Oracle8*i* Cache

Concepts and Administration Guide

Release 1.0

May 2000

Part No. A83544-01

ORACLE®

Oracle8*i* Cache Concepts and Administration Guide, Release 1.0

Part No. A83544-01

Contents

3 Managing Oracle8*i* Cache

4 Tuning Your Oracle8*i* Cache Environment

5 Troubleshooting Oracle8*i* Cache

A Error Messages

Index

Send Us Your Comments

Oracle8*i* Cache Concepts and Administration Guide, Release 1.0

Part No. A83544-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: nedc_doc@us.oracle.com
- FAX: 603.897.3819   Attn: Oracle8*i* Cache Documentation
- Postal service
  Oracle Corporation
  Oracle8*i* Cache Documentation
  One Oracle Drive
  Nashua, NH 03062-2698
  USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

Oracle8*i* Cache improves scalability and performance for applications that access Oracle databases in a multiple-tier environment. By caching frequently accessed data on the middle tier, Oracle8*i* Cache reduces the load on the back-end database and speeds query execution. Oracle8*i* Cache is ideal for enhancing scalability and performance for Web servers and Web portals, as well as front-office and back-office application servers.

This manual presents the concepts of Oracle8*i* Cache and describes how to set up and manage your Oracle8*i* Cache environment.

Intended Audience

This manual is intended for Web server and database administrators who are interested in using Oracle8*i* Cache to increase the capacity of their applications.

Structure

This manual contains the following chapters:

Chapter 1	Introduces Oracle8*i* Cache and provides an overview of the concepts of Oracle8*i* Cache.
Chapter 2	Describes how to set up your Oracle8*i* Cache environment.
Chapter 3	Describes how to manage your Oracle8*i* Cache environment.
Chapter 4	Describes how to tune your Oracle8*i* Cache environment to improve scalability and performance. For applications that use OCI static libraries, it describes how to relink your applications.

Related Documents

For more information about Oracle8*i* Cache, see the following:

- *Oracle8i Cache Readme,* which contains important information about Oracle8*i* Cache, including known problems and restrictions. It is available on the software CD-ROM and is installed with the software.

- The installation guide, particularly the installation instructions for Oracle8*i* Cache.

- Oracle8*i* Cache Quick Tour, which is available from Cache Manager, a component of Oracle DBA Studio.

- The online help for Cache Manager.

In addition, the following books from the Oracle8*i* documentation set contain related information:

- *Oracle Call Interface Programmer's Guide*

- *Oracle8i Administrator's Reference* for your platform

Conventions

The following conventions are used in this manual:

Convention	Meaning
. . .	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
. . .	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted.
boldface text	Boldface type in text indicates a term defined in the text.
italic text	Italic text indicates emphasis, a user-supplied name, or the title of a document.

Convention	Meaning
[ ]	Brackets enclose optional clauses from which you can choose one or none.

1

Introducing Oracle8*i* Cache

Oracle8*i* Cache improves the scalability and performance of applications that access Oracle databases by caching frequently used data on a middle-tier system. With Oracle8*i* Cache, your applications can process several times as many requests as their original capacity. In addition, you do not need to modify your existing applications to use Oracle8*i* Cache. Oracle8*i* Cache is transparent to your end users.

Who Should Use Oracle8*i* Cache?

If your applications meet the following criteria, you can use Oracle8*i* Cache to boost the scalability of your Web sites and the performance of your applications:

- Your applications access Oracle databases.

- You use a multiple-tier environment, where the clients, Web or application servers, and Oracle database servers are located on separate systems.

- Your applications communicate with an Oracle database through Oracle Call Interface (OCI), or an access layer built on OCI, like JDBC-OCI, ODBC, or OLE. The applications can be written using scripting or programming languages.

- Web or application users generate mostly read-only queries.

- Data residing in a subset of tables is more frequently accessed than other data in the database.

- Your applications can tolerate some synchronization delay with data in the origin database. That is, the data in the cache does not need to be as up-to-date as the data in the origin database. (You decide how often to refresh the data.)

Many applications meet these characteristics, including e-business and content publishing applications.

What Are the Benefits of Oracle8*i* Cache?

Oracle8*i* Cache maintains a cache of frequently requested data on the middle tier (the Web server or application server tier), and it processes read-only requests using this dynamic cache.

Oracle8*i* Cache provides performance and scalability benefits by:

- Processing database queries on the middle tier, thereby reducing the time spent sending and receiving the data over the network. As a result, Web and application servers can process more queries faster using the same amount of hardware resources.

- Reducing the load on the database server tier by processing the most common read-only requests on the Web or application server tier. As a result, existing databases can support more users.

There is no need to modify your existing applications or your existing databases to achieve these benefits.

What Is the Oracle8*i* Cache Environment?

In a traditional multiple-tier environment running Web applications, the client (typically a Web browser) displays HTML or XML sent by an application running on the middle tier (usually a Web or application server). The middle tier gathers the information from the database server and translates the information into an HTML or XML page that can be displayed by the client. The database server tier stores, searches for, and retrieves data. The following figure shows an example of this Web-based, traditional, multiple-tier environment:
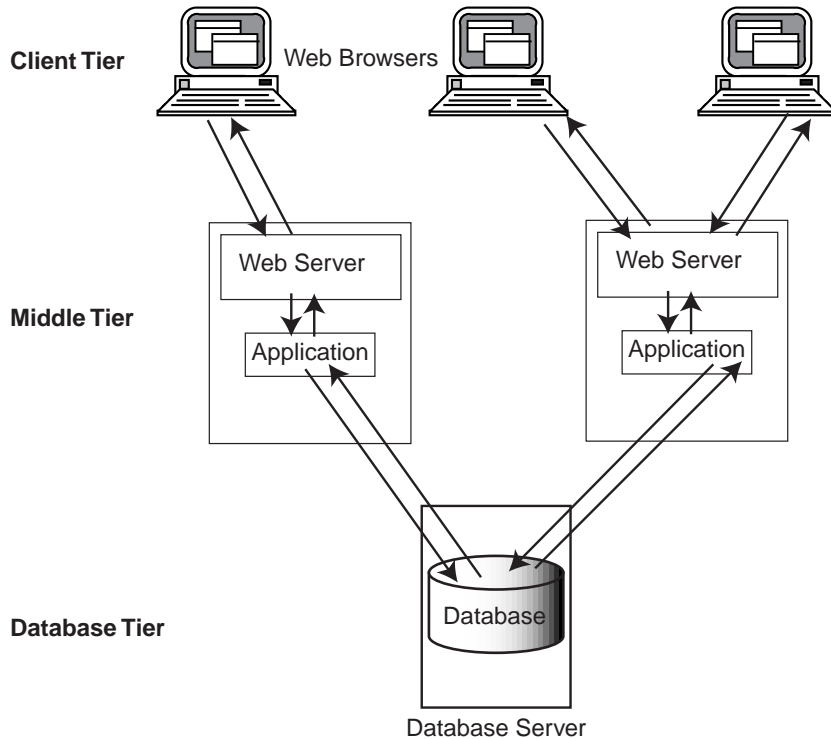
Database Server

In this environment, each user executes a database request using a distinct path from the Web browser through the Web server to the database, even if more than one user is requesting the same information. For example, assume that your database contains information about books, such as the book title, author, description, and ranking in a best-seller list. If User A and User B both request the list of the top 10 mystery books, each user's request is routed through the application or Web server to the database on the database server. For each user, the database server returns the data through the application or Web server, which sends it to the Web browser for display. Each user receives a separate copy of the data. Later, if User A makes the same request, it is routed once again through the application or Web server to the database on the database server.

Other types of applications, such as content publishing, running in a multiple-tier environment, function in a similar way.

Using Oracle8*i* Cache, you can store frequently requested data on the Web or application server tier. The requests of User A and User B from the client tier are

routed to the cache on the middle tier, which returns the data. The following figure shows the multiple-tier environment using Oracle8*i* Cache:
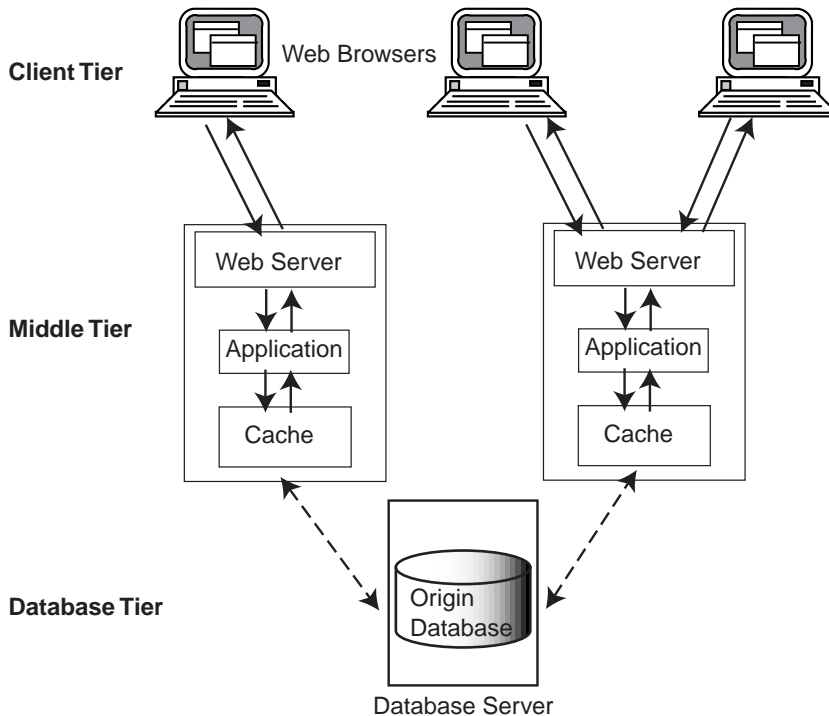
In this environment, Oracle8*i* Cache software, consisting of a **middle-tier cache** for storing data and intelligent software to route queries, resides on each Web or application server on the middle tier. Frequently accessed data is stored in the middle-tier cache. The **origin database** (on the database tier) is the original and primary storage for the data. For this release, a middle-tier cache can cache data from only one origin database.

When users request the frequently accessed data, the requests pass from the client, such as a Web browser, to the Web or application server, which returns the data. For example, assume you cache information about books, such as the book title, author, description, and ranking in a best-seller list, in the middle-tier cache. If User A and User B both request the list of the top 10 mystery books, each user's request is routed to the middle-tier cache. Because the data is stored on the same tier, the application returns the data quickly and does not need to pass the request to the origin database server for retrieval of the data. Furthermore, each user can click on

one of the titles to get more detailed information about the book. The query is routed to the middle-tier cache, which returns the data quickly.

How Does Oracle8*i* Cache Work?

Oracle8*i* Cache provides a middle-tier cache for storing data and an easy-to-use management interface for managing the Oracle8*i* Cache environment. Oracle8*i* Cache caches and manages data in entities called data sets. A **data set** is any collection of data items that can be expressed using a SQL SELECT statement. A data set can be a table or any subset of a table, or data from more than one table.

> **Note:** For this release of Oracle8*i* Cache, you must cache entire tables; you cannot cache just the parts of tables that satisfy the query. As a result, a data set must be an entire database table.

Oracle8*i* Cache also provides components that transparently route queries to the middle-tier cache, enforce data caching policies, and store statistics about cache hits and misses. A **hit** is a query that was satisfied in the middle-tier cache. A **miss** is a query that could not be satisfied in the cache and that was routed to the origin database. Oracle8*i* Cache provides transparent connectivity to the origin database.

You use a Java-based interface, Cache Manager (a component of DBA Studio), to set up and manage your Oracle8*i* Cache environment. When you set up a middle-tier cache, you specify the origin database, the data sets (or tables) to be cached, and cache properties, such as memory allocation, disk space, and the synchronization policy. The **synchronization policy** establishes how and how often the data in the cache is refreshed from the origin database.

The following figure illustrates the Oracle8*i* Cache environment in more detail:

```
         Client Tier
                       Cache Manager          Web Browsers

                                                    Web Server

                                                     Application

         Middle Tier
                                            Cache Router
                                            Data in Cache?   No
                                                        Yes

                                              Cache

         Database Tier
                                                              Origin
                                                              Database

                                                          Database Server
```
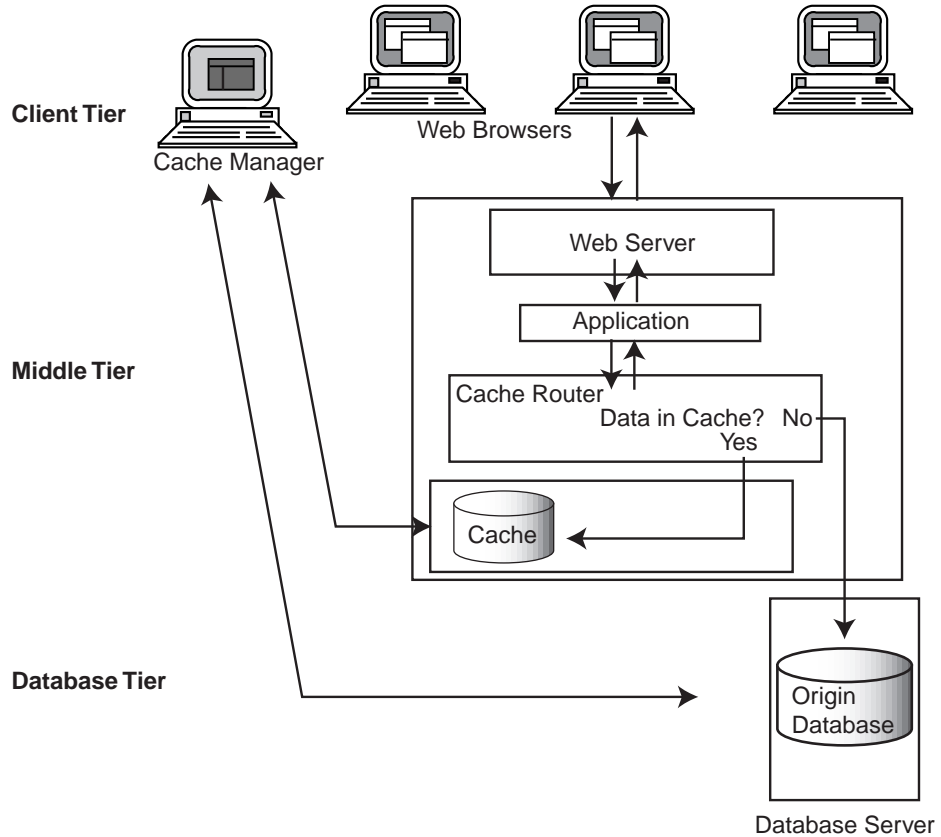
Oracle8*i* Cache maintains a history of queries and, based on that history, intelligently routes queries either to the middle-tier cache or to the origin database. It also maintains statistics about the queries. These statistics are accessed by all clients attached to a particular middle-tier cache.

