Oracle® Database Cache

Concepts and Administration Guide

Release 1.0.2.1

February 2001

Part No. A88706-01

ORACLE®

Oracle Database Cache Concepts and Administration Guide, Release 1.0.2.1

Part No. A88706-01

Contents

3   Managing Oracle Database Cache

4   Tuning Your Oracle Database Cache Environment

5 Troubleshooting Oracle Database Cache

6 Cache Management API

A Oracle Database Cache Views

B Error Messages

Index

Send Us Your Comments

Oracle Database Cache Concepts and Administration Guide, Release 1.0.2.1

Part No. A88706-01

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: 603.897.3825   Attn: Oracle Database Cache Documentation
- Postal service
  Oracle Corporation
  Oracle Database Cache Documentation
  One Oracle Drive
  Nashua, NH 03062-2804
  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

Oracle Database 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, Oracle Database Cache reduces the load on the back-end database and speeds query execution. Oracle Database Cache is ideal for enhancing scalability and performance for Web servers and Web portals, as well as front-office and back-office application servers.

> **Note:**   In previous releases, this product was known as Oracle8*i* Cache. The former name is still reflected in the Quick Tour, in some messages generated by the software, and in some screens in Cache Manager and the Configuration Assistant.

This guide presents the concepts of Oracle Database Cache and describes how to set up and manage your Oracle Database Cache environment.

Intended Audience

This guide is intended for Web server and database administrators who are interested in using Oracle Database Cache to increase the capacity of their applications.

Structure

This guide contains the following chapters:

Related Documents

For more information about Oracle Database Cache, see the following:

- *Oracle Database Cache Readme,* which contains important information about Oracle Database 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 Oracle Database Cache.

- Oracle Database 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

- *PL/SQL User's Guide and Reference*

Conventions

The following conventions are used in this guide:

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

As much as possible, this guide attempts to provide generic documentation, that is, documentation that is not specific to any operating system or interface. Nevertheless, it is sometimes necessary for illustrative purposes to show how the syntax works at the operating-system level. In these cases, this guide uses examples from a UNIX command-line interface and the Windows NT interface.

1

Introducing Oracle Database Cache

Oracle Database Cache improves the scalability and performance of applications that access Oracle databases by caching frequently used data on a middle-tier system. With Oracle Database 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 Oracle Database Cache. Oracle Database Cache is transparent to your end users.

Who Should Use Oracle Database Cache?

If your applications meet the following criteria, you can use Oracle Database 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 Oracle Database Cache?

Oracle Database 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.

Oracle Database 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 Oracle Database 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:
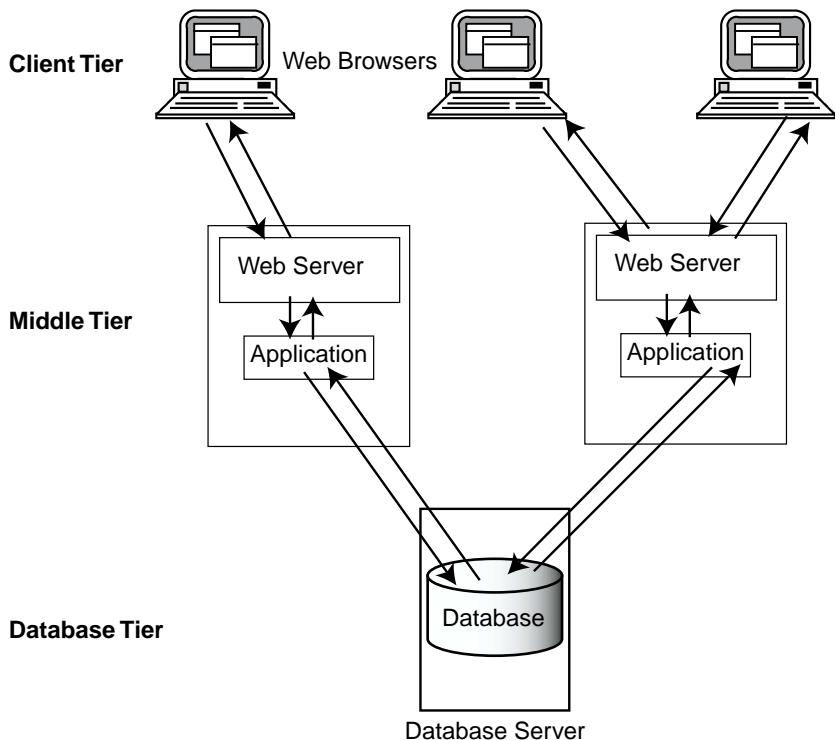
Database Server

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 Oracle Database 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 Oracle Database Cache:
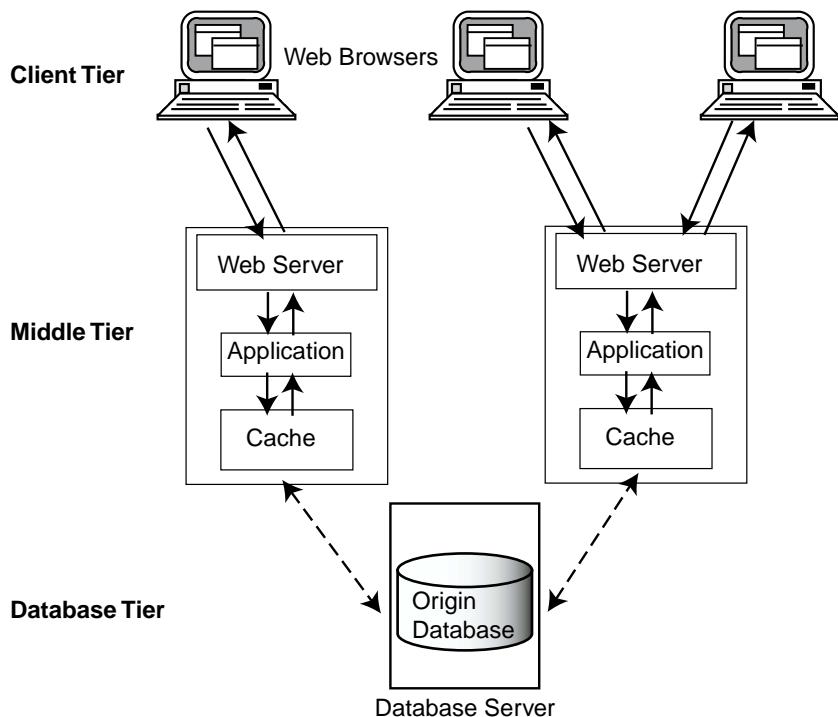
In this environment, Oracle Database Cache software, consisting of a **middle-tier cache** for storing data and intelligent software to route queries, resides on each Web or application server on the middle tier. Frequently accessed data is stored in the middle-tier cache. The **origin database** (on 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 Oracle Database Cache Work?

Oracle Database Cache provides a middle-tier cache for storing data and an easy-to-use management interface for managing the Oracle Database Cache environment. Oracle Database 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 Oracle Database 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.

Oracle Database 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. Oracle Database 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 Oracle Database 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 Oracle Database Cache environment in more detail:
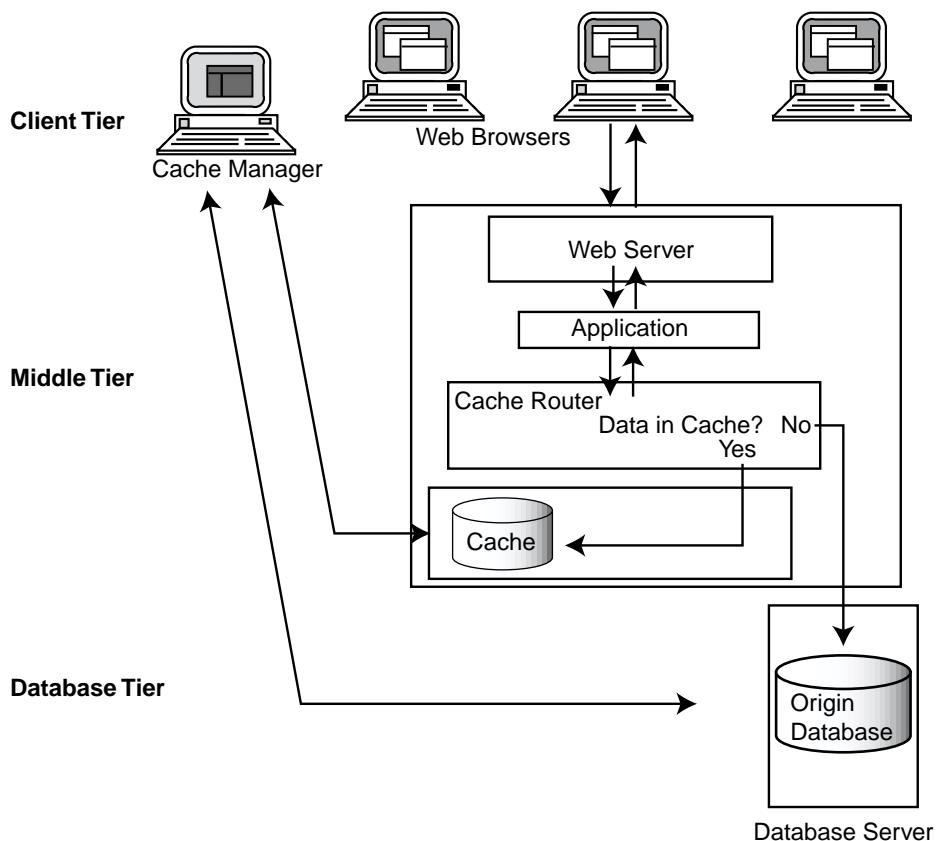
Oracle Database Cache maintains a history of queries and, based on that history, intelligently routes queries either to the middle-tier cache or to the origin database. It also maintains statistics about the queries. These statistics are accessed by all clients attached to a particular middle-tier cache.

When a user requests data, Oracle Database 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, Oracle Database 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, Oracle Database Cache attempts to execute the query using the middle-tier cache. If the query can be satisfied on the middle-tier cache, Oracle Database 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, Oracle Database Cache routes the request to the origin database, which returns the data, through the Web or application server, to the user. In either case, Oracle Database Cache records information about the query.

If the request is an insert, update, or delete, rather than a query, Oracle Database Cache immediately routes the request to the origin database.

If the request is a PL/SQL subprogram (procedure or function) or anonymous block or a call to a PL/SQL stored package, function, or procedure and if routing of PL/SQL is enabled, Oracle Database Cache attempts to execute the request. If Oracle Database Cache encounters an insert, update, or delete request, the request fails and Oracle Database Cache routes the request 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 Oracle Database Cache?

You do not need to modify your applications to take advantage of the benefits of Oracle Database Cache. You only need to configure the environment of your applications, as described in "Configuring Your Application Environment" on page 2-26. 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 5-10 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 Oracle Database Cache. See "Linking Applications That Use Static OCI Libraries" on page 4-1 for more information.

2

Setting Up Oracle Database Cache

To set up Oracle Database 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 and PL/SQL objects (packages, procedures, and functions) to be cached in each, and specify the synchronization policy of each table in each cache.

> **Note:**   Oracle Database Cache provides a cache management API, which is loaded into the cache during installation. The API provides PL/SQL procedures and functions that allow you to use a utility such as SQL*Plus to perform many of the tasks you perform with Cache Manager. For more information about the API, see Chapter 6.
>
> Oracle Database Cache also provides views that allow you to use SQL to access information, such as a list of cached objects, from the cache. For more information about the views, see Appendix A.

Setting Up Oracle Database Cache: General Steps

To set up Oracle Database Cache, you take the following general steps:

1. Design your Oracle Database Cache environment, deciding which data to cache. See "Designing Your Oracle Database Cache Environment" on page 2-2.

2. Install Oracle Database 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 Oracle Database Cache" on page 2-7.

3. Invoke Oracle DBA Studio. See "Invoking Oracle DBA Studio" on page 2-7.

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. In addition, you can specify the PL/SQL packages, procedures, and functions to be cached. (You should cache only read-only PL/SQL objects.) See "Setting Up the Caches" on page 2-13.

5. Enable routing of queries, including PL/SQL subprograms and anonymous blocks, to the cache. See "Enabling Routing to the Cache" on page 2-24.

6. Configure your application environment to use caching. See "Configuring Your Application Environment" on page 2-26.

Designing Your Oracle Database Cache Environment

Before you begin setting up Oracle Database Cache, take time to design your Oracle Database 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 2-5 for information on synchronization policies and your options.

- Do you want to cache any PL/SQL objects (packages, procedures, or functions)? If so, which PL/SQL objects do you want to cache? See "Deciding Which PL/SQL Objects to Cache" on page 2-7.

When you have answered these questions, you will have the necessary information to set up and deploy Oracle Database Cache.

Determining the Data to Be Cached

The first step in designing your Oracle Database 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 Oracle Database Cache routes only read requests to the cache, you get the greatest benefit when most of the requests for the data are read-only.

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.

- 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.

- 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 2-5 for information about synchronizing data.

Now you have identified the data to be cached. Oracle Database 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 Oracle Database 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 Oracle Database 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, in an online news publishing company, 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 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 Oracle Database 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.

Deciding Which PL/SQL Objects to Cache

You can cache PL/SQL objects (packages, procedures, and functions). Consider the following when deciding which, if any, PL/SQL objects to cache:

- You should cache PL/SQL objects that contain only read requests.

  If you cache PL/SQL objects that contains both read-only and read/write statements, calls to these objects may generate inconsistencies. For example, if a PL/SQL procedure contains a query, then an UPDATE statement for the same table, and subsequently another query of the same data, the first query will return data, but the UPDATE statement will fail. Oracle Database Cache will mark the call to the PL/SQL procedure as a miss.

- When you cache a PL/SQL object, you should make sure that you cache any tables or other objects that are referred to by the PL/SQL object. If you do not, the procedure will execute on the origin database, not on the cache.

