does not store data itself</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td>dbname.app</td>
<td>Application file containing application settings</td>
<td>ARBORPATH/app/</td>
</tr>
<tr>
<td>dbname.db</td>
<td>Database file containing database settings</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td>x.lro</td>
<td>Linked reporting objects</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td>essbase.sec3</td>
<td>Essbase security file</td>
<td>ARBORPATH/bin/</td>
</tr>
<tr>
<td>essbase_timestamp.bak</td>
<td>Backup of the Essbase security file</td>
<td>ARBORPATH/bin/</td>
</tr>
<tr>
<td>essbase.cfg</td>
<td>Essbase Server configuration file</td>
<td>ARBORPATH/bin/</td>
</tr>
</tbody>
</table>
Table 4-7  (Cont.) Essbase Files to Back Up

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>.otl</td>
<td>Database artifact files</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td>.csc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.rul</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.rep</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.rep</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.eqd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.sel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESSCMD or MaxL scripts</td>
<td>No defined storage location</td>
<td></td>
</tr>
</tbody>
</table>

1 Back up all .ind files related to a database, because a database may have multiple .ind files.
2 Back up all .pag files related to a database, because a database may have multiple .pag files.
3 Shut down the Agent before backing up essbase.sec.

When performing a file system backup, use the file system backup software of your choice. You can back up specific directories or files or the entire Essbase directory structure. Back up data on every disk volume.

To restore a database, ensure that the application is stopped and replace the files on disk with the corresponding backed up files.

**Backing Up Files During Runtime**

If any Essbase databases must be running at the time of the backup, follow these steps:

1. **Putting a Database in Read-Only Mode**
2. **Performing a File Backup**
3. **Returning a Database to Read-Write Mode**
4. **Restoring the File Backup**

**Putting a Database in Read-Only Mode**

Putting the database in read-only (or archive) mode protects it from updates during backup.

To put a database in read-only mode, use a tool:

<table>
<thead>
<tr>
<th>Tool</th>
<th>Topic</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxL</td>
<td>alter database begin archive</td>
<td>Oracle Essbase Technical Reference</td>
</tr>
<tr>
<td>ESSCMD</td>
<td>BEGINARCHIVE</td>
<td>Oracle Essbase Technical Reference</td>
</tr>
</tbody>
</table>

After performing the backup, return the database to read-write mode.
The `BEGINARCHIVE` command performs the following tasks:

- Commits modified data to disk.
- Switches the database to read-only mode.
- Reopens the database files in shared, read-only mode.
- Creates, in the `ARBORPATH/app/appname/dbname` directory, a file (default name `archive.lst`) that lists files to be backed up.

Attempting to modify data during the backup process results in an error message that data is in read-only mode.

Begin archive does not perform the backup; it protects the database during backup. If you cancel the `BEGINARCHIVE ESSCMD` command or the `alter database begin archive MaxL` statement, and you receive a “can’t cancel” message, the system may be in the final stage of writing items to the drive, when the operation cannot be canceled.

**Caution:**

If you back up data without using `BEGINARCHIVE`, ensure that all Essbase applications are closed and that all users are logged off during the backup, to avoid corrupting the database.

### Performing a File Backup

Put the database in read-only mode before you perform the backup.

To back up data, use a third-party backup utility to back up the files listed in `archive.lst` and in Table 7, or back up the entire Essbase directory structure.

### Returning a Database to Read-Write Mode

After performing a backup, return the database to read-write mode.

To return the database to read-write mode, use a tool:

<table>
<thead>
<tr>
<th>Tool</th>
<th>Topic</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxL</td>
<td><code>alter database end archive</code></td>
<td>Oracle Essbase Technical Reference</td>
</tr>
<tr>
<td>ESSCMD</td>
<td><code>ENDARCHIVE</code></td>
<td>Oracle Essbase Technical Reference</td>
</tr>
</tbody>
</table>

The end-archive utility performs the following actions:

- Returns the database to read-write mode.
- Reopens database files in exclusive, read-write mode.
Note:

You must use the end-archive utility to put the database back into read-write mode, even if you shut down and restart the database. The end-archive utility does not restart the database.