When a user requests data, Oracle8*i* Cache checks to see if the query has been executed previously. If it has and if the history indicates that the data is not found in the middle-tier cache, Oracle8*i* Cache immediately routes the request to the origin database, which returns the data, through the Web or application server, to the user.

If the query has not been executed previously, Oracle8*i* Cache attempts to execute the query using the middle-tier cache. If the query can be satisfied on the middle-tier cache, Oracle8*i* Cache returns the data from the middle-tier cache to the user. (Note that a query may return no data, as when a user requests books by an

author whose information is not stored in the database. In this case, the query is regarded as being satisfied by the cache.) If the query cannot be satisfied by the cache, Oracle8*i* Cache routes the request to the origin database, which returns the data, through the Web or application server, to the user. In either case, Oracle8*i* Cache records information about the query.

If the request is an insert, update, or delete, rather than a query, Oracle8*i* Cache immediately routes the request to the origin database. If the request is a PL/SQL statement or anonymous block, Oracle8*i* Cache routes the statement or block to the origin database.

Cache Manager helps you monitor the ratio of hits to misses. You can use this information to assess whether you are caching the appropriate data sets, you need to cache additional data sets, or you need to drop the currently cached data sets and cache different data sets.

How Do Applications Use Oracle8*i* Cache?

You do not need to modify your applications to take advantage of the benefits of Oracle8*i* Cache. You only need to configure the environment of your applications, as described in "Configuring Your Application Environment" on page 2-25. Applications that use SQL statements to access the database are routed to the middle-tier cache automatically. Similarly, applications that are linked with OCI using *dynamic* libraries and layered directly on OCI are routed to the middle-tier cache automatically. (Applications layered directly on OCI are those applications that call OCI directly or use ODBC, JDBC, or other interfaces that call OCI directly.)

Note that applications must use OCI in threaded mode. If they do not, refer to "Applications Must Be Linked in Multithreaded Mode" on page 5-5 for information about linking in multithreaded mode.

Applications that are linked with OCI using *static* libraries and layered directly on OCI require that you relink those applications with the OCI library that ships with Oracle8*i* Cache. For more information, see "Linking Applications That Use Static OCI Libraries" on page 4-1.

2

Setting Up Oracle8*i* Cache

To set up Oracle8*i* Cache, you use Cache Manager, a component of Oracle DBA Studio. Cache Manager is an easy-to-use, Java-based interface that helps you to set up the middle-tier caches, specify the data to be cached in each, and specify the synchronization policy of each table in each cache.

Setting Up Oracle8*i* Cache: General Steps

To set up Oracle8*i* Cache, you take the following general steps:

1. Design your Oracle8*i* Cache environment, deciding which data to cache. See "Designing Your Oracle8i Cache Environment" on page 2-1.

2. Install Oracle8*i* Cache on the middle-tier nodes, such as Web or application servers, on which you want to cache data. Then, install the management interface on a client node. See "Installing Oracle8i Cache" on page 2-6.

3. Invoke Oracle DBA Studio. See "Invoking Oracle DBA Studio" on page 2-6.

4. Set up the caches on the middle-tier nodes by specifying the data to be cached in each middle-tier cache and by specifying other characteristics of the cache, such as the synchronization policies, memory allocation, and disk allocation. See "Setting Up the Caches" on page 2-13.

5. Configure your application environment to use caching. See "Configuring Your Application Environment" on page 2-25.

Designing Your Oracle8*i* Cache Environment

Before you begin setting up Oracle8*i* Cache, take time to design your Oracle8*i* Cache environment. In doing so, you gather the following information:

- What data do you want to cache? See "Determining the Data to Be Cached" on page 2-2.

- Where do you want to cache the data? See "Deciding Where to Cache the Data" on page 2-4.

- What synchronization policy do you want for each data set? The **synchronization policy** establishes how and how often the data in the cache is refreshed from the origin database. See "Deciding Synchronization Policies" on page 2-4 for information on synchronization policies and your options.

When you have answered these questions, you will have the necessary information to set up and deploy Oracle8*i* Cache.

Determining the Data to Be Cached

The first step in designing your Oracle8*i* Cache environment is determining which data you should cache. To that end, answer the following questions:

- Do your Web site users or application users generate mostly read-only requests?

  Because Oracle8*i* Cache routes only read requests to the cache, you get the greatest benefit when most of the requests for the data are read-only.

- What is the most frequently requested data? In other words, what are the most frequently executed queries?

  Your Web sites or applications may have one query that is executed very often or they may have several queries in this category. Before you set up your caches, you can use Oracle Performance Manager or SQL Analyze to help you determine which queries are executed most frequently. Alternatively, you can set up a cache and monitor the queries that are executed by viewing the Hit/Miss Statistics page of Cache Manager (see "Monitoring Hit and Miss Statistics" on page 3-3).

- In which database does this frequently requested data reside? In other words, what is the origin database?

  If the data resides in more than one database, which database contains most of the frequently accessed, read-only data? For this release, you can cache data from only one origin database.

- Which database tables contain this data?

  Each query may access data in one or more tables in a database.

- Can the requests tolerate some synchronization delay with data in the origin database?

  That is, does it matter if there is a delay between when new data is entered into the origin database and when that data in the caches is synchronized with the origin database? If a delay is acceptable, then the data is well suited to be cached.

  Read/write transactions may generate some inconsistencies. For example, if a transaction contains a query, then an UPDATE statement for the same table, and subsequently another query of the same data, the data that is returned for the second query will not contain the updated data. The updated data is written to the origin database; the query is executed by the cache. To ensure consistency of read/write transactions, you can put the statements executed during a transaction into a PL/SQL statement. Oracle8i Cache routes PL/SQL statements directly to the origin database.

- How often is this frequently requested data updated?

  If the data is updated constantly and extensively and the read-only applications must have current data, you will have to synchronize the data at frequent intervals. See "Deciding Synchronization Policies" on page 2-4 for information about synchronizing data.

Now you have identified the data to be cached. Oracle8i Cache caches and manages data in entities called data sets. A **data set** is any collection of data items that can be expressed using a SQL statement. A data set can be a table, any subset of a table, or data from more than one table.

> **Note:** For this release of Oracle8i Cache, you must cache entire tables; you cannot cache just the parts of tables that satisfy the query. As a result, a data set must be an entire database table.

As an example, after answering the preceding questions, you determine that your company Web site, GreatStore.com, has several frequently requested queries:

- One query asks for the top 100 books sold by your Web site. That data is stored in the BOOKS_ORDERED table, along with information such as book description, author, author's biography, number of pages, and ISBN number. Users can click on the book titles to get more detailed information about each book.

- Six queries request the New York Times best-seller list by category, such as hardcover fiction and paperback nonfiction. That data is stored in the NYT_ BEST table. Both tables are stored in the origin database BOOKS.

- Another query requests a page showing the specials on computer software. That data is stored in the table SOFT_SPEC in a different origin database, SOFTWARE.

In this example, you decide that you need to cache the BOOKS_ORDERED and NYT_BEST tables from the BOOKS database and the SOFT_SPEC table from the SOFTWARE database. Each table to be cached is a data set.

Deciding Where to Cache the Data

Using Oracle8*i* Cache, you cache the data in a cache on a middle-tier node.

In most cases, a company may have several middle-tier servers, such as Web servers or application servers, that handle all requests. If the data is stored in only one origin database, you can cache the same data on all middle-tier servers. That is, you can have identical caches on all nodes.

In some cases, a company may have several middle-tier servers, but it routes different types of requests to different Web servers. For example, Web servers A and B are responsible for requests for the weather information; Web servers C and D are responsible for requests for sports information. In this situation, you cache different data on Web servers A and B than on C and D.

> **Note:** For this release, data from only one origin database can be cached in any particular cache.

Using the GreatStore.com scenario described in the previous section, you must cache your data on at least two different middle-tier nodes because your data resides in two separate origin databases. You decide to cache data about books in the caches on several middle-tier nodes. You cache identical data sets in these caches. Then, you decide to cache data about computer software in the caches located on several other middle-tier nodes. You cache identical data about computer software in these caches.

Deciding Synchronization Policies

When you **synchronize** data, you make the data in the cache consistent with the data in the origin database by refreshing the cache with the current data from the

origin database. The **synchronization policy** establishes how and how often the data in the cache is refreshed from the origin database. You can specify different synchronization policies for each data set.

You can choose incremental or complete synchronization:

- Incremental: Synchronize the data by refreshing only the data that has changed since the data set was cached or since it was last synchronized.

    Use this option when only a portion of the data changes.

- Complete: Completely refresh the data set by deleting all data in the data set and replacing it with the current data from the origin database.

    Use this option when a large percentage of your data changes at the same time, as when the batch updates are loaded into the origin database.

If there is a large amount of data in the data set, a complete synchronization can take considerably more time than an incremental synchronization. However, if a large percentage of data in a data set changes, a complete synchronization is more efficient than an incremental synchronization.

For either of these types of synchronization, you can schedule synchronization at specified intervals or you can use the Cache Manager to synchronize the data at any point in time.

When you can predict how often the data needs to be synchronized, schedule the synchronization to occur at specified intervals.

For example, your Web site may list the 100 books that are the top sellers on your Web site. The results of such a query change quite frequently, so you decide to synchronize the data once each hour.

In setting the interval, consider the frequency of updates to the data in the origin database. In addition, consider your environment's tolerance for stale data. If there is low tolerance, synchronize often; if there is high tolerance, synchronize less often.

If you cannot predict how often the data needs to be synchronized, or if you want to synchronize the data at a time other than the scheduled time, you can use Cache Manager to synchronize the data.

For example, a Web site for ordering sporting goods contains a catalog of items. New items are added to the catalog at irregular intervals, making it difficult to predict a schedule for synchronization. In this case, you can synchronize the data when the changes in data in the origin database necessitate it.

When you plan the synchronization of your data, consider carefully how you want the data to be refreshed throughout your Oracle8i Cache environment. You may

want to schedule synchronization at a time of day when connectivity is guaranteed or when communication costs are lowest. If your caches or origin database experiences a high volume of transactions during the business day, consider scheduling the synchronization during off-peak hours. Furthermore, you might want to stagger the synchronization to distribute the load on the origin database over time, rather than scheduling the synchronization of many caches at the same time.

If the synchronization intervals are very short or the data changes rapidly, you may see an effect on performance because of the processing required to keep the middle-tier cache synchronized with the origin database.

If the synchronization takes longer than the interval specified, the table will be continually refreshed. For example, if you specify an interval of 5 minutes, but the synchronization takes 6 minutes to complete, the next scheduled synchronization will start immediately after the first one completes.

If two or more requests to synchronize a particular table are issued simultaneously, one of the requests will succeed. The other requests will fail with an error.

Note that while you can easily change the synchronization policy from scheduled to not scheduled, changing the policy from Complete to Incremental or Incremental to Complete consumes more resources and may take considerable time if the table is very large.

Installing Oracle8*i* Cache

You install Oracle8*i* Cache on all nodes in the middle tier where you want to cache data. When you install Oracle8*i* Cache, you specify which Oracle database you want to use as the origin database. The installation procedure creates a cache on the Oracle8*i* Cache node and prepares the origin database for use by Oracle8*i* Cache.

In addition, you install Oracle Enterprise Manager, including DBA Studio. Cache Manager is a component of DBA Studio.

For more information about how to install Oracle8*i* Cache, see the installation guide.

Invoking Oracle DBA Studio

Oracle DBA Studio is a Java-based interface that you use to manage your Oracle environment. Cache Manager, a component of DBA Studio, provides the interface to manage your Oracle8*i* Cache environment and provides wizards that take you through the process of setting up and deploying your Oracle8*i* Cache environment.

You can start DBA Studio through the Oracle Enterprise Manager console or as a standalone application.

To invoke the Oracle Enterprise Manager console on Windows NT systems, from the Windows NT Start menu, select **Programs** -> **Oracle_*home_name*** -> **Oracle Enterprise Manager** -> **Console.**

To invoke the Oracle Enterprise Manager console on UNIX systems, such as Sun Solaris, enter the following command:

```
oemapp console
```

Then, from the console, select **Tools** -> **Database Applications** -> **DBA Studio** or select the Database Applications icon from the toolbar and then select the DBA Studio icon.

Note that Oracle8*i* Cache does not support the jobs and events subsystem or discovery of cache nodes.

To start DBA Studio as a standalone application on UNIX systems, such as Sun Solaris, enter the following command:

```
oemapp dbastudio
```

To start DBA Studio as a standalone application on Windows NT systems, from the Windows NT Start menu, select **Programs** -> **Oracle_*home_name*** -> **DBA Management Pack** -> **DBA Studio.**

When you invoke DBA Studio, it displays the Oracle Enterprise Manager Login dialog box:
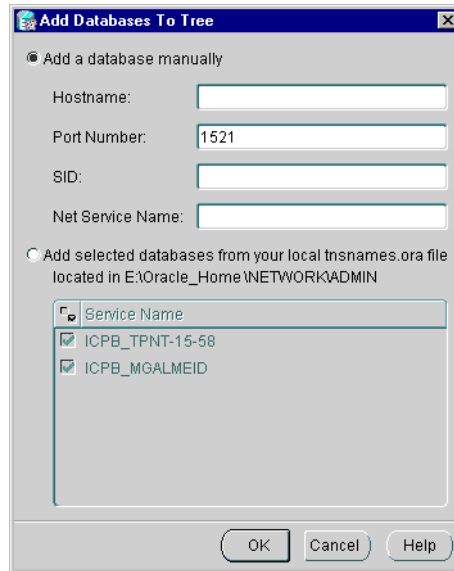

Select **Launch DBA Studio standalone.** Click **OK.**

When you log in successfully, DBA Studio is displayed:

Before you can connect to an origin database, you must add it to the list in the Navigator tree. Take the following steps:

1. Choose **File** -> **Add Database to Tree.**

2. In the Add Databases to Tree dialog box, select **Add a database manually.**

3. In the dialog box, enter the following information:

   - For **Hostname,** enter the name of a node that contains the origin database.

   - For **Port Number**, enter the number of the port of the origin database. By default, the port number is 1521.

   - For **SID,** enter the Oracle system identifier.

   - When you enter the Hostname and SID, DBA Studio completes the **Net Service Name.**

   Alternatively, if the origin database is listed in the local tnsnames.ora file, it is displayed in the **Service Name** list. In this case, you can select **Add selected databases from your local tnsnames file.** Then, select the database service name.

4. Click **OK.**

DBA Studio lists the origin database in the Navigator tree. When you log in to DBA Studio in the future, the database will be listed in the tree.

