

Oracle Reports Release 6.0: Reports Server

Technical White Paper
February, 1999

TABLE OF CONTENTS

Architecture.....	3
Reports Server Choices.....	9
A Case Study: Web CGI using any Web server	11
A Case Study: Web Cartridge using the Oracle Applications Server	26
Related Documentation.....	44
Glossary	45
Index.....	48

Oracle Reports Release 6.0: Reports Server Configuration

In today's fast-moving, competitive business world, clear and up-to-date information is needed for the accurate, expedient decision making requirements of an often geographically distributed workforce. The timely distribution of that information must be reliable, cost effective, and accessible to everyone who requires it. Oracle Reports provides an unbounded, easy-to-use, scalable, and manageable solution for high-quality database publishing and reporting.

Oracle Reports is a powerful Enterprise Reporting tool used by IS developers to create sophisticated dynamic reports for the Web and across the enterprise.

Its Application Server based architecture means report consumers require only a Web browser to view reports in industry standard formats. The Oracle Reports Server supports on-demand delivery of high-quality reports over the Web through native generation of HTML with Cascading Style Sheets and Adobe's Portable Document Format (PDF). Maintenance overheads are cut as reports are administered and maintained centrally and there is no requirement to install complex software on every users PC.

This white paper explains the following aspects of the Reports Server:

- architecture
- choices
- a case study: Web CGI using any Web server
- a case study: Web Cartridge using the Oracle Applications Server
- list of related documentation
- glossary

REPORTS SERVER ARCHITECTURE

The Reports Server can be configured in a number of ways depending upon your requirements. When used in a Web environment, the Reports Server architecture consists of four tiers¹:

- the thin client tier
- the Web server tier
- the Reports Server tier
- the database tier

The range of possible configurations runs from having all of these tiers on one machine to having each of these tiers on a separate machine. The most common configurations typically have the tiers spread

¹ In this white paper, the term tier refers to the logical location of the components that comprise the Reports Server architecture. Each of the tiers, though, could reside on the same or different machines.

across three or four machines. The graphics below provide a conceptual view of these common configurations.

Note: In the non-Web case, which will be discussed later, there are only three tiers because the Web server tier is not necessary.

Web Architecture: Server Configurations

The diagrams below illustrate two of the most common configurations for the Reports Server in a Web environment. The key difference between the two configurations is whether the Reports Server and Web server tiers are on the same or different machines. In the first case, the Web server and Reports Server reside on the same machine. In the second case, they are on different machines. The latter case requires a slightly different setup from the first.

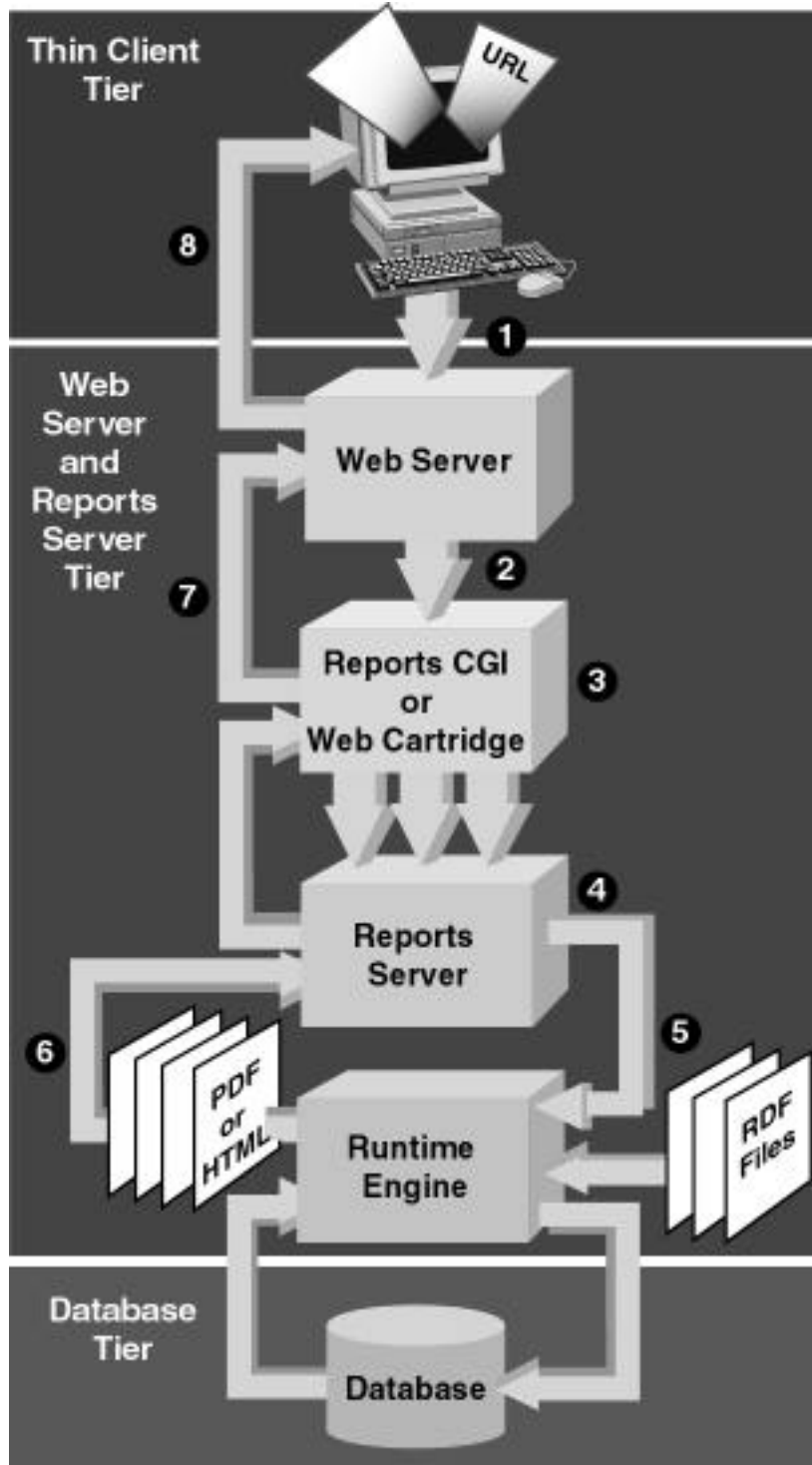


Figure 1: Web architecture, three machine configuration

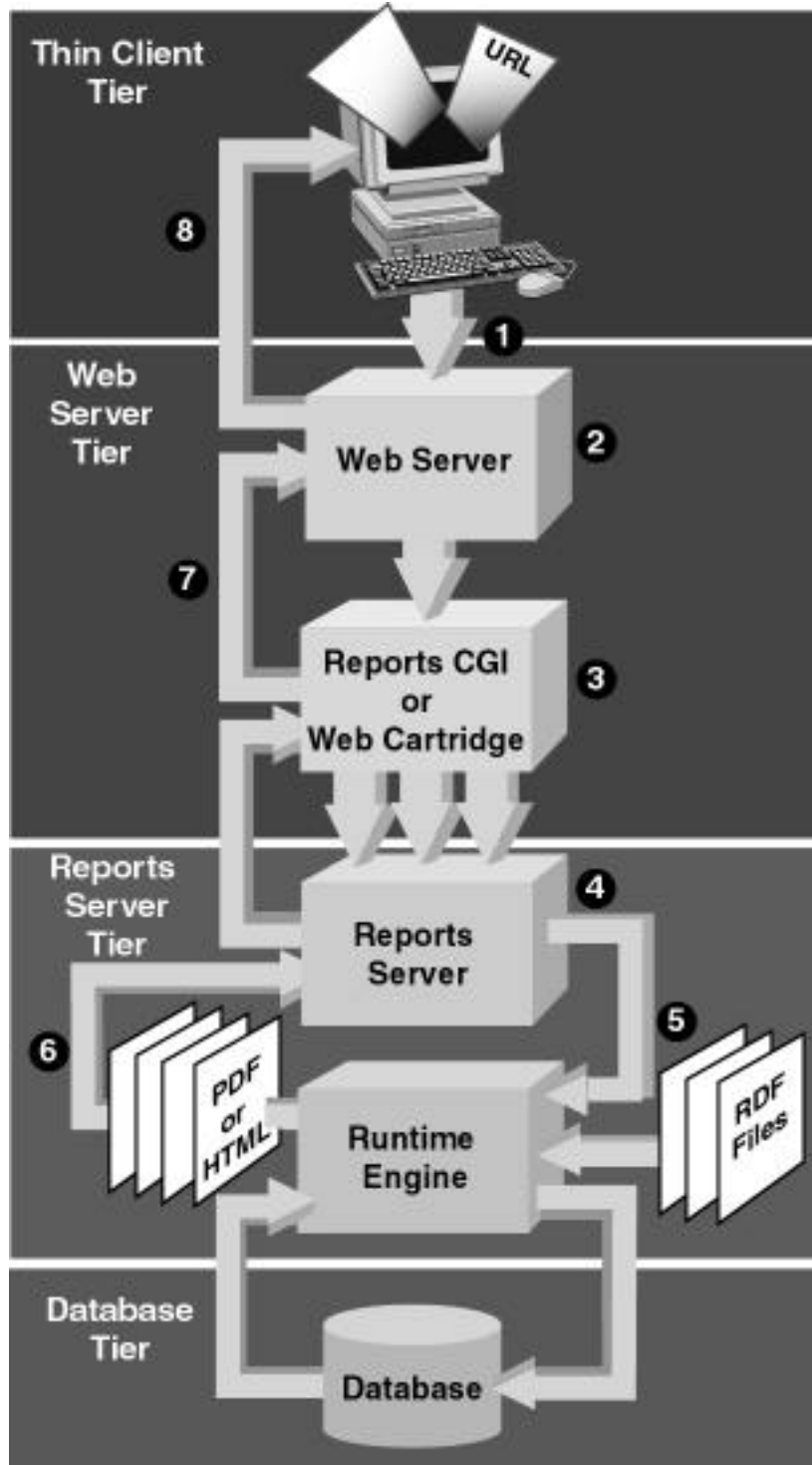


Figure 2: Web architecture, four machine configuration

Processing Web Reports

1. The client requests the report from their Web browser either by typing a URL or clicking a hyperlink. The Web browser passes the URL to the Web server.
2. To handle the request, the Web server invokes either the Reports CGI or the Reports Web Cartridge, depending upon which one you have configured.²
3. The Reports CGI or Web Cartridge parses the request. If necessary, the user is prompted for their database logon. The Reports Web Cartridge or CGI converts the request to a command line that can be executed by the Reports Server and submits it to the specified Reports Server.
4. If the request includes a time tolerance³, the Reports Server will check its output cache to determine whether it already has output that will satisfy the request. If it finds acceptable output in its cache, it will immediately return that output rather than executing the report.
5. The Reports Server receives the job request and queues it. When one of its runtime engines becomes available,⁴ it sends the command line to that runtime engine for execution.
6. The runtime engine executes the report.
7. The Reports Web Cartridge or CGI receives the report output from the Reports Server and sends it to the Web server.
8. The Web server sends the report output to the Web browser.

