IBM® DB2® OLAP Server™ for iSeries™

Administrator's Guide for OLAP on iSeries

Version 8.1

SC18-7226-00

IBM® DB2® OLAP Server™ for iSeries™

Administrator’s Guide for OLAP on iSeries

Version 8.1

SC18-7226-00

Note

Before using this information and the product it supports, be sure to read the information in “Notices” on page 61.

First Edition (November 2002)

This edition applies to version 8.1 of DB2 OLAP Server for iSeries (program number 5724–B78) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 1998, 2002. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Portions of this book © 2002 SPSS Corporation. All Rights Reserved.

Contents

© Copyright IBM Corp. 1998, 2002

Preface

Welcome to the DB2 OLAP Server for iSeries V8.1 *Administrator's Guide for OLAP on iSeries*. This guide was written by ShowCase (a division of SPSS, Inc.) for their Essbase/400 product under the title *Essbase/400 6.5 Administrator's Guide*. This guide discusses the administration and maintenance of the OLAP components of DB2 OLAP Server for iSeries V8.1, which is called Essbase/400 throughout this guide.

This guide contains references to ShowCase components, functions, and books that are not part of the IBM product. Disregard these references. For a description of exactly which books are shipped with DB2 OLAP Server for iSeries V8.1, see the preface to the *Installation Guide for iSeries*.

This preface provides the following information:

- What you will find in this guide
- Where to find additional information
- How to contact the documentation team to provide feedback about this guide

About This Guide

- Chapter 1, "Starting and Stopping the Servers" on page 1, explains how to start and stop the Essbase/400 servers.
- Chapter 2, "Controlling and Managing Performance" on page 5, explains how to manage performance to ensure successful deployment.
- Chapter 3, "Essbase/400 Server Maintenance" on page 11, explains how to use and maintain your Essbase/400 library. If you are familiar with Essbase on other platforms, we recommend reading this chapter to understand how Essbase on an iSeries platform differs.
- "Commands" on page 51, contains a list of commands used to maintain Essbase/400 libraries.

© Copyright IBM Corp. 1998, 2002

Chapter 1. Starting and Stopping the Servers

Note: Before performing a task on the iSeries that may jeopardize your system stability, such as applying PTFs, it is recommended that you stop the servers.

Before You Begin

If this is the first time that Essbase/400 is being run and you did not enter Essbase information during installation, you will be prompted for the following information when you start the servers:

Enter your company name
Enter the name of the company to which Essbase/400 is licensed.

Enter your initial system supervisor user ID
Enter an Essbase/400 system user ID. This user ID is independent of your OS/400 user profiles. It does not need to match an existing OS/400 user profile name, although you may choose a name that corresponds to an iSeries user ID.

Enter the system password
Enter a password for the user ID that you specified in the previous step.

Confirm the system password
Reenter the password.

Starting the Servers

The `STRESSSVR` command will start one or all of the Essbase/400 servers. It has two parameters: the `SERVER` parameter indicates which server to start, and the `JOBD` parameter indicates which OS/400 job description to use for the server job.

The `SERVER` parameter may have one of three values:

`*ALL` indicates that every Essbase/400 server should be started.

`*OLAP` indicates that Essbase/400 OLAP Server should be started. This is the default value.

`*INTEGRATION` indicates that Essbase/400 Integration Server should be started.

The `JOBD` parameter indicates which job description to use for the server job. If the `SERVER` parameter uses `*ALL`, all server jobs will use the same job

© Copyright IBM Corp. 1998, 2002

description. If the `SERVER` parameter uses a specific server, such as `*OLAP` or `*INTEGRATION`, only that server will use the specified job description.

To use different job descriptions for different servers, run the `STRESSSVR` command multiple times. For example, you could start the Essbase/400 OLAP Server with this command: `STRESSSVR SERVER(*OLAP) JOBD(ESSBASE/ESSJOBD)`.

Next you could start the Essbase/400 Integration Server with this command: `STRESSSVR SERVER(*INTEGRATION) JOBD(ESSBASE/EISJOBD)`

To Start the Servers

Note: Be sure that iSeries TCP/IP (`STRTCP`) is running *before* starting the servers.

1. Log on with `QSECOFR` or equivalent authority (`*ALLOBJ`, `*IOSYSCFG`, `*SAVSYS`, `*JOBCTL`, and `*SECADM` special authorities are needed).

   Note: The `GRTESSAUT` command allows a user who lacks the proper authority to start or stop the servers.
2. Add the Essbase/400 OLAP Server library to your library list.

To start all Essbase/400 Servers:

3. Enter the command `STRESSSVR SERVER(*ALL) JOBD(ESSBASE/ESSJOBD)` where `ESSBASE` is your Essbase library and `ESSJOBD` is your job description.

To start only Essbase/400 OLAP Server:

4. Enter the command `STRESSSVR SERVER(*OLAP) JOBD(ESSBASE/ESSJOBD)` where `ESSBASE` is your Essbase library and `ESSJOBD` is your job description.

To start only Essbase/400 Integration Server:

5. Enter the command `STRESSSVR SERVER(*INTEGRATION) JOBD(ESSBASE/EISJOBD)` where `ESSBASE` is your Essbase library and `EISJOBD` is your job description.

Stopping the Servers

The `ENDESSSVR` command will stop one or all of the Essbase/400 servers. The `SERVER` parameter indicates which server to stop.

The `SERVER` parameter has three values:

`*ALL` indicates that every Essbase/400 server and server job should be stopped. This is the default value.

`*OLAP` indicates that Essbase/400 OLAP Server and server job should be stopped.

***INTEGRATION**
indicates that Essbase Integration Server and server job should be stopped.

To Stop the Servers

The ENDESSSVR command will terminate all current TCP/IP connections to the specified library.

Note: To stop the OLAP Server, a username and password are required. If they aren't specified, the *CURUSR value is used, which may not be a valid for the OLAP Server.

1. Log on with QSECOFR or equivalent authority (*ALLOBJ, *IOSYSCFG, *SAVSYS, *JOBCTL, and *SECADM special authorities are needed).

   Note: The GRTESSAUT command allows a user who lacks the proper authority to stop or start the servers.
2. Add the Essbase/400 OLAP Server library to your library list.

To stop all Essbase/400 Servers:

3. Enter the command `ESSBASE/ENDESSSVR SERVER(*ALL) username password` where ESSBASE is your Essbase library.

To stop Essbase/400 OLAP Server:

4. Enter the command `ESSBASE/ENDESSSVR SERVER(*OLAP) username password` where ESSBASE is your Essbase library.

To stop Essbase/400 Integration Server:

5. Enter the command `ESSBASE/ENDESSSVR SERVER(*INTEGRATION)` where ESSBASE is your Essbase library.

Verifying That the Servers are Running

You can use the WRKACTJOB command to verify that your Essbase/400 servers are running. The Essbase/400 OLAP Server job will have a name identical to the job description name, a job type of BCH, and a function of PGM-ESSBASE.

The Essbase/400 Integration Server will have a name identical to the job description name, a job type of BCH, and a function of PGM-OLAPISVR.

Chapter 2. Controlling and Managing Performance

This chapter provides insight into performance planning and tuning of multidimensional databases.

Essbase/400 Security

Essbase/400 security planning is a necessary part of building a multidimensional data warehouse. For detailed information on defining Essbase/400 security settings, see Chapter 2, “Controlling and Managing Performance” in this book, or the Hyperion Essbase *Database Administrator’s Guide*.

Optimizing Essbase/400 Performance

There are two important areas of optimization: optimizing the Essbase/400 OLAP Server, and optimizing the iSeries system on which the server is running.

Tuning Essbase/400 Performance

Load, calculation, and retrieval performance are critical to a successful Essbase/400 implementation. Performance varies based on the relationships between the source data, the amount of data, the configuration of the Essbase/400 outline, the size and complexity of your retrievals, and the tuning applied to the database settings.

Note: It is important that you understand the concepts in the *Hyperion Essbase Database Administrator’s Guide* and IBM’s *iSeries Work Management Guide*.

Designing an Optimized SQL Data Load Plan

To optimize an Essbase/400 data load:

- Sort your input records so that the same sparse member combinations are grouped consecutively. In addition, sort your input records so that the sparse dimensions appear in the same order as the sparse dimensions in the outline. Sorting data on the sparse dimension members minimizes the number of times an Essbase data block must be read from disk, decompressed, enlarged, recompressed, and written back to disk. To sort your input data, specify the appropriate `ORDER BY` clause on the SQL statement in your rule file.
- Load the data from a data source on the server. If the data is not on the source, moving it to the server will improve load times since the data does not have to be transported over the network at load time.

© Copyright IBM Corp. 1998, 2002

For additional information and other optimization tips, see the *Hyperion Essbase Database Administrator's Guide*.

Preventing Storage Overflow

Essbase data load or calc operations may exceed the predetermined Auxiliary Storage lower-level limit and eventually crash the system. You can set up the iSeries to stop the data load and/or calc operation before storage overflow to prevent the system from crashing.

To stop the data load and/or calc operation before storage overflow:

1. Set the QSTGLOWACN system value to *REGFAC.
2. Register SCSTGEXIT to QIBM_QWC_QSTGLOWACN by typing the following at the iSeries command line:

   ```
   ADDEXITPGM EXITPNT(QIBM_QWC_QSTGLOWACN) FORMAT(STGL0100) PGMNBR(1) PGM(ESSBASE/SCSTGEXIT)
   ```

   where *ESSBASE* is the name of the Essbase/400 installation library.

   Note: The PGMNBR parameter must not already be in use.
3. When there is enough storage available, run the following command on an OS/400 command line before Essbase can resume normal functions:

   ```
   RMVENVVAR ENVVAR(OS400STGLOWLMTREACHED) LEVEL(*SYS)
   ```

Optimizing the Essbase Calculation Process

Calculating an Essbase/400 model is CPU-intensive, and decisions you make about the calculation process can affect performance. For more information about optimizing your Essbase calculations, see the *Hyperion Essbase Database Administrator's Guide*.

Tuning iSeries Performance

Performance can change depending on the system resources available to Essbase/400 (CPU, number and type of disk units, and main memory), the number of users accessing the data, and other concurrent activities on the system. For more information on tuning your iSeries, see the IBM *Work Management Guide*.

Once you understand your analyst's requirements, examine your implementation environment to determine the limits placed on your solution. The following sections introduce issues to consider and questions to answer before continuing with the next phase of your implementation.

Balancing Priority between Warehouse Builder and Essbase

If you attempt to run Warehouse Builder distributions while running Essbase servers, the two activities will compete for system resources. This is especially true if you are running on a single-processor system.

Depending on how you installed Warehouse Manager, Warehouse Builder distributions run a default priority of 20. If you start the Essbase/400 Server as a typical batch job, it runs at a default batch priority of 50. Both of these tasks are iSeries resource-intensive. Since Warehouse Builder jobs run at a higher priority, your Essbase operations (load/calc/retrievals) will run very slowly.

Note: To prevent slow Essbase operations, set identical tuning priorities for your server and data distributions. For example, if you run the server at priority 50, set the data distribution to priority 50 too. In addition, set your time slice under 25 milliseconds to prevent starvation of one of the jobs.

Controlling Memory Allocations

Essbase/400 performance is largely affected by the amount of memory it is allowed to use, and you can manipulate the standard OS/400 work management techniques to control memory allocations. On the iSeries, memory is allocated to various memory pools by OS/400. For example, memory can be allocated to the `*BASE` pool, `*INTERACTIVE` pool, or other user-defined pools. For more information on memory pools, consult the IBM *Work Management Guide*.

You can run Essbase/400 under the `*BASE` pool or under a dedicated memory pool. Use a sufficiently large memory pool that does not compete excessively with other applications running on your iSeries. We suggest starting with a memory pool of 256MB for small databases accessed by a limited number of users. As you increase the number of databases, the size of the databases, or the number of users, consider increasing the amount of memory available for Essbase/400.

Because every Essbase/400 database is unique, we cannot recommend specific system size and memory requirements.

Grouping Databases into Applications

Typical Essbase/400 installations run many Essbase/400 databases on a single iSeries. While you can group several databases into a single application, a single application and database for each business problem is usually the best choice to simplify administrative tasks.

Essbase/400 and Symmetric Multiprocessing

Symmetric multiprocessing (SMP) refers to a system with more than one main processor. For example, some iSeries models can be equipped with two, four, eight, or more processors. In this case, SMP is a feature of the hardware. Multiple processors let the system simultaneously process more than one task.

On the iSeries, SMP can also mean a DB2/400 software offering that can improve DB2 query performance regardless of the number of system processors. This software offering gives OS/400 the ability to process a query in several simultaneous steps run in separate jobs.

Essbase/400 is implemented as a multithreaded program. Usually this allows Essbase/400 to take advantage of multiple processors.

Currently, the Essbase calculation process of a given database is single threaded and uses only one processor. To use multiple processors during database calculation, subdivide the database by creating transparent partitions. You can then use multiple processors by calculating each partition at the same time. For more information on transparent partitions, see the *Hyperion Essbase Database Administrator's Guide*.

Since each database is calculated under a separate thread, you can use multiple processors when calculating more than one database at the same time. Performance will vary depending on other iSeries resources such as the amount of available memory and the performance of your disk units. Performance also depends on other system activity.

If you have a single-processor system, limited memory, or limited I/O capacity (few disk arms), you will achieve better results by calculating only one database at a time.

Essbase/400 takes advantage of multiple processors while processing end user retrievals. This applies to retrievals submitted using the spreadsheet add-ins, Analyzer, or user-written report scripts. For each retrieval, Essbase uses two threads. As multiple users run simultaneous retrievals, Essbase uses multiple threads to process these retrievals. Because Essbase is using multiple threads, it takes advantage of SMP, and the work is spread across multiple processors.

Essbase Query Optimization Tips

This section contains tips for improving the performance of your Essbase queries. Many of these Essbase query optimization features can be found in the Essbase Options dialog box in Query and Report Writer.

Limiting Rows

If you have created a complex Essbase query or one that might return a large quantity of data, test the query first by limiting the number of rows returned. By doing so, the query will return only a specified number of rows, indicating that the query runs correctly. To limit rows, you can use any of the following three options:

Top/bottom sorting

Use top/bottom sorting to return an exact number of top- or bottom-ranked values in your report. For example, if your database contains 2,000 cities, and you want to sort sales by city and display only the top 10 cities by sales, you can apply top/bottom sorting to one column per report. To apply top/bottom sorting, see "Applying top/bottom sorting" in the index of the Query or Report Writer online Help.