To connect to the database, take the following steps:

1. Select the origin database from the Navigator tree and click the **Connect** icon. DBA Studio displays the Database Connect Information dialog box.
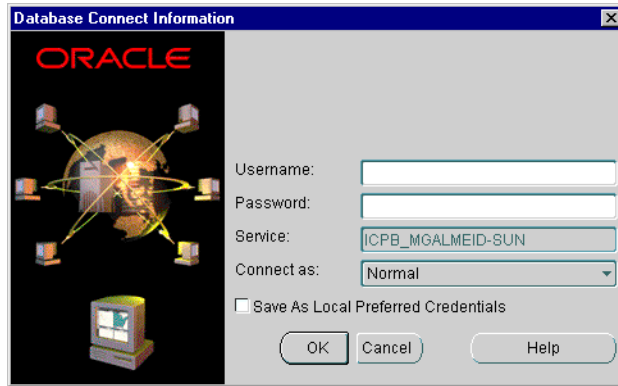
2. Enter the following information:

   ■   For **Username**, enter the name of a valid user for the origin database. The user must have DBA privileges, because many of the operations you can perform require this level of privileges.

   ■   For **Password**, enter the password for the specified user.

3. Click **Save As Local Preferred Credentials.** This saves the connect information and allows you to access the database in the future without specifying the connect information.

4. Click **OK.**

DBA Studio authenticates the user, checking that the user is a valid user with the requisite privileges in the origin database. When you log in successfully, DBA Studio expands the database in the Navigator tree.

To manage your Oracle8*i* Cache environment, select the **Caches** folder from the Navigator tree. DBA Studio displays information about Cache Manager and a button to view the Oracle8*i* Cache Quick Tour.

Expand **Caches** to see the caches associated with the origin database. Oracle8*i* Cache attempts to attach to the associated caches. The icons for each cache show the state of the cache.
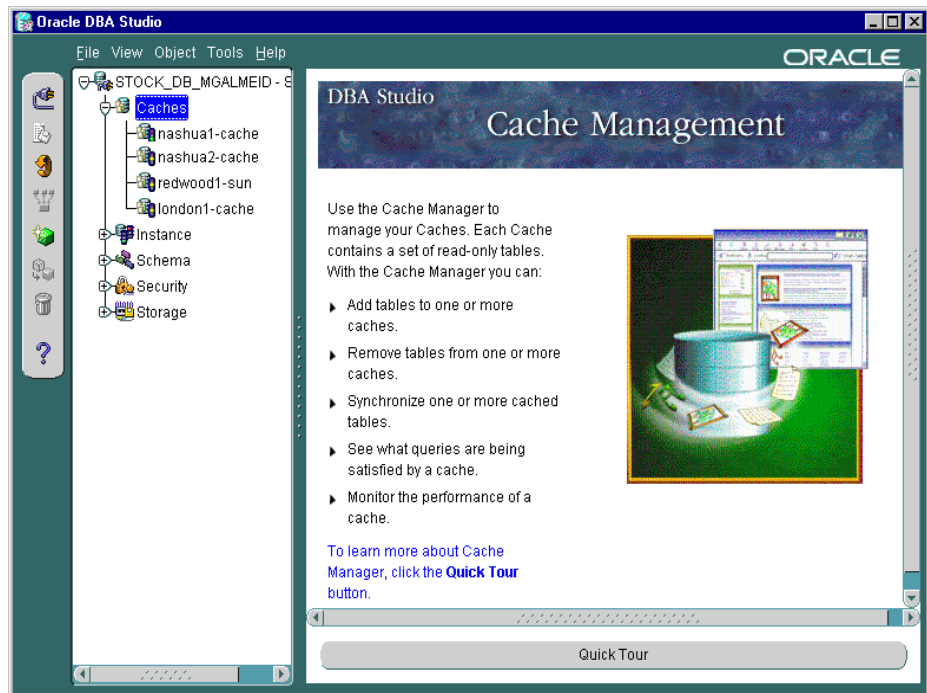
Now, select a cache from the Navigator tree to see information about that cache:
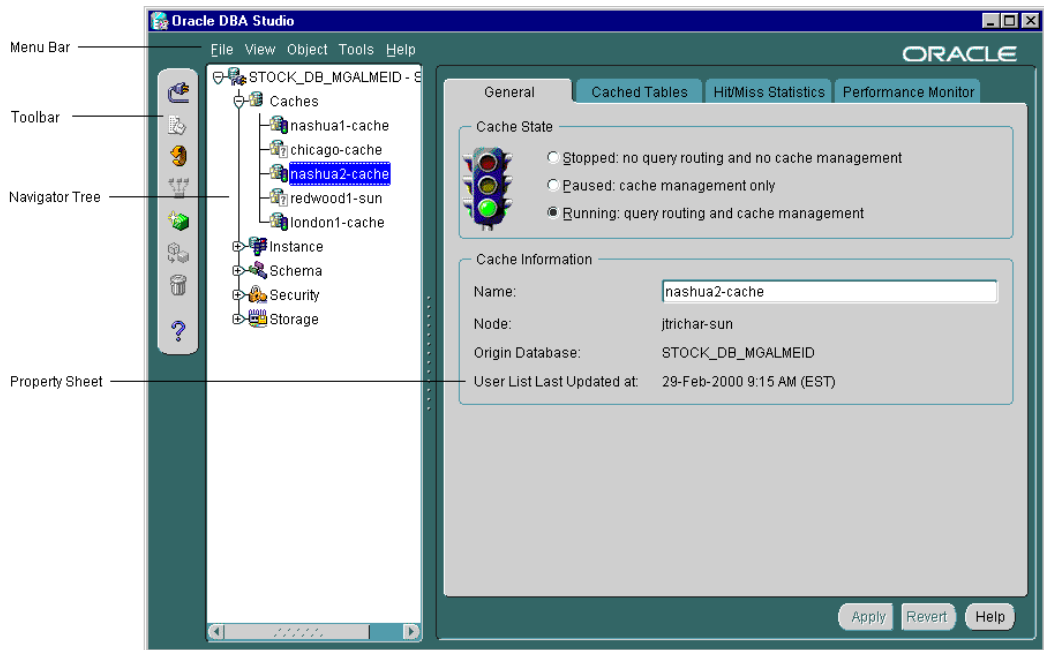
The preceding illustration points out some of the features of DBA Studio and Cache Manager:

- The menu bar contains menus that let you perform many tasks on objects such as caches and tables.

- The toolbar contains buttons for the most common functions, such as changing connections to a different origin database, refreshing the Navigator tree, and adding and removing objects from the tree.

- The Navigator tree shows a hierarchical view of your databases and caches. Under the **Caches** folder, it shows the caches associated with the origin database.

- The property sheet displays information about the cache selected in the Navigator tree.

  When you select a cache, the General property page is displayed, along with tabs for other property pages. If the cache is stopped, only the General page is displayed. If Oracle8*i* Cache is not running on the node or the node is not

reachable, the General property page will appear dimmed and no other property pages will be shown.

For detailed information about each property page, dialog box, and wizard, see the online Help.

Setting Up the Caches

The caches were created when you installed Oracle8*i* Cache. Now, you use Cache Manager to specify which data is cached in which middle-tier caches and the synchronization policy for each table in the caches.

> **Note:** The tables are stored in the cache using the same character set as specified as the character set of the origin database.

Cache Manager provides the **Add Tables Wizard,** which lets you set up and populate one or more caches at the same time. With the wizard, you can add tables to one or more caches and specify the synchronization policy for each table.

When you are setting up your Oracle8*i* Cache environment for the first time, consider using the **Add Tables Wizard** to set up one cache. Then, verify and test that you set up the cache in the best way for your environment. When you are satisfied with the setup, you can easily copy it to other caches. The following sections describe how to use this method.

If you accepted the default value for the amount of disk space allocated to the cache, you should consider increasing the amount of disk space before you add tables to the caches. The following section describes how to add disk space.

Increasing the Disk Space Allocated to a Cache

If you accepted the default value for the amount of disk space allocated to the cache and you plan to cache a large amount of data, you may need to increase the disk space allocated to the cache.

You can increase the amount of disk space available for a cache in the following ways:

- Adding a new file to hold additional disk space
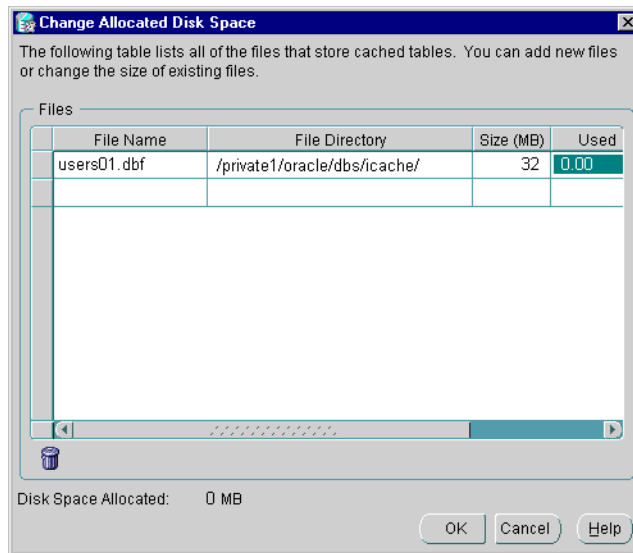- Increasing the amount of disk space in a particular file

Note that performance improves if you allocate the disk space in large amounts. That is, performance is better if you allocate disk space using a few large files rather than many smaller files.

To modify the amount of disk space allocated to a cache, take the following steps to invoke the Change Allocated Disk Space dialog box:

1. From the Navigator tree, select the cache.

2. From the **Cached Tables** tab, click **Change.**

3. Cache Manager displays the Change Allocated Disk Space dialog box, which shows the files that hold the allocated disk space for the cache.

To increase the amount of disk space allocated to a cache by adding a new file, take the following steps:

1. In the Change Allocated Disk Space dialog box, enter the following information in the last row:

    ■ For **File Name,** enter the name for the file.

    ■ For **File Directory,** enter the full specification for an existing directory on the middle-tier node that will hold the file. Note that the naming convention must follow the rules for the middle-tier node.

    ■ For **Size,** enter the size, in megabytes, for the file.

2. Click **OK.** Cache Manager updates the **Disk Space Allocated** field on the Cache Tables page.

3. Click **Apply.** Cache Manager adds the new file to the cache.

To increase the amount of disk space allocated to a cache by increasing the size of an existing file, take the following steps:

1. In the Change Allocated Disk Space dialog box, enter the new amount, in megabytes, in the **Size** field for a particular file.

2. Click **OK.** Cache Manager updates the **Disk Space Allocated** field on the Cache Tables page.

3. Click **Apply.** Cache Manager adds the additional disk space to the cache.

Using the Add Tables Wizard to Set Up Caches

To invoke the **Add Tables Wizard,** select the **Caches** folder from the Navigator tree and choose **Object** -> **Add Tables to Caches.** Then, take the following steps:

1. The wizard displays the Welcome page. Click **Next.**

2. The **Select Caches** page shows the caches to which you can add tables. This page displays all caches that are configured for the current origin database and that are running or paused.

   To cache tables in particular caches, select the caches in which you want to cache tables and click the right arrow (>) to move them from the **Available Caches** list to the **Selected Caches** list. To remove caches in which you do not want to cache tables from the **Selected Caches** list, select the caches and click the left arrow (<).
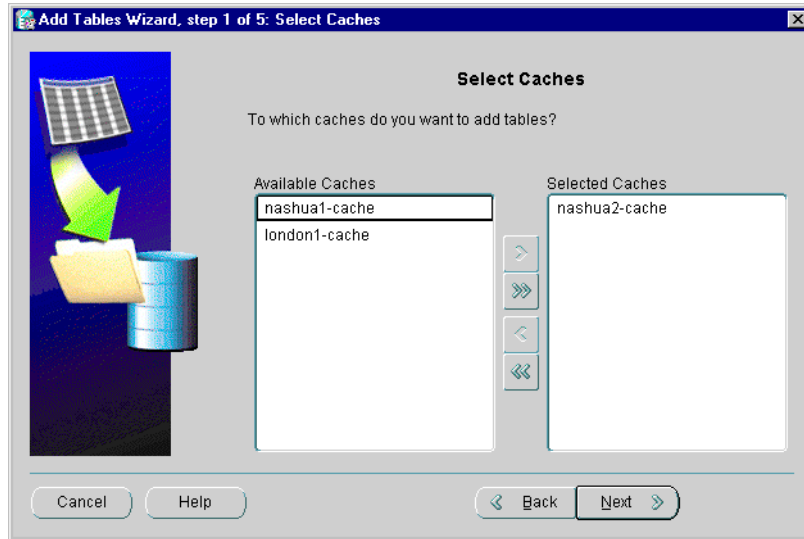
3. Click **Next.**

4. The **Select Tables** page lets you select which tables to cache in the selected caches. In the **Available Tables** list, expand the schemas. The list displays all tables that can be added to the caches you selected. If a table is cached in all selected caches, the table will not be displayed in either list.

   From the **Available Tables** list, select the tables that you want to include in the caches. Click the right arrow (**>**) to move the tables to the **Selected Tables** list.
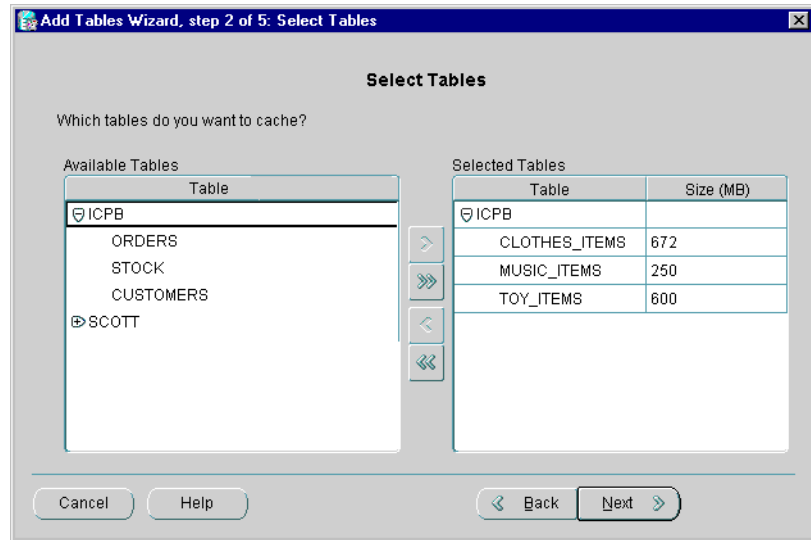
5. When you have selected the tables, click **Next.**

6. On the Synchronization Policy Overview page, you indicate whether you want to use the same synchronization policy for all tables in the caches or to specify different policies for some tables.

   To specify the same synchronization policy for all tables, select **Yes, use the same policy for every table.** If any caches contain any of the tables you selected on the previous page, the wizard will not add those particular tables to those caches and will not change the synchronization policies of those tables on those caches.