Non-Web Architecture: Server Configuration

The non-Web architecture differs from the Web architecture in that there is no Web browser or Web server. Report requests are sent to the Reports Server from a thin client such as the Reports Launcher or command line, RWCLI60. The non-Web architecture is useful to those who cannot use the Web to deploy their reports for some reason.

² When you use the Reports Web Cartridge, the Oracle Application Server must be your Web server. If you are not using the Oracle Application Server, you should use the Reports CGI.

³ For any job request that you send to the Reports Server, you can include a TOLERANCE argument. TOLERANCE defines the oldest output that the requester would consider acceptable. For example, if the requester specified 5 minutes as the TOLERANCE, the Reports Server would check its cache for duplicate report output that had been generated within the last five minutes.

⁴ When you configure the Reports Server, you can specify the maximum number of runtime engines it can use. If the Reports Server is under this maximum, it may start new runtime engines to handle requests. Otherwise, the request must wait until one of the runtime engines completes its current job.

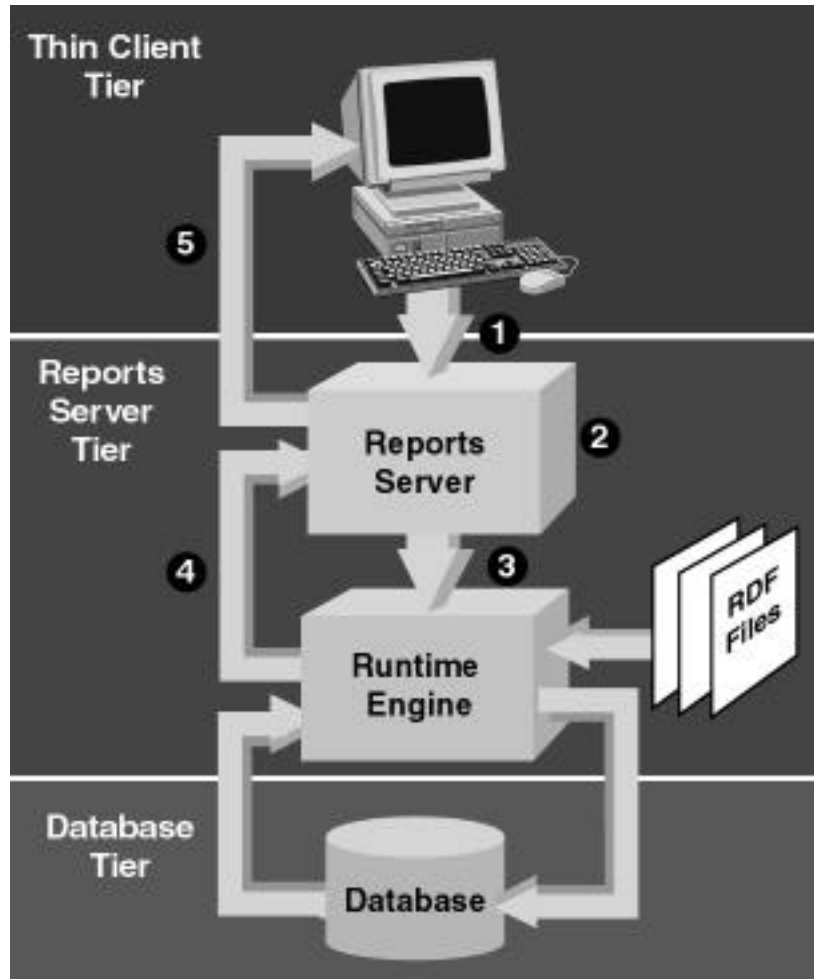


Figure 3: Non-Web architecture

Processing Non-Web Reports

1. The client requests the report using the command line (RWCLI60), the Reports Queue Manager, or the Reports Launcher (ActiveX control).
2. The Reports Server receives the job request and queues it. When one of its runtime engines becomes available, it sends the request to that runtime engine for execution.
3. The runtime engine executes the report.
4. The Reports Server is notified that the job has been completed.
5. If the Reports Server was called synchronously, it signals the client that the job has been completed. Note also that if the destination type (DESTTYPE) for the command line client is set to localfile in the job request, the output is transferred to the client.

REPORTS SERVER CONFIGURATION CHOICES

The configuration of the Reports Server can vary widely depending upon the requirements of your system. Before attempting to configure the Reports Server, you must make a number of important decisions based upon your requirements. By making these decisions beforehand, you can greatly simplify the configuration process.

Enable Web and non-Web requests

As you saw in the previous section, the Reports Server can accept job requests from both Web and non-Web thin clients. In the Web case, users run reports by clicking or typing a URL in their Web browser and, depending on the URL, the report output is served back to them in their browser or sent to a specified destination (e.g., a printer). In the non-Web case, users launch job requests using client software installed on their machines (i.e., Net8 and the Reports Thin Client, which is comprised of the Reports Launcher, the Reports Queue Manager, and RWCLI60).

To enable users to launch reports from a Web client, you need to install either the Reports Web CGI or Cartridge with your Web server to communicate between the Web server and the Reports Server. The Web CGI or Cartridge is required for your Web server to process report requests from Web clients. For more information, refer to the section entitled **Choose the Reports Web CGI or Cartridge**. To enable users to launch reports from a non-Web client, you need to install the required client software (i.e., Net8 and the Reports Thin Client) on each machine from which you plan to launch report requests.

From the perspective of configuration, the key differences between enabling Web and non-Web requests is as follows:

- Enabling Web requests requires that you install some additional software with your Web server, namely the Reports Web CGI or Cartridge, but obviates the need to install any client software beyond a Web browser.
- Enabling non-Web requests requires that you install and maintain client software on each machine from which you want to send job requests to the Reports Server.

The Web case is clearly the most cost effective because it reduces client maintenance costs, but there may be cases where launching non-Web requests is a necessity for other reasons. The Reports Server supports both Web and non-Web requests and they are not mutually exclusive.

Choose the Reports Web CGI or Cartridge

As discussed in the previous section, to use the Reports Server in a Web environment, you must install and configure either the Reports Web CGI or Reports Web Cartridge to handle the transmission of job requests and output between your Web server and the Reports Server. The key consideration in this choice is whether you are using or are planning to use the Oracle Application Server Release 4.0. The Reports Web Cartridge is designed specifically to work with the Oracle Application Server. If you are not using the Oracle Application Server, you should use the Reports Web CGI, which works with any CGI-aware Web server.

Choose the location of the Reports Server

As described in the **Reports Server Architecture** section, you can place the Reports Server on the same machine as your Web server or on a different machine. As you make this decision, you should consider the following:

- Having the Reports Server on the same machine with the Web server, of course, requires more of the machine's resources. If you plan to have both on the same machine, you need to take that into account when determining the machine's resource requirements (i.e., memory and disk space).
- Having the Reports Server and the Web server on the same machine reduces network traffic. The Reports CGI and Web Cartridge must reside on the same machine as the Web server. If the Reports Server is on a different machine, its transmissions to the Reports CGI and Web Cartridge must travel across a network. If it is on the same machine, the transmissions do not have to travel across the network.
- It is easier to share the Reports Server's cache with the Web server if both reside on the same machine. Refer to the next section, "Choose whether to share the cache," for more information. If the Reports Server is on a different machine and you want to share its cache, you must place the cache on a file system that is shared with the Web server machine.

Choose whether to share the cache

The Web server and the Reports Server both have cache locations for file storage and retrieval. You can choose to have the Reports Server and the Web server caches share the same location. As you make this decision, you should consider the following:

- Sharing the cache can improve performance because, if the caches are shared, the Reports Server does not have to copy output files from its cache to the Web server's cache.
- Sharing the cache can also reduce disk space requirements. If the cache is not shared, one copy of the report output is maintained in the Reports Server cache and one copy in the Web server cache. If the cache is shared, only one copy of the report output is maintained, effectively halving the amount of disk space needed.
- Sharing the cache simplifies cache maintenance. The Reports Server automatically cleans up its own cache by removing report output files. If the cache is shared, the Reports Server will also clean up report output files in the Web server cache. If the cache is not shared, the Web server cache must be cleaned up by some other mechanism.
- If the Reports Server and the Web server reside on different machines, sharing the cache requires that you place it on a network drive that is accessible to both machines. The users running the servers will also need to have privileges on the network drive where the cache is located.

Typically, if the Reports Server and the Web server are on the same machine, it makes sense to have a shared cache.

A CASE STUDY: WEB CGI USING ANY WEB SERVER

This section contains step-by-step instructions on how to configure the Reports Server using the Web CGI and any CGI-aware Web server. In order to make these configuration procedures meaningful, it is necessary to make several assumptions. For the purposes of this section, we assumed the following:

- You are configuring the Reports Server to enable Web requests.
- You are using the Reports Web CGI with the CGI-aware Web server of your choice.
- The Reports Server is installed on a different machine than the Web server.
- The cache is not shared between the Reports Server and the Web server.

Tip: This case assumes that you will install software, configure your server settings, and save .RDFs on the "C" drive.

Configuration Overview

Step 1. Install your software.

Step 2. Configure your Web server. Refer to your Web server documentation.

Step 3. Configure the Reports Web CGI on the same machine as your Web server.

Step 4. Configure the Reports Server.

Step 5. Make reports available to your users.

Step 6. (Optional) Further customize the Reports Server configuration.

Step 1. Install your software

Before you begin, you have to:

1. On your Web server machine, install the Oracle Reports Thin Client component after installing your Web server.
2. On your Reports Server machine, install the Oracle Reports Multitier Server component and the Report Builder. Note that the Report Builder is required only if you need to build reports. If the Reports Server machine is used only as a deployment machine, then do not install the Report Builder component.