Installing Oracle Database Cache

You install Oracle Database Cache on all nodes in the middle tier where you want to cache data. When you install Oracle Database Cache, you specify which Oracle database you want to use as the origin database. The installation procedure creates a cache on the Oracle Database Cache node and prepares the origin database for use by Oracle Database 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 Oracle Database 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 Oracle Database Cache environment and provides wizards that take you through the process of setting up and deploying your Oracle Database 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 UNIX systems, such as Sun Solaris, enter the following command:

```
oemapp console
```

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.**

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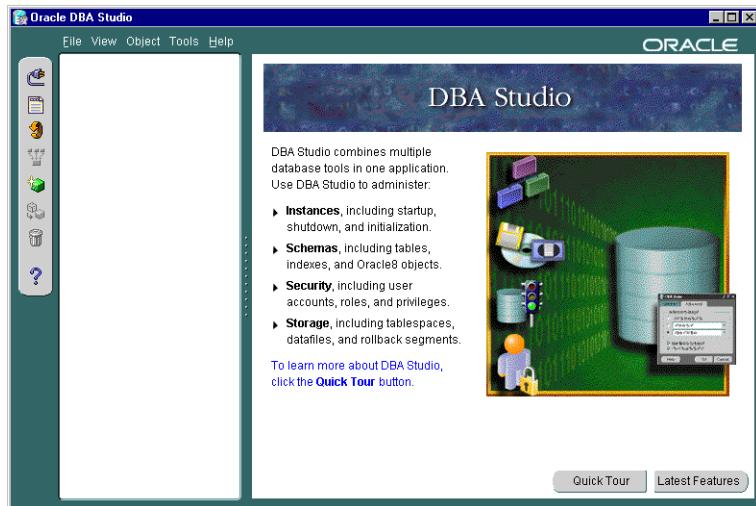
Before you can connect to an origin database, you must add it to the list in the Navigator tree. Take the following steps:

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

2. In the Add Databases to Tree dialog box, select **Add a database manually.**
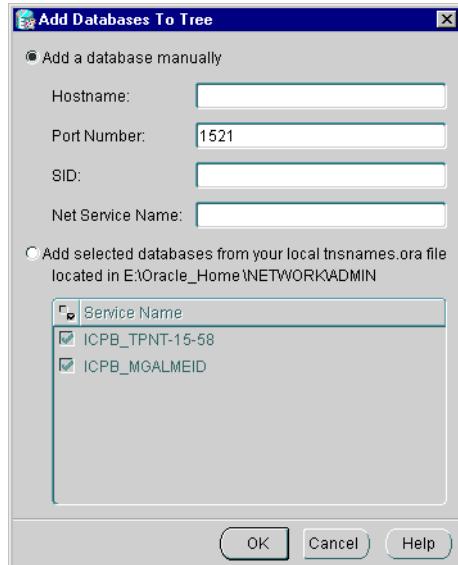
3. In the dialog box, enter the following information:

    ■   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.ora 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.
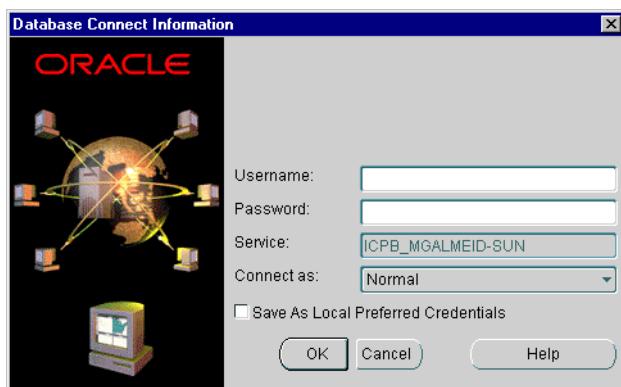
2. Enter the following information:

    ■   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 Oracle Database Cache environment, select the **Caches** folder from the Navigator tree. DBA Studio displays information about Cache Manager and a button to view the Oracle Database Cache Quick Tour.

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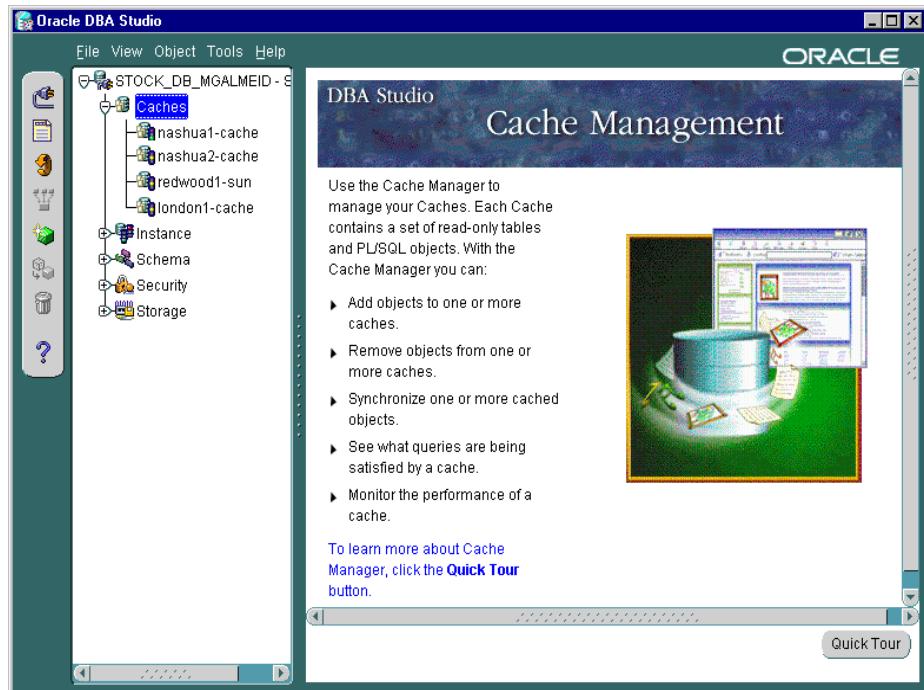
Now, select a cache from the Navigator tree to see information about that cache:
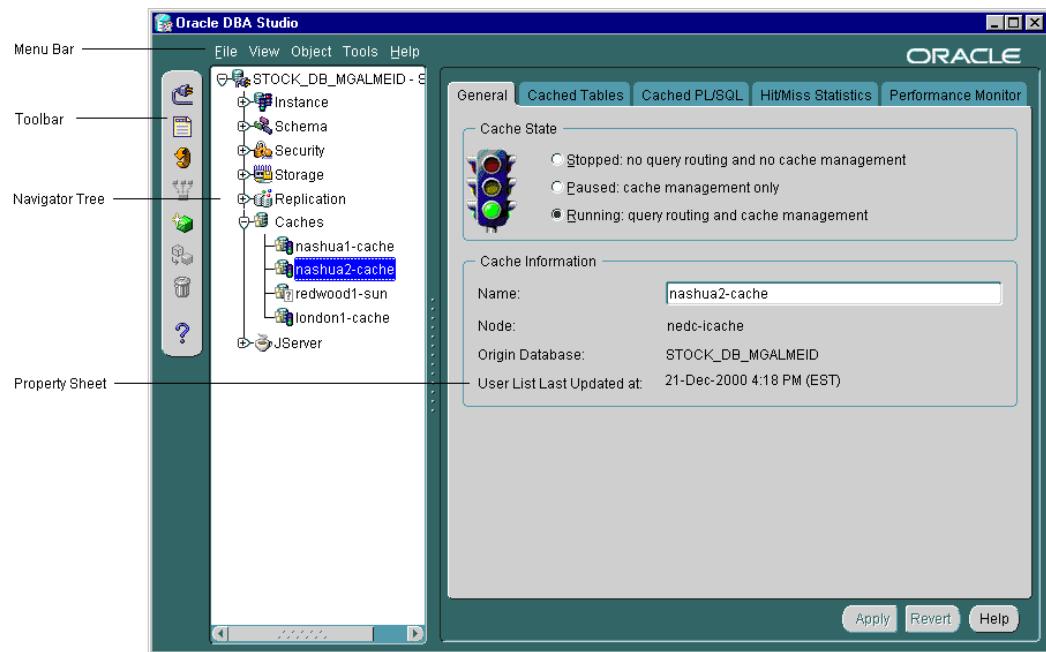
The preceding illustration points out some of the features of DBA Studio and Cache Manager:

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

- 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 Oracle Database 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 Oracle Database Cache. Now, you use Cache Manager to specify which data and PL/SQL objects are 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 Wizard,** which lets you set up and populate one or more caches at the same time. With the wizard, you can add tables and PL/SQL objects to one or more caches and specify the synchronization policy for each table.

When you are setting up your Oracle Database Cache environment for the first time, consider using the **Add 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 Wizard to Set Up Caches

To invoke the **Add Wizard,** select the **Caches** folder from the Navigator tree and choose **Object** -> **Add Objects 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 objects (tables and PL/SQL objects.) This page displays all caches that are configured for the current origin database and that are running or paused.

   To cache objects in particular caches, select the caches in which you want to cache objects 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 objects from the **Selected Caches** list, select the caches and click the left arrow (<).
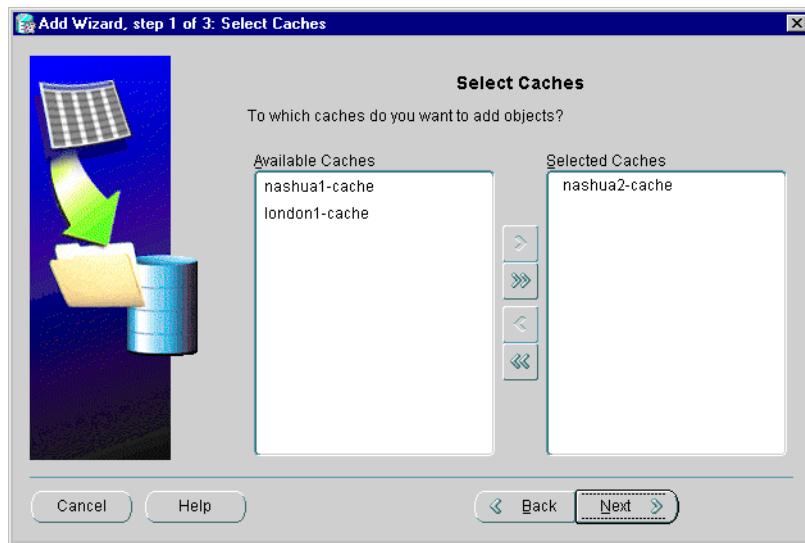
3. Click **Next.**

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 already cached in all selected caches, the table will not be displayed in either list.

   To check the dependencies of a table, from the **Available Tables** list, select the table and click the **Show Dependencies** icon.

   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.
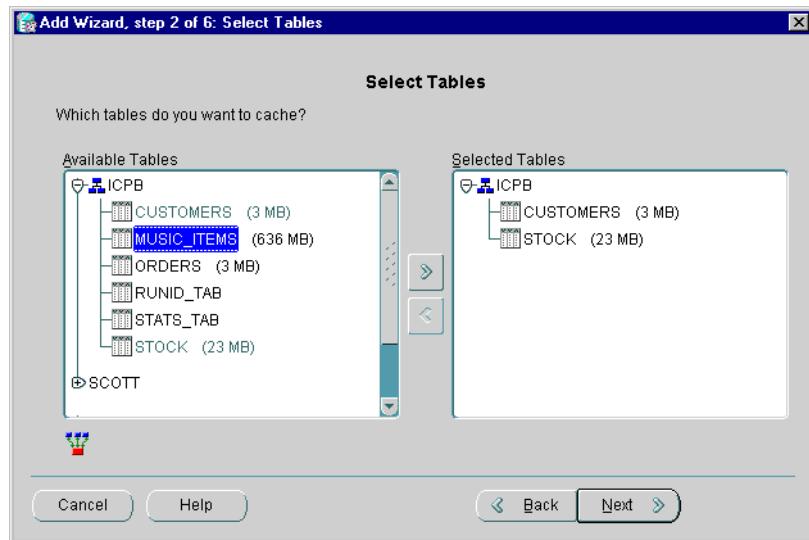
5. When you have selected the tables, click **Next.**

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.

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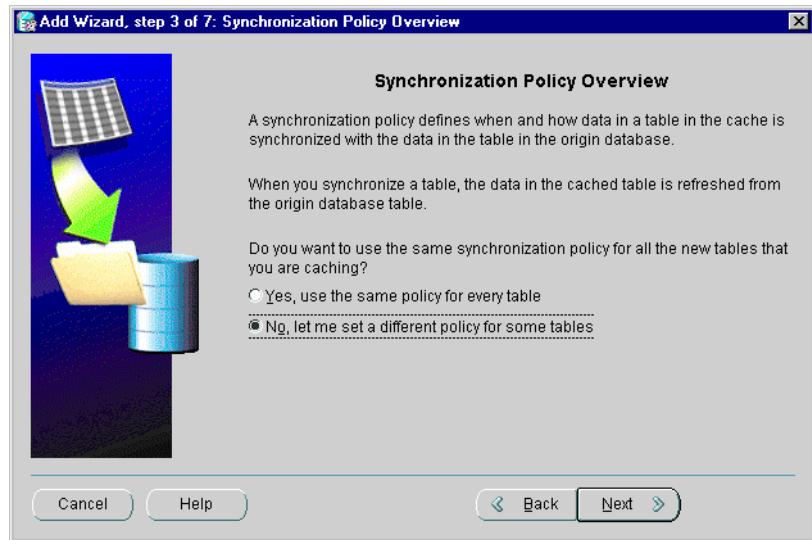
7. Click **Next.**

8. On the Synchronization Policy Method page, you specify the synchronization method 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, Oracle Database Cache deletes all data in the table and replaces it with the current data from the origin database.
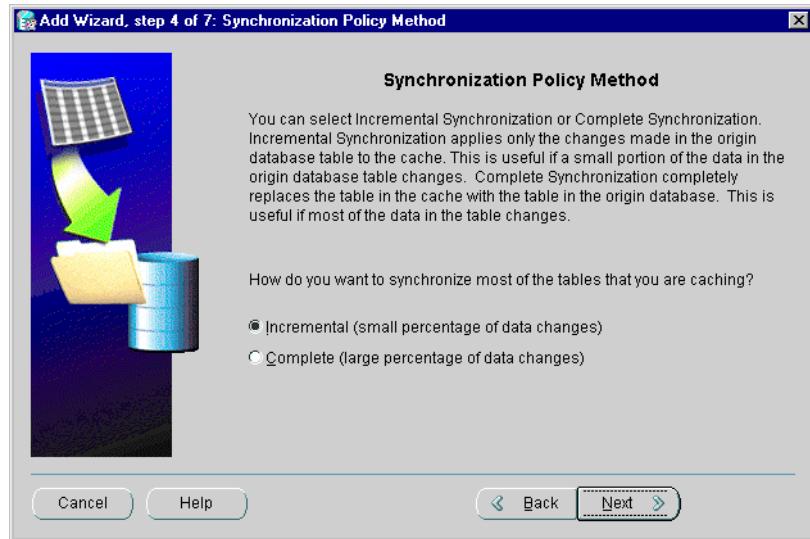
See "Deciding Synchronization Policies" on page 2-5 for more information about synchronization policies.