Suppressing rows that are missing all values

Use this option to exclude rows containing only #MISSING values from your result data. #MISSING means the data in a given cell does not exist. This may be because the member combination is not meaningful (for example, a given company does not sell toboggans in Miami) or has never been entered. This option can be handy if you have many rows with no values. To activate this option, from the Query menu, choose **Essbase Options** and then in the Essbase Options dialog box, select **Suppress Rows that are Missing All Values**.

Suppressing rows that are all zeros

Use this option to exclude rows containing only zeros from your result data. To activate this option, from the Query menu, choose **Essbase Options** and then in the Essbase Options dialog box, select **Suppress Rows that are All Zeros**.

Suppressing Duplicate Shared Members

Suppress Duplicate Shared Members is an Essbase command that suppresses the display of duplicate shared members, reducing the amount of output data returned. Use this option to remove duplicate rows from your result data by excluding these duplicate shared members from your query. To activate this option, from the Query menu, choose **Essbase Options,** and then in the Essbase Options dialog box, select **Suppress Duplicate Shared Members**. For more information about shared members, see "Suppress Duplicate Shared Members" in the index of the Query or Report Writer online Help.

Creating Sparse Reports

Sparse reporting data retrieval is an Essbase feature that may improve data retrieval. Query and Report Writer can use this method to request data from the Essbase/400 Server. The Generate a Sparse Report option uses the sparse data extraction method, which optimizes performance when a high proportion of the requested data rows are #MISSING. For more information on how sparse reporting works and when to use it, see "sparse reporting" in the index of the Query or Report Writer online Help.

Intelligent Sparse Reporting (ISR) is a separately licensed enhancement to Essbase/400 that enables faster query/retrieval processing for Essbase databases containing related dimensions comprised of a large number of members (defined as sparse dimensions to the Essbase engine). For example, a database may contain several dimensions representing product SKUs such as product, color, and size.

Chapter 3. Essbase/400 Server Maintenance

This chapter discusses maintenance procedures and operating considerations for Essbase/400 servers. For information about starting and stopping the servers, see Chapter 1, "Starting and Stopping the Servers" on page 1.

Understanding the Essbase/400 Configuration

Essbase/400 servers provide two mechanisms for configuring information on the iSeries: the configuration file and environment settings. The following sections explain:

- The objects that Essbase/400 installs
- The Essbase/400 environment and how to use the iSeries configuration file
- How the configuration file relates to the Essbase environment settings

Essbase/400 Installation Library and Directory

During installation, you can specify an Essbase/400 installation library name. The default name is `ESSBASE`. The installation program uses the library name to create both a *QSYS.LIB* library and a root file system directory. Thus, the default root file system directory name is also *ESSBASE*.

For example, if you specify `MYBASE` as the installation library name, the installation program creates a library named `MYBASE` in the *QSYS.LIB* file system and a directory named *MYBASE* in the root file system.

To differentiate between the areas that Essbase/400 uses to store information, we will refer to the library as the *Essbase library* and the root file system directory as the *Essbase directory*.

Essbase/400 QSYS.LIB Objects Created During Installation

Essbase/400 uses the root file system for user data associated with Essbase application databases. The data in the Essbase library consist of configuration and program information.

The following sections explain which objects are stored in the *QSYS.LIB* file system. If you are familiar with an Essbase file not mentioned here, it can probably be found in the Essbase directory structure in the root file system.

Files The configuration file is a source physical file named *CFG* and contains a member called *ESSBASE*. In most Essbase documentation, the file is referred to as *ESSBASE.CFG*.

© Copyright IBM Corp. 1998, 2002

Server and application logs are written in ASCII format to IFS. To view the logs from an iSeries command line, use the DSPESSLOG command. DSPESSLOG can also copy the logs to the *QSYS.LIB* file system.

Miscellaneous Objects

Essbase/400 also includes miscellaneous objects that implement iSeries-specific functions. The miscellaneous objects include user spaces to store environment information, commands to provide user interfaces, a message queue for additional logging support, and a message file that logs iSeries-specific error messages.

Using Essbase/400 Configuration File Settings

Where *ESSBASE.CFG* is referenced in the documentation, substitute the Essbase/400 equivalent, *essbaselib/CFG member ESSBASE* (where *essbaselib* is the Essbase/400 installation library).

When Essbase/400 is installed, the configuration file is created as an empty source physical file unless it already exists. If the file exists, it is converted to a source physical file, maintaining the settings it contains.

The configuration file contains keywords and values that define characteristics about the Essbase/400 OLAP Server. The *Hyperion Essbase Technical Reference*, accessible by choosing Documentation from the ShowCase STRATEGY menu, provides specific information about each keyword and value that can be used in the configuration file. The configuration file can contain the same keywords and values used by Essbase on other platforms, but keywords and values not supported by Essbase/400 OLAP Server are ignored.

If the configuration file is deleted, use the following command to re-create it:

```
CRTSRCPF FILE(essbaselib/CFG) RCDLEN(240) MBR(ESSBASE)
ACCPTHSIZ(*MAX4GB) SIZE(*NOMAX) CCSID(37)
```

where *essbaselib* is the Essbase/400 installation library.

Using Configuration Files from a 5250 Display Session

Essbase/400 uses files in the *QSYS.LIB* file system to store information that an administrator or user may want to view or edit using a 5250 display session. The following sections provide iSeries information for displaying or editing the configuration files from a 5250 display session.

Editing or Displaying the ESSBASE.CFG File

To display or edit the *ESSBASE.CFG* file from a 5250 display session, use the iSeries EDTF command. Use the following syntax:

```
EDTF FILE(essbaselib/CFG) MBR(ESSBASE)
```

where *essbaselib* is the Essbase installation library.

Editing or Displaying the ESSBASE.LOG File

To edit or display the *ESSBASE.LOG* file from a 5250 display session, use the iSeries EDTF command:

```
EDTF STMF('/essbaselib/ESSBASE.LOG')
```

where *essbaselib* is the Essbase installation library.

Using Essbase/400 Environment Settings

Each Essbase/400 installation library has unique settings. Use parameters to tailor the Essbase/400 environment. Some parameters are configured during installation. Others are configured only when you need them. The environment parameters are implemented with OS/400 environment variables. Each Essbase/400 installation has its own set of environment variables.

Two Essbase commands allow you to work with Essbase/400 environment variables:

CHGESSENV allows you to change the standard Essbase/400 environment variables and additional generic Essbase environment variables.

DSPESSENV displays the currently configured Essbase environment.

See the following sections for details about configuring, displaying, and removing environment parameters.

Understanding When the Environment Is Used

The Essbase environment is activated when Essbase/400, MaxL/400, or ESSCMD/400 is started. Even though the Essbase environment can be changed while Essbase operations are active, the changes are not active until all Essbase operations are ended and restarted.

Configuring Environment Variables

A standard Essbase environment is configured at installation time. After installation, it can be changed with the CHGESSENV command. Table 1 lists the standard environment variables and their associated command parameters.

Table 1. Standard Essbase environment parameters

Command Parameter	Environment Variable
LIB	ARBORLIB
PATH	ARBORPATH
SERIALNUM	ESSBASE_SERIAL
IPADDR	ESSBASE_IP_ADDR
THREADS	SERVER_THREADS

Table 1. Standard Essbase environment parameters (continued)

Command Parameter	Environment Variable
EBCCCSID	ESSBASE_EBCDIC_CCSID
ASCCCSID	ESSBASE_ASCII_CCSID
SCRSYNTAX	N/A
ALLOBJ	Run with all object authority
CLASSPATH	CLASSPATH

The generic variables in Table 2 can also configure Essbase/400 operations.

Table 2. Generic Essbase environment variables

Environment Variable	Description
ESSBASE_UPPERCASE	When this variable is defined with a non-NULL value, all text output that Essbase generates in messages and log files is displayed in uppercase English. This environment variable is most effective using Essbase on systems outside the United States that do not support lowercase English characters.
LINES	Specifies the number of lines to be displayed by the Help command in the ESSCMD interface.
ESSBASE_JOURNAL	Specifies the library and name of a journal where Essbase will log debug information. Developers may use this information to investigate problems. **Note:** Use this variable only at the direction of technical support.

Displaying Environment Variables

To display the currently configured environment, run the DSPESSENV command. This command displays all of the defined environment variables and their values.

Removing Environment Variables

The standard Essbase environment variables cannot be removed. Once set, they can be changed only to another value. The generic variables can be removed by specifying *REMOVE as the value on the CHGESSENV command.

Understanding the Configuration and Environment Relationship

Because Essbase/400 configures operating characteristics in two ways, it is important to understand how the characteristics relate to one another.

- The Essbase configuration file allows characteristics common among all Essbase platforms to be defined.

- The Essbase environment settings define characteristics that are unique to Essbase/400, characteristics that have different value ranges, or Essbase/400 defaults.

Server threads can be configured in both places. If a value is configured in both places, the environment setting takes precedence over the configuration file setting. To reduce confusion, it is recommended that server threads be configured using only the environment settings.

Configuring Essbase/400 for Java

Essbase/400 supports Java-related functionality. See the *Essbase/400 Installation Guide* for any Essbase/400 Java requirements. The `JVMModuleLocation` parameter, which is set in the *ESSBASE.CFG* file, is ignored in Essbase/400. Instead, a standard Essbase/400 environment variable called `CLASSPATH` has been created. `CLASSPATH` specifies the location of the Java files needed to run with Essbase/400. Essbase/400 will always seek the `CLASSPATH` default value. Do not change the default value. Any changes must be appended or Java will not work.

The `CLASSPATH` default value is:

```
\arborpath\java:
\arborpath\java\essbase.jar:
\arborpath\java\maxljni.jar:
\arborpath\java\jaxp.jar:
\arborpath\java\parser.jar:
\arborpath\java\hisapi.jar:
\arborpath\java\impexp.class:
```

Using Essbase/400 National Language Support

Essbase/400 interacts with client PC workstations using an ASCII CCSID to represent character data. To represent character data with DB2/400 and OS/400, Essbase/400 uses EBCDIC CCSID. Defaults for both CCSID values are calculated at installation time based on your iSeries system settings. To view and/or change the EBCDIC and ASCII CCSIDs for Essbase/400 Server, see the following sections.

Migrating to the Euro CCSID

If you need support for the Euro CCSID, you must change the EBCDIC CCSID. See the following table for easy conversions. If the conversion you need is not in the table, see “Changing ASCII and EBCDIC CCSIDs” on page 17.

Table 3. Euro CCSIDs

Country, region or language group	EBCDIC CCSID	Euro CCSID
United States/Canada/Brazil	37	1140
Germany/Austria	273	1141

Table 3. Euro CCSIDs (continued)

Country, region or language group	EBCDIC CCSID	Euro CCSID
Denmark/Norway	277	1142
Finland/Sweden	278	1143
Italy	280	1144
Spain/Latin America	284	1145
United Kingdom	285	1146
France	297	1147
Multinational #5	500	1148
Iceland	871	1149

To change the CCSID:

1. Type the iSeries command:
   ```
   essbaselib/CHGESSENV EBCCCSID(euro CCSID)
   ```
   where *essbaselib* is the name of the Essbase/400 installation library and *euro CCSID* is the newEuro CCSID.
2. Press Enter.
3. Stop and restart the server by following the instructions in Chapter 1, “Starting and Stopping the Servers” on page 1.

Displaying the Current EBCDIC CCSID

1. To display the current EBCDIC CCSID value, type the iSeries command:
   ```
   essbaselib/DSPESSENV INFO (*ALL)
   ```
   where *essbaselib* is the name of the Essbase/400 installation library.
2. Press Enter.
3. Locate the `ESSBASE_EBCDIC_CCSID` variable. The value displayed is the current EBCDIC CCSID.
4. To display help text, set the cursor on the Essbase/400 Server Attributes message and press F1.

Displaying the Current ASCII CCSID

1. To display the current ASCII CCSID value, type the iSeries command:
   ```
   essbaselib/DSPESSENV INFO (*ALL)
   ```
   where *essbaselib* is the name of the Essbase/400 installation library.
2. Press Enter.
3. Locate the `ESSBASE_ASCII_CCSID` variable. The value displayed is the current ASCII CCSID.

4. To display help text, set the cursor on the Essbase/400 Server Attributes message and press F1.

Changing ASCII and EBCDIC CCSIDs

When changing Essbase/400 CCSID values, consider the potential impact on your Essbase/400 environment. You may have to re-create part of your Essbase/400 environment because certain characters may be interpreted differently once the change is complete.

In most environments, an EBCDIC CCSID change will not impact the interpretation of character data stored in Essbase/400 OLAP Server. If the ASCII CCSID is changed and only invariant (universal codes among CCSIDs) alphabetic characters are used in Essbase/400, the change will have little or no impact. However, if the ASCII CCSID is changed and language-specific characters are used in Essbase/400, you must re-create some information.

Before attempting a change, examine the differences between the CCSIDs that you are considering changing and the importance of the differences. Consult the IBM manual *iSeries National Language Support* to assist in your research. If you still believe a CCSID change is necessary, follow the steps below.

To change CCSIDs:

1. Save outlines, rules files, calculation scripts, and report scripts to the client PC.
2. Type the iSeries command:

   ```
   essbaselib/CHGESSENV ASCCCSID(ccsid1) EBCCCSID(ccsid2)
   ```

   where:

essbaselib	is the name of the Essbase/400 installation library
ccsid1	is the new ASCII CCSID
ccsid2	is the new EBCDIC CCSID

3. Verify that the change is correct.
4. Reload all outlines, rules files, calculation scripts, and report scripts from the client PC to Essbase/400 Server.
5. Import any Essbase/400 data that was interpreted differently as a result of the CCSID change.

Administering Essbase/400

iSeries users can administer the Essbase/400 OLAP Server from a command line using MaxL/400. Traditionally, ESSCMD/400 has provided functionality that is consistent with the PC ESSCMD (ESSCMD and ESSCMDW), but MaxL is positioned as the next generation of ESSCMD.