   If you want to specify different synchronization policies for one or more tables, select **No, let me set a different policy for some tables.**
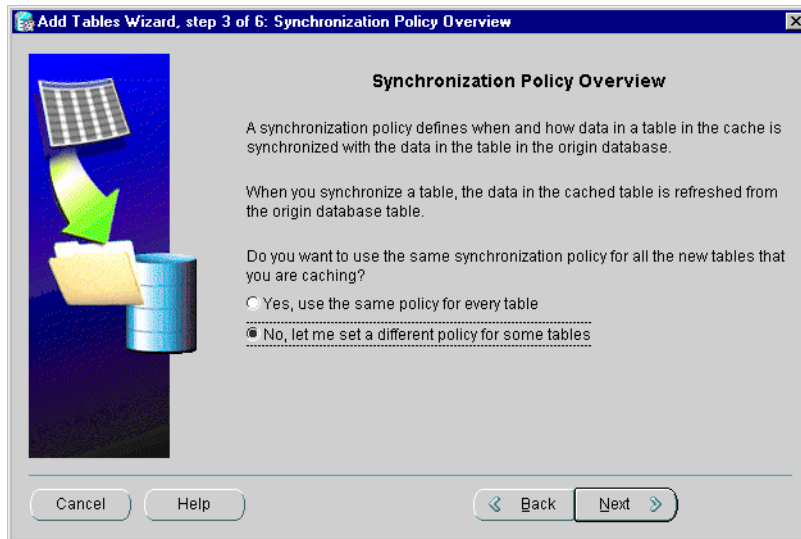
7. Click **Next.**

8. On the Synchronization Policy Method page, you specify the synchronization policy for all or most of the tables you have selected. You can select one of the following methods:

    ■   **Incremental:** Synchronize the data by refreshing only the data that has changed since the table was cached or since it was last synchronized.

    ■   **Complete:** Synchronize the data by completely refreshing the table. When you select this method, Oracle8*i* Cache deletes all data in the table and replaces it with the current data from the origin database.
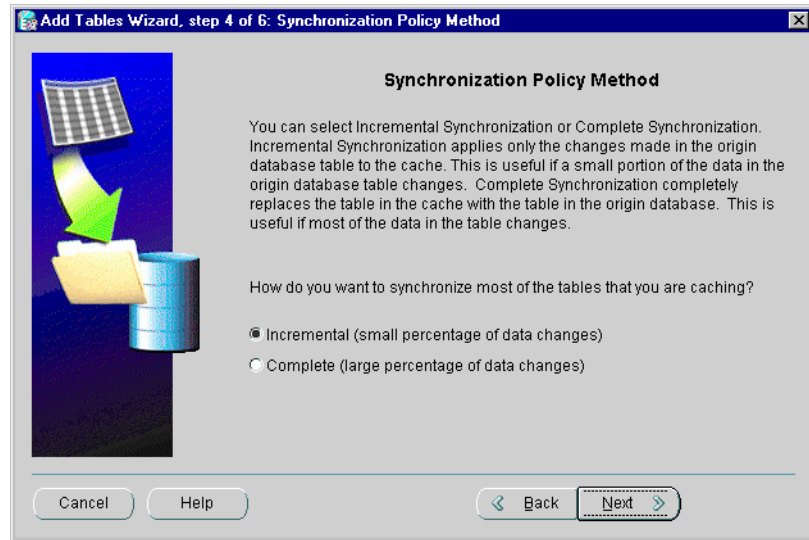
See "Deciding Synchronization Policies" on page 2-4 for more information about synchronization policies.

9. Click **Next.**

10. On the Synchronization Policy Schedule page, you specify when you want the synchronization to occur for most or all of the tables.
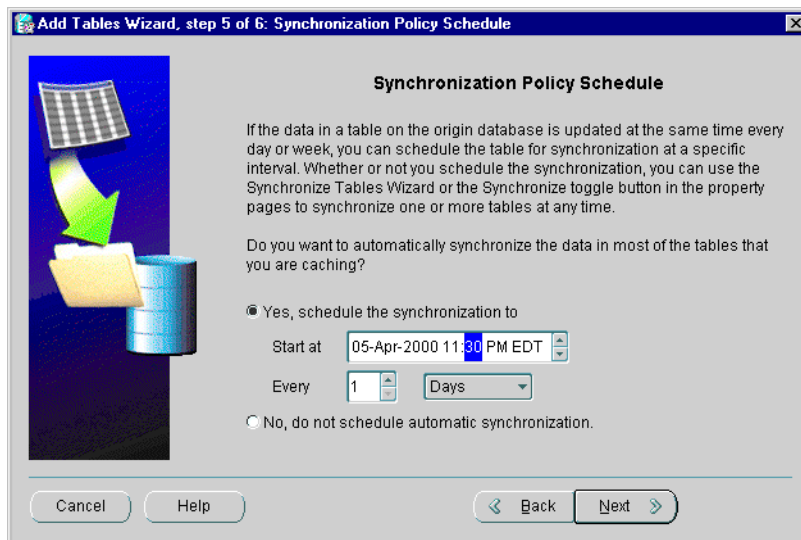
You can choose one of the following:

- **Yes, schedule the synchronization to:** The synchronization occurs at specified intervals. Use this option when you can predict how often the data needs to be synchronized.
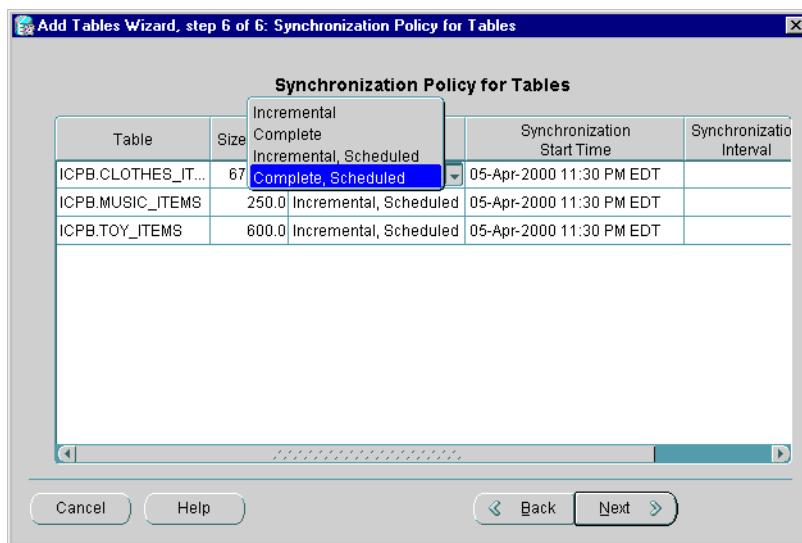
  If you choose this option, you specify the start time and the interval between synchronizations:

  * In the **Start at** field, enter the date and time to start the synchronization. You can select each field in the date and time string and use the arrow to select the correct time. Note that the time reflects the time zone defined for the middle-tier node.

  * In the **Every** field, enter a number for the interval in the first box and select a unit, such as minutes, hours, days, or weeks, in the second box.

- **No, do not schedule automatic synchronization:** The synchronization occurs only when you initiate the action. Use this option when you cannot predict how often the data needs to be synchronized.

11. Click **Next.**

12. If you selected **Yes, use the same policy for every table** in step 6, skip to step 14.

If you selected **No, let me set a different policy for some tables,** Cache Manager displays the next page, **Synchronization Policy for Tables.**

Add Tables Wizard, step 6 of 6: Synchronization Policy for Tables	☒

Synchronization Policy for Tables

Table	Size		Synchronization Start Time	Synchronizatio Interval
		Incremental		
		Complete		
		Incremental, Scheduled		
ICPB.CLOTHES_IT...	67	Complete, Scheduled ▼	05-Apr-2000 11:30 PM EDT	
ICPB.MUSIC_ITEMS	250.0	Incremental, Scheduled	05-Apr-2000 11:30 PM EDT	
ICPB.TOY_ITEMS	600.0	Incremental, Scheduled	05-Apr-2000 11:30 PM EDT	

◀		▶

Cancel	Help		« Back	Next »

For those tables for which you want a different synchronization policy, take these steps:

a. From the **Synchronization Policy** column, select the type of synchronization.

b. If you select Incremental, Scheduled, or Complete, Scheduled, enter the following information:

* From the **Synchronization Start Time** column, select the date and time for the synchronization to begin.

* From the **Synchronization Interval** column, enter a number for the interval between synchronizations.

* From the **Synchronization Unit** column, select a unit of time for the interval, such as minutes, hours, days, or weeks.

13. Click **Next.**

14. The wizard displays the Summary page. Click **Finish.**

Oracle8*i* Cache adds the tables, their data, and their dependent objects to the caches. The wizard displays a progress box that shows each operation as it completes. If an

operation results in an error, the wizard displays an error in the progress box and displays the warning when it completes all operations.

Now you can verify that the caches are set up correctly.

Verifying Your Cache Setup

To verify that you set up your cache correctly, select the cache from the Navigator tree. Cache Manager displays the property sheet for the cache. Then, take the following steps:

1. Select the **Cached Tables** tab and check that all tables that you want cached are listed. See "Modifying Which Data Is Cached" on page 3-6 for information on adding or removing tables.
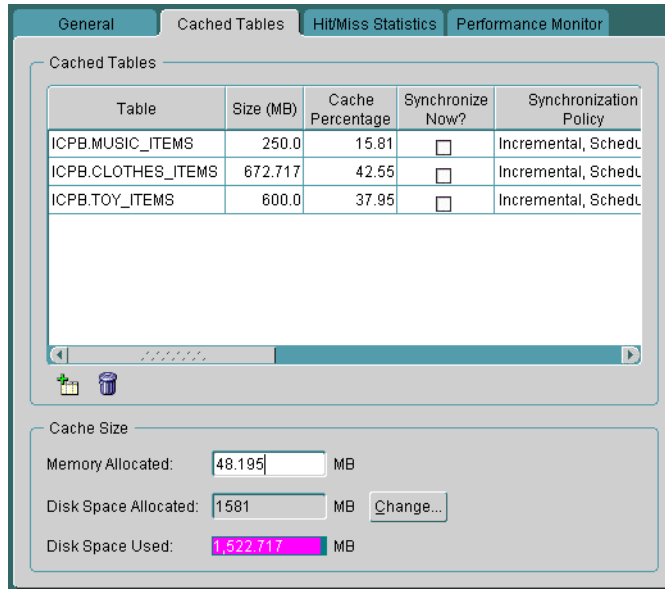
2. On the Cached Tables page, check the synchronization policy and intervals for each table. To change the synchronization policy for a table, click in the **Synchronization Policy** column and select an option.

   If you selected Incremental, Scheduled, or Complete, Scheduled, select a **Synchronization Start Time.** Then, enter a **Synchronization Interval** and select a **Synchronization Unit,** such as minutes, hours, days, or weeks.

3. On the Cached Tables page, check **Memory Allocated**, which indicates the amount of memory allocated to the cache. To change the amount, enter the new amount in the **Memory Allocated** box.

> **Note:** The change to memory allocation will not take effect until the next time the cache is started.

4. On the Cached Tables page, check **Disk Space Allocated,** which indicates the amount of disk space allocated to the cache. To change the disk space allocated, click **Change.** Cache Manager displays the Change Allocated Disk Space dialog box. You can increase or decrease the size of a particular file or add a new file.

   To increase the disk space by increasing the size of a particular file, enter the increased size in the **Size** column.
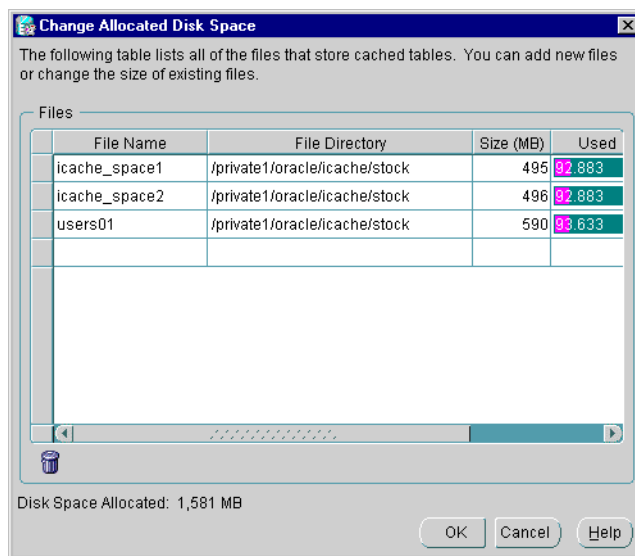
"Modifying the Disk Space Allocated to a Cache" on page 3-11 provides more information about increasing or decreasing the disk space allocated.

5. Click **OK.**

6. If you made any changes, click **Apply.**

If the cache is set up correctly, you can enable caching and test the cache.

Enabling Caching

To enable caching on a middle-tier cache, take the following steps:

1. Select the cache from the Navigator tree. Cache Manager displays the property sheet of the cache.

2. On the **General** page, click **Running.**

3. Click **Apply.**

Oracle8*i* Cache starts the middle-tier cache and enables caching.

Now you can test the cache to see that you cached the appropriate tables and that you specified appropriate amounts of disk and memory. For information about monitoring caches, see Chapter 3.

If the cache is set up correctly, you can copy the setup to other middle-tier caches.

Copying Cached Tables to Other Caches

If you have set up one cache and want to duplicate those cache settings to other nodes, you can easily do so using the Copy Cached Tables dialog box:

1. In the Navigator tree, select the cache that you want to duplicate. Right-click and choose **Copy Cached Tables.**

2. In the Copy Cached Tables dialog box, from the **Available Caches** list, select the caches to which you want to copy the setup and click the right arrow (**>**) to move the caches to the **Selected Caches** list.
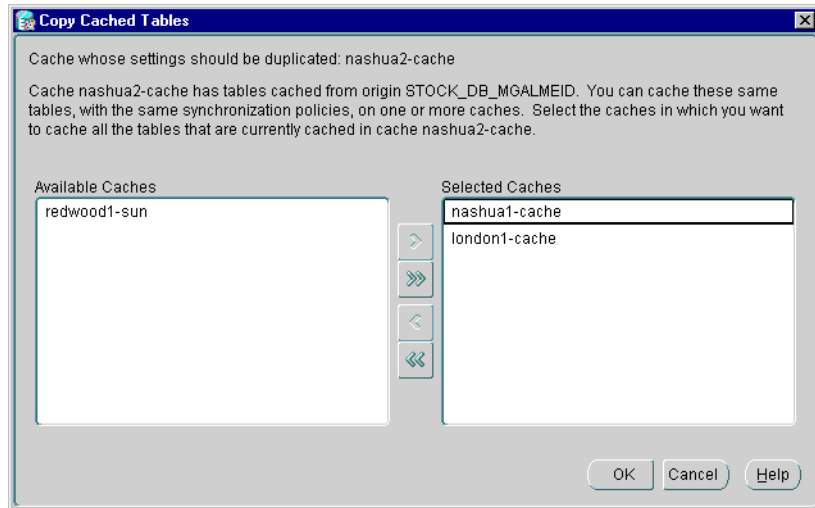
3. Click **OK.**

Repeat the steps outlined in "Verifying Your Cache Setup" on page 2-22 to verify that these caches are set up correctly.

