IBM® DB2 ® OLAP Server™ for iSeries™

Administrator's Guide for Builder on iSeries

Version 8.1

SC18-7227-00

IBM® DB2 ® OLAP Server™ for iSeries™

Administrator’s Guide for Builder on iSeries

Version 8.1

SC18-7227-00

Note

Before using this information and the product it supports, be sure to read the information in "Notices" on page 119.

First Edition (November 2002)

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

Contents

Preface

Welcome to the DB2 OLAP Server for iSeries V8.1 *Administrator's Guide for Builder on iSeries*. This guide was written by ShowCase (a division of SPSS, Inc.) for their ShowCase Suite (STRATEGY) product under the title *STRATEGY Administrator's Guide*. This guide discusses the administration and maintenance of the Warehouse Manager and Warehouse Builder components of DB2 OLAP Server for iSeries V8.1.

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 find documentation online
- How to contact the documentation team to provide feedback about this guide

For installation information, see the DB2 OLAP Server for iSeries V8.1 *Installation Guide for iSeries*. This guide contains the information found in the ShowCase

About This Guide

- Chapter 1, "Starting and Stopping the Servers" on page 1, explains how to start and stop Warehouse Manager and how to prepare the servers for client connections.
- Chapter 2, "Controlling and Managing Performance" on page 3, explains how to manage performance to ensure successful deployment.
- Chapter 3, "Managing Data on the iSeries" on page 33, describes how to maintain your Warehouse Manager library, including backing up and restoring files, checking for usage, and configuring TCP/IP.
- Chapter 4, "Warehouse Manager Server Maintenance" on page 49, describes typical server maintenance procedures.
- Appendix A, "Commands" on page 85, contains a list of commands commonly used to maintain iSeries libraries.

- Appendix B, “Sample Databases” on page 95, describes the sample databases installed with Warehouse Manager Server.

Chapter 1. Starting and Stopping the Servers

This chapter outlines the procedures for starting and stopping the Warehouse Manager Server and Orion Server.

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.

Warehouse Manager Server

This section outlines the commands to start and stop Warehouse Manager Server.

Note: Users of STRATEGY 3.5 and earlier releases will notice that since release 4.1, AOS no longer has its own "start" and "stop" server commands. To simplify managing the servers, the Analyzer OLAP functionality has been integrated into one Warehouse Manager Server. Now, when you start and stop the Warehouse Manager Server, you are also starting and stopping Analyzer OLAP features.

Starting Warehouse Manager Server

Running `STRSCSVR` will start Warehouse Manager Server and enable Analyzer OLAP features that you are licensed to use. If you are using the TCP/IP communications protocol, the Warehouse Manager TCP/IP Server job should start automatically when the installation is finished, provided that TCP/IP was configured correctly.

However, if you opted to set up Warehouse Manager TCP/IP after installation, or if you stopped Warehouse Manager Server, you must start Warehouse Manager TCP/IP Server.

1. Log on with `QSECOFR` or equivalent authority (`*JOBCTL` authority is required).
2. Add the Warehouse Manager Server library to your library list. This is the name of the library in which the server is installed.
3. Run the Warehouse Manager command `serverlib/STRSCSVR`, where `serverlib` is the name of the library in which the server is installed.

Stopping Warehouse Manager Server

To stop the Warehouse Manager Server and Warehouse Manager TCP/IP Server connections:

1. Log on with QSECOFR or equivalent authority (*ALLOBJ and *JOBCTL authorities are required).
2. Add the Warehouse Manager Server library to your library list.
3. Run the Warehouse Manager command `serverlib/ENDSCSVR`, where `serverlib` is the name of the library in which the servers are installed.

The ENDSCSVR command will terminate all current TCP/IP connections to the specified library, including Analyzer OLAP functionality. For more information on this and related topics, see “Uninstalling a Warehouse Manager Library” on page 52.

Orion Server

Instructions for starting and stopping Orion server are located in the *Enterprise Reporting Administration Guide.* For a list of commands dealing with installing, starting, and stopping the Orion Server, see Table 15 on page 86.

Chapter 2. Controlling and Managing Performance

Managing the performance of relational databases can be a challenge for the most experienced database administrator. This chapter provides insight into performance planning and tuning relational databases.

These sections focus on performance tuning DB2/400 databases:

- Planning security
- Controlling Query/Report Writer system usage
- Controlling level of parallelism
- Enabling expert cache
- Tuning system performance
- DB2 symmetric multiprocessing for OS/400
- Using auditing to understand queries
- Improving query performance
- Optimizing performance for Warehouse Builder

Planning Security

This section outlines the security issues to consider when deploying STRATEGY for relational data access.

iSeries Security Considerations

When deploying Warehouse Manager products, consider your current iSeries security environment. Items to address when creating a security plan include:

- What information do you currently store or plan to store on your system?
- Will you use STRATEGY applications on an iSeries system that is also currently used for transaction data processing, or will you use STRATEGY software on an iSeries system that is dedicated to data warehouse processing?
- Who needs access to the information? Which applications will they use to gain access to the information?
- What is your overall approach to security? Do you have a restrictive or a non-restrictive security approach?
- What type of information will you secure (confidential, competitive, and operations information)?
- What security scheme do you use for client-server applications other than Warehouse Manager products?

- What other vendors' applications affect the way that you set up your system security?
- What iSeries system security level will you use?

It is important to plan your overall iSeries security approach or have a thorough understanding of the existing iSeries security approach. This allows you to choose the best way to implement Warehouse Manager security to meet your current and future security needs.

Refer to IBM documentation for more detailed iSeries system security information.

Warehouse Manager Server Security

When you set security for Warehouse Manager Server, it affects all client applications connecting to the Warehouse Manager server. These include:

- Warehouse Manager Client
- Warehouse Builder
- Query
- Report Writer

Several methods are provided that affect your server security. These methods interact with your current iSeries security settings, your current applications, applications purchased from other vendors, and your additional security measures (such as exit programs) to provide your total security solution. The primary Warehouse Manager security methods include:

- Using Warehouse Manager Client to apply library security, table security, column security, or row security for specific users or user groups.
- Using Warehouse Manager Client to change server options. These options affect all users.
- Using Warehouse Builder or Query with batch output to a file to create new iSeries objects.

Although the following sections make recommendations for your Warehouse Manager security plan based on specific situations, in general, you will always follow these steps:

1. Designate at least one user profile as a data administrator.
2. Review your user-profile scheme and update it if desired.
3. Decide on a general security policy. Choose whether to restrict global access to files and then authorize specific users to specific files, or to allow global access to files and then exclude specific users from specific files. Implementing this decision requires the following:
   - Set the server options to allow all users to see all libraries and tables, or set the server options to exclude all users from all libraries and tables.

- Use Warehouse Manager Client to apply authorities to specific tables for specific users.

4. Decide whether to restrict users to libraries within their iSeries library lists.
5. Apply any necessary column filters.
6. Apply any necessary row filters.
7. Ensure the correct users are authorized to, or restricted from, specific STRATEGY applications.
8. If desired, use Warehouse Manager products to output directly to libraries that users are authorized to access.

Designating Data Administrators

Using Warehouse Manager Client, designate one or more user profiles as data administrators. The data administrator(s) are always able to view all libraries and tables in Warehouse Manager products, and they are also able to set server options. If you choose a data administrator user profile without *ALLOBJ authority, you must also complete the following steps:

1. Using a profile with *ALLOBJ authority, use Warehouse Manager Client to authorize that user to use Warehouse Manager Client software. See the Warehouse Manager Client online Help for more information.
2. Using a profile with *ALLOBJ authority, give the intended data administrator iSeries *ALL authority to the user profiles that he or she will need to administer. For example, if you want the data administrator to administer all user profiles, use the following iSeries command:

   ```
   GRTOBJAUT OBJ(QSYS/*ALL) OBJTYPE(*USRPRF) USER(userid) AUT(*ALL)
   ```
3. With a profile that has *ALLOBJ authority, give the intended data administrator iSeries *ALL authority to the libraries and tables that he or she will need to administer. For example, if you want the data administrator to administer all libraries, use the following iSeries command:

   ```
   GRTOBJAUT OBJ(QSYS/*ALL) OBJTYPE(*LIB) USER(userid) AUT(*ALL)
   ```
4. With a profile that has *ALLOBJ authority, give the intended data administrator iSeries *USE authority to the following commands in the Warehouse Manager Server library on the iSeries:
   - CHGSCAUT
   - CHKSRVRCAT
   - CHGSRVRDFT
   - CHGSRVRSTS
   - DSPSCAUT
   - RMVSCAUT
   - SETLIBACC

iSeries *USE authority allows the data administrator to run these commands and change server options. For example, use the following iSeries command to grant the user authority to CHGSRVRDFT:

```
GRTOBJAUT OBJ(SCSERVER/CHGSRVRDFT) OBJTYPE(*CMD)USER(userid) AUT(*USE)
```

5. With a profile that has *ALLOBJ authority, give the data administrator iSeries *CHANGE authority to the object SCVERSION in the Warehouse Manager Server library on the iSeries. This allows the data administrator to change the security and *ALLOBJ server options. For example, use the following iSeries command:

```
GRTOBJAUT OBJ(SCSERVER/SCVERSION) OBJTYPE(*USRSPC) USER(userid) AUT(*CHANGE)
```

Advantages of designating data administrators

Designating data administrators leaves you free to have people administering Warehouse Manager security without requiring you to grant those people *ALLOBJ authority. If the user who is in charge of implementing Warehouse Manager security does not have *ALLOBJ authority, it is a good idea to make that user a data administrator.

Disadvantages of designating data administrators

You cannot lock data administrators out of any data in any products (that is Query or Warehouse Builder) that use Warehouse Manager as their server. In other words, data administrators can see any objects for which they have iSeries authority.

Reviewing and Updating Your User Profile Scheme

In several of the recommended practices that follow, you are encouraged to apply specific security settings to libraries and tables on a user-profile basis. This level of security administration can be tedious if you have many users. As an alternative, you may consider using group profiles. If the same security settings will apply to several users, for example all members of a given department, then it would be efficient for you to create a group profile and make those user profiles members of that group. You can then administer Warehouse Manager security for the entire group rather than for each individual member.

You can use Warehouse Manager Client to create group profiles and to make users members of the group.

Warehouse Manager products will honor authority if it is gained through an iSeries supplemental group. However, you cannot administer supplemental groups with Warehouse Manager Client.

Using Menu-Level Security

In a menu-level security scheme, you rely on a set of menus or application interfaces to restrict user access to underlying data. If an option is not on a menu, your users do not have access from an iSeries display session.

Generally, in this scheme, your data files will not have object-level security applied to them; that is, you have not excluded specific users from specific files. Almost all client-server tools will bypass this security, because outside of the environment of the iSeries display session, you no longer have a menu interface protecting access to your files.

Menu-level security is considered a nonrestrictive security plan, and your goal when deploying STRATEGY should be to restrict users from gaining access to data that they must not see.

Option 1—To restrict all access and authorize users to only those files that they need

1. Using Warehouse Manager Client, change server options as follows:
   - Select to exclude users from libraries by default.
   - Select to exclude users from tables by default.
2. Using Warehouse Manager Client, authorize specific users or groups of users to files they need to access.

Advantages

This option provides the most secure and least error-prone protection of sensitive data files. As libraries and tables are added to your database in the future, they are automatically protected from users until you specifically authorize them.

Disadvantages

The process of authorizing users correctly can be time consuming. This effort can be reduced if your data files are organized so that all files in a given library can be opened up to a user or group of users. If this is the case, you can set the server options to exclude users from libraries by default, without excluding them from tables by default. Then you need only to authorize a user to a library to allow him or her access to all tables in that library.

Option 2—To restrict access to specific files

1. Using Warehouse Manager Client, restrict specific users or groups of users from files that they should not be allowed to read.

Advantages

This method can be implemented quickly if you have only a few sensitive files and know exactly which files should not be viewed by which users.

Disadvantages

This method leaves a lot of room for error and is not recommended. If you do not remember or do not know about a sensitive table and fail to secure it, the data will be available to anyone. If sensitive tables are added to your database in the future, anyone can read them until you restrict them.

Option 3—To run at security level 20

Running your iSeries at security level 20 is a variation on menu-level security. At security level 20, all users have the special authority *ALLOBJ, meaning that they can access any information in your data files. If this is the case, you are probably relying on menu security to restrict access to those files. However, outside your iSeries display session environments, users will not be restricted. If this is the case, you need to take additional measures to restrict access to your system.

1. Using Warehouse Manager Client, change server options to **Enforce STRATEGY Security on *ALLOBJ profiles.**
2. Decide whether to use the approach described in the previous two options. If you want to restrict access globally and authorize users to specific files, continue by following the steps in option 1. If you want to allow access globally and restrict users from specific files, refer to option 2.

Advantages

This is the only way to allow Warehouse Manager security settings to take effect when the system security level is set to 20.

Application-Level Security

With an application-level security scheme, all tables have object-level security applied to exclude users from accessing them. Applications then adopt authority in order to access necessary tables. When users attempt to use the tables with a client-server tool or any other application that is not designed to adopt the correct authority, they will still be excluded from the tables.

Application-level security is considered a restrictive security plan, and your goal when deploying STRATEGY should be to allow users to access the tables that they need but still be restricted from using the tables that they should not see.

Security on a New, Dedicated, Data Warehousing iSeries

If you decide to deploy security on a new iSeries that is dedicated to your data warehousing applications, you will not need to consider an existing security scheme as the previous scenarios do. We recommend that you use a restrictive approach when deploying a new data warehousing system.

- Run the iSeries at system security level 30 or higher.
- Do not grant users *ALLOBJ authority unless it is necessary for system operations or other (non-STRATEGY) applications.
- Create all new database files with *PUBLIC authority set to *EXCLUDE.

- Use Warehouse Manager Client to authorize specific users to database files they will need when using Warehouse Manager products.

Advantages

This approach allows you to completely control Warehouse Manager security for users without the risk of users accessing database files that they should not see. If you use other applications on this iSeries in addition to your STRATEGY applications, you can set authorities for specific files needed by those applications without affecting your Warehouse Manager security.

Disadvantages

The flexibility of applying different security schemes for applications from different vendors means that you will need to administer these schemes separately.

Restricting Users to Libraries in their iSeries Library Lists

Library list security allows you to restrict users so that they can access only the set of libraries in their library list. Without library list security, users have access to all libraries.

To set library list security

1. Open Warehouse Manager Client and from the Manage menu, choose **Server Options**.
2. Select the **Library Access** tab in the Server Options dialog box.
3. Select `Return only libraries in user portion of the library list.`

Advantages

If you are currently using the library lists in job descriptions or system values to control user access to libraries as part of your system security policy, choosing library-list security allows you to continue using this mechanism for database file access. For example, the J.D. Edwards package uses library-list security in some cases.

Disadvantages

If library lists are not currently part of your database access scheme, you will probably find no advantage to using library-list security.

Applying Column Filters

Warehouse Manager column security allows users to see data in some columns of a database file, while restricting them from seeing data in other columns. For example, an employee master file might contain employees' names, titles, department numbers, hire dates, and social security numbers. Some of your users may need access to the information, but you want to keep the social security numbers confidential. Using Warehouse Manager column security, you can exclude all users or specific users from seeing the social security column, while allowing them to access all other columns.

To apply column security with Warehouse Manager Client

1. Ensure that the users have authority to see the database file and highlight the particular column that you want to protect.
2. Choose a security option. If you want to restrict everyone from seeing the column, choose the user profile `*PUBLIC`. If you want to restrict specific users, choose those user profiles and specify the `Exclude` authority setting.

Advantages

Without column security, you would not be able to restrict users from any of the data in files that contain columns with sensitive information. Column security gives you much more flexibility in deciding which files your users may access.

Disadvantages

Warehouse Manager column security applies only when you are using the client products that use Warehouse Manager products.

Applying Row Security

Warehouse Manager row security allows users to see data in some rows of a database file while restricting them from seeing data in other rows. For example, a prospective customer file might contain prospect information for your company's potential clients around the world. However, you want only the sales managers in each country to be able to view information about prospects in their country. Using Warehouse Manager row security, you can exclude users from seeing the rows that contain information about prospects in other countries.

If Warehouse Manager row security is applied to a logical file (data view), that security is used. If there is no security on the logical file, then security on the base physical file is used. The field used for the row security constraint for the base physical file must also be in the logical file. Otherwise, the security will fail. This applies only to row security.

To apply row security with Warehouse Manager Client

1. In our example, you would ensure that each sales manager has access to the prospect file, highlight the name of the table, and choose row security.
2. For the U.S. sales manager, select that person's user profile and, in the Row Security text box, enter `\COUNTRY\='USA'` where `COUNTRY` is the name of the column containing country information. Then you would select, for example, the German sales manager and enter `\COUNTRY\='DEU'` and repeat this for the sales manager in each country.

   Another approach to row security is to write user exit programs. The `PCSACC` exit program on the `CHGNETA` command is supported. (However, `*REGFAC` is not supported and is treated as `*OBJAUT`.) User exit programs specified on the `PCSACC` parameter can be used to essentially apply row security for all client-server programs that support this parameter.

Advantages

Row security gives you much more flexibility in deciding which files users may access. Row security enables you to restrict users access to data in files containing rows with information that they should not see.

Warehouse Manager Client gives you a way to specify this row security through a graphical user interface, without forcing you to write user exit programs. It also allows you to apply row security for STRATEGY applications without affecting your other applications.

Disadvantages

Warehouse Manager row security only takes effect when you are using the client products that use Warehouse Manager Server.

Setting Application Security

Warehouse Manager Client allows you to specify which users can run STRATEGY applications. Some applications allow users to do more than access files. For example, Warehouse Builder can create new files, and Warehouse Manager Client can apply security, alias, and resource settings. In most cases, you should exclude all users from using these applications except those specific users whose job it is to administer that application.

Authority to use Query and Report Writer can generally be given to all users. However, it is possible to create batch output files with these products. See the next section for tips on maintaining security with batch output.

System Security and Application Output

Warehouse Builder, Query, and Report Writer are all capable of creating new files on your iSeries. You should manage this capability carefully. Your goals when managing output from applications are:

- Make sure that new objects do not unintentionally expose data to users who were previously restricted from viewing it.
- Make sure that existing data is not accidentally overwritten.

Making Sure That Restricted Data Is Not Exposed

In some circumstances, it is possible for new objects to give users unintended access to data. For example, suppose you have set Warehouse Manager security to exclude some users from certain tables. But a user who does have authority to access that table might run a Query batch query and store the output in a file. If that new file is placed in a library to which the previously unauthorized users have access, they will now be able to view the data in that file. To prevent this situation with query output to a batch file, you should implement at least one of the following suggestions:

- Choose the server option to exclude users from libraries by default.

- Specify the name of the library to which a user can write batch output using the Resource Settings option for Query or Report Writer, and then ensure that library is secured from other users.

The same situation could happen with Warehouse Builder target files. It is recommended that you exclude users from access to these applications. Only the person who deploys data warehousing or administers database access should use Warehouse Builder.

Making Sure That Existing Data Is Not Overwritten

You can control the security required to clear the target table when running a Query or Report Writer query in batch mode. This is to prevent a user from accidentally replacing a production file to which they have authority to update.

There are two modes for handling batch output security:

- You must be the owner of the batch target table or have `Manage` authority to the target table before being able to replace the results.
- You must have system object-level security and at least `Read` authority to the target table before being able to replace the results.

To decide which mode to use, first examine the files on your system:

- If many of your production files have `*ALL` or `*CHANGE` authority for your query users, you should consider using the first mode to gain additional Warehouse Manager security. This way a user will not be able to accidentally replace production data with the results of a query.
- If you have appropriate object-level security, you may disable the additional Warehouse Manager security.

To change batch output security

1. Open Warehouse Manager Client and from the Manage menu, choose **Server Options**.
2. Select the **Batch** tab in the Server Options dialog box.
3. Select one of the following options to determine who is allowed to clear a result table:
   - The owner of the table or any user with `Manage` authority to the table.
   - Any users with iSeries `*CHANGE` authority to the table.

Controlling Query/Report Writer System Usage

One aspect of improving your overall system performance is to control how Query and Report Writer use system resources. This means that you must manage certain aspects of your iSeries system to control the impact your PC applications will have on your iSeries system. This can be done by setting maximum query run limits, setting server job run priorities, and controlling the level of parallelism on DB2 Symmetric Multiprocessing systems. These performance improvement methods are generally implemented by using Warehouse Manager Client.

Setting Maximum Query Run Limits

Using the query run limit, you can protect your system from long-running or ill-formed queries. Query run limit support is built on the predictive query governor support of OS/400. Using Warehouse Manager, you can set the maximum run length of a user’s query. If the query optimizer on the iSeries predicts the query will take longer than the limit, it stops the query *before* it is run.

Advantages

Setting query run limits allows you to control the system resources that any query user is allowed to use. It also prevents system resources from being consumed by a single, ill-formed query.

Disadvantages

It limits the user to running queries that meet the criteria set by the database administrator.

Since the query optimizer is making an educated guess about the run time for a query, it may not be completely accurate (the query could actually have a shorter or longer run time than the estimate given).

Assigning Query Run Limits

To determine the query run limits, you should consider the types of queries that your users run and the appropriate protection that you need to provide for the overall system. To make this analysis:

1. Determine the amount of time that it takes to run several typical queries. You can use the Performance Analyzer feature in Query to estimate the amount of time a query will take to complete, or you can use auditing in Warehouse Manager Client and run the Detailed Select Statement query.
2. Determine the other requirements of your system and the job run priorities established earlier.
3. Using this information, break your queries into two groups:
   - The set of queries that completes quickly enough to be run interactively.
   - The set of queries that should be run in batch.
4. Use Warehouse Manager to set up the appropriate query limits for `*PUBLIC`, individual users, and groups of users.

Example

In this example, you want to protect the system from CPU-intensive queries. Use the Performance Analyzer in Query to estimate the run limit on a set of queries commonly used by a user named `QUSER`. Once the analysis is made, you will set up the following limits for user `QUSER`:

- For any queries estimated to run less than 5 minutes, run them interactively at priority 50.
- For queries estimated to take more than 5 minutes, run them in batch at priority 50.
- If the query is estimated to run more than 30 minutes, it will not be allowed to run.

Verifying the Run Limit

The query run limit is enforced by estimating how long a query will run. Since this is only an estimate, some queries that take longer than the limit are allowed to run. Also, some queries may not be allowed to run even though they will finish within the specified time. To verify whether a query will run within the limits, use the Performance Analyzer in Query, which will inform you of the query’s estimated run time.

Setting Server Job-Run Priorities

The job-run priority is used to prioritize the server job relative to other work on the system. The job-run priority is the standard iSeries work management run priority. The lower the number, the higher the run priority.

Advantages
Provides control of system resources by prioritizing jobs relative to other work on the system.

Disadvantages
Other work can starve the query if it has a higher priority.

Assigning Job-Run Priorities

1. Consider all work performed on your system and rank it by importance.
2. Based on ranking, choose the job-run priority that you want to use when running queries.
3. Using Warehouse Manager, assign job-run priority for `*PUBLIC`.
4. You may further refine priorities by assigning job-run priorities based on user ID or groups of users. This allows you to prioritize query usage among users.

Example

After evaluating your system, you must run the following jobs:

- Order-entry application
- End-user queries
- Nightly batch, order processing application

Based on this, you assign the following job-run priorities:

- Order-entry application (priority 20)
- End-user queries (priority 20-30)

- Nightly batch, order processing application (priority 50)

The order-entry application and batch processing application already had the run priorities defined. For queries, you defined a range of priorities so that you can further prioritize individual users as needed.

Using Warehouse Manager, you set up a general run priority of 20 for interactive queries and 30 for batch queries. By setting the value for *PUBLIC, you are applying it to all user profiles on the system. On some later date, you may define specific run priorities for individual users and groups of users. These will then override the *PUBLIC value.

Additional Information

Once the server is started, the job priority may be changed using the CHGJOB CL command. When a user connects to the iSeries or submits a query to batch, the following shows which run priority is used:

- Warehouse Manager setting for the user profile.
- Warehouse Manager setting for the group profile if the user is a member of a group.
- Warehouse Manager setting for *PUBLIC.
- The class description used by Warehouse Manager Server.

If the run priority does not exist at a particular level, the next level is used. In the previous example, you would use the value of *PUBLIC because there are no settings for the specific user profiles or the user's group profile.

For batch jobs, the Warehouse Manager setting is determined when the batch job is submitted, not when it is run.

Controlling Level of Parallelism

This section applies only if you have the DB2 Symmetric Multiprocessing for OS/400 feature on your iSeries. Controlling the level of parallelism will determine whether the system will allow a query to be logically split into multiple, independent tasks that can be carried out in parallel.

The iSeries system value QQRYDEGREE can be used to tune the level of parallelism when running queries on your system.

Assigning the Level of Parallelism

1. Consider all work performed on your system and decide if queries can be allowed to consume additional CPU and I/O resources in order to finish quickly.
2. Decide on the amount of parallelism that you are going to allow. Currently, the system value gives you the following control:

*NONE Do not allow any parallelism. Queries are not the highest priority on the system and you cannot afford to allow the system to devote the resources necessary to an individual query.

*IO Allow IO parallelism. You are willing to let the system use additional IO and mainstore resources to complete a query faster. In some cases, allowing the system this freedom to use more resources while running a query will improve the cumulative total resources used.