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.
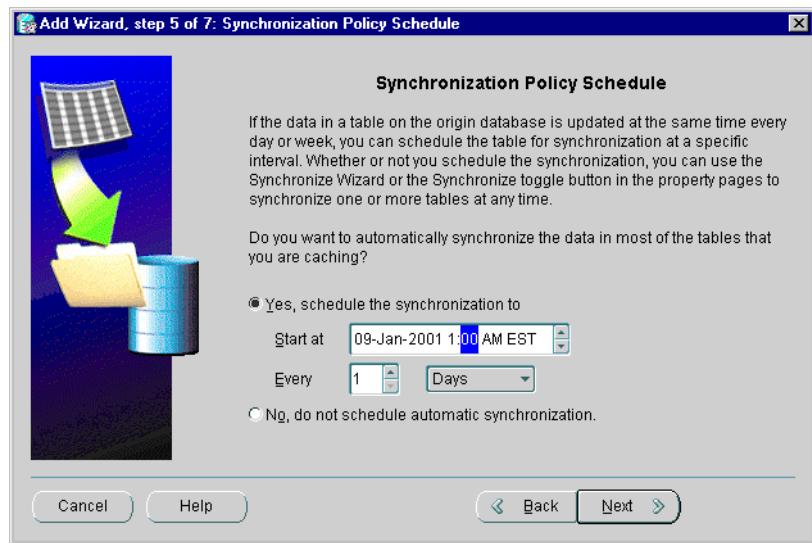
You can choose one of the following:

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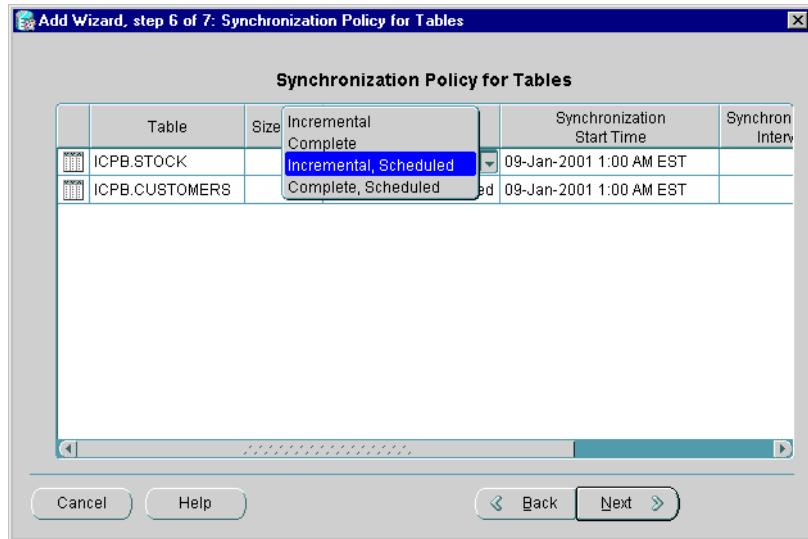
For those tables for which you want a different synchronization policy, take
these steps:

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 Select PL/SQL Objects page lets you select which PL/SQL Objects to cache
    in the selected caches. In the **Available PL/SQL Objects** list, expand the
    schemas, then expand the folders for procedures, functions, or packages. The
    list displays all PL/SQL objects that can be added to the caches you selected. If

a PL/SQL object is cached in all selected caches, the PL/SQL object will not be displayed in either list.

To check the dependencies of a PL/SQL object, from the **Available PL/SQL Objects** list, select the table and click the **Show Dependencies** icon.

From the **Available PL/SQL Objects** list, select the PL/SQL objects that you want to include in the caches. Click the right arrow (>) to move the objects to the **Selected PL/SQL Objects** list.
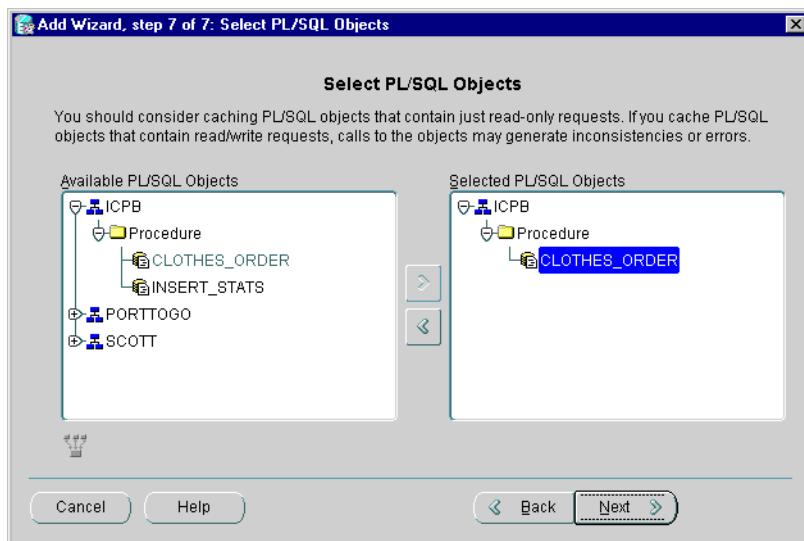
15. When you have selected the PL/SQL objects, click **Next.**

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

Oracle Database Cache adds the tables, their data and their dependent objects, and the PL/SQL 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.

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 3-7 for information on adding or removing tables.
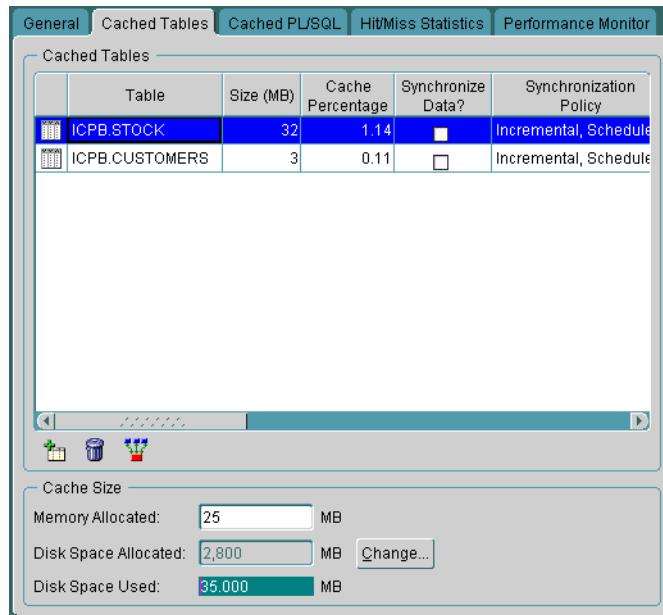
2. On the Cached Tables page, check the synchronization policy and intervals for each table. To change the synchronization policy for a table, click in the **Synchronization Policy** column and select an option.

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.
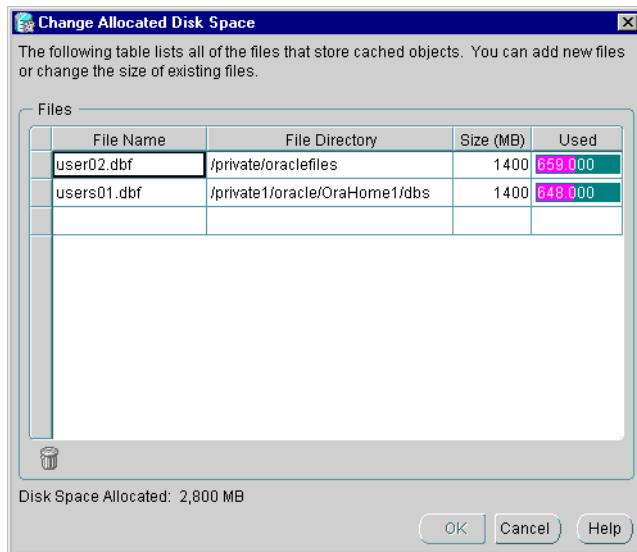
"Modifying the Disk Space Allocated to a Cache" on page 3-14 provides more information about increasing or decreasing the disk space allocated.

5. Click **OK.**

6. On the Cached PL/SQL page, check that the correct PL/SQL objects are cached. To check the dependencies of an object, select the object and click **Show Dependencies**.

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

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

Enabling Routing to the Cache

To enable routing of queries, including PL/SQL subprograms (procedures and functions) and anonymous blocks, to the 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.**

Oracle Database Cache starts the middle-tier cache and enables routing of queries, including PL/SQL subprograms and anonymous blocks, to the cache.

You can disable the routing of PL/SQL subprograms and anonymous blocks by using the DBMS_ICACHE.SET_FLAG procedure. See "SET_FLAG Procedure" on page 6-17 for more information.

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 Objects 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 Objects dialog box:

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

2. In the Copy Cached Objects 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.
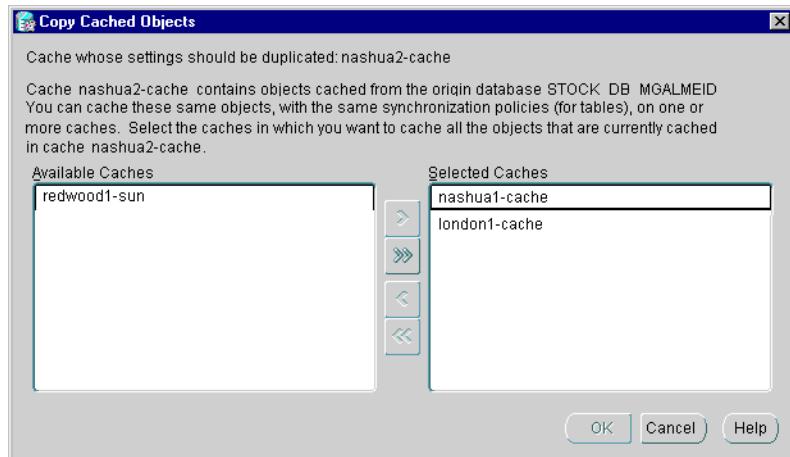
3. Click **OK.**

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 Oracle Database Cache, you must set the environment variable ORA_OCI_CACHE and you must ensure that your applications are using the Oracle home in which Oracle Database 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 Oracle Database Cache.

For example, on UNIX, you use a shell environment variable; on Windows NT, you use an environment variable.

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

If ORA_OCI_CACHE is not set, applications will not use Oracle Database Cache. (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 Oracle Database 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 Oracle Database Cache.

You must also make sure that your applications are using the Oracle home in which Oracle Database Cache is installed, including OCI libraries from that Oracle home. For example, on UNIX, set the value of the environment variable ORACLE_HOME to the Oracle home for Oracle Database Cache. Set the value of the environment variable LD_LIBRARY_PATH so that the Oracle Database Cache library directory ($ORACLE_HOME/lib) precedes other Oracle library directories.

On Windows NT, from the Start menu, select **Oracle for Windows NT** -> **Oracle Home Selector.** Then, select the Oracle home in which you installed Oracle Database Cache. Set the value of the environment variable PATH so that the Oracle Database Cache library directory (*ORACLE_HOME*\lib) precedes other Oracle library directories.

For more information about using the Oracle home and libraries for Oracle Database Cache, see the installation guide.

3

Managing Oracle Database Cache

To manage your Oracle Database 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 Routing to the Cache

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 Oracle Database 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 routing 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, including PL/SQL subprograms and anonymous blocks, are routed to the origin database. If a cache is stopped, you can only view general information about it. You cannot add or remove objects 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 objects 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 | Cached PL/SQL | 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: nashua2-cache

Node: nedc-icache

Origin Database: STOCK_DB_MGALMEID

User List Last Updated at: 21-Dec-2000 4:18 PM (EST)
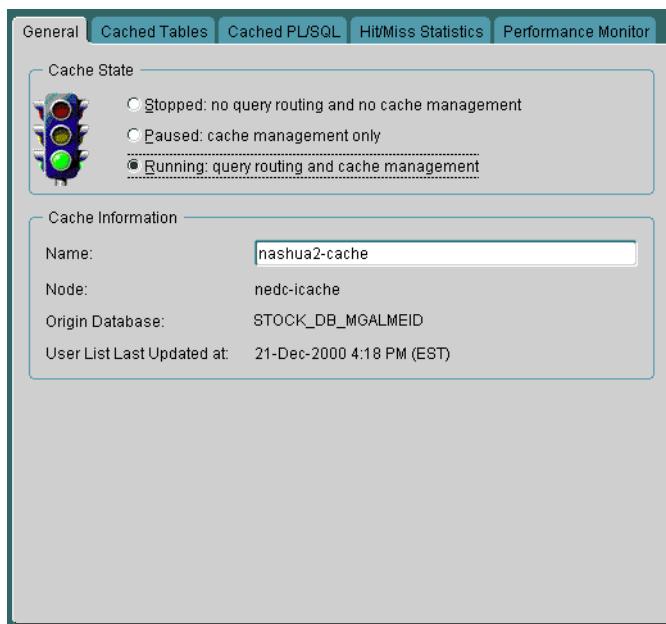
Select an option to change the state of the cache:

- To start a stopped cache, click **Running.** Oracle Database Cache starts the selected cache and enables routing of queries, including PL/SQL subprograms and anonymous blocks, to the cache.

- To disable the routing of queries to the cache, click **Paused.** Oracle Database 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.** Oracle Database Cache stops the selected cache and routes queries directly to the origin database.