Configuring Your Application Environment

To configure your application environment to use Oracle8*i* Cache, you must set the environment variable ORA_OCI_CACHE and you must ensure that your applications are using the Oracle home in which Oracle8*i* Cache is installed.

You set the environment variable ORA_OCI_CACHE to the value "1" in the environment from which your application is invoked. All applications invoked from this environment take advantage of Oracle8*i* Cache.

For example, on Solaris, you use a shell environment variable.

> **Note:** To use Oracle8*i* Cache, you must run your application from a node on which Oracle8*i* Cache is installed.

If ORA_OCI_CACHE is not set, applications will not use Oracle8*i* Cache. (Note, for this release, if you set the environment variable to a value other than "1", caching will be enabled, although only the value "1" will be supported in future releases.)

See "Controlling Which Applications Use Oracle8i Cache" on page 4-2 if you use applications that directly call OCI functions and you want to exercise greater control over which applications use Oracle8*i* Cache.

You must also make sure that your applications are using the Oracle home in which Oracle8*i* Cache is installed, including OCI libraries from that Oracle home. For example, on Sun Solaris, set the value of the environment variable ORACLE_HOME to the Oracle home for Oracle8*i* Cache. Set the value of the environment variable LD_LIBRARY_PATH so that the Oracle8*i* Cache library directory ($ORACLE_HOME/lib) precedes other Oracle library directories. For more information about using the Oracle home and libraries for Oracle8*i* Cache, see the installation guide.

3

Managing Oracle8*i* Cache

To manage your Oracle8*i* Cache environment, you use Cache Manager, a component of Oracle DBA Studio. Cache Manager provides a Java interface that lets you start and stop caching for one or more middle-tier caches, monitor the ratio of hits to misses, modify which data is cached, and modify the synchronization policy.

Enabling and Disabling Caching

Occasionally, as when you perform maintenance on a node, upgrade software, or change the memory allocation for a cache, you may need to start or stop a cache. Likewise, when you want to test the performance of the queries without Oracle8*i* Cache, you can pause the routing of queries to a middle-tier cache.

Using Cache Manager, you can start or stop caches or pause or resume caching on one or more caches. The **General** page shows the current state of the cache and lets you change the state:

- **Stopped:** Indicates that the cache is currently stopped. As a result, the queries are routed to the origin database. If a cache is stopped, you can only view general information about it. You cannot add or remove tables or synchronize a stopped cache. Statistics are not gathered for the cache while it is stopped.

- **Paused:** Indicates that the cache is running, but that queries are routed directly to the origin database. If a cache is paused, you can view all information about the cache. You can add and remove tables and synchronize a paused cache. Some statistics are gathered while the cache is paused, but the Hit/Miss Percentage and Hit/Miss Count per Second statistics are not gathered.

- **Running:** Indicates that the cache is running and queries are routed to the cache. You can perform all operations on the cache and all statistics are gathered.

| General | Cached Tables | Hit/Miss Statistics | Performance Monitor |

Cache State

- Stopped: no query routing and no cache management
- Paused: cache management only
- Running: query routing and cache management

Cache Information

Name: london1-cache

Node: jmcclung-sun

Origin Database: STOCK_DB_MGALMEID

User List Last Updated at: 25-Mar-2001 12:34 AM (EST)
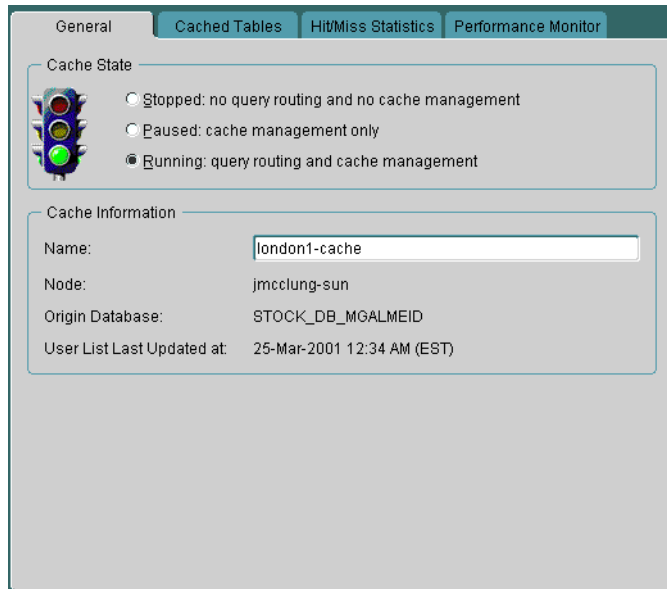
Select an option to change the state of the cache:

- To start a stopped cache, click **Running.** Oracle8*i* Cache starts the selected cache and enables routing of queries to the cache.

- To disable the routing of queries to the cache, click **Paused.** Oracle8*i* Cache does not stop the cache, but it routes queries directly to the origin database.

- To resume the routing of queries to a cache that is paused, click **Running.**

- To stop a running or paused cache, click **Stopped.** Oracle8*i* Cache stops the selected cache and routes queries directly to the origin database.

When you click **Apply,** Oracle8*i* Cache changes the state of the cache.

> **Note:** If you stop a cache and then restart it, you must also restart your application so that the application can reconnect to the cache.

To start or stop caches or to pause or resume query routing on all caches, select the Caches folder and choose one of the following from the **Object** menu:

- Start All

- Stop All
- Pause All

A progress box displays the results of the operation. When the operation is complete, click **Close** to close the progress box.

Monitoring Your Oracle8*i* Cache Environment

Cache Manager helps you to monitor your Oracle8*i* Cache environment. It provides information about which queries are hits and which are misses, as well as information about hits and misses per second, processor time, and memory usage.

To monitor a cache, select the cache from the Navigator tree. Cache Manager displays the property sheet for the cache.

Monitoring Hit and Miss Statistics

To view information about which queries are hits and which are misses, select the **Hit/Miss Statistics** tab. A **hit** is a query that was satisfied in the middle-tier cache. A **miss** is a query that could not be satisfied in the cache and that was routed to the origin database.

By selecting an option from the **View** box, you can display statistics for the following categories of queries:

- Most frequent queries, both hits and misses

- Most frequent queries, hits only

- Most frequent queries, misses only

- Least frequent queries, both hits and misses

- Least frequent queries, hits only

- Least frequent queries, misses only

The page shows the following information about the top 50 queries in the selected category:

- Whether or not each query was a hit or a miss

  Yes denotes a hit; **No** denotes a miss. **Unknown** denotes that Oracle8*i* Cache cannot determine whether the query is a hit or a miss, because new data has recently been cached and the query has not yet been reissued. The next time the query is issued, Oracle8*i* Cache will determine whether the query is a hit or a miss. It attempts to find the data in the cache. If the query cannot be satisfied in the cache, Oracle8*i* Cache routes the query to the origin database and marks the query as a miss. If the query can be satisfied in the cache, Oracle8i Cache returns the data to the user and marks the query as a hit.

- The number of times the query was executed since the statistics were last reset

- The text of the queries that were executed

- The name of the user who executed the query

If a query is a miss, but has been executed many times, consider adding the tables listed in the query to the cache.

Sometimes, the query may be marked as a miss, but the table is already cached. In this case, the user information in the cache may need to be updated. For example, the user's password may have changed in the origin database, but that information has not been propagated to the cache. To update the user information in the cache, select the cache from the Navigator tree and choose **Update User List** from the drop-down menu.

To refresh the statistics, click **Refresh.** However, even when you click Refresh, you may not receive the most up-to-date statistics, because the statistics are gathered once every 5 minutes.

Resetting Hit and Miss Statistics

You can reset the hit and miss statistics for the cache by clicking **Reset.** Consider resetting when you add or remove several tables from the cache. Resetting the statistics resets the hit and miss count to zero, making it easier to see which queries are being executed most frequently at the current time.

Monitoring Performance

To view other performance information about the cache, select the **Performance Monitor** tab.
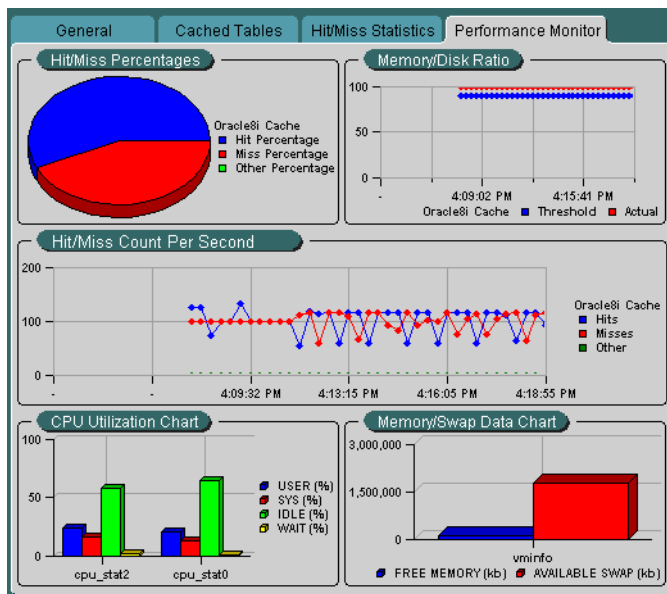
The page shows the following:

- Hit/Miss Percentages Chart: The percentage of hits and misses for a cache

  This chart shows you the percentage of hits (**Hit Percentage**) and the percentage of misses (**Miss Percentage**) for the cache. **Other Percentage** refers to the percentage of requests that were requests other than queries, such as updates, inserts, or deletes. Reviewing this information can help you determine whether or not you have cached the appropriate data.

  If the Hit Percentage is high or the Miss Percentage is low, you have probably cached the appropriate data. If this is not the case, select the **Hit/Miss Statistics**

tab. For **View,** select **Most Frequent Queries, Misses Only.** Consider caching the tables that are accessed by the top queries in this category.

If the Other Percentage is high, your application may not be ideal for Oracle8*i* Cache. Consider moving the update, insert, or delete requests to another application.

- Memory/Disk Ratio Chart: The ratio of queries (hits) that are satisfied using the cache memory instead of using data stored on disk

  For better performance, a high percentage of queries should be satisfied by the data in cache memory because access to memory is much faster than access to disk. The optimal range is 90% or higher. If the percentage is low, consider increasing the amount of memory allocated to the cache. (First, check the Memory Swap Data chart to verify that you have available memory on your system. Then, select the **Cached Tables** tab and specify the new amount in the **Memory Allocated** box.)

- Hit/Miss Count per Second Chart: The number of hit queries per second for the cache and the number of miss queries per second for the cache

  This chart, along with the Hit/Miss Percentage chart, helps you to determine if you have cached the appropriate data.

- CPU Utilization Chart: The CPU usage or non-idle processor time for the cache node

  For caches on UNIX systems, this chart shows the CPU utilization for all processes on the cache node. For caches on Windows NT, this chart shows the non-idle processor time utilization for all processes on the cache node. This chart shows you whether the CPU of the cache node is overloaded or underutilized.

- Memory/Swap Data Chart: The memory swap data or the memory usage for the cache node

  This chart shows whether the memory of the cache node is overloaded or underutilized. The numbers refer to the memory on the cache node.

You can customize the charts and view additional information by right-clicking. The context menu shows the options for that chart.

Modifying Which Data Is Cached

Using Cache Manager, you can add tables to one or more caches or remove tables from one or more caches, as the following sections describe.

> **Note:**   When you add a table to or remove a table from a cache, Oracle8*i* Cache changes the status of all queries that were marked as misses to "Unknown." When Oracle8*i* Cache receives a query marked as Unknown, it attempts to find the data in the cache. If the query cannot be satisfied in the cache, Oracle8*i* Cache routes the query to the origin database and marks the query as a miss.

Adding Tables to One Cache

You can add tables to one cache by using the property sheet of the cache. Take the following steps:

1. From the Navigator tree, select the cache.

2. Select the **Cached Tables** tab and then click the **Add Cached Tables** icon.

3. Cache Manager displays the Add Cached Tables dialog box.
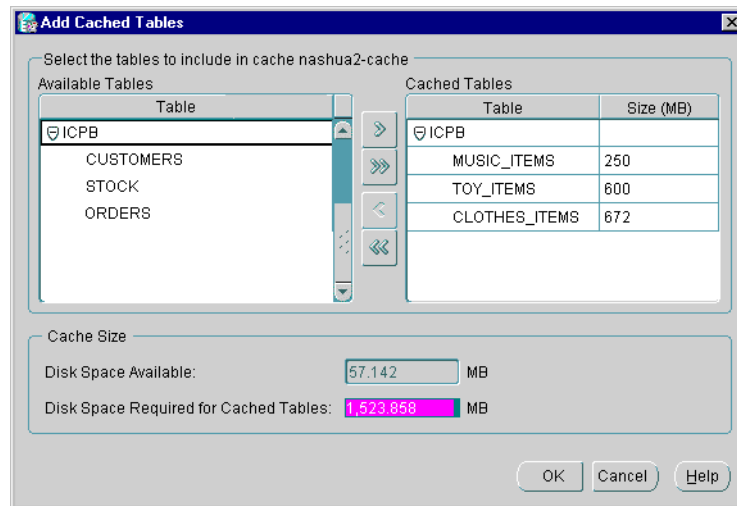
To add a table to the cache, expand the schemas and select the table from the **Available Tables** list. Click the right arrow (**>**) to move the table to the **Cached Tables** list.

You can also remove tables using this dialog box by selecting the table from the **Cached Tables** list. Click the left arrow (**<**) to move the table to the **Available Tables** list.

As you move tables to or from the Cached Tables list, Cache Manager updates the information in the **Disk Space Available** and **Disk Space Required for Cached Tables** boxes. If the tables use more disk space than the **Disk Space Available**, Cache Manager warns you that it cannot add all of the tables. It adds as many tables as the disk space allows.

4. Click **OK.**

5. Click **Apply.**

When you add very large tables to a cache, the operation may take some time to complete.

Adding Tables to One or More Caches

To add tables to one or more caches at the same time, use the **Add Tables Wizard.** With the wizard, you can add tables to one or more caches and specify the synchronization policy for each table.

To invoke the **Add Tables Wizard,** select the **Caches** folder from the Navigator tree and choose **Object** -> **Add Tables to Caches.** Then, take the steps outlined in "Using the Add Tables Wizard to Set Up Caches" on page 2-15.