Restoring the File Backup

To restore a database, replace the files on disk with the corresponding files from the backup.

The application should be stopped unless you are restoring from an export file. In that case, ensure that the application is not accepting client connections.

Backing Up and Restoring Data

Exporting data copies the data to a file that you specify.

When you export data to a text file, the data is not compressed. The text export file contains only data; it does not include control, outline, or security information.

Consider exporting data to accomplish the following tasks:

- Transfer data across platforms
- Back up only part of the data; for example, level-0 blocks
- Create an exported file in text format rather than binary format

The advantages of exporting data:

- You can use text export files to load data from the source database into databases on other platforms. Exporting a database in column format enables you to manipulate the export file and use it with a rules file to load the data back or to another database.

  Loading an export file into a database fails if the database outline does not contain all the members found within the export file.

  If the outline changes between the time that the export file is created and reloaded (and the new outline contains all the members found within the export file), the load time might be significantly longer than if the outlines were identical.

- Data integrity is verified because every block is checked to confirm whether corresponding page and index files match.

- Exporting data, clearing all data from the database, and reloading the text file can reduce fragmentation.

The disadvantages of exporting data:

- Unless dynamic calculations are executed at the time of the export, only stored data and data from previously calculated Dynamic Calc and Store members are included in the export.

- During a database export, users cannot write to the database. After an export has started, users can do read operations. Exports of large databases require considerable time, during which users can only read the data.
Exporting Data

The same basic information is required, regardless of the method used to export data:

- Names of export data files
- The amount of data to export
  - All data
  - Level-0 blocks only (blocks containing only level-0 sparse member combinations; these blocks may contain data for upper-level dense dimension members)
  - Data from input blocks only (blocks containing data from a previous data load or spreadsheet lock and send)
- Whether to export data in a columnar or noncolumnar format

In each row, the columnar format displays a member name from every dimension. Names can be repeated from row to row.

The columnar format provides a structure to the exported data, so that it can be used for further data processing by applications other than Essbase tools; for example, relational databases. In noncolumnar format, sparse members identifying a data block are included only once for the block. Because the export file in noncolumnar format is smaller than in columnar format, reloading a file in noncolumnar format is faster.

To export data, use a tool:

<table>
<thead>
<tr>
<th>Tool</th>
<th>Topic</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration Services</td>
<td>Exporting Databases</td>
<td>Oracle Essbase Administration Services Online Help</td>
</tr>
<tr>
<td>MaxL</td>
<td>export data</td>
<td>Oracle Essbase Technical Reference</td>
</tr>
<tr>
<td>ESSCMD</td>
<td>EXPORT or PAREXPORT</td>
<td>Oracle Essbase Technical Reference</td>
</tr>
<tr>
<td>Calculation script (block storage only)</td>
<td>DATAEXPORT calculation commands</td>
<td>Oracle Essbase Technical Reference</td>
</tr>
</tbody>
</table>
Note:
To improve export performance, you can export data in parallel to a specified number of files.

Exporting Files Larger Than 2 GB

Some file systems do not support text files larger than 2 GB. On any OS, if Essbase anticipates that an export file exceeds 2 GB, it creates two or more export files, as needed. The requested file name for the main file is used. An underscore and a sequential cardinal number are appended to the names of the additional files, starting with _1. For example, if the requested file name is `expJan.txt`, and the exported data would exceed 4 GB, Essbase creates three files, naming them `expJan.txt`, `expJan_1.txt`, and `expJan_2.txt`. Exported data files can be reloaded in any sequence.

Restoring the Data Backup

When you reload data that has been exported, it is marked as input data. If you reload data exported from level-0 blocks or input blocks, you must recalculate the database after reloading. When Essbase recalculates the database, it recalculates every data block.

If you export all data in a database and then reload, Essbase marks all blocks in the database as input blocks. Consequently, you cannot clear data, because the database does not contain noninput blocks.