When you click **Apply,** Oracle Database 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.

You can also use the cache management API to enable and disable routing. To enable or disable routing of queries, you can use the DBMS_SET_ROUTING_STATE procedure. For more information, see "SET_ROUTING_STATE Procedure" on page 6-20.

To enable or disable routing of PL/SQL subprograms (procedures and functions) and anonymous blocks to the cache, you can use the DBMS_ICACHE.SET_FLAG procedure. See "SET_FLAG Procedure" on page 6-17 for more information.

To check whether the routing of PL/SQL subprograms is enabled or disabled, you can use the DBMS_ICACHE.GET_FLAG function. See "GET_FLAG Function" on page 6-8 for more information about this function.

Monitoring Your Oracle Database Cache Environment

Cache Manager helps you to monitor your Oracle Database 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 Oracle Database 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, Oracle Database 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, Oracle Database Cache routes the query to the origin database and marks the query as a miss. If the query can be satisfied in the cache, Oracle Database 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, by default, the statistics are gathered once every 5 minutes.

You can also use the DBA_ICACHE_QUERY_STATS view, provided by Oracle Database Cache, to retrieve information about hit and miss statistics. For more information, see Appendix A.

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.

You can also reset the statistics using the DBMS_ICACHE.RESET_STATS procedure. See "RESET_STATS Procedure" on page 6-16 for more information.

Monitoring Performance

To view other performance information about the cache, select the **Performance Monitor** tab.
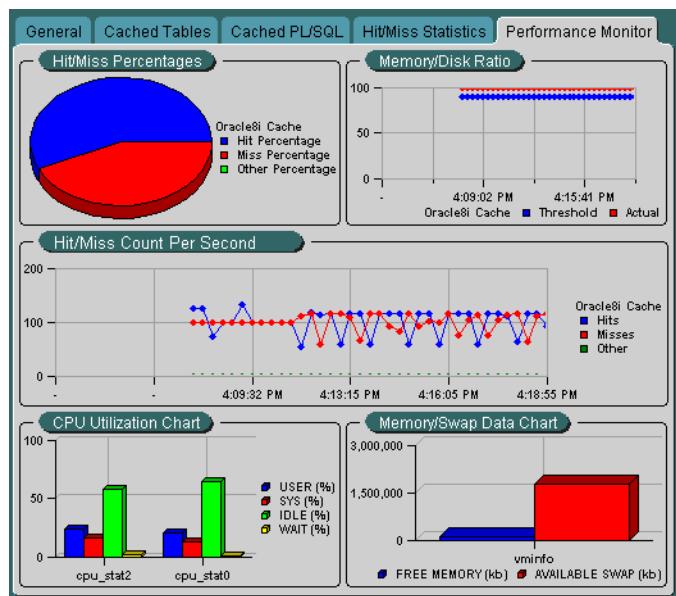
- Hit/Miss Percentages Chart: The percentage of hits and misses for a cache

  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 Oracle Database 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 Percentages 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.

You can also monitor performance using the DBA_ICACHE_AGGREGATE_STATS view, provided by Oracle Database Cache. For more information, see Appendix A.

Modifying Which Data Is Cached

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

> **Note:**   When you add a table to or remove a table from a cache, Oracle Database Cache changes the status of all queries that were marked as misses to "Unknown." When Oracle Database 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, Oracle Database 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 Tables to Cache** icon.

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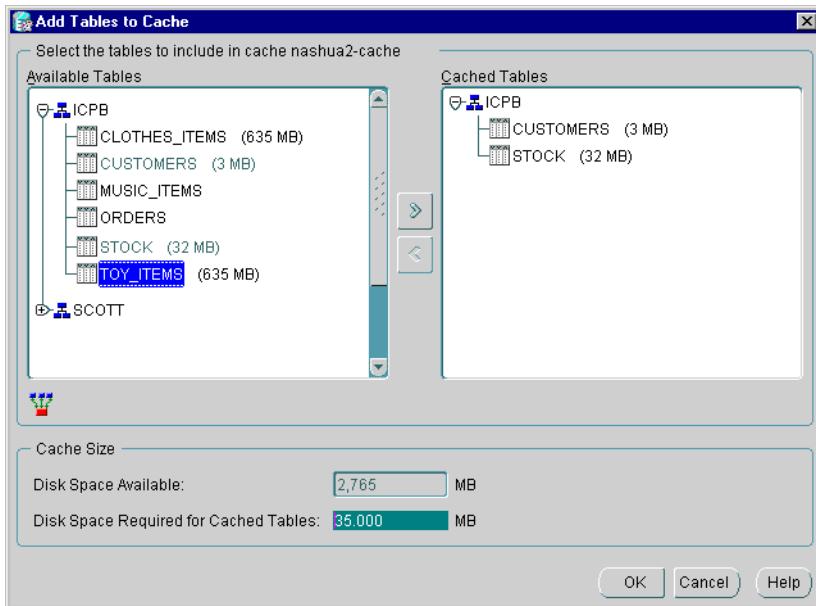
4.   To add a table to the cache, expand the schemas and select the table from the **Available Tables** list. Click the right arrow (>) to move the table to the **Cached Tables** list.

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.

5. Click **OK.**

6. Click **Apply.**

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

Adding PL/SQL Objects to One Cache

You can add PL/SQL objects 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 PL/SQL** tab and then click the **Add PL/SQL Objects to Cache** icon.

3. Cache Manager displays the Add PL/SQL Objects to Cache dialog box.
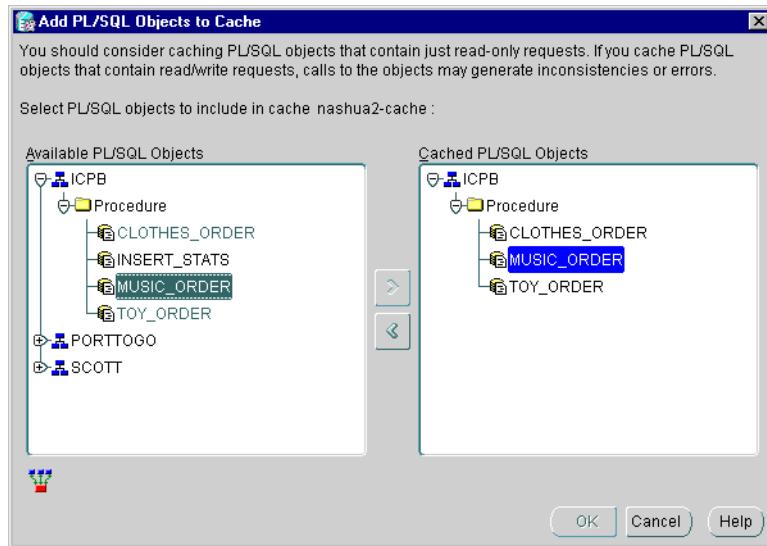
To add a PL/SQL object to the cache, expand the schemas and folders in the **Available PL/SQL Objects** list. Then, select the PL/SQL object. Click the right arrow (>) to move the PL/SQL object to the **Cached PL/SQL Objects** list.

You can also remove PL/SQL object using this dialog box by selecting the PL/SQL object from the **Cached PL/SQL Objects** list. Click the left arrow (<) to move the object to the **Available PL/SQL Objects** list.

4. Click **OK.**

5. Click **Apply.**

Adding Tables and PL/SQL Objects to One or More Caches

To add tables or PL/SQL objects (packages, procedures, and functions) to one or more caches at the same time, use the **Add Wizard.** With the wizard, you can add tables to one or more caches and specify the synchronization policy for each table and add PL/SQL objects to one or more caches.

To invoke the **Add Wizard,** select the **Caches** folder from the Navigator tree and choose **Object** -> **Add Objects to Caches.** Then, take the steps outlined in "Using the Add 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. Select the **Cached Tables** tab. Then, select the table and click the **Remove Cached Tables** icon.

3. Click **Apply.**

Removing PL/SQL Objects from One Cache

You can remove PL/SQL objects from 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 PL/SQL** tab. Then, select the PL/SQL object and click the **Remove Cached PL/SQL Objects** icon.

3. Click **Apply.**

Removing Tables and PL/SQL Objects from One or More Caches

To remove tables and PL/SQL objects from one or more caches, you can use the **Remove Wizard.** Take the following steps:

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

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

3. The Select Caches page shows the caches from which you can remove objects. 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 objects.
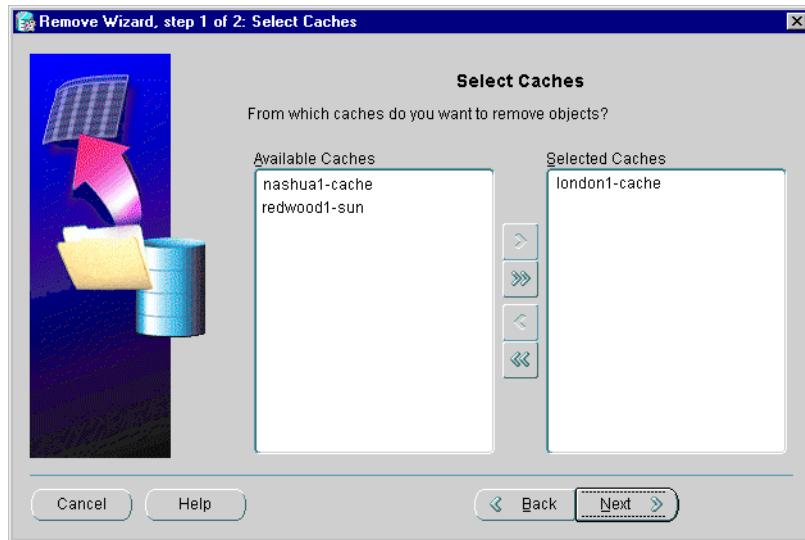
4. Click **Next.**

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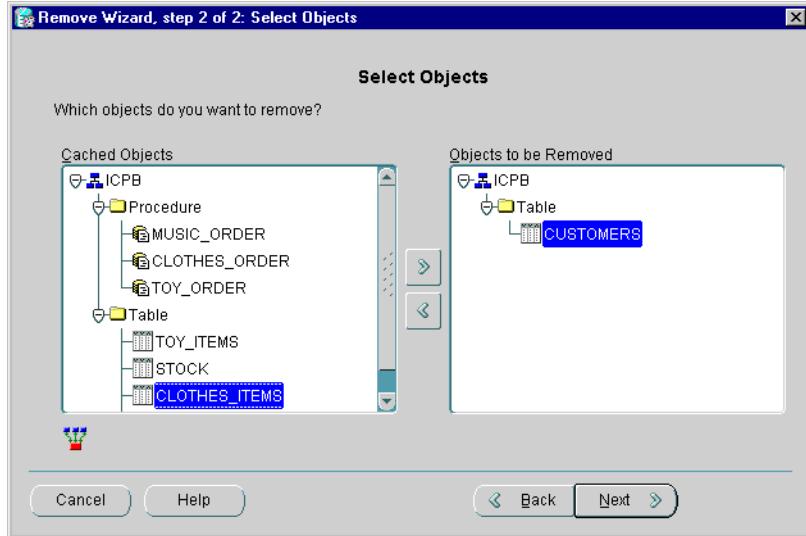
6. Click **Next.**

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

The wizard displays a progress box. When it completes the operation, the selected objects 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. Select the **Cached Tables** tab, and then enter the new amount in the **Memory Allocated** box.
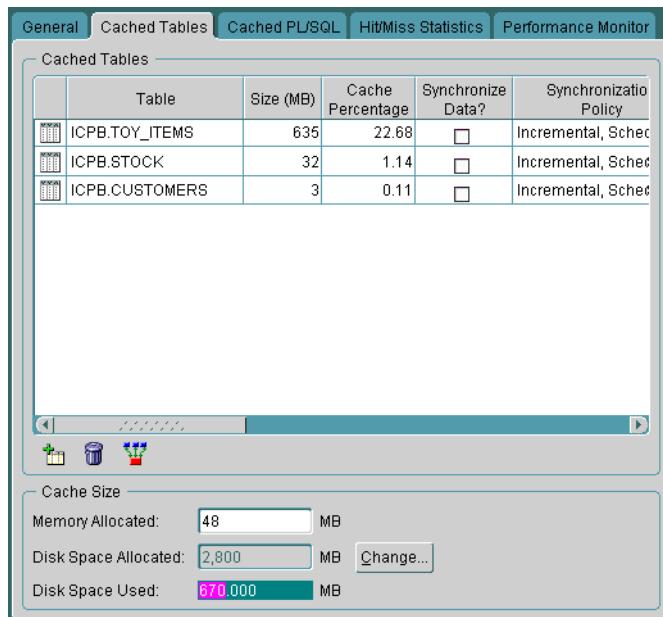
3. Click **Apply.**

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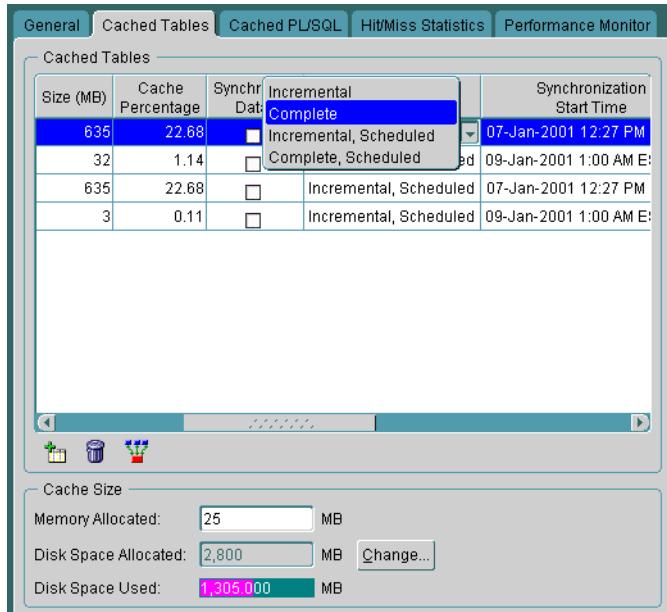
4. If you selected Incremental, Scheduled, or Complete, Scheduled, select a **Synchronization Start Time.** Then, enter a **Synchronization Interval** and select a **Synchronization Unit,** such as minutes, hours, days, or weeks.