Removing Tables from One Cache

You can remove tables from one cache by using the property sheet of the cache. Take the following steps:

1. From the Navigator tree, select the cache.

2. From the **Cached Tables** tab, select the table and click the **Remove Cached Tables** icon.

3. Click **Apply.**

Removing Tables from One or More Caches

To remove tables from one or more caches, you can use the **Remove Tables Wizard.** Take the following steps:

1. To invoke the wizard, select the **Caches** folder from the Navigator tree and choose **Object** -> **Remove Tables from Caches.**

2. The wizard displays the Welcome page. Click **Next.**

3. The **Select Caches** page shows the caches from which you can remove tables. By default, the **Available Caches** list is empty because all available caches are listed in the **Selected Caches** list. Click the left arrow (**<**) to move caches from the **Selected Caches** list so that the list contains only those caches from which you want to remove tables.
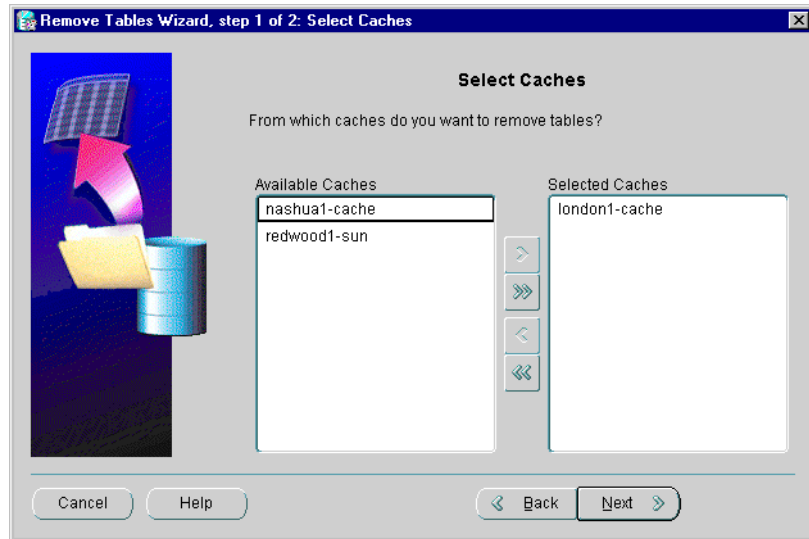
4. Click **Next.**

5. In the Select Tables page, the **Cached Tables** list shows the tables that are cached in one or more of the caches. In the **Cached Tables** list, expand the schemas and select the tables that you want to remove from the caches. Click the right arrow (**>**) to move the tables to the **Tables to Be Removed** list.
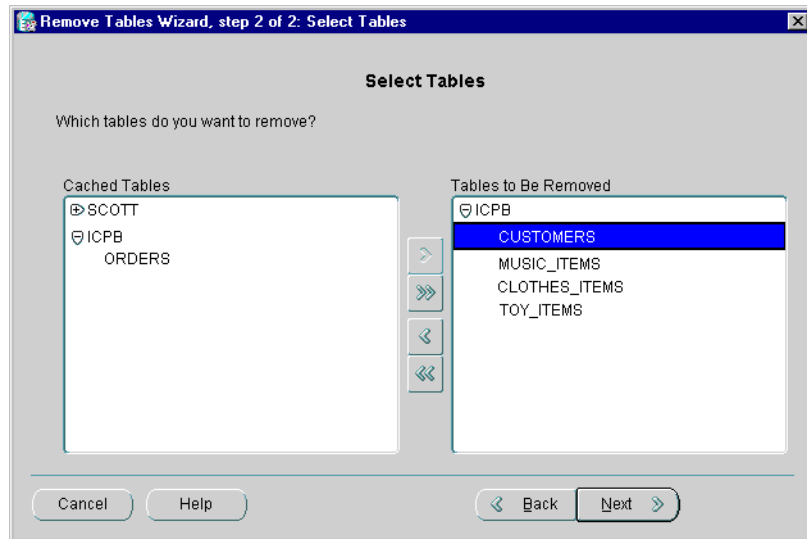
6. Click **Next.**

7. The wizard displays the Summary page. Click **Finish.**

The wizard displays a progress box. When it completes the operation, the selected tables are removed from the caches.

Modifying Characteristics of a Cache

Using the property sheet of a cache, you can change the characteristics of that cache, including the memory and disk space allocated to the cache and the synchronization policy of a table.

Modifying the Memory Allocated to a Cache

To change the amount of memory allocated to a cache, take the following steps:

1. From the Navigator tree, select the cache.

2. From the **Cached Tables** tab, enter the new amount in the **Memory Allocated** box.
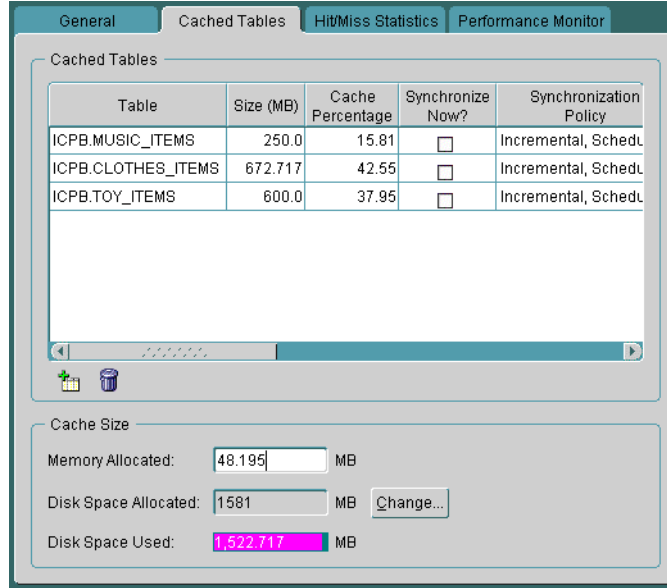
3. Click **Apply.**

Note that the change will not take effect until the next time the cache and applications are started.

Modifying the Disk Space Allocated to a Cache

You can modify the amount of disk space available for a cache in the following ways:

- Add a new file to hold additional disk space

- Increase the amount of disk space in a particular file

- Decrease the amount of disk space in a particular file

Note that performance improves if you allocate the disk space in large amounts. That is, performance is better if you allocate disk space using a few large files rather than many smaller files.

To modify the amount of disk space allocated to a cache, take the following steps to invoke the Change Allocated Disk Space dialog box:

1. From the Navigator tree, select the cache.

2. From the **Cached Tables** tab, click **Change.**

3. Cache Manager displays the Change Allocated Disk Space dialog box, which shows the files that hold the allocated disk space for the cache.

To increase the amount of disk space allocated to a cache by adding a new file, take the following steps:

1. In the Change Allocated Disk Space dialog box, enter the following information in the last row:

   ■ For **File Name,** enter the name for the file.

   ■ For **File Directory,** enter the full specification for an existing directory on the middle-tier node that will hold the file. Note that the naming convention must follow the rules for the middle-tier node.

   ■ For **Size,** enter the size, in megabytes, for the file.

2. Click **OK.** Cache Manager updates the **Disk Space Allocated** field on the Cache Tables page.

3. Click **Apply.** Cache Manager adds the new file to the cache.

You can delete a file if you have not yet clicked **Apply** to add the additional file**.** The row will display an icon in the left column. To delete the file, take the following steps:

1.   From the Change Allocated Disk Space dialog box, select the file.

2.   Click the **Remove File** icon.

3.   Click **OK,** then **Apply.**

To increase the amount of disk space allocated to a cache by increasing the size of an existing file, take the following steps:

1.   In the Change Allocated Disk Space dialog box, enter the new amount, in megabytes, in the **Size** field for a particular file.

2.   Click **OK.** Cache Manager updates the **Disk Space Allocated** field on the Cache Tables page.

3.   Click **Apply.** Cache Manager increases the size of the data file.

To decrease disk space allocated to the cache, you can decrease the size of one or more existing files. Take the following steps:

1.   In the **Size** column for a particular file, enter the new size. It must not be less that the amount of disk space shown in the **Used** column.

2.   Click **OK.** Cache Manager updates the **Disk Space Allocated** field on the Cache Tables page.

3.   On the property page, click **Apply**. Cache Manager decreases the size of the file.

Modifying the Synchronization Policy

The **synchronization policy** establishes how and how often the data in the cache is refreshed from the origin database. When you set up a middle-tier cache, you specified the synchronization policy for each table in that cache. Sometimes, though, the frequency or extent at which the data in the origin database is modified does not stay the same, necessitating a change to the synchronization policy for a table.

For example, for an online catalog, prices and merchandise are changing more rapidly than when you first created the cache. Instead of refreshing the data once a week, you want to refresh it once a day.

To modify the synchronization policy for a table, take the following steps:

1.   From the Navigator tree, select the cache.

2. Select the **Cached Tables** tab. This page displays the tables in the cache and the synchronization policy for each.

3. For each table that you want to modify, click in the **Synchronization Policy** column and select an option from the drop-down list.
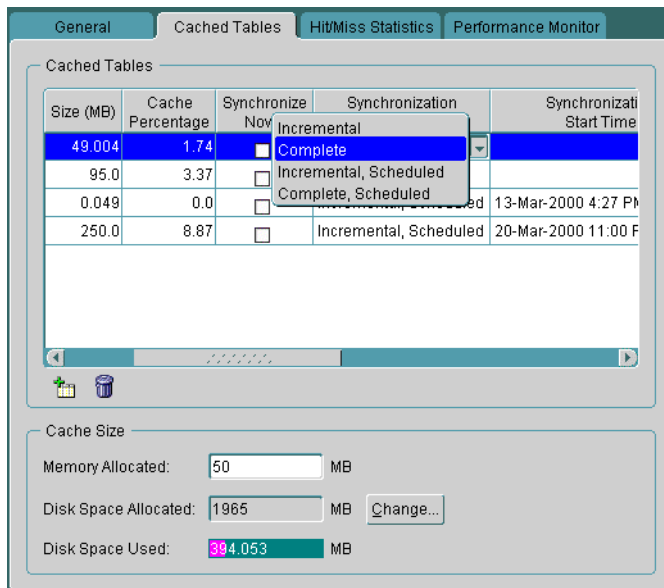
4. If you selected Incremental, Scheduled, or Complete, Scheduled, select a **Synchronization Start Time.** Then, enter a **Synchronization Interval** and select a **Synchronization Unit,** such as minutes, hours, days, or weeks.

   For example, for the table BOOKS, you could select a start time of 02-Feb-2000 3:00 AM, an interval of 1, and a unit of 1 Week. In this case, the table would be refreshed once a week at 3:00 a.m., beginning on February 2.

5. Click **Apply.**

Note that while you can easily change the synchronization policy from scheduled to not scheduled, changing the policy from Complete to Incremental or Incremental to Complete consumes more resources and may take considerable time if the table is very large.

Synchronizing Data

When the synchronization policy of a table is not scheduled, you must initiate the synchronization of the table using Cache Manager. Even if the synchronization policy is Scheduled, you may decide that you want to initiate the action, rather than wait until the next scheduled synchronization.

For example, the table SPORTSWEAR for your online catalog is scheduled for synchronization in two days, but the data in the origin database has changed substantially with the addition of new spring clothing and markdowns on winter clothing. You do not want to wait for the scheduled synchronization to update the data in the caches.

Cache Manager lets you synchronize tables in two ways:

- You can synchronize tables in one cache by using the Cached Tables tab.
- You can synchronize tables in more than one cache by using the Synchronize Tables Wizard.

The following sections describe these methods.

Synchronizing Data in One Cache

To initiate the synchronization of some or all of your tables in one cache, take the following steps:

1. Select the cache from the Navigator tree.
2. From the **Cached Tables** tab, click the check box in the **Synchronize Now?** column for the table or tables you want to synchronize.
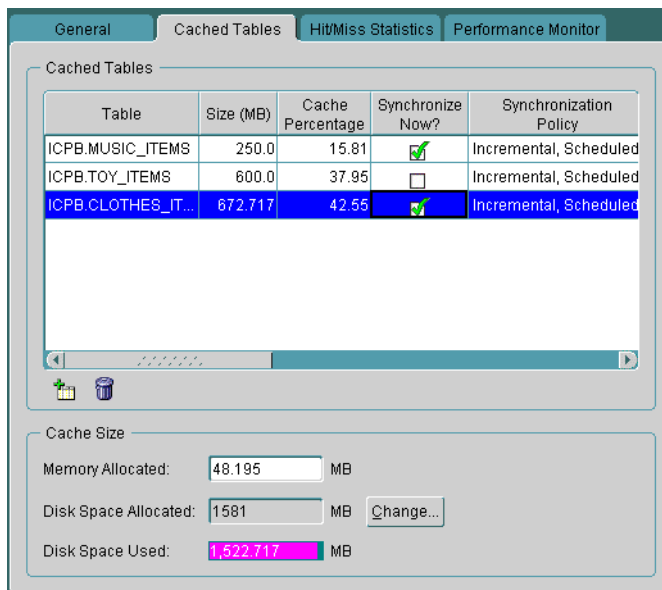
3. Click **Apply.**

If the synchronization policy of a table is Complete, Cache Manager performs a Complete synchronization on the selected table immediately. Similarly, if the synchronization policy of a table is Incremental, Cache Manager performs an Incremental synchronization immediately.

Synchronizing Data in More than One Cache

To initiate the synchronization of some or all of your tables in more than one cache, you use the **Synchronize Tables Wizard.** The wizard and Oracle8*i* Cache synchronize the tables in parallel. For example, if you want to synchronize TABLE_A and TABLE_B on Cache1 and Cache2, the wizard starts the synchronization of TABLE_A on both Cache1 and Cache2. The operations on multiple caches run in parallel. When the synchronization of TABLE_A is completed on Cache1, the wizard starts the synchronization of TABLE_B on Cache1.

If the synchronization policy of a table is Complete, the **Synchronize Tables Wizard** performs a Complete synchronization on the table immediately. Similarly, if the synchronization policy of a table is Incremental, the **Synchronize Tables Wizard** performs an Incremental synchronization immediately.

To synchronize tables immediately, take the following steps:

1. From the Navigator tree, select the **Caches** folder. Choose **Object** -> **Synchronize Cached Tables.**

2. Cache Manager displays the **Synchronize Tables Wizard** Welcome page. Click **Next.**

3. The **Select Caches** page shows the caches that you can synchronize. By default, the **Available Caches** list is empty because all available caches are listed in the **Selected Caches** list. To synchronize tables in only particular caches, select the caches that you do not want to synchronize and click the left arrow (**<**) to move those caches to the **Available Caches** list.