*OPTIMIZE
Allow the system to choose whether IO and CPU parallelism should be used to improve query performance and overall system performance. In some cases, allowing the system to use more resources while running a query will improve the cumulative total resources used.

*MAX Allow IO and CPU parallelism. You want parallelism to be used whenever possible if it will help improve the individual query. You only use this system for running queries, so there is little impact on other types of applications.

3. Once you have set the level of parallelism, review system impact and adjust it as necessary until you have the setting that works best for your environment.

Example

You have three systems in your company. You run queries on all systems, so you bought DB2 Symmetric Multiprocessing for OS/400 to get the additional data access methods and now want to use the appropriate level of parallelism. The following shows how each of your systems is used:

SYS-A You run an order-entry application on this system and run occasional queries.

SYS-B This is a development and test system with which you enhance your application and test your queries.

SYS-C This machine is a dedicated data warehouse system and is where most of your queries are run.

Based on analysis, you choose to set the QQRYDEGREE system value as follows for each of your systems.

SYS-A=*NONE
Your order-entry application is your first priority. You cannot afford to have a single query use a majority of the system resources even for a short time.

SYS-B=*OPTIMIZE
You let the system decide. You have no mission critical applications running on this system, and you want to test queries that sometimes use parallelism.

SYS-C=*MAX
This system is used only for decision support. You want the queries to be fast, and you are willing to allow the query to use any resources necessary to complete.

Enabling Expert Cache

Expert cache is a set of improved paging algorithms used for database files and related objects. By enabling expert cache, you are improving query completion time and reducing system paging and faulting rates. Enabling expert cache should be the first step in tuning your system for running queries.

To Enable Expert Cache

1. Determine the storage pool that the Warehouse Manager Server job is using. Normally, this is *BASE (storage pool 2).
2. On the iSeries, use the WRKSYSSTS command with the assistance level set to 2=Intermediate.
3. Press F11 until you see the heading **Paging Option**. Enter *CALC over the top of the existing value to enable the expert cache algorithms. Press Enter.

You can also use the CHGSHRPOOL command.

Tuning System Performance

System performance tuning refers to adjusting pool sizes, time slice, and subsystem allocations to improve response time, throughput, paging rates, and contention.

System performance tuning can improve overall usage of system resources and individual query response times. However, in most cases, you must tune the *entire* system to realize the benefits. For this reason, we recommend the following:

- Always consider the entire system and all system work when adjusting system-wide resources.
- Never adjust system resources for the server without considering the rest of the system load.
- System tuning starts with system performance analysis. Do not tune the system without first completing this analysis.

- Tune system performance *after* following the other performance suggestions in this section.
- In most cases, it is best to hire a consultant who specializes in system tuning. System tuning can be an exercise in trial and error. A good consultant already knows what works and what does not in many situations.

If you decide that system-level performance tuning is right for your system, see the following sources for information to help you get started:

- IBM *OS/400 Work Management Guide*
- IBM *Performance Tools/400—Getting Started*
- An iSeries consultant specializing in system performance

DB2 Symmetric Multiprocessing for OS/400

The DB2 Symmetric Multiprocessing feature improves query performance by adding the following to DB2/400:

- Additional data access methods and optimizer improvements
- CPU parallelism for query processing
- Support for the `QQRYTIMLMT` and `QQRYDEGREE` system values

Warehouse Manager Server takes advantage of the functionality provided by this feature. And while its name indicates that it is for multiprocessor systems, it also provides improvements on single-processor machines.

Using Auditing to Understand Your Queries

STRATEGY can provide you with information about the impact users have on your system when accessing the iSeries relational databases using SQL. To collect this information, you must use Warehouse Manager Client and choose **Auditing** from the Manage menu. This information can help you make tuning and performance decisions about specific queries that your users are running.

Impact to Overall Performance

Turning on the auditing feature will have an impact on your system's performance, and this may be noticeable to end users. Auditing works by collecting information on every query run and writing that information to a database file; both the additional overhead of collecting information and storing output operations to disk can affect your performance. Because environments differ greatly, it is difficult to predict the exact degradation that you might see.

Because of the possibility of performance degradation, we suggest that you turn on auditing for a limited time only. Collect information for a period of time, turn auditing off, and analyze what you've collected. Do not leave the auditing feature permanently turned on. Rather, auditing should be used as part of a planned cycle of information collection and analysis.

Collecting the Information

1. Using Warehouse Manager Client 3.0 or greater, connect to the iSeries data source (Warehouse Manager Server at release 3.0 or greater) in which you would like to audit your users' queries.
2. From the Manage menu, choose **Auditing.**
3. Set the options to collect data for a specified period of time or until the auditing output file reaches a specified size, and then click **Start Auditing**. If you decide to end auditing before the specified time or file size is reached, click **Stop Auditing**.

After you start auditing, information is collected on SQL SELECT statements for each user who connects to the server library in which you started auditing using an ODBC connection. These will probably be Query and Report Writer users. Local distributions run by Warehouse Builder will also be recorded. STRATEGY applications released prior to release 3.0, including Data View Manager and Analyzer Data Modeler, will also be audited if they connect to a 3.0 server library in which auditing is started.

Note that some applications, including Warehouse Builder and Warehouse Manager Client, use SQL statements internally to process user requests. In order to provide an accurate picture of the impact applications have on your SQL usage, these statements are also audited.

In some cases, Warehouse Manager Server is able to optimize a user's Query or Report Writer query so iSeries SQL is not used. In these cases, you will not see that user's query recorded in the auditing file, because it did not affect SQL performance.

Analyzing the Information

After auditing, it is time to analyze the information. Because the analysis involves running SQL queries against the database of information collected, it is recommended that you begin analysis only after auditing has ended. Otherwise, your own queries on the auditing information are added to the database file as you analyze it. This may skew your view of your users' actual SQL usage.

In Warehouse Manager Client, you will find several queries designed to help you analyze this data. From the Reports menu, choose **Audit Querie**s. Each query is provided in both a Query (*.dbq)* format and Report Writer (*.rpt*) format. To use these queries, you must have at least version 3.0 of Query or Report Writer. An explanation of each query follows, with some suggestions on how to use them.

Select Statements: Basic Information

This report returns basic information about the queries run while auditing was turned on. Use it when you simply want to see which users are running SQL SELECT statements and what the statements are.

Select Statements: Detailed Information

This report returns detailed information about the queries run while auditing was turned on. Use it when you want to start analysis of the SQL SELECT statements that have run on your system. You may choose to do further analysis after seeing detailed information on the STRATEGY applications used, the time to run statements, the query optimizer's access methods, etc.

For each query run, there is at least one record in the report per file used in the query. Each record shows which access method the DB2/400 optimizer chose to use when retrieving your data from that file. Always note the job number and unique statement number of the query you are interested in so you can find information about that particular query in the other reports.

Ordering

For different analysis purposes, you may want to order the data in this report differently than the way it is presented. Because this report, in the Query format, uses Break Groups, there is a special way to do this. To order by different columns in the group, from the Format menu, choose **Break Groups** and drag the column by which you want to order to the top of the Break Groups list.

Timings

The detailed information report includes the run time for each query. The time reported is not "wall clock" time, it is an internal processor time as reported by DB2/400. This information is best used for comparisons between queries.

This time may not match exactly a time reported by Query or Report Writer performance analysis. Again, the auditing time is an internal database-reported time, while the performance analysis is an estimate of all processor time used by the application. All comparisons made between queries should be done with timing reports from the same tool, either auditing or performance analysis, but the results of the tools should not be mixed.

You may notice that the same SELECT statement has varying run times. The reason for this might be the system environment, including the load on the system at the time the query was run and the job priority given to the user. Another reason might be that the SELECT statement was run multiple times in the same job, and the application was able to reuse the statement without going through some of the overhead associated with its first run.

When users query views instead of files

Note that if users are directly querying a view or logical file, this report shows that view as the name of the file used. Similarly, other auditing queries count that view as the “file used” for the purpose of most frequently queried files. If you need to find which physical file is actually being used by a view or logical file, add columns `QQPTLN` and `QQPTFN` to your report.

Multiple records for a queried file

You might see more than one record for a file in this report because it might be joined to itself, so it actually was used more than once. Or the query might have been implemented with an encoded vector index (EVI), in which case the access plan is recorded as both an arrival and an existing index. You may use the Queries Which Used Existing Indexes query to verify whether an EVI was used. Another scenario is that the filename is actually a view or logical file that references multiple physical files.

Full Select Statement

This report returns the entire text of the `SELECT` statement. You may need to use this report if a `SELECT` statement is longer than 1024 characters. The auditing file breaks `SELECT` statements into several records if the text exceeds 1024 characters.

This report is most useful if you use the Report Writer (*.rpt*) version rather than the Query (*.dbq*) version because the *.rpt* allows you to see multiple lines of the statement text.

Queries That Build Temporary Indexes

Temporary index builds may be of interest to administrators because often they are time consuming. Use this report to find which queries the DB2/400 optimizer chose to build temporary indexes for, and whether the optimizer advises building a permanent index.

Queries That Used Existing Indexes

This report returns the list of all SQL queries that used an index already on the system. You may be interested in this information if you created indexes specifically for the purpose of improving a query’s performance, and you want to verify the index was used.

You may notice for some queries that the index name and library are the same as the filename and library. In this case, it means the key on the physical file was used to run the query.

Indexes Advised

The DB2/400 optimizer sometimes advises you to build a permanent index to assist in query performance. This report tells you which queries had indexes suggested for them, what files the indexes should be built over, and what the key fields of the indexes should be. By building the suggested indexes, you may improve your users' query performance.

Most Frequently Queried Files

This report returns a list of the files most often specified in your users' SELECT statements. Use it when you want to know which are the most "popular" files to query. This may assist you in making information more readily available to users or in deciding how to optimize your environment.

Which Queries Use a Given File

If you see something in the previous report that makes you wonder exactly why a certain file is being queried so often, you may use this report to find out exactly which SELECT statements ran against that file and which users ran them. To use this report, you will need to enter the name of the file that you are interested in and its library.

Note: The information you enter in the prompts is case sensitive.

Query Governor Information

DB2/400's predictive query governor prevents queries from running if it estimates that they will take longer than a time value that you have specified. This report shows whether the query governor was active at the time a query was run and whether any users' queries were prevented from running because the predictive query governor estimated their processing time to be too long.

Run this report when you've used Warehouse Manager Client's Resource Settings to set Query Limits for a user. Setting Query Limits for a user enables the DB2/400 predictive query governor when that user connects to the server. This report can help you see whether you are preventing too many queries from running because the Query Limit is set too low.

Additional Analysis Tips

If you need more detailed information about a query than you can get from the auditing queries, you can cut the SELECT statement out of the report (this is easiest using Report Writer) and paste it into the SELECT statement edit box in either Query or Report Writer. Then choose **Performance** from the Run menu to get detailed messages about the access paths considered, temporary indexes or result sets built, or indexes used.

Managing Your Auditing Environment

Saving Auditing Results

When you start auditing, if you select the Result Data option to `Overwrite existing file` instead of `Append to existing file`, any previous data collected is cleared. If you would like to save this previous data for reference at a later date, follow these steps before restarting auditing with the Overwrite option:

1. Use the iSeries CRTDUPOBJ command to create a copy of the file *SCDBMON*, which you will find in the server library in which auditing started. Be sure to specify *YES for the Duplicate data option.
2. In the same way, make a copy of the file *SCAUDINF*, also found in the server library.
3. Make a note of the names of your new files and their libraries.
4. Open the auditing query or report that you want to run against the copied data.
5. From the File menu in Query or Report Writer, choose **Save As** and save a copy of the query or report.
6. Using your new, saved copy of the query or report, choose **Tables** from the Query menu. For each reference to either *SCDBMON* or *SCAUDINF* in the query, click the table name, and then click the **Change Link** button. In the Change Link dialog box, enter the library name and the filename of the copy of *SCDBMON* or *SCAUDINF* that you made earlier. (You may leave any references to *SCAUDMAP* alone, because that file is neither cleared when you begin auditing nor updated when users run queries.)
7. Repeat step 6 for each query or report that you need to run against the saved audit files.
8. Remember to delete your copies of *SCDBMON* and *SCAUDINF* when you are finished with them. The copies that you save are no longer under the application's control so Warehouse Manager Server will not automatically clean them up.

Subsequent Installations

If you reinstall Warehouse Manager Server in a library where auditing has been run (for example, to upgrade the library to the next release of Warehouse Manager Server), be aware that your auditing information is not preserved during such an installation. If you want to save your data, you may follow the steps in the previous section, being sure to save the files in a library other than your server library.

Size of the Auditing File

The auditing file *SCDBMON* can become quite large. The record length of the file in V4R3 and V4R4 is 2958 bytes, and each SELECT statement can generate several records.

While you do have the ability to specify that auditing should end when the file reaches a certain size or when a given time period expires, be aware that any users who are currently connected when these parameters are reached will continue to have their queries audited and information recorded in the auditing file until they disconnect.

Upgrading OS/400

Auditing should always be ended before you upgrade your version of OS/400. If you want to preserve any previous auditing information, follow the steps at the beginning of "Managing Your Auditing Environment" on page 22. When you restart auditing after the upgrade, choose **Overwrite existing file** for the Result Data option.

Improving Query Performance

After improving your overall system performance by controlling Query and Report Writer system usage, it is possible to fine tune your queries by taking advantage of methods used by the iSeries to process queries and by following some general query creation tips.

Understanding the Query Optimizer

The iSeries uses a sophisticated piece of software called the **query optimizer** to efficiently retrieve end user and applications data. Understanding how the query optimizer processes queries will help you create queries that will run efficiently, since the query optimizer's primary function is to turn an SQL statement into a set of structures that enable fast and efficient data retrieval from the iSeries. Like most optimizing software, query optimizer produces a query as follows:

- Breaks the SQL statements into functional "chunks."
- Validates each chunk, for example, verifies the existence of the files and fields referenced in the query.
- Optimizes the query.
- Builds structures needed by database run-time code for data retrieval.
- Passes those structures to the database run-time component.

One of the key outputs of an optimized query is a control structure called an **access plan**. An access plan is a bookkeeping structure for tracking the objects used by DB2/400 and determining how the database functions will access and process each record in the database.

A key piece of information held by the access plan is the access path to be used by the query. The access path refers to the method(s) that the database management system (DBMS) used to perform the query. View the access plan as a road map and the access path as the road itself.

During the optimization phase, the query optimizer will evaluate many alternatives to implement a query. These alternatives may use a single access method or a combination of methods. After coming up with a list of possible access plans, the “cost” of each access plan is calculated based on the expected amount of CPU and I/O work each requires. Once the cost of each plan is available, the query optimizer chooses the cheapest method for processing the query.

For a more detailed explanation of how the query optimizer processes information, see the IBM document *DB2 for OS/400 Database Programming*.

Using Query’s Performance Analyzer to Evaluate Your Queries

Improving query performance can often be an exercise in trial and error. You may need to analyze your query, make changes, then analyze it again until you have achieved an acceptable level of performance.

To tune your query

1. Understand how the query optimizer chose to implement the query.
2. Analyze the run-time performance of the query and determine if it is I/O or CPU bound.
3. Based on the results from steps 1 and 2, experiment with the techniques described in the remainder of this document to eliminate unnecessary CPU processing and disk I/O.
4. Go back to step 1 until the performance does not improve or all possibilities have been attempted.

The most important step in improving query performance is to understand what choices the optimizer made for the query. Once this is understood, you can experiment and influence the optimizer’s future decisions.

Query provides an easy way to analyze your query through its Performance Analyzer feature.

To analyze your query

1. After creating your query, from the Run menu, click **Performance**. The Performance dialog box appears.
2. Click **Analyze** to analyze the efficiency of your query.
3. Once the analysis is complete, click the **Log** tab to view the implementation messages for your query. The meaning of each of the optimizer messages can be found in the Performance Analyzer online Help.

To effectively use the detailed information provided by the log, see the following sources:

- Online Help. The online Help contains a description of the provided information along with some general performance-tuning tips.

- IBM document *DB2 for OS/400 Database Programming*
- IBM document *DB2 for OS/400 SQL Programming*

The Performance Analyzer feature of Query allows you to quickly and easily test queries before they run to determine how they were implemented and the estimated run time. This estimate is provided by the DB2/400 query optimizer and is fairly accurate. However, as mentioned previously, the optimizer does not always have adequate information for an accurate estimate. Note that Performance Analyzer is also available in Report Writer.

Creating Indexes to Improve Performance

Creating indexes can dramatically improve query performance. However, creating indexes will also slow the performance of OLTP applications (for example, order entry) because the index is updated for every insert, update, and delete operation. Indexes built over large files will also consume additional DASD.

When should an index be built? It depends on the importance of query performance weighed against that of OLTP performance, unless you have separated queried data from updated data (a data mart or data warehouse environment). The rest of this section will focus on creating indexes to help query performance. The trade-offs should be considered before creating indexes for production files.

To determine what kind of indexes will help your query performance:

1. Look at 5–10 of your most resource-intensive or frequently run queries. Using the performance analyzer, go through the query optimizer messages as described in the previous section.
2. Identify the following attributes of these queries:
   - Join conditions
   - Group-by columns
   - `WHERE` clauses
3. If the query messages gathered in step 1 point to join conditions or group-by clauses as the source of spontaneous index builds, seriously consider building permanent indexes above those files. If an index does not exist for a join condition, it is probably best to build an index by defining a join-logical file. Join-logical files not only supply the necessary index, but they eliminate the need for your end users to specify a join condition in their query.
4. Examine your `WHERE` clauses. Pick out the most selective parts of the `WHERE` clause—that is, those that tend to eliminate the most records from the file. If those conditions are typically joined by `AND` with the rest of the `WHERE` criteria, those fields/columns are usually the best candidates for a permanent index. If multiple fields are usually joined by `AND`, the index

should usually be built with those fields in the key. Multikey indexes should be ordered from most selective to least selective WHERE criteria.

The following documents provide more information on creating indexes:

- IBM document *DB2 for OS/400 SQL Reference*
- IBM document *DB2 for OS/400 SQL Programming*
- IBM document *DDS Reference*

The following examples present sample SQL statements and the types of indexes that might improve their performance:

Example 1

```
SELECT partnum, partname FROM orders WHERE orddate > 960530
```

The candidate for an index is orddate. The index should be created when queries referencing orddate eliminate 80% of all rows.

Example 2

```
SELECT custnum, partname FROM orders WHERE orddate > 960601 AND custnum = 976771
```

The candidate index would have key fields of custnum and orddate. The order of the key fields will depend on which is most selective (put the most selective first).

Example 3

```
SELECT ordnum, custname, partdesc FROM orders, parts WHERE orders.partnum = parts.partnum
```

The candidate index for partnum would be in either or both files. It could create a join-logical file instead of an index for even better performance.

Example 4

```
SELECT custname, sum(price) FROM orders GROUP BY custname
```

The candidate index is custname.

Example 5

```
SELECT custname, sum(price) FROM orders WHERE custnum = 976771 GROUP BY custname
```

The candidate index is custnum if the WHERE clause is very selective.

Note: To perform the group-by clause, the system builds an index at run time if one does not exist over custname. This is fast when custnum is very selective because the existing index for custnum contains few records. The index for custnum is used to build the index for the group-by clause.

Changing ODBC.INI Settings

Several settings in the *ODBC.INI* file also influence the performance of Query. The first setting is SQLOptimizationLevel. It can be set to one of two values: 0 or 3. The default value is 3, which tells the STRATEGY components to optimize the query so that it fully utilizes all available APIs on the iSeries to minimize query run time. A value of 0 indicates that the query should be passed directly to the SQL components on the iSeries. There are some queries that will run better if the query is passed to SQL. While the recommended setting is 3, this is another technique that might help if other approaches do not work.

Another *.INI* setting is AllowParameterization. The default value for this option is Yes, which allows the ODBC driver to parse your SQL statements and to replace the literal values found in these statements with parameter markers. This parameterized statement is prepared and held for reuse during your Query session. If the statement is being run more than once during a Query session and only the literal values used in the statement are changed, the default option of Yes will help you avoid the constant SQL overhead at run time. If the majority of your statements are run once during a Query session and the literal values used in the statement are not the only values changed, then No for this option will help the iSeries Query Optimizer make better estimates and decisions on your SQL, improving run time.

Query Creation Tips

Always Specify Join Conditions

If the query contains more than one table, it is almost always necessary to specify a join relationship between the files. If no relationship is defined, the system will perform a Cartesian product join that matches every row in one table with every row in another table. The typical result returns extremely long response times and large amounts of (nonsensical) data.

Limit the Use of Deeply Nested Subqueries

Subqueries complicate query optimization. As a result, the optimization phase has more opportunity to build an inefficient access plan. In general, it is a good idea to break the statement down into simpler queries that are run back-to-back if possible.

Limit Use of Data Type Conversion

Whenever possible, avoid forcing the system to do data-type conversions. For example, if you have a WHERE clause, such as WHERE INTEGER = 1.4, the system must convert the INTEGER field to another data type to accurately compare it to 1.4. The optimizer will not use any index built over INTEGER.

Avoid String Padding

In cases where you compare a string literal to a field, ensure the length of the literal matches the length of the field. If the literal is longer, the system must convert the field to the longer length (and pad with blanks) to do an accurate comparison. As with numeric conversion, this eliminates use of any index built over the character field. A shorter literal string will not cause problems, since the system will pad the literal appropriately to use the index.

Minimize the Use of LIKE

Avoid using pattern matching when possible. Particularly bad is the use of % or _ at the start of a pattern. Since the optimizer depends on the use of leading key fields in indexes, the use of a leading wild card character eliminates index use in these situations.

Avoid the Use of Arithmetic Expressions in WHERE Clauses

The use of arithmetic expressions often eliminates the use of an index that would otherwise apply to the selection criteria.

Optimizing Performance for Warehouse Builder

Following are several ways to improve the performance of your Warehouse Builder relational database distributions.

Choose the Most Efficient Type of Distribution

There are several types of distributions that Warehouse Builder provides. The various types of distributions are shown in Figure 1. The type of distribution that you choose will have a significant impact on the performance of your distribution and its affect on your iSeries system. The performance characteristics of the various types of distributions are described in Table 1.

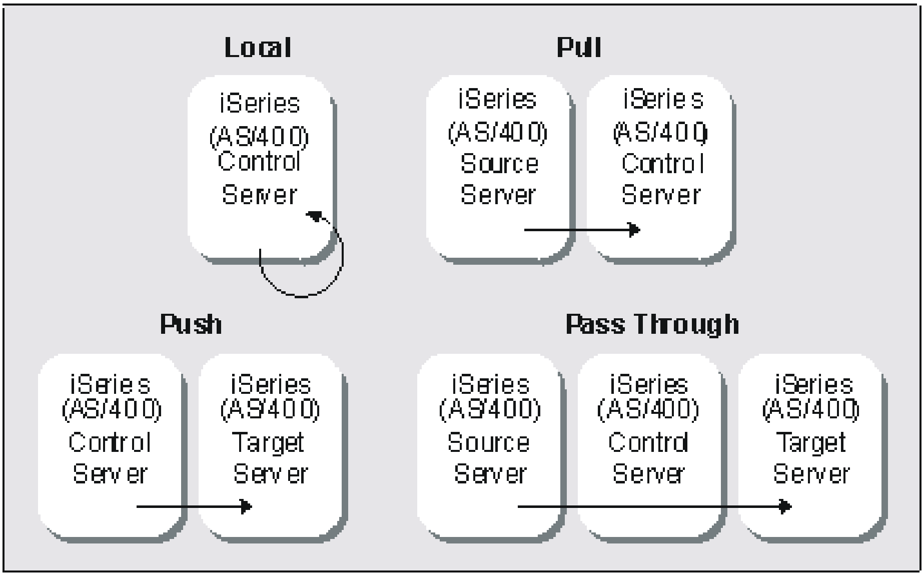

Local
iSeries (AS/400) Control Server
Pull
iSeries (AS/400) Source Server
iSeries (AS/400) Control Server
Push
iSeries (AS/400) Control Server
iSeries (AS/400) Target Server
Pass Through
iSeries (AS/400) Source Server
iSeries (AS/400) Control Server
iSeries (AS/400) Target Server

Figure 1. Warehouse Builder distribution types

Table 1. Distribution types and performance levels

Type of distribution	Performance level
Local (within the same machine)	Very good
Pull (from a remote iSeries data source to a target table on the control server)	Very good
Push (from the control server to a remote iSeries)	Not very good
Pass through (from a remote iSeries to another remote iSeries)	Not very good

Use Pull Distributions

By far, the fastest type of distribution is a pull distribution. Pulling data from a target system is 10 to 15 times faster than pushing data to another system. This is due to the lack of blocking support in DRDA when doing inserts into a remote file. Therefore, we recommend that all performance-critical distributions pull data from the source system. This means the control server and the destination server should be the same system.

Creating Libraries

To store distributed data, Warehouse Builder will create libraries on the target system by default, improving distribution performance. To use journaling, you can specify that Warehouse Builder create collections instead. However, using collections and activating journaling may dramatically reduce distribution performance.

To create collections or libraries with Warehouse Builder

1. From the Tools menu, choose **Options.**
2. Select the **Default** tab.
3. Select **Create Target SQL Collection**. If this option is not selected, Warehouse Builder may create libraries.