   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 page.

- You can synchronize tables in more than one cache by using the Synchronize 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. Select the **Cached Tables** tab. Then, click the check box in the **Synchronize Data?** column for the table or tables you want to synchronize.
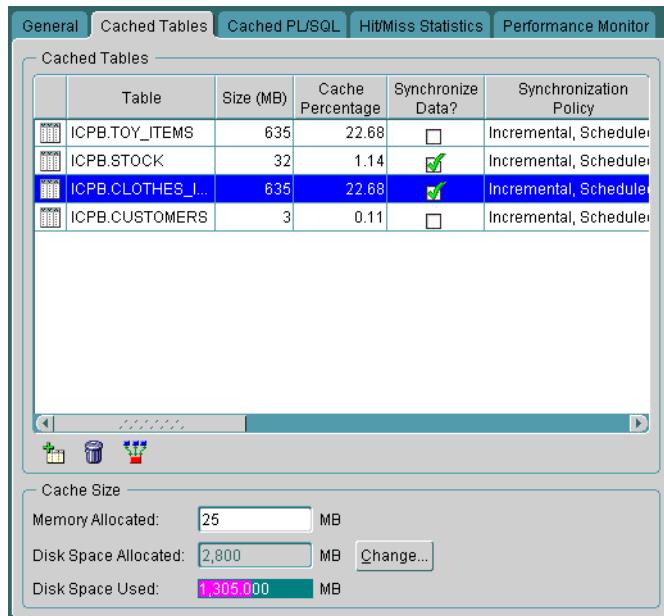
3. Click **Apply.**

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 Wizard.** The wizard synchronizes 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 Wizard** performs a Complete synchronization on the table immediately. Similarly, if the synchronization policy of a table is Incremental, the **Synchronize 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 Objects.**

2. Cache Manager displays the **Synchronize 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 Objects 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 Objects** list, expand the schemas and the Table folder. Then, select the tables that you want to synchronize. Click the right arrow (>) to move the tables to the **Objects to Be Synchronized** list.
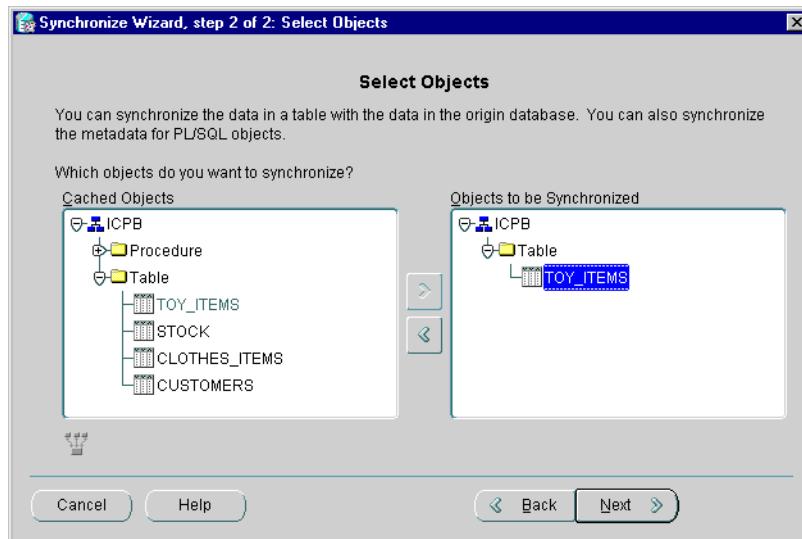
On this page, you can also select PL/SQL objects to be synchronized. See "Synchronizing PL/SQL Objects" on page 3-21 for information about synchronizing the metadata of PL/SQL Objects.

6. Click **Next.**

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

Oracle Database 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), Oracle Database Cache immediately performs a complete synchronization. If a table has a synchronization policy of Incremental (whether or not it is Scheduled), Oracle Database 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.

Synchronizing PL/SQL Objects

If you update the definition (the metadata) of a PL/SQL object (a package, procedure, or function) in the origin database, you should update the definition in the cache by synchronizing the PL/SQL object in the cache with the PL/SQL object in the origin database.

Note that you cannot schedule the synchronization of a PL/SQL object.

Synchronizing PL/SQL Objects in One Cache

To initiate the synchronization of some or all of the PL/SQL objects in one cache, take the following steps:

1. Select the cache from the Navigator tree.

2. Select the **Cached PL/SQL** tab. Then, click the check box in the **Synchronize?** column for the PL/SQL object or objects you want to synchronize.
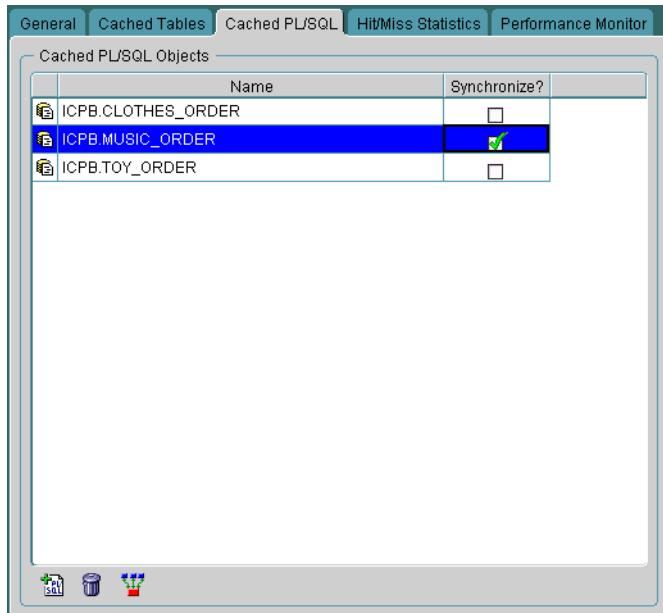
3. Click **Apply.**

Oracle Database Cache synchronizes the metadata of the PL/SQL objects in the cache with the metadata of the PL/SQL objects in the origin database.

Synchronizing PL/SQL Objects in One or More Caches

To initiate the synchronization of some of all of your PL/SQL objects in one or more caches, you use the Synchronize Wizard. Take the following steps:

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

2. Cache Manager displays the **Synchronize 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 PL/SQL objects 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 Objects page shows the tables and PL/SQL objects that are cached in one or more of the selected caches. In the **Cached Objects** list, expand the schemas and the Procedure, Function, or Package folders. Then, select the PL/SQL objects that you want to synchronize. Click the right arrow (>) to move PL/SQL objects to the **Objects to Be Synchronized** list.
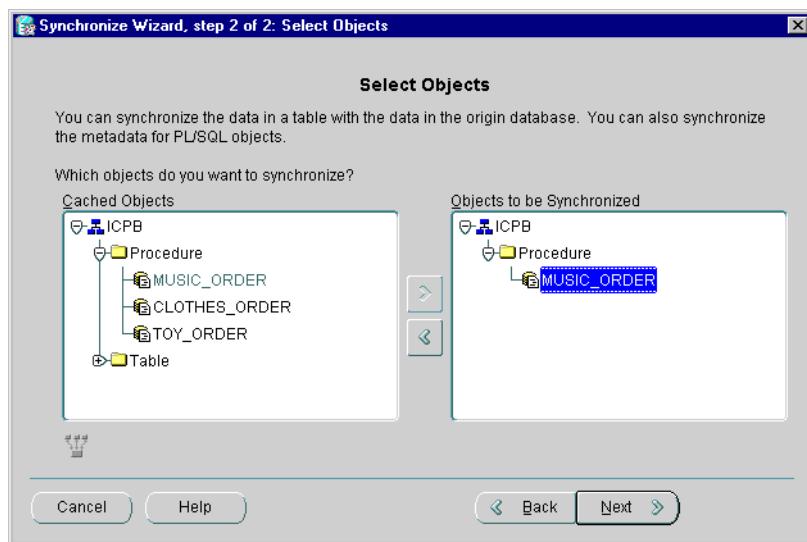
On this page, you can also select tables to be synchronized. See "Synchronizing Data" on page 3-18 for information about synchronizing the data in tables.

6. Click **Next.**

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

Oracle Database Cache immediately synchronizes the selected PL/SQL objects in the selected caches.

Updating Users

When you install Oracle Database 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 objects 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 objects 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 Oracle Database Cache.

However, you must use the Oracle Database 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 Oracle Database Cache

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

The Configuration Assistant file is named wtacca and is located in the bin subdirectory of the Oracle home. To modify a cache, invoke the Configuration Assistant by using the following command on the cache node:

```
$ wtacca
```

The Configuration Assistant displays the Welcome page. To modify your cache, click **Next** and take the following steps:

1. On the 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.
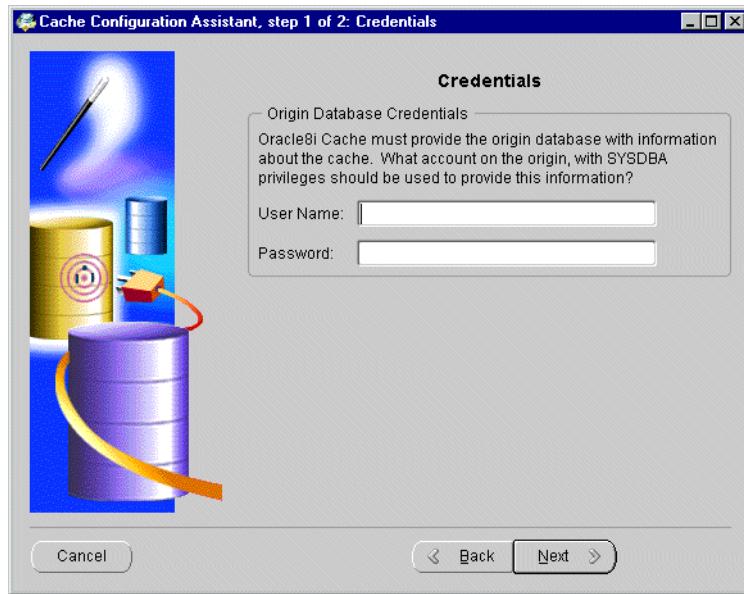
If you enter a user name that is not valid or that does not have SYSDBA privileges, or if you enter an invalid password, the Configuration Assistant returns an error and allows you to enter another value.

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 Oracle Database 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 (').
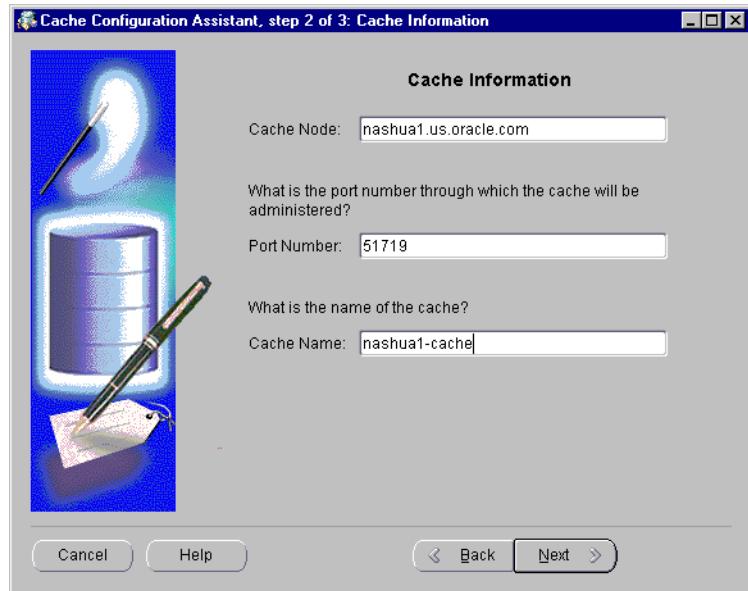
4. Click **Next.**

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 Oracle Database Cache Environment

In most cases, after you use Cache Manager to set up your Oracle Database 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 Oracle Database 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 Oracle Database 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 Oracle Database 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 Oracle Database Cache middle-tier cache. (See "Configuring Your Application Environment" on page 2-26 for information about this environment variable.) If ORA_OCI_CACHE is not set, applications will not use Oracle Database Cache.

If you want to exercise greater control over which applications use Oracle Database 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 Oracle Database 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 Oracle Database 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 Oracle Database Cache

You can control which statements within an application use Oracle Database 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 Oracle Database 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 Oracle Database Cache

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

Logging Errors

To help you troubleshoot problems in your environment, Oracle Database Cache writes error messages to a log file or an event viewer.

By default, on UNIX, the file specification of the log file is:

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

On Windows NT, Oracle Database Cache writes error messages to the Event Viewer. From the Windows NT Start menu, select **Programs** -> **Administrative Tools** -> **Event Viewer.** From the **Log** menu of the Event Viewer, choose **Application.** Then, select events named OracleWtcIcache.*sid,* where *sid* is the Oracle system identifier (SID) of the cache.

If error messages cannot be written to the event log, errors are written to the log file:

```
ORACLE_HOME\admin\icache\bdump\wtcme.log
```

Note that the contents of the log file are always written using the UTF8 character set. If the log file contains non-ASCII characters, you may need to use a text viewer that can display characters in the UTF8 character set.

You can change the level of errors written to the log file and, on UNIX, the name and location of 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:

- On UNIX: $ORACLE_HOME/admin/icache/pfile/wtcme.ora

- On Windows NT: *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.
port	The port number of the listener for Oracle Database Cache. The listener establishes a communication path between Oracle Database 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 3-25.
	By default, the port number is 51719.
logfile	The full file specification for the log file. By default, the value is set to the following:
	■ On UNIX:
	$ORACLE_HOME/admin/icache/bdump/wtcme.log
	■ On Windows NT:
	ORACLE_HOME\admin\icache\bdump\wtcme.log

Parameter	Description
loglevel	The level of error messages written to the log file wtcme.log or to the Windows NT Event Viewer.
	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 Oracle Database Cache management engine are written to the file.
	Because the volume of messages written to the log file or event viewer is very great, use this value only when you are investigating a problem and the other values do not provide enough information.