When you reload data that has been exported, Essbase also marks the data blocks as dirty. If you calculated the database before exporting it, to save time during the next calculation, set the status of the blocks as clean. If you did not calculate the database before exporting it, you need not set the status of the blocks as clean.

To reload exported data, use a tool:

<table>
<thead>
<tr>
<th>Tool</th>
<th>Topic</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration Services</td>
<td>Performing a Data Load or Dimension Build</td>
<td><a href="#">Oracle Essbase Administration Services Online Help</a></td>
</tr>
<tr>
<td>MaxL</td>
<td>import data</td>
<td><a href="#">Oracle Essbase Technical Reference</a></td>
</tr>
<tr>
<td>ESSCMD</td>
<td>IMPORT</td>
<td><a href="#">Oracle Essbase Technical Reference</a></td>
</tr>
<tr>
<td>Calculation script</td>
<td>DATAIMPORTBIN (exported binary files only)</td>
<td><a href="#">Oracle Essbase Technical Reference</a></td>
</tr>
</tbody>
</table>

To clean block status in a database after exporting all data and reloading, run the following calculation script:

```plaintext
Set ClearUpdateStatus Only;
Calc All;
```
Maintaining LRO Links

LROs (linked reporting objects) are not restored with backed-up databases; therefore, you must export them and then import them.

As part of your database backup procedure, export the LROs.

To export LRO links, use a tool:

<table>
<thead>
<tr>
<th>Tool</th>
<th>Topic</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration Services</td>
<td>Exporting LROs</td>
<td>Oracle Essbase Administration Services Online Help</td>
</tr>
<tr>
<td>MaxL</td>
<td>export lro</td>
<td>Oracle Essbase Technical Reference</td>
</tr>
</tbody>
</table>

As part of your database restore procedure, import the LROs that you previously exported.

To import LRO links, use a tool:

<table>
<thead>
<tr>
<th>Tool</th>
<th>Topic</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration Services</td>
<td>Importing LROs</td>
<td>Oracle Essbase Administration Services Online Help</td>
</tr>
<tr>
<td>MaxL</td>
<td>import lro</td>
<td>Oracle Essbase Technical Reference</td>
</tr>
</tbody>
</table>

Resolving Corrupt Database Files

The database does not start on Essbase Server if database files are corrupt.

To restore a corrupt database:

1. Delete these files:
   - `esaxxxxx.pag`
   - `dbname.esm`
   - `esaxxxxx.ind`
   - `dbname.tct`
   - `dbname.ind`
2. Restart the database.
3. Reload from data or export files that you created before the corruption.

Preserving Environment Variable Values

Note the values of the ARBORPATH and ESSLANG environment variables.

Backing Up and Restoring Aggregate Storage Applications

The file structure described in Table 8 is unique to aggregate storage applications.
### Table 4-8  Aggregate Storage Application Files

<table>
<thead>
<tr>
<th>Directory or File</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>appname</code></td>
<td>Application directory</td>
<td>ARBORPATH/app/appname</td>
</tr>
<tr>
<td><code>appname.app</code></td>
<td>Application file containing application settings</td>
<td>ARBORPATH/app/appname/</td>
</tr>
<tr>
<td><code>appname.log</code></td>
<td>Application log file</td>
<td>ARBORPATH/app/appname/</td>
</tr>
<tr>
<td><code>dbname</code></td>
<td>Database directory</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td><code>dbname.db</code></td>
<td>Database file containing database settings</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td><code>dbname.dbb</code></td>
<td>Backup of database file</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td><code>dbname.ddb</code></td>
<td>Partition definition file</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td><code>dbname.otl</code></td>
<td>Outline file</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td><code>dbname.otl.keep</code></td>
<td>Temporary backup of <code>dbname.otl</code> (created by operations that modify the outline and write it to a new file.)</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td><code>trigger.trg</code></td>
<td>Trigger file</td>
<td>ARBORPATH/app/appname/dbname/</td>
</tr>
<tr>
<td><code>default</code></td>
<td>Tablespace directory (can be in multiple locations that you define)</td>
<td>(Default location)</td>
</tr>
<tr>
<td><code>temp</code></td>
<td>Tablespace directory (can be in multiple locations that you define)</td>
<td>(Default location)</td>
</tr>
<tr>
<td><code>log</code></td>
<td>Tablespace directory</td>
<td>ARBORPATH/app/appname/log</td>
</tr>
<tr>
<td><code>metadata</code></td>
<td>Tablespace directory</td>
<td>ARBORPATH/app/appname/metadata</td>
</tr>
<tr>
<td><code>essn.dat</code></td>
<td>Aggregate storage data file</td>
<td>ARBORPATH/app/appname/default/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARBORPATH/app/appname/log/</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARBORPATH/app/appname/metadata/</td>
</tr>
</tbody>
</table>

