

Oracle9i Reports Developer

Release Notes

Release 2 (9.0.2)

April 2002

Part No. A96189-01

This document summarizes the differences between Oracle9i Reports Developer and its documented functionality. For recent updates to these release notes and other Oracle9i Reports Developer documentation, please refer to the Oracle Technology Network (<http://otn.oracle.com/products/reports/>).

See Also: *Oracle9i Application Server Release Notes*

1 General Issues and Workarounds

This section describes general issues and their workarounds for Oracle9i Reports Developer.

1.1 Migration

For information about migration, please refer to *Oracle9i Application Server Migrating from Oracle9iAS Release 1 (1.0.2.2.x) to Release 2 (9.0.2)*, part number A96157-01.

1.2 Deprecated Features from Earlier Versions

Some features available in Oracle Reports 6i have been deprecated or removed from Oracle9i Reports Developer.

Following is a list of the deprecated features. Existing reports using these features will continue to run without modification, but these features are no longer documented and their further use is strongly discouraged:

- user exits
- RWCGI Web executable

ORACLE[®]

Copyright © 2002 Oracle Corporation.
All Rights Reserved.

Oracle is a registered trademark, and Oracle9i and PL/SQL are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

- command line options: CURRENCY, THOUSANDS, DECIMAL, PROFILE, ERRFILE, LOGFILE, BACKGROUND, KEYIN, KEYOUT
- SRW.SET_ATTR built-in
- OLE2 object support

Following is a list of features that have been from Oracle9i Reports Developer:

- RWRUNC character mode runtime
- client/server GUI report previewer in rwrunc
- RWRBE60 background engine
- RWOWS60 OAS cartridge
- OBE60 query builder
- OBS60 schema builder

More detailed information about these deprecated and obsolete features can be found in the Oracle Reports Statement of Direction white paper available from the Oracle Technology Network (<http://otn.oracle.com/products/reports/>).

1.3 Inserting Multiple Report Blocks In Web Source

If you are using the Report Block Wizard to insert multiple report blocks that share one or more data columns between them, the generated JSP tags will end up with duplicate IDs. These duplicate tags will cause a JSP compilation failure and the report will not execute. You can work around this issue by manually editing the Web source to make the tag IDs unique.

1.4 PDF Font Sub Setting

In Oracle9i Reports Developer, you can subset the TrueType fonts for multi-byte, Unicode, and single byte PDF output. Oracle9i Reports Developer internally converts the TrueType fonts to Adobe's Type 3 format and embeds them within the document.

To view the Type 3 fonts in the Acrobat Reader smoothly, Acrobat 5.0 provides the following workaround:

1. Open Adobe Acrobat 5.0.
2. Choose **Edit->Preferences**.
3. Choose **Display->Smoothing**.
4. Check all of the check boxes (smooth Text, smooth Artline, smooth Images).

5. If you are using a laptop, also check the CoolType check box.

1.5 XML and Text Data Sources

- For the XML and text pluggable data source Edit dialog boxes in Reports Builder, all referenced parameters in a URL should be wrapped as follows:

```
&<parameter>
```

For example, the following URL references a parameter called stock_symbol:

```
http://host.domain.com/vdir/get_stock_quote.jsp?symbol=&<stock_symbol>
```

If stock_symbol was not inside of the angle brackets, the ampersand would be read as an argument separator and stock_symbol as a report parameter.

- The XML data must follow the same order of elements as found in the DTD or XML schema file. For example, suppose your DTD or XML schema defines two elements, PRODUCT_ID coming first and WAREHOUSE_ID coming second. In this scenario, PRODUCT_ID must come before WAREHOUSE_ID in your XML data file, too.

1.6 Supported Data Types in Pluggable Data Sources

The Pluggable Data Source API supports the number, date, and string data types. An individual pluggable data source should appropriately map its types to these three types.

For example, XML schema data types would be mapped to the nearest ones of the three supported types. These types would then be passed to the XML pluggable data source. In the case of the JDBC pluggable data source, if the query tries to retrieve a BLOB column from the database, the pluggable data source driver would give an error, `Column type not supported`.