Refer to the *Getting Started* manual for more information on installing the Oracle Reports Thin Client and Multitier Server component and the Report Builder.

Refer to your Web server documentation for more information on installing the Web server.

Tip: To install just the Reports Thin Client, Multitier Server, or the Report Builder, start the Oracle Installer and choose the Custom Installation. From the Available Products list box, expand the Oracle Developer - Reports node and choose Reports Thin Client, Multitier Server, or Report Builder as desired. Proceed with the installation as described in the *Getting Started* manual.

Step 2. Configure your Web server

This step is performed on the Web server machine.

Refer to your Web server documentation for more information on configuring your Web server.

1. Start your Web Server.
2. Start your browser.
3. Create a listener.
4. Configure your Web server mapping and make note of the following physical and virtual directories:

Directory Description	Physical directory example	Virtual directory example	Permissions required
Web script	C:\YOUR_WEBSERVER\BIN	/CGI-BIN	execute
Cache	C:\YOUR_WEBSERVER\CACHE	/CACHE	read
HTML Document	C:\YOUR_WEBSERVER\DOCS	/DOCS	read

Table 1: Web server physical and virtual directories

Note: The Web script directory on your Web server contains CGI executables.

Step 3. Configure the Reports Web CGI

Before you begin this step, we recommend that you back up the registry before making any changes. Refer to your operating system's documentation for more information.

This step is performed on the Web server machine.

To configure Reports Web CGI on Microsoft Windows NT:

1. Copy `rwsgi60.exe` (located in the `ORACLE_HOME\BIN` directory) to your Web script directory.

Tip: The Web script directory is defined in your Web server configuration. Refer to your Web server documentation for more information. In the Table 1 example, the Web script directory is `C:\YOUR_WEBSERVER\BIN`.

2. On your desktop, choose **Start**→**Run**.
3. Type `regedit` to display the Registry Editor.
4. Expand **Hkey_Local_machine**→**Software**→**Oracle**.
5. Choose **Edit**→**New**→**String value** to add the following environment variables:

Note: Refer to your operating system's documentation to determine if the values you set are case sensitive.

Name	Description	Example Values
REPORTS60_SHARED_CACHE	<p>This environment variable specifies whether the location of the Reports Server's cache is shared with the Web server's cache.</p> <p>The values are YES or NO.</p> <p>When you set REPORTS60_SHARED_CACHE, REPORTS60_PHYSICAL_MAP, and REPORTS60_VIRTUAL_MAP, the Reports Server ignores REPORTS60_WEBLOC and REPORTS60_WEBLOC_TRANSLATED.⁵</p>	NO
REPORTS60_VIRTUAL_MAP	<p>Specifies where the Web server looks for the Reports Server output (file cache). It is a virtual directory, as defined in the Web server configuration file. It may be mapped directly to the Reports Server file cache directory (defined in the .ORA file for the Reports Server being used) only if that directory is mapped from the Web server machine.</p> <p>REPORTS60_VIRTUAL_MAP is usually mapped to the physical directory defined by the REPORTS60_PHYSICAL_MAP parameter, which is valid regardless of whether the Reports Server and the Web server share the same file system.</p> <p>If this parameter is not set, the Web CGI will not display HTML/PDF output in the client browser ("Cannot access report output" message will result), although the report will run and output will be produced.</p> <p>When you set REPORTS60_SHARED_CACHE, REPORTS60_PHYSICAL_MAP, and REPORTS60_VIRTUAL_MAP, the Reports Server ignores REPORTS60_WEBLOC and REPORTS60_WEBLOC_TRANSLATED.</p>	<p>/CACHE</p> <p>Note: You may need to include another slash (e.g., /CACHE/) depending on the operating system you are using. Refer to your operating system's documentation for more information.</p>

⁵ The REPORTS60_WEBLOC and REPORTS60_WEBLOC_TRANSLATED environment variables are provide for backward compatibility only. We recommend that you do not use them.

Name	Description	Example Values
REPORTS60_PHYSICAL_MAP	<p>Specifies which directory the report output will be transferred to on the Web server machine if the Reports Server and Web server do not share file systems. This must be the absolute physical path of the virtual directory specified in REPORTS60_VIRTUAL_MAP and defined in the Web listener configuration.</p> <p>If this variable is set to an invalid value, the Reports CGI will be not able to display HTML/PDF output in the client browser ("Cannot access report output" message will result), although the report will run and output will be produced.</p> <p>When you set REPORTS60_SHARED_CACHE, REPORTS60_PHYSICAL_MAP, and REPORTS60_VIRTUAL_MAP, the Reports Server ignores REPORTS60_WEBLOC and REPORTS60_WEBLOC_TRANSLATED.</p>	C:\YOUR_WEB_SERVER\CACHE

Table 2: Environment variables for Web CGI

Step 4. Configure the Reports Server

This step is performed on the Reports Server machine.

To modify the **REPORTS60_PATH**:

1. On the Reports Server machine desktop, choose **Start → Run**.
2. Type `regedit` to display the Registry Editor.
3. Expand **Hkey_Local_machine → Software → Oracle**.
4. Double click **REPORTS60_PATH**.
5. Add a new reports source path. For example:

`C:\WEB_REPORTS`

To create a tnsnames.ora entry for the Reports Server:

1. Open the tnsnames.ora file (located in ORACLE_HOME\NET80\ADMIN) in a text editor.
2. Add the following Reports Server entry to tnsnames.ora:

```
repserver.world =(ADDRESS = (PROTOCOL = TCP)(Host = repserver_machine.mydomain)
(Port = 1949))
```

where:

repserver.world is the name of the server instance and *.world* is the domain specified in the NAMES.DEFAULT_DOMAIN setting in the sqlnet.ora file. If the NAMES.DEFAULT_DOMAIN setting is not defined in the sqlnet.ora, then omit *.world* from the name of the server instance.

repserver_machine.mydomain is the host name or IP address of the machine.

1949 is the port number to which the server is listening.


3. Repeat steps 1 and 2 on the Web server machine.

To configure the Reports Server on Windows NT as a service:

1. On the Reports Server machine desktop, choose **Start→Run**.
2. Type the following command line argument:

```
rwmts60 -install repserver tcpip
```

Tip: Repserver does not need to have the domain qualifier appended to it (e.g., *.world*).

3. Click **Yes** to install the Reports Server as an NT Service and Click **OK** to start up the service.
4. On your desktop, choose **Start→Settings→Control Panel** and double-click  (Services).
5. At the Services dialog box, choose **Oracle Reports Server [repserver]** and click **Startup**.
6. From the startup dialog, select This Account in the Log On As section and type or select an operating system user name and password. This specifies that the server is run as that user.

Important: Ensure that the user running the Reports Server service has access to a default printer.

7. Set the service to start automatically (when the system is booted up).
8. Click **OK**.
9. Click **Start**. A Service Control message box indicates when your Reports Server has started. If your Reports Server cannot start, see the Troubleshooting section for more information.

Tip: When you start the Reports Server for the first time, a Reports Server configuration file (e.g., *repserver.ora*) is created in the ORACLE_HOME\REPORT60\SERVER directory. Refer to this file to determine the default setting for your Reports Server cache.

Step 5. Make your reports available to end users

To build a report:

1. On the Reports Server machine, create the reports source directory for saving .RDF files using the path that was previously defined in the REPORTS60_PATH (e.g., C:\WEB_REPORTS).
2. Start the Report Builder and build an .RDF. Save the .RDF in the reports source directory on the Reports Server machine (e.g., C:\WEB_REPORTS). Refer to the Report Builder online help for more information.

To specify report run requests:

Specify a URL run request

1. Open your Web browser.
2. Make the following request:

```
http://your_webserver/cgi-bin/rwcgi60.exe?report=your_report.rdf+
userid=user_name/password@mydb+server=repserver+desformat=html+destype=cache
```

If the report does not display in your Web browser, see the Troubleshooting section for more information.

Specify a URL run request using key mapping

1. Ensure that the cgicmd.dat is located in the ORACLE_HOME\REPORT60 directory on your Web server machine.

Tip: Type: `http://your_webserver/cgi-bin/rwcgi60.exe/showmap?` in your Web browser to verify the mapping file that is being used.

2. Open cgicmd.dat in a text editor.
3. Add the following:

```
key1: REPORT=your_report.rdf USERID=user_name/password@mydb DESFORMAT=html
SERVER=repserver DESTYPE=cache
```

4. Open your Web browser and make the following request:

```
http://your_webserver/cgi-bin/rwcgi60.exe?key1
```


Add the URL to a sample web page

1. Copy the `replist.htm` and associated `.GIFs` (located in `ORACLE_HOME\REPORT60\DEMO\WEBFILES`) to your Web server's HTML document directory.

Tip: The Web server's HTML document directory is defined in your Web server configuration.

2. Open `replist.htm` in a text editor.
3. Add the URL to the bottom of the file (before `</body>`). For example:

```
<P>
<a href="http://your_webserver/cgi-bin/rwcgi60.exe?key1">Test Report</a>
<P>
(This tests a Reports Server Configuration)
<P>
```

4. Save the file.
5. Open your Web browser and make the following request:

```
http://your_webserver/docs/replist.htm
```

6. At the bottom of the page, click `Test Report`.

This completes the Report Server configuration case study. You have successfully configured the Reports Server for dynamic reporting using the Reports Web CGI.

Step 6. (Optional) Customize the Reports Server

There are a number of ways in which you can refine your Reports Server configuration:

- implement database user authentication
- map URL parameters
- modify the configuration file
- update environment variables

Implement database user authentication

You can configure the Reports Web CGI to prompt users for their database logons when they run reports. When users successfully log on, their browser is sent an encrypted cookie. By default, this cookie expires after 30 minutes. When a cookie has expired, subsequent requests (i.e., ones that are configured for database logon) must be re-authenticated. Note that the `REPORTS60_COOKIE_EXPIRE` environment variable sets the expire time of the cookie. See Table 4: Environment variables for Web CGI for more information.

1. In the key mapping file (cgicmd.dat), search for the dummy key and change the `SERVER` parameter to the tnsname of your Reports Server. An example of the dummy key is shown below:

```
dummy: dummy.rdf destype=cache desformat=html server=repserver
```

Note: The Reports Server uses this dummy report to verify the user's logon information. The dummy.rdf should be located in the `ORACLE_HOME\REPORT60` directory and defined in the `REPORTS60_PATH` environment variable.