While ESSCMD/400 and the ESSCMD API continue to be supported, many Essbase features are accessible only through the MaxL/400 interface. Thus, we recommend using MaxL/400 to automate and maintain your Essbase/400 OLAP Server. This documentation explains how to use MaxL/400 with Essbase/400 and how to use MaxL/400 with existing ESSCMD/400 and ESSCMD API automation. Since Essbase/400 fully supports the MaxL language, see the *Hyperion MaxL User's Guide* for more information.

MaxL/400 and ESSCMD/400 use the iSeries standard input, output, and error devices to retrieve input, write output, and write errors, respectively. MaxL/400 uses IFS files exclusively for its input/output files. ESSCMD/400 uses *QSYS.LIB* physical or source physical files for input/output files. Using parameters from the respective commands, these files can be switched from using the device interface to using a file on the iSeries. Thus, an administrator can script and run a sequence of Essbase commands. From this point forward, we will refer to running MaxL/400 or ESSCMD/400 commands with an input file as the *scripted interface,* and running commands without an input file as the *nonscripted interface* or *command line*.

An alternative to ESSCMD/400 is the ESSCMD API Toolkit, which provides OS/400 command interfaces and OS/400 error handling for all ESSCMD/400 operations. For more information, see “Using the ESSCMD API Toolkit” on page 24.

Command Interfaces

You can run commands in script mode, interactive mode, or command mode.

Script mode
Use a scripting language, such as CL, along with commands to automate processing. No shell is invoked, and commands are processed from an input file.

Interactive mode
Enter statements and receive responses on the MaxL/400 or ESSCMD/400 shell or command line. This mode is effective for day-to-day administrative tasks, error/syntax checking, determining status, or checking information on the fly. In ESSCMD/400, interactive and command modes are very similar.

Command mode
Pass MaxL or ESSCMD statements to the respective interpreter and eventually to Essbase/400. Use the respective command with parameters to pass the statement and use iSeries commands to specify the statement, the output file, and the escape message level.

Using an Input Script File

To use the script interface, place the commands in an input file and specify an output or error file, depending on the script's intent. If you are running a short script or testing a script, you can run the script interactively while directing the output and error information to the workstation. Thus, the script is properly coded and displays what the script did or did not do. While running scripts in this manner is useful for testing, script information is most often directed to a file. This allows you to start scripts, let them run unattended, and verify later that they finished correctly. When an output or error file is specified, the output or error information that is normally displayed on the screen is written to the specified file.

A script file may contain ESSCMD/400 and MaxL/400 commands. The ESSCMD/400 and/or MaxL/400 session must be initiated with the `LOGIN` command.

MaxL/400 Scripts (ESSMSH Command)

All filenames referenced in the `ESSMSH` parameters must be in IFS format. The input and output files are IFS files and in ASCII format. Create the input files in the IFS before specifying them in the command.

The syntax for running a script file with the MaxL/400 shell interpreter specifies an input, output, and error file:

```
essbaselib/ESSMSH INFILE(infile)OUTFILE(outfile) ERRFILE(errfile)
```

where:

essbaselib is the library in which Essbase/400 is installed

infile is the input script file that contains the input script file member

outfile is the file that contains the output file member

errfile is the file that contains the error file member

Following is an example script file containing MaxL/400 commands:

```
login user1 password on 172.20.1.54;
display application all;
display user all;
drop user mea;
create user mea identified by password member of group '!supervisors' comment 'delete';
display user all;
msh '/essm40ga/drop.mxl' larry;
msh '/essm40ga/register.mxl' larry;
export database larry.datload all data to data_file '/essm40ga/exp.txt';
display partition all;
display disk volume all;
display macro larry;
display function larry;
display system;
display filter row all;
logout;
exit;
```

ESSCMD/400 Scripts (STRESSCMD)

To create ESSCMD/400 script files, create an iSeries *QSYS.LIB* file and member to contain the Essbase commands. The file can be created as a physical or source physical file. Set the record size between 80 and 240 characters to allow easy iSeries-based editing and viewing. Once the file and member are created, update the file to contain the Essbase commands that ESSCMD/400 will run.

The syntax for running ESSCMD/400 with an input, output, and error file is:

```
essbaselib/STRESSCMD INFILE(inlibrary/infile) INMBR(inmember)
OUTFILE(outlibrary/outfile) OUTMBR(outmember)
ERRFILE(errlibrary/errfile) ERRMBR(errmember)
```

where:

essbaselib is the library in which Essbase/400 is installed

inlibrary is the library in which the input script file is located

infile is the input script file that contains the input script file member

inmember is the input script file member that contains valid Essbase/400 commands

outlibrary is the library in which the output file is located

outfile is the file that contains the output file member

outmember is the output member where ESSCMD/400 screen output is written

errlibrary is the library in which the error file is located

errfile is the file that contains the error file member

errmember is the error file member where ESSCMD/400 error information is written

Specifying ESSCMD/400 Output and Error Files in ESSCMD/400 Script Files

When creating ESSCMD/400 scripts, several ESSCMD commands let you specify input, output, and error files for a given command. If a command that you are running requires a filename as a parameter, enter the filename in the syntax that Essbase understands.

However, if you specify a file in the *QSYS.LIB* file system as shown in the example, the library and file must exist before running ESSCMD/400. ESSCMD/400 does not create the library and file. ESSCMD/400 will create the specified member if it does not exist. However, if the member does not exist, verify that the MAXMBRS setting on the file will allow the creation of a new member. If the MAXMBRS limit is reached, the member cannot be created.

The files can be created as physical or source physical files. We recommended a record size between 80 and 240 characters for easy iSeries-based editing and viewing. If the ESSCMD/400 output file is not created in advance, the file is created as a source physical file with a data field of 254 characters. The size of the data field makes it difficult to view or edit the output. If the record size is not large enough to store the ESSCMD/400 output record, the output record is truncated. Verify that the file is created with a record size large enough to accommodate the error messages and any failing input records to be logged to the file.

For example, when using the `BUILDDIM` or `IMPORT` commands, you can specify a filename to use for recording errors found during the operation. You must enter an IFS path name. You may store the file in the root file system or the *QSYS.LIB* file system.

The following example stores the information in a library named `MYLIB`, a file named *MYFILE*, and a member named `MYMEMBER` in the QSYS file system:

```
/qsys.lib/MYLIB.lib/MYFILE.file/MYMEMBER.mbr
```

The following example stores the information in the *DIMBUILD.ERR* file in the Essbase directory for the Sample Basic application database of the root file system:

```
/ESSBASE/app/Sample/Basic/dimbuild.err
```

When a root file system path is specified, the directories must exist before running ESSCMD/400. The file will automatically be created if it does not exist. If the file exists, it is overwritten.

Using the Command Line Interface

To run commands from a command line interface, issue the respective commands from an OS/400 command line. Do not specify any parameters. The respective command interpreter will start.

MaxL/400 (RUNMAXLSTM)

Issue `RUNMAXLSTM` from an OS/400 command line. F1 command help explains the possible values for each parameter. Following is the syntax:

```
essbaselib/RUNMAXLSTM COMMAND(command)
OUTPUT(outfile) ESCLEVEL(esclevel)
```

where:

essbaselib	is the library in which Essbase/400 is installed
command	is the MaxL/400 command
outfile	is the file that contains the output file

esclevel specifies the level of the messages that are returned. F1 Help lists the parameter values.

ESSCMD/400 (STRESSCMD)

Issue the `STRESSCMD` command from an OS/400 command line. Do not specify any parameters. The ESSCMD/400 interpreter shell starts.

ESSCMD/400 File Differences

ESSCMD/400 provides nearly identical support for the PC ESSCMD commands. Although the interface that ESSCMD/400 provides is very similar to the interfaces provided by ESSCMD utilities on other platforms, there are differences in coding an ESSCMD/400 script file.

Creating ESSCMD/400 Output and Error Files

Before creating or viewing a QSYS file that holds error records, determine the EBCDIC server CCSID that Essbase/400 is configured to use:

1. Find the server CCSID. Run the `DSPESSENV` command to determine the value of the `ESSBASE_EBCDIC_CCSID` variable. Place the cursor on the Essbase/400 Server Attributes message and press F1.
2. Create the source file in QSYS with the following command:

   ```
   CRTSRCPF FILE(mylib/filename) RCDLEN(length) CCSID(value)
   ```

   where:

   mylib is the QSYS library name

   filename is the QSYS filename

   length is the record length. (Specify a length long enough to handle the longest rejected record. Note that tab characters are expanded to spaces at a ratio of eight spaces per tab character.)

   value is the single-byte CCSID that Essbase/400 uses. (On servers using double-byte character sets, *value* must be the single-byte CCSID that is associated with the mixed CCSID of Essbase/400 Server.)

Viewing Output and Error Files

If you wrote the error file to the root file system and want to view the error file, use the iSeries command `DSPF`. For example:

```
DSPF STMF('/path/filename')
```

where:

path is the location of the file

filename is the name of the output or error file to view

If you wrote the error file to the *QSYS.LIB* file system and want to view the error file, use the iSeries command DSPF. For example:

```
DSPF FILE(mylib/filename) MBR ESSBASE
```

where:

mylib is the QSYS library name.

filename is the name of the output or error file to view.

Output Command

ESSCMD/400 The OUTPUT command directs process information output from the session to an iSeries file.

```
OUTPUT lib/file(member)
```

where *lib/file(member)* is the library and filename of the output file.

MaxL/400 The OUTPUT command directs process information output from the session to a text file.

```
OUTPUT ('/path/file.ext')
```

where *'/path/file.ext'* is the path and filename of the output text file.

Reloading Rejected Records

You can specify an error file that receives rejected records if a dimension build or data load error occurs. This file can reside in either the root file system or the *QSYS.LIB* file system. If you want to use the error file to reload rejected records, direct it to a stream file in the root file system. If you want to display only the error file, store the file in the QSYS file system. In this situation, tabs in the file are expanded to spaces, making the file easier to read. If you direct the error file to the root file system, you can use the CPYFRMSTMF command to copy the error file to the QSYS file system to read:

1. Create a source physical file using the CRTSRCPF command described previously.
2. Copy the file from the root file system to the QSYS file system using the command:

   ```
   CPYFRMSTMF FROMSTMF('file name') TOMBR('/qsys.lib/mylib.lib/pf.file/member.mbr')
      TABEXPN(*NO)
   ```

   where:

   file name is the full path and name of the file in the root file system in which the error records were originally logged (for example, */essbase/app/smstbc/dbcs/dimerr.txt*)

   mylib is the QSYS library name

pf is the QSYS filename

member is the name of the member to create in the QSYS file system (for example, *dimerr.mbr*)

Using the ESSCMD API Toolkit

The ESSCMD API Toolkit provides OS/400 command interfaces and OS/400 error handling for all of the ESSCMD/400 operations. The ESSCMD API Toolkit is automatically enabled and sits atop the ESSCMD/400 function, transforming the script-based interface of ESSCMD/400 into an OS/400 command interface.

Use the ESSCMD API Toolkit to open and maintain an ESSCMD/400 session across multiple OS/400 commands. You can intersperse ESSCMD/400 commands with other OS/400 commands. You can also code your CL program to respond to errors reported by Essbase on a command-by-command basis, using standard OS/400 error- and message-handling mechanisms.

The *Hyperion Essbase Technical Reference*, accessible by choosing `Documentation` from the ShowCase STRATEGY menu, includes ESSCMD API Toolkit information. For details about individual ESSCMD API commands, see the iSeries online Help. To use the Help, enter an ESSCMD API command, press F4, and then press F1 for help on a parameter.

Command Structure

The ESSCMD API Toolkit consists of four base OS/400 commands and many supplemental commands that are discussed later. OS/400 command-level help is available for each of the base and supplemental commands. The four base commands are:

- `LOGINESS`
- `RUNESSCMDC`
- `LOGOUTESS`
- `RTVESSSTS`

Note: The `STRESSCMD` command is not an ESSCMD API command. `STRESSCMD` acts independently from the ESSCMD API commands.

LOGINESS Command

Issue the `LOGINESS` command before running other ESSCMD API commands. After issuing the `LOGINESS` command, `RUNESSCMDC` (or a supplemental command) interacts with Essbase/400 Server under the session opened by the `LOGINESS` command. Multiple logons with `LOGINESS` are not allowed.

RUNESSCMDC Command

RUNESSCMDC lets you run any ESSCMD command directly from the OS/400 command line, or from within a CL or other program. With RUNESSCMDC, type the underlying ESSCMD/400 command as follows:

```
RUNESSCMDC COMMAND('UNLOADAPP MYAPP')
```

Typically, you will use the LOGINESS command to initiate the session before using RUNESSCMDC.

LOGOUTESS Command

LOGOUTESS ends the connection and closes the ESSCMD API session by issuing the ESSCMD EXIT command within the regular ESSCMD interface.

RTVESSSTS Command

From within a CL program, RTVESSSTS can retrieve information about a previous ESSCMD API Toolkit command if the session associated with that command has not been closed.

There are three pieces of information that can be retrieved:

- Essbase message number
- Essbase message severity
- Essbase message text

See the online command Help for more information about the data types of these variables and the possible return values.

Auto-logoff Considerations

When you use ESSCMD API commands within another program, consider how Essbase's Auto-logoff feature will affect the program. Essbase is installed with a default 60-minute auto-logoff time, but this may be reduced. This may be a problem in a program such as the following:

```
LOGIN SVRUSR(userid) SVRPW(password)
ENDESSAPP APPNAME(mycube)
SAVLIB  DEV(TAP01) ...
STRESSAPP APPNAME(mycube)
```

The SAVLIB operation may take a long time, particularly if loading or changing a tape is required to complete the operation. If the time between the completion of the ENDESSAPP and start of the STRESSAPP command is longer than the auto-logoff setting, the STRESSAPP will fail.

Unfortunately, the auto-logoff setting applies to all users, so you cannot adjust it for the user ID used in the program. If you prefer not to change the time-out for all Essbase users (it can be set to no auto-logoff), run LOGOUTESS and LOGINESS again wherever a problem might occur. For example:

```
LOGIN SVRUSR(userid) SVRPW(password)
ENDESSAPP APPNAME(mycube)
LOGOUTESS
SAVLIB  DEV(TAP01) ...
LOGINESS userid passwordSTRESSAPP APPNAME(mycube)
```

Supplemental Commands

In addition to the commands described previously, the ESSCMD API Toolkit provides alternative commands to RUNESSCMDC. There's no need to remember the ESSCMD command syntax. The underlying ESSCMD syntax is mapped to OS/400 command keywords.