Using CLRPFM Command Instead of Mass Deletes

When the target table is defined in Warehouse Builder, you can specify that the target table is deleted on each distribution. There are two methods to accomplish this: SQL `DELETE` or `CLRPFM` (Clear Physical File Member).

When you define in Warehouse Builder to delete the target table, an SQL `DELETE` statement is issued to assure that the client code will work against any SQL data source. The SQL `DELETE` method is slower than `CLRPFM` but is applicable with all targets.

On the iSeries, the other method to remove all records from a file is to issue a `CLRPFM` command. `CLRPFM` is a faster method for deletion, but it is only valid with iSeries targets. To enhance performance, use the `CLRPFM` command before the data distribution is started. `CLRPFM` has two other distinct advantages:

- Removing old records is extremely fast and consumes few system resources.
- Disk space used by the deleted records is recovered.

To specify how to delete the target table

1. From the Tools menu, choose **Options**.
2. Select the **Default** tab.
3. In the Table Overwrite section, select the **SQL DELETE** or the **Clear File Member Command** option.

Sending Only Changed Records

Another way to improve distribution performance is to send only changed records between the source and target servers. While there is currently no automatic method to send only changed records between a source and target system (for example, as with products having journal-based replication services), the following paragraphs describe several methods to facilitate sending only changed records if you have a file getting only inserts and not getting updates or deletes.

Using CURDATE()

If there is a date in the file(s) referenced on a source system, you can use a variety of built-in SQL functions to determine what data are new since the last distribution.

For example, if you move data nightly from an orders file and that file has a date (either a date data type or a date that can be manufactured out of several fields and converted to a real date), then you can compare the date in the record with the current date (for example, ORDERDATE = CURDATE()).

Using a Sequence Number

If records on the source system have a field containing a sequence number (perhaps used as a key) that grows in value as new records are inserted, then you may be able to use this field to select only the recently inserted records. To use this method:

1. Create a new file on the source system that has a field representing the last_sequence_number. Initialize this field to 0.
2. When creating a definition, join it to the table created in step 1 and specify a WHERE clause to select only the records that are greater than last_sequence_number.
3. Create a post-SQL statement that selects the maximum sequence number from the source tables and updates last_sequence_number.

This method assumes that no new records are being added to the source table between the time the main transfer runs and the time the post-SQL statement is run to update last_sequence_number.

Other Performance Tuning Issues and Solutions

The efficiency of the SELECT statement running on the source system will have a direct influence on both the distribution time and the amount of system resources used. Therefore, when SELECT statements run against large files, it is recommended that you first use the Performance Analyzer feature of Query.

- Determine if the statement is optimized and/or how long, roughly, the query will run.

or

- Turn on auditing and run it as a local distribution or in Query to get the auditing information logged.

Based on this information, you may either adjust the SQL statement, build indexes to speed up the query, or perform the query in multiple passes (for example, divide the query into several smaller queries to reduce the complexity of the statement or perform more of the work on the target system instead of the source).

Chapter 3. Managing Data on the iSeries

Warehouse Manager Server supports STRATEGY PC applications. Data produced by STRATEGY applications is stored on the iSeries. You will need to perform the following tasks to protect this data:

- Move data from a test to a production environment
- Back up data
- Change physical files that have dependent data views
- Change data views that have dependent data views
- Manage passwords
- Manage information workflow

See Chapter 4, "Warehouse Manager Server Maintenance" on page 49 for more information on Warehouse Manager Server maintenance.

Moving Data from a Test to a Production Environment

Query, Report Writer, and Warehouse Builder give you the powerful advantage of developing queries, reports, data models, data views, and data distributions directly against iSeries or select NT or AIX databases. The environment in which you develop these files is critical to the success of daily database operations. If you develop these files in a production environment, you could adversely affect the performance of other operations. To prevent this problem, develop files against test data, either on a separate iSeries machine or in a separate library. Once testing is complete, you can put your queries, reports, data models, data views, and/or data distributions into production.

There are various ways to move data from a test environment to a production environment, as described in the following sections.

Saving and Restoring Data

This is a global approach, allowing you to back up all data residing on the iSeries. This approach is the most efficient, because you can move all information at once, but it limits you from moving individual queries, reports, data views, and so on. Use the Save/Restore procedure to save and back up server information from one library and restore it to another. The target library can be on a different system or on the same system. See "Backing Up Data" on page 35 for backup recommendations.

Redirecting Data

With this approach, you redirect a query, report, or data view from a test environment to run in a production environment. You do this on a file-by-file basis, which is helpful if you do not want to move all data at once through the Save/Restore procedure. However, this method requires more planning and structure. The key is to keep the structure of your libraries, tables, columns, and column data types consistent in both your test and production environments.

Sometimes, it’s necessary to modify the database structure by renaming libraries or tables or by removing or renaming columns. However, we recommend that you keep structural modifications to a minimum. If you are performing many structural modifications, it is wise to do a thorough database design review. Existing queries, for example, may cease to run after these kinds of changes. Change Link error messages are invoked when PC applications determine that libraries, tables, or columns are missing, or if column data types have changed. Consider the following two scenarios to help you decide how to structure your iSeries database:

Scenario 1: Testing and Production on Separate iSeries

If testing and production are performed on separate iSeries, maintain an identical database structure on both systems. That is, the library, table, and column names should match, and the column data types should be equivalent on both systems (for example, numbers to numbers or characters to characters).

System		Library	Table	Column
From	SYS1	FY02	CUSTOMERS	CUSTNAME
	SYS1	FY02	ORDERS	SHIPDATE
To	SYS2	FY02	CUSTOMERS	CUSTNAME
	SYS2	FY02	ORDERS	SHIPDATE

After testing is complete, use the Change Data Source feature to switch a query, report, data view, data model, or data distribution to run on the production system. This feature is available in STRATEGY PC applications. Use the Performance Analyzer feature to evaluate the performance of a query on your production iSeries. This feature is available in Query and Report Writer.

Scenario 2: Testing and Production on the Same iSeries

If testing and production are performed on the same iSeries, maintain a set of test tables in a test library, parallel to the production library. The same concerns regarding matching names and data types apply here, except for the library names, which are different

System		Library	Table	Column
From	SYS1	SAMPLE	CUSTOMERS	CUSTNAME
	SYS1	SAMPLE	ORDERS	SHIPDATE
To	SYS1	FY02	CUSTOMERS	CUSTNAME
	SYS1	FY02	ORDERS	SHIPDATE

After testing is complete, use the Change Link feature to convert a query, report, data view, data model, or data distribution to run against the production library. This feature is available in STRATEGY PC applications.

Creating Custom Library Lists for Users

Another approach for moving data from a test to a production environment is to create custom library lists for users. In this case, developers or testers run against a library list that points to test libraries; users run against a library list that points to production libraries.

To give users access to only those iSeries libraries in their library lists:

1. Open Warehouse Manager Client and from the Manage menu, choose **Server Options.**
2. In the Server Options dialog box, select the **Library Access** tab.
3. Select **Return only libraries in user portion of the library list**.

This simplifies the database for users because all the data they need is contained in only a few libraries. However, it is difficult to troubleshoot database errors because tables are implicit in the libraries, thus making the exact location of data less obvious.

Backing Up Data

We recommend that you periodically back up Warehouse Manager Server. The purpose of the backup dictates how you back up the data. Will you be backing up temporarily so you can, for example, move server information between systems? Or will you be backing up as part of your disaster recovery plan?

Temporary Backups

The following situations require you to back up Warehouse Manager Server information temporarily:

- You need to install a new release in the Warehouse Manager Server library.
- You need to delete the Warehouse Manager Server library but want to save the library’s contents, such as security settings, queries, and data views.

- You need to move the Warehouse Manager Server library from a test system to a production system.
- You need to move Warehouse Manager Server contents from a test library to a production library on the same system.
- You need to distribute the data after you purchase a license for an additional iSeries.

For any of these situations, use the Save/Restore procedure to save and back up server information from one library and restore it to another. (See "After Restoring Server Information" on page 38 for configuration issues that you will need to consider once Warehouse Manager Server information has been restored.) The target library can be on a different system or on the same system.

Using this procedure, you have the choice of saving server information to a file or a device such as a tape. In these situations, we recommend saving to a file. If you are moving data between systems, it is easier to send files electronically, especially for remote sites. If you cannot send files electronically between systems, you can always back up to a device.

Disaster Recovery Backups

To back up Warehouse Manager Server information for disaster recovery purposes, you should follow the same procedure as for other iSeries backups. Whether you use the iSeries `GO SAVE` command to back up the entire system or the `SAVLIB` command to back up individual libraries, you should always back up to a tape or other media device and then store the media off-site.

If you typically back up the entire system, Warehouse Manager Server information is backed up when you perform the system save. For complete details about an entire system backup, refer to the IBM *OS/400 Backup and Recovery* documentation.

If you typically just back up individual libraries, simply include the Warehouse Manager library (for example, `SCSERVER`) in the list of libraries that you already back up.

To save your libraries

1. On an iSeries command line, enter `SAVLIB` and press F4.
2. Enter the names of the libraries that you usually back up, plus the Warehouse Manager library name.
3. In the Device prompt, enter the name of the device to save to (for example, `TAP01` if you are saving to a tape).

To restore the Warehouse Manager library

1. On an iSeries command line, enter `RSTLIB` and press F4.

2. At the Saved Library prompt, enter the name of the Warehouse Manager library to be restored (for example, the original SCSERVER).
3. At the Device prompt, enter the name of the device from which to restore this library.
4. Press F9 to display more prompts and then page down to the next screen.
5. In the Restore to Library prompt, enter the library name of your original Warehouse Manager Server (for exdample, SCSERVER) to restore teh Warehouse Manager Server library.

After restoring the original Warehouse Manager library, install a fresh copy in a new library (for example, SCNEWLIB). The installation procedure is described in the *STRATEGY Installation Guide.*

To recap, you have restored the SCSERVER library from a tape back to the SCSERVER library on the new system, and then reinstalled STRATEGY in a new SCNEWLIB library.

Next, we recommend saving the server information in the SCSERVER library and restoring it using the Save/Restore procedure. That is, run the SAVSRVRINF command from the SCNEWLIB library against the SCSERVER library and then run the RSTSRVRINF command from the SCNEWLIB library to restore the information.

See Chapter 4, “Warehouse Manager Server Maintenance” on page 49 for detailed steps of the Save/Restore procedure. An abbreviated version of this procedure, as it relates to disaster recovery backups, is described below:

1. Save the server information from the SCNEWLIB library. On an iSeries command line, type SCNEWLIB/SAVSRVRINF and press F4. This command will run from the new SCNEWLIB library you installed. Enter the following:
   - Enter your original Warehouse Manager Server library name (for example, SCSERVER).
   - Enter *SAVF to save the data to a file. A file is the most convenient choice, because you will keep the server information “live” instead of archiving it to tape.
   - Enter the save filename and the name of the library in which you want the file to be created. The file is created if it does not already exist.
2. Press Enter to save the information.
3. Restore the server information to the SCNEWLIB library. On an iSeries command line, enter SCNEWLIB/RSTSRVRINF and press F4. This command will run from the new SCNEWLIB library you installed. Enter the following:
   - Enter *SAVF to restore the data from a save file.
   - Enter the name of the save file from which to restore and the name of the library in which you created that file.
4. Press Enter to restore the information.

Backing Up Enterprise Reporting and Analyzer OLAP Data

The following command will save the server information including Analyzer OLAP functionality and any Enterprise Reporting information:

```
SAVSRVRINF SRVRLIB(serverlib) DEVICE(*SAVF) SAVF(savelib/savefilename)
   INFTYPE(*inftype) TGTRLS(*CURRENT) DLTVIEW(*NO)
```

where:

serverlib
is the library in which Warehouse Manager Server is installed.

savelib is the name of the library in which the backup file is stored.

savefilename
is the name of the backup file.

**inftype*
is the information type value.

To back up Enterprise Reporting data, use `*AOS` for the `*inftype` value. For general backup procedures, we recommend using `*ALL` as the `*inftype` value.

After Restoring Server Information

After running `RSTSRVRINF` to restore Warehouse Manager Server information, you may need to further configure your server so that it will run correctly. If you restored Warehouse Builder information, consider the following:

- If you restored the information to an iSeries different from the one on which it was saved, you may want to change your distribution sets and definitions. For example, if you were performing “pulls” of data from one iSeries to a test system, and you moved the distribution information to your production system, change the relational database (RDB) name of the target system so that the “pull” will now pull data to your production system. To make these changes, from the Tools menu in Warehouse Builder, choose **Definition Properties**.
- If you restored to a different iSeries, you should also ensure that TCP/IP communications are configured and active between the new system and the systems defined as source and target servers for the distributions.
- If you restored to a new iSeries or to a new library on the same iSeries, any schedule information defined for your distribution sets and any scheduled queries/reports are not restored. From the Sets menu in Warehouse Builder, choose **Work with Warehouse Builder Sets** to edit your sets and set up schedules.
- If you restored Warehouse Builder information into a server library already containing distribution definitions, the object IDs of the distribution sets will change. If you are running the sets from your own CL programs with the object IDs hard-coded in either the `STRDD` or `SBMDD` command, the hard-coded references may need to be changed. To avoid this problem in

the future, use the object name of the distribution set in the STRDD or SBMDD commands. See “Running Distribution Sets with STRDD and SBMDD” on page 42.

If you restored Warehouse Manager administration or Query information, consider the following:

- You can display the physical file RSTSRVPF to see the actions performed on the server catalog files.
- If you restored Warehouse Manager administration or Query information to a different iSeries, or if your environment has changed substantially since setting up this information, clean up the catalogs. From the Manage menu in the Warehouse Manager PC application, choose **Clean Up ShowCase Catalogs**.

Changing Physical Files That Have Dependent Data Views

Query lets you create and manage simplified views of any iSeries database. This simplified view, or data view, is saved on the iSeries as an SQL view. Since SQL views are saved on the iSeries and reference physical files (also known as tables), you may not be able to change physical files that have dependent views. Query includes several predefined queries and reports that you can run to find out which data views are built over a given iSeries table. You can run these predefined queries and reports from the Tools menu in Query.

If you find there are data views dependent on a physical file that you need to change, you can use the CHGPF command, which allows you to change a physical file without deleting dependent views. As a precaution and for recovery purposes, back up your views before using CHGPF.

To do this, use the Save/Restore procedure. For more information see “Saving and Restoring Data” on page 33. That is, you will want to run the SAVSRVRINF command on the iSeries to save data views. In this case, leave **Delete Views After Saving** set to ***NO** when running the SAVSRVRINF command.

Changing Data Views That Have Dependent Data Views

Just as data views can reference physical files, data views can also reference other data views. If you built a data view over another view, you cannot delete, modify, or replace the base view because it has a dependent view. For example, if VIEW2 is built over VIEW1, you cannot delete, modify, or replace VIEW1. You have three options, all of which can be done within Query:

- Delete the dependent view, VIEW2, if it’s no longer needed.
- Edit the dependent view to remove the dependency.

- Create a copy of the base view, VIEW1, and give the copied view a unique name (for example, VIEW1TEMP). Change the dependent view, VIEW2, to use the new view, VIEW1TEMP. This removes the dependency on VIEW1. You are now free to make your changes to VIEW1. After completing the changes to VIEW1, link VIEW2 back to VIEW1. You can then delete the temporary view VIEW1TEMP.

For complete details about data view tasks, see the Query online Help.

Managing Passwords

Scheduled reports and Warehouse Builder distributions rely on consistent passwords to connect to the iSeries. When a profile tries to connect to an iSeries through a STRATEGY application, the password is passed through to logon to the iSeries. In time, changed and/or expired passwords can cause the STRATEGY control tables to have incorrect passwords for a user. This may cause Warehouse Builder distributions and scheduled reporting to fail.

Changing Passwords with a STRATEGY Application

If the user's iSeries password has changed or expired, the user can update the applicable product 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. This ensures uninterrupted scheduled reports and Warehouse Builder distributions.

Updating Passwords with the SCCHKPWD Command

To update an iSeries user profile's password in the product control tables, use the SCCHKPWD command. For the command details, press F1 after entering SCCHKPWD on the command line or see Appendix A, "Commands" on page 85. SCCHKPWD allows you to:

- Specify whether to update the tables or to generate a list of the items matching the search criteria.
- Limit the generated list to matching entries for DB2 databases, Warehouse Manager Servers, and non-ShowCase ODBC data sources.

Examples

- Update passwords for an iSeries RDB (Warehouse Builder distributions and Enterprise Reporting ODBC data sources):

```
SCCHKPWD USERID(myuserprofile) OLDPWD(MYOLDPWD) UPDATE(*YES) RDB(iSeriesname)
AOS(*NONE) ODBC(*NONE) NEWPWD(MYNEWPWD) PWDCONFIRM(MYNEWPWD)
```

- Update passwords for a non-STRATEGY data source (ER scheduled reports and WB Oracle or SQL Server pulls):

```
SCCHKPWD USERID(myuserprofile) OLDPWD(MYOLDPWD) UPDATE(*YES) RDB(*NONE)
AOS(*NONE) ODBC(mysource) NEWPWD(MYNEWPWD) PWDCONFIRM(MYNEWPWD)
```

- Update passwords for the local system (RDB and AOS):

```
SCCHKPWD USERID(myuserprofile) OLDPWD(MYOLDPWD) UPDATE(*YES) RDB(*LOCAL)
AOS(*LOCAL) ODBC(*NONE) NEWPWD(MYNEWPWD) PWDCONFIRM(MYNEWPWD)
```

- Get a list of items affected by a user password change:

```
SCCHKPWD USERID(myuserprofile) OLDPWD(MYOLDPWD) UPDATE(*NO) RDB(*ALL) AOS(*ALL)
ODBC(*ALL)
```

- Get a summary of RDB, AOS, and ODBC names affected by a user password change:

```
SCCHKPWD USERID(myuserprofile) OLDPWD(MYOLDPWD) UPDATE(*NO) DETAILS(*NO)
RDB(*ALL) AOS(*ALL) ODBC(*ALL)
```

Using SCCHKPWD with Essbase Build Plans

Using the SCCHKPWD command, you can update SQL passwords (for DRDA connection) that are used with an Essbase build plan.

Note that the DRDA system name is not stored in the control table (that information is stored in the Essbase rules file and read by Essbase during actual data load or dimension build). To update the SQL password, use the associated Essbase Server to identify the correct record. Be aware that because the same Essbase Server can use different DRDA data sources for data load using the same user ID and password combination, you may update all of those passwords at once.

Example

One user has Essbase Server SCEssbase and uses:

- UserA and pwd123 for data load from DRDA data source SYS1
- UserA and pwd234 to load from SYS2
- UserA and pwd123 to load from SYS3

Running the following command will update user A’s password for *both* SYS1 and SYS3:

```
SCCHKPWD USRID(userA) OLDPWD(pwd123) UPDATE(*YES) DETAILS(*NO) RDB(*none) AOS(*none)
ODBC(*none) ESSBASE(SCEssbase) NEWPWD(newPwd) CONFNEWPWD(newPwd)
```

The result may be undesirable if the user only intended to update the SYS1 password.

There are two ways to avoid this problem:

- Change user A’s password on SYS3.
- Use different/unique user ID for each DRDA system connection.

Managing Information Workflow

Running Distribution Sets with STRDD and SBMDD

The STRDD (start data distribution) and SBMDD (submit data distribution) commands run Warehouse Builder distribution sets by set name (OBJNAME) or ID (OBJID). For more information on these commands, see Appendix A, "Commands" on page 85.

For example, the following commands run the distribution set Sales:

```
STRDD OBJNAME('Sales') OBJID(*NONE) or STRDD OBJNAME('Sales')
```

The following commands run the distribution set that has an ID 51:

```
STRDD OBJNAME(*OBJID) OBJID(51) or STRDD OBJID(51)
```

The SBMDD command works in the same way.

The STRDD and SBMDD commands can be inserted into a user program to run distribution set(s). These commands return an RSC6336 status message when a distribution set fails. The user program can monitor for the RSC6336 message, and the appropriate response can be coded into the user program.

For example, if one set in the program fails, the user program can stop instead of proceeding to the next set. This enhancement does not affect existing user programs that use STRDD or SBMDD. This is a status message and the program is not required to monitor for it.

Example

Program using STRDD:

```
PGM /* PGM1 */
ADDLIBLE LIB(SCSERVER)
STRDD OBJNAME('Sales')
MONMSG MSGID(RSC6336) EXEC(DO)
    SNDMSG 'Set 1 Failed, call application support - Sets 2
    and 3 not run'
    GOTO END
ENDDO
SBMJOB CALL(PGM2)
SBMJOB CALL(PGM3)
END:
ENDPGM
```

Pulling iSeries Data into Essbase NT

It is possible to submit your Essbase NT load plans from Warehouse Builder to Essbase on the NT (6.0 patch 1 or later) platform. Once created, you can use all the features of Essbase on the NT platform.

Requirements

Be sure to meet these requirements before you begin this procedure.

- The SCODBC ODBC driver must be installed on the Essbase NT Server. Use the STRATEGY installation CD to install one of the following PC Client components: Query, Warehouse Builder, Warehouse Manager, Analyzer, or Analyzer Designer.
- Set up an ODBC data source on the Essbase NT (6.0 patch 1 or later) Server. From the STRATEGY menu, select **Microsoft ODBC Administrator**.
- Set up Hyperion Essbase SQL Interface on Essbase NT (6.0 patch 1 or later). This is a separately licensed add-on offered by Hyperion. You can verify licensed products in the Essbase Application Manager by clicking **Server | Information | License Info**.
- In Warehouse Manager, you must enable STRATEGY licensing for this feature by adding a password for the Essbase NT application. This password is available by contacting a service representative. You can also add this password with the `CHGSCAUT` command on the iSeries. See "Add Password command" in the Warehouse Manager online Help.
- Start your Warehouse Manager Server and Essbase NT Servers.

Creating Application and Database, Database Outline, and Data Load Rule Files

Before you start, make sure your Warehouse Manager Server and Essbase NT Servers are running.

Note: For instructions on creating the new application and database, saving the database outline to Essbase NT Server, and saving the data load rule files to the Essbase NT Server, see the Hyperion Essbase documentation.

Verify and Edit the Data Load Rule Files on Essbase NT: After the data load rules are saved to Essbase NT Server:

1. Highlight the Application Desktop Window for Essbase NT Server. Select the application and database that you just created.
2. Click **Data Load Rules**.
3. Select a data load rule file and click **Open**. The data load rule file will appear.
4. From the File menu, choose **Open SQL**. The Selected Application, Server, and Database dialog box will appear. Click **OK**. The Define SQL dialog box will appear.
5. In the SQL Data Sources list box, verify that the selected data source points to the correct SCODBC data source.
6. In the Select text box, make any necessary changes to the SQL Select Statement, which determines which data you're extracting from the iSeries data warehouse.
7. In the From list box, verify that the library listed (the format is `library.table`) is the library on the iSeries containing the data files.

8. In the Define SQL dialog box, click **OK/Retrieve** to retrieve sample data. The iSeries data will appear.
9. Finish creating and editing the rule file. For more information, see the *Hyperion Essbase Database Administrator's Guide*.

- You may now use Application Manager to perform a dimension build/data load.
- In Warehouse Builder Client, you can reference this load rule in your Essbase build plan. When defining the Essbase build plan, type the Essbase NT IP address.

STRATEGY Integration with J. D. Edwards (JDE)

STRATEGY applications support versions A7.*x* and A8.*x* of JDE World and version B7.33.*x* of JDE OneWorld. The following sections describe enhancements to product integration with JDE for this release. For additional JDE information such as data dictionary and security information, go to http://support.spss.com and log on to the ShowCase division support site.

Non-iSeries Profile Sign-On

Non-iSeries profiles cannot be used with Warehouse Builder or Warehouse Manager Client, and they cannot be administrated with Warehouse Manager Client. Also, non-iSeries profiles are not supported through Enterprise Reporting (ER). This means you cannot sign on to ER via a browser with non-iSeries profiles, and you cannot save to, open from, or log on to the Server in Query or Report Writer with a non-iSeries profile.

Previously, JDE OneWorld users needed an OS/400 user profile for product applications to recognize their environment information. Administrators can now use the SETJDEOWA command to enable product applications to recognize environment information for OneWorld users who do not have an iSeries user profile.

- To enable SETJDEOWA, use the following syntax:

  ```
  SETJDEOWA SNGLUSRPRF(*ENABLE) JDEOWUSER(owprofile) JDEOWPWD(owpassword)
     JDEOWSYSLB(*SEARCHSYS)
  ```

  where *owprofile* is the user's JDE OneWorld profile and *owpassword* is the user's JDE OneWorld password.

  Note: The job description for the iSeries profile used in the SETJDEOWA command must have the JDEOW, QTEMP, and QGPL library lists. The profile must also have an initial program of BV3C in QGPL in order for the JDE OneWorld application user to be recognized as a OneWorld user. For more information about iSeries user profiles and job descriptions, refer to your iSeries documentation.
- To disable SETJDEOWA, use the following syntax:

```
SETJDEOWA SNGLUSRPRF(*DISABLE)
```

Editing ShowCase.cfg to Use an Alternate F0094 File

Usually, the F0094 file in JDE OneWorld contains the Master Library List data. JDE OneWorld integration with STRATEGY applications relies on the data in this file to set the library list upon connecting to the database through the Web interface or the application interface. On occasion, the JDE OneWorld environment administrator may choose to omit the F0094 file in its OneWorld client environment and use information contained in a client access data source. Since STRATEGY applications cannot determine the data source used for the library lists, a method has been provided so that users can manually specify an alternate file (which must contain F0094 data). This method involves duplicating the F0094 file and editing the configuration file (*ShowCase.cfg*).