2. If you want users to authenticate *every time* they run a particular report, include the `AUTHTYPE=D` argument in the report URL or include the `%D` argument in the key map entry in the key mapping file (cgicmd.dat).

If you want users to authenticate and *remain authenticated* until the cookie expires, omit the `USERID` parameter.

Note: To hide parameters in your URL requests, you could use a key mapping file entry. Refer to the Map URL Parameters section.

Database Authentication Demo

For your convenience, sample templates (welcome.htm and replist.htm) have been provided to demonstrate various run requests. If you wish to use these sample templates to help you implement database authentication, do the following:

1. Copy the files (welcome.htm, replist.htm, and associated .GIFs) located in the `ORACLE_HOME\REPORT60\DEMO\WEBFILES` directory to the HTML document directory defined in your Web server configuration. You can modify the templates to your own UI specifications as desired.
2. In welcome.htm, search for `http:`, and change the URLs to point to your Web script directory.
3. In replist.htm, search for `http:`, and change the URLs to point to your Web script directory. Additionally, search for `report=`, and change the reports to your .RDFs.

Map URL parameters

To simplify or hide parameters in your URL requests, you can use a key mapping file. The mapping file takes a URL parameter and maps it to the command line arguments that govern the running of the report. For example, one parameter in the URL could map to all of the command line arguments needed to run the report. By using key mapping, the command line arguments are all hidden from the user. Below is an example of a key mapping for a restricted run with a Parameter Form:

A submission of:

```
http://your_webserver/cgi-bin/rwcgi60.exe?key+par1+par2+parN
```

where the key mapping file contains:

```
KEY: module=myreport deptno=%1 myparam=%2 %*
```

generates the equivalent of the following command line request:

```
RWCLI60 module=myreport deptno=par1 myparam=par2 parN
```

1. On the Web server machine, modify the key mapping file, ORACLE_HOME\REPORT60\cgicmd.dat. Except for the special parameters that are described in the file itself, the command line arguments follow the syntax rules of RWCLI60, which is described in the Report Builder online help.

Tip: Type: `http://your_webserver/cgi-bin/rwcgi60.exe/showmap?` in your Web browser to verify the mapping file that is being used.

2. Update the hyperlinks in your Web page (e.g., replist.htm) to use the keys you defined in cgicmd.dat.

Modify the Reports Server configuration file

1. Open ORACLE_HOME\REPORT60\SERVER\repsrvr.ora and modify the parameters as desired.

Parameters	Description
IDENTIFIER	IDENTIFIER is an internal setting that contains the encrypted queue administrator user ID and password. You should not attempt to modify it. If IDENTIFIER is not specified or is deleted, or the configuration file is not present, anyone can supply any user ID and password from the Reports Queue Manager to log on as the queue administrator. Once someone has logged on in this way, the user ID and password they specified becomes the queue administrator user ID and password until it is changed from the Queue Manager.
MAXCONNECT	MAXCONNECT is the maximum number of processes that can communicate with the server process at any one time. This setting is the sum of the number of engines and clients, and must be greater than two (at least one engine and one client). Default value=20
SOURCEDIR	SOURCEDIR is a path to be searched before REPORTS60_PATH when searching for reports and other runtime files. This setting is useful when you have more than one Reports Server sharing the same ORACLE_HOME because each Reports Server can search different directories.
CACHEDIR	CACHEDIR is the cache for the Reports Server. CACHEDIR can be set to any directory or logical drive on the machine. If it is not specified, the default is ORACLE_HOME\REPORT60\SERVER\CACHE.
TEMPDIR	TEMPDIR is a directory that will be used instead of REPORTS60_TMP when creating temporary files. TEMPDIR can be set to any directory or logical drive on the machine.
CACHESIZE	CACHESIZE is the size of the cache in megabytes. If you expect to store the output of many of your reports in the Reports Server cache, you may want to increase this setting. If you do not expect to store a lot of output in the cache and have limited system resources, you may want to reduce it. Once the cache grows beyond the set size, the Reports Server cleans up the cached files on a first in, first out basis. Default value=50

Parameters	Description
MINENGINE	<p>MINENGINE is the minimum number of runtime engines the Reports Server should have available to run reports. The server process will attempt to keep at least this many engines active. Ensure that you have sufficient memory and resources available to accommodate this many engines.</p> <p>Default value=0</p>
MAXENGINE	<p>MAXENGINE is the maximum number of runtime engines available to the Reports Server to run reports. The server process will attempt to keep no more than this many engines active. Ensure that you have sufficient memory and resources available to accommodate this number of engines.</p> <p>Default value=1</p>
INITENGINE	<p>INITENGINE is the initial number of runtime engines started by the Reports Server. The server process will spawn this many engines when it is started. It will wait two minutes for these engines to connect to it and will shut itself down if they fail to do so. If the engines cannot connect in this amount of time, there is usually some setup problem.</p> <p>Default value=1</p>
MAXIDLE	<p>MAXIDLE is the maximum amount of time an engine is allowed to be idle before being shut down. Note that the Reports Server will not shut down the engine if doing so would reduce the number of available engines to less than those defined in the MINENGINE.</p> <p>Default value=30</p>
SECURITY	<p>SECURITY is the security level (0, 1, 2, or 3) for accessing cached output files through the Reports Queue Manager. 0 means that anyone can access a job's cached output. 1 means that only a user whose user ID is identical to that of the user who ran the job can access the job's cached output. 2 means that only the same process that sent the job can access the job's cached output. 3 means that the cached output cannot be accessed.</p> <p>Default value=1</p>
ENGLIFE	<p>ENGLIFE is the maximum number of reports that an engine will run before shutting itself down. The Reports Server will then bring up fresh engines for new requests.</p> <p>Default value=50</p>
FAILNOTEFILE	<p>FAILNOTEFILE is path and file name of the notification message template that is sent to specified email addresses for jobs that fail to run.</p>
SUCCNOTEFILE	<p>SUCCNOTEFILE is the path and file name of the notification message template that is sent to specified email addresses for jobs that run successfully.</p>

Parameters	Description
CLUSTERCONFIG	<p>CLUSTERCONFIG is the configuration of slave servers to the master server. Clustering allows you to run reports on multiple Reports Servers. The master server can identify available slave servers and start their engines as needed. You can set up many servers as slaves to the master server. Use the following syntax in the master server configuration file:</p> <pre>Clusterconfig="(server=<servername> minengine=<minimum number of master engines> maxengine=<maximum number of master engines> initengine=<initial number of master engines> cachedir=<directory of central cache>)"</pre> <p>Refer to the Oracle Reports 6.0 Reports Server Clustering technical white paper for more information.</p>
LOGOPTION	<p>LOGOPTION is the type of log information you want inserted into the log file. The options are alljob, failedjob, and succeededjob.</p>
PERSISTFILE	<p>PERSISTFILE indicates the location of the Reports Server's .DAT file, which contains the details of scheduled jobs. If PERSISTFILE is not specified, the default is ORACLE_HOME\REPORT60\SERVER.</p>
MAILPROFILE	<p>(Windows NT only) MAILPROFILE specifies the mail profile and password to be used when mailing reports.</p>

Table 3: Reports Server configuration file parameters

Update environment variables

1. Modify any of the following environment variables to customize your configuration.

Variable	Description
REPORTS60_COMMON_AUTH	<p>Specifies the template used to determine the common properties of the authentication form, such as the size of the form and whether a toolbar is visible. This template is used in conjunction with the HTML template set in the REPORTS60_DB_AUTH environment variable.</p> <p>Default value=common.htm</p> <p>It is recommended that you keep the default.</p>
REPORTS60_COOKIE_EXPIRE	<p>Determines the idle time of the cookie for database authentication in minutes.</p> <p>Default value=30</p> <p>Cookies save encrypted user names and passwords on the client-side when users first authenticate themselves. When the server receives a cookie from the client, the server compares the time saved in the cookie with the current system time. If the time is longer than the number minutes defined in the REPORTS60_COOKIE_EXPIRE environment variable (e.g., 30 minutes), the server rejects the cookie and returns the database authentication form and an error message to the client. Users must re-authenticate to run the report.</p>
REPORTS60_DB_AUTH	<p>Specifies the database authentication template used to authenticate the user name, password, and database.</p> <p>Default value=dbauth.htm</p>
REPORTS60_ENCRYPTION_KEY	<p>Specifies the encryption key used to encrypt the user name and password for the cookie. The encryption key can be any character string.</p> <p>Default value= reports6.0</p>
REPORTS60_SSLPORT	<p>If you are using SSL and you want to use a port number other than 443, you can use this variable to set a different port number.</p> <p>Default value= 443</p>
REPORTS60_CGIDIAGBODYTAGS	<p>For the Reports Web CGI, specifies HTML tags that will be inserted as a <BODY...> tag in the RWCGI60 diagnostic/debugging output. For instance, you may want to use this environment to set up text/background color, image, etc.</p>

Variable	Description
REPORTS60_CGIDIAGHEADTAGS	For the Reports Web CGI, specifies HTML tags to insert between <HEAD> ...</HEAD> tags in the RWCGI60 diagnostic/debugging output. For instance, you may want to use this environment to set up <TITLE> or <META...> tags.
REPORTS60_CGIHELP	For the Reports Web CGI, defines URL/URI of the RWCGI60 help file, which will be navigated to when RWCGI60 is invoked with the empty request: http://your_webserver/rwsgi60?. For example., setting it to http://www.yahoo.com will go to that URL; setting it to myhelpfile.htm will display the file: http://your_webserver/myhelpfile.htm. If this parameter is not defined, a default help screen will be displayed.
REPORTS60_CGIMAP	For the Reports Web CGI, defines fully qualified filename/location of the RWCGI60 map file (if map file configuration is used), e.g., C:\ORANT\REPORT60\cgicmd.dat.
REPORTS60_CGINODIAG	For the Reports Web CGI, when defined, disables all debugging/diagnostic output, such as help and showmap, from RWCGI60. For example, http://your_webserver/rwsgi60/help? will not work when REPORTS60_CGINODIAG is defined.