The following command specifies a log level of debug:

```
wtcme loglevel=debug
```

Important Cautions

The following describes some important information that can help you to prevent problems and fix problems if they occur:

■ Oracle Database 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, Oracle Database 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, Oracle Database 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 Oracle Database Cache and related software is started. See "Starting Oracle Database Cache and Related Software" on page 5-8 for more information.

- Do not modify or delete the entries that Oracle Database Cache places in the tnsnames.ora file. If you have modified or deleted the entries, see "Restoring tnsnames.ora Entries" on page 5-5 for information on creating the entries again.

- Because Oracle Database 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 5-10 for more information.

Port Number Conflicts

If the port number used by Oracle Database Cache is the same number as an existing port, the Configuration Assistant does not return an error during installation. It writes a message to the installation log indicating that the listener is already started.

However, when you attempt to access the cache using the listener (as when clients are routed to the cache), you receive the following error:

```
ORA-12541 - TNS: no listener
```

To resolve the problem, edit the listener.ora and tnsnames.ora files and specify a port number that is not used.

Restoring tnsnames.ora Entries

If you modified or deleted the entries that Oracle Database Cache placed in the tnsnames.ora file on the cache node, Oracle Database Cache will not operate correctly. Edit the tnsnames.ora file to match the following entries:

- The Oracle Database Cache management engine communicates with the cache through the ora_icache entry, shown in the following excerpt:

```
# Name for the cache.
ora_icache =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = <your_dbSid>))
    )
    (CONNECT_DATA = (SERVICE_NAME = <HostName-GlobaldbName>)
    )
  )
```

- The Oracle Database Cache management engine communicates with the origin database through the ora_icache_origin entry, shown in the following excerpt:

```
# Oracle Database Cache's origin database
ora_icache_origin =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)
                 (HOST = <originHostName>)
                 (PORT = <originPortNumber>))
    )
    (CONNECT_DATA =(SERVICE_NAME = <originServiceName>)
    )
  )
```

- The following entry supports external procedures within Oracle Database Cache:

```
# Support for external procedures within Oracle Database Cache
extproc_connection_data =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = ipc)(KEY = <your_dbSid>)
    )
    (CONNECT_DATA = (SID = iCache_extproc) (SERVER = DEDICATED)
    )
  )
```

- The following entry provides a descriptive name to connect to the cache from the Oracle Enterprise Manager Performance Monitor:

```
<HostName-GlobaldbName> =
(DESCRIPTION =
  (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC) (KEY = <your_dbSid>))
      )
      (CONNECT_DATA = (SERVICE_NAME = <HostName-GlobaldbName>)
      )
)
```

- The following entries support Oracle Servlets for Java:

```
# Support for mod_ose over normal TCP connections
inst1_http =
  (DESCRIPTION =
    (ADDRESS =
      (PROTOCOL = TCP)
      (HOST = <hostName>)
      (PORT = 1521)
    )
    (CONNECT_DATA =
      (SERVICE_NAME = MODOSE)
      (SERVER = shared)
      (PRESENTATION = http://admin)
    )
  )

# To support mod_ose over TCP with SSL connections, remove the comments
# from the following entry:
#inst1_https =
#  (DESCRIPTION =
#   (ADDRESS =
#      (PROTOCOL = TCPS)
#      (HOST = %s_hostName%)
#      (PORT = 2484)
#    )
#   (CONNECT_DATA =
#      (SERVICE_NAME = MODOSE)
#      (SERVER = shared)
#      (PRESENTATION = http://admin)
#   )
#  )
```

Restoring listener.ora Entries

If you modified or deleted the entries that Oracle Database Cache placed in the listener.ora file on the cache node, Oracle Database Cache will not operate correctly. Edit the listener.ora file to match the following entries:

```
CONNECT_TIMEOUT_LISTENER = 0
LOGGING_LISTENER = OFF

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =                    # Local connections
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY = <your_dbSid>))
      )
    )
    (DESCRIPTION =                    # General TCP connections
      (ADDRESS =
          (PROTOCOL = TCP)(HOST = <hostName>)(PORT = 1521))
    )

# For secure connections over SSL, remove the comment characters
# from the following entry:
#    (DESCRIPTION =                   # Secure TCP connections
#      (ADDRESS =
#          (PROTOCOL = TCPS)(HOST = <hostName>)(PORT = 2484))
#    )

    (DESCRIPTION =                    # IIOP connections
      (PROTOCOL_STACK = (PRESENTATION = GIOP) (SESSION = RAW))
      (ADDRESS = (PROTOCOL = TCP)(HOST = <hostName>)(PORT = 2481))
    )
# For secure IIOP connections over SSL, remove the comment characters
# from the following entry:
#    (DESCRIPTION =                   # Secure IIOP connections
#      (PROTOCOL_STACK = (PRESENTATION = GIOP)(SESSION = RAW))
#      (ADDRESS = (PROTOCOL = TCPS)(HOST = <hostName>)(PORT = 2482))
#    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = <hostName>-<globalDBName>)
      (ORACLE_HOME = <ORACLE_HOME>)
```

```
       (SID_NAME = <your_dbSid>)
     )
     (SID_DESC =
       (ORACLE_HOME = <ORACLE_HOME>)
       (SID_NAME = iCache_extproc)
       (PROGRAM = extproc)
     )
   )
```

Starting Oracle Database Cache and Related Software

To use all features of the Oracle Database Cache product, Oracle Database Cache, the Oracle Database Cache management engine, Oracle Data Gatherer, and TNS Listener must be started.

To use Cache Manager to manage your caches, the Oracle Database 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 Oracle Database Cache or some functionality of the product.

On UNIX systems, if you followed the procedures for automating startup and shutdown in the installation guide, Oracle Database Cache (as well as the Oracle Database Cache management engine, Oracle Data Gatherer, and TNS Listener) is started when your system starts.

On Windows NT systems, Oracle Database Cache, the Oracle Database Cache management engine, and TNS Listener are services that are started automatically when your system starts.

On UNIX, if Oracle Database Cache is stopped, you can restart it, the Oracle Database 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
```

On UNIX, to stop Oracle Database 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 Oracle Database Cache was installed.

On Windows NT, if Oracle Database Cache is stopped, you can restart it by starting the service:

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

2. Select **OracleServiceicache** and click **Start.**

On Windows NT, you must start the Oracle Database Cache management engine separately:

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

2. Select **OracleWTCiCache** and click **Start.**

Oracle Database Cache uses the user name and password of the Windows NT user under which Oracle Database Cache was installed to start the service for the Oracle Database Cache management engine. The user listed in the Windows NT Services dialog box must be a member of the Administrators group and have the user right "Log on as a service."

To start the Data Gatherer and the TNS Listener, see the following sections.

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 UNIX, 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 **Oracle*home_name*DataGatherer** and click **Start.**

Oracle Database Cache uses the user name and password of the Windows NT user under which Oracle Database Cache was installed to start the service for Oracle Data Gatherer. The user listed in the Windows NT Services dialog box must be a member of the Administrators group and have the user right "Log on as a service."

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 UNIX, 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 **Oracle*home_name*TNSListener** and click **Start.**

Applications Must Be Linked in Multithreaded Mode

Oracle Database 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 UNIX, you use the -lthread option in the link line.

On Windows NT, you link with the msvcrt.lib library.

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

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

6

Cache Management API

Oracle Database Cache provides a cache management API that allows you to manage caches using scripts or a utility such as SQL*Plus. The API consists of a PL/SQL package, DBMS_ICACHE, that is loaded into the cache during installation. This package provides the same functionality as that provided by Cache Manager. In addition, it allows you to:

- Disable routing of PL/SQL subprograms (procedures and functions) and anonymous blocks without disabling routing of standard SQL queries.

- Set intervals for statistics gathering and enable and disable statistics gathering

The package is installed in the SYS schema during the Oracle Database Cache installation.

The following tables provide brief descriptions of each PL/SQL procedure and function. Full descriptions of each procedure and function are provided later in this chapter.

The following table lists the PL/SQL functions and procedures that you can use to pause and restart caching, enable routing of PL/SQL procedures, enable gathering of statistics, control the interval at which statistics are gathered, and reset statistics:

Name	Description
GET_FLAG	Indicates whether or not PL/SQL is routed to the cache or whether or not statistics are being collected.
GET_ROUTING_STATE	Indicates whether the cache is running or paused.
GET_STATS_INTERVAL	Returns the interval at which the specified type of statistics is collected.
GET_STATS_RESET_TIME	Returns the date and time that the specified statistics were last reset.

Name	Description
RESET_STATS	Resets the specified type of statistics to zero.
SET_FLAG	Enables or disables routing of PL/SQL calls to the cache or enables or disables the gathering of statistics.
SET_ROUTING_STATE	Starts a paused cache or pauses a running cache.
SET_STATS_INTERVAL	Sets the interval for the specified type of statistics collection.

The following table lists the PL/SQL functions and procedures that you can use to add objects to or remove objects from the cache. The objects include tables and PL/SQL packages, procedures, and functions:

Name	Description
ADD_PLSQL_OBJECT	Adds the named PL/SQL object (package, procedure, or function) to the cache.
ADD_TABLE	Adds the named table, its data, and its dependent objects to the cache and sets the synchronization policy of the table.
GET_TABLE_SIZE	Calculates the amount of disk space required to cache the named table.
REMOVE_PLSQL_OBJECT	Removes the named PL/SQL object from the cache.
REMOVE_TABLE	Removes the named table, its data, and its dependent objects from the cache.
SET_ORIGIN_CREDENTIALS	Sets the credentials to be used on the origin database.
SET_SYNC_METHOD	Modifies the synchronization method of the named table.
SET_SYNC_SCHEDULE	Modifies the synchronization schedule of the named table.
SYNC_PLSQL_OBJECT	Synchronizes the named PL/SQL object with the PL/SQL object on the origin database.
SYNC_TABLE	Synchronizes the named table with the table on the origin database.
SYNC_USERS	Synchronizes the list of users on the cache with the list of users on the origin database.

Privileges Required

To use the PL/SQL package, you connect to the cache as a user with DBA privileges. Some procedures, such as DBMS_ICACHE.ADD_TABLE, require access to the origin database. Before you call those procedures, you must call the DBMS_ICACHE.SET_ORIGIN_CREDENTIALS procedure. See "SET_ORIGIN_CREDENTIALS Procedure" on page 6-18 for more information.

Errors

The PL/SQL procedures and functions in this package raise the application error ORA-20001, following by the number and text of an Oracle Database Cache error message. For example, if you attempt to remove a table from the cache, but the table is not cached, you receive the following error:

```
ORA-20001: WTE-03806 Object not cached.
ORA-06512: at "SYS.DBMS_ICACHE", line 1227
ORA-06512: at line 1
```

See Appendix B for a description of the Oracle Database Cache errors.

For information about handling errors returned by PL/SQL procedures, see the *PL/SQL User's Guide and Reference.*

ADD_PLSQL_OBJECT Procedure

This procedure adds the named PL/SQL object (package, procedure, or function) to the cache. It also adds the associated object privileges and any synonyms to the cache.

Format

```
DBMS_ICACHE.ADD_PLSQL_OBJECT (
      schema_name IN VARCHAR2,
      object_name IN VARCHAR2 );
```

Parameters

schema_name
The name of the schema on the origin database that contains the PL/SQL object to be cached.

object_name
The name of the PL/SQL object to be cached.

Usage Notes

- To execute this procedure, you must connect to the cache using AS SYSDBA in the connect statement.

- You must execute the SET_ORIGIN_CREDENTIALS procedure before you execute this procedure.

- If you add a PL/SQL object that refers to an object, such as a table, that is not cached, you cannot cache the PL/SQL object. To correct the problem, cache the object to which the PL/SQL object refers, then add the PL/SQL object to the cache.

- You should cache only PL/SQL objects that contain read-only statements. Insert, update, or delete statements are routed to the origin database.

Examples

Assume that you have a PL/SQL procedure called GET_TOY_ITEMS in the schema ICPB. Because this procedure contains only queries, it is a good candidate for caching. To cache the procedure, take the following steps:

1. Make sure that any tables that are referenced by the GET_TOY_ITEMS procedure are already cached.

2. Connect to the cache using a utility such as SQL*Plus. Connect as a user with SYSDBA privileges:

```
connect system/manager @mycache.us.oracle.com AS SYSDBA;
```

3. Because the DBMS_ICACHE.ADD_PLSQL_OBJECT procedure must connect to the origin database, you must pass user credentials to the origin database. Execute the DBMS_ICACHE.SET_ORIGIN_CREDENTIALS procedure, specifying the user name and password of a user on the origin database who has DBA privileges:

```
execute DBMS_ICACHE.SET_ORIGIN_CREDENTIALS ('system','manager');
```

4. Execute the DBMS_ICACHE.ADD_PLSQL_OBJECT procedure, specifying the schema name and the procedure name:

```
execute DBMS_ICACHE.ADD_PLSQL_OBJECT ('ICPB', -
            'GET_TOY_ITEMS');
```

5. To verify that the procedure is cached, use the DESCRIBE statement:

```
describe ICPB.GET_TOY_ITEMS
```

ADD_TABLE Procedure

This procedure adds the specified table, its data, and its dependent objects to the cache. It also sets the synchronization policy of the table.

Format

```
DBMS_ICACHE.ADD_TABLE(
     schema_name IN VARCHAR2,
     table_name IN VARCHAR2,
     sync_method IN NUMBER,
     auto_refresh IN BOOLEAN,
     start_date IN DATE,
     interval IN INTEGER,
     build_indexes IN BOOLEAN,
     column_list IN VARCHAR2,
     exclusion_list IN VARCHAR2 );
```

Parameters

schema_name
The name of the schema that contains the table to be cached.

table_name
The name of the table to be cached.

sync_method
The synchronization method to be used for the table. Possible values are the following constants:

- SYNC_INCREMENTAL: Synchronizes the data by refreshing only the data that has changed since the table was cached or since it was last synchronized.

- SYNC_COMPLETE: Synchronizes the data by completely refreshing the table. When you select this method, Oracle Database Cache deletes all data in the table and replaces it with the current data from the origin database.

The default is SYNC_COMPLETE.

auto_refresh
A flag to indicate whether the table should be synchronized automatically on a given schedule or synchronized explicitly. To specify that the synchronization be

performed automatically, use the value TRUE. To specify that the synchronization be performed only when it is explicitly initiated, use the value FALSE.

The default is FALSE.

start_date
The date and time to start the automatic synchronization. If you specify TRUE for auto_refresh, but you do not specify a start_date, the start_date is set to the value SYSDATE.

The default is NULL.

For information about date and time formats, see the *Oracle8i SQL Reference.*

interval
The interval, in seconds, between automatic synchronization. If you specify TRUE for auto_refresh, but you do not specify an interval, Oracle Database Cache returns the error "Automatic refresh interval is not provided."

build_indexes
This parameter is currently ignored.

column_list
This parameter is currently ignored.

exclusion_list
This parameter is currently ignored.

Usage Notes

You must execute the SET_ORIGIN_CREDENTIALS procedure before you execute this procedure.

Examples

The following example adds the TOY_ITEMS table from the schema STOCK to the cache. It specifies an incremental, scheduled synchronization.