1.7 Links Between Queries Using Pluggable Data Sources

When you create an invalid link between two queries of pluggable data sources, you will receive the following error message:

```
Fail to fetch Plugin Data Source java.lang.NumberException:G
```

When you receive this error, you should check the validity of your links (for example, are the data types of the columns compatible).

1.8 Oracle9iAS Portal Portlet/Page Parameters

Within Oracle9iAS Portal, you can choose whether to make a report's parameters visible to users through the Customization page of a report portlet. Click **Customize** at the bottom of the Manage Component page for the report and click **Visible to user** for each parameter you want to expose. You can also set the default value of the parameter from this page.

If the parameter you are exposing has a corresponding Oracle9iAS Portal page parameter, the default value from the Manage Component page will be used as the default value in the Customize page for the portlet. If the user leaves the parameter value empty in the Customize page, the portlet inherits the page parameter's value. If the user enters a value for the report portlet's parameter, that value will override the page parameter value.

For more information about Oracle9iAS Portal, refer to the Oracle Technology Network (<http://portalcenter.oracle.com>).

1.8.1 Customize Page

The SSOCONN and CGI/servlet key parameters are missing from the JSP report Customize page in Oracle9iAS Portal. The work around for this issue is to create additional user parameters for the missing parameters.

1.9 HTTPS from within Oracle9iAS Portal

Oracle9i Reports Developer does not support https when running reports from within Oracle9iAS Portal. You should not use https for the Reports Server URL Gateway when defining Reports Server Access object Oracle9iAS Portal.

1.10 Reports Portlet and Netscape Communicator 4.7

In Oracle9iAS Portal with Netscape Communicator 4.7, if you add a reports portlet to a page, you will get an error message when you try to take another action such as adding another portlet. This problem does not occur in Netscape Communicator 6.0 or Microsoft Internet Explorer 5.5.

1.11 Destination Parameter and Microsoft Internet Explorer

In some cases, Microsoft Internet Explorer ignores the mimetype of a URL's return stream and instead sets the type by looking at the URL. This can be a problem when you are using the distribution feature of Oracle9i Reports Developer because your URL might end with the destination parameter, for example:

```
...distribute=yes destination=c:\oracle\reports\distribution\mydist.xml
```

In this scenario, your URL ends with the extension `xml` and Internet Explorer treats the return stream as XML, when in fact it is HTML. As a result, you will receive a browser error. To work around this issue, you should never use recognized file extensions at the end of a URL. In the example above, you could switch the positions of the `distribute` and `destination` parameters in your URL.

1.12 Templates

If you are using one of the default report templates, you cannot combine two report blocks that use different default templates in a single report. All of your report-blocks in any one report must use the same default template.

1.13 Oracle9iAS Portal Security, Portal Destination, and Job Status Repository

If you use these features, the JDBC database connections made by Oracle9iAS Reports Services may override the initial `NLS_LANG` setting. This change may in turn affect the behavior of the running report, such as bidirectional output in PDF. On UNIX platforms, you can work around this issue by setting the `NLS_LANG` explicitly in `report.sh`.

1.14 Command Line Arguments

1.14.1 SUPPRESSLAYOUT Keyword

[Table 1](#) indicates which commands can use the `SUPPRESSLAYOUT` keyword.

Table 1 *Commands that can use SUPPRESSLAYOUT*

<code>rwclient</code>	<code>rwrn</code>	<code>rwbuilder</code>	<code>rwconverter</code>	<code>rwervlet</code>	<code>rwsgi</code>	<code>rwserver</code>
yes	yes	yes	no	yes	yes	no

Description The `SUPPRESSLAYOUT` command line option prevents any paper objects in the report layout from being formatted. The option allows users to control whether the paper layout in a report is executed at runtime. The most common use of this option is to increase the performance of JSP reports. Since a JSP report may have a paper layout and reference objects in it via an `<rw:include>` tag, Reports formats the paper layout before running the JSP section of the report. To improve the performance of single source JSP reports that store both paper and Web layouts but do not reference paper layout objects, set `SUPPRESSLAYOUT=YES` on the command line.