To set up this alternate configuration:

1. Find the JDE system library name by using the `WRKOBJ` command:

   ```
   WRKOBJ OBJ(*ALL/F980WSEC) OBJTYPE(*FILE)
   ```

   The object and library name are displayed.
2. A duplicate of the F0094 file is required. The `CRTDUPOBJ` command may be used, with the *F0094SPSS* filename in *MYLIB* and the JDE system library name *SYSB733* as examples:

   ```
   CRTDUPOBJ OBJ(F0094) FROMLIB(SYSB733) OBJTYPE(*FILE) TOLIB(MYLIB) NEWOBJ(F0094SPSS)
      DATA(*YES)
   ```

   Note: The `DATA(*YES)` option duplicates the data. If you want to populate an empty file with the library list information, you can set the option to `DATA(*NO)`.
3. The main configuration file (*ShowCase.cfg*) in the Warehouse Manager Server library must contain two entries specifying the location of the alternate filename and library name. The iSeries `EDTF` command may be used to edit the file.

   Client access may be used for the same purpose. Following is the `EDTF` example, where *SCSERVER* is the Warehouse Manager Server library name:

   ```
   EDTF '/SCSERVER/proddata/config/ShowCase.cfg'
   ```

   Add the following two lines to the configuration file:

   ```
   showcasecorp.JDEOneWorldF0094File=F0094SPSS
   showcasecorp.JDEOneWorldF0094Library=MYLIB
   ```
4. To activate this change, stop and restart Warehouse Manager Server. For more information, see Chapter 1, "Starting and Stopping the Servers" on page 1.

User-Defined Codes (UDC)

During the initial setup of the JDE system, an administrator will customize descriptions for users based on category codes that exist with the JDE system. A user can select these UDC customized descriptions in the columns display. UDC customized descriptions are available in Query, Report Writer, and when running interactive reports via a browser using Enterprise Reporting.

Note: To work with Warehouse Builder definitions that contain JDE data and UDC columns, OS/400 V4R5 users must apply the following PTF or the definitions will not run: `5769SS1 V4R5M0 SF64377.`

Multiple Library List Environments

STRATEGY applications retrieve library list environments that have been set up within the JDE environment. A user who has been set up in the JDE system to have multiple library list environments will have the option to select an environment on the initial connection or have one set as the default. The prompting for library list environments can take place when running interactive reports via a browser using Enterprise Reporting. An enhancement has also been made allowing the prompting for library list environments if the environment is defined for the user group instead of the individual profile.

Selecting Group Library Lists

There is an order that JDE must follow to select group library lists. This order differs for OneWorld and World users.

OneWorld

The order of library list selection for OneWorld users is:

- User with a single library list (taken from the *F0092* file)
- Group with a single library list (taken from the *F0092* file)
- User with a multiple library list (taken from the *F0093* and *F0094* files)
- Group with a multiple library list (taken from the *F0093* and *F0094* files)

World

Currently, JDE World depends on the initial programs—J98INIT or J98INITA.

The order of library list selection for World users with J98INIT is:

- User with a single library list (taken from the *F0092* file)
- Group with a single library list (taken from the *F0092* file)

The order of library list selection for World users with J98INITA is:

- User with a multiple library list (taken from the *F0093* and *F0094* files)
- Group with a multiple library list (taken from the *F0093* and *F0094* files)
- User with a single library list (taken from the *F0092* file)
- Group with a single library list (taken from the *F0092* file)

Note: If the library list cannot be set due to incorrect data, missing libraries, or the lack of a library list, Query, Report Writer, and Enterprise Reporting will not function properly.

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 Essbase/400 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 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.

Chapter 4. Warehouse Manager Server Maintenance

This chapter discusses typical maintenance procedures that system administrators must perform on Warehouse Manager Server. For instructions on maintaining the server for Analyzer OLAP functions, see "Analyzer OLAP Maintenance" on page 69.

Using TCP/IP

The product software is used in a transmission control protocol/Internet protocol (TCP/IP) network environment. Client and server products communicate with each other using connections through a TCP port.

Note: Whether you installed Warehouse Manager Server with one port number or two port numbers, you can use the `CHGSCSRVA` command to configure the port(s).

To change the TCP port number or port numbers after the install, use the `CHGSCSVRA` command. This makes it easier to migrate from a test environment to a production environment. See "Migrating Users to a New Warehouse Manager Library" on page 50.

Configuring Warehouse Manager TCP/IP with STRSCSVR

If you configured Warehouse Manager TCP/IP during installation, you can skip this section. Before configuring TCP/IP, verify that OS/400 TCP/IP is configured and active. To enable Warehouse Manager TCP/IP support, run the `CHGSCSRVA` command.

The following additional commands are available in the server library to help users maintain Warehouse Manager Server TCP/IP support:

- `CHGSCSVRA` to change subsystem information and TCP/IP-specific objects.
- `STRSCSVR` to start the manager job.
- `ENDSCSVR` to end the manager and client jobs (to which the client is talking).

Finding Available Port Numbers

A TCP/IP port number is a numeric alias for an application. You assign a specific port to a Warehouse Manager Server installation library so the client software knows to which TCP/IP server application to connect. To enable the server for TCP/IP, you must assign a unique port number for each library.To view a list of the ports that are in use, run the OS/400 command `WRKSRVTBLE`.

The command CHGSCSVRA assigns the port number 43419 by default. If this number already appears when you run the command WRKSRVTBLE, choose another port number. The next available number should work.

Migrating Users to a New Warehouse Manager Library

At times, it may be necessary to have your users change libraries. For example, you may install a new release of the Warehouse Manager Server and first verify that it works in your test environment. You then want users to stop using the old release and begin using the new one.

In a TCP/IP network, users' client configuration employs the TCP/IP port to determine which Warehouse Manager installation its connects to. This allows you to stop the old server, configure the new library with the port number from the old library, and start the new server. This way, users can automatically connect to the new library without any changes to their client configuration.

Checking Usage of a Warehouse Manager Library

Listing Connections to Relational Applications Only

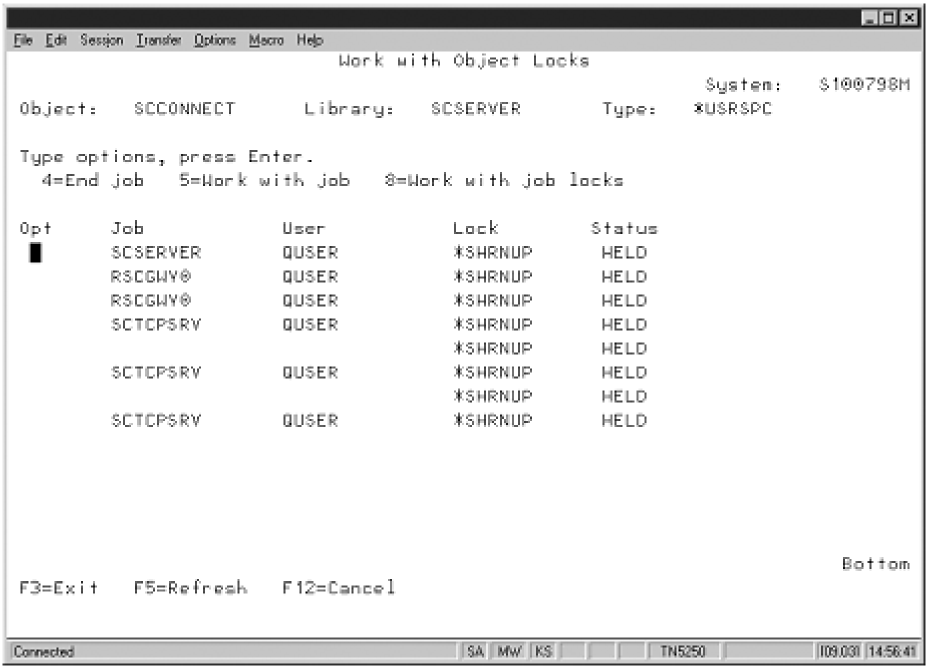
The following procedures list any TCP/IP active jobs to relational products such as Warehouse Builder. To list all the active connections to applications including Analyzer OLAP, see "Listing All Connections to All Applications" on page 51.

For Warehouse Manager libraries, run the following command:

```
WRKOBJLCK OBJ(serverlib/SCCONNECT) OBJTYPE(*USRSPC)
```

where *serverlib* is the name of the library that you want to check.

A screen similar to the following will appear. If TCP/IP is active, the screen shows the TCP/IP Manager job for the library (job name SCSERVER, which is the same as the library name) and may show TCP/IP connections (job name SCTCPSRV).



```
                              Work with Object Locks
                                                             System:   S100798M
Object:   SCCONNECT      Library:   SCSERVER       Type:   *USRSPC

Type options, press Enter.
  4=End job   5=Work with job   8=Work with job locks

Opt     Job             User            Lock         Status
        SCSERVER        QUSER           *SHRNUP       HELD
        RSCGWY0         QUSER           *SHRNUP       HELD
        RSCGWY0         QUSER           *SHRNUP       HELD
        SCTCPSRV        QUSER           *SHRNUP       HELD
                                        *SHRNUP       HELD
        SCTCPSRV        QUSER           *SHRNUP       HELD
                                        *SHRNUP       HELD
        SCTCPSRV        QUSER           *SHRNUP       HELD





                                                                         Bottom
F3=Exit   F5=Refresh   F12=Cancel
```

Figure 2. Using the WRKOBJLCK command to check library usage

To see which users are connected (for example, to ask them to disconnect before an installation), use **Work With Job Locks** (option 8). You should see a lock on the user profile (enter `*USRPRF`). (If you see `QUSER`, this indicates no user is connected). From the user profile name, you should be able to determine who is using the product. Alternatively, you can use the Warehouse Manager Client software to view active connections.

Listing All Connections to All Applications

This method lists active connections to all applications, including those using Analyzer OLAP functionality.

1. Open the Warehouse Manager Client application and connect to the data source.
2. Double-click the data source for which you want to see connections.
3. From the Warehouse Manager menu, choose **Applications**.
4. From the Manage menu, choose **Active Connections**.

Ending TCP/IP (Stopping the Server)

Before ending TCP/IP, verify that no one is using the library (Refer to "Checking Usage of a Warehouse Manager Library" on page 50). If you see the library name in the list of jobs with locks, or jobs named `SCTCPSRV` are listed, TCP/IP is enabled.

From the library in which you want to end TCP/IP, run the command ENDSCSVR. This command will end TCP/IP clients and stop the TCP/IP manager job for the library.

Uninstalling a Warehouse Manager Library

You may occasionally need to uninstall Warehouse Manager libraries on your iSeries system, such as test libraries.

Uninstalling Warehouse Manager Server Libraries

1. Log on as QSECOFR or with equivalent authority (*ALLOBJ, *IOSYSCFG, *SAVSYS, *JOBCTL, and *SECADM special authorities are needed).
2. Back up any Warehouse Manager settings that you want to save. See "Backing Up Warehouse Manager Server Information" on page 53 for instructions.
3. Verify that no users are connected to the library. See "Checking Usage of a Warehouse Manager Library" on page 50.
4. Restore the UNINSTSC utility from the UNINSTSC save file that exists in the Warehouse Manager Server library and enter the following command:

   ```
   RSTOBJ OBJ(*ALL) SAVLIB(QTEMP) DEV(*SAVF) SAVF(serverlib/UNINSTSC) MBROPT(*ALL)
   ALWOBJDIF(*ALL) RSTLIB(QSYS)
   ```

   where *serverlib* is the name of the Warehouse Manager library to be uninstalled.

   Note: This will restore three objects into QSYS.
5. Interactively enter the Warehouse Manager command UNINSTSC and then press Enter. This command cannot be submitted to batch.
6. Enter the name of the Warehouse Manager library that you want to uninstall and then press Enter to confirm.

Uninstalling Warehouse Manager Server Libraries (pre-product)

1. Log on as QSECOFR or with equivalent authority (*ALLOBJ, *IOSYSCFG, *SAVSYS, *JOBCTL, and *SECADM special authorities are needed).
2. Back up any Warehouse Manager settings that you want to save. See "Backing Up Warehouse Manager Server Information" below for instructions.
3. Restore the UNINSTSC utility from the UNINSTSC save file that exists in the Warehouse Manager Server library. This must be restored from a 3.0 or later release.

   ```
   RSTOBJ OBJ(*ALL) SAVLIB(QTEMP) DEV(*SAVF) SAVF(serverlib/UNINSTSC) MBROPT(*ALL)
   ALWOBJDIF(*ALL) RSTLIB(QSYS)
   ```

   where *serverlib* is the name of the Warehouse Manager library to be uninstalled.

4. Interactively enter the Warehouse Manager command `UNINSTSC`. It cannot be submitted to batch.
5. Enter the name of the Warehouse Manager Server library that you want to uninstall.
6. To uninstall the library, press Enter.

Backing Up Warehouse Manager Server Information

Because of the way Warehouse Manager is installed, we recommend backing up your installation library with `SAVSRVRINF`, rather than using conventional means. Following is the series of procedures that you should use to back up a Warehouse Manager Server library.

Saving iSeries Media and Passwords

Be sure to save the iSeries media and passwords that you received. You will need them to restore the server library. These passwords are valid when restoring to the same iSeries.

Saving Server Information

Note: If you are saving information and the device is tape, objects must be the first or the only objects on the tape. Otherwise, the save to a tape will fail.

This task is required only if you are using any of the following applications:

- Warehouse Manager (server options such as iSeries library access and default public authority to iSeries libraries and tables; security and resource settings, aliases, and column attributes; and application security).
- Query (data views).
- Enterprise Reporting.
- Warehouse Builder (such as data distribution sets and definitions).
- Analyzer applications (such as forms, models, pinboards, and views).

The Warehouse Manager, Query, Warehouse Builder, and Report Writer applications store information on the Warehouse Manager Server on the iSeries. Analyzer applications store information such as views and pinboards. Save and back up the information on the server with the `SAVSRVRINF` command, located in the server library, and then restore using the `RSTSRVRINF` command.

The Save and Restore feature, which consists of these two commands, is also useful for moving data from one server library to another. See “Moving Data from a Test to a Production Environment” on page 33 and “Backing Up Data” on page 35 for recommendations on moving data from a test to a production environment and backing up data.

To back up information that applications store on the Warehouse Manager Server:

1. Before saving server information, stop the Warehouse Manager Server and the Orion Server with the ENDSCSVR and ENDORI commands. For more information, see Chapter 1, "Starting and Stopping the Servers" on page 1.
2. Log on with a user ID that has *ALLOBJ authority.
3. Add the Warehouse Manager source library to your iSeries library list and ensure that you have only one Warehouse Manager library in your library list. Run the command:

   ```
   ADDLIBLE serverlib
   ```

   where *serverlib* is the name of the Warehouse Manager library with which you want to work.
4. Enter the command SAVSRVRINF and press F4.
5. At the Library prompt, enter the name of the server library from which server information should be saved. This library is typically SCSERVER.
6. At the Device prompt, enter the name of the device to save to, or type *SAVF to save to a file. Press Enter. If you entered *SAVF, enter the save filename and the iSeries library in which you want to create the save file.
7. At the Type of Information to Save prompt, enter one of the following:

   *ALL to save all of the information described in the following bullets.

   *ADMIN to save only Warehouse Manager administration information (such as security settings and aliases).

   *AOS to save only Analyzer OLAP information (such as forms, models, pinboards, reports, and views) and Information Delivery applications' user information (such as shared queries).

   *DIST to save only Warehouse Builder information (such as data distribution sets and definitions).

   *DATAVIEW to save only Query data views.

   Note: Even if you save all information, you can restore individual groups of settings.
8. Press F10 to see additional prompts for the SAVSRVRINF command. These prompts do not need to be changed under most circumstances. However, the prompts are described below, in case they might be useful to you:
   - At the Target Release prompt, enter *PRV if you intend to restore saved information on the previous release of the OS/400, or leave the default of *CURRENT if you intend to restore saved information on the same or a later release of the OS/400.

- At the Delete Views After Saving prompt, specify whether or not to delete data views after saving. It is recommended that you leave this option set to*NO. However, if you need to change physical files that have dependent views, enter *YES to delete the data views on your system. (The views can later be restored with the RSTSRVRINF command.) Another option is to use the CHGPF command, which allows you to change a physical file without deleting dependent views.

9. Press Enter to save the server information.

Restoring Warehouse Manager Server Information

Installing iSeries Media and Passwords

1. Follow the procedures in the “Installing Server Software” chapter of the *STRATEGY Installation Guide*.
2. Enter your Warehouse Manager passwords. For more information, see the *STRATEGY Installation Guide*.

Restoring Server Information

To restore the server information that was saved with the SAVSRVRINF command:

1. Before saving server information, stop the Warehouse Manager Server and the Orion Server with the ENDSCSVR and ENDORI commands. For more information, see Chapter 1, “Starting and Stopping the Servers” on page 1.
2. Distribute your save file or move your tape or other device to the target system, if it’s different from the first system.
3. Add the Warehouse Manager source library to your iSeries library list and ensure you only have one Warehouse Manager library in your library list. Run the command:

   ```
   ADDLIBLE serverlib
   ```

   where *serverlib* is the name of the Warehouse Manager library with which you want to work.
4. Enter the command RSTSRVRINF and press F4.
5. At the Device prompt, enter the name of the device from which to restore or enter *SAVF to restore from a save file. Press Enter. If you entered *SAVF, enter the save filename and the iSeries library in which you created the save file.
6. Choose whether or not to restore Warehouse Manager Server administration information. Press Enter. If *YES, the Run Catalog Check Program prompt displays. Choose one of the following values for this prompt:

*NO to not run the catalog check program on the restored information.

*AUTODELETE to delete any records that do not apply to objects found on the system.

*LISTONLY to simply create a list of records that do not apply to objects found on the system. You can later delete these records by running the CHKSRVRCAT command and specifying *YES on the DLTORPHAN parameter.

Note: To view the results of a catalog cleanup operation, use the Warehouse Manager Client application. On the Manage menu, choose the **Clean Up ShowCase Catalogs** command and then click **View Results** to display the results.

7. Choose whether or not to restore Analyzer OLAP information. Press Enter. If *YES, the Duplicate User Option and Restore Configuration File prompts display. Choose one of the following values for the Duplicate User Option prompt:

   *KEEP to keep the existing AOS data in the target user-permission file. That is, existing system manager privileges and user settings will not be replaced with any duplicate AOS data being restored.

   *REPLACE to replace the existing AOS data in the target user-permission file with any duplicate AOS data being restored.

   Choose one of the following values for the Restore Configuration File prompt:

   *YES to replace the existing AOS configuration file with the saved (SAVSRVRINF) version of this file.

   *NO to not restore this file and thus not overwrite the existing AOS configuration file.

8. If you specified *YES for either the Restore Warehouse Manager Information or Restore AOS Information prompt, the Duplicate Record Option prompt is displayed. This prompt specifies the action that restore takes if it finds any saved Warehouse Manager or AOS data that duplicates data in the target server library. Choose *KEEP or *REPLACE.
9. Choose whether or not to restore Warehouse Builder information. Press Enter. If *YES, the Restore Logs prompt is displayed.
10. Choose whether or not to restore the logging and message information from previous runs of the data distribution sets and definitions that you saved. Choosing to restore Warehouse Builder information also displays a

duplicate record option. Choose whether to keep or replace duplicate records by entering *KEEP or *REPLACE.

11. Choose whether or not to restore Data View information.
12. You can press F10 to see additional prompts for the RSTSRVRINF command. The following prompt does not need to be changed under most circumstances. However, it is described below in case it might be useful to you.
    - In the Catalog File to be Processed prompt, the default *ALL restores all catalog files that contain application administration information. These files contain application resource settings, Warehouse Manager security settings, aliases, and column attributes. Alternately, you can specify a single catalog file to restore. Press F1 on this prompt to display the Help, which lists the catalog files and their contents.
13. Press Enter to restore the server information. Several status messages will flash and then a screen with additional configuration steps will appear. Read this information carefully, especially if you restored Warehouse Builder, Warehouse Manager, Query, or Data View information.

    For example, if you are restoring Warehouse Builder information to a new server library on the same system or to a different system, we recommend you use Warehouse Builder to make any necessary changes, to create target tables, and to reschedule jobs.

Detecting and Removing Obsolete Information

When using Warehouse Manager's Alias Manager and Security Manager, you define alias names or security rules. Alias names reference existing objects on your system. If those objects change, the alias names may reference objects or fields that no longer exist. Similarly, Security Manager defines access rules for a particular object and user. In this case, not only can the referenced objects change but the users may no longer exist.

When you use Query to create data views, Warehouse Manager stores information about these data views. It is possible that these data views could be renamed or deleted by programs other than Query. The Warehouse Manager information could then reference objects that no longer exist.

Given these situations, Warehouse Manager provides a way to resynchronize the catalog information with your changed system. On the Manage menu, click **Clean Up ShowCase Catalogs**. The command allows you to determine and view information that is no longer valid, and remove it if you choose. For additional information on the command, see the Warehouse Manager Client online Help.

Restricting Access to a Warehouse Manager Library

It may be necessary to put a Warehouse Manager Server library in a restricted state. This restricted state prevents users from connecting to the server library. Once a server library is in a restricted state, you can easily remove the restricted state so users can reconnect.

Applying the Restricted State

To put a server library in a restricted state:

1. Log on as QSECOFR or equivalent authority (*ALLOBJ, *IOSYSCFG, *SAVSYS, *JOBCTL, and *SECADM special authorities are needed).
2. Add the Warehouse Manager library to your iSeries library list and ensure that you have only one Warehouse Manager library in your library list. Run the command:

   ```
   ADDLIBLE serverlib
   ```

   where *serverlib* is the name of the Warehouse Manager library with which you want to work.
3. Change the server status by entering the command CHGSRVRSTS and pressing **F4**.
4. At the Library prompt, enter the name of the Warehouse Manager library that you want to put in a restricted state. This library is typically SCSERVER.
5. At the Restrict Server Library prompt, enter *YES to prevent users from connecting to the server library. Only your current job (running the CHGSRVRSTS command) and the TCP/IP master job (if it's running) remain connected to the restricted server library.

   Note: If users are already connected to the server library, it will not be put in a restricted state unless you force the restriction.
6. Press Enter.
7. At the Force Server Restricted State prompt, choose whether or not to force the server library into a restricted state:

   *NO to attempt to put the server library in a restricted state. However, if users are connected to the server library, the attempt will fail.

   *YES to force all connections to the server library to end (except for those listed in the fifth step—your current job and TCP/IP master job). This will end the jobs of any users who are currently connected to the server library.
8. Press Enter. If you specified *NO in the previous step, the server library may or may not be put in a restricted state. If you specified *YES, the Maximum Force Wait Time prompt displays. This prompt allows you to set the maximum time that the CHGSRVRSTS command should run while

waiting for all connections to the server library to end. The default time interval is 30 seconds. After you specify a time limit, press Enter.

The countdown begins *after* all connections to the server library have started to close. The server is put in a restricted state once either of the following occurs:

- No more connections are found during the specified time interval.

or

- The time interval expires and there are no more connections.

If the time expires and there are still connections to the server library, the server will not be put in a restricted state. In this case, you will need to retry until the restricted state is achieved so that you can change the Enforce iSeries Security option.

Note: Regardless of whether or not the server library is put in a restricted state, when the `CHGSRVRSTS` command is issued with the `Force=*YES` option, all active connections will begin to close. This process cannot be reversed once it has started. If the server library fails to enter a restricted state in the specified time interval, users whose connections were ended in the process are allowed to reconnect, as are any other users.

Removing the Restricted State

To release a server library from a restricted state:

1. Log on as `QSECOFR` or equivalent authority (`*ALLOBJ`, `*IOSYSCFG`, `*SAVSYS`, `*JOBCTL`, and `*SECADM` special authorities are needed).
2. Add the Warehouse Manager library to your iSeries library list and ensure you only have one Warehouse Manager library in your library list. Run the command:

   ```
   ADDLIBLE serverlib
   ```

   where *serverlib* is the name of the Warehouse Manager library with which you want to work.
3. Change the server status by entering the command `CHGSRVRSTS` and pressing F4.
4. At the Library prompt, enter the name of the server library that you want to release from a restricted state.
5. At the Restrict Server Library prompt, enter `*NO` to allow users to access the server library again.
6. Press Enter to remove the restricted state.

Working with Journals

Warehouse Manager creates two journals when it is installed. These journals are used to maintain data integrity when catalog files are updated. The two journals are RSCSPCTLJ, which is associated with stored procedure catalogs, and RSCSCCTLJ, which is associated with Warehouse Manager, Warehouse Builder, and Query Client catalogs.