To back up an aggregate storage database:

1. Stop the application.
2. Use the file system to copy the contents of the application directory (ARBORPATH/app/appname), excluding the temp directory.

**Exporting and Importing Partition Definitions**

The section applies to block storage and aggregate storage databases.
If you have Database Manager permissions for a partitioned database, you can create backup copies of partition definitions by exporting them. Partition definitions must first be saved to the file system or network as .xml files. You can import only partition definition files that have the .xml extension.

For instructions on exporting and importing partition definitions, see “Exporting Partition Definitions” and “Importing Partition Definitions” in the *Oracle Essbase Administration Services Online Help*.

**Essbase Studio**

To back up and restore Oracle Essbase Studio components to a previous state:

1. Stop the Essbase Studio server.
2. Perform a complete backup of the Essbase Studio metadata repository.
3. Keep a current copy of installed software and all property files (server.properties, essbase.properties, and eas.properties).

**Provider Services**

Back up these files and folders:

- **EPM_ORACLE_HOME/products/Essbase/aps/bin/datasources.xml** file—Contains all data source connection information for the various providers.
- **EPM_ORACLE_HOME/products/Essbase/aps/data** folder—Contains all Smart Slice definitions managed centrally by Oracle Hyperion Provider Services
  - Back up the entire folder, including subfolders.
- **EPM_ORACLE_INSTANCE/products/Essbase/aps/bin/Essbase.properties**

You can restore the backed up files and folders by returning them to their original locations.
5

Financial Performance Management
Applications

Related Topics
• Planning
• Financial Management
• Tax Provision
• Financial Close Management
• Tax Governance
• Profitability and Cost Management

Planning

To enable Oracle Hyperion Planning recovery from a failure:

1. Back up related components, as described in Common Backup Tasks.

   Note:
   Be sure to back up the Planning system database as well as all individual application databases.

   Backups of applications and related application databases must include these items:
   • The application in Oracle Essbase
   • The relational database for Planning
   • Required Planning components

   Tip:
   Rather than back up the entire Windows registry, you can back up only these keys for Planning: HKLM/Software/Oracle* nodes and HKLM/System/CurrentControlSet/Services/Oracle*.

2. Back up the Essbase outline files for your applications.
3. Back up the full export of Essbase data. See Essbase.
4. Back up any calculation scripts and substitution variables associated with Essbase.
5. Back up the file system folder that contains Planning shared libraries:
   • 32-bit: \EPM_ORACLE_HOME\products\Planning\lib
   • 64-bit: \EPM_ORACLE_HOME\products\Planning\lib64

6. Back up common components in \EPM_ORACLE_HOME\common\planning\11.1.2.0\lib.

7. Ensure that the backup of \EPM_ORACLE_HOME\products\Planning\directory includes these files, and back them up manually if it does not:
   • log4J.properties
   • essbase.properties

To recover after a failure:

1. Stop all product and related services. See the Oracle Enterprise Performance Management System Installation and Configuration Guide.
2. Replace backed up files and directories in the original locations.
3. Restore the Planning databases.
4. Restart all servers and services.