```
! Change the default date format to specify hours, minutes, seconds.
alter session set nls_date_format='DD-MON-YYYY HH24:MI:SS';
execute DBMS_ICACHE.ADD_TABLE ('ICPB', -
        'TOY_ITEMS', -
         DBMS_ICACHE.SYNC_INCREMENTAL, -
         TRUE, -
         '04-OCT-2000 01:10:00', 3600*24);
```

GET_FLAG Function

This function returns a flag that indicates whether or not the routing of PL/SQL to the cache is enabled or whether or not the collection of statistics is enabled.

Format

```
DBMS_ICACHE.GET_FLAG (
    flag IN NUMBER )
  RETURN BOOLEAN;
```

Parameters

flag

The name of the flag to be retrieved. Possible values are the following constants:

- PLSQL_ROUTING_ENABLED: Whether or not the routing of PL/SQL subprograms (procedures and functions) and anonymous blocks to the cache is enabled

- STATS_REPORTING_ENABLED: Whether or not the collection of statistics is enabled

Return Value

Returns TRUE if the specified flag is enabled. Returns FALSE if the specified flag is disabled.

Examples

The following example uses the GET_FLAG function to check whether or not PL/SQL subprograms are routed to the cache. If they are, it disables routing of PL/SQL.

```
begin
  if DBMS_ICACHE.GET_FLAG (DBMS_ICACHE.PLSQL_ROUTING_ENABLED)
     then DBMS_ICACHE.SET_FLAG(DBMS_ICACHE.PLSQL_ROUTING_ENABLED, FALSE);
  end if;
end;
```

GET_ROUTING_STATE Function

This function returns a constant that indicates whether the cache is running or paused.

Format

```
DBMS_ICACHE.GET_ROUTING_STATE
   RETURN NUMBER;
```

Return Value

Returns the constant RUNNING if the cache is running. If the cache is running, queries are routed to the cache. You can perform all operations on the cache, and all statistics are gathered.

Returns the constant PAUSED if the cache is paused. If the cache is paused, the cache is running, but 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.

Examples

The following example uses the GET_ROUTING_STATE procedure to check if the cache is paused. If it is, the example calls the procedure SET_ROUTING_STATE to start the paused cache.

```
begin
  if DBMS_ICACHE.GET_ROUTING_STATE ( ) = DBMS_ICACHE.PAUSED
     then DBMS_ICACHE.SET_ROUTING_STATE (DBMS_ICACHE.RUNNING);
  end if;
end;
```

GET_STATS_INTERVAL Function

This function returns the interval, in seconds, at which a specified type of statistics is collected.

Format

```
DBMS_ICACHE.GET_STATS_INTERVAL (
     interval IN NUMBER )
   RETURN NUMBER;
```

Parameters

interval

The type of statistics collection for which the interval is to be retrieved. Possible values are the following constants:

- AGGREGATE_STATS: The interval at which clients report to the cache the number of hits, misses, and other queries they have executed.

- QUERY_STATS: The interval at which clients report to the cache the queries they have executed, their status, and the hit miss counts.

- STATS_COMPILATION: The interval at which the cache compiles the query statistics (QUERY_STATS) reported by all clients.

Return Value

The interval, in seconds, for the specified type.

Examples

The following example returns the interval for QUERY_STATS to the screen in SQL*Plus:

```
set serveroutput on
execute DBMS_OUTPUT.PUT_LINE -
     (DBMS_ICACHE.GET_STATS_INTERVAL(DBMS_ICACHE.QUERY_STATS));
```

GET_STATS_RESET_TIME Function

This function returns the date and time that the specified statistics were last reset.

Format

```
DBMS_ICACHE.GET_STATS_RESET_TIME (
     category IN NUMBER)
    RETURN DATE;
```

Parameters

category
The category of statistics collection for which you want the last reset time returned. Possible values are the following constants:

- AGGREGATE_COUNTERS: The number of hits, misses, and other queries executed by the clients. If you specify this value, the function returns the last time the number of hits, misses, and other queries was reset to zero.

    These statistics can be viewed on the Hit/Miss Percentages chart and the Hit/Miss Count per Second chart.

- QUERY_COUNTERS: The number of hits and misses for each query. If you specify this value, the function returns the last time the number of hits and misses per query was reset to zero.

    This reset time and these statistics can viewed on the Hit/Miss Statistics property page.

- QUERY_ENTRIES: All queries executed by the clients. If you specify this value, the function returns the last time the queries were deleted from the statistics view, DBA_ICACHE_QUERY_STATS, in the cache.

    The queries can be viewed on the Hit/Miss Statistics property page.

Return Value

Returns the date and time that the specified type of statistics was reset to zero or deleted.

Examples

The following example returns the last reset time for AGGREGATE_COUNTERS to the screen in SQL*Plus:

```
set serveroutput on
execute DBMS_OUTPUT.PUT_LINE -
 (DBMS_ICACHE.GET_STATS_RESET_TIME (DBMS_ICACHE.AGGREGATE_COUNTERS));
```

GET_TABLE_SIZE Procedure

This procedure calculates the amount of disk space required to cache the named table. It calculates the size of the table and its dependent objects. Use this procedure to ensure that sufficient disk space exists before you add a table to the cache.

Format

```
DBMS_ICACHE.GET_TABLE_SIZE (
    schema_name IN VARCHAR2,
    table_name IN VARCHAR2,
    table_size OUT NUMBER );
```

Parameters

schema_name
The name of the schema that contains the table to be sized.

table_name
The name of the table to be sized.

table_size
The parameter to hold the size of the table, in megabytes (MB).

Usage Notes

You must execute the SET_ORIGIN_CREDENTIALS procedure before you execute this procedure.

Examples

The following example shows how to get the size for the SCOTT.EMP table:

```
declare tab_size number;
begin
DBMS_ICACHE.GET_TABLE_SIZE ('scott', 'emp', tab_size);
DBMS_OUTPUT.PUT_LINE:('Table size is:' || tab_size);
end;
/
Table size is: .0078125
```

REMOVE_PLSQL_OBJECT Procedure

This procedure removes the named PL/SQL object (package, procedure, or function) from the cache.

Format

```
DBMS_ICACHE.REMOVE_PLSQL_OBJECT (
      schema_name IN VARCHAR2,
      object_name IN VARCHAR2 );
```

Parameters

schema_name
The name of the schema that contains the PL/SQL object to be removed.

object_name
The name of the PL/SQL object to be removed from the cache.

Usage Note

You must execute the SET_ORIGIN_CREDENTIALS procedure before you execute this procedure.

Examples

The following example removes the PL/SQL procedure GET_TOY_ITEMS in schema ICPB from the cache:

```
execute DBMS_ICACHE.REMOVE_PLSQL_OBJECT('ICPB','GET_TOY_ITEMS' );
```

REMOVE_TABLE Procedure

This procedure removes the named table, its data, and its dependent objects from the cache. If no other objects owned by the named schema are in the cache, Oracle Database Cache performs additional cleanup operations.

Format

```
DBMS_ICACHE.REMOVE_TABLE (
      schema_name IN VARCHAR2,
      table_name IN VARCHAR2 );
```

Parameters

schema_name
The name of the schema that contains the table to be removed.

table_name
The name of the table to be removed from the cache.

Usage Notes

You must execute the SET_ORIGIN_CREDENTIALS procedure before you execute this procedure.

Examples

The following example removes the table ICPB.TOY_ITEMS from the cache:

```
execute DBMS_ICACHE.REMOVE_TABLE ('ICPB','TOY_ITEMS' );
```

RESET_STATS Procedure

This procedure resets the specified category of statistics.

Format

```
DBMS_ICACHE.RESET_STATS (
     category IN NUMBER );
```

Parameters

category
The category of statistics that you want to reset. Possible values are the following constants:

- AGGREGATE_COUNTERS: The number of hits, misses, and other queries executed by the clients. If you specify this value, the number of hits, misses, and other queries is reset to zero.

    These statistics can be viewed on the Hit/Miss Percentages chart and the Hit/Miss Count per Second chart.

- QUERY_COUNTERS: The number of hits and misses for each query. If you specify this value, the number of hits and misses per query is reset to zero.

    These statistics can viewed on the Hit/Miss Statistics property page.

- QUERY_ENTRIES: All queries executed by the clients. If you specify this value, the queries are deleted from the statistics view, DBA_ICACHE_QUERY_STATS, in the cache.

    The queries can be viewed on the Hit/Miss Statistics property page.

Examples

The following example resets the AGGREGATE_COUNTERS to zero:

```
execute DBMS_ICACHE.RESET_STATS (DBMS_ICACHE.AGGREGATE_COUNTERS);
```

SET_FLAG Procedure

This procedure enables or disables the routing of PL/SQL subprograms (procedures and functions) and anonymous blocks to the cache or it enables or disables the collection of statistics.

Format

```
DBMS_ICACHE.SET_FLAG (
    name IN NUMBER,
    value IN BOOLEAN );
```

Parameters

name
The name of the flag to be set. Possible values are the following constants:

- PLSQL_ROUTING_ENABLED: Whether to enable or disable the routing of PL/SQL subprograms and anonymous blocks to the cache.

- STATS_REPORTING_ENABLED: Whether to enable or disable the collection of statistics.

value
The value to which to set the flag. To enable a flag, set the value to TRUE. To disable a flag, set the value to FALSE.

Examples

The following example enables the collection of statistics:

```
execute DBMS_ICACHE.SET_FLAG ( -
            DBMS_ICACHE.STATS_REPORTING_ENABLED, -
            TRUE );
```

The following example disables the routing of PL/SQL subprograms and anonymous blocks to the cache:

```
execute DBMS_ICACHE.SET_FLAG ( -
            DBMS_ICACHE.PLSQL_ROUTING_ENABLED, -
            FALSE );
```

SET_ORIGIN_CREDENTIALS Procedure

This procedure sets the user name and password to be used to authenticate the user as a DBA on the origin database. You must execute this procedure before you execute procedures that involve the origin database.

Format

```
DBMS_ICACHE.SET_ORIGIN_CREDENTIALS (
      username IN VARCHAR2,
      password IN VARCHAR2,
      connect_type IN INTEGER );
```

Parameters

username
The name of a user on the origin database who has DBA privileges.

password
The password associated with the specified user.

connect_type
The type of connection to be made to the origin database. Possible values are the following constants:

- CONNECT_NORMAL: Connects to the origin database as the named user, with that user's privileges. This type of connection is sufficient to execute the Oracle Database Cache PL/SQL procedures and functions.

- CONNECT_SYSDBA: Connects to the origin database as the named user, with SYSDBA privileges. If the user does not have SYSDBA privileges on the origin database, an error is returned.

The default is CONNECT_NORMAL.

Usage Notes

You must execute this procedure before you execute the following procedures, which involve the origin database:

- ADD_PLSQL_OBJECT Procedure
- ADD_TABLE Procedure

- GET_TABLE_SIZE Procedure
- REMOVE_PLSQL_OBJECT Procedure
- REMOVE_TABLE Procedure
- SET_SYNC_METHOD Procedure
- SET_SYNC_SCHEDULE Procedure
- SYNC_PLSQL_OBJECT Procedure
- SYNC_TABLE Procedure
- SYNC_USERS Procedure

Examples

The following example shows how to call this procedure from SQL*Plus. You must be connected to the cache as a user with DBA privileges.