The main concern with journaled files is the space consumed by the journal receivers. To ensure the journal receivers do not consume too much space on your system, you can use one of the following approaches:

- Monitor the size of the receivers and issue a CHGJRN command when they get too large. Once the CHGJRN command is issued and new journal receivers are attached to the journal, you can (optionally) save the receivers and then delete them.
- Let the system maintain the journal receivers. Enter the commands in the following steps to do so:

```
CRTJRNRCV JRNRCV(serverlib/receiver-name) THRESHOLD(500000)

CHGJRN JRN(serverlib/RSCSPCTLJ) JRNRCV(serverlib/receiver-name) SEQOPT(*RESET)
MNGRCV(*SYSTEM) DLTRCV(*YES)

CRTJRNRCV JRNRCV(serverlib/receiver-name) THRESHOLD(500000)

CHGJRN JRN(serverlib/RSCSCCTLJ) JRNRCV(serverlib/receiver-name) SEQOPT(*RESET)
MNGRCV(*SYSTEM) DLTRCV(*YES)
```

Warehouse Manager and iSeries Work Management

This section describes how Warehouse Manager is influenced by iSeries work management controls and provides information for adjusting these controls. Before reading this section, you should be familiar with iSeries work management concepts. For information on iSeries work management see the *IBM Work Management* document.

Work Management Controls

Following is an overview of what you can control with each of the work management constructs as they relate to Warehouse Manager and running queries. Included is information for changing the constructs to influence Warehouse Manager behavior.

In general, you should not need to adjust the work management behavior of Warehouse Manager. This information gives you a basic understanding of how Warehouse Manager is interacting with your system and provides the information necessary for advanced systems management, such as system tuning or configuring job classes to control temporary storage limits.

Job Class

The job class used by Warehouse Manager can be used to control the following attributes with minimum impact to other jobs. Other attributes can also be controlled through the class description, such as time slice.

- Job run priority
- Maximum temporary storage
- Default wait time

Job run priority can be used to control Warehouse Manager’s run priority relative to other jobs on the system. If this is the only attribute you want to control, using the Resource Settings feature of Warehouse Manager Client is an easier alternative.

Maximum temporary storage can be used to control the amount of storage a job uses at run time. Temporary storage is the storage required for Warehouse Manager and system programs at run time, internal system objects used by the system for this job, and temporary objects used when running a query.

Default wait time can be used to control the maximum time, in seconds, that the Warehouse Manager job will allow for an instruction to finish running. Typically, this is the time Warehouse Manager waits to obtain either an object or record lock while processing a query.

Use the default wait time if locks are common in your environment and are affecting your query. Reducing the wait time will allow Warehouse Manager to return record and object lock errors more quickly. Lengthening the wait time will allow Warehouse Manager additional time for the object to become available.

In most environments a wait time of 30 seconds to 2 minutes is recommended. A wait time of `*NOMAX` is allowed but is not recommended as lock conditions may cause the PC to appear stopped for long periods of time.

To create a job class, use the `CRTCLS` command. Add or update the routing entries to use the new job class.

Storage Pools

The **storage pool** on the iSeries is a portion of the main memory isolated for a set of jobs. All jobs using a storage pool compete for the resource but do not compete against jobs using a different storage pool. Storage pools can be used to:

- Isolate memory-intensive jobs from other jobs on the system
- Give particular jobs more memory by reducing contention
- Keep highly used objects in main memory

Use caution when creating storage pools since the memory is subdivided and is not used unless a job is actively using the storage pool.

More than one storage pool can be associated with a subsystem. The routing entry indicates the storage pool that should be used by the job when running in the subsystem.

Subsystems

The subsystem Warehouse Manager runs in can be used to control the storage pools available to the server and to configure job classes within the subsystem.

To ensure the storage pool you are using has expert cache enabled, see "Enabling Expert Cache" on page 17.

The subsystem the server job uses is determined as follows.

TCP/IP
TCP/IP uses the subsystem specified during the Warehouse Manager installation or with the `CHGSCSVRA` command. By default, this is either `QCMN` or `QBASE` on most systems. If you need different PC workstations in different subsystems using TCP/IP, you must have multiple Warehouse Manager libraries installed.

Batch queries
Batch queries use the subsystem servicing the job queue. This job queue was specified in the job description and is used when the query is scheduled by the user.

Routing Entries

Routing entries allow you to choose the job class used by Warehouse Manager and the storage pool within the subsystem the job uses.

TCP/IP jobs: TCP/IP jobs use the job class associated with the routing entry added when the product was installed or when the CHGSCSVRA command was issued. By default, the product uses the QWCPCSUP job class for TCP/IP connections.

To change the job class used by TCP/IP, use the `CHGRTGE` command. The routing data format of the TCP/IP routing data is listed as follows:

Table 2. TCP/IP routing data format

Data	Position	Length
Library	1	10
`SCTCPSRV`	10	8

The iSeries also manages TCP/IP. These jobs simply start the associated client jobs when a connection is made to the iSeries using ODBC. The routing data is as follows:

Table 3. iSeries TCP manager job routing data

Data	Position	Length
Library	1	10
`SCTCPMGR`	10	8

Batch queries: For batch queries, you can add your own routing data in the job description. When that job description is a specified batch job schedule, the routing data is passed to the system. You can set up routing entries based on the routing data you define to assign the appropriate job class.

Job Descriptions

The job description allows you to control the initial library list used by the server, job log logging options, and job accounting.

TCP/IP jobs
For TCP/IP, the job description of `QUSER` is used initially. The library list of the job description specified in the actual user's user profile then replaces the library list of `QUSER`. The job log logging and job accounting are not updated.

Batch queries
For batch queries, it's the job description specified when scheduling the batch job.

User Profiles

The user profile allows you to control:

- Object level security and management attributes applied when running the server.
- Job description for the user.
- Permanent storage limits.
- National Language Support attributes of the job.
- Output queue used when producing a job log or other spooled output.

Warehouse Manager will always run using the user profile specified at ODBC logon when connecting to the iSeries.

When using WRKACTJOB, WRKUSRJOB, or a similar command, the job name may contain the actual user profile or it may contain QUSER, depending on how the job was started. QUSER is the user profile many jobs run under. QUSER must not have a storage limit. To make sure that QUSER has no limit, run the following command:

```
CHGUSRPRF USRPRF(QUSER) MAXSTG(*NOMAX)
```

To identify a job by user, you can use one of the following techniques. To list all jobs by user, use the command:

```
WRKOBJLCK OBJ(myuser) OBJTYPE(*USRPRF)
```

where *myuser* is the user profile for which you want to list jobs.

To list all jobs for a particular library, use the command:

```
WRKOBJLCK OBJ(libraryname/SCCONNECT) OBJTYPE(*USRSPC)
```

Security Issues

Command Authorities

This section lists the provided commands and their authorities as set at installation.

On installation, existing command authority is preserved for all users except *PUBLIC. *PUBLIC is unconditionally set to *EXCLUDE or *USE as described in the following tables.

Existing Data Administrators will not have authority to access the commands. You may:

- Grant authority to those users by employing the iSeries security management commands.
- Use Warehouse Manager Client to grant the Data Administrator privilege.

The Warehouse Manager-provided iSeries commands in the following table are shipped with *PUBLIC authority set to *EXCLUDE. A security officer can grant appropriate authority using the standard OS/400 security management commands.

*Table 4. Commands shipped as *PUBLIC = *EXCLUDE*

Command	Description
CHGSCSVRA	Configure TCP/IP connectivity support, subsystems, and various Analyzer parameters
APYSCPTF	Apply STRATEGY PTF
CFGDD	Configure data distribution
CHGAOSA	Use CHGSCSVRA command
CHGSCAUT	Change STRATEGY product authority
CHGSP	Change stored procedure
CHGSRVRDFT	Change server default attributes
CHGSRVRSTS	Change server status
CHKSP	Check stored procedure
CHKSRVRCAT	Check server catalog information
CPYSP	Copy stored procedure
CMPSCJAR	Compile the STRATEGY Java files
CRTSP	Create stored procedure
DLTSP	Delete stored procedure
DSPSP	Display stored procedure
ENDSCSVR	End TCP/IP connectivity support
RMVSCAUT	Remove STRATEGY authority
RSTSMPLB	Restore STRATEGY sample database
RSTSP	Restore stored procedure
RSTSRVRINF	Restore server information
SAVSP	Save stored procedure
SAVSRVRINF	Save server information
SETLIBACC	Set library access
STRSCSVR	Start TCP/IP connectivity support
UNINSTSC	Uninstall server

The commands in Table 5 are shipped with *PUBLIC authority set to *USE.

*Table 5. Commands shipped as *PUBLIC = *USE*

Command	Description
DSPAOSINF	Display AOS information
DSPSCINF	Display STRATEGY tips
DSPSCAUT	Display product authority
SBMDD	Submit Warehouse Builder distribution set
STRDD	Start Warehouse Builder distribution set
STRRPTEXEC	Start job executor

When granting a user Data Administrator privilege with Warehouse Manager Client, that user must be given iSeries *CHANGE authority to access the Warehouse Manager commands in the following table. If the user is not given *CHANGE authority, some administration functions will not be allowed.

*Table 6. Data administrator privilege = *CHANGE commands*

Command	Description
CHGSCAUT	Change product authority
CHGSRVRDFT	Change server default attributes
CHGSRVRSTS	Change server status
CHKSRVRCAT	Check server catalog information
DSPSCAUT	Display product authority
RMVSCAUT	Remove product authority
SETLIBACC	Set library access

Table 7 shows the product commands, but they should not be used. They are provided for server use and are listed here as reference information only. Using these commands may cause unpredictable results.

Table 7. Procust commands

Command
AOSMAST
CACHEMGR
RSCGWY

Table 7. Procust commands (continued)

Command
SCJOBMGR
SCLLQRY
SPBCHPRC
TCPMAST

User Exit Programs

A user exit program can be specified on the iSeries to accept or reject a request before the function is actually carried out. This allows for additional user-defined security not provided by the iSeries or Warehouse Manager. Specify the user exit program name on the client support request access (PCSACC) parameter of the iSeries change network attributes (CHGNETA) command. See the example below.

The PCSACC exit program on the CHGNETA command is supported. *REGFAC is also supported, but if the PCSACC value is *REGFAC the user will need to run the ADDEXITPGM command—with SC_QUERY_ROW_SEC for the Exit point parameter and SCRS0100 for the Exit point format parameter—to use User Exit Programs.

Example:

```
CHGNETA PCSACC(mylib/myuepgm)
```

This tells Client support to call this program before it runs a file transfer, virtual printer, or message function. Warehouse Manager uses the same network attribute and program parameters as Client Access/400 before it prepares each SQL statement or before submitting a batch query.

Following are descriptions of the two parameters that are passed to the user exit program when using Warehouse Manager or Client Access/400. More detailed documentation on user exit programs can be found in the IBM *OS/400 Server Concepts and Administration* document.

Parameter 1
: One byte exit program return code. The request is allowed if the user exit program sets this value to hex F1 (the character "1"). The request is rejected if this parameter is set to any other value.

Parameter 2
: This parameter contains information about the request. The following table describes the structure used by the product and is identical to the Client Access/400 File Transfer structure.

Table 8. Structure of the second parameter passed to a user exit program

Value	Type	Description
User profile	Character 10	Contains the user profile making the request.
Application name	Character 10	Contains the name of the PC application making the request.[2]
Requested function	Character 10	Contains the type of SQL statement, SELECT, JOIN, INSERT, UPDATE, DELETE, CREATE, DROP, GRANT, etc.[3]
Object name	Character 10	Contains the object (table) name.[4,5]
Library name	Character 10	Contains the library name or *LIBL.[4,5]
Member name	Character 10	Contains the member name, *FIRST or *LAST.[4,5] *ALL is not supported by iSeries SQL.
Record format name	Character 10	Contains the record format name. The product passes a value of *ONLY for the record format name.[5]
Length of next field	Zoned 5,0	Length of the statement in the following request.
Entire request	char(32500)	SQL statement

Notes:

1. When developing a user exit program, be aware that Client Access/400 Virtual Print may continue using the first user exit program it calls even though you use CHGNETA to change the user exit program.
2. An asterisk for the first character of the name signifies a Client application. Otherwise, the name is the ODBC Client application name specified by the PC on the ODBC connection string.
3. For SELECT statements involving multiple files, the requested function value is passed as JOIN and the user exit program is called once for each table involved in the join select.
4. The library name does not appear in the SQL statement when *LIBL is implied. The member name never appears in the SQL statement (members are handled through file overrides).
5. Blanks are passed for this field for any SQL statement other than SELECT, INSERT, UPDATE or DELETE.

Distributed Relational Database Architecture (DRDA)

DRDA is the IBM DB2 implementation for database operations on remote DB2s. The correct operation of Warehouse Builder requires that connections to remote iSeries databases be set up correctly. If there are errors with the configuration or if the IBM DDM server is not running, Warehouse Builder sets will not extract and/or write data from one iSeries database to another.

To make sure that remote databases are configured properly:

1. Enter the command `WRKRDBDIRE` and press Enter.

   There should be a list of remote database names corresponding to other iSeries systems. If the required systems are present, the following steps may be skipped.
2. The first step in adding a remote database entry is to decide which communications protocol to use. You have the choice of SNA or TCP/IP.
3. When configuring the entry for SNA, the APPC information for the remote system is required. Consult the IBM APPC manual to determine and configure the name, the device, the local location, the remote network identifier, and the mode. When configuring the entry for TCP/IP, the TCP/IP address is required. Determine the TCP/IP address of the remote iSeries.
4. Use the command `ADDRDBDIRE` and fill in the required parameters depending on the communications protocol chosen. Usually, the relational database being added is the system name of the remote iSeries.

If the communications protocol for the remote database entry is TCP/IP, the DRDA server must be started on the remote iSeries system. Use the command `STRTCPSVR SERVER(*DDM)` on the remote iSeries to ensure that the server is running.

If the communications protocol for the remote database entry is SNA, ensure that the device being used is in the varied-on state and is active. Consult the IBM APPC documentation for instructions, if necessary.

Analyzer OLAP Maintenance

This section discusses typical setup and maintenance procedures that system administrators must perform on the Analyzer OLAP functions of the Warehouse Manager Server. It also includes information about customizing the Analyzer for the Web applets, setting up the iSeries HTTP Server, and other miscellaneous notes.

Configuring the iSeries HTTP Server

This section gives an overview of configuring the iSeries HTTP Server for use with Analyzer for the Web. It is assumed you have a basic understanding of the iSeries HTTP Server. If you need more detailed information regarding the server or the configuration of the server, see *IBM TCP/IP Configuration and Reference Version 3*.

Configuring a New HTTP Server for Analyzer for the Web

If the iSeries HTTP Server on the iSeries has never been configured, at a minimum you must use the following procedure to add basic configuration settings. IBM ships a default configuration file with the iSeries HTTP Server.

However, all entries in the file are commented out using the "#" symbol. Figure 3 on 71 shows a sample HTTP configuration file with the required settings in bold text.

To configure a new HTTP Server:

1. At an iSeries command line, enter `CFGTCPHTTP` and press Enter.
2. Choose option 2, **Work with HTTP Configuration**.
3. Specify the configuration name. `CONFIG` is the default name.
4. Press Enter.
5. Choose **Add** (option 1) and enter the following above any map entries:

   ```
   Pass /serverlib/* /serverlib/ProdData/Analyzer/*
   ```

   where *serverlib* is the name of the library in which Warehouse Manager is installed.

   This entry specifies that the HTTP Server is allowed to serve up documents residing in the `serverlib/ProdData/Analyzer` directory. With a Web browser, the end user would need to specify only the `serverlib` as a path to the requested document.

   For example, if the Warehouse Manager Server has been installed in a library named `SCSERVER`, enter `/SCSERVER/index.html` to bring up the first Analyzer for the Web page. The HTTP Server would actually find the `index.html` page in your system IP or in `alias/SCSERVER/ProdData/Analyzer`.

   Note: The URLs may be case sensitive, depending on the configuration. If you want users to be able to enter the URL in lowercase, repeat step 5 and add another entry to the configuration file. The second entry would be the same as the first entry, except that the `serverlib` name would be in lowercase.
6. Press F3 twice to exit the HTTP configuration.
7. Recycle the HTTP Server.
   a. To end the server: `ENDTCPSVR SERVER(*HTTP) HTTPSVR(*ALL)`
   b. To start the server: `STRTCPSVR SERVER(*HTTP) HTTPSVR(*ALL)`
8. Press Enter.

Note: If you do not restart the HTTP Server, your changes will not take effect. You should now be able to access Analyzer for the Web from your browser by specifying the URL `http://TCPaddress/serverlib/index.html`

where:

`TCPaddress` is your iSeries TCP/IP address.

`serverlib` is the name of the library in which the Warehouse Manager Server is installed.

Note: If, after starting the server, you cannot access or display *index.html*, double-check the HTTP configuration file. Specifically, make sure that the entries that you made are not preceded by the “#” symbol. The “#” symbol is a comment character in the HTTP configuration file and all entries following it are ignored.

The file that contains the HTTP configuration on the iSeries is `QUSRSYS/QATMHTTPC`, member `CONFIG`. You can query against this file to retrieve the settings. A sample HTTP configuration file with the required settings in bold text is shown in Figure 3.

```
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *#
#   iSeries Internet Connection Server Configuration                                                              #
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ** * * * * * * * * #
#    HostName          your.full.host.name                                                                        #
#    The default port for HTTP is 80. If you change this use a port number greater than 1024                      #
#    Port              80                                                                                         #
#    Enable            GET                                                                                        #
#    Enable            HEAD                                                                                       #
#    Enable            POST                                                                                       #
#    Entries added for Analyzer for the Web                                                                       #
         Pass /showcasecorp/* /scserver/proddata/web/*
         Pass /SERVERLIB/* /SERVERLIB/ProdData/Analyzer/*
         Pass /serverlib/* /serverlib/ProdData/Analyzer/*
#        End of Analyzer for the Web section                                                                       #
          Welcome index.html
#   Map             /test/*          /as400/*                                                                     #
#   Pass            /as400/*         QDLS/400HOME/*                                                               #
#   Pass            /httpfile        /*QSYS.LIB/AS400LIB.LIB/HTML.FILE/*                                          #
#   Pass            /doc/*           /QDLS/graphics/*                                                             #
#   Pass            /file/*          /www/webdata/*                                                               #
#   Fail            /QDLS/TESTING/*                                                                               #
#   Redirect        /wsg             http://hostname:5061/WSG                                                     #
#   Exec            /cgi-bin/*       /QSYS.LIB/MYCGI.LIB/*                                                        #

# ***  DIRECTORY LISTINGS  ***                                                                                    #
#   DirAccess       On               DirShowSize                        On                                        #
#   Welcome         Welcome.html     DirShowBytes                       On                                        #
#   Welcome         index.html       DirShowOwner                       Off                                       #
#   AlwaysWelcome   On               DirShowDescription                 On                                        #
#   DirReadme       Top              DirShowMaxDescrLength              25                                        #
#   DirShowDate     On                                                                                            #

# ***  LOGGING  ***                                                                                               #
#   AccessLog          ACCESSLOG     InputTimeOut                       2 mins                                    #
#   ErrorLog           ERRORLOG      OutputTimeOut                      20 mins                                   #
#   LogFormat          COMMON        ScriptTimeOut                      5 mins                                    #
#   LogTime            LocalTime                                                                                  #
# * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *#
```

Figure 3. Sample HTTP configuration file

Additional configuration options are available to secure the HTTP Server. For more information on these settings, see *IBM TCP/IP Configuration and Reference Version 3*.

Configuring the Orion Server for Analyzer for the Web

To use Analyzer for the Web, you must install and configure an HTTP server. If you also use Enterprise Reporting, we recommend using Orion Application Server as an HTTP server. It is available during server installation.

To configure Orion Server:

1. Use the iSeries `ENDORI` command to stop the Orion Server and the `STRORI` command to restart it. For more information about these commands, see "Warehouse Manager Server Commands" on page 85 and the *Enterprise Reporting Administrator's Guide*.
2. Use the iSeries `WRKLNK` command to open the following XML file for editing:
   - Type `WRKLNK OBJ('/scserver/orion/application-deployments/default/defaultWebApp/orion-web.xml')`
   - Select option 2 to open *orion-web.xml*

   Note: As a precaution, you may want to create a backup of *orion-web.xml*.
3. Add the following bold lines of code to *orion-web.xml* (where `scserver` and `SCSERVER` is your server library), save the file, and close it.

```
<?xml version="1.0"?>
<!DOCTYPE orion-web-app PUBLIC "-//Evermind//DTD Orion Web Application 2.3//EN"
              "http://www.orionserver.com/dtds/orion-web.dtd">
<orion-web-app
  deployment-version="1.5.2"
  jsp-cache-directory="./persistence"
  temporary-directory="./temp"
  servlet-webdir="/servlet/"
>
  <virtual-directory virtual-path="/scserver/temp" real-path="/scserver/proddata/ANALYZER/TEMP" />
         <virtual-directory virtual-path="/showcasecorp" real-path="/SCSERVER/proddata/web/" />
  <virtual-directory virtual-path="/scserver" real-path="/scserver/proddata/Analyzer/" />
  <virtual-directory virtual-path="/SCSERVER" real-path="/SCSERVER/proddata/Analyzer/" />
          <orion-web-app>
```

4. To load this change, use the iSeries `ENDORI` command to stop the Orion server and the `STRORI` command to restart it. For more information about these commands, see Appendix A, "Commands" on page 85 and the *Enterprise Reporting Administrator's Guide*.
5. To access Analyzer for the Web, use the following URL: *http://iSeries:ORIONPORT/scserver/index.html*

The Orion port number was chosen during Orion installation. For information on installing the Orion Application Server, see the Enterprise Reporting Administrator's Guide.

Analyzer for the Web Support for Firewalls and Proxy Servers

Analyzer for the Web requires a direct socket connection to the Warehouse Manager Server to retrieve views and data. In a typical client/server installation, the Analyzer for the Web Java applet communicates with the Warehouse Manager Server using standard TCP/IP over a socket connection. A firewall or proxy server can prevent the applet from establishing a communication link to the server, making it necessary to configure the Analyzer for the Web Java applet to use the standard HTTP protocol in combination with the AOSServlet. The AOSServlet passes the requests from the applet to the Warehouse Manager Server.

Note: You must use the servlet with the IBM WebSphere Server or the Orion Server. The servlet and the server must be running on the same iSeries.

To Configure the AOSServlet with the IBM Websphere Server:

1. Add the *ShowCase.jar* file to the Java Classpath for your Web server. The new servlet is packaged as part of the same JAR file.
2. Add the new servlet. Choose a servlet name, such as *AOSServlet.* The class name should be *com.showcasecorp.aosservlet.AOSServlet*. There are no additional servlet properties required.
3. To configure the usage of the servlet interface in your HTML documents, you must change two applet parameters. You will change the `applet.html` document, which can be found in the IFS folder called */serverlib/ProdData/Analyzer/javaClient* where *serverlib* is the name of the library in which the Warehouse Manager Server is installed. Find the following two applet parameters:

   ```
   <PARAM name=UseCGI value="False">
   <PARAM name=CGIURL value="Servlet URL">
   ```

   Change the `UseCGI` parameter to `True` and the `CGIURL` to the URL for your servlet. For example, enter `http://IPADDRESS/servlet/`*AOSServlet*where `IPADDRESS` is the IP address your iSeries Web Server is running and *AOSServlet* is the same name that you configured in the third step.

   To test if the servlet is configured correctly, enter the CGIURL directly into your browser. You should see a message indicating that you have connected to the Analyzer for the Web servlet. If you do not get the message, then your servlet is either not configured correctly or the URL is incorrect.

   If the servlet is configured correctly and Analyzer for the Web still is unable to connect to the server, you may have to configure the `ServerIP` parameter to the applet in your html documents (for example, `applet.html`) as follows:

   ```
   <PARAM name=ServerIP value="localiseriesip">
   ```

where *localiseriesip* is the native IP address of the iSeries on which Warehouse Manager Server is running.

Note: If you want some clients to use the servlet interface and some to not use the servlet interface, you should have different HTML documents for each type.

To Configure the AOSServlet with the Orion Server:

1. Use the iSeries `WRKLNK` command to open the following xml file for editing:
   - Enter `WRKLNK OBJ('/scserver/orion/default-web-app/WEB-INF/web.xml')`
   - Select 2 to open *web.xml*

   Note: As a precaution, you may want to create a backup of *web.xml*.
2. Add the following bold linese of code to *web.xml* (where `TCPaddress` is your iSeries TCP/IP address and `OrionPort` is your Orion port number), save the file, and close it.

```
<?xml version="1.0"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<web-app>
  <!-- A demo servlet, add servlets below -->
  <servlet>
    <servlet-name>snoop</servlet-name>
    <servlet-class>SnoopServlet</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>AOSServlet</servlet-name>
    <servlet-class>com.showcasecorp.aosservlet.AOSServlet</servlet-class>
      <init-param>
          <param-name>UseCGI</param-name>
        <param-value>True</param-value>
      </init-param>
      <init-param>
        <param-name>CGIURL</param-name>
        <param-value>http://TCPaddress:OrionPort/servlet/AOSServlet</param-value>
      </init-param>
  </servlet> </web-app>
```

3. Follow the instructions in the third step of “To Configure the AOSServlet with the IBM Webspere Server,” on p. 79 but when changing the `CGIURL` to the URL for your servlet, enter *http://TCPaddress:OrionPort/servlet/AOSServlet* where *TCPaddress* is the TCP/IP address your iSeries Web Server is running and *OrionPort* is your Orion port number.
4. Use the iSeries `ENDORI` command to stop the Orion server and the `STRORI` command to restart it. For more information about these commands, see “Warehouse Manager Server Commands” on page 85 and the *Enterprise Reporting Administrator's Guide*.
5. To test if you've successfully configured the AOSServlet, use the following URL: `http://TCPaddress:OrionPort/servlet/AOSServlet`

where `TCPaddress` is the TCP/IP address your iSeries Web Server is running and `OrionPort` is your Orion port number.