4. Click **Next.**

5. The **Select Tables** page shows the tables that are cached in one or more of the selected caches and the synchronization policy of each table. In the **Cached Tables** list, expand the schemas and select the tables that you want to synchronize. Click the right arrow (**>**) to move the tables to the **Tables to Be Synchronized** list.
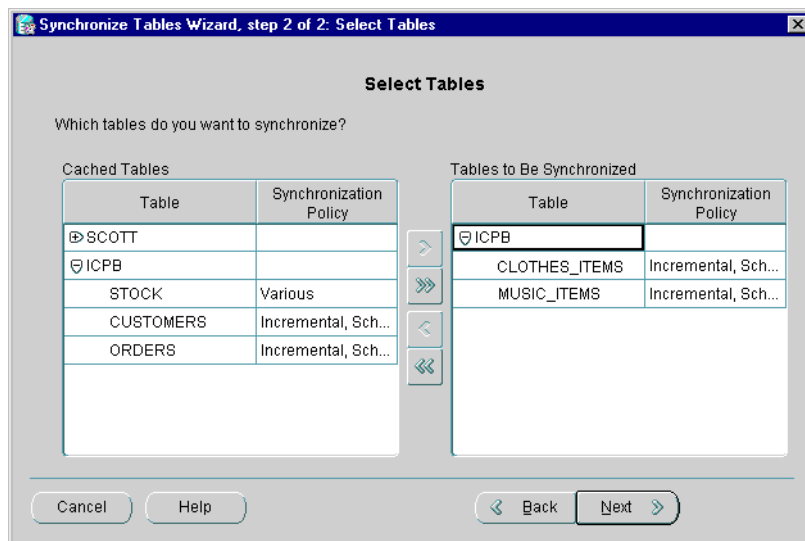
If the synchronization policy is listed as Various, the table is cached in more than one cache and the synchronization policy of the table is not the same in all caches.

6. Click **Next.**

7. The wizard displays the summary page. Click **Finish.**

Oracle8*i* Cache immediately synchronizes the selected tables using the synchronization policy that is specified for each table. That is, if a table has a synchronization policy of Complete (whether or not it is Scheduled), Oracle8*i* Cache immediately performs a complete synchronization. If a table has a synchronization policy of Incremental (whether or not it is Scheduled), Oracle8*i* Cache immediately performs an incremental synchronization.

If the tables are very large and the synchronization method is Complete, the operation may take some time to complete. Similarly, if the tables are very large and the synchronization method is Incremental *and* there are many changes to the data in the origin database, the operation may take some time to complete.

Updating Users

When you install Oracle8*i* Cache, the installation procedure reads the list of users in the origin database and propagates the list of users and the attributes of the users to the cache. (The procedure copies only the user-defined database users, not system-defined database users, such as SYS and SYSTEM.) As a result, you can cache tables from the schemas associated with these users in your cache, and applications can access data owned by these users.

If a new user-defined database user has been added to the origin database, you must add that user to the cache before you can cache tables from that user's schema. Similarly, if an attribute, such as the password, of an existing user is modified in the origin database, you must update the attribute in the cache. If you do not update the password, for example, applications invoked by that user will not be able to access data in the cache.

If a user is dropped from the origin database, you must drop the user from the cache. If you do not, queries on tables in that user's schema continue to access the data in the cache. In addition, synchronization of the table may fail.

To add or drop a user or update the user's attributes, use Cache Manager to update the list of users for one cache or for all caches associated with an origin database. To maintain consistency between the origin database and all caches, Oracle Corporation recommends that you update users for all caches at the same time.

To update the list of users and their attributes for all caches associated with one origin database, take the following steps:

1. From the Navigator tree, select the Caches folder.

2. Right-click, then choose **Update All User Lists** from the drop-down menu.

3. Cache Manager displays a confirmation box. Click **OK.**

Cache Manager updates the user list for all caches and displays a progress box. If Cache Manager cannot update the user list for a cache, perhaps because the cache is not running, it displays a message.

To update the list of users and their attributes for one cache, take the following steps:

1. From the Navigator tree, select the cache.

2. Right-click, then choose **Update User List** from the drop-down menu.

3. Cache Manager displays a confirmation box. Click **OK.**

Cache Manager updates the user list for the selected cache and displays a progress box.

To see the date when the user list was last updated for a cache, check the **User List Last Updated at** field on the General page.

> **Note:** For applications to make use of the updated user list, you must restart your application.

Using the Configuration Assistant to Reconfigure a Cache

To make most changes to a middle-tier cache, such as adding or removing tables or modifying the memory or disk space allocation, you can use Cache Manager, the management interface for Oracle8*i* Cache.

However, you must use the Oracle8*i* Cache Configuration Assistant to reconfigure the following characteristics of the cache:

- Changing the host name for the middle-tier cache

- Changing the port number for the listener for Oracle8*i* Cache

In addition, you can use the Configuration Assistant to change the cache name, although you can also accomplish this using Cache Manager.

For example, to modify a cache on a Sun Solaris system, invoke the Configuration Assistant by using the following commands on the cache node:

```
$ cd $ORACLE_HOME/bin
$ wtacca
```

The Configuration Assistant displays the Welcome page. To modify your cache, click **Next** and take the following steps:
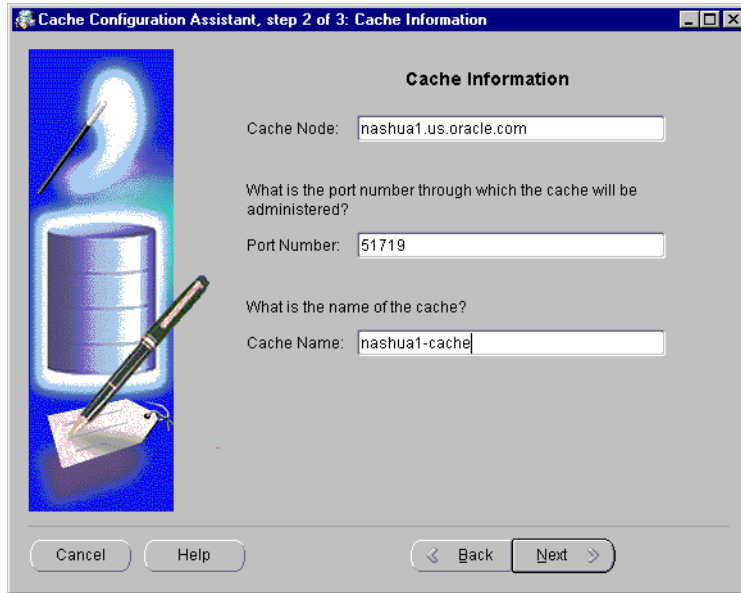
1. On the Origin Database Credentials page, enter the following information:

   - For **User Name,** enter the name of a user on the origin database who has the SYSDBA role.

   - For **Password,** enter the password of the specified user.
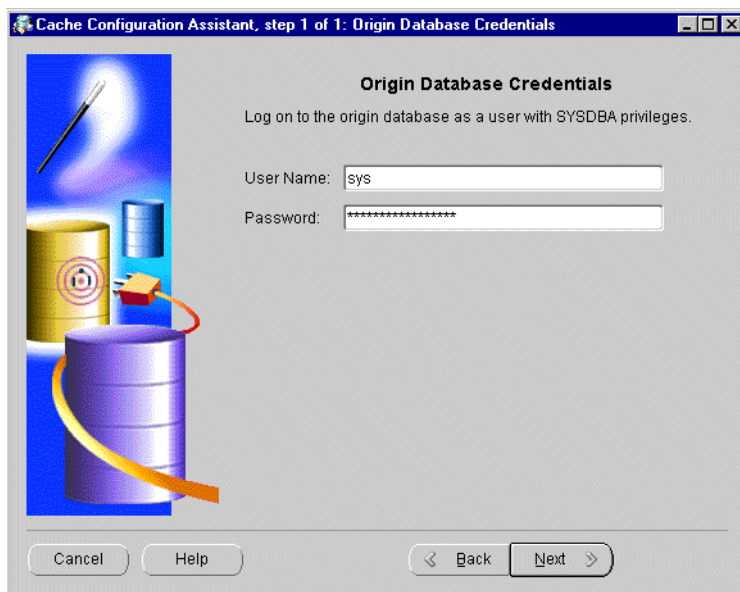
If you enter a user name that is not valid or that does not have SYSDBA privileges, or if you enter an invalid password, the Configuration Assistant returns an error and allows you to enter another value.

2. Click **Next.**

3. On the Cache Information page, you can modify the following information:

   - For **Cache Node,** if your system contains more than one network card, you can enter a host name for the middle-tier node.

   - For **Port Number,** you can change the port number for the listener for Oracle8*i* Cache. You can enter any valid port number that is not in use.

■ For **Cache Name,** you can change the name of the cache. You can enter any string that uniquely identifies the cache, although Oracle Corporation recommends that you use only alphanumeric characters. Do not use a single quotation mark (').

4. Click **Next.**

5. The Summary page provides information about the origin database, the cache, and the changes to be made. Click **Finish.**

The Configuration Assistant displays a progress box informing you of the results of the modifications.

4

Tuning Your Oracle8*i* Cache Environment

In most cases, after you use Cache Manager to set up your Oracle8*i* Cache
environment, you do not need to take any further action to realize improvements in
throughput and performance. However, this chapter provides some hints that may
be useful, depending on your configuration.

Linking Applications That Use Static OCI Libraries

To take advantage of the benefits of Oracle8*i* Cache, you do not need to modify your
applications. Applications that use SQL statements to access the database are routed
to the middle-tier cache automatically when you have defined the environment
variable ORA_OCI_CACHE.

Similarly, applications that are linked with Oracle Call Interface (OCI) using *dynamic*
libraries and layered directly on OCI are routed to the cache automatically.
Dynamic libraries, also called **shared libraries,** are libraries that are linked into the
application at runtime. (Applications layered directly on OCI are those applications
that call OCI directly or use ODBC, JDBC-OCI, or other interfaces that call OCI
directly.)

Applications that are linked with OCI using *static* libraries and layered directly on
OCI require that you relink them with the OCI library in the Oracle8*i* Cache Oracle
home. **Static libraries** are libraries that are linked together into a single executable
program with objects of the whole application.

For more information about linking applications with OCI, see the *Oracle Call
Interface Programmer's Guide* and the *Oracle8i Administrator's Reference* for your
platform.

Controlling Which Applications Use Oracle8*i* Cache

All applications that are started from and running in a process in which the environment variable ORA_OCI_CACHE is set to "1" and that call OCI (directly or indirectly), use the Oracle8*i* Cache middle-tier cache. (See "Configuring Your Application Environment" on page 2-25 for information about this environment variable.) If ORA_OCI_CACHE is not set, applications will not use Oracle8*i* Cache.

If you want to exercise greater control over which applications use Oracle8*i* Cache, you can use a combination of the environment variable ORA_OCI_CACHE and either the OCI_CACHE or OCI_NO_CACHE value to the mode parameter for the OCIEnvCreate() function call. Setting the mode creates an environment handle with the characteristics of that mode.

To specify that an application uses the Oracle8*i* Cache middle-tier cache, specify OCI_CACHE as a value to the mode parameter for the OCIEnvCreate() function call. To specify that an application does not use Oracle8*i* Cache, specify OCI_NO_CACHE as a value to the mode parameter for the OCIEnvCreate() function call.

The following shows an example of specifying OCI_CACHE in the OCIEnvCreate() function call.

```
OCIEnvCreate((dvoid**)&envhp, (ub4) OCI_DEFAULT | OCI_CACHE,
                        (dvoid *)0,
                        (dvoid * (*)(dvoid *, size_t)) 0,
                        (dvoid * (*)(dvoid *, dvoid *, size_t))0,
                        (void (*)(dvoid *, dvoid *)) 0,
                        (size_t)0, (dvoid **)0))
```

Note the following:

- The OCI_CACHE or OCI_NO_CACHE value to the mode parameter overrides the ORA_OCI_CACHE environment variable setting. It will affect the behavior of all sessions, connections, and statement executions under that environment handle.

- Only one of the two values should be used in an application. If both are used, the environment handle is created with OCI_NO_CACHE.

- After you modify your application, you must recompile and relink the application.

Controlling Which Statements Use Oracle8*i* Cache

You can control which statements within an application use Oracle8*i* Cache by using the OCI_ATTR_NO_CACHE statement handle attribute. By default, all statements are routed to the cache.

To specify that a statement not be routed to the cache, use the OCIAttrSet() function to set the value of the attribute to 1. If you set the value to 0 or if the attribute is not set, the statement will be routed to the cache for execution. You use the OCIAttrGet() function to retrieve the value of the attribute.

The following example shows how to use the OCI_ATTR_NO_CACHE attribute:

```
if (OCIAttrSet((dvoid *)stmthp1, (ub4)OCI_HTYPE_STMT,
               (dvoid *) &rem, (ub4)0, (ub4)OCI_ATTR_NO_CACHE,
               errhp1))
```

Consider using this attribute to route complex retrieval statements that you want to be executed in the origin database directly to the origin database.

Note the following:

- If caching has been disabled (by not setting the environment variable ORA_OCI_CACHE or by specifying the OCI_NO_CACHE value to the mode parameter for OCIEnvCreate()), this attribute has no effect.

- After you modify your application, recompile and relink the application.

- This attribute is unrelated to the attribute OCI_ATTR_CACHE.

Tuning Hints

The following tips contain information to help you get the best performance from Oracle8*i* Cache:

- If you need to increase the disk space allocated to the cache, note that performance improves if you allocate the disk space in large amounts. That is, performance is better if you allocate disk space using a few large files rather than many smaller files.

- For better performance, a high percentage of queries should be satisfied by the data in cache memory, because access to memory is much faster than access to disk. The optimal range is 90% or higher. To see the percentage, select the Performance Monitor tab and check the Memory/Disk Ratio chart. If the percentage is low, consider increasing the amount of memory allocated to the

cache. (Select the **Cached Tables** tab and specify the new amount in the
Memory Allocated box.)

5

Troubleshooting Oracle8*i* Cache

This chapter describes error logging, cautions against certain actions, describes some problems you may encounter in your Oracle8*i* Cache environment and solutions to those problems, and notes restrictions to the software.

Logging Errors

To help you troubleshoot problems in your environment, Oracle8*i* Cache writes error messages to a log file. By default, the file specification of the log file is:

```
$ORACLE_HOME/admin/icache/bdump/wtcme.log
```

You can change the name and location of the log file and the level of errors written to the log file. To make these changes, modify the parameters in the file `wtcme.ora` or specify a parameter to the `wtcme` command.

The file `wtcme.ora` is located in the following directory:

```
$ORACLE_HOME/admin/icache/pfile/wtcme.ora
```

The following table describes the parameters that are specified in the wtcme.ora file:

Parameter	Description
authenticationserver	The tnsname of the origin database. Do not use the command line or edit the file to change this parameter.
	The tnsname is set during installation.

Parameter	Description
port	The port number of the listener for Oracle8*i* Cache. The listener establishes a communication path between Oracle8*i* Cache on the middle tier and Cache Manager on the client tier.
	Do not use the command line or edit the file to change this parameter. The port number is set during installation or by using the Configuration Assistant as described in "Using the Configuration Assistant to Reconfigure a Cache" on page 3-19.
	By default, the port number is 51719.
logfile	The full file specification for the log file. By default, the value is set to $ORACLE_HOME/admin/icache/bdump/wtcme.log
loglevel	The level of error messages written to the log file wtcme.log
	Valid values are:
	■ none: No error messages are written to the file.
	■ error: Error messages are written to the file.
	■ debug: Error messages and other information (such as the start time for the management engine, the TCP port, client connects, client disconnects) are written to the file.
	■ trace: Error messages and other information (such as the start time for the management engine, the TCP port, client connects, client disconnects) are written to the file. In addition, all messages sent and received by the Oracle8*i* Cache management engine are written to the file.
	Because the volume of messages written to the log file is very great, use this value only when you are investigating a problem and the other values do not provide enough information.