Table 4: Environment variables for Web CGI

Web CGI case study reference files

If you wish to check your Reports Server configuration, refer to the following:

tnsnames.ora (on the Reports Server machine and Web server machine)

Location: ORACLE_HOME\NET80\ADMIN\tnsnames.ora

```
repserver.world =(ADDRESS = (PROTOCOL = TCP)(Host = repserver_machine.mydomain)
(Port = 1949))
```

where *repserver.world* the name of the server instance and *.world* is the domain specified in the NAMES.DEFAULT_DOMAIN setting in the sqlnet.ora file. If the NAMES.DEFAULT_DOMAIN setting is not defined in the sqlnet.ora, then omit *.world* from the name of the server instance.

repserver_machine.mydomain is the host name or IP address of the machine.

1949 is the port number to which the server is listening.

sqlnet.ora (on the Reports Server machine and Web server machine)

Location: ORACLE_HOME\NET80\ADMIN\sqlnet.ora

Note that the following is a typical default setting:

```
TRACE_LEVEL_CLIENT=OFF
names.directory_path = (TNSNAMES)
names.default_domain = world
name.default_zone = world
```

Reports Server configuration file (on the Reports Server machine)

Location: ORACLE_HOME\REPORT60\SERVER\repserver.ora

```
maxconnect=20
cachedir="c:\ORACLE_HOME\REPORT60\server\cache"
cachesize=50
minengine=0
maxengine=1
initengine=1
maxidle=30
security=1
englife=50
```


Environment Variables (on the Web server machine)

In DOS, type `set` to view the list of environment variables set on your Web server machine.

```
REPORTS60_SHARED_CACHE "NO"
REPORTS60_PHYSICAL_MAP "C:\YOUR_WEBSERVER\CACHE"
REPORTS60_VIRTUAL_MAP "/CACHE"
```

Note: For `REPORTS60_VIRTUAL_MAP`, You may need to include another slash (e.g., `/CACHE/`) depending on the operating system you are using. Refer to your operating system's documentation for more information.

URL run request (on the client machine)

Enter the following report run request:

```
http://your_webserver/cgi-bin/rwcgi60.exe?report=your_report.rdf+
userid=username/password@mydb+server=repserver+desformat=html+destype=cache
```

URL key mapping file (on the Web server machine)

Location: `ORACLE_HOME\REPORT60\cgicmd.dat`

```
key1: REPORT=your_report.rdf USERID=username/password@mydb DESFORMAT=html
SERVER=repserver DESTYPE=cache
```

URL specified on a web page (on the Web server machine)

Location: `YOUR_WEBSERVER\DOCS\REPLIST.HTM`

```
<P>
<a href="http://your_webserver/cgi-bin/rwcgi60.exe?key1">Test Report</a>
<P>
(This tests a Reports Server Configuration)
<P>
```

CASE STUDY: USING WEB CARTRIDGE ON THE ORACLE APPLICATIONS SERVER

This section contains step-by-step instructions on how to configure the Reports Server using the Web Cartridge on the Oracle Applications Server (OAS). In order to make these configuration procedures meaningful, it is necessary to make several assumptions. For the purposes of this section, we assumed the following:

- You are configuring the Reports Server to enable Web requests.
- You are using the Reports Web Cartridge with Oracle Application Server, Release 4.0.
- The Reports Server is on a different machine than the Oracle Application Server.
- The cache is not shared between the Reports Server and the Oracle Application Server.

Configuration Overview

Step 1. Install your software.

Step 2. Configure your Oracle Application Server. Refer to your OAS documentation.

Step 3. Configure the Reports Web Cartridge on the same machine as your Oracle Application Server.

Step 4. Configure the Reports Server.

Step 5. Make reports available to your users.

Step 6. (Optional) Further customize the Reports Server configuration.

Step 1. Install your software

Before you begin, you have to:

1. Install your Oracle Application Server, Release 4.0 and the Oracle Reports Thin Client component on the Web server machine.
2. Install the Oracle Reports Multitier Server component and the Report Builder on the Reports Server machine. Note that the Report Builder is required only if you need to build reports. If the Reports Server machine is used only as a deployment machine, then do not install the Report Builder component on the Reports Server machine.

Refer to your OAS documentation for more information on installing the Oracle Application Server.

Refer to the *Getting Started* manual for more information on installing the Oracle Reports Thin Client and Multitier Server component and the Report Builder.

Tip: To install just the Reports Thin Client, Reports Multitier Server, or Report Builder start the Oracle Installer and choose the Custom Installation option. From the Available Products list box, expand the Oracle Developer - Reports node and choose Reports Thin Client, Reports Multitier Server, or Report Builder as desired. Proceed with the installation as described in the *Getting Started* manual.

Step 2. Configure your Web server

This step is performed on your Web server machine.

To configure your Web server:

For more information on these steps, refer to your OAS server documentation.

1. Start the OAS Node Manager.
2. Start your browser.
3. Point your browser to the OAS Node Manager port. Expand the site and HTTP listeners node.
4. Create a listener if necessary, or expand the listener you want use.
5. Click **Directory** and configure the Oracle Application Server directory mapping:

Directory Description	Physical directory example	Virtual directory example	Permissions required
Cache	C:\OAS\CACHE	/CACHE	read

Table 5: OAS physical and virtual directories

To create a default key mapping file:

1. Copy and rename the ORACLE_HOME\REPORT60\cgicmd.dat file to ORACLE_HOME\REPORT60\owscmd.dat.




Note: The owscmd.dat is required for configuring database user authentication and for using key map entries. See the **Implement database user authentication** section for more information on database user authentication. See the **Map URL parameters** section for more information on key map entries.

Step 3. Configure the Reports Web Cartridge

This step is performed on your Web server machine.

To configure the Reports Web Cartridge (on Windows NT and Unix)

The steps that follow assume that you have already configured Oracle Application Server, Release 4.0. For more information, refer to your Oracle Application Server documentation.

1. In the Oracle Application Server Welcome page, click **OAS Manager**.
2. In the OAS Manager navigational tree, click the + icon beside the web  site icon.
3. Click the Applications  icon.
4. Click the Add  icon to display the Add Application form.
5. In the Add Application form, Application Type field, specify C Web.
6. Click **Apply**.
7. In the Add form, Application Name field, specify the name of your application (e.g., Reports).

8. In the Display Name field, specify the display name (e.g., Oracle Reports).
9. In the Application Version field, specify a version number (e.g., 6.0).
10. Click **Apply**.
11. In the Success dialog, click **Add Cartridge** to this Application.
12. In the Add C Web Cartridge form, Cartridge Name field, type: Reports60Cartridge
13. In the Display Name field, type: Reports60Cartridge
14. In the Cartridge Shared Object field, specify the location of the shared object file (rrows60.dll).
For example, type: D:\ORANT\BIN\RROWS60.DLL
15. In the Entry Point (Shared Object) field, type: rrows_start
16. In the Virtual Path field, type: /rrows
17. Click **Apply**.
18. To set cartridge configuration properties navigate to your application's Cartridge Configuration folder by expanding the appropriate tree nodes in the Navigator.
19. Within the Cartridge Configuration folder, click **Cartridge Parameters**.
20. Define the environment variables described in the following table.

Note: Refer to your operating system's documentation to determine if the values you set are case sensitive.

Name	Description	Example Values
REPORTS60_SHARED_CACHE	<p>This environment variable specifies whether the location of the Reports Server's cache is shared with the Web server's cache.</p> <p>Values are YES or NO.</p> <p>When you set REPORTS60_SHARED_CACHE, REPORTS60_PHYSICAL_MAP, and REPORTS60_VIRTUAL_MAP, the Reports Server ignores REPORTS60_WEBLOC and REPORTS60_WEBLOC_TRANSLATED.⁶</p>	NO
REPORTS60_VIRTUAL_MAP	<p>Specifies where the Web server looks for the Reports Server output (file cache). It is a virtual directory, as defined in the Web server configuration file. It may be mapped directly to the Reports Server file cache directory (defined in the .ORA file for the Reports Server being used) only if that directory is mapped from the</p>	<p>/CACHE</p> <p>Note: You may need to include another slash (e.g., /CACHE/) depending on the operating system you are using. Refer to your operating system's</p>

⁶ The REPORTS60_WEBLOC and REPORTS60_WEBLOC_TRANSLATED environment variables are provide for backward compatibility only. We recommend that you do not use them.

Name	Description	Example Values
	<p>Web server machine.</p> <p>REPORTS60_VIRTUAL_MAP is usually mapped to the physical directory defined by the REPORTS60_PHYSICAL_MAP parameter, which is valid regardless of whether the Reports Server and the Web server share the same file system.</p> <p>If this parameter is not set, the Web Cartridge will not display HTML/PDF output in the client browser ("Cannot access report output" message will result), although the report will run and output will be produced.</p> <p>When you set REPORTS60_SHARED_CACHE, REPORTS60_PHYSICAL_MAP, and REPORTS60_VIRTUAL_MAP, the Reports Server ignores REPORTS60_WEBLOC and REPORTS60_WEBLOC_TRANSLATED.</p>	documentation for more information.)
REPORTS60_PHYSICAL_MAP	<p>Specifies which directory the report output will be transferred to on the Web server machine if the Reports Server and Web server do not share file systems. This must be the absolute physical path of the virtual directory specified in REPORTS60_VIRTUAL_MAP and defined in the Web listener configuration.</p> <p>If this variable is set to an invalid value, the Reports Web Cartridge will be not able to display HTML/PDF output in the client browser ("Cannot access report output" message will result), although the report will run and output will be produced.</p> <p>When you set REPORTS60_SHARED_CACHE, REPORTS60_PHYSICAL_MAP, and REPORTS60_VIRTUAL_MAP, the Reports Server ignores REPORTS60_WEBLOC and REPORTS60_WEBLOC_TRANSLATED.</p>	C:\OAS\CACHE

Table 6: Parameters for the Reports Web Cartridge

25. Click **Apply**.

26. In the Success dialog box, click **OK**.

27. Within the Cartridge Configuration folder, click **Tuning**.
28. In the Maximum Number of Instances field, type: 100.
29. Click **Apply**.
30. In the Success dialog box, click **OK**.

Step 4. Configure the Reports Server

You can run the Reports Server on Windows NT or Unix. On Windows NT, you can configure it as an NT service or not as a service. The sections that follow describe the setup process for each of these possibilities.

This step is performed on your Reports Server machine.

To create a tnsnames.ora entry for the Reports Server

Before you configure the Reports Server, you must give it an entry in your tnsnames.ora file.