This URL should display a simple page informing you that the servlet is properly configured and running.

Configuring an Existing HTTP Server for Analyzer for the Web

If the iSeries running the Warehouse Manager Server already has an HTTP Server running, the configuration may be more complex. At a minimum, you must specify to the HTTP Server that the documents in the IFS directory `/serverlib` (where `serverlib` is the name of the library in which Warehouse Manager Server is installed) can be served to client browsers.

In some cases, it is necessary to only add the configuration file entries described in the previous section. As long as you insert your pass entry above any map entries, you should not experience any problems. However, in other instances, the HTTP configuration file may contain a map entry pointing to a different IFS directory. In that situation, you must create a symbolic link that makes it appear as though the library containing Analyzer for the Web resides inside the directory specified in the map entry.

Note: If you are using an HTTP Server other than the iSeries HTTP Server, such as I/NET HTTP Server, it may be necessary to create a symbolic link to load the Analyzer for the Web HTML pages.

To Create a Symbolic Link:

1. From an iSeries command line, enter the command:

   ```
   ADDLNK OBJ('/serverlib')
   NEWLNK('/mapdirectory/serverlib')
   LNKTYPE(*SYMBOLIC)
   ```

   where:

   serverlib is the library in which Warehouse Manager Server is installed

   mapdirectory is the directory specified in the map entry of the HTTP configuration file.
2. Press Enter.
3. Restart the HTTP Server.
   a. To stop the server: `ENDTCPSVR SERVER(*HTTP) HTTPSVR(*ALL)`
   b. To start the server: `STRTCPSVR SERVER(*HTTP) HTTPSVR(*ALL)`

After creating a symbolic link, Analyzer for the Web users connect by specifying the library name of Warehouse Manager Server (for example, `http://serverlib/index.html`, where `serverlib` is the name of the library in which Warehouse Manager is installed). The HTTP Server automatically adds a prefix for the mapped directory.

Depending on your existing configuration, you may need to make additional changes to the configuration file. See IBM *TCP/IP Configuration and Reference Version 3* for more information.

Using National Language Support

The Warehouse Manager Server must interact with Analyzer using an ASCII CCSID to represent character data. Warehouse Manager Server uses an EBCDIC CCSID for Essbase/400, DB2/400, and OS/400. Defaults for both CCSID values are calculated at installation time based on your iSeries system settings. For additional information, including the supported EBCDIC and ASCII CCSIDs, see the “Installing Server Software” chapter in the *STRATEGY Installation Guide*.

The client CCSID setting is used during startup time. It must be a valid CCSID supporting translation to/from the EBCDIC CCSID. This does not affect what code page or encoding windows and web clients are able to run. The Analyzer OLAP part of the Warehouse Manager Server will support multiple windows and web clients on any code page or encoding that supports translation to a specific EBCDIC CCSID. To view and/or change the EBCDIC and ASCII CCSIDs for Analyzer OLAP, see the following sections.

Displaying the Current EBCDIC CCSID

1. To display the current EBCDIC CCSID value, enter the command:

   ```
   DSPFD serverlib/DBDEF
   ```

   where *serverlib* is the name of the Warehouse Manager Server library.
2. Press Enter.
3. Page down to locate the line specifying the coded character set identifier. This is the EBCDIC CCSID.

Changing the EBCDIC CCSID

If you need to change the EBCDIC CCSID value, you must reinstall Warehouse Manager Server and reconfigure the EBCDIC CCSID. See the “Installing Server Software” chapter in the *STRATEGY Installation Guide* for details.

Displaying the Default ASCII CCSID

1. To display the current ASCII CCSID value, enter the command:

   ```
   DSPPFM serverlib/CFG system
   ```

   where *serverlib* is the name of the Warehouse Manager Server library.

2. Press Enter.
3. Locate the line starting with `ClientCCSID=`. The value displayed is the default ASCII CCSID.

Changing the ASCII CCSID

1. Add the Warehouse Manager Server library to your library list.
2. From an iSeries command line, enter the command `CHGSCSVRA` and press Enter.
3. Change the value of the `ClientCCSID`.
4. Restart the server with the `STRSCSVR` command. See Chapter 1, "Starting and Stopping the Servers" on page 1 for details.

HTML Documents

Several HTML documents are installed when the Warehouse Manager Server is installed. These documents are installed into the IFS directory with the same name as the library in which it is installed.

Note: Analyzer for the Web is a Java-based application. An HTML document must exist before you can use a Java applet.

The Analyzer subdirectory contains sample Web pages. It employs the VMDetector applet to test the capabilities of the user's browser. As a result, the correct Web page locally installs the archives or launches the selected edition. You can use the pages as provided or modify them to meet custom needs.

The following document is installed in \\*serverlib**ProdData**ANALYZER:*

index.html is a home page from which users can launch Analyzer for the Web by linking to pages in the *javaClient* directory.

The following documents are installed in \\*serverlib**ProdData**ANALYZER**javaClient*:

launcher.html inside the *javaClient* directory launches the VMDetector with parameters specifying the splash screen image and version number.

applet.html inside the *javaClient* directory will launch Analyzer for the Web. See "Tailoring Analyzer Web Pages" on page 79 for more information.

default.html inside the *javaClient* directory acts as the all-purpose launch page for browsers with JDK 1.1 that are not Internet Explorer 4.0 or Netscape Communicator/Navigator 4.05 or higher.

troubleShoot.html
notifies the user in the event that the browser is not JDK

1.1-compliant and offers links to the latest versions of Microsoft Internet Explorer or Netscape Communicator/Navigator.

All code for the VMDetector, common images, and archives are stored in the shared directory. Analyzer's directory must be accessible to your Web Server (also known as an HTTP Server). To allow for this, define an additional document directory or a directory link using your Web Server configuration software. For additional information, see "Configuring the iSeries HTTP Server" on page 69.

Getting Users Started

Zero deployment is a major benefit of the browser client version of Analyzer for the Web. Provide all users with the Analyzer for the Web home page address or a specific launcher page. An example might be:

- `//system/serverlib/ProdData/Analyzer/Index`
- `//system/serverlib/Index.html`

Adding Databases

Before users can access databases from Analyzer or Analyzer for the Web, you must specify which databases are accessible. This is done through Warehouse Manager Client.

To add databases to be accessed with Analyzer or Analyzer for the Web:

1. Start Warehouse Manager Client and connect to the data source containing the Warehouse Manager Server.
2. In the list of objects, double-click **Analyzer OLAP**.
3. An item called Databases will appear. Right-click on **Databases** and choose **Add Database**.
4. In the Essbase System Login dialog box, enter the server IP address, Essbase/400 user ID, and password and click OK. A list of the databases on the server will appear.
5. Select the database(s) that you want to access from Analyzer by pressing Ctrl and clicking the database name.
6. Click **OK**.

The databases you selected will appear beneath the Databases object in the Warehouse Manager Client window (you may need to expand the Databases object to display the names). These databases are now accessible to users of Analyzer and Analyzer for the Web.

Working with iSeries Profiles

Although by default Analyzer and Analyzer for the Web require users to enter an iSeries profile to connect, it is not necessary to have an iSeries profile to use Analyzer or Analyzer for the Web.

1. From an iSeries command line, run the command `CHGSCSVRA`.
2. Change the value of the `AOValidateUser` setting. This setting controls whether users logging on to the server from either Analyzer or Analyzer for the Web must have a valid iSeries profile.

If the value is 0, an iSeries log on check is not performed and the log on is passed directly to Essbase/400. A non-zero value causes the server to validate against an iSeries profile. This is the default value.

Note: If you change the ValidateUser value to 0 so that iSeries profiles are not required, you cannot use Warehouse Manager Client to control application security for Analyzer or Analyzer for the Web. Application security requires an iSeries profile. Restart the server with the `STRSCSVR` command after changing this setting. See Chapter 1, "Starting and Stopping the Servers" on page 1 for details.

To simplify your security scheme for Analyzer and Analyzer for the Web, we highly recommend that you use the Integrated Security feature with Essbase/400. For more information, see the *Essbase/400 4.5 Administrator's Guide*.

Note: Some Analyzer OLAP features require a valid iSeries user ID and password. These features are not currently used by Analyzer, but they appear in Query and Report Writer. If you want to use scheduled queries or reports or access remote scheduling, users must log on with a valid iSeries profile.

You may turn off integrated security and still have some users enter valid iSeries user IDs and passwords. In that case, only those users are able to access the Analyzer OLAP functionality in Query and Report Writer.

Specifying an Analyzer Default User Profile: You may find it easier to set up one iSeries profile and use its settings as the default user profile settings. The `AODefaultUserProfile` parameter in the `CHGSCSVRA` command specifies an iSeries user profile as the default user profile. The first time a user connects to the server, the default user profile settings are assigned to the new user. For more information about the AODefaultUserProfile parameter, see "CHGSCSVRA Analyzer OLAP-Specific Settings" on page 88.

Tailoring Analyzer Web Pages

The sample set of Web pages provided is enough to support most installations. Feel free to copy and modify *index.html* and the pages in the *javaClient* directory with your own style and logos. However, if you decide to modify the pages, rename them so they will not be overwritten during upgrades or patches.

Note: The Web pages, Java classes, and archives in the shared directory should not be modified.

The Java components of specific Analyzer launcher pages—the VMDetector—stored in archives, can be modified by setting parameters passed to them by their associated APPLET tags. The *launcher.html* and *applet.html* files may be customized as long as they retain a valid APPLET tag.

```
<APPLET
width="0"
height="0"
code="VMDetector.class"
codebase="../shared/VMDetector"
name="Analyzer VMDetector"
>
<PARAM name="Splash" value="../shared/w_inter.gif">
<PARAM name="BuildNo" value="3.0.3.15902">
</APPLET>
```

Understanding APPLET Tag Components

An APPLET tag can be broken into two groups of components:

- Applet modifiers describe the applet to the browser. The applet modifiers are specified in the HTML standard and are available for use with all applets.
- Applet parameters pass values to the applet itself. The applet parameters are specific to the particular applet.

Applet Modifiers:

`WIDTH` Amount of room the browser will allow the applet to use horizontally.

`HEIGHT` Amount of room the browser will allow the applet to use vertically.

> **Note:** In addition to supporting fixed width and height, Internet Explorer and Netscape Communicator can specify the percentage of browser window width and height. For example: `WIDTH=99% HEIGHT=80%`

`CODE` The name of the applet's main class. Changing this value will not load the appropriate classes.

`CODEBASE` The URL of the applet's main class.

`NAME` Specifies a name for the applet instance. This makes it possible for applets on the same page to find and communicate with each other.

`ARCHIVE` The URL of the JAR archive that contains the classes.

Netscape browsers download the archive and use the classes within to accelerate startup. When specified in the *applet.html*, it should not be changed.

Note: When classes are installed locally, omit the ARCHIVE modifier.

Parameters: To set parameters for an applet, you specify a name and value. The names and values are specific to the applet and should be documented in its accompanying documentation.

Table 9. VMDetector parameters

Splash	The URL of the image to be displayed as the VMDetector is running.
BuildNo	Associates a version number with the VMDetector to be compared to that of the locally installed classes. If you use a custom page to launch the VMDetector, manually update this parameter when you install future versions of Analyzer.

Table 10. Analyzer supported parameters

AutoLogin	Default: "False". If set to "True", attempts a log on upon applet start using the values specified by the "UserID", "Password", and "Language" parameters.
UserID	The user name used for auto login.*
Password	The password used for auto login.*
Language	The language used for auto login.*
UseFrame	Default: "True". Causes the applet to run in a resizable frame outside the browser. To turn off this feature, set the parameter value to "False".
HelpItemText	Default: " " The value of this parameter is displayed on the Help menu.
HelpItemURL	Default: " " The URL of the page to load when the user selects the value of HelpItemText from the Help menu.
BGColor	Default: " " The RGB color specification for the background color of the applet. The default value specifies light gray.
ChartLegLimit	Default: " " Specifies how many items will show on the X axis before they are turned to display vertically. For example, if the value is "5" and your chart has six items, all six are turned so that they display vertically.
Port	The port number to use for connecting to the Warehouse Manager Server. This value must match the configuration of the server.
LoginHint	Default: " " Changes the hint on the Login dialog box to provide a customizable logon message.
TmpImageDirURL	Default: " " The URL of the directory where AOS is storing images. The default specifies the default location.

Table 10. Analyzer supported parameters (continued)

ServerIP	Default: " " The IP address or the fully qualified DNS name for the server. If the value is specified, the Warehouse Manager Server must reside on that machine.
UseCGI	For firewall and proxy support. See "Analyzer for the Web Support for Firewalls and Proxy Servers" on page 73.
CGIURL	For firewall and proxy support. See "Analyzer for the Web Support for Firewalls and Proxy Servers" on page 73.

*Used only if the AutoLogin parameter is set to "True".

About Distributed Archive Files

Advances in Web browser and Java technology make it possible to locally install the classes necessary to run Analyzer for the Web. The advantages of local installation are faster applet startup and less network traffic. The necessary classes will only be reinstalled and redownloaded from the Web server when new versions of Analyzer for the Web are installed. Currently, Analyzer for the Web supports automatic installation of local archive files for use with Netscape Communicator 4.05 and Microsoft Internet Explorer 4.0 or later versions.

Microsoft Internet Explorer and CAB files

Microsoft Internet Explorer uses CAB archives for locally installed classes. Using Microsoft-specific applet parameters, the files are updated by the Microsoft Internet Explorer's Package Manager whenever new versions are located on the server.

In the installed sample, *IESample.html* in the *samples* folder contains the necessary applet tag with the parameters to install the CAB contained in the same folder.

Table 11. Microsoft-specific CAB parameters used in IESample.html

namespace	Default: "ShowCaseAnalyzer". The name space into which the classes contained in the CAB should be installed. *This value should not be changed*.
useslibrary	Default: "ShowCaseAnalyzer." The name space of the library classes. *This value should not be changed*.
useslibrarycodebase	Default: "shared\ShowCase Analyzer.cab". The URL to the Analyzer for the Web CAB file.
useslibraryversion	Default: "3,0,3,15902". The version number corresponding to the version of the CAB files. When installing new versions or patches of Analyzer for the Web, this parameter must be manually updated in custom pages using CAB files. This number will vary.

Uninstall CAB Files: To manually uninstall CAB files through Internet Explorer:

1. Open Internet Explorer, making sure that Analyzer for the Web is not loaded.
2. From the Tools menu, choose **Internet Options**.
3. In the Internet Options dialog box, select the **General** tab.
4. In the Temporary Internet Files section, click **Settings**. The Settings dialog box will appear.
5. To open a folder with all the installed, downloaded CAB files and components, click the **View Objects** button. The Downloaded Program Files dialog box will appear.
6. Select the Analyzer component that you want to uninstall.

To uninstall the Analyzer classes:

1. In the Downloaded Program Files dialog box, select the component(s) that you want to remove.
2. From the File menu, choose **Remove Program File**.
3. From the File menu, choose **Close**.
4. In the Settings dialog box, click **OK**.
5. In the Internet Options dialog box, click **OK**.

Netscape Communicator and JAR Files

The sample Web pages use *netscape.html* in the shared folder to install new versions of the classes as they are available from your server. The file contains JavaScript that uses Netscape's SmartUpdate to install the JAR files and maintain version information. As noted earlier, this page should not be changed. However, *smartupdate.js* (in the *shared* directory) may be used by your custom pages to install the local classes for Netscape browsers. The *NSLauncher.html* sample page demonstrates its use.

Uninstalling JAR Files: Netscape 4.5 users can uninstall the locally installed classes:

1. Open Netscape Communicator, making sure that Analyzer for the Web is not running.
2. From the Edit menu, choose **Preferences**.
3. From the category list, double-click **Advanced**.
4. Click **SmartUpdate** to open the SmartUpdate panel.
5. Select **Analyzer**.
6. Click **Uninstall**.
7. Click **OK**.

Sample Pages

The directory contains several pages showing how to customize the pages that launch Analyzer for the Web. Feel free to use and modify these pages to meet your needs.

Table 12. Sample pages

IESample.html	Microsoft Internet Explorer only. Illustrates the use of the Microsoft-specific parameters to locally install the CAB and launch the applet.
NoFrame.html	Launches the applet within the browser′s window embedded in an HTML page.
NSLauncher.html	Netscape Communicator only. Launches the applet within the browser′s window embedded in an HTML page and installs the JAR file as needed.
Analyzer640.html	Launches the applet within the browser′s window embedded in an HTML page and sized for a 640 monitor resolution.

Appendix A. Commands

This appendix lists many of the common iSeries commands, Warehouse Manager Server commands, and Essbase/400 commands you will need for maintaining your iSeries software.

iSeries Commands

Table 13 lists common iSeries commands for maintaining server libraries.

Table 13. 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.

Warehouse Manager Server Commands

The following tables list the common iSeries Warehouse Manager Server commands for maintaining a Warehouse Manager library. For related security information, see "Command Authorities" on page 64.

Note: Before using these commands, add the Warehouse Manager library to your iSeries library list and make sure that you have only one Warehouse Manager library in your library list. Run the command `ADDLIBLE serverlib`, where `serverlib` is the name of the library running the Warehouse Manager Server that you want to work with

.

Table 14. Common Warehouse Manager Save/Restore commands

Command	Description	Usage
SAVSRVRINF	Save server information.	To back up the Warehouse Manager Server or move data stored on the server to another library on the same iSeries or a different iSeries.
RSTSRVRINF	Restore server information.	To restore the server data saved using the SAVSRVRINF command.

Table 15. Common Warehouse Manager TCP/IP commands

Command	Description	Usage
CHGSCSVRA	Add TCP/IP connectivity support or control Analyzer OLAP functionality (see Table 20).	To enable or disable TCP/IP support after installation. Requires *ALLOBJ, *SECADM, *IOSYSCFG, *JOBCTL, and *SAVSYS authorities. **Note:** Before using the CHGSCSVRA command, end the server with ENDSCSVR. After using the CHGSCSVRA command, restart the server with STRSCSVR.
ENDORI	End Orion server.	To end the Orion server. Requires *ALLOBJ and *JOBCTL authorities. For more information about the Orion server, see the *Enterprise Reporting Administrator's Guide*.
ENDSCSVR	End Warehouse Manager Server.	To end TCP/IP. For example, if you need to restart TCP/IP, you should end TCP/IP first. Requires *ALLOBJ and *JOBCTL authorities.
INSTORI	Install Orion server.	To install the Orion server. For more information about the INSTORI command, including parameters, see the *Enterprise Reporting Administrator's Guide*.
MIGSCREP	Migrate objects and documents to 4.5 Web deployment framework.	To migrate 3.0, 3.5, 4.0, or 4.1 Information Directory objects and documents to the 4.5 Web deployment framework. Requires QSECOFR or equivalent authority (*ALLOBJ, *IOSYSCFG, *SAVSYS, *JOBCTL, and *SECADM special authorities are needed). For more information, see the *Enterprise Reporting Administrator's Guide*.
RSTSMPLB	Restore sample database.	To install the sample database if not previously installed, or to restore the sample database back to its original structure (if database files were altered or destroyed).

Table 15. Common Warehouse Manager TCP/IP commands (continued)

Command	Description	Usage
SETJDEOWA	Enable/Disable non-iSeries profile sign-on for J. D. Edwards OneWorld users.	To enable/disable J. D. Edwards OneWorld connections for users who do not have an iSeries user profile. For more information, see “Non-iSeries Profile Sign-On” on page 44.
STRORI	Start Orion server.	To start the Orion server. Requires *JOBCTL authority. For more information about the Orion server, see the *Enterprise Reporting Administrator's Guide*.
STRSCSVR	Start Warehouse Manager Server.	After an IPL, ENDSCSVR, or after configuring TCP/IP with CHGSCSVRA. Requires at least *JOBCTL authority.
STRTCP	Start iSeries TCP.	To start iSeries TCP/IP. iSeries TCP/IP must be started before you can enable or start TCP/IP support.

Table 16. Common Warehouse Manager Server information commands

Command	Description	Usage
CHGSCAUT	Change/enter Warehouse Manager passwords.	To enter license information.
CHKSRVRCAT	Check server catalog information.	To inspect Warehouse Manager catalogs and delete obsolete Warehouse Manager settings.
CHGSRVRSTS	Change server status.	To change the status of the Warehouse Manager Server library (for example, SCSERVER) to and from a restricted state. A restricted state prevents users from connecting to the server library.
DSPSCAUT	Display Warehouse Manager authority.	To see the version of Warehouse Manager installed and the products and licensing information.
DSPSCINF	Display STRATEGY hints and tips.	To locate hints and tips provided by SPSS Inc.

Table 17. Common Warehouse Manager Server options commands

Command	Description	Usage
CHGSRVRDFT	Change server defaults.	To change the default *PUBLIC authority to all libraries or tables and enforce Warehouse Manager security rules on *ALLOBJ profiles.
SETLIBACC	Set library list access.	To restrict users so they can access only the libraries in their iSeries library lists, or to give users access to all libraries on the iSeries to which they are authorized.

Table 17. Common Warehouse Manager Server options commands (continued)

Command	Description	Usage
`SCCHKPWD`	Manage iSeries passwords stored in control tables (see Table 21).	Update the STRATEGY 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.
`CMPSCJAR`	Compile STRATEGY Java files.	The Warehouse Manager Server uses Java. STRATEGY will install several required *.jar* files in your Integrated File System (IFS).To reduce server start-up time, compile these files on the system running Warehouse Manager Server before you start the server. Compiling the files may affect system performance during the compile. To prevent this, you can compile immediately after the installation finishes or enter a date and time to begin compiling.

Table 18. Warehouse Manager Server commands for Warehouse Builder

Command	Description	Usage
`STRDD`	Start data distribution.	Runs a Warehouse Builder distribution set immediately. Can run by set name (`OBJNAME`) or ID (`OBJID`). For example, `STRDD OBJNAME ('Sales')` runs the Sales distribution set.
`SBMDD`	Submit data distribution.	Submits a Warehouse Builder distribution set to be run. Can run by set name (`OBJNAME`) or ID (`OBJID`). For example, `SBMDD OBJID(51)` runs the Sales distribution set with an ID of 51.

Table 19. Uninstall command

Command	Description	Usage
`UNINSTSC`	Uninstall a Warehouse Manager or Essbase/400 library.	To remove a library from the system. For more information about the `UNINSTSC` save file, see "Uninstalling a Warehouse Manager Library" on page 52.

CHGSCSVRA Analyzer OLAP-Specific Settings

Before running `CHGSCSVRA`, stop the Warehouse Manager Server with `ENDSCSVR`. Analyzer, Analyzer for the Web, Analyzer Designer, Query, and Report Writer use part of the server as a common repository for storing various objects, including queries, reports, views, forms, pins, pinboards, and lists. Additionally, the server loads the view definitions, retrieves the data from Essbase/400, and packages and sends the data to the Analyzer for the Web Client. Table 20 contains `CHGSCSVRA` settings for maintaining this functionality.

Table 20. CHGSCSVRA settings for Analyzer functionality

Setting	Description
AOEncrypt	Enables data stream encryption. The default is encryption enabled. This is a Boolean value with 1 being "enabled" and 0 being "disabled."
AOAllowNewUserDatabaseAccess	Enables a prompt for username and password when the user ID and password used to log on to Analyzer are not valid for a particular Essbase/400 database. The default is no prompting. This is a Boolean value with 1 being "enable prompting" and 0 being "disable prompting."
AOSavePasswordMode	Enables saving of the default preferences whenever users who do not currently have preferences set log on to the Server. Setting the preferences enables them to be viewed in Warehouse Manager. The default is not to save these settings. This is a Boolean value with 1 being "save the default settings" and 0 being "do not save the settings."
AODefaultUserProfile	Used the first time a user connects to the Server. The user settings associated with the default user profile, including the system manager setting, are assigned to the new user. You can also (optionally) reset the settings for all existing users. **Note:** If you change the existing default user's settings and want to apply them to the current users, you must run the CHGSCSVRA command and set the Update All Users option to *YES. Enter a blank ('') profile to turn off the use of the default profile for new users. When specifying default user profile, a conditional prompt asks whether to update the settings for all users. Choosing this option will apply the user settings associated with the default profile to all users. If users are connected to the server at the time this command is entered, their user settings will not be updated. The AOValidateUser parameter cannot equal 0 if you want to use the AODefaultUserProfile setting.
AOAllowDataExport	Disables or enables the data export functions within Analyzer for an individual user or for all users, including e-mail, Lotus Notes, Excel, Text File, Lotus 123W, and the Windows clipboard. If the AOAllowDataExport setting is not specified, the default is to always allow data exporting. **Note:** This setting affects only existing users who have previously signed on to Analyzer. To apply this setting to new users as they sign on, turn on the AODefaultUser option using CHGSCSVRA. The new settings are returned to the Analyzer Client when the user signs on. It will disable the appropriate commands at that time.