Note: If there is an <rw:include> tag, then no output will be created for the tag.

Syntax SUPPRESSLAYOUT=[YES|NO]

Values YES means that the paper layout objects are not formatted when the report is executed. NO means that the paper layout objects will be formatted.

Default NO

1.14.2 UPGRADE_PLSQL

[Table 2](#) indicates which commands can use the UPGRADE_PLSQL keyword.

Table 2 *Commands that can use UPGRADE_PLSQL*

rwclient	rwrun	rwbuilder	rwconverter	rservlet	rwcgi	rwserver
no	no	no	yes	no	no	no

Description The UPGRADE_PLSQL command line option upgrades any PL/SQL code in the specified report to the latest version required by Reports9i Developer.

Syntax UPGRADE_PLSQL=[YES|NO]

Values YES means that the PL/SQL code will be upgraded automatically if necessary. NO means that the PL/SQL code will not be updated.

Default YES

1.14.3 RECURSIVE_LOAD

[Table 3](#) indicates which commands can use the RECURSIVE_LOAD keyword.

Table 3 *Commands that can use RECURSIVE_LOAD*

rwclient	rwrun	rwbuilder	rwconverter	rservlet	rwcgi	rwserver
yes	yes	no	yes	yes	yes	no

Description The RECURSIVE_LOAD keyword is used to determine whether Oracle9i Reports Developer should validate all of the external references of the program units in a report at runtime. If this keyword is set to YES, an invalid external reference will cause the program unit to be automatically recompiled. Setting RECURSIVE_LOAD to NO is useful

when running your report against a different database than the one against which its PL/SQL was originally compiled.

Syntax RECURSIVE_LOAD=[YES|NO]

Values YES means that the external references will be validated. NO means that the external references will not be validated.

Default YES

1.14.4 SQLTRACE

[Table 3](#) indicates which commands can use the SQLTRACE keyword.

Table 4 *Commands that can use RECURSIVE_LOAD*

rwclient	rwrun	rwbuilder	rwconverter	rwervlet	rwcgi	rwserver
yes	yes	yes	no	yes	yes	no

Description The SQLTRACE keyword enables you to perform SQL tracing on your report without having to modify the report definition.

Syntax SQLTRACE=[YES|NO]

Values YES means that SQL tracing will be performed on the report. NO means that SQL tracing will not be performed on the report.

Default NO

1.15 Built-ins

1.15.1 SRW.GET_VALUE

SRW.GET_VALUE is equivalent to the Oracle9i Forms Developer NAME_IN built-in. It permits developers to get the value of a field at runtime indirectly. This method of obtaining a field's value is useful if you are writing business logic in a PL/SQL library but need to obtain report values directly.

Instead of using :field_name, the user can use SRW.GET_VALUE(field_name) to obtain the value of a field. For example:

```
function func_one return varchar2
is
  the_fieldname varchar2(20):='ENAME';
begin
  return(srw.get_value(the_fieldname));
```

```
end func_one;
```

1.15.2 SRW.GET_REPORT_NAME

SRW.GET_REPORT_NAME can be used to obtain the file name of the report being executed. For example:

```
function AfterPForm return boolean is
    my_var varchar2(80);
begin
    srw.get_report_name(my_var);
    srw.message(0, 'Report Filename = ' || my_var);
    return (TRUE);
end;
```

1.16 Default Character Set for JSPs

By default, a new JSP created in Reports Builder contains the following:

```
<%@ page contentType="text/html;charset=ISO-8859-1" %>
```

If you are creating your JSP outside of Reports Builder, you should ensure that it contains similar encoding information.