Each supplemental command corresponds to an ESSCMD/400 command. For example, the ESSCMD API command STRESSAPP corresponds to the ESSCMD/400 LOADAPP command. These OS/400 commands have the look and feel of standard OS/400 commands, complete with prompted keywords, help, and so on. For ESSCMD/400 commands with no corresponding ESSCMD API command, use RUNESSCMDC.

With ESSCMD/400, choose an application and database using the SELECT command so that subsequent commands (that do not specify an application or database) can operate upon the selected database. The ESSCMD API supplemental commands require that you specify the application and database on the command itself. The appropriate SELECT command is issued implicitly for you.

Most supplemental commands let you use the full power of the underlying ESSCMD/400 commands. This list contains all the Essbase ESSCMD API supplemental commands and indicates the corresponding ESSCMD command issued. For more information about each ESSCMD command, see the ESSCMD documentation in the *Hyperion Essbase Technical Reference*, accessible by choosing **Documentation** from the ShowCase STRATEGY menu.

Note: The IMPESSSQL and BLDESSDIM commands do not permit all forms of their respective ESSCMD commands to be issued. In each case, only the SQL form with a server-based rule file is supported, since this is the most common usage. To handle other cases, use RUNESSCMDC.

Table 4. ESSCMD API supplemental commands

ESSCMD/400 API Command	ESSCMD/400 Command
ADDESSGRPU	ADDUSER
BLDESSDIM	BUILDDIM
CLCESSDFT	CALCDEFAULT
CPYESSAPP	COPYAPP
CPYESSDB	COPYDB

Table 4. ESSCMD API supplemental commands (continued)

ESSCMD/400 API Command	ESSCMD/400 Command
CRTESSAPP	CREATEAPP
CRTESSDB	CREATEDB
CRTESSGRP	CREATEGROUP
CRTESSUSR	CREATEUSER
CRTESSVAR	CREATEVARIABLE
DISESSLGN	DISABLELOGIN
DLTESSAPP	DELETEAPP
DLTESSDB	DELETEDB
DLTESSGRP	DELETEGROUP
DLTESSUSR	DELETEUSER
DLTESSVAR	DELETEVARIABLE
ENBESSLGN	ENABLELOGIN
GETESSAPP	GETAPPSTATE
IMPESSSQL	IMPORT
STRESSAPP	LOADAPP
LOGINESS	LOGIN
LOGOUTESS	EXIT
LSTESSAPP	LISTAPP
LSTESSDB	LISTDB
LSTESSGRPS	LISTGROUPS
LSTESSGRPU	LISTGROUPUSERS
LSTESSUSR	LISTUSERS
LSTESSVAR	LISTVARIABLES
CLRESSDB	RESETDB
RNMESSAPP	RENAMEAPP
RNMESSDB	RENAMEDB
RNMESSUSR	RENAMEUSER
RMVESSGRPU	REMOVEUSER
RUNESSCLC	RUNCALC
ENDESSSVR	SHUTDOWNSERVER
ENDESSAPP	UNLOADAPP
UPDESSVAR	UPDATEVARIABLE

Output Processing

RUNESSCMDC and other ESSCMD/400 API commands each have an OUTFILE keyword. This keyword follows normal OS/400 OUTFILE conventions. If the file does not exist, it is created with a record length of 240 characters.

Controlling Error Handling When Programming

The Essbase/400 messaging mechanism defines five message severity levels:

- Informational
- Warning
- Error
- Serious
- Fatal

If Essbase returns only *Informational* and/or *Warning* messages, by default an ESSCMD/400 API command will end with a normal OS/400 completion message. Conversely, if Essbase returns an *Error*, *Serious*, or *Fatal* message, the ESSCMD/400 API command issues an OS/400 Escape message.

To override this behavior, use the optional ESCLEVEL keyword on most ESSCMD/400 API commands. For example, you could use the following sequence of commands to delete an existing application and then re-create it by copying another application:

```
LOGINESS myuserid mypasswordDLTESSAPP myapp ESCLEVEL(*SERIOUS)
CPYESSAPP yourapp myapp ESCLEVEL(*ALL)
LOGOUTESS
```

The DLTESSAPP uses ESCLEVEL(*SERIOUS) so the sometimes expected error message doesn't cause the program to fail:

```
(1051030) ERROR: Application myapp does not exist.
```

Note: In the previous example, the Essbase ERROR level message will behave the same way, resulting in an OS/400 Completion message. Serious and Fatal messages will still result in an OS/400 Escape message.

In the previous ESCLEVEL example, the CPYESSAPP command uses ESCLEVEL(*ALL) because the copy should always work after successfully deleting the application. CPYESSAPP doesn't return any Essbase messages, not even an Informational one. By using *ALL, the program will fail if anything other than what is expected (no messages) occurs. See the online Help for more information about the ESCLEVEL keyword.

In addition to the ESCLEVEL keyword, you can use the RTVESSSTS command within your program to retrieve status information about the most recent Essbase message issued. This lets you programmatically retrieve the level of the message (for example, Informational, Warning), the Essbase message number, and the message text. For example, the following message:

```
(1051030) ERROR: Application myapp does not exist.
```

would return 1051030 for the STATUS (message number), ″ERROR″ for the MSGLEVEL, and ″Application myapp does not exist.″ for the MESSAGE.

See the online Help for RTVESSSTS for more information about the types and sizes of the keywords for RTVESSSTS.

Debugging ESSCMD API

It’s possible to see the commands that the ESSCMD/400 API passes to the ESSCMD/400 engine. To log each ESSCMD/400 command, issue the following command in the job ESSCMD/400 is running under. This command will also log the password.

```
ADDENVVAR ENVVAR(ESSBASE_CMDAPI_LOG) VALUE('your_library/your_file')
```

Once the environment variable is set, turn off logging with the next command:

```
CHGENVVAR ENVVAR(ESSBASE_CMDAPI_LOG) VALUE(*NULL)
```

Logging can be turned on and off again with the CHGENVVAR command.

You can use all the standard OS/400 built-in debugging support for your CL program, RPG program, or other program that uses the ESSCMD/400 API.

To more easily debug CL programs that use the ESSCMD API, you may want to create your CL program to log CL Program Commands:

```
CRTCLPGM ... LOG(*YES)
```

In addition, change your job to default to not log CL Program Commands by issuing:

```
CHGJOB LOGCLPGM(*NO)
```

The combination of these changes will result in a job log (via DSPJOBLOG) that shows only the commands issued directly by your CL program and not those issued internally by product software.

Maintaining Essbase/400 Log Files

While Essbase/400 is running, entries are added to various log files. These files reside in the IFS root file system in the application directories. Over time, these files can grow very large. If the logs are not properly maintained, they may fill up or consume an unnecessary amount of storage space.

To control the size of the Essbase/400 log file, use the `OS400DAYSLOGRETAIN` parameter in the Essbase/400 configuration file. For information about how to edit the configuration file, see "Understanding the Essbase/400 Configuration" on page 11. When starting an Essbase/400 server or an Essbase/400 application, the server uses `OS400DAYSLOGRETAIN` to determine how many days of log records to keep. Deleted records are not recoverable.

Table 5. OS400DAYSLOGRETAIN value

Value	Results
None or Invalid	Retain all records
0	Retain records from current session
1	Retain records logged in last 24 hours
2 and greater	Retain records from specified number of days

You can also manually clear the server and application logs. In Essbase Application Manager, from the Server menu, choose **Delete Event Log**, or from the Application menu, choose **Delete Event Log**.

Index Cache Size

The index cache size of an Essbase/400 database depends on the size of the database index page. For every 1 kilobyte (KB) of index page size, you can specify up to 125 megabytes (MB) for the index cache size. The following table shows the maximum sizes allowed for the index cache, depending on the size of the index page.

Table 6. Maximum sizes for the index cache

Index Page Size	Maximum Index Cache Size
1KB (1024B)	125MB (131072000B)
2KB (2048B)	250MB (262144000B)
4KB (4096B)	500MB (524288000B)
8KB (8192B)	1000MB (1048576000B)

If the index cache size limit is exceeded, Essbase/400 will reset the cache size to the maximum size allowed for the index page size. For example, if you specify a cache size of 1000MB but your index page size is 2KB, Essbase/400 will reset the cache size to 250MB.

Note: Data file cache is not used. Therefore, a value of 0 is correct.

Loading DB2/400 Data into Essbase/400

Essbase/400 data load and dimension build operations are most efficient by using the SQL interface support. The SQL interface support is shipped as part of the base Essbase/400 package; you do not need to install a separate SQL interface module.

Setting Up Your iSeries to Allow SQL Loads

1. Type `WRKRDBDIRE` and press Enter.
2. If you do not see an entry for `*LOCAL`, use option 1 to add a `*LOCAL` entry.

Loading Data Using the SQL Interface Support

1. Using the Essbase Application Manager, go to a Data Prep Editor window, and from the File menu, choose **Open SQL**.
2. Select your application and database in the Select Server, Application and Database dialog box and click **OK**.
3. In the SQL Data Sources selection box of the Define SQL dialog box, the `*LOCAL` Relational Database Directory entry is highlighted. The Connect information should *not* be filled in.

   Note: The `*LOCAL` entry allows your rules files to be portable between Essbase/400 servers on different systems. You can also select the name of your system from the list, but your rules file must always be used only on that system.
4. Use the Select information to specify the SQL statement to load data into the Data Prep Editor. In the From box, enter the names of the tables in the SQL naming convention, as opposed to the iSeries naming convention. In the SQL naming convention, a period (.) is used between a library and filename. For example:

   ```
   library.filename
   ```

   You cannot specify a specific OS/400 database file member name in this interface.
5. If you want to add `GROUP BY` or `ORDER BY` clauses, you can do so in the Where box. However, you must first specify at least one `WHERE` condition. If you do not have a `WHERE` condition to specify, simply use `1 = 1` as a placeholder `WHERE` condition in the Where box. For example: `1 = 1 ORDER BY Product`.
6. Click **OK/Retrieve**.

Substitution Variables in Rule File SQL Statements

Essbase substitution variables can be used in calc scripts, report scripts, and spreadsheets. With Essbase/400, you can also use substitution variables within the SQL statement specified in a Data Load or Dimension Build rule file. This allows easy reuse of rule files for multiple data files of the same format or for retargeting loads for test versus production libraries. Modify the value of substitution variables as needed, which modifies the behavior of the scripts and rule files that use the variables.

To use substitution variables in the SQL statement, simply specify the variable prefixed with an ampersand (&). If a variable is used in the Where box and the result of the entire WHERE clause after all variables are substituted is all blanks or an empty string, the entire WHERE clause is omitted from the SQL statement.

Note: If a variable reference does not have a value assigned to it, no substitution occurs and the original variable name along with the ampersand are passed in as part of the SQL statement.

Following is an example of the message that is logged if the variable reference is not assigned a value:

```
Local/Appname/Dbname/Userid/Info(1012042) Substitution variable <xy> doesn't exist.
```

Substitution variables can be managed by choosing **Substitution Variables** from the Server menu in Essbase/400 Application Manager, or via the ESSCMD commands: CREATEVARIABLE, UPDATEVARIABLE, and DELETEVARIABLE. See the Hyperion Essbase documentation for more information about substitution variables.

Flat File Support

Many non-iSeries systems support character-delimited files, also known as flat files. Flat files often use a delimiter character, such as a tab or space, to separate individual fields within a record.

Essbase/400 can load flat files stored in the IFS application/database directory with a *.txt* extension. ESSCMD/400 can also read from flat files from IFS. The recommended way to load data is to use a load rule with a SQL statement. To load data from DB2 files, specify the appropriate SQL statement to select the fields that you want to load. You can also load a flat file using the same technique. In this case, the standard OS/400 database file contains a single character field. This field is used to hold character-delimited data.

For example, suppose you have a tab-delimited text file on a PC. You can load data directly into Essbase/400 from this file using ESSCMD on the PC. Alternatively, you can copy this file to the iSeries and load the data using SQL. Following are the steps that you would use:

1. Create a database file on the iSeries using the appropriate record length:
   ```
   CRTPF FILE(MYLIB/FILE) RCDLEN(132)
   ```
2. Using the FTP PUT command, copy the data from the PC to the iSeries:
   ```
   put c:\pcdata.txt   mylib/file.file
   ```

3. The file MYLIB/FILE now contains the tab-delimited data. Using Application Manager, create or edit the load rule to read this data by using the following select statement:

   ```
   SELECT * FROM MYLIB.FILE
   ```

4. From the Options menu, choose **Data File Attributes** and use the File Delimiter tab in the Data File Attributes dialog box to set the appropriate delimiter.

Using Warehouse Builder with Essbase/400

Using Warehouse Builder, you can perform several operations against an Essbase/400 database (for example, clear a database, build dimensions, load a database, calculate a database). When using Warehouse Builder to build dimensions or load data, Essbase/400 uses SQL to read records and load them into the database. Essbase validates each record as it is processed and may reject records if it encounters an error.

Dimension Build and Data Load Rejected Records

The member names identified by each record must match the member names specified in the Essbase/400 outline. If the member names do not match (or some other error occurs), Essbase/400 rejects the record. In this situation, if the Essbase/400 build plan defined in Warehouse Builder is set to Continue on Error, Warehouse Builder logs the rejected record in the JOBMSGS table and processing continues.

The records causing an error are logged with a MSG_ID of 'TXT0208' (in the case of a data load) or 'TXT0207' (in the case of a dimension build). The record itself is logged in the MSG_HELP field. The object ID for the Essbase/400 build plan is stored in the OBJECTID field. The sequence number of the Essbase/400 build plan identifies the Essbase/400 operations within the Essbase/400 build plan and is stored in SEQ_NUMBER.

Reloading Rejected Records

After fixing your Essbase/400 outline, it is possible to reprocess only the rejected records (as opposed to reprocessing all of the data). To reprocess the rejected records, use the Essbase Application Manager to create a dimension build or data load rule that points to the JOBMSGS table. Following is an example SQL statement that reprocesses the rejected records of an Essbase/400 build plan whose object ID is 101 and whose sequence number is 1:

```
SELECT MSG_HELP FROM SCSERVER.JOBMSGS
WHERE MSG_ID = 'TXT0208' AND OBJECTID = 101 AND SEQ_NUMBER =1
```

Note: To reprocess the rejected records for a dimension build, the MSG_ID value is 'TXT0207'.