```
! Set the credentials for the origin database.
execute DBMS_ICACHE.SET_ORIGIN_CREDENTIALS ('system','manager');
```

SET_ROUTING_STATE Procedure

This procedure starts a paused cache or pauses a running cache.

Format

```
DBMS_ICACHE.SET_ROUTING_STATE (
    state IN NUMBER );
```

Parameters

state
The state to which you want to set the cache. To start a paused cache, set the value to the constant RUNNING. To pause a running cache, set the value to the constant PAUSED.

Usage Notes

- See "GET_ROUTING_STATE Function" on page 6-9 for information about the meaning of RUNNING and PAUSED.

- You cannot use PL/SQL to start a stopped cache or to stop a cache. To do either operation, see "Starting Oracle Database Cache and Related Software" on page 5-8.

Examples

The following example pauses the cache:

```
execute DBMS_ICACHE.SET_ROUTING_STATE (DBMS_ICACHE.PAUSED );
```

SET_STATS_INTERVAL Procedure

This procedure sets the interval, in seconds, for the specified type of statistics collection.

Format

```
DBMS_ICACHE.SET_STATS_INTERVAL (
    interval IN NUMBER,
    seconds IN NUMBER );
```

Parameters

interval

The type of statistics collection for which the interval is to be set. Possible values are the following constants:

- AGGREGATE_STATS: The interval at which clients report to the cache the number of hits, misses, and other queries they have executed.

  This interval reflects the rate at which the Hit/Miss Percentages chart and the Hit/Miss Count per Second chart are refreshed. (The charts are on the Performance Monitor property page of Cache Manager.) By default, Oracle Database Cache sets this interval to 5 seconds. If you are viewing the charts in Cache Manager, the rate is lowered. When the charts are closed, the rate reverts to the specified rate.

- QUERY_STATS: The interval at which clients report to the cache the queries they have executed, their status, and the hit miss counts. By default, Oracle Database Cache sets the interval to 60 seconds.

- STATS_COMPILATION: The interval at which the cache compiles the query statistics (QUERY_STATS) that are reported by all clients.

  This interval affects the rate at which the Hit/Miss Statistics property page in the Cache Manager is refreshed. By default, Oracle Database Cache sets the interval to 10 seconds. This interval should be equal to or less than the interval for QUERY_STATS.

seconds

The length of the interval, in seconds.

Examples

The following example sets the interval for collecting AGGREGATE_STATS to 10 seconds:

```
execute DBMS_ICACHE.SET_STATS_INTERVAL ( -
        DBMS_ICACHE.AGGREGATE_STATS, -
        10 );
```

SET_SYNC_METHOD Procedure

This procedure modifies the synchronization method of a table in the cache.

Format

```
DBMS_ICACHE.SET_SYNC_METHOD (
    schema_name IN VARCHAR2,
    table_name IN VARCHAR2,
    sync_method IN NUMBER );
```

Parameters

schema_name
The name of the schema that contains the table to be synchronized.

table_name
The name of the table to be synchronized.

sync_method
The type of synchronization to be used for the named table. Possible values are the following constants:

- SYNC_INCREMENTAL: Synchronizes the data by refreshing only the data that has changed since the table was cached or since it was last synchronized.

- SYNC_COMPLETE: Synchronizes the data by completely refreshing the table. When you select this method, Oracle Database Cache deletes all data in the table and replaces it with the current data from the origin database.

The default is SYNC_COMPLETE.

Usage Notes

- You must execute the SET_ORIGIN_CREDENTIALS procedure before you execute this procedure.

- This procedure does not change whether the synchronization is scheduled or unscheduled. To do that, use the SET_SYNC_SCHEDULE procedure.

Examples

The following example changes the synchronization method of ICPB.TOY_ITEMS to complete synchronization.

```
execute DBMS_ICACHE.SET_SYNC_METHOD ('ICPB', -
          'TOY_ITEMS', -
          DBMS_ICACHE.SYNC_COMPLETE);
```

SET_SYNC_SCHEDULE Procedure

This procedure modifies the synchronization schedule for a table in the cache. You can specify whether or not you want to schedule the table for automatic synchronization and, if so, the start time and interval for the synchronization.

Format

```
DBMS_ICACHE.SET_SYNC_SCHEDULE (
    schema_name IN VARCHAR2,
    table_name IN VARCHAR2,
    auto_refresh IN BOOLEAN,
    start_date IN DATE,
    interval IN INTEGER );
```

Parameters

schema_name
The name of the schema that contains the named table.

table_name
The name of the table for which you want to modify the synchronization schedule.

auto_refresh
A flag to indicate whether the table should be synchronized automatically on a given schedule or synchronized explicitly. To specify that the synchronization be performed automatically, use the value TRUE. To specify that the synchronization be performed only when it is explicitly initiated, use the value FALSE.

The default is FALSE.

start_date
The date and time to start the automatic synchronization. If you specify TRUE for auto_refresh, you must specify a start_date for the synchronization to be performed.

The default is NULL.

For information about date and time formats, see the *Oracle8i SQL Reference.*

interval

The interval, in seconds, between automatic synchronization of the table. If you specify TRUE for auto_refresh, you should specify an interval. If you do not, the synchronization will be performed only once, at the start_date.

The default is zero.

Usage Notes

- You must execute the SET_ORIGIN_CREDENTIALS procedure before you execute this procedure.

- This procedure does not change whether the synchronization is Complete or Incremental. To do that, use the SET_SYNC_METHOD procedure.

Examples

The following example changes the schedule of the synchronization of the table ICPB.TOY_ITEMS.

```
execute DBMS_ICACHE.SET_SYNC_SCHEDULE( 'ICPB', -
          'TOY_ITEMS', -
          TRUE, -
          '05-NOV-2000 02:00:00', -
          3600*24);
```

The following example changes the synchronization of the table ICPB.TOY_ITEMS to unscheduled.

```
execute DBMS_ICACHE.SET_SYNC_SCHEDULE( 'ICPB', -
          'TOY_ITEMS', -
          FALSE );
```

SYNC_PLSQL_OBJECT Procedure

This procedure initiates the immediate synchronization of a PL/SQL object. When you call this procedure, it synchronizes the definition (the metadata) of the specified PL/SQL object in the cache with the object in the origin database. Use this procedure when the definition of a PL/SQL object in the origin database is modified.

Format

```
DBMS_ICACHE.SYNC_PLSQL_OBJECT (
      schema_name IN VARCHAR2,
      object_name IN VARCHAR2 );
```

Parameters

schema_name
The name of the schema that contains the PL/SQL object to be synchronized.

object_name
The name of the PL/SQL object to be synchronized.

Usage Notes

- You must execute the SET_ORIGIN_CREDENTIALS procedure before you execute this procedure.

- You cannot set a synchronization policy or schedule for PL/SQL objects. You must manually initiate the synchronization using this procedure.

Examples

The following example updates the definition of the PL/SQL procedure GET_MUSIC_ITEMS:

```
execute DBMS_ICACHE.SYNC_PLSQL_OBJECT ('ICPB', -
            'GET_MUSIC_ITEMS');
```

SYNC_TABLE Procedure

This procedure initiates the immediate synchronization of a table. When you call this procedure, it synchronizes the specified table in the cache with the table in the origin database. Use this procedure when the synchronization policy of a table is not Scheduled. 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.

Format

```
DBMS_ICACHE.SYNC_TABLE (
      schema_name IN VARCHAR2,
      table_name IN VARCHAR2 );
```

Parameters

schema_name
The name of the schema that contains the table to be synchronized.

table_name
The name of the table to be synchronized.

Usage Notes

- You must execute the SET_ORIGIN_CREDENTIALS procedure before you execute this procedure.
- If the synchronization policy of a table is Complete, this procedure performs a Complete synchronization on the specified table immediately. Similarly, if the synchronization policy of a table is Incremental, this procedure performs an Incremental synchronization immediately.

Examples

The following example synchronizes the table ICPB.TOY_ITEMS immediately:

```
execute DBMS_ICACHE.SYNC_TABLE ( 'ICPB', 'TOY_ITEMS');
```

SYNC_USERS Procedure

This procedure synchronizes the list of users on the cache with the list of users on the origin database. It updates the list of nonsystem users and the attributes of the users on the cache, so that it is consistent with the list of nonsystem users on the origin database.

Format

```
DBMS_ICACHE.SYNC_USERS ;
```

Parameters

None.

Usage Notes

- You must execute the SET_ORIGIN_CREDENTIALS procedure before you execute this procedure.

- You must synchronize the user list in the following circumstances:

  - 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.

  - 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.

Examples

The following procedure synchronizes the user list:

```
execute DBMS_ICACHE.SYNC_USERS ;
```

A

Oracle Database Cache Views

Oracle Database Cache provides the following views, which let you use SQL to obtain information, such as statistics and objects cached, from the cache:

- DBA_ICACHE_AGGREGATE_STATS: Contains information about the number of queries that were hits, misses, or other.

- DBA_ICACHE_OBJECTS: Contains information about the tables and PL/SQL objects that are cached.

- DBA_ICACHE_QUERY_STATS: Contains information, such as status, hit and miss count, and SQL text, about each query executed.

DBA_ICACHE_AGGREGATE_STATS View

The view contains statistics about the queries executed by clients. It contains a single row that represents the aggregate statistics for the entire cache.

The following table shows the columns in the view:

Column Name	Data Type	Description
HIT_COUNT	NUMBER(8)	The number of queries that were satisfied by the cache.
MISS_COUNT	NUMBER(8)	The number of queries that were not satisfied by the cache and that were routed to the origin database.
OTHER_COUNT	NUMBER(8)	The number of requests that were requests other than queries, such as updates, inserts, or deletes.
TOTAL_COUNT	NUMBER(8)	The total number of queries executed.

The following example shows a SQL query that retrieves information from this view:

```
SQL> select * from dba_icache_aggregate_stats;

 HIT_COUNT MISS_COUNT OTHER_COUNT TOTAL_COUNT
---------- ---------- ----------- -----------
   3670945     196282        2075     3869302
```

DBA_ICACHE_OBJECTS View

This view contains information about the objects in the cache. It contains one row for each object in the cache.

The following table shows the columns in the view:

Column Name	Data Type	Description
OBJECT_NAME	VARCHAR2(128)	The name of the cached object.
OBJECT_TYPE	VARCHAR2(128)	The type of cached object. Valid values are: ■ TABLE ■ PACKAGE ■ FUNCTION ■ PROCEDURE
OWNER	VARCHAR2(128)	The owner of the cached object.
OBJECT_SIZE	NUMBER	The size of the cached table, in kilobytes (KB).
SYNC_METHOD	VARCHAR2(128)	The type of synchronization method for tables. Valid values are: ■ INCREMENTAL ■ COMPLETE Because you cannot set a synchronization method for PL/SQL objects, this column is empty for rows describing PL/SQL objects.

Column Name	Data Type	Description
AUTO_REFRESH	CHAR(1)	Whether or not the synchronization is scheduled to execute automatically at specified intervals. Valid values are:
		■ Y: Yes, the synchronization is scheduled.
		■ N: No, the synchronization is not scheduled.
		Because you cannot schedule synchronization for PL/SQL objects, this column is empty for rows describing PL/SQL objects.
INTERVAL	NUMBER	The interval, expressed in seconds, between scheduled synchronizations.
		Because you cannot set a synchronization interval for PL/SQL objects, this column is empty for rows describing PL/SQL objects.
START_DATE	DATE	The date and time for the synchronization to begin.
		Because you cannot schedule synchronization for PL/SQL objects, this column is empty for rows describing PL/SQL objects.

The following example shows a query that returns information about the cached table EMP:

```
SQL> alter session set nls_date_format= 'DD-MON-YYYY HH24:MI:SS';
Session altered.

SQL> select object_name, sync_method, start_date, interval from dba_icache_
objects where object_name='EMP';

OBJECT_NAME
--------------------------------------------------------------------------------
SYNC_METHOD
--------------------------------------------------------------------------------
START_DATE
--------------------------------------------------------------------------------
  INTERVAL
----------
EMP
COMPLETE
05-NOV-2000 01:01:00
     86400
```

DBA_ICACHE_QUERY_STATS View

This view contains information, such as status, hit and miss count, and SQL text, about each query executed against the cache. It contains one row for each query.

The following table shows the columns in the view:

Column Name	Data Type	Description
STATUS	VARCHAR2(7)	The last known status of this query. Valid values are: ■ 'HIT' ■ 'MISS' ■ 'UNKNOWN'
HIT_COUNT	NUMBER(8)	The number of times the query was satisfied by the cache.
MISS_COUNT	NUMBER(8)	The number of times the query was not satisfied by the cache and was routed to the origin database.
TOTAL_COUNT	NUMBER	The total number of times the query was executed.
USERNAME	VARCHAR2(20)	The user who executed this query.
SQL_TEXT	VARCHAR2(150)	The SQL text of the query.

The following example shows the number of queries executed against the EMP table and the total number of times they were executed:

```
SQL> select count(*), sum(TOTAL_COUNT) from DBA_ICACHE_QUERY_STATS
where upper(SQL_TEXT) like '%SCOTT.EMP%';

  COUNT(*)   SUM(TOTAL_COUNT)
---------- ----------
      23     1300
```

B

Error Messages

This appendix describes the error messages that you may encounter using Oracle Database Cache.

Message Formats

To help you find and fix errors, Oracle Database 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. Oracle Database Cache uses the prefix WTE. For lists of the prefixes used by other Oracle products, see *Oracle8i Error Messages*.

Note that directory specifications in the error messages usually use the UNIX convention. Unless otherwise noted, the path of the directory specification is the same for other platforms. Only the directory and file delimiters, such as slashes, may be different.

Oracle Database 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 in your environment. For example, on UNIX, set 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 Oracle Database Cache management engine is already started or another application is using the port.

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

WTE-01507 Error: OCI environment initialization failed

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

Action: On UNIX, check that the LD_LIBRARY_PATH environment variable exists and that the value of the environment variable is set so that the Oracle Database Cache library directory ($ORACLE_HOME/lib) precedes library directories for other Oracle homes. On Windows NT, check that the PATH environment variable exists and that the value of the environment variable is set so that the Oracle Database Cache library directory (*ORACLE_HOME*\lib) precedes library directories for other Oracle homes. If the environment variable is set correctly, 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 Oracle Database 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 Oracle Database 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: Oracle Database 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: Oracle Database Cache failed to create a file pointer to the log file wtcme.log.

Action: Check that the user who started the Oracle Database 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-01539 Error: Memory allocation error

Cause: Memory allocation call failed.

Action: Contact customer support.

WTE-01540 Error: Failed to determine extent size for the cache

Cause: A call to the size package on the cache failed.

Action: Review the associated error message for more information.

WTE-01557 Error: TCP port *number* not available for SSL server

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

Action: Verify that the Oracle Database Cache management engine is not already running. If another application is using the port, run the Oracle Database Cache Configuration Assistant to change the port number for Oracle Database 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: Oracle Database 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 priv-
ileges.

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-01608 Error: Could not write export file

   Cause: Oracle Database Cache could not write the export file to the directory $ORACLE_HOME/dbs.

   Action: Check that the user who started Oracle Database Cache has write access to the directory. Check that there is available disk space.

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

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

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

WTE-01610 Warning: Failed to set the environment character set to be unicode, use default in System

   Cause: Not enough system space to add a new environment variable.

   Action: Delete some unused environment variables.

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 Oracle Database Cache management engine is written to the log file.

   Action: Verify that the Oracle Database 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 Oracle Database 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-**02550 SSL server listening to TCP port** *number*

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

Action: Verify that the Oracle Database Cache management engine is using the correct port number.

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

Cause: Oracle Database Cache could not copy the export file to the cache node.

Action: Check that the user who started Oracle Database 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: Oracle Database 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, Oracle Database Cache prevented the decrease in size. If you tried to increase the size of the data file but there was insufficient disk space, Oracle Database 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: Oracle Database 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 is 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 is 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 is 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