1.16.1 REPORTS-NLS_XML_CHARSETS

Currently, some Oracle NLS_CHARSET values have no equivalent IANA character set. The XML saved by Oracle9i Reports Developer for reports with these character sets cannot be opened by some XML viewers, such as Internet Explorer, unless you set REPORTS-NLS_XML_CHARSETS to the following:

```
WINDOWS-950=BIG5;CSEUCKR=EUC-KR;
```

1.17 Java Importer

In order to run reports that rely on Java classes, you must:

- Add the necessary jar files to the REPORTS_CLASSPATH (for the Reports Builder).
- Add the necessary jar files to the classPath attribute of the engine element in the Reports Server configuration file (ORACLE_HOME\reports\conf\

1.18 Buttons

If you open an existing report that contains buttons in Oracle9i Reports Developer, the buttons will be converted into text items. You cannot add new buttons in Oracle9i Reports Developer.

1.19 Parameter Forms

If you open an existing report that contains user parameters and you save it as a JSP, the parameter form is lost. If you create a new JSP with user parameters, the parameter form will appear when you run it in the Reports Builder, but you have to create your own parameter form for runtime.

1.20 RTF Output in Microsoft Word 95 for Japanese

When you open the RTF output from Oracle9i Reports Developer in Microsoft Word 95 for Japanese, you may encounter anomalies in the output, such as dashes not appearing correctly. These issues are specific to Microsoft Word 95 and do not occur in Microsoft Word 97 for Japanese.

2 Configuration Issues and Workarounds

This section describes configuration issues and their workarounds for Oracle9i Reports Developer.

2.1 Reports Server Installed with Reports Builder

The Reports Builder requires an instance of the Reports Server. Hence, when you install the Reports Builder, the Installer prompts you for some information that it requires to configure the Reports Server, for example the name of your mail server.

2.2 REPORTS_CLASSPATH

Oracle9i Reports Developer uses the environment variable `REPORTS_CLASSPATH` when looking for Java classes; it does not use the system `CLASSPATH` variable. As a result, any JavaBeans that you want to use within Oracle9i Reports Developer must be locatable from the `REPORTS_CLASSPATH`.

`REPORTS_CLASSPATH` is limited to 511 bytes in length. For Windows systems, `REPORTS_CLASSPATH` is set in the registry. For Unix systems, it is set from the command prompt or in a shell script.

2.3 X-terminals and Graphical Terminals

If you are planning to run reports on an X-terminal or graphical terminal, the DISPLAY variable must be set appropriately. For more information on configuring for X-terminals and graphical terminals, refer to the platform-specific documentation for Oracle9iAS.

2.4 REPORTS60_DEFAULT_PIXEL_SIZE

REPORTS60_DEFAULT_PIXEL_SIZE is an environment variable that overrides the operating system's default pixel size when rendering a report. Normally, Oracle9i Reports Developer takes its pixel size from the operating system. If you are working with older reports that rely upon a pixel size that is different from that of the operating system (for example, a pixel size of 80), you can use this variable to maintain the same behavior in your older reports.

For Windows, REPORTS60_DEFAULT_PIXEL_SIZE is set in the registry. For Unix, it is set from the command prompt or in a shell script.

2.5 Oracle Reports 6i Access to Oracle9i Reports Developer

The Oracle Reports 6i executables, such as RWCLI, can send job requests to the Oracle9iAS Reports Services through a proxy server, as long as the Reports Server is not secured.

3 UI Issues and Workarounds

3.1 Source Control (Windows only)

To enable the use of Oracle Source Control Management with Oracle9i Reports Developer, you must set a number of registry variables. Oracle Source Control Management provides a Start menu item that will update the registry variables for you:

Oracle 9i Developer Suite-ORACLE_HOME -> Oracle 9i Software Configuration Manager -> Use as Source Control For Forms/Reports

This menu item runs a file named drsc61.reg. By running this file, you are updating the registry variables required by Oracle9i Reports Developer.

3.1.1 Check In/Check Out Restriction

If you are using the source control integration feature in Reports Builder, you should not select the check out after check in option. Doing so will

result in the report being placed in Read-only mode. To avoid this problem, you should always perform check ins and checkouts as separate operations.

3.2 Some Languages Not Appearing Correctly in Web Source View

In the Web Source view of the Report Editor, the following languages may appear garbled: Japanese, Thai, Arabic, and Hebrew. To work around this issue, you can set the font names for Reports Builder in `uifont.ali` as follows:

```
[rwbuilder]
....jal6sjis="MS Gothic"
....ar8mwin1256="Courier New"
```

4 Documentation Errata

This section describes known errors or omissions in the documentation.