To determine the correct object ID and sequence number to use, run the queries (*.DBQ) shipped with Warehouse Builder against the JOBMSGS table. The queries are located in the *REPORTS* folder in which Warehouse Builder is installed.

Naming Essbase/400 Server

When using Warehouse Builder with Essbase/400, the value you specify for Essbase/400 OLAP Server in the Build Plan dialog box must be either an IP address in numeric format or a valid (or usable) name on the control server iSeries. To check whether a name is valid, enter the iSeries command CFGTCP from an iSeries session window on the control server iSeries and choose option 10. On the screen that appears, you can verify that the name associated with the TCP/IP address for Essbase/400 Server is the same as the name defined on your PC.

For additional information about using Warehouse Builder with Essbase/400, see the Warehouse Builder online Help.

Security

All Essbase security functions are supported by Essbase/400. However, individual Essbase/400 data files reside in the OS/400 Integrated File System (IFS) and may be accessible via other OS/400 interfaces. You should verify that these files are properly secured with respect to the iSeries defined security mechanisms. In particular, the Essbase security file itself, named *ESSBASE.SEC*, is stored within the IFS file system.

ALLOBJ Authority

With a profile that has *ALLOBJ authority, a user has full iSeries access, including all databases, programs, folders, and files. If you want a user profile to have access to certain iSeries objects but do not want to assign *ALLOBJ authority, you can grant users authority access to certain objects and not to others. For example, a profile at the *USER level can be granted access to library ABC, but not to library XYZ.

Essbase security controls user access. Using a profile that has been assigned the *SECOFR class is the easy way to start the servers. However, to enable a user to run Essbase/400 with anything less than *ALLOBJ authority, follow these steps:

1. Create a user profile with the desired security class to be dedicated to starting Essbase. Make sure that Essbase will always be started with this user profile.
2. Type the command GRTESSAUT to grant the user profile all of the necessary authorities to start Essbase and press Enter.
3. Type CHGESSENV and press Enter.

4. Change the Run with All Object Authority field to `*NO`.
5. To verify if a user profile is authorized to start Essbase, run the `CHKESSAUT` command.

Integrated Security

Essbase/400 provides support for Integrated Security. By using OS/400 security controls, Integrated Security lets you control who has access to Essbase/400. As a result, your security scheme for Essbase/400, Analyzer, and Analyzer for the Web is greatly simplified.

Waning: If you are setting up Integrated Security for Essbase/400, create at least one Essbase superuser profile that is not an OS/400 user, and be sure this user profile does not get deleted or demoted. If an Essbase system supervisor ID is the same as any Essbase/400 ID, you may lose your supervisory status.

Once you set up the Integrated Security environment between Essbase/400 and OS/400, OS/400 user profile operations are applied to Essbase/400 security. Then you can perform most routine Essbase/400 user security changes from OS/400 security interfaces or from Warehouse Manager. You will no longer need to add or remove users or change group memberships via the Essbase/400 security interfaces. However, you will still use Essbase/400 security interfaces to establish which applications and databases a group can access.

If you choose *not* to use the Integrated Security feature, you need not do anything. By default, Essbase/400 runs with its own security system, independent of OS/400 security.

Note: Whether or not Integrated Security is enabled, access to DB2/400 for data loading and dimension building will still require OS/400 authorization information to be entered.

Setting Up Essbase/400 Integrated Security

To set up Essbase/400 Integrated Security, you should be familiar with basic Essbase security and basic OS/400 security, including OS/400's group profile support. This includes experience creating users and groups in Essbase/400 using Application Manager and OS/400 using Warehouse Manager Client.

For detailed information about basic Essbase/400 and OS/400 security, see the *Hyperion Essbase Database Administrator's Guide* and the IBM *OS/400 Security Reference* document. For more information about Warehouse Manager Client, see the Warehouse Manager Client online Help. You should also be familiar with editing the Essbase/400 configuration files (see "Editing or Displaying the ESSBASE.LOG File" on page 13.)

Essbase/400 requires that actual Essbase user IDs be defined for many operations. For example, locks on objects such as outlines, rule files, or calc scripts must be associated with an Essbase/400 user ID. For this reason, the Integrated Security feature of Essbase/400 will automatically create and delete Essbase/400 user IDs in response to OS/400 user profile creations, deletions, and other operations.

Essbase/400 Integrated Security can be configured in one of two ways: you may give *all* OS/400 users access to Essbase/400, or you may give only a *subset* of OS/400 users access to Essbase/400.

To implement Integrated Security, follow these steps:

Step 1. Start Essbase/400 OLAP Server.

Decide which OS/400 user profiles should have access to Essbase. You can give *all* valid OS/400 user profiles (both existing profiles and those created in the future) access to Essbase, or you can give a *subset* of OS/400 users access to Essbase. Regardless of your choice, you will still be able to give non-OS/400 users access to Essbase as well.

To give *all* current and future OS/400 users access to Essbase/400, skip to Step 6. If you want to give only a *subset* of OS/400 users access to Essbase/400, continue with Step 2.

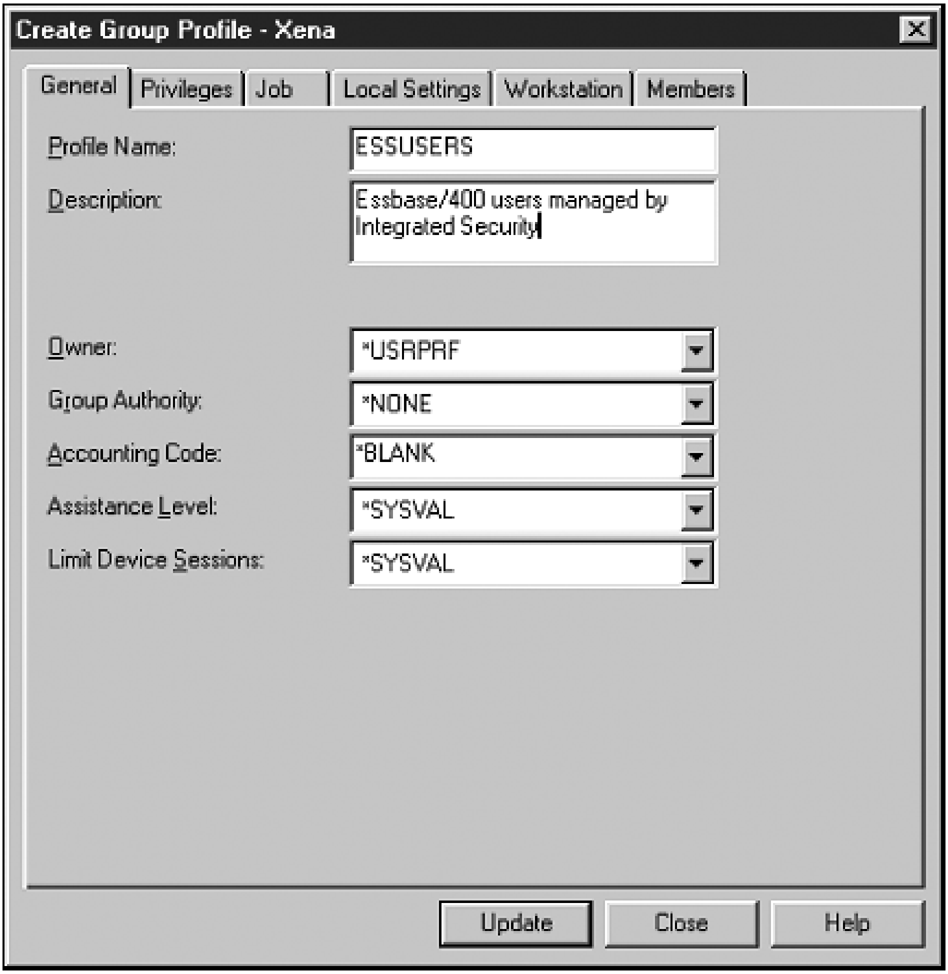
Step 2. First, you'll create an OS/400 group. Open Warehouse Manager Client. To add a Group User Profile, follow these steps:

a. In the Explorer, connect to the iSeries data source that you want to create the profile on.

b. On the Manage menu, choose **Profile**, and then choose **Create Group**. The Create Group Profile dialog box appears.

c. Type a name and description for the profile.

The name for the group profile can be anything. In this guide, we will assume that you are using the name `ESSUSERS`.



Create Group Profile - Xena

Profile Name: ESSUSERS

Description: Essbase/400 users managed by Integrated Security

Owner: *USRPRF

Group Authority: *NONE

Accounting Code: *BLANK

Assistance Level: *SYSVAL

Limit Device Sessions: *SYSVAL

Figure 1. Warehouse Manager Create Group Profile dialog box

d. Define any other settings on the other tabs. When setting up Integrated Security, you can accept the defaults that are displayed on these tabs. No special authorities are required.

e. Click **Update** to update the iSeries and refresh the profiles list in Explorer. Then, click **Close** to close the dialog box.

You have now created a Group User Profile (ESSUSERS) that identifies OS/400 users with access to Essbase/400. Now you will update the Essbase/400 configuration file to identify the OS/400 group (and corresponding Essbase/400 group, to be created later), which will specify that the OS/400 users have Essbase/400 access.

Step 3. Without stopping the server, type the following line in the Essbase configuration file. (For information about editing the configuration file, see "Editing or Displaying the ESSBASE.CFG File" on page 12.)

```
OS400USERSGROUP ESSUSERS
```

where *ESSUSERS* refers to the name of the OS/400 group you just created.

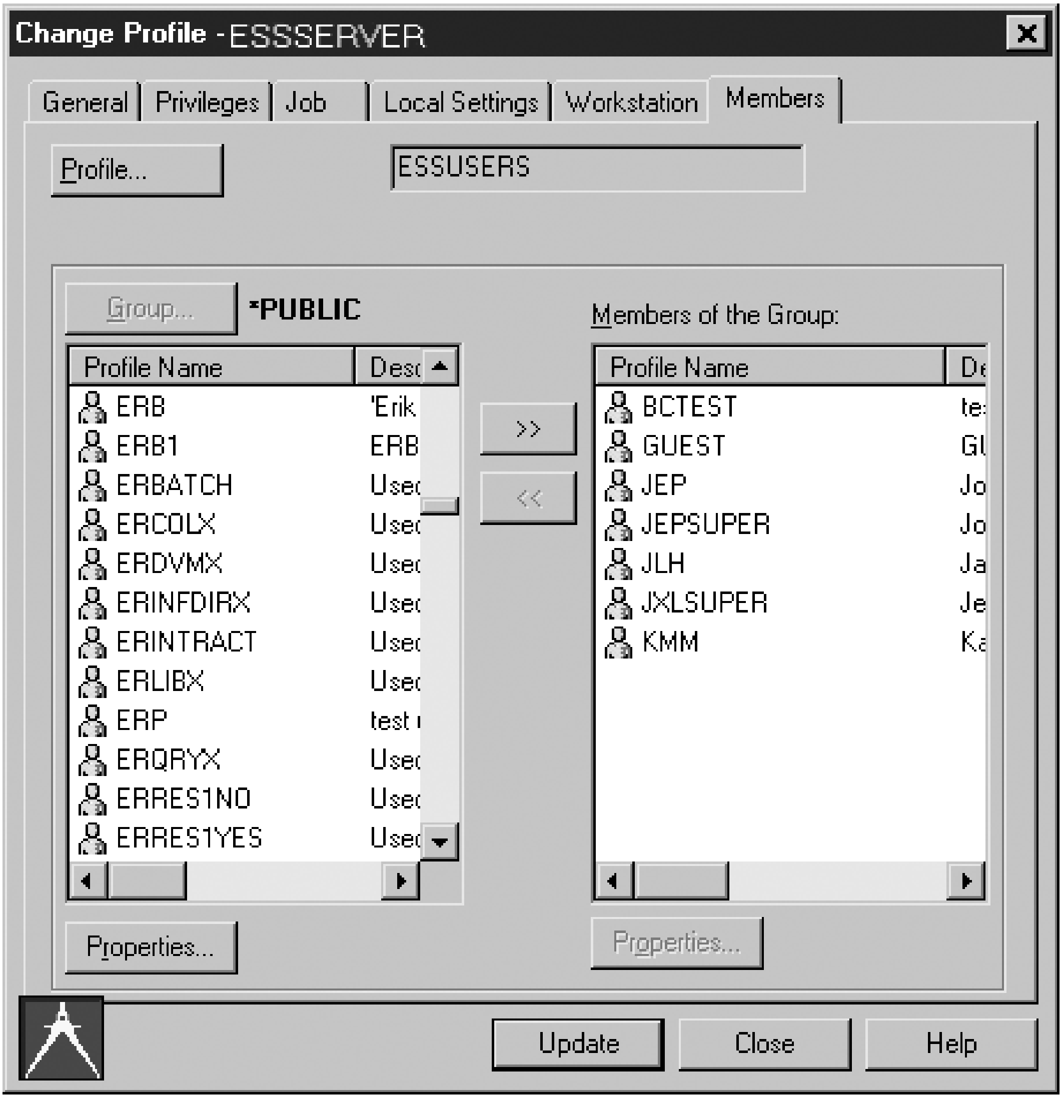
Step 4. You can now add OS/400 users to your new group, `ESSUSERS`. Each OS/400 user profile can be associated with one *primary* Group User Profile and several *supplemental* Group User Profiles. By establishing the `ESSUSERS` group as either the primary or supplemental group for an OS/400 user, you will indicate that the user has access to Essbase/400.

Note: `ESSUSERS` can be either the primary *or* supplemental group for any of your users. For example, it could be the primary group for Joe and the supplemental group for Bob.

Step 5. To associate a Group User Profile with a User Profile, use Warehouse Manager Client to add users to the `ESSUSERS` group. How you do this depends on whether `ESSUSERS` is the primary or secondary user group.

For those users who *do not* have a primary group:

a. Find `ESSUSERS` in the Explorer view. Right-click and choose **Change**.
b. The Create Group Profile dialog box will appear. Click the **Members** tab and add all the necessary members, as in Figure 2.
c. When finished, click **Update**.