1. Add the following Reports Server entry to tnsnames.ora:

```
repserver.world =(ADDRESS = (PROTOCOL = TCP)(Host = repserver_machine.mydomain)
(Port = 1949))
```

where:

repserver.world is the name of the server instance and *.world* is the domain specified in the NAMES.DEFAULT_DOMAIN setting in the sqlnet.ora file. If the NAMES.DEFAULT_DOMAIN setting is not defined in the sqlnet.ora, then omit *.world* from the name of the server instance.

repserver_machine.mydomain is the host name or IP address of the machine.

1949 is the port number to which the server is listening.

2. Repeat steps 1 and 2 on the Web server machine.

To configure the Reports Server on Windows NT as a service:

1. Create an NT service by running the following command line:

```
rwmts60 -install repserver tcpip
```

Tip: Repserver does not need to have the domain qualifier appended to it (e.g., .world).

2. From the Start menu, choose **Start**→**Settings**→**Control Panel**, then double-click **Services**.
3. Within the Service dialog, select the Reports Server60 service entry.
4. From the startup dialog, select This Account in the Log On As section and type or select an operating system user name and password on that machine. This specifies that the server is run as that user.

Important: Ensure that the user running the Reports Server service has access to a default printer.

5. Set the service to be started automatically (when the system is booted up).
6. Click **OK**.
7. Click **Start**.

To configure the Reports Server on Windows NT as a non-service

1. Run the following command line:

```
rwmts60 -listen repserver
```

To configure the Reports Server on Unix

1. Set the TNS_ADMIN environment variable to point to the location of tnsnames.ora (e.g., \$ORACLE_HOME/NETWORK/ADMIN).
2. Set the REPORTS60_PATH environment variable to locate the reports (e.g., \$ORACLE_HOME/MY_REPORTS).
3. Run the following command line to start the Reports Server:

```
rwmts60 name=repserver
```

Tip: You may want to create a shell script that sets the environment variables and launches RWMTS60. Printers must be configured in the uiprint.txt file.

Step 5. Make your reports available to end users

1. Create and save your .RDF files in Report Builder.
2. Copy your .RDF files to a directory that is accessible to the Reports Server. The Reports Server looks for files using the SOURCEDIR parameter in its configuration file (e.g., repserver.ora) or, if SOURCEDIR is not specified, using the REPORTS60_PATH environment variable.
3. Provide users with an HTML page that contains URLs to run reports using the Reports Server. An example of such a page is found in
ORACLE_HOME\REPORT60\DEMO\WEBFILES\replist.htm.

Step 6. (Optional) Customize the Reports Server

There are a number of ways in which you can refine your Reports Server configuration:

- implement database user authentication
- map URL parameters
- modify the configuration file
- update environment variables

Implement database user authentication

You can configure the Reports Web Cartridge to prompt users for their database logons when they run reports. Once users successfully log on, their browser is sent an encrypted cookie. By default, this cookie expires after 30 minutes. When a cookie has expired, subsequent requests (i.e., ones that are configured for database logon) must be re-authenticated. Note that the REPORTS60_COOKIE_EXPIRE environment variable sets the expire time of the cookie. See Table 8: Environment variables for Web Cartridge for more information.

1. In the key mapping file (owscmd.dat), search for the dummy key and change the SERVER parameter to the tnsname of your Reports Server: An example of the dummy key is shown below:

```
dummy: dummy.rdf destype=cache desformat=html server=repserver
```

Note: The Reports Server uses this dummy report to verify the user's logon information. The dummy.rdf should be located in the ORACLE_HOME\REPORT60 directory and defined in the REPORTS60_PATH environment variable.

2. If you want users to authenticate *every time* they run a particular report, include the AUTHTYPE=D argument in the report URL or include the %D argument in the key map entry in the key mapping file (owscmd.dat).

If you want users to authenticate and *remain authenticated* until the cookie expires, omit the USERID parameter.

Note: To hide parameters in your URL requests, you could use a key mapping file entry. Refer to the Map URL Parameters section.

Database Authentication Demo

For your convenience, sample templates (welcome.htm and replist.htm) have been provided to demonstrate the various run requests. If you wish to use these sample templates to help you implement database authentication, do the following:

1. Copy the files (welcome.htm, replist.htm, and associated .GIFs) located in the ORACLE_HOME\REPORT60\DEMO\WEBFILES directory to the HTML document directory defined in your Web server configuration. You can modify the templates to your own UI specifications as desired.
2. In welcome.htm, search for `http:`, and change the URLs to point to your Web script directory.
3. In replist.htm, search for `http:`, and change the URLs to point to your Web script directory. Additionally, search for `report=`, and change the reports to your .RDFs.

Map URL parameters

To simplify or hide parameters in your URL requests, you can use a key mapping file. The mapping file takes a URL parameter and maps it to command line arguments that govern the running of the report. For example, one parameter in the URL could map to all of the command line arguments needed to run the report. By using key mapping, the command line arguments are all hidden from the user.

1. On the Web server machine, modify the key mapping file, ORACLE_HOME\REPORT60\owscmd.dat. Except for the special parameters that are described in the file itself, the command line arguments follow the syntax rules of RWCLI60, which is described in the Report Builder online help.

Tip: Type: `http://your_webserver/bin/rwows/showmap?` in your Web browser to verify the mapping file that is being used.

2. Update the hyperlinks in your Web page (e.g., replist.htm) to use the keys you defined in owscmd.dat.

Modify the Reports Server configuration file

1. Open ORACLE_HOME\REPORT60\SERVER\repserver.ora and modify the parameters as desired.

Parameters	Description
IDENTIFIER	IDENTIFIER is an internal setting that contains the encrypted queue administrator user ID and password. You should not attempt to modify it. If IDENTIFIER is not specified or is deleted, or the configuration file is not present, anyone can supply any user ID and password from the Reports Queue Manager to log on as the queue administrator. Once someone has logged in this way, the user ID and password they specified becomes the queue administrator user ID and password until it is changed from the Queue Manager.
MAXCONNECT	MAXCONNECT is the maximum number of processes that can communicate with the server process at any one time. This setting is the sum of the number of engines and clients, and must be greater than two (at least one engine and one client). Default value=20
SOURCEDIR	SOURCEDIR is a path to be searched before REPORTS60_PATH when searching for reports and other runtime files. This setting is useful when you have more than one Reports Server sharing the same ORACLE_HOME because each Reports Server can search different directories.
CACHEDIR	CACHEDIR is the cache for the Reports Server. CACHEDIR can be set to any directory or logical drive on the machine. If it is not specified, the default is ORACLE_HOME\REPORT60\SERVER\CACHE.
TEMPDIR	TEMPDIR is a directory that will be used instead of REPORTS60_TMP when creating temporary files. TEMPDIR can be set to any directory or logical drive on the machine.
CACHESIZE	CACHESIZE is the size of the cache in megabytes. If you expect to store the output of many of your reports in the Reports Server cache, you may want to increase this setting. If you do not expect to store a lot of output in the cache and have limited system resources, you may want to reduce it. Once the cache grows beyond the set size, the Reports Server cleans up the cached files on a first in, first out basis. Default value=50
MINENGINE	MINENGINE is the minimum number of runtime engines the Reports Server should have available to run reports. The server process will keep at least this many engines active. Ensure that you have sufficient memory and resources available to accommodate this many engines. Default value=0

Parameters	Description
MAXENGINE	<p>MAXENGINE is the maximum number of runtime engines available to the Reports Server to run reports. The server process will keep no more than this many engines active. Ensure that you have sufficient memory and resources available to accommodate this number of engines.</p> <p>Default value=1</p>
INITENGINE	<p>INITENGINE is the initial number of runtime engines started by the Reports Server. The server process will spawn this many engines when it is started. It will wait two minutes for these engines to connect to it and will shut itself down if they fail to do so. If the engines cannot connect in this amount of time, there is usually some setup problem.</p> <p>Default value=1</p>
MAXIDLE	<p>MAXIDLE is the maximum amount of time an engine is allowed to be idle before being shut down. Note that the Reports Server will not shut down the engine if doing so would reduce the number of available engines to less than the number defined in the MINENGINE.</p> <p>Default value=30</p>
SECURITY	<p>SECURITY is the security level (0, 1, 2, or 3) for accessing cached output files through the Reports Queue Manager. 0 means that anyone can access a job's cached output. 1 means that only a user whose user ID is identical to that of the user who ran the job can access the job's cached output. 2 means that only the same process that sent the job can access the job's cached output. 3 means that the cached output cannot be accessed.</p> <p>Default value=1</p>
ENGLIFE	<p>ENGLIFE is the maximum number of reports that an engine will run before shutting itself down. The Reports Server will then bring up fresh engines for new requests.</p> <p>Default value=50</p>
FAILNOTEFILE	<p>FAILNOTEFILE is path and file name of the notification message template that is sent to specified email addresses for jobs that fail to run.</p>
SUCCNOTEFILE	<p>SUCCNOTEFILE is the path and file name of the notification message template that is sent to specified email addresses for jobs that run successfully.</p>

Parameters	Description
CLUSTERCONFIG	<p>CLUSTERCONFIG is the configuration of slave servers to the master server. Clustering allows you to run reports on multiple Reports Servers. The master server can identify available slave servers and start their engines as needed. You can set up many servers as slaves to the master server.</p> <p>Use the following syntax in the master server configuration file:</p> <pre>Clusterconfig="(server=<servername> minengine=<minimum number of master engines> maxengine=<maximum number of master engines> initengine=<initial number of master engines> cachedir=<directory of central cache>)"</pre> <p>For more information, refer to the Oracle Reports 6.0 Reports Server Clustering technical white paper.</p>
LOGOPTION	<p>LOGOPTION is the type of log information you want inserted into the log file. The options are alljob, failedjob, succeededjob.</p>
PERSISTFILE	<p>PERSISTFILE indicates the location of the Reports Server's .DAT file, which contains the details of scheduled jobs. If PERSISTFILE is not specified, the default is ORACLE_HOME\REPORT60\SERVER.</p>
MAILPROFILE	<p>(Windows NT only) MAILPROFILE specifies the mail profile and password to be used when mailing reports.</p>