- If you are looking for information about REP_UTIL_PPL, refer to the Getting Started on the Oracle Technology Network (<http://otn.oracle.com/products/reports/>).
- The PAGESIZE parameter cannot be set from the Runtime Parameter Form. It can only be set from the command line (e.g., `rwclient`).
- The PAGESIZE parameter's value will be applied to all sections in the report (header, body, and trailer).
- In order to run the barcode example report that is referenced in the Oracle9i Reports Developer Getting Started on the Oracle Technology Network (<http://otn.oracle.com/products/reports/>) and the *Oracle9i Reports Developer Building Reports* manual, you must:
 - Add `oraclebarcode.jar` to the REPORTS_CLASSPATH (for the Reports Builder).
 - Add `oraclebarcode.jar` to the classPath attribute of the engine element in the Reports Server configuration file (`ORACLE_HOME\reports\conf\<server_name>.conf`).
- The Reports Builder online help incorrectly states the requirements for Acrobat 4.0 for PDF output. The help states that you need Acrobat 4.0 packs in all cases when you really only need it for font aliasing:
 - If you are building a multi-byte report for multi-byte languages, such as Chinese and Japanese, and you need to alias the font in PDF output, you need the CID fonts named within the Acrobat 4.0

packs. Otherwise, you do not need the CID fonts in the Acrobat 4.0 packs.

- In Chapter 6 of *Oracle9iAS Reports Services Publishing Reports to the Web*, the method described for generating public and private keys has changed. To generate a new Reports Server key on Unix, enter the following command:

```
$ORACLE_HOME/bin/rwgenkey.sh <public_key_file> <private_key_file>
```

To generate a new Reports Server key on Windows, enter the following command:

```
$ORACLE_HOME/bin/rwgenkey.bat <public_key_file> <private_key_file>
```

- In Chapter 3 of *Oracle9iAS Reports Services Publishing Reports to the Web*, the syntax for the security and destination elements incorrectly describes the securityUserid and portalUserid properties. They should be as follows:

```
<property name="securityUserid" value="portal_db_username/portal_
password@portal_db_connection" confidential="yes" encrypted="no"/>
```

```
<property name="portalUserid" value="portal_db_username/portal_
password@portal_db_connection" confidential="yes" encrypted="no"/>
```

5 Oracle Enterprise Manager

You can monitor and manage your Reports Servers through Oracle Enterprise Manager. The sections that follow describe the Reports Server pages available in Oracle Enterprise Manager.

You will find more information on the Reports Server in *Oracle9iAS Reports Services Publishing Reports to the Web*, included on the Oracle9iAS documentation CD.

5.1 Reports Server Main Page

This page summarizes the status of the selected Reports Server.

Table 5 Reports Server Main Page

Setting	Description
General Section	
Current Status	Indicates whether the server is up or down.
Start Time	Indicates the date and time the selected Reports Server was last started.

Table 5 Reports Server Main Page

Setting	Description
Stop Server/Start Server/Restart Server	These buttons enable you to stop, start, or restart the selected Reports Server from inside Oracle Enterprise Manager. Click Stop Server to stop the selected Reports Server; click Start Server to start the selected Reports Server. Click Restart to stop then start the selected Reports Server. The Start and Stop buttons display when the server is down; the Restart and Stop buttons display when the server is up.
Configuration Section	
Cluster Name	If the selected Reports Server is a member of a server cluster, the cluster name is listed here.
Trace Option	If you have entered an Oracle Trace option in your Reports Server configuration file, <server_name>.conf, this field lists the option(s) you have entered.
Trace Mode	Indicates the trace mode specified in the Reports Server configuration file for the trace log file, either replace (the default) or append. Trace replace replaces the existing text in the trace log file with new information. Trace append appends new information to the end of the existing trace log file.
Maximum Queue Size	Lists the value you have entered for the maximum queue size under the queue element in your Reports Server configuration file (<server_name>.conf). The queue element specifies the maximum number of jobs that can be held in the Reports queue, including the scheduled, current, and finished job queues. If the maximum is reached, the oldest job(s) are automatically purged to make room for the newest (first in/first out, or FIFO).
Status Section	
Active Engines	Indicates the number of engines currently running on the selected Reports Server.
CPU Usage (%)	Lists the percentage of the host machine's CPU currently employed by the selected Reports Server.
Memory Usage (MB)	Lists the number of megabytes (MB) of the host machine's RAM currently employed by the selected Reports Server.
Average Response Time (ms)	Lists the average number of milliseconds it takes for the selected Reports Server to process a request from the client.
Response and Load Section	