ESSUSERS

*PUBLIC

Profile Name	Desc
ERB	'Erik
ERB1	ERB
ERBATCH	Use
ERCOLX	Use
ERDVMX	Use
ERINFDIRX	Use
ERINTRACT	Use
ERLIBX	Use
ERP	test
ERQRYX	Use
ERRES1NO	Use
ERRES1YES	Use

Members of the Group:

Profile Name	De
BCTEST	te
GUEST	GU
JEP	Jo
JEPSUPER	Jo
JLH	Ja
JXLSUPER	Je
KMM	Ka

Figure 2. Warehouse Manager Change Profile dialog box, Members tab

For those users who already have a primary group, you must add the users individually via the Change Profile dialog box:

a. Find the user in the Explorer view. Right-click and choose **Change Profile**.

b. The Change Profile dialog box will appear. Click the **Members** tab.

c. From the list of all groups, choose **ESSUSERS**, and then click the double-arrow button to add the group to the user's list of current supplemental groups.

d. Click **Update**. Repeat this for as many users as necessary.

Step 6. If you have completed steps 2–5, skip to step 7.

You have chosen to allow all valid OS/400 users to access Essbase/400.

To give *all* valid OS/400 users access to Essbase/400, *without* stopping server, add the following line to the Essbase configuration file. (For information about editing the configuration file, see "Editing or Displaying the ESSBASE.CFG File" on page 12.)

```
OS400UsersGroup *ALL ESSUSERS
```

where ESSUSERS is the group name for Essbase/400.

Note: Be sure to choose a group name that is *not* currently in use by Essbase.

Step 7. At this point, you have specified the OS/400 users who access Essbase/400 (either a subset of them, or all of them.) Now, using Essbase Application Manager, create the Essbase/400 group (which was specified in the Essbase/400 configuration file in step 3 or step 5) using the following steps:

a. Open Essbase Application Manager on your PC. Connect to Essbase/400 OLAP Server. From the Security menu, choose **Users/Groups**. The User/Group Security dialog box will appear.

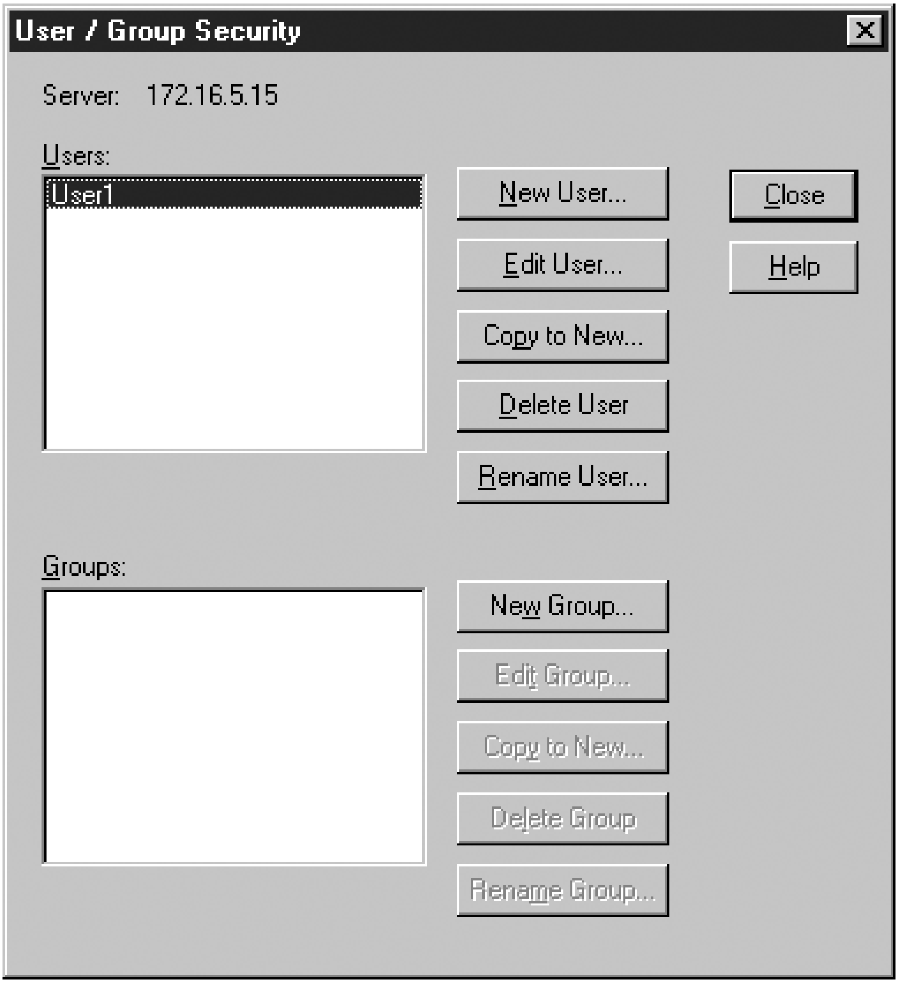
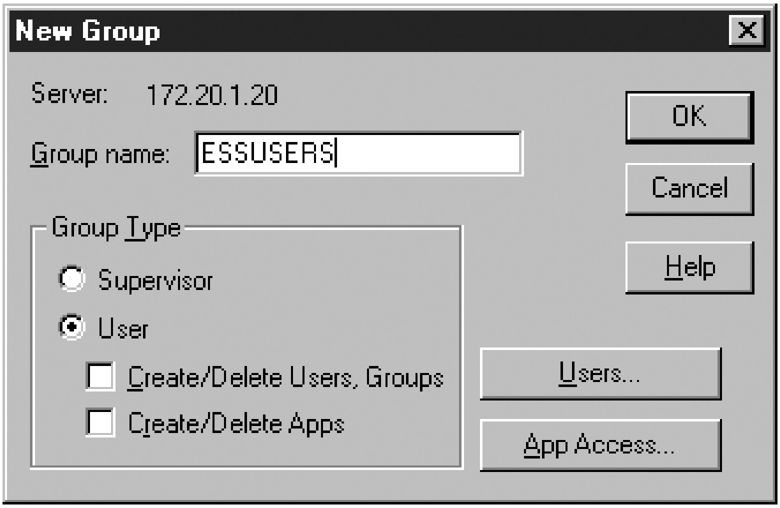


Figure 3. Essbase User/Group Security

b. To create a new Essbase group, click **New Group**. The New Group dialog box will appear.

c. Create an Essbase/400 group that matches the group name specified in the OS400UsersGroup configuration file entry (in this case, `ESSUSERS`). Specify the group type as `User`.



New Group

Server: 172.20.1.20

Group name: ESSUSERS

Group Type

○ Supervisor

● User

□ Create/Delete Users, Groups

□ Create/Delete Apps

Figure 4. Essbase New Group dialog box

d. Click **OK**. The new Essbase group is added. Click **Close** to close the User/Group Security dialog box.

Note: Implementing Integrated Security shifts user management to OS/400 and Warehouse Manager Client. As a result, you will no longer add users to Essbase via Application Manager security. You should use Application Manager security only to create the Essbase group; after that, any user maintenance (for example, adding or deleting users) should be done via OS/400.

Step 8. Stop and restart Essbase/400 OLAP Server. Integrated Security is now in effect.

Step 9. Using Application Manager, from the Server menu, choose **View Event Log**. Scroll to the bottom of the log to ensure that there are no errors related to Integrated Security. If you had restarted the server before completing the necessary steps (such as creating the OS/400 or Essbase/400 groups), you may see some error messages. All necessary groups must be created before stopping the server.

Step 10. When you are finished viewing the log file, from the File menu, choose **Close**.

Now, as users log on to Essbase/400, it will verify that the user name is an eligible Integrated Security OS/400 user and validate the password against the OS/400 user password. If the OS/400 user is deleted, the Essbase/400 user is automatically deleted. If the OS/400 user is disabled, the Essbase/400 user is disabled. Likewise, if the OS/400 password is expired, the Essbase/400 password is expired. In addition, the OS/400 user profile description is reflected in the Essbase/400 user description.

At this point, all Integrated Security users will have access only to the Essbase/400 applications and databases to which the Essbase/400 ESSUSERS group has access. For instructions on controlling which OS/400 users have access to which applications and databases, see “Setting Up Group-Level Control” below.

Note: If all Essbase users will have the same level of access, you may choose to modify the rights of the Essbase/400 ESSUSERS group so that as each Integrated Security user is automatically added to Essbase/400, the appropriate rights are granted via the ESSUSERS group. See the *Hyperion Essbase Database Administrator's Guide* for information about how to do this. However, if you want to have different levels of access for different groups, do not give ESSUSERS any additional access; instead, continue with the following section.

Setting Up Group-Level Control

Integrated Security can give you even more control by enabling you to specify at a group level which OS/400 users have access to particular applications and databases. For example, members of the Sales group can see the Sales database, while members of the Human Resources group can see the Human Resources database.

Note: To implement group-level security, you must define an OS400USERSGROUP (such as ESSUSERS). If you have not done that yet, see “Setting Up Essbase/400 Integrated Security” on page 35.

You may already have OS/400 group user profiles established for different departments or roles in your company. For example, you may have an OS/400 group user profile for access to sales data, one for access to accounting information, and another for access to HR data. If these group user profiles provide an appropriate method of defining who should have access to Essbase/400 applications, you can use Integrated Security so that as users are added or removed from the OS/400 groups, access to Essbase/400 applications and databases are adjusted accordingly. If you already have these groups defined, skip the first step. Remember that members of these OS/400 groups must also be members of OS/400 group ESSUSERS (or the equivalent) to participate in Essbase/400 Integrated Security.

Step 1. If OS/400 groups are not appropriately defined for your Essbase/400 application and database security requirements, you must revise the OS/400 group definitions. Open Warehouse Manager. To add a Group User Profile:

a. In Explorer, connect to the iSeries data source that you want to create the profile on.

b. From the Manage menu, choose **Profile**, and then choose **Create Group**. The Create Group Profile dialog box appears.

c. Enter a name and description for the profile.

d. The name for the group profile is your choice. In this guide, we will assume that you are creating two new groups, Sales and HR.

e. Define any other settings on the General through Workstation tabs. These settings will not affect Essbase/400 usage rights. Generally, you can accept the defaults.

f. Click **Update** to update the iSeries and refresh the profiles list in Explorer. Then click **Close**.

g. Repeat the first four substeps of this step for each group that you want to create.

h. You can add users to these groups now or later. For users who do not have a primary group, find ESSUSERS in the Explorer view. Right-click and choose **Change Profile**. The Create Group Profile dialog box will appear. Click the **Members** tab and add all the necessary members. When you are finished, click **Update**. Remember that members in these groups must also be members of ESSUSERS or the equivalent.

Step 2. At this point, you have created Group User Profiles (Sales, HR) that identify users who can access the Sales and HR applications and databases. Now, you will create these groups in Essbase/400:

a. Open Essbase Application Manager on your PC. Connect to Essbase/400 OLAP Server. From the Security menu, choose **Users/Groups**. The User/Group Security dialog box will appear (see Figure 3).

b. To create a new Essbase group, click **New Group**. The New Group dialog box will appear.

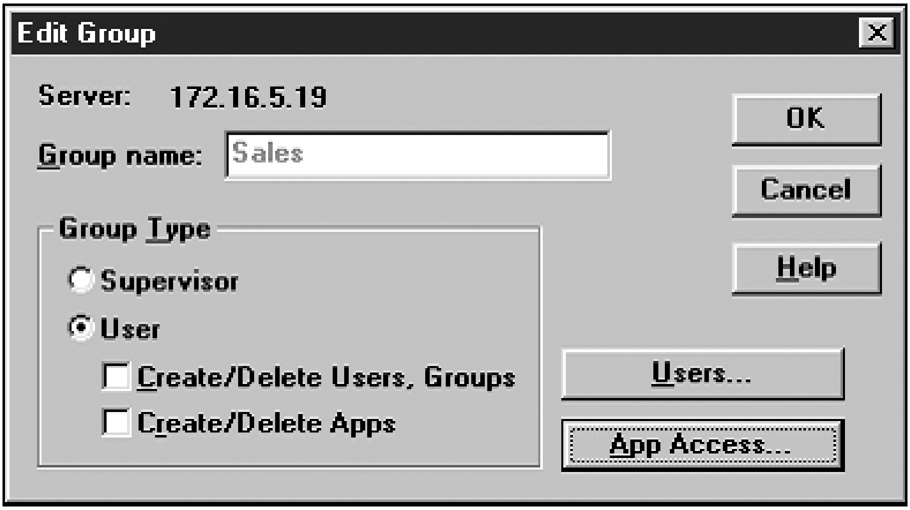
c. Create an Essbase/400 Group that matches the name of the group you just created in Warehouse Manager (in this case, HR). Specify the group type as **User**.

d. Click **OK.** The new Essbase group is added.

e. Repeat the previous four substeps to create the Sales group.

Step 3. Adjust the application and database access as needed for each group. For example, the Sales group should have access to the Sales database and the HR group should have access to the HR database.

a. In Application Manager, in the User/Group Security dialog box (see Figure 3), select a group in the Groups box and click **Edit Group**. The Edit Group dialog box will appear.



Server: 172.16.5.19

Group name: Sales

Group Type

○ Supervisor

● User

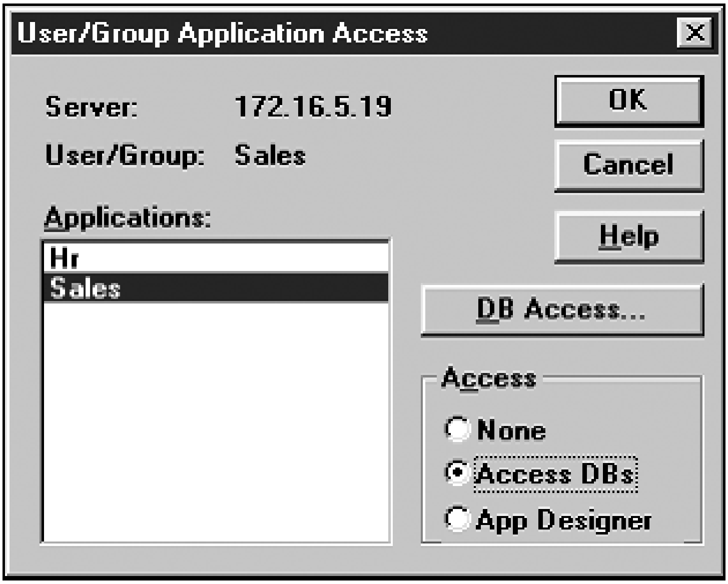
□ Create/Delete Users, Groups

□ Create/Delete Apps

Figure 5. Essbase Edit Group dialog box

b. In the Group Type box, select **User**.

c. To specify the group's application access, click **App Access**. The User/Group Application Access dialog box will appear, as shown in Figure 6.