Table 20. CHGSCSVRA settings for Analyzer functionality (continued)

Setting	Description
AOAllowPrint	Disables or enables the print functions within Analyzer for an individual user or for all users, including e-mail, Lotus Notes, Excel, Text File, Lotus 123W, and the Windows clipboard. If the AllowPrint setting is not specified, the default is to always allow printing. **Note:** This setting affects only existing users who have previously signed on to Analyzer. To apply this setting to new users as they sign on, turn on the DefaultUser option using CHGSCSVRA. The new settings are returned to the Analyzer Client when the user signs on. It will disable the appropriate commands at that time.
AOBlockLoginAfter	Limits the number of logon attempts for a particular user ID. When the number of consecutive logon failures surpasses the value specified, no further attempts are allowed until a specified time has passed. The default is 3. **Note:** Logon blocking is based on the client IP and user ID. If a particular user fails to log on, the user is not allowed to log on from the same IP until the expiration period has expired (600 seconds; see BlockLoginFailureRetrySecs below). Upon reaching the point where logins are blocked, the full Windows version of Analyzer will exit. However, Analyzer for the Web does not exit the browser. Nonetheless, the user may disconnect and reconnect at any time.
AOBlockLoginFailureRetrySecs	Specifies the time a user must wait before attempting another logon when the failure count is exceeded. This value is in seconds and the default is 600 (10 minutes).
AOInitialThreadCount	Defines the initial number of threads created when the Server is started. Since the number of threads will automatically increase if necessary, this setting does not normally need to be specified. Blank for default or number 0 to N. The default is 6 initial threads.
AOMaximumIdleThreadCount	Defines the maximum number of idle threads that the server allows. A certain number of idle threads process additional client requests quickly. However, excessive idle threads may use unnecessary system resources. By default, a small number of threads are kept idle. Idle threads in excess of this amount are terminated naturally. This setting does not normally need to be modified. This setting applies to idle threads, *not* idle client connections. Blank for default or number 0 to N. The default is to have InitialThreadCount + 12 idle threads.

Table 20. CHGSCSVRA settings for Analyzer functionality (continued)

Setting	Description
AOMaximumThreadCount	Defines the maximum number of threads allowed by the Server. Client requests are processed in the order in which they are received if there are more client requests than threads available to do the work. If this setting is 0, all client requests are processed on the main thread. This setting takes precedence over the InitialThreadCount and MaximumIdleThreadCount settings. Blank for default or number 0 to N. The default is to have an unlimited amount of threads.
SecondaryTCPIPPort	Configures the port number that is associated with the Warehouse Manager Server secondary TCP/IP service. A Warehouse Manager Server can have a secondary port only if it has been upgraded from 3.5 or an earlier release. The service name is formed internally by the Warehouse Manager Server support. It is formed by appending the library name of the Warehouse Manager Server support to SCOLAPSVR. For a Warehouse Manager Server named SCSERVER, the service name becomes SCOLAPSVRSCSERVER. To check for available port numbers, use the OS/400 WRKSRVTBLE command to display existing port assignments. Valid KEYVAL settings can be 5000 to 65535. The default is 23001. When specifying TCPIPPort, a conditional prompt asks whether to update the Analyzer OLAP HTML files with the specified port number. If you are using the Analyzer for the Web JAVA client to connect to the Server through the secondary server TCP/IP port, you must configure the HTML files to use the port the Warehouse Manager Server is listening on.
AOTempImagePath	Define the path on the Server for temporary storage of images used with Analyzer for the Web.
AOValidateUser	Controls whether users logging on to the Server from Analyzer or Analyzer for the Web must have a valid iSeries logon. If the setting is 0, an iSeries logon check is performed and the logon is passed directly to Essbase/400. This setting is used if your system has only Essbase/400 profiles and no iSeries profiles. A setting of non-zero causes the Server to validate against an iSeries profile. This is the default.
AOPrimaryLanguage	Defines the primary language the Server should use for iSeries job logging. This value has no effect on the language used by the client PC connecting to the Server. Valid settings are ENP (uppercase English) and ENU (lowercase English). The default is ENU.

Table 20. CHGSCSVRA settings for Analyzer functionality (continued)

Setting	Description
AOClientCCSID	Specifies the default Windows ANSI code page that clients run under. This setting is used only for the initial communication with the client and if a translation table between the Server CCSID and the client-specified CCSID cannot be established. The default client CCSID is 1252.
TCPIPPort	Configures the port number associated with TCP/IP service. The number uniquely identifies the TCP/IP service provided by each installation of support. The number specified here creates a TCP/IP service table entry and cannot already be defined for any other services. If multiple copies of Warehouse Manager Server support are installed on the iSeries, each copy has its own service name, which is formed by appending the library name of the support to SCSERVER. For a server named SCAOSVR, the service name becomes SCSERVERSCAOSVR. Valid settings are from 5000 to 65535. The default is 43419. When specifying TCPIPPort, a conditional prompt asks whether to update the Analyzer HTML files with the specified port number. If you are using the Analyzer for the Web Java Client to connect to AOS, you must configure the HTML files to use the port associated with AOS.

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 21 lists the *optional* SCCHKPWD parameters.

Table 21. SCCHKPWD optional 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.
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-ShowCase 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.

Appendix B. Sample Databases

This appendix describes the sample databases that you can install with Warehouse Manager Server on the iSeries. The databases are:

- SCSAMPLE45, a more summarized database in a star schema format that represents an OLAP-type database
- SCDB45, a fairly normalized database that represents an OLTP-type database

Installing the Sample Databases

When you install Warehouse Manager Server on the iSeries, you have the option to install the sample database. Selecting this option creates a library on the iSeries called SCSAMPLE45. This library contains the files for the SCSAMPLE45 sample database and the iSeries save files for creating other sample databases, such as SCDB45.

SCSAMPLE45

The files (tables) for the SCSAMPLE45 sample database are installed in the SCSAMPLE45 library. To install the SCSAMPLE45 library on the iSeries, use one of the following methods:

- Select the option to install the sample library as part of the initial installation of Warehouse Manager Server.
- Run the Restore Sample Library (RSTSMPLB) command on the iSeries command line. For more information, see “Installing the Sample Library after Installation with RSTSMPLB” on page 118.

SCDB45

Before you can use the SCDB45 sample database, you must do a small amount of setup on the iSeries. The files for the SCDB45 sample database are shipped in an iSeries save file, called SCDB, within the SCSAMPLE45 library.

Note: An iSeries save file is an iSeries object with type *FILE and attribute *SAVF. A save file contains data that have been saved using one of the iSeries save commands. To use the data from a save file, you must extract the data from the save file using one of the iSeries restore commands.

To set up the SCDB45 sample database:

1. Install the SCSAMPLE45 database (library).
2. Run the following command on the iSeries command line:

```
RSTLIB SAVLIB(SCDB45) DEV(*SAVF) SAVF(SCSAMPLE45/SCDB) MBROPT(*ALL)
ALWOBJDIF(*ALL) RSTLIB(SCDB45)
```

This command extracts the files (tables) from the SCDB save file in the SCSAMPLE45 library and restores them into the SCDB45 library.

Contents of the Sample Databases

The sample databases are the basis for the STRATEGY tutorials and the training manuals. The databases contain data for a fictitious company called The Outdoor Connection. The Outdoor Connection is a sporting goods company that sells through three channels: retail stores, specialty stores, and catalogs. Its primary product lines are skiing equipment, biking equipment, camping gear, and clothing. The Outdoor Connection is based in the United States. It also does business in Canada, Belgium, Germany, France, the United Kingdom, Japan, Australia, and New Zealand.

Note: We recommend that you do not modify the original sample database in any way. If it is necessary for you to customize the sample database, copy it to a new library and make your modifications there.

This appendix describes the sample database that you can install with Warehouse Manager Server. The sample database tables are installed into a library called SCSAMPLExx, where xx is the current product version (for example, SCSAMPLE45).

Figure 4 depicts the general structure of the sample database.

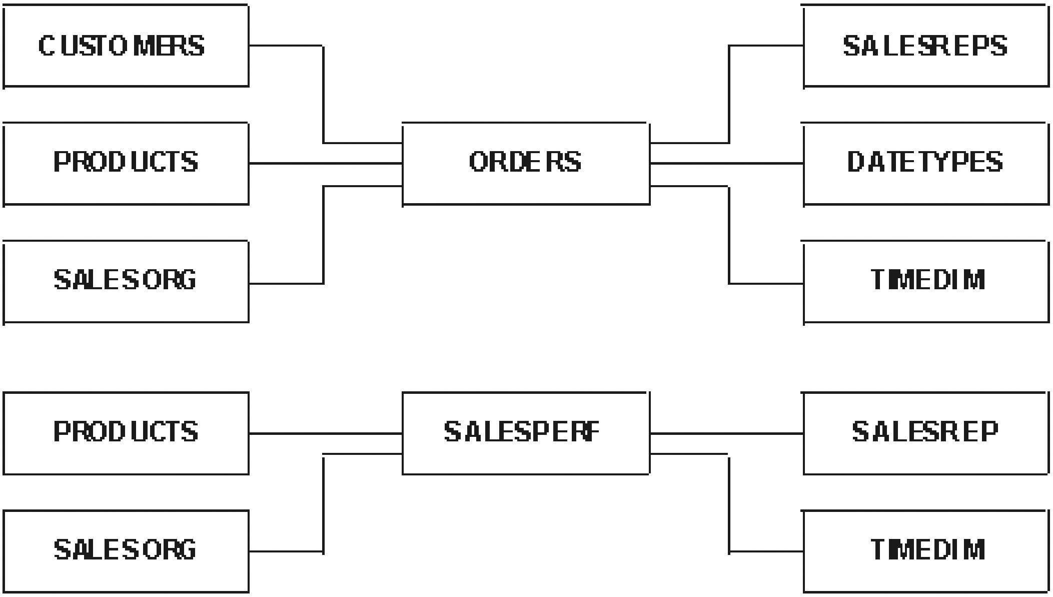


Figure 4. Sample database structure

SCSAMPLE45 Database

ORDERS Table

The ORDERS table is the basis of all transaction information. It contains the order number, the location placing the order, the date of the order, the promised delivery date, the date shipped, etc. The table also includes a column that records the location to which each order is shipped. This accommodates the situations in which the order is shipped to a location other than where the order originates.

The ORDERS table also contains the product-related specifics of each order, such as the quantity ordered, quantity shipped, quantity backlogged, and quantity canceled for each product. In addition, the table tracks the unit price, unit cost, extended price, and extended cost for each product ordered. Finally, for international sites, the table includes exchange rates. The rates are used to calculate the foreign extended price and foreign extended cost for each product based on the exchange rate at the time of the order.

The exchange rate is an unsigned percentage and can be mapped to a currency code for each country in which The Outdoor Connection conducts business. It is assumed that all monetary amounts entered into the database are entered in U.S. dollars. Therefore, by definition, the exchange rate for U.S. dollars is 1.0. The exchange rate for other currencies is based on the amount that one U.S. dollar will buy. For example, $1 U.S. recently bought $1.4575 CA. Therefore, the exchange rate for Canadian dollars is 1.4575. To convert an amount in the database into local values, multiply the amount by the exchange rate:

$10 US x 1.4575 = $14.58 CA or $10 US x 0.6105 = £6.11

This table contains 14,379 rows of data.

Table 22. ORDERS table

Column	Description	Heading	Data Type	Contents
ORDNUM	Order Number	Order Number	Decimal(8)	Order number
CUST_ID	Customer ID	Customer ID	Numeric(8)	Account ID of client
SHIP_TO	Shipping Destination	Ship To	Numeric(8)	Account ID of shipping destination
PRODUCTID	Product ID (SKU)	Product ID	Numeric(8)	Product ID
PRDLVL03	Product Level 3	Product Level 3	Char(8)	Third-tier product category
QTYORD	Quantity Ordered	Quantity Ordered	Decimal(15)	Quantity ordered
QTYSHP	Quantity Shipped	Quantity Shipped	Decimal(15)	Quantity shipped
QTYBACK	Quantity Backordered	Quantity Backordered	Decimal(15)	Quantity back-ordered
QTYCAN	Quantity Cancelled	Quantity Cancelled	Decimal(15)	Quantity canceled
DATEREQ	Date Requested	Date Requested	Date	Date requested for delivery
DATEORD	Date Ordered	Date Ordered	Date	Date ordered
DATEPROM	Date Promised	Date Promised	Date	Date promised for delivery
DATESHIP	Date Shipped	Date Shipped	Date	Date shipped
DATEINV	Date Invoiced	Date Invoiced	Date	Date invoiced
UNIT_PRICE	Unit Price	Unit Price	Decimal(15,4)	Unit retail price
UNIT_COST	Unit Cost	Unit Cost	Decimal(15,4)	Unit wholesale cost (cost of goods sold)
EXTPRICE	Extended Price	Extended Price	Decimal(15,4)	Extended retail total (the product of QTYORD and UNIT_PRICE)
EXTCOST	Extended Cost	Extended Cost	Decimal(15,4)	Extended wholesale total (the product of QTYORD and UNIT_COST)
MEASURE	Unit of Measure	Unit of Measure	Char(4)	Unit of measure code
CURR_CODE	Currency Code	Currency Code	Char(3)	Currency code

Table 22. ORDERS table (continued)

Column	Description	Heading	Data Type	Contents
EXCHG_RATE	Exchange Rate	Exchange Rate	Decimal(15,7)	Exchange rate
FRGN_XPRIC	Foreign Extended Price	Foreign Ext Price	Decimal(15,4)	Foreign extended retail total
FRGN_XCOST	Foreign Extended Cost	Foreign Ext Cost	Decimal(15,4)	Foreign extended wholesale total
DIVSN_ID	Division ID	Division ID	Char(8)	Division code
REGION_ID	Region ID	Region ID	Char(8)	Region code
AREA_ID	Area ID	Area ID	Char(8)	Area code
DIST_ID	Distribution Center ID	Distribution Cntr ID	Char(8)	Distribution center code
REPCODE	Sales Rep Code	Rep Code	Char(4)	Account representative's code

CUSTOMERS Table

The CUSTOMERS table contains information about The Outdoor Connection's clients. It includes names and addresses and divides the clients into various geographical categories. The Outdoor Connection consists of 3 divisions (North America, Europe, and Asia/Pacific); 12 regions (East, West, South, Midwest, Canada, Belgium, Germany, France, United Kingdom, Japan, Australia, and New Zealand); 70 areas (one for each state and province, including the District of Columbia, plus one for each European and Asia/Pacific country); and 3 distribution facilities (Seattle, Kansas City, and New York). In addition, the CUSTOMERS table includes a currency code for each client, a status code that tracks whether a client is active or inactive, and a code that identifies each client's sales representative.

The CUSTOMERS table contains sufficient data to analyze customers according to two separate hierarchical structures:

Sales organization structure

- Division
- Region
- Area

Geographic location

- Country
- State/Province
- City
- Postal code

This table contains 157 rows of data.

Table 23. CUSTOMERS table

Column	Description	Heading	Data Type	Contents
CUST_ID	Customer ID	Customer ID	Numeric(8)	Account ID of client
CUSTNAME	Customer Name	Customer Name	Char(100)	Name of client
LNAME	Surname	Last Name	Char(50)	Surname of client contact
FNAME	Given Name	First Name	Char(50)	Given name of client contact
ADDRESS1	Address 1	Address 1	Char(100)	Address line 1
ADDRESS2	Address 2	Address 2	Char(100)	Address line 2
CITY	City	City	Char(50)	City name
STATE	State/Province Abbreviation	State/Province Abbr	Char(2)	State/province postal abbreviation
STATEDESC	State/Province Description	State/Province	Varchar(80)	State/province name
POSTALCODE	Postal Code	Postal Code	Char(16)	Postal code
COUNTY	County	County	Varchar(80)	County name
COUNTRY	Country Code	Country Code	Varchar(12)	Country code
COUNTRYDESC	Country Description	Country	Varchar(80)	Country name
PHONE	Phone Number	Phone Number	Char(24)	Phone number
E_ADDRESS	E-mail Address	E-mail Address	Char(100)	E-mail address of client contact
LOB	LOB	LOB Code	Char(8)	Line of business code
LOBDESC	Line of Business Description	Line of Business	Char(80)	Line of business description
REPCODE	Sales Rep Code	Rep Code	Char(4)	Account representative's code
ENTRY_DATE	Entry Date	Entry Date	Date	Date added
ACTIVE	Active Code	Active	Char(1)	Active status (Y or N)
DIVSN_ID	Division ID	Division ID	Char(8)	Division code
DIVSN	Division	Division	Char(80)	Division name
REGION_ID	Region ID	Region ID	Char(8)	Region code
REGION	Region	Region	Char(80)	Region name
AREA_ID	Area ID	Area ID	Char(8)	Area code
AREA	Area	Area	Char(80)	Area name
DIST_ID	Distribution Center ID	Distribution Cntr ID	Char(8)	Distribution center code

Table 23. CUSTOMERS table (continued)

Column	Description	Heading	Data Type	Contents
DIST_LOC	Distribution Center	Distribution Center	Char(80)	Distribution center name
CURR_CODE	Currency Code	Currency Code	Char(3)	Currency code

PRODUCTS Table

The PRODUCTS table contains all of The Outdoor Connection's product information and groups products hierarchically. Each product has three levels associated with it. The first identifies a product group (for example, clothing, biking, skiing, etc.), the second, a type (for example, outerwear, shirts, pants, etc.), and the third, a style (for example, black, red, yellow, etc.). In addition, the PRODUCTS table includes a unit price and a unit cost for each product. It also contains a currency code so the unit price and cost can be converted for various countries. Finally, it includes a measures column with a unit type (for example, *one* bicycle, *a pair* of skis, etc.) and a date that records when information was last changed.

This table contains 120 rows of data.

Table 24. PRODUCTS table

Column	Description	Heading	Data Type	Contents
PRODUCTID	Product ID (SKU)	Product ID	Numeric(8)	Product ID
PRODUCT	Product Description	Product Description	Char(80)	Product description
PRDLVL01	Product Level 1	Product Level 1	Char(8)	First tier product category
PRDDESC01	Product Description 1	Description Lvl 1	Char(80)	First tier product description
PRDLVL02	Product Level 2	Product Level 2	Char(8)	Second tier product category
PRDDESC02	Product Description 2	Description Lvl 2	Char(80)	Second tier product description
PRDLV03	Product Level 3	Product Level 3	Char(8)	Third tier product category
PRDDESC03	Product Description 3	Description Lvl 3	Char(80)	Third tier product description
PRDLVL04	Product Level 4	Product Level 4	Varchar(8)	Fourth tier product category
PRDDESC04	Product Description 4	Description Lvl 4	Varchar(27)	Fourth tier product description

Table 24. PRODUCTS table (continued)

Column	Description	Heading	Data Type	Contents
PRDLV05	Product Level 5	Product Level 5	Varchar(8)	Fifth tier product category
PRDDESC05	Product Description 5	Description Lvl 5	Varchar(27)	Fifth tier product description
UNITPRICE	Unit Price	Unit Price	Decimal(15,4)	Unit retail price
UNITCOST	Unit Cost	Unit Cost	Decimal(15,4)	Unit wholesale cost (cost of goods sold)
CURR_CODE	Currency Code	Currency Code	Char(3)	Currency code
MEASURE	UOM Code	UOM Code	Char(4)	Unit of measure code
MEASURE_DESC	Unit of Measure	Unit of Measure	Char(80)	Unit of measure description
ENTRY_DATE	Entry Date	Entry Date	Date	Entry/update date

SALESORG Table

The SALESORG table contains the geographic structure used in the database. Essentially, it maps the various divisions, regions, and areas to each other. It also indicates which distribution facility serves each area or foreign region. As explained previously, the database contains 3 divisions, 12 regions, 70 areas, and 3 distribution facilities.

This table contains 70 rows of data.

Table 25. SALESORG table

Column	Description	Heading	Data Type	Contents
DIVSN_ID	Division ID	Division ID	Char(8)	Division code
DIVSN	Division	Division	Char(80)	Division name
REGION_ID	Region ID	Region ID	Char(8)	Region code
REGION	Region	Region	Char(80)	Region name
AREA_ID	Area ID	Area ID	Char(8)	Area code
AREA	Area	Area	Char(80)	Area name
DIST_ID	Distribution Center ID	Distribution Cntr ID	Char(8)	Distribution center code
DIST_LOC	Distribution Center	Distribution Center	Char(80)	Distribution center name

SALESREPS Table

The SALESREPS table maps a salesperson’s code to a name, employee ID, and manager code. It also maps the salespeople to regions so they can be tracked and evaluated by geography. Each domestic region contains three salespeople; Canada and each European region have two, and each Asia/Pacific region has one.

This table contains 25 rows of data.

Table 26. SALESREPS table

Column	Description	Heading	Data Type	Contents
EMP_ID	Employee ID	Employee ID	Numeric(8)	Employee number
REPCODE	Sales Rep Code	Rep Code	Char(4)	Account representative's code
MGRCODE	Manager Code	Manager Code	Char(4)	Supervisor's code
LASTNAME	Surname	Last Name	Varchar(100)	Surname of employee
FIRSTNAME	Given Name	First Name	Varchar(100)	Given name of employee
REGION_ID	Region ID	Region ID	Char(8)	Region code
REGION	Region	Region	Varchar(80)	Region name
ENTRY_DATE	Entry Date	Entry Date	Date	Entry date (date of hire)
ACTIVE	Active	Active	Char(1)	Active status (Y or N)

DATETYPES Table

The DATETYPES table is used to demonstrate how the product applications handle date values stored as a data type other than `DATE`. Specifically, the `DATEINV` column has been replicated into several formats. You can use this table independently or join it to the ORDERS table on `ORDNUM`.

This table contains 14,379 rows of data.

Table 27. DATETYPES table

Column	Description	Heading	Data Type
ORDNUM	Order Number	Order Number	Decimal(8)
DATEINV	Date Invoiced	Date Invoiced	Date
DATEINV_YY	Two Digit Year Value	YY	Decimal(2)
DATEINV_MM	Two Digit Month Value	MM	Decimal(2)
DATEINV_DD	Two Digit Day Value	DD	Decimal(2)
DATEINV_EXCELSRL	Five Digit Excel Serial Date	EXCELSRL	Decimal(5)
DATEINV_YYDDD	Five Digit YYDDD Date	YYDDD	Decimal(5)
DATEINV_YYMMDD	Six Digit YYMMDD Date	YYMMDD	Decimal(6)
DATEINV_YYDDMM	Six Digit YYDDMM Date	YYDDMM	Decimal(6)
DATEINV_DDMMYY	Six Digit DDMMYY Date	DDMMYY	Decimal(6)

Table 27. DATETYPES table (continued)

Column	Description	Heading	Data Type
DATEINV_MMDDYY	Six Digit MMDDYY Date	MMDDYY	Decimal(6)
DATEINV_CYYDDD	Six Digit CYYDDD Date	CYYDDD	Decimal(6)
DATEINV_CYYMMDD	Seven Digit CYYMMDD Date	CYYMMDD	Decimal(7)
DATEINV_CYYDDMM	Seven Digit CYYDDMM Date	CYYDDMM	Decimal(7)
DATEINV_YYYYDDD	Seven Digit YYYYDDD Date	YYYYDDD	Decimal(7)
DATEINV_YYYYMMDD	Eight Digit YYYYMMDD Date	YYYYMMDD	Decimal(8)
DATEINV_YYYYDDMM	Eight Digit YYYYDDMM Date	YYYYDDMM	Decimal(8)
DATEINV_MMDDYYYY	Eight Digit MMDDYYYY Date	MMDDYYYY	Decimal(8)
DATEINV_DDMMYYYY	Eight Digit DDMMYYYY Date	DDMMYYYY	Decimal(8)
DATEINV_CHAR	Eight Character YYYYMMDD Date	YYYYMMDD	Char(8)
DATEINV_HYF	Five Digit Infinium Date (EXCELSRL – 1)	HYF	Decimal(5)

TIMEDIM Table

The TIMEDIM table contains time-related information for specific dates. This table enables you to perform a variety of time-related analyses.

This table contains 2,192 rows of data.

Table 28. TIMEDIM table

Column	Description	Heading	Data Type	Contents
TDATE	TDATE	TDATE	Date	Date
TYEAR	TYEAR	TYEAR	Integer	Year
TMONTH	TMONTH	TMONTH	Integer	Month number
TDAY	TDAY	TDAY	Integer	Day of month
DAYOFWEEK	DAYOFWEEK	DAYOFWEEK	Integer	Day of week number
DAYNAME	DAYNAME	DAYNAME	Varchar(9)	Day
DAYOFYEAR	DAYOFYEAR	DAYOFYEAR	Integer	Day of year number
TWEEK	TWEEK	TWEEK	Integer	Week number
HOLCODE	HOLCODE	HOLCODE	Integer	Holiday code
HOLIDAY	HOLIDAY	HOLIDAY	Varchar(30)	Holiday name

SALESPERF Table

The SALESPERF table tracks a number of performance measurements. It includes sales projections and actual sales information. It also includes sales quota information for each sales representative and allows you to determine actual and projected sales, as well as quotas, on a geographical basis.

This table contains 12,008 rows of data.