Table 5 Reports Server Main Page

Setting	Description
Current Jobs	Provides the total number of currently running jobs in the Job Queue. When this number is higher than 0, it links to the Current Jobs Queue, where you can view details and cancel currently running jobs.
Failed Jobs	Provides the total number of jobs currently in the selected Reports Server's Job Queue that were stopped before completion. This includes cancelled jobs as well as those terminated with error. When this number is higher than 0, it links to the Failed Jobs Queue, where you can get detail on why a job failed, view the job's trace file, and resubmit the job.
Finished Jobs	Provides the total number of jobs that have finished running successfully. When this number is higher than 0, it links to the Finished Jobs Queue, where you can get more detail on the finished job, view the job's trace file, view the job result from cache, and resubmit the job.
Scheduled Jobs	Provides the total number of jobs currently in the Scheduled Jobs Queue. When this number is greater than 0, it links to the Scheduled Jobs Queue, where you can view details and canceled the scheduled job.

Other Servers Running in the Cluster Section

Previous/Next Buttons	Click the Previous or Next button to page through the list of other cluster members, or select a range of clusters from the drop-down list.
Server Name	Lists the names of each of the other Reports Servers that are members of the same cluster that the selected Reports Server belongs to. Click the server's name to hyperlink to the OEM home page for that server.
Finished Jobs	Provides the total number of finished jobs currently in the listed Reports Server's Job Queue.
Current Jobs	Provides the total number of currently running jobs in the listed Reports Server's Job Queue.
Scheduled Jobs	Provides the total number of scheduled jobs currently in the listed Reports Server's Job Queue.
Failed Jobs	Provides the total number of jobs for the listed Reports Server that were stopped before completion. This includes cancelled jobs as well as those terminated with error.
Average Response Time	Lists the average number of milliseconds it takes for the listed Reports Server to process a request from the client.

Performance Section

Table 5 Reports Server Main Page

Setting	Description
Response Metrics	Provides details about average response time; scheduled, finished, current, and failed jobs in the Job Queue; and number of jobs transferred from one server to another in a clustered environment.
Engine Information	Lists the types and numbers of currently running engines on the selected Reports Server.
System Usage Metrics	Provides percentages of CPU and memory usage on the selected Reports Server.
Administration Section	
Configuration	Leads to the selected Reports Server's current configuration file. Here you can alter the file, check file syntax, and save your changes. Changes take effect after the next server restart.
Server Trace	Leads to the results of any trace you ran on jobs running on the selected Reports Server. Specify whether you will use the Trace option in the Reports Server's configuration file, available in OEM through the Configuration link.
Server Log	Leads to a log of general sever events, such as when the selected server was started and stopped.

5.2 Reports Server Performance Page

This page provides performance details about the selected Reports Server.

Table 6 Reports Server Performance Page

Setting	Description
System Usage Metrics Section	
CPU Usage (%)	Lists the percentage of the host machine's CPU currently employed by the selected Reports Server.
Memory Usage (MB)	Lists the number of megabytes (MB) of the host machine's RAM currently employed by the selected Reports Server.
Response Metrics Section	
Average Response Time (ms)	Lists the average number of milliseconds it takes for the selected Reports Server to process a request from the client.