Server: 172.16.5.19

User/Group: Sales

Applications:

- Hr
- Sales

Access

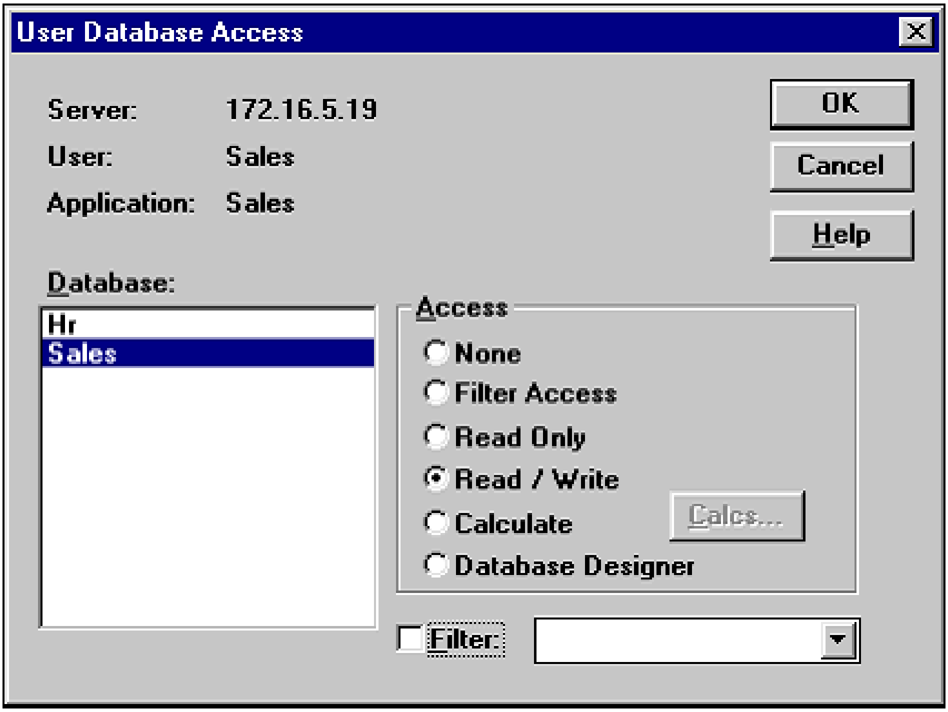
○ None

● Access DBs

○ App Designer

Figure 6. Essbase User/Group Application Access dialog box

d. Choose an application in the Applications box and select the access in the Access box. If you have created databases under the application, the **DB Access** button is enabled if you select **Access DBs**. To set the database access, click **DB Access.** The User Database Access dialog box will appear.



User Database Access

Server: 172.16.5.19

User: Sales

Application: Sales

Database:

- Hr
- Sales

Access

○ None

○ Filter Access

○ Read Only

● Read / Write

○ Calculate

○ Database Designer

□ Filter:

Figure 7. Essbase User Database Access dialog box

e. Choose a database in the Database box and select the access level for the group in the Access box.

f. When you are finished, click **OK**. The User/Group Application Access dialog box will appear (see Figure 6). Click **OK**. The Edit Group dialog box will appear. Click **OK.** The User/Group Security dialog box will appear (see Figure 3). Click **Close.**

g. Repeat the previous six substeps for each group for which you want to set access.

Step 4. Add entries to the Essbase configuration file to tell Essbase/400 which Essbase/400 groups are participating in Integrated Security. Without stopping the server, add one line, as shown below, to the Essbase configuration file for each group you have defined. (For information about editing the configuration file, see “Editing or Displaying the ESSBASE.CFG File” on page 12.)

```
OS400COMMONGROUP GROUPNAME
```

where *GROUPNAME* refers to the name of the Essbase/400 group you just created—for example, Sales or HR.

Step 5. Stop and restart Essbase/400 OLAP Server.

Step 6. Using Application Manager, from the Security menu, choose **View Event Log.** Scroll to the bottom of the log to ensure that there are no errors related to Integrated Security. If you had restarted the server before completing the necessary steps (such as creating the OS/400 or Essbase/400 groups), you may see error messages. All necessary groups must be created before stopping the server.

Step 7. When you finish viewing the log file, from the File menu, choose **Close**.

You have completed setup for Integrated Security.

For users who are members of the `OS/400USERSGROUP` (that is, `ESSUSERS`), membership in the "common" Essbase/400 groups (for example, Sales, HR) is dynamically synchronized with the current membership status in the corresponding OS/400 common group.

Automatic creation of an Essbase/400 user occurs when the user first logs on to Essbase/400. Automatic deletion of users (if required) and synchronization of group memberships occurs at every logon, at Essbase/400 startup, and when the Security dialog box in Application Manager is opened.

Troubleshooting Integrated Security

If you've followed all of the previous instructions and Integrated Security still does not seem to be working, check this list for common problems and their fixes. After you fix any problems, stop and restart Essbase/400 OLAP Server to apply your changes.

- Your log file has any of the following error messages:

  ```
  1180002: OS400USERSGROUP (ESSUSERS) is not an OS/400 Group User Profile, OS400USERSGROUP disabled
  1180003: OS400USERSGROUP (ESSUSERS) is not an Essbase/400 group, OS400USERSGROUP disabled
  1180004: OS400COMMONGROUP (HR) ignored. It is not an OS/400 Group User Profile.
  11800005: OS400COMMONGROUP (HR) ignored. It is not an Essbase/400 group.
  ```

  Verify that you have created the group profile in Warehouse Manager Client, entered the group in the Essbase configuration file, and added the group in Application Manager. Remember that the group name must be the same in all three locations.

- You've set up your special groups, HR and Sales, but the users are rejected when they try to log on.

  Verify that the users are also members of the OS/400 `ESSUSERS` group. If you are enabling a subset of your users to access Essbase, those users must first be members of OS/400 ESSUSERS. Then you can add them to any additional OS/400 groups (for example, Marketing, Sales) that you define.

- Your error file includes the following message:

  ```
  1180008: OS400USERSGROUP (ESSUSERS) has been deleted, OS400USERSGroup disabled
  ```

  Verify that the OS400USERSGROUP is still defined in Application Manager. If the group has been deleted, it must be re-created for Integrated Security to work properly.

Automatic Deletion of Essbase/400 Users

There are two ways that Essbase/400 users are automatically deleted:

- If you have specified only a group name (for example, ESSUSERS) in the OS400UsersGroup configuration entry (that is, you did not specify *ALL), then any user in the Essbase/400 group specified on the configuration entry who is not a member of the OS/400 group is automatically deleted from Essbase. This ensures that when an OS/400 user is deleted, the Essbase/400 user is also deleted.
- If you have specified *ALL on the OS400UsersGroup entry, automatic deletion will occur when the OS/400 user profile is deleted.

What about Passwords?

Passwords for Essbase/400 user IDs created by the Integrated Security feature must be changed by changing the OS/400 password via the CHGPWD or CHGUSRPRF command. However, some Essbase/400 Client programs (including Application Manager and Spreadsheet Add-ins) will allow the end user to change a password at Essbase/400 logon time if the password has expired. For an Integrated Security user, this will also change the OS/400 password.

Note: Changing a password via the Essbase/400 Application Manager Security menu option or via ESSCMD will not affect the OS/400 password.

Essbase/400 normally requires that passwords be at least six characters long. OS/400 allows a wide variety of password rules to be optionally enforced and even allows a user exit for password screening. When using the Integrated Security feature, the normal Essbase/400 requirement of six-character passwords is no longer enforced. This will allow existing OS/400 users with shorter passwords to use Essbase without needing a new password. Note that any rules or exits defined under OS/400 will also apply to Integrated Security users when they attempt to change an expired password via an Essbase interface.

Changing the Password with a Product Application

If a user's Essbase/400 password was changed or expired, the user can update the applicable control table settings with the SCCHKPWD command. The SCCHKPWD command searches the local system and lists and/or updates the necessary control table settings with the new password. For more information about the SCCHKPWD command and its ESSBASE parameter, see "SCCHKPWD Command Parameters" on page 52.

Renaming Users and Groups

When Integrated Security is enabled, certain renaming operations within Essbase/400 are not permitted. The special-purpose groups specified in the Essbase configuration file entries for OS400UsersGroup and OS400CommonGroup may not be renamed. In addition, any Essbase/400 integrated user may not be renamed. These restrictions prevent the Integrated Security configuration from becoming inconsistent.

Essbase/400 Jobs

You can use the WRKACTJOB command to verify that your Essbase/400 servers are running. The Essbase/400 OLAP Server job will have a name identical to the job description name, a job type of BCH, and a function of PGM-ESSBASE. Applications run under separate jobs with the job name identical to the application name and a job type of BCI.

The Essbase/400 Integration Server will have a name identical to the job description name, a job type of BCH, and a function of PGM-OLAPISVR.

Message Logging

When running Essbase Application Server on platforms other than the iSeries, messages that track the requests and subsequent error messages are displayed on the terminal. Because of internal iSeries restrictions, most of these messages are recorded in the Essbase/400 log file and are not displayed on the terminal. To access the Essbase/400 log file on the iSeries, use the DSPESSLOG command. For information about maintaining the Essbase/400 log file, see, "Maintaining Essbase/400 Log Files" on page 29.

Backing Up Essbase/400 Files

Essbase/400 does not provide direct backup capability. Rather, it assumes that the underlying operating system will provide adequate backup mechanisms. You can use the SAVLIB or SAV commands to save Essbase/400 applications and libraries. The following procedure creates two save files in QGPL, which hold the QSYS file system and IFS file system information important to Essbase/400. This procedure assumes that Essbase/400 is installed in the default library of ESSBASE.

1. Shut down Essbase/400 servers before backing up files.
2. Type CRTSAVF FILE(QGPL/SAVEQSYS) and press Enter.

3. Type `SAVLIB LIB(ESSBASE) DEV(*SAVF) SAVF(QGPL/SAVEQSYS)` *or*`SAV DEV('/QSYS.LIB/QGPL.LIB/SAVEQSYS.FILE') OBJ(('/QSYS.LIB/ESSBASE.LIB'))` and press Enter.
4. Type `CRTSAVF FILE(QGPL/SAVEIFS)` and press Enter.
5. Type `SAV DEV('/QSYS.LIB/QGPL.LIB/SAVEIFS.FILE') OBJ(('/ESSBASE'))` and press Enter.

Note: All files found in the *APP* directory, as well as the *ESSBASE.SEC* file, must be backed up and restored together to ensure consistency.

Restoring Essbase/400 Files

If you used the procedure described in "Backing Up Essbase/400 Files" on page 49, use the following procedure to restore your Essbase/400 files:

1. Type `RSTLIB SAVLIB(ESSBASE) DEV(*SAVF) SAVF(QGPL/SAVEQSYS)` *or*`RST DEV('/QSYS.LIB/QGPL.LIB/SAVEQSYS.FILE') OBJ(('/QSYS.LIB/ESSBASE.LIB'))` and press Enter.
2. Type `RST DEV('QSYS.LIB/QGPL.LIB/SAVEIFS.FILE') OBJ(('/ESSBASE')` and press Enter.

Appendix. Commands

This appendix lists many iSeries and Essbase/400 commands needed to maintain the software.

iSeries Commands

Table 7 lists common iSeries commands for maintaining server libraries.

Table 7. Common iSeries commands

Command	Description	Usage
ADDLIBLE	Add a library to your library list.	To add a Warehouse Manager Server or Essbase/400 server library to the library list when you need to run a server command from the iSeries command line.
CFGTCP	Configure TCP/IP.	To configure iSeries TCP/IP.
DSPPTF	Display PTFs.	To find the latest PTFs installed on your system.
DSPSYSVAL	Find system information.	To find system information, such as model and serial number. For example, to find the model number, enter DSPSYSVAL SYSVAL(QMODEL); to find the serial number, use QSRLNBR for the system value.
DSPUSRPRF	Display user profile.	To display a user's profile.
EDTF	Edit file.	To display or edit iSeries files.
WRKACTJOB	View active jobs.	To see if the Warehouse Manager Server or Essbase/400 servers are running or to view active jobs, such as Essbase/400 and Warehouse Manager jobs.
WRKOBJLCK	Check for active jobs against a Warehouse Manager library.	To check for iSeries jobs that may be active against a Warehouse Manager Server or Essbase/400 server library; in particular, to check for TCP/IP jobs. Useful when installing over the top of an existing Warehouse Manager library or debugging system-related problems (for example, when TCP/IP connections fail).
WRKSRVTBLE	View port numbers already in use.	To find an available port number when configuring TCP/IP.
SCCHKPWD	Manage iSeries passwords stored in control tables (see Table 8).	Update the control tables with a user's new password, or set the UPDATE parameter to *NO to simply generate a list. Type the SCCHKPWD command on the iSeries command line, and press F1 for parameter information.

© Copyright IBM Corp. 1998, 2002

SCCHKPWD Command Parameters

The SCCHKPWD command searches the local system and lists and/or updates the necessary control table settings with the new password. This ensures uninterrupted scheduled reports and Warehouse Builder distributions.

The following are *required* SCCHKPWD parameters:

USERID (no default) - The name of the user profile to list or update passwords OLDPWD (no default) - The current password for the USERID

Table 8 lists the *optional* SCCHKPWD parameters.

Table 8. SCCHKPWDoptional parameters

Parameter	Description
UPDATE (default = *NO)	Indicates whether the passwords for this USERID should be updated in the control tables. Supported values for this parameter are *YES and *NO. Regardless of which option is used, a report of the items matching the specified USERID, OLDPWD, and server(s) is generated and stored in a spool file for the user.
DETAILS (default = *YES)	Indicates whether the generated report lists ALL entries matching the specified criteria or if only a summary of the RDBs, servers, and ODBC data sources affected are included.
RDB (default = *LOCAL)	The RDB name of a DB2 database for which passwords may need to be checked or updated. This will affect Warehouse Builder distributions using DB2 sources or targets and scheduled Enterprise Reporting items using ODBC data sources to the specified RDB. (Non-STRATEGY data sources must use the ODBC parameter to update passwords.) Special values supported are: *LOCAL, which references the local iSeries database, *ALL, which references all DB2 databases used by items in the catalogs, and *NONE, which indicates that no RDB-related passwords should be checked or updated.
AOS (default = *LOCAL)	The name of a server for which passwords may need to be checked or updated. This will affect only Enterprise Reporting scheduled items using a source or target server. Special values supported are *LOCAL, which references all servers installed on the local system, *ALL, which references any server regardless of the system it's installed on, and *NONE, which indicates that no server-related passwords should be checked or updated.