The following shows the command that specifies a log level of debug:

```
$ wtcme loglevel=debug
```

Important Cautions

The following describes some important information:

- Oracle8*i* Cache creates some users on the origin database to facilitate communication between the caches and the origin database. The following users are created:

    – During the installation, the user ORACACHE is created.

    – When you add tables to a cache, Oracle8*i* Cache adds a user for each schema that contains cached tables. The users are named using the following format:

    ORACACHE*$n*

    In the format, *n* represents an integer between 1 and 1000.

    Do *not* delete these users from the origin database or change their passwords.

- During installation, Oracle8*i* Cache creates the following packages in the SYS schema in the origin database:
  - DBMSSIZE
  - DBMS_CACHE_MASTER
  - DBMS_CACHE_MASTER_UTL
  - WTCSINI
  - WTCSYS

  Do *not* modify or delete these objects.

- You must make sure that Oracle8*i* Cache and related software is started. See "Starting Oracle8i Cache and Related Software" on page 5-3 for more information.

- Because Oracle8*i* Cache uses threads to monitor and maintain statistics about the cache, your application must use OCI in threaded mode. See "Applications Must Be Linked in Multithreaded Mode" on page 5-5 for more information.

Starting Oracle8*i* Cache and Related Software

If you followed the procedures for automating startup and shutdown in the installation guide, Oracle8*i* Cache (as well as the Oracle8*i* Cache management engine, Oracle Data Gatherer, and TNS Listener) is started when your system starts.

To use Cache Manager to manage your caches, the Oracle8*i* Cache management engine must be started. When the management engine is not started on a middle-tier node, Cache Manager cannot access the cache. Cache Manager displays the General page for that cache as dimmed.

If Oracle Data Gatherer and TNS Listener are not started, you will not be able to use Oracle8*i* Cache or some functionality of the product.

If Oracle8*i* Cache is stopped, you can restart it, the Oracle8*i* Cache management engine, Oracle Data Gatherer, and TNS Listener by executing the `cachstrt` script, which is located in the $ORACLE_HOME/bin directory. Use the following command:

```
cachstrt
```

To stop Oracle8*i* Cache and the TNS Listener, execute the `cachshut` script, which is located in the $ORACLE_HOME/bin directory:

```
cachshut
```

To execute these commands, you must be logged in as the user under which Oracle8*i* Cache was installed.

In addition, you can start the related software separately, as the following sections describe.

Starting Oracle Data Gatherer

Oracle Data Gatherer, which manages the collection of statistics for use by Oracle Enterprise Manager components such as DBA Studio, must be started on the middle-tier machine. If it is not, you may receive the following error when you click on the Performance Monitor tab for a cache:

```
Cannot connect to Data Gatherer for cache. Some performance charts will not be
available.
```

On Sun Solaris, use the following command to start Oracle Data Gatherer:

```
vppcntl -start
```

On Windows NT, take these steps to start Oracle Data Gatherer:

1. From the Windows NT Start menu, select **Settings** -> **Control Panel.** Click **Services.**

2. Select **OracleDataGatherer** and click **Start.**

Starting the TNS Listener

TNS Listener must be started on the middle-tier node. If it is not, queries are routed to the origin database.

Note that the listener on the middle-tier node is configured to listen for IPC protocol connection requests. The IPC protocol is used by applications that reside on the same node as the listener to communicate with a cache.

On Sun Solaris, use the following command to start the listener:

```
lsnrctl start listener
```

On Windows NT, take these steps to start the listener:

1. From the Windows NT Start menu, select **Settings** -> **Control Panel.** Click **Services.**

2. Select **OracleTNSListener** and click **Start.**

Applications Must Be Linked in Multithreaded Mode

Oracle8*i* Cache uses threads to monitor and maintain statistics about the behavior of the cache. As a result, if an application already uses OCI in threaded mode, you do not need to make any changes to the application. If the application is a single-threaded application, you must relink the application using the multithreaded mode.

For example, on Sun Solaris, you use the -lthread option in the link line.

If the application is not linked in multithreaded mode, Oracle8*i* Cache may not execute properly and may return errors, such as the following error:

```
Failed: Register Aggregate Stats Queue
```

A

Error Messages

This appendix describes the error messages that you may encounter using Oracle8*i* Cache.

A.1 Message Formats

To help you find and fix errors, Oracle8*i* Cache embeds object names, numbers, and character strings in some messages. In this appendix, these embedded variables are represented by *string*, *number*, *date*, or *character*.

All messages displayed by Oracle software are prefixed by text that indicates which program issued the message. Oracle8*i* Cache uses the prefix WTE. For lists of the prefixes used by other Oracle products, see *Oracle8i Error Messages*.

A.2 Oracle8*i* Cache Error Messages

WTE-01500 Error: Unable to get necessary parameters for initialization

Cause: An error occurred attempting to get necessary parameters for initialization, including port and AuthenticationServer. The parameter file wtcme.ora may not exist in the correct location.

Action: Ensure that the parameter file wtcme.ora exists in the directory $ORACLE_HOME/admin/icache/pfile and that it contains the parameters port and AuthenticationServer or that the command line includes the parameters.

WTE-01504 Error: ORACLE_HOME is not defined

Cause: ORACLE_HOME is not defined.

Action: Define ORACLE_HOME by setting the environment variable.

WTE-01505 Error: Origin database service name (AuthenticationServer) is not provided

Cause: The parameter file wtcme.ora file does not exist, or the parameter AuthenticationServer is not specified in the wtcme.ora file.

Action: Ensure that the parameter file exists and that the parameter AuthenticationServer is specified in the parameter file.

WTE-01506 Error: TCP port *number* is not available

Cause: The port number specified is already in use. This may occur if the Oracle8*i* Cache management engine is already started or another application is using the port.

Action: Verify that the Oracle8*i* Cache management engine is not already running. If another application is using the port, run the Oracle8*i* Cache Configuration Assistant to change the port number for Oracle8*i* Cache.

WTE-01507 Error: OCI environment initialization failed

Cause: The Oracle8*i* Cache management engine is unable to initialize an OCI environment handle. This may occur if the LD_LIBRARY_PATH environment variable is not set or if the OCI client library needs to be re-created.

Action: Check that the LD_LIBRARY_PATH environment variable exists. If it does, re-create the OCI client library.

WTE-01508 Error: Failed to initialize OCI server for database *string*

Cause: The service name of the origin database may not be valid.

Action: In the tnsnames.ora file on the cache node, check the service name of the origin database.

WTE-01509 Error: Failed to connect to server for database *string*

Cause: The user may not be a valid user or the password specified may not be valid.

Action: Refer to the OCI error message to determine the precise reason for the error.

WTE-01512 Error: Unable to get status of the cache

Cause: The client is unable to connect to the cache.

Action: Verify that ORACLE_SID is defined to be "icache" and the user account from which the Oracle8*i* Cache management engine is running is a member of the operating system group that is designated to have DBA privileges.

WTE-01516 Error: Client limit exceeded

Cause: The Oracle8*i* Cache management engine already has reached its maximum number of simultaneous clients.

Action: Exit from some clients, such as Cache Manager, and then reconnect.

WTE-01521 Error: Failed to log on to origin database

Cause: Oracle8*i* Cache failed to log on to the origin database.

Action: Refer to the OCI error message to determine the precise reason for the error.

WTE-01524 Error: Failed to execute OCI statement - *string*

Cause: The OCI statement failed to execute.

Action: Refer to the OCI error message to determine the precise reason for the error.

WTE-01526 Error: Open or create log file, *string*, failed

Cause: Oracle8*i* Cache failed to create a file pointer to the log file wtcme.log.

Action: Check that the user who started the Oracle8*i* Cache management engine has access privileges to the file and the directory that contains the file.

WTE-01529 OCI error - *string*

Cause: The specified OCI error occurred.

Action: Refer to the OCI error message to determine the precise reason for the error.

WTE-01532 OCI returned no data

Cause: OCI was unable to return data for a table.

Action: Refer to the OCI error message to determine the precise reason for the error.

WTE-01534 Error: Failed to open OCI Server connection to origin database: *string*

Cause: The origin database might be stopped. It must be started to complete this operation.

Action: If the origin database is stopped, start the database.

WTE-01535 Error: Failed to connect to the cache (Is the cache stopped?)

Cause: The cache might be stopped. It must be started to complete this operation.

Action: If the cache is stopped, use Cache Manager to start the cache.

WTE-01536 Error: Failed to initialize metadata manager context (Is cache stopped?)

Cause: The cache might be stopped. It must be started to complete this operation. This message is displayed in the log only if the log level is set to debug.

Action: If the cache is stopped, use Cache Manager to start the cache.

WTE-01582 Error: Failed to open a message file for Cache Manager

Cause: The message file is not in the directory $ORACLE_HOME/icache/mesg/.

Action: Check if the message file wte*locale*.msb exists in the directory. If it does not, contact customer support.

WTE-01583 Warning: Failed to use a locale provided by Cache Manager, using default

Cause: The user may have provided an invalid locale. The default locale is being used.

Action: Provide a valid locale setting.

WTE-01584 Error: Failed to connect to origin database

Cause: The origin database might be stopped. It must be started to complete this operation.

Action: If the origin database is stopped, start the database.

WTE-01587 Error: Unable to get table size

Cause: Oracle8*i* Cache was unable to get the table size. This may be caused by a problem during installation.

Action: Contact customer support.

WTE-01592 Error: Cache is not available

Cause: The cache might be stopped. This may be caused by another administrator using Cache Manager to stop the cache. The cache must be started to complete this operation.

Action: If the cache is stopped, use Cache Manager to start the cache.

WTE-01594 Error: Error sending message. Client dropped connection?

Cause: Client may have dropped the connection or lost the connection while a message was being sent to the client. This message is displayed in the log only if the log level is set to debug.

Action: No action is required.

WTE-01602 Error: Insufficient privilege, DBA privileges required

Cause: The operation requires DBA privileges, but the user who logged in does not have DBA privileges.

Action: Log in using the user name and password of a user who has DBA privileges.

WTE-01607 Error: Could not initialize LOB handle

Cause: External procedure capability may not be enabled in the origin database.

Action: Enable external procedure capability in the origin database, then update the user list using Cache Manager.

WTE-01609 Error: Failed to attach to origin database server *string*

Cause: The service name in the tnsname.ora file could be invalid, or the TNS Listener for the origin server is stopped.

Action: Check the service name in the tnsname.ora file on the origin database. If the TNS Listener is stopped, start it.

WTE-01611 Error: Failed to connect to the origin database as user oracache

Cause: The password for the user oracache may have been changed in the origin database, or the data link between the cache and the origin database might be broken.

Action: Contact customer support.

WTE-02506 Origin database service name (AuthenticationServer): *string*

Cause: If the log level is set to debug, the service name of the origin database is written to the log file.

Action: Verify that the service name is correct.

WTE-02507 Listening to TCP port *number*

Cause: If the log level is set to debug, the port number used by the Oracle8*i* Cache management engine is written to the log file.

Action: Verify that the Oracle8*i* Cache management engine is using the correct port number.

WTE-02508 Client connection accepted. Spawning thread *number*

Cause: If the log level is set to debug, the thread number is written to the log file when a new client connects to the Oracle8*i* Cache management engine.

Action: No action is required.

WTE-02509 Closing connection

Cause: If the log level is set to debug, this message is written to the log file when a connection is closed.

Action: No action is required.

WTE-03502 Error updating list of users: Failed to copy export file to cache node

Cause: Oracle8*i* Cache could not copy the export file to the cache node.

Action: Check that the user who started Oracle8*i* Cache has write access to the directory. Check that there is available disk space on the cache node.

WTE-03504 Error updating list of users: Failed to drop users already deleted from origin database

Cause: Oracle8*i* Cache could not drop one or more users from the cache when the list of users was updated. The user or users were already dropped from the origin database. However, the users could not be dropped from the cache because they were connected to the cache.

Action: When the users disconnect from the cache, use Cache Manager to update the list of users.

WTE-03506 Error altering the size of data file *string*

Cause: The size of the data file could not be altered. If you tried to decrease the size of the file and doing so would result in loss of data, Oracle8*i* Cache prevented the decrease in size. If you tried to increase the size of the data file but there was insufficient disk space, Oracle8*i* Cache prevented the operation.

Action: Use Cache Manager to check the amount of unused space in each file before decreasing the size of the file. If insufficient disk space prevents increasing the size of an existing file, add a new data file to a different disk (one that has sufficient disk space).

WTE-03507 Error adding the new data file *string*

Cause: Oracle8*i* Cache could not add a new data file. Possible causes are: the directory does not exist, the Cache Manager user does not have appropriate privileges to create a file in the directory, the file already exists, or there is insufficient disk space.

Action: If the directory does not exist, create the directory. If the user does not have the appropriate privileges to create a file in the directory, grant the user the appropriate operating system privileges. If the file already exists, specify a different file name. If there is insufficient disk space, specify a directory on a different disk or delete unneeded files from the current disk.

WTE-03508 Error getting cache information: Failed to get number of blocks from init.ora file

Cause: The parameter db_block_buffers was not found in the init.ora file.

Action: Check the init.ora file to verify that it contains the parameter db_block_buffers.

WTE-03513 Error setting memory quota: Failed to set number of blocks

Cause: The Cache Manager user may not have the appropriate privileges to modify the init.ora file.

Action: Grant the user DBA privileges.

WTE-03608 Error caching table *string.string*: User *string* does not exist in cache

Cause: The owner of the table was not found in the cache. The owner may have been created since the cache was created or the last time the user list was updated.

Action: Use Cache Manager to update the user list from the origin database.

WTE-03610 Error caching table *string.string*: Table has already been cached

Cause: The table already exists in the cache. Another administrator may have added the table.

Action: Cancel the cache operation, then verify that the table exists in the cache.

WTE-03617 Error removing table *string.string* from cache: Table not in cache

Cause: The table does not exist in the cache. Another administrator may have dropped the table from the cache.

Action: Cancel the operation, then verify that the table exists in the cache before repeating the operation.

WTE-03621 Error synchronizing table *string.string*: Table not in cache

Cause: The table does not exist in the cache. Another administrator may have dropped the table from the cache.

Action: Cancel the operation, then verify that the table exists in the cache before repeating the operation.

WTE-03624 Error altering synchronization policy for table *string*.*string*: Table not in cache

Cause: The table does not exist in the cache. Another administrator may have dropped the table from the cache.

Action: Cancel the operation, then verify that the table exists in the cache before repeating the operation.

Index