Table 7: Reports Server configuration file parameters

Update environment variables

1. Modify any of the following environment variables to customize your configuration.

Variable	Description
REPORTS60_COMMON_AUTH	<p>Specifies the template used to determine the common properties of the authentication form, such as the size of the form and whether a toolbar is visible. This template is used in conjunction with the HTML template set in the REPORTS60_DB_AUTH environment variable.</p> <p>Default value=common.htm</p> <p>It is recommended that you keep the default.</p>
REPORTS60_COOKIE_EXPIRE	<p>Determines the idle time of the cookie for database authentication in minutes.</p> <p>Default value=30</p> <p>Cookies save encrypted user names and passwords on the client-side when users first authenticate themselves. When the server receives a cookie from the client, the server compares the time saved in the cookie with the current system time. If the time is longer than the number minutes defined in the REPORTS60_COOKIE_EXPIRE environment variable (e.g., 30 minutes), the server rejects the cookie and returns the database authentication form and an error message to the client. The user must re-authenticate to run the report.</p>
REPORTS60_DB_AUTH	<p>Specifies the database authentication template used to authenticate the user name, password, and database.</p> <p>Default value=dbauth.htm</p>
REPORTS60_ENCRYPTION_KEY	<p>Specifies the encryption key used to encrypt the user name and password for the cookie. The encryption key can be any character string.</p> <p>Default value= reports6.0</p>
REPORTS60_SSLPORT	<p>If you are using SSL and you want to use a port number other than 443, you can use this variable to set a different port number.</p> <p>Default value= 443</p>
REPORTS60_OWSDIAGBODYTAGS	<p>For the Reports Web Cartridge, specifies HTML tags that will be inserted as a <BODY...> tag in the RWOWS diagnostic/debugging output. For instance, you may want to use this environment to set up text/background color, image, etc.</p>

Variable	Description
REPORTS60_OWSDIAGHEADTAGS	For the Reports Web Cartridge, specifies HTML tags to insert between <HEAD> ...</HEAD> tags in the RWOWS diagnostic/debugging output. For instance, you may want to use this environment to set up <TITLE> or <META...> tags.
REPORTS60_OWSHELP	For the Reports Web Cartridge, defines URL/URI of the RWOWS help file, which will be navigated to when RWOWS is invoked with the empty request: http://your_webserver/rwows?. For example, setting it to http://www.yahoo.com will go to that URL; setting it to myhelpfile.htm will display the file: http://your_webserver/myhelpfile.htm. If this parameter is not defined, a default help screen will be displayed.
REPORTS60_OWSMAP	For the Reports Web Cartridge, defines fully qualified filename/location of the RWOWS map file (if map file configuration is used), e.g., C:\ORANT\REPORT60\OWSCMD.DAT.
REPORTS60_OWSNODIAG	For the Reports Web Cartridge, when defined, disables all debugging/diagnostic output, such as help and showmap, from RWOWS. For example, http://your_webserver/rwows/help? will not work when REPORTS60_OWSNODIAG is defined.

Table 8: Environment variables for Web Cartridge

TROUBLESHOOTING

Problem Description	Probable Cause and Solution
Reports Server appears to hang when you start it.	You may have made a syntactical error in the tnsnames.ora file and the Reports Server cannot resolve the tnsname. Alternatively, you could try rebooting in case the cause is a memory problem.
You get the error “Daemon failed to listen to port.”	If you start up a Reports Server that is listening to the same port as an already running Reports Server, you will get this error. It could also be a problem with your Net8 or TCP/IP setup.
You get an error about being unable to initialize the printer (REP-3002)	Make sure that the Reports Server has access to printers. On Windows NT, the System Account does not usually have access to printers. It could be that you installed the Reports Server as an NT service and used the System Account or another account without printer access in the Log On As field. You must specify an account in the Log On As field that has a default printer access. Note that this printer does not have to exist, but the driver must be installed. For Unix, the printer must be configured in the uiprint.txt file.

Problem Description	Probable Cause and Solution
<p>Upon starting the Reports Server, you get server specific error 186.</p>	<p>Typically this indicates a problem in tnsnames.ora or sqlnet.ora. Check the entry for the Reports Server in tnsnames.ora. A typical entry should look something like the following:</p> <pre data-bbox="646 390 1247 453">repserver.world = (ADDRESS=(PROTOCOL=tcp) (HOST=144.25.87.182)(PORT=1951))</pre> <p>Note that in this example <i>.world</i> is appended to the name because it is the domain specified in the sqlnet.ora file. If the NAMES.DEFAULT_DOMAIN setting is not defined in the sqlnet.ora, then omit <i>.world</i> from the name of the server instance.</p> <p>If your tnsnames.ora file appears to be correct, check your sqlnet.ora file. Good default settings to use in this file are:</p> <pre data-bbox="646 695 1130 831">TRACE_LEVEL_CLIENT=OFF names.directory_path = (TNSNAMES) names.default_domain = world name.default_zone = world</pre> <p>If your protocol is TCP, ensure that the Net8 TCP/IP adapter and Net8 have been installed.</p> <p>Lastly, be sure that your installed version of Net8 is not older than the version that came with the Reports Server.</p>
<p>Error reported when opening the report.</p>	<p>Check the name and extension carefully. On UNIX machines, the actual report name must be in the same case as specified in the URL. If you are using Windows Explorer in Windows, be sure not to hide extensions for the displayed files that you are copying and renaming. (Check View→Options in the Explorer window.) This prevents you from creating files with names like your_report.rdf.txt. Alternatively, use a DOS window for file manipulation.</p> <p>Alternatively, ensure that the report is located in the path defined by the REPORTS60_PATH environment variable.</p>

Problem Description	Probable Cause and Solution
Problems running the Reports Server as a Windows NT Service.	<p>If you install the Reports Server service to run under a user other than SYSTEM, make sure that the user account:</p> <ul style="list-style-type: none"> • Has the Password Never Expires option selected in the User Manager. • Has membership in the appropriate groups to run the Reports Server and access the report files. • Has at least print permission to a default printer. • Can log on to a service. Choose Start→Programs→Administrative Tools→User Manager, then Policies User Rights. Check Show Advanced User Rights. From the Right list, choose Log on as a service. If the user is not already in the Grant To list, click the Add. <p>When starting the service, you may need to explicitly specify the domain as well as the user name (username/domain). If you get a Windows NT error reporting that the service failed and returning the error message number, you can look up the message number in the Report Builder online help.</p>
ops\$ account is not working.	<p>For security reasons, ops\$ accounts are not supported by the Reports Server. If you pass a command line with USERID=/ to the Reports Server, an error is generated because it will try to use the user name of the Reports Server process rather than the user name of the client.</p>
Database roles not working as expected.	<p>If you are using database roles, the Reports Server gets and then sets the default roles for the job request's database connection. If the default roles require a password, the Reports Server will log off and then back on to the database. As a result, it is best to include roles that require passwords in the report itself using the Role Name report property. Since the Reports Server gets and then sets the default roles on a per job basis, you cannot share roles between jobs. This is done to preserve security.</p>
PDF reports run, but HTML reports generate an error saying that the requested URL was not found.	<p>It could be that REPORTS60_PHYSICAL_MAP points to the same location as the Reports Server cache (i.e., CACHEDIR in the Reports Server configuration file). In this scenario, the Reports Web CGI is trying to copy the report output from the cache directory to the root directory of the Web listener, but, since those directories are the same, you receive an error.</p> <p>Check your REPORTS60_PHYSICAL_MAP environment variable. If CACHEDIR is the same as your REPORTS60_PHYSICAL_MAP, set REPORTS60_SHARED_CACHE to Yes. If CACHEDIR is different, change the value of REPORTS60_PHYSICAL_MAP to the absolute physical path of the virtual directory defined in REPORTS60_VIRTUAL_MAP.</p>

Problem Description	Probable Cause and Solution
URL mapping is not working.	<p>Make sure you have a valid key mapping file. It must either be named owscmd.dat (for the Reports Web Cartridge) or cgicmd.dat (for the Reports Web CGI) in the REPORT60 directory, or named according to the value set in the REPORTS60_OWSMAP or REPORTS60_CGIMAP environment variable.</p> <p>To ensure the key mapping file can be found, first try (the following is a Web CGI example):</p> <pre>http://your_webserver/your_virtual_cgi_dir/rwcgi60.exe/showmap?</pre> <p>and verify that your key entry has been correctly parsed in the resulting page.</p> <p>Then try, running the report using the key map entry:</p> <pre>http://your_webserver/your_virtual_cgi_dir/rwcgi60.exe?your_key</pre> <p>where your_key is a valid key entry in the key mapping file.</p>
Cannot shutdown the queue from the Reports Queue Manager.	<p>You should not leave the username and password blank the first time that you log in as the administrator. The first time that you log in as the queue administrator from the Reports Queue Manager (Options→Privileges→Administrator), you can specify any username and password. The username and password that you specify the first time will be the administrator's until you change it.</p>
Cannot run the Reports Server as an NT Service under LocalSystem.	<p>If the Reports Server is to be run as an NT service under the LocalSystem userid, then the system administrator must ensure that the following line is in the sqlnet.ora file, otherwise the server cannot be accessed:</p> <pre>sqlnet.authentication_services=(NONE)</pre>
Problems finding files.	<p>Since network drives are mapped to a drive letter on a per-user basis, these mappings are no longer in effect when the NT user logs off. The Reports Server must not refer to these drives through their drive letters. Instead you should use UNC path names (e.g., \\sales\documents\reports). This applies to Reports Server parameters, Web Cartridge and Web CGI command mappings, and each hard-coded path name in each report being run.</p>
The Web server reports an error opening the report output.	<p>If the Web server reports an error opening the report output, check the name and extension carefully. On UNIX machines, the actual report name must have the same case as specified in the URL. If you are on Windows using the Windows Explorer, be sure not to hide extensions for displayed files (View→Options in the Explorer window) that you are copying and renaming. This prevents you from creating files with names like your_report.rdf.txt. Alternatively, use a DOS window for file manipulation.</p>
Report runs fine on design platform (e.g., Windows), but fails on server platform (e.g., UNIX).	<p>Check whether the release you are using on the design platform is the same as that on the server. If they are not the same, it could be that a difference between the two releases is causing the problem.</p>