Table 29. SALESPERF table

Column	Description	Heading	Data Type	Contents
PERIOD	Period	Period	Date	Time period
REPCODE	Sales Rep Code	Rep Code	Char(4)	Account representative's code
AREA_ID	Area ID	Area ID	Char(8)	Area code
PRODUCTID	Product ID (SKU)	Product ID	Numeric(8)	Product ID
PRJ_UNITS	Projected Unit Sales	Projected Units	Numeric(15,0)	Projected unit sales
ACT_UNITS	Actual Unit Sales	Actual Units	Numeric(15,0)	Actual units sold
PRJ_SALES	Projected Sales	Projected Sales	Numeric(15,4)	Projected sales amount
ACT_SALES	Actual Sales	Actual Sales	Numeric(15,4)	Actual sales amount
QUOTA	Quota	Quota	Numeric(15,4)	Sales quota

EMPLOYEES Table

The EMPLOYEES table contains human resources information about The Outdoor Connection's employees. Specifically, it lists an employee's home address and phone number, e-mail address, gender, date of birth, and, where applicable, social security number. It also contains the employee's compensation type and rate, hire date, and, where applicable, termination date and reason.

This table contains 39 rows of data.

Table 30. EMPLOYEES table

Column	Description	Heading	Data Type	Contents
EMP_ID	Employee ID	Employee ID	Numeric(8)	Employee number
LASTNAME	Surname	Last Name	Varchar(100)	Surname of employee
FIRSTNAME	Given Name	First Name	Varchar(100)	Given name of employee
ADDRESS1	Address 1	Address 1	Char(100)	Address line 1
ADDRESS2	Address 2	Address 2	Char(100)	Address line 2
CITY	City	City	Char(50)	City name
STATE	State/Province Abbreviation	State/Province Abbr	Char(2)	State/province postal abbreviation
POSTALCODE	Postal Code	Postal Code	Char(16)	Postal code

Table 30. EMPLOYEES table (continued)

Column	Description	Heading	Data Type	Contents
COUNTRY	Country	Country	Char(50)	Country name
PHONE	Phone Number	Phone Number	Char(24)	Phone number
E_ADDRESS	E-mail Address	E-mail Address	Char(100)	E-mail address of client contact
SEX	Gender	Sex	Char(1)	Gender
DOB	Date of Birth	Date of Birth	Date	Date of birth
SSN	Social Security Number	Social Security No.	Char(11)	Social security number
COMPTYPE	Compensation Type	Comp Type	Char(1)	Compensation type (commission, salary, hourly)
SALARY	Base Monthy Salary	Base Salary	Numeric(10,2)	Base monthly salary
HOURLY	Hourly Rate	Hourly Rate	Numeric(4,2)	Hourly wage
COMMISSION	Commission Rate	Commission	Numeric(4,4)	Commission percentage
CURR_CODE	Currency Code	Currency Code	Char(3)	Currency code
HIREDATE	Hire Date	Hire Date	Date	Hire date
TERMDATE	Termination Date	Termination Date	Date	Termination date
TERMTYPE	Termination Reason	Termination Reason	Char(20)	Reason for termination

SCDB45 Database

The SCDB45 database represents an OLTP-type database. Its tables are mostly normalized, which means that the data are stored in only one place. This requires that tables be joined to find detailed information that may be stored as codes in other tables. For example, the ORDERMAST (Order Master) table contains an order number and a customer ID for each order. To find the details for the order, the ORDERMAST table must be joined to the ORDERDET (Order Detail) table. Likewise, to find information about the customer who placed the order, the ORDERMAST table must be joined to the CUSTMAST (Customer Master) table.

ORDERMAST Table

The ORDERMAST (Order Master) table is the basis of all transaction information. It contains the “header” information for each order placed, including the order number, the location placing the order, the date of the order, the promised delivery date, the date shipped, etc. The table also includes a column that records the location to which each order is shipped. This accommodates the situations in which the order is shipped to a location other than where the order originates.

This table contains 12,411 rows of data.

Table 31. ORDERMAST table

Column	Description	Heading	Data Type	Contents
ORDNUM	Order Number	Order Number	Decimal(8)	Order number
CUST_ID	Customer ID	Customer ID	Numeric(8)	Account ID of client
SHIP_TO	Shipping Destination	Ship To	Numeric(8)	Account ID of shipping destination
DATEREQ	Date Requested	Date Requested	Date	Date requested for delivery
DATEORD	Date Ordered	Date Ordered	Date	Date ordered
DATEPROM	Date Promised	Date Promised	Date	Date promised for delivery
DATESHIP	Date Shipped	Date Shipped	Date	Date shipped
DATEINV	Date Invoiced	Date Invoiced	Date	Date invoiced
DIST_ID	Distribution Center ID	Distribution Center ID	Char(8)	Distribution center code
REPCODE	Sales Rep Code	Rep Code	Char(4)	Account representative's code

ORDERDET Table

The ORDERDET (Order Detail) table contains the product-related specifics of each order, such as the quantity ordered, quantity shipped, quantity backlogged, and quantity canceled for each product. In addition, the table tracks the unit price, unit cost, extended price, and extended cost for each product ordered. Finally, for international sites, the table includes exchange rates. The rates are used to calculate the foreign extended price and foreign extended cost for each product based on the exchange rate at the time of the order.

This table contains 14,379 rows of data.

Table 32. ORDERDET table

Column	Description	Heading	Data Type	Contents
ORDNUM	Order Number	Order Number	Decimal(8)	Order number
PRODUCTID	Product ID (SKU)	Product ID	Numeric(8)	Product ID
PRDLVL03	Product Level 3	Product Level 3	Char(8)	Third-tier product category
QTYORD	Quantity Ordered	Quantity Ordered	Decimal(15)	Quantity ordered
QTYSHP	Quantity Shipped	Quantity Shipped	Decimal(15)	Quantity shipped

Table 32. ORDERDET table (continued)

Column	Description	Heading	Data Type	Contents
QTYBACK	Quantity Backordered	Quantity Backordered	Decimal(15)	Quantity back-ordered
QTYCAN	Quantity Cancelled	Quantity Cancelled	Decimal(15)	Quantity canceled
UNIT_PRICE	Unit Price	Unit Price	Decimal(15,4)	Unit retail price
UNIT_COST	Unit Cost	Unit Cost	Decimal(15,4)	Unit wholesale cost (cost of goods sold)
EXTPRICE	Extended Price	Extended Price	Decimal(15,4)	Extended retail total (the product of QTYORD and UNIT_PRICE)
EXTCOST	Extended Cost	Extended Cost	Decimal(15,4)	Extended wholesale total (the product of QTYORD and UNIT_COST)
MEASURE	Unit of Measure	Unit of Measure	Char(4)	Unit of measure code
CURR_CODE	Currency Code	Currency Code	Char(3)	Currency code
EXCHG_RATE	Exchange Rate	Exchange Rate	Decimal(15,7)	Exchange rate
FRGN_XPRIC	Foreign Extended Price	Foreign Ext Price	Decimal(15,4)	Foreign extended retail total
FRGN_XCOST	Foreign Extended Cost	Foreign Ext Cost	Decimal(15,4)	Foreign extended wholesale total

CUSTMAST Table

The CUSTMAST (Customer Master) table contains information about The Outdoor Connection’s clients. It includes names and addresses and divides the clients into various geographical categories. The Outdoor Connection consists of 3 divisions (North America, Europe, and Asia/Pacific); 12 regions (East, West, South, Midwest, Canada, Belgium, Germany, France, United Kingdom, Japan, Australia, and New Zealand); 70 areas (one for each state and province, including the District of Columbia, plus one for each European and Asia/Pacific country); and 3 distribution facilities (Seattle, Kansas City, and New York). In addition, this table includes a currency code for each client, a status code indicating whether a client is active or inactive, and a code that identifies each client’s sales representative.

This table contains 157 rows of data.

Table 33. CUSTMAST table

Column	Description	Heading	Data Type	Contents
CUST_ID	Customer ID	Customer ID	Numeric(8)	Account ID of client

Table 33. CUSTMAST table (continued)

Column	Description	Heading	Data Type	Contents
CUSTNAME	Customer Name	Customer Name	Char(100)	Name of client
LNAME	Surname	Last Name	Char(50)	Surname of client contact
FNAME	Given Name	First Name	Char(50)	Given name of client contact
ADDRESS1	Address 1	Address 1	Char(100)	Address line 1
ADDRESS2	Address 2	Address 2	Char(100)	Address line 2
CITY	City	City	Char(50)	City name
STATE	State/Province Abbreviation	State/Province Abbreviation	Char(2)	State/province postal abbreviation
POSTALCODE	Postal Code	Postal Code	Char(16)	Postal code
COUNTY	County	County	Varchar(80)	County name
COUNTRY	Country Code	Country Code	Varchar(12)	Country code
PHONE	Phone Number	Phone Number	Char(24)	Phone number
E_ADDRESS	E-mail Address	E-mail Address	Char(100)	E-mail address of client contact
LOB	LOB	LOB Code	Char(8)	Line of business code
REPCODE	Sales Rep Code	Rep Code	Char(4)	Account representative's code
ENTRY_DATE	Entry Date	Entry Date	Date	Date added
ACTIVE	Active Code	Active	Char(1)	Active status (Y or N)
DIVSN_ID	Division ID	Division ID	Char(8)	Division code
REGION_ID	Region ID	Region ID	Char(8)	Region code
AREA_ID	Area ID	Area ID	Char(8)	Area code
DIST_ID	Distribution Center ID	Distribution Center ID	Char(8)	Distribution center code
CURR_CODE	Currency Code	Currency Code	Char(3)	Currency code

LOB Table

The LOB (Line of Business) table maps the line of business code to a description. A line of business can be considered a subclassification for the client. The database contains three lines of business: retail, specialty, and catalog.

This table contains 3 rows of data.

Table 34. LOB table

Column	Description	Heading	Data Type	Contents
LOB	LOB	LOB Code	Char(8)	Line of business code
LOBDESC	Line of Business Description	Line of Business	Char(80)	Line of business description

CURRENCY Table

The CURRENCY table contains exchange rates used to convert U.S. dollars into foreign currency amounts. The exchange rate is an unsigned percentage and can be mapped to a currency code for each country in which The Outdoor Connection conducts business. It is assumed that all monetary amounts entered into the database are entered in U.S. dollars. Therefore, by definition, the exchange rate for U.S. dollars is 1.0. The exchange rate for other currencies is based on the amount that one U.S. dollar will buy. For example, $1 U.S. recently bought $1.4575 CA. Therefore, the exchange rate for Canadian dollars is 1.4575. To convert an amount in the database into local values, multiply the amount by the exchange rate:

$10 US x 1.4575 = $14.58 CA or $10 US x 0.6105 = £6.11

This table contains 9 rows of data.

Table 35. CURRENCY table

Column	Description	Heading	Data Type	Contents
CURR_CODE	Currency Code	Currency Code	Char(3)	Currency code
COUNTRY	Country Code	Country Code	Varchar(12)	Country code
CURR_DESC	Currency Description	Currency	Char(50)	Currency description
EXCHG_RATE	Exchange Rate	Exchange Rate	Decimal(15,7)	Exchange rate
EXCHG_DATE	Effective Date	Exchange Date	Date	Effective date of exchange rate

CNTRYMAST Table

The CNTRYMAST (Country Master) table maps the country code to the country name.

This table contains 9 rows of data.

Table 36. CNTRYMAST table

Column	Description	Heading	Data Type	Contents
COUNTRY	Country Code	Country Code	Varchar(12)	Country code

Table 36. CNTRYMAST table (continued)

Column	Description	Heading	Data Type	Contents
COUNTRYDESC	Country Description	Country	Varchar(80)	Country name

STATEMAST Table

The STATEMAST (State Master) table maps the postal abbreviation for the U.S. state or Canadian province to the name of the state or province. The database contains 50 states, 12 provinces, and the District of Columbia.

This table contains 63 rows of data.

Table 37. STATEMAST table

Column	Description	Heading	Data Type	Contents
STATE	State/Province Abbreviation	State/Province Abbreviation	Char(2)	State/province postal abbreviation
STATEDESC	State/Province Description	State/Province	Varchar(80)	State/province name

PRODMAST Table

The PRODMAST (Product Master) table contains all of The Outdoor Connection's product information and groups products hierarchically. Each product has three levels associated with it. The first identifies a product group (for example, clothing, biking, skiing, etc.), the second, a type (for example, outerwear, shirts, pants, etc.), and the third, a style (for example, black, red, yellow, etc.).

In addition, the PRODMAST table includes a unit price and a unit cost for each product. It also contains a currency code so the unit price and cost can be converted for various countries. Finally, it includes a measures column with a unit type (for example, *one* bicycle, *a pair* of skis, etc.) and a date that records when information was last changed.

This table contains 120 rows of data.

Table 38. PRODMAST table

Column	Description	Heading	Data Type	Contents
PRODUCTID	Product ID (SKU)	Product ID	Numeric(8)	Product ID
PRODUCT	Product Description	Product Description	Char(80)	Product description
PRDLVL01	Product Level 1	Product Level 1	Char(8)	First tier product category

Table 38. PRODMAST table (continued)

Column	Description	Heading	Data Type	Contents
PRDLVL02	Product Level 2	Product Level 2	Char(8)	Second tier product category
PRDLV03	Product Level 3	Product Level 3	Char(8)	Third tier product category
PRDLVL04	Product Level 4	Product Level 4	Varchar(8)	Fourth tier product category
PRDLV05	Product Level 5	Product Level 5	Varchar(8)	Fifth tier product category
UNITPRICE	Unit Price	Unit Price	Decimal(15,4)	Unit retail price
UNITCOST	Unit Cost	Unit Cost	Decimal(15,4)	Unit wholesale cost (cost of goods sold)
CURR_CODE	Currency Code	Currency Code	Char(3)	Currency code
MEASURE	UOM Code	UOM Code	Char(4)	Unit of measure code
ENTRY_DATE	Entry Date	Entry Date	Date	Entry/update date

PLVL01MAST Table

The PLVL01MAST (Product Level 1 Master) table maps the first product level code to the product level description. This level identifies four product groups: clothing, bicycles, skiing, and camping.

This table contains 4 rows of data.

Table 39. PLVLO1MAST table

Column	Description	Heading	Data Type	Contents
PRDLVL01	Product Level 1	Product Level 1	Char(8)	First tier product category
PRDDESC01	Product Description 1	Description Level 1	Char(80)	First tier product description

PLVL02MAST Table

The PLVL02MAST (Product Level 2 Master) table maps the second product level code to the product level description. This level identifies four product types for each of the product groups. For example, the clothing product group contains the product types outerwear, pants, shirts, and sweaters.

This table contains 16 rows of data.

Table 40. PLVLO2MAST table

Column	Description	Heading	Data Type	Contents
PRDLVL02	Product Level 2	Product Level 2	Char(8)	Second tier product category
PRDDESC02	Product Description 2	Description Level 2	Char(80)	Second tier product description

PLVL03MAST Table

The PLVL03MAST (Product Level 3 Master) table maps the third product level code to the product level description. This level identifies five product styles: black, blue, green, red, and yellow. The product style can be considered a subclassification for a product.

This table contains 5 rows of data.

Table 41. PLVLO3MAST table

Column	Description	Heading	Data Type	Contents
PRDLV03	Product Level 3	Product Level 3	Char(8)	Third tier product category
PRDDESC03	Product Description 3	Description Level 3	Char(80)	Third tier product description

UOMMAST Table

The UOMMAST (Unit of Measure Master) table maps the unit of measure code to the measure description. The unit of measure identifies whether a product is sold as a single unit or in pairs.

This table contains 2 rows of data.

Table 42. UOMMAST table

Column	Description	Heading	Data Type	Contents
MEASURE	UOM Code	UOM Code	Char(4)	Unit of measure code
MEASURE_DESC	Unit of Measure	Unit of Measure	Char(80)	Unit of measure description

ORGMAST Table

The ORGMAST (Organization Master) table contains the geographic structure used in the database. Essentially, it maps the various divisions, regions, and areas to each other. Each division has several regions, and each region has several areas. This table also indicates which distribution facility serves each area or foreign region. The database contains 3 divisions, 12 regions, 70 areas, and 3 distribution facilities.

This table contains 70 rows of data.

Table 43. ORGMAST table

Column	Description	Heading	Data Type	Contents
DIVSN_ID	Division ID	Division ID	Char(8)	Division code
REGION_ID	Region ID	Region ID	Char(8)	Region code
AREA_ID	Area ID	Area ID	Char(8)	Area code
DIST_ID	Distribution Center ID	Distribution Center ID	Char(8)	Distribution center code

DIVSNMAST Table

The DIVSNMAST (Division Master) table maps each division code to a division name. The database contains three divisions: North America, Europe, and Asia/Pacific.

This table contains 3 rows of data.

Table 44. DIVSNMAST table

Column	Description	Heading	Data Type	Contents
DIVSN_ID	Division ID	Division ID	Char(8)	Division code
DIVSN	Division	Division	Char(80)	Division name

REGIONMAST Table

The REGIONMAST (Region Master) table maps each region code to a region name. The database contains three North American regions, four European regions, and three Asia/Pacific regions.

This table contains 12 rows of data.

Table 45. REGIONMAST table

Column	Description	Heading	Data Type	Contents
REGION_ID	Region ID	Region ID	Char(8)	Region code
REGION	Region	Region	Char(80)	Region name

AREAMAST Table

The AREAMAST (Area Master) table maps each area code to an area name. The database contains 63 areas in the North American region, 4 areas in the European region, and 3 areas in the Asia/Pacific region.

This table contains 70 rows of data.

Table 46. AREAMAST table

Column	Description	Heading	Data Type	Contents
AREA_ID	Area ID	Area ID	Char(8)	Area code
AREA	Area	Area	Char(80)	Area name

DISTMAST Table

The DISTMAST (Distribution Center Master) table maps each distribution center code to a distribution center name. The database contains three distribution centers: Seattle, Kansas City, and New York. Each distribution center can serve all divisions.

This table contains 3 rows of data.

Table 47. DISTMAST table

Column	Description	Heading	Data Type	Contents
DIST_ID	Distribution Center ID	Distribution Center ID	Char(8)	Distribution center code
DIST_LOC	Distribution Center	Distribution Center	Char(80)	Distribution center name

DATETYPES Table

The DATETYPES table is used to demonstrate how product applications handle date values stored as a data type other than `DATE`. Specifically, the `DATEINV` column has been replicated into several formats. You can use this table independently or join it to the ORDERMAST table on `ORDNUM`.

This table contains 12,411 rows of data.

Table 48. DATETYPES table

Column	Description	Heading	Data Type
ORDNUM	Order Number	Order Number	Decimal(8)
DATEINV	Date Invoiced	Date Invoiced	Date
DATEINV_YY	Two Digit Year Value	YY	Decimal(2)
DATEINV_MM	Two Digit Month Value	MM	Decimal(2)
DATEINV_DD	Two Digit Day Value	DD	Decimal(2)
DATEINV_EXCELSRL	Five Digit Excel Serial Date	EXCELSRL	Decimal(5)
DATEINV_YYDDD	Five Digit YYDDD Date	YYDDD	Decimal(5)
DATEINV_YYMMDD	Six Digit YYMMDD Date	YYMMDD	Decimal(6)

Table 48. DATETYPES table (continued)

Column	Description	Heading	Data Type
DATEINV_YYDDMM	Six Digit YYDDMM Date	YYDDMM	Decimal(6)
DATEINV_DDMMYY	Six Digit DDMMYY Date	DDMMYY	Decimal(6)
DATEINV_MMDDYY	Six Digit MMDDYY Date	MMDDYY	Decimal(6)
DATEINV_CYYDDD	Six Digit CYYDDD Date	CYYDDD	Decimal(6)
DATEINV_CYYMMDD	Seven Digit CYYMMDD Date	CYYMMDD	Decimal(7)
DATEINV_CYYDDMM	Seven Digit CYYDDMM Date	CYYDDMM	Decimal(7)
DATEINV_YYYYDDD	Seven Digit YYYYDDD Date	YYYYDDD	Decimal(7)
DATEINV_YYYYMMDD	Eight Digit YYYYMMDD Date	YYYYMMDD	Decimal(8)
DATEINV_YYYYDDMM	Eight Digit YYYYDDMM Date	YYYYDDMM	Decimal(8)
DATEINV_MMDDYYYY	Eight Digit MMDDYYYY Date	MMDDYYYY	Decimal(8)
DATEINV_DDMMYYYY	Eight Digit DDMMYYYY Date	DDMMYYYY	Decimal(8)
DATEINV_CHAR	Eight Character YYYYMMDD Date	YYYYMMDD	Char(8)
DATEINV_HYF	Five Digit Infinium Date (EXCELSRL – 1)	HYF	Decimal(5)

SALESPERF Table

The SALESPERF table tracks a number of performance measurements. It includes sales projections and actual sales information. It also includes sales quota information for each sales representative and allows you to determine actual and projected sales, as well as quotas, on a geographical basis.

This table contains 12,008 rows of data.

Table 49. SALESPERF table

Column	Description	Heading	Data Type	Contents
PERIOD	Period	Period	Date	Time period
REPCODE	Sales Rep Code	Rep Code	Char(4)	Account representative’s code
AREA_ID	Area ID	Area ID	Char(8)	Area code
PRODUCTID	Product ID (SKU)	Product ID	Numeric(8)	Product ID
PRJ_UNITS	Projected Unit Sales	Projected Units	Numeric(15,0)	Projected unit sales
ACT_UNITS	Actual Unit Sales	Actual Units	Numeric(15,0)	Actual units sold
PRJ_SALES	Projected Sales	Projected Sales	Numeric(15,4)	Projected sales amount
ACT_SALES	Actual Sales	Actual Sales	Numeric(15,4)	Actual sales amount
QUOTA	Quota	Quota	Numeric(15,4)	Sales quota

EMPMAST Table

The EMPMAST (Employee Master) table contains human resources information about The Outdoor Connection's employees. Specifically, it lists an employee's home address and phone number, e-mail address, gender, date of birth, and, where applicable, social security number. It also contains the employee's compensation type and rate, hire date, and, where applicable, termination date and reason. If an employee is part of the sales organization, the table contains the employee's sales representative code, the representative code of the employee's manager, the date the employee entered the sales organization, and a status code indicating whether the employee is currently active or inactive in the sales organization.

This table contains 39 rows of data.

Table 50. EMPMAST table

Column	Description	Heading	Data Type	Contents
EMP_ID	Employee ID	Employee ID	Numeric(8)	Employee number
LASTNAME	Surname	Last Name	Varchar(100)	Surname of employee
FIRSTNAME	Given Name	First Name	Varchar(100)	Given name of employee
ADDRESS1	Address 1	Address 1	Char(100)	Address line 1
ADDRESS2	Address 2	Address 2	Char(100)	Address line 2
CITY	City	City	Char(50)	City name
STATE	State/Province Abbreviation	State/Province Abbreviation	Char(2)	State/province postal abbreviation
POSTALCODE	Postal Code	Postal Code	Char(16)	Postal code
COUNTRY	Country	Country	Varchar(12)	Country code
PHONE	Phone Number	Phone Number	Char(24)	Phone number
E_ADDRESS	E-mail Address	E-mail Address	Char(100)	E-mail address of employee
SEX	Gender	Sex	Char(1)	Gender
DOB	Date of Birth	Date of Birth	Date	Date of birth
SSN	Social Security Number	Social Security No.	Char(11)	Social security number
COMPTYPE	Compensation Type	Comp Type	Char(1)	Compensation type (commission, salary, hourly)
SALARY	Base Monthly Salary	Base Salary	Numeric(10,2)	Base monthly salary
HOURLY	Hourly Rate	Hourly Rate	Numeric(4,2)	Hourly wage

Table 50. EMPMAST table (continued)

Column	Description	Heading	Data Type	Contents
COMMISSION	Commission Rate	Commission	Numeric(4,4)	Commission percentage
CURR_CODE	Currency Code	Currency Code	Char(3)	Currency code
HIREDATE	Hire Date	Hire Date	Date	Hire date
TERMDATE	Termination Date	Termination Date	Date	Termination date
TERMTYPE	Termination Reason	Termination Reason	Char(20)	Reason for termination
REPCODE	Sales Rep Code	Rep Code	Char(4)	Sales rep code
MGRCODE	Manager Code	Manager Code	Char(4)	Sales manager code
ENTRY_DATE	Entry Date as Sales Rep	Entry Date	Date	Entry date into sales organization
ACTIVE	Active Sales Rep	Active	Char(1)	Active status of sales rep

Installing the Sample Library after Installation with RSTSMPLB

RSTSMPLB (Restore Sample Library) will install the sample databases if they were not previously installed. This command can also restore the sample database back to its original structure (if database files were altered or destroyed). To use this command, type RSTSMPLB at the command prompt and press Enter.

The command has two parameters:

- CLRLIB (*YES, *NO). If *YES is chosen, the sample database library (if it exists) is cleared (all files within destroyed) before the sample files are restored. Sample files are restored to their original version installed with this release.
- PRVLGL (*YES,*NO). If *YES is chosen, all logical files on the system that are dependent on the sample files within the sample database library are preserved. Logical files that are dependent on non-product files within the sample library are not preserved.

Note: If the user wants to preserve logical files not dependent on sample files, choose *NO for CLRLIB.

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 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
Language Environment
Net.Data
Notes
iSeries
OS/390
RS/6000

DRDA
IBM
OpenEdition
S/390
SecureWay
OS/400

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.

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

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.

Index

Program Number: 5724–B78

Printed in U.S.A.

SC18-7227-00

Spine information:

IBM® DB2™ ® OLAP Server™ for iSeries

DB2 OLAP Server Administrator's Guide for Builder on iSeries

Version 8.1