See also Essbase Components.

---

Financial Management

Use the following procedure on every application and Web server to back up Oracle Hyperion Financial Management.

To enable recovery of Financial Management:

1. Perform a full database backup once per cycle:
   a. Stop Financial Management Web Service and Financial Management Java Service:
      • Oracle Hyperion Financial Management - Java Server (epmsystem1)
      • Oracle Hyperion HFM Server - Java Web Application (epmsystem1)
   b. Stop the related process xfmdatasource.exe.
2. Back up the Oracle Hyperion Shared Services Registry.
3. Back up any customized style sheets.

**Note:**

You must also back up Oracle Hyperion Shared Services.

To recover from an application server or Web server failure:

1. Reinstall and configure Financial Management on the failed application or Web server.
2. Stop all Financial Management processes and services on the newly configured server.
3. Restore the backed-up files to their original locations.
4. Restart the services.

To recover from a database server failure:

1. Stop all Financial Management processes and services on all application and Web servers.
2. Restore the database from the last full backup, applying transaction logs as needed.
3. Restart the database.
4. Restart the application and Web servers.

Tax Provision

To enable recovery of Oracle Hyperion Tax Provision, follow the procedures described in Financial Management.

Financial Close Management

To enable recovery Oracle Hyperion Financial Close Management data, back up the database. You can perform hot backups or cold backups. See Database Backup Types and your RDBMS documentation.

For additional details about how to clone a production environment in order to create a test environment, see Knowledge Article 1903665.1 on My Oracle Support (https://support.oracle.com/rs?type=doc&id=1903665.1), which provides instructions for a single node environment.

Tax Governance

To enable recovery of Oracle Hyperion Tax Governance, follow the procedures described in Financial Close Management.

Profitability and Cost Management

To enable Oracle Hyperion Profitability and Cost Management recovery from a failure:

1. Back up related components as described in Common Backup Tasks.
2. Back up the Profitability and Cost Management import staging area and the operational data store.
   Use the backup tools for your relational database. This could include using scripting or scheduler scripts.
   See Essbase Components, and the Oracle Essbase Database Administrator’s Guide.

To recover from a failure, restore the backed-up components to their original locations and restore the database.
Financial Reporting


To back up the Document Repository:

2. From the Navigate menu, select Administer, and then select Shared Services Console.
3. In Application Management, expand the Application Groups node, and then expand Default Application Group.
4. Right-click Document Repository and then click Explore.
5. Select the repository objects you want to back up, and then click Export.
6. Enter a descriptive name in File System Folder, and then click Export.
7. In Application Management, expand the File System node, right-click the file you exported, and then click Download.

Back up the exported file with other artifacts as described in Common Backup Tasks.

You can download log files using the ziplogs.bat utility in the EPM_ORACLE_INSTANCE\bin.
7

Data Management

Related Topics
- Data Relationship Management
- FDMEE

Data Relationship Management

Oracle Data Relationship Management uses separate databases and schemas for each application that you create.

To enable Data Relationship Management recovery from a failure:
1. Back up related components and application databases as described in Common Backup Tasks.
2. Regularly back up these files in EPM_ORACLE_HOME/products/DataRelationshipManagement/server/config to a storage device:
   - drm-config.xml — Connection information for all Data Relationship Management applications that are hosted on the machine
4. Back up scripts for the Data Relationship Management Batch Client that are used to run automated processes. The location of these files depends on the computer where the Batch Client program is run.
5. Back up XML files created by the Data Relationship Management Migration Utility (for archival purposes). The location of these files depends on the computer where the Migration Utility is run.

To recover after a failure:
1. Restore the copied database and configuration files to their original locations.
2. Reenter the passwords for the database in the configuration file.

FDMEE

To enable Oracle Hyperion Financial Data Quality Management, Enterprise Edition recovery after a crash, regularly back up the database schema that stores the mapping tables and data staging tables.

To recover after a failure, restore the directory to its original location and restore the database as described in the database vendor documentation.