Problem Description	Probable Cause and Solution
<p>Reports are not running when the URL is requested.</p>	<ul style="list-style-type: none"> • Make sure the Web server is responding (e.g., by trying to bring up your Web server administration page). Refer to your Web server installation documentation. • Make sure your Web Cartridge or Web CGI executable has been found and is responding. For Windows 95/NT, type one of the following: <pre>http://your_webserver/your_virtual_cgi_dir/rwcgim60.exeor,</pre> <pre>http://your_webserver/rwows</pre> <p>For UNIX, type: <pre>http://your_webserver/your_virtual_cgi_dir/rwcgi60 or,</pre> <pre>http://your_webserver/rwows</pre> <p>in your browser URL field. A help page should appear. If it does not, check the mapping of your_virtual_cgi_dir (usually called cgi-bin) in your Web server configuration file. It should be mapped to an existing physical directory on your Web server. You must have a copy of the RWCGI60 executable in this physical directory.</p> </p> • Make sure that the REPORTS60_OWSNODIAG (for Web Cartridge) or REPORTS60_CGINODIAG (for Web CGI) environment variable is <i>not</i> defined, otherwise all diagnostic output will be disabled. Test this by typing one of the following: <pre>http://your_webserver/your_virtual_cgi_dir/rwcgi60.exe/showenv?</pre> <pre>http://your_webserver/rwows/showenv?</pre> <p>This will also allow you to view the other cartridge parameters or environment variables.</p> • Check the list of cartridge parameters or environment variables and make sure that REPORTS60_SHARED_CACHE, REPORTS60_VIRTUAL_MAP, and REPORTS60_PHYSICAL_MAP are defined. The former is defined to be a virtual directory on the Web server; the latter is defined to be the absolute path of the directory to which the virtual directory maps. • Make sure the REPORTS60_PATH environment variable is defined. Check the environment variable by typing one of the following: <pre>http://your_webserver/you_virtual_cgi_dir/rwcgi60.exe/showenv?</pre> <pre>http://your_webserver/rwows/showenv?</pre>

Problem Description	Probable Cause and Solution
	<ul style="list-style-type: none"> • Try running a simple report to your browser, by typing one of the following: <pre data-bbox="760 310 1565 541"> http://your_webserver/your_virtual_cgi_dir/rwcgi60.exe ?server=your_repserver+report=your_report.rdf+ userid=scott/tiger@mydb+desformat=html http://your_webserver/rwows?server=your_repserver+ report=your_report.rdf+userid=scott/tiger@mydb+ desformat=html </pre> <p data-bbox="688 562 1253 594">If the report does not display, check to be sure that:</p> <ul style="list-style-type: none"> ◦ Your_report.rdf runs correctly from Report Builder or Reports Runtime Your_report.rdf is located in a directory specified under REPORTS60_PATH. ◦ The database connection string is correct. <p data-bbox="688 793 1503 863">Remember that the Reports Server must have access to the report and any external files used by the report.</p> <p data-bbox="688 884 1562 1098">When sending a report to the Reports Server, you should only use the In Report value for parameters if they have their values explicitly set in the report definition. For example, suppose that you are launching a report from the Reports Queue Manager (Job→New). If you specify In Report for the Report Mode and Orientation parameters, and neither of them has a value specified in the report definition, the job will fail.</p>

Table 9: Troubleshooting information

RELATED DOCUMENTATIONS

For more information about the Reports Server, refer to the following documents:

[Oracle Developer Guidelines for Building Applications](#), Part No. A58766-02

Oracle Developer Server online configuration guide, odcfg.htm, which is installed in
ORACLE_HOME/TOOLS/DOC60/us

Oracle Reports 6.0 Reports Server Clustering technical white paper

GLOSSARY

Authentication

The process of verifying the identity of a user, device, or other entity in a computer system, often as a prerequisite to allowing access to resources in a system.

Cache

A temporary storage place for database data that is currently being accessed or changed by users, or for data that Oracle Server requires to support users. The terms are often used interchangeably.

Cartridge

A manageable object that plugs in to the client, application server, or database through an open API.

CGI

Acronym for Common Gateway Interface. The industry-standard technique for running applications on a Web server. Whereas standard HTML documents retrieved from a Web server are static (the exact same text is retrieved every time) CGI enables a program running on the Web server to communicate with another computer to generate "dynamic" HTML documents in response to user-entered information.

CSS

Acronym for Cascading Style Sheets. HTML with CSS allows developers to control the style and layout of multiple Web pages all at once. A style sheet works like template, a collection of style information, such as font attributes and color. Cascading refers to a set of rules that Web browsers use to determine how to use the style information. Navigator 4.0 and Internet Explorer 4.0 support cascading style sheets.

Domain

A grouping of network objects, such as databases, that simplifies the naming of network services.

HTML

Acronym for Hypertext Markup Language A tag-based ASCII language used to specify the content and links to other documents on Web servers on the Internet. End users with Web browsers view HTML documents and follow links to display other documents.

HTTP

Acronym for Hypertext Transfer Protocol. The protocol used to carry Web traffic between a Web browser computer and the Web server being accessed.

IP

Acronym for Internet Protocol. The basic protocol of the Internet. It enables the delivery of individual packets from one host to another. It makes no guarantees about whether or not the packet will be delivered, how long it will take, or if multiple packets will arrive in the order they were sent. Protocols built on top of this add the notions of connection and reliability.

Net8

Oracle's remote data access software that enables both client-server and server-server communications across any network. Net8 supports distributed processing and distributed database capability and runs over and interconnects many communication protocols.

Oracle Application Server

Oracle Application Server is a strategic platform for network application deployment. By moving application logic to application servers and deploying network clients, organizations can realize substantial savings through reduced complexity, better manageability, and simplified development and deployment. Oracle Application Server provides the only business-critical platform that offers easy database web publishing and complete legacy integration while transition from traditional client-server to network application architectures.

ORACLE_HOME

An alternate name for the top directory in the Oracle directory hierarchy on some directory-based operating systems. An environment variable that indicates the root directory of Oracle products.

PDF

Acronym for Portable Document Format. A file format (native for Adobe Acrobat) for representing documents in a manner that is independent of the original application software, hardware, and operating system used to create the documents. A PDF file can describe documents containing any combination of text, graphics, and images in a device-independent and resolution independent format.

Port

A number that TCP uses to route transmitted data to and from a particular program.

Reports Queue Manager

Enables you to monitor and manipulate job requests that have been sent to the Reports Server.

Reports Launcher

An application that utilizes the functionality provided by the Reports ActiveX control, such submitting a request to run the specified report to the Report Server.

Reports Server

Enables you to run reports on a remote server in a multi-tier architecture. It can be installed on Windows NT, Windows 95, or Unix. The Reports Server handles client requests to run reports by entering all requests into a job queue.

Reports Web Cartridge

An interface between your Oracle Application Server and Reports Runtime, enabling you to run reports dynamically from your Web browser.

RWCLI60

Parses and transfers the command line to the specified Reports Server (RWMTS60).

TCP/IP

Acronym for Transmission Control Protocol based on Internet Protocol (IP). An Internet protocol that provides for the reliable delivery of streams of data from one host to another.

Tnsnames.ora

An Net8 file that contains connect descriptions mapped to service names. The file may be maintained centrally or locally, for use by all or individual clients.

URI

Acronym for Uniform Resource Identifier. A compact string representation of a location (URL) for use in identifying an abstract or physical resource. URI is one of many addressing schemes, or protocols, invented for the Internet for the purpose of accessing objects using an encoded address string.

URL

Acronym for Uniform Resource Locator. A URL, a form of URI, is a compact string representation of the location for a resource that is available via the Internet. It is also the text-string format clients use to encode requests to Oracle Application Server.

Web browser

A program that end users utilize to read HTML documents and programs stored on a computer (serviced by a Web server).

Web Server

A server process (HTTP daemon) running at a Web site which sends out Web pages in response to HTTP requests from remote Web browsers.

INDEX

- A**
- architecture, 2
 - non-Web, 6
 - Web, 3
 - authentication, database
 - CGI, 16
 - Web cartridge, 29
- C**
- cache
 - sharing, 9
 - case study
 - CGI, 10
 - Web cartridge, 23
 - concepts, 8
 - configuration
 - CGI case study, 10
 - choices, 8
 - parameters, 18
 - steps for CGI, 11
 - steps for Reports Server, 13
 - steps for Web cartridge, 25
 - Web cartridge case study, 23
- D**
- database authentication
 - CGI, 16
 - Web cartridge, 29
 - deploying
 - reports with CGI, 15
 - documentation, related, 41
- E**
- environment variables, 12
 - CGI, 21
 - Reports Web Cartridge, 33
- G**
- glossary, 42
- I**
- introduction, 2
- K**
- key mapping
 - CGI, 17
 - Web cartridge, 30
- M**
- mapping, key
 - CGI, 17
 - Web cartridge, 30
- N**
- non-Web
 - architecture, 6
 - processing reports, 7
 - requests, enabling, 8
- O**
- Oracle Application Server
 - configuration of cartridge, 25
- P**
- parameters
 - Reports Server, 18
 - processing non-Web reports, 7
 - processing Web reports, 6
- R**
- registry variables, 12
 - CGI, 21
 - related documentation, 41
 - reports
 - deploying with CGI, 15
 - non-Web processing, 7
 - Web processing, 6
 - Reports CGI
 - case study, 10
 - choice, 8
 - steps for configuration, 11
 - Reports Server
 - architecture, 2
 - cache sharing choice, 9
 - CGI case study, 10
 - configuration choices, 8
 - configuration file, 18
 - introduction, 2
 - location choice, 9
 - steps for configuration, 13
 - Web cartridge case study, 23
 - Reports Web Cartridge
 - case study, 23
 - choice, 8
 - environment variables, 33
 - steps for configuration, 25

S

sharing cache, 9

T

terminology, 42

troubleshooting, 36

W

Web

architecture, 3

cache sharing, 9

cartridge choice, 8

CGI choice, 8

processing reports, 6

requests, enabling, 8



Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
+1.650.506.7000
Fax +1.650.506.7200
<http://www.oracle.com>

Oracle Reports Release 6.0: Reports Server Configuration
Technical White Paper
Authors: Carol Menzigian, Frank Rovitto
Contributors: Chan Fonseca, Shaun Lin, Paul Narth,
Ashok Natesan, Danny Richardson

Copyright © Oracle Corporation 1999
All Rights Reserved

This document is provided for informational purposes only, and the information herein is subject to change without notice. Please report any errors herein to Oracle Corporation. Oracle Corporation does not provide any warranties covering and specifically disclaims any liability in connection with this document.

Oracle is a registered trademark.

All other company and product names mentioned are used for identification purposes only and may be trademarks of their respective owners.