Table 8. SCCHKPWDoptional parameters (continued)

Parameter	Description
ODBC (default = *NONE)	The name of an ODBC data source for which passwords may need to be checked or updated. This will affect both Enterprise Reporting scheduled items using DBQs that reference non-STRATEGY data sources and data sources used in WB non-DB2 "pulls." Due to the nature of ODBC, these names are arbitrary, and there is no implicit relationship between the data source name and the underlying system/database (that is, different users may use the same data source name to reference different databases). As a result, take care when using SCCHKPWD to update non-STRATEGY ODBC passwords to ensure that the correct data sources are updated. Special values supported are: *ALL, which references all ODBC data sources, and *NONE, which indicates that no (non-STRATEGY) ODBC-related passwords should be checked or updated.
ESSBASE (default = *NONE)	The name of an Essbase server for which a password may need to be checked or updated. Special values supported are: *ALL, which indicates all ESSBASE servers, and *NONE, which indicates that no ESSBASE server-related passwords should be checked or updated.
NEWPWD (no default)	The new password to use for the USERID. This setting is ignored if UPDATE = *NO.
PWDCONFIRM (no default)	Confirmation of the new password to use for the USERID (to make sure it was typed correctly). This setting is ignored if UPDATE = *NO.

Essbase/400 Commands

Table 9 lists common Essbase/400 commands for maintaining an Essbase/400 library.

Note: Before using the following commands, add your Essbase library to your iSeries library list and make sure that you have only one Essbase library in your library list. Run the command ADDLIBLE ESSBASE, where ESSBASE is the name of the Essbase library you want to work with.

Table 9. Common Essbase/400 commands

Command	Description	Usage
CHGESSENV	Change Essbase/400 environment variables.	To change the environment.
DSPESSENV	Display Essbase/400 environment variables.	To display the version or the environment.
DSPESSLOG	Display Essbase/400 Logs.	To read the server and application logs from the iSeries.
STRESSSVR	Start the Essbase/400 servers.	To start the servers and specify job descriptions to assign to the instances.

Table 9. Common Essbase/400 commands (continued)

Command	Description	Usage
ENDESSSVR	End the Essbase/400 command processor.	To stop the servers.
ESSMSH	Start MaxL/400 or run MaxL statements.	To start the MaxL/400 shell or run MaxL statements from a script.
GRTESSAUT	Grant Essbase authority.	To allow users lacking proper authority to end/start the Essbase/400 servers.
RUNMAXLSTM	Run MaxL statements with return codes.	To run MaxL statements and receive return codes (completion status) that you can use in another program.
STRESSCMD	Start the ESSCMD command processor.	To run any Essbase/400 command or run scripts.
UNINSTSC	Uninstall the Essbase/400 or Warehouse Manager Library.	To remove a library from the system. For more information about the UNINSTSC save file, see the *STRATEGY Administrator's Guide*.

Table 10 lists common configuration file settings for implementing Integrated Security and specifying how many days of log records to preserve. These keywords are added to the Essbase configuration file.

Table 10. Common Essbase/400 configuration file settings

Setting	Description	Usage
OS400UsersGroup	Enable all OS/400 users or a subset of OS/400 users to access Essbase/400.	When setting up Integrated Security, this setting gives Essbase/400 access to all users or to a specified group of users.
OS400CommonGroup	Assign OS/400 user profiles to Essbase/400 groups.	When setting up Integrated Security, as OS/400 users are added or removed from OS/400 groups, they are automatically added or removed from the corresponding Essbase/400 group.
OS400AutoDelete	Enable automatic deletion of Essbase/400 users.	When setting up Integrated Security, this setting automatically deletes all users in the auto-delete group each time the Essbase/400 OLAP Server is started.
OS400DAYSLOGRETAIN	Specify how many days of Essbase log files to preserve.	To set the number of days of log records to keep when the Essbase/400 OLAP Server or an Essbase application is started. Example: A value of 2 will retain two days of records.

Table 10. Common Essbase/400 configuration file settings (continued)

Setting	Description	Usage
SCHASHCALCWOSHAREDMBR	Improve performance when calculating databases without shared members.	If CALCOPTCALCHASHTBL is true, insert a SCHASHCALCWOSHAREDMBR line in config file specifying each database that does not have shared members. Must be applied once per database.**Example:** SCHASHCALCWOSHAREDMBR tbc demo

Glossary

This glossary defines terms that are used in this book and throughout the IBM DB2 OLAP Server library.

accounts dimension. A dimension type that makes accounting intelligence available. You can tag one dimension as an accounts dimension, but an accounts dimension is not required.

administrator. A person who is responsible for installing and maintaining DB2 OLAP Server and for setting up user accounts and security.

agent. A process that starts and stops applications and databases, manages connections from users, and handles user-access security.

alias ID table. A table created by DB2 OLAP Server in your relational database that contains a mapping of multidimensional alias table names to ID numbers allocated by DB2 OLAP Server.

alias ID view. A view created by DB2 OLAP Server in your relational database that contains one row for each multidimensional alias table used with a relational cube. There is one alias ID view for each relational cube.

alias name. An alternate name for a dimension or member.

anchor dimension. A dense dimension that is specified as the dimension that DB2 OLAP Server uses to help define the structure of the fact table it creates in the relational cube for a multidimensional database.

API. Application programming interface. The DB2 OLAP Server API is a library of functions that you can use in a custom C or Visual Basic program to access DB2 OLAP Server.

Application Manager. A tool that you can use to create and maintain OLAP applications.

block. A string of data elements recorded or transmitted as a unit.

cache. A component of memory. Each multidimensional database contains a data cache and an index cache.

calculation. An equation within a database outline, a calculation script, or a report script that calculates a value for a particular member or point in a report.

calculation script. A text file that contains instructions to perform calculations within a multidimensional database. Also called a calc script.

Commit Block parameter. A parameter on the Transaction page of the Database Settings notebook in the Application Manager that you use to set the number of blocks that can be changed before DB2 OLAP Server commits the blocks.

cube catalog table. A table that DB2 OLAP Server creates in your relational database that contains a list of all the multidimensional databases that are stored in your relational database. The cube catalog table also shows the application with which each cube is associated. Each time that you create a multidimensional database, DB2 OLAP Server creates a new row in this table.

cube catalog view. A view that DB2 OLAP Server creates in your relational database that allows an SQL user to access a list of OLAP applications and relational cubes.

cube table. A table that DB2 OLAP Server creates in your relational database that contains a list of dimensions in a relational cube and information about each dimension.

cube view. A view that DB2 OLAP Server creates in your relational database that allows an

© Copyright IBM Corp. 1998, 2002

SQL user to access the names of all dimensions in a relational cube and associated information for each dimension. There is one cube view for each relational cube in your relational database.

data load. The process of populating a multidimensional database with data. Loading data establishes actual values for the values of the cells defined in the database outline for the database.

data load rules. A set of operations that DB2 OLAP Server performs on data as it is loaded from an external source file.

database administrator. A person responsible for administering a relational database.

database log file. A set of primary and secondary log files consisting of log records that record all changes to a database. The database log file is used to roll back changes for units of work that are not committed and to recover a database to a consistent state.

database managed space (DMS). Space in a table that is managed by the database.

database name. The name of the relational database where you want DB2 OLAP Server to store your multidimensional data.

database outline. The structure that defines all elements of a database within DB2 OLAP Server. It contains definitions of dimensions and members, dense or sparse dimension tags and attributes, the anchor dimension attribute, calculations, shared members, and alternations to the basic roll-up structure of the database.

database password. The password for the user ID that you want DB2 OLAP Server to use to log on to your relational database.

database section. A section in the Relational Storage Manager configuration file that contains values for parameters that override the values for the same parameters for the current application.

database settings. Settings that you can modify to improve performance and space utilization for your relational database. You can change database settings by using utilities or commands supplied with your relational database management system.

database user ID. The user ID that you want DB2 OLAP Server to use to log on to your relational database. The default is the supervisor ID that you specify when you start DB2 OLAP Server for the first time.

dense dimension. A dimension with a high probability for occupying one or more data points in every combination of dimensions that occurs.

dimension. A data category, such as time, accounts, products, or markets. In a multidimensional database outline, the dimensions represent the highest consolidation level.

dimension table. A table that DB2 OLAP Server creates in your relational database that contains detailed information about the members in a dimension. There is one dimension table for each dimension in an outline.

dimension view. A view that DB2 OLAP Server creates in your relational database that allows an SQL user to access information about members contained in a dimension.

ESSCMD. A command-line interface used to perform server operations interactively or through a batch file.

generation name. A unique name that describes a generation in a database outline.

generation table. A table that DB2 OLAP Server creates in your relational database that contains generation numbers and names for each named generation specified when you created the outline. There is one generation table for each dimension in an outline.

level name. A unique name that describes a level in a database outline.

member. A discrete component within a dimension. For example, January 1997 or 1Qtr97 are typical members of a Time dimension.

metaoutline. A template containing the structure and rules for creating a database outline from an OLAP model. Using the metaoutline, you can create a database outline and load data into it. The DB2 OLAP Integration Server desktop includes a tool called OLAP Metaoutline that you can use to create one or more metaoutlines from an OLAP model.

model. A logical model (star schema) that you create from tables and columns in a relational database. You can use the OLAP model to create a metaoutline that generates the structure of a multidimensional database. The DB2 OLAP Integration Server desktop includes a tool called OLAP Metaoutline that you can use to design and create an OLAP model based on a relational data source.

multidimensional data. The data in a multidimensional database. Data can include basic data values (loaded from an external source) that represent combinations of the lowest level of members in the dimensions of the database; data values that are calculated from the base data values; and rolled up data values that are created by combining values for members in dimension hierarchies.

multidimensional database. An OLAP database that you create using the Application Manager or ESSCMD commands (in DB2 OLAP Server) or with the DB2 OLAP Integration Server desktop, which is in the Starter Kit. An OLAP database includes a database outline, data, associated optional calculation scripts, optional report scripts, and data load rules. DB2 OLAP Server stores the actual data and a shadow of the database outline in tables in a relational database.

named pipes. An API used for special node-to-node applications and particularly for access to communications and database servers.

OLAP application. An application that you create using the Application Manager or the ESSCMD commands (in DB2 OLAP Server) or with the DB2 OLAP Integration Server desktop, which is in the Starter Kit. An OLAP application can contain one or more multidimensional databases and any associated calculation scripts, report scripts, and data load rules.

OLAP metaoutline. See *metaoutline*.

OLAP model. See *model*.

online analytical processing (OLAP). A multidimensional, multi-user, client server computing environment for users who need to analyze consolidated enterprise data in real time. OLAP systems feature zooming, data pivoting, complex calculations, trend analyses, and modeling.

outline. See *database outline*.

RDBMS. Relational database management system. A database that can be perceived as a set of tables and manipulated in accordance with the relational model of data.

relational attribute. A characteristic of a dimension table, represented by a column. You can run SQL statements against the data in relational attribute columns.

relational database. A database that is organized and accessed according to relationships between data items. A relational database contains a collection of relational tables, views, and indexes.

report script. An ASCII file that contains Report Writer commands that generate one or more production reports. Report scripts can be run in batch mode, using the ESSCMD command-line interface, or through the Application Manager. The script is a text file that contains data retrieval, formatting, and output instructions.

restructure. An operation to regenerate or rebuild the tables and views that DB2 OLAP Server created in your relational database.

shadow. Information stored in relational tables that shadows the information stored in a multidimensional database outline.

shared member. A member that explicitly shares storage space with another member of the same name. This member has an attribute that designates it as shared. Shared members prevent making extra calculations on a member that appears in more than one location in a database outline.

sibling. A child member at the same branch level.

sparse dimension. A dimension with a low percentage of available data positions filled. For example, a product that is not sold in all of a company's available markets would be a good sparse candidate.

Spreadsheet Add-in. Software that merges with Microsoft Excel and Lotus 1-2-3 to enable analysis of multidimensional databases. The software library appears as a menu Add-In to the spreadsheet and provides such features as connect, zoom-in, and calculate.

SQL application. An application that uses SQL statements. You can use SQL applications to access data in a relational cube.

SQL. Structured Query Language. A standardized language for defining and manipulating data in a relational database.

star schema. The type of relational database schema used by DB2 OLAP Server. When you use the Application Manager to create a multidimensional database, DB2 OLAP Server creates a main fact table and a set of dimension tables. The fact table holds the actual data values for the database, and the dimension tables hold data about members and their relationships.

star view. A relation view that DB2 OLAP Server creates in your relational database that allows an SQL user to access data from the star schema in a single view with the JOIN already done.

table space. An abstraction of a collection of containers into which database objects are stored. A table space provides a level of indirection between a database and the tables stored within the database. A table space:

- Has space on media storage devices assigned to it.
- Has tables created within it. These tables will consume space in the containers that belong to the table space. The data, index, long field, and LOB portions of a table can be stored in the same table space, or can be individually broken out into separate table spaces.

time dimension. A dimension type that defines how often you collect and update data. You can tag only one dimension as Time, although you do not need to have a Time dimension.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will

© Copyright IBM Corp. 1998, 2002

be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Trademarks

1–2–3
AIX
AS/400
DB2
DB2 OLAP Server
DB2 Universal Database
Net.Data
Notes
OpenEdition
iSeries
OS/390
OS/400

DRDA
IBM
iSeries
Language Environment
RS/6000
S/390
SecureWay

The following terms are trademarks or registered trademarks of other companies:

Lotus and 1-2-3 are trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Microsoft, Windows, and Windows NT are trademarks or registered trademarks of Microsoft Corporation.

Java or all Java-based trademarks and logos, and Solaris are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, other countries, or both.

UNIX is a registered trademark in the United States, other countries or both and is licensed exclusively through X/Open Company Limited.

Other company, product, or service names may be trademarks or service marks of others.

Index

© Copyright IBM Corp. 1998, 2002

Program Number: 5724–B78

Printed in U.S.A.

SC18-7226-00