Table 6 Reports Server Performance Page

Setting	Description
Number of Jobs Transferred	In a clustered server environment, provides the total number of jobs transferred between the selected Reports Server and other cluster members. For example, if the selected Reports Server receives a request for a job that was run earlier on another cluster member, the request is transferred to the cluster member that provided the earlier result and the result is delivered to the client from the cluster member's cache. Such a transaction would be counted as one transfer within the cluster.
Number of Failed Jobs	Provides the total number of jobs currently in the Job Queue that were stopped before completion. This includes cancelled jobs as well as those terminated with error.
Current Jobs	Provides the total number of currently running jobs in the selected Reports Server's Job Queue. When there are jobs currently running, the number in the Value column links to the Current Jobs queue.
Finished Jobs	Provides the total number of finished jobs currently in the selected Reports Server's Job Queue. When there are finished jobs in the queue, the number in the Value column links to the Finished Jobs queue.
Scheduled Jobs	Provides the total number of scheduled jobs currently in the selected Reports Server's Job Queue. When there are scheduled jobs in the queue, the number in the Value column links to the Scheduled Jobs queue.
Engine Information Sections	
Engine ID	Lists the type of engines available for processing jobs on the selected Reports Server.
Number of Running Engines	Provides the total number of this type of engine that is currently running on the selected Reports Server.

5.3 Reports Server Queue Page

This page provides a detailed look at all jobs currently running on the selected Reports Server and supplies the means of cancelling a currently running job. Click the **Previous** or **Next** button to page through the Current Job Queue, or select a range of records to view from the drop-down list.

To cancel a currently running job:

1. Click the **Select** radio button next to the job you want to cancel.
2. Click the **Cancel Job** button.

If you wish, you can resubmit a cancelled job from the Failed Job Queue.

Note: The Cancel Job button does not appear on this page when no jobs are currently running.

Table 7 Reports Server Queue Page

Setting	Description
Select	Use this radio button to select a particular job. On this page, this function is most useful when you wish to cancel a job. Click the Select radio button next to a job you wish to cancel, then click the Cancel button near the top of the page.
Id	Displays a unique job identifier assigned to this job by the Reports Server. This number is strictly under the server's control and cannot be reset by a user.
Job Name	If you specified a job name in the command line you used to run this job, that name is listed here. Otherwise, it is the name of the job provided for the "report=" or "module=" parameter of the job request.
Owner	Displays the user ID under which this job is running.
Output Type	Displays the destination type (destype) specified for this job at runtime.
Output Format	Displays the output format (desformat) specified for this job at runtime.
Queued At	Displays the date and time this job request was placed in the Job Queue.
Started At	Displays the date and time this job started running.
Interval	The frequency at which the job will be run, for example, daily, monthly, and so on. This setting only appears on the Reports Server Scheduled Job Queue page.

5.4 Reports Server Scheduled Job Queue Page

This page functions very much like Reports Server Queue page. Refer to the [Section 5.3, "Reports Server Queue Page"](#) for more information.

5.5 Reports Server Finished Job Queue Page

This page provides a detailed look at all successfully completed jobs in the Job Queue on the selected Reports Server.

Additionally, it provides a means of viewing a completed job's trace file, displaying job output from cache, or resubmitting a job request.

Click the **Previous** or **Next** button to page through the Finished Job Queue, or select a range of records to view from the drop-down list.

To view a job's trace file, the Trace option must have been specified in the Reports Server configuration file or the runtime command line.

To view a job's trace file:

1. In the Select column, click the radio button next to the finished job whose trace file you want to view.
2. Click the View Trace button near the top of the page.

To view a result from cache:

1. In the Select column, click the radio button next to the finished job you want to view.
2. Click the View Result button near the top of the page.

The result opens in a second browser window.

To resubmit a job:

1. In the Select column, click the radio button next to the job you want to resubmit.
2. Click the Rerun Report button near the top of the page.

Table 8 Reports Server Finished Job Queue Page

Setting	Description
Select	<p>Use this radio button to select a particular job. On this page, this function is most useful for selecting a report and:</p> <ul style="list-style-type: none">▪ Viewing a Web version of its output (click the Select radio button next to a job you want to view, then click the View Result button near the top of the page)▪ Viewing the selected job's trace results (click the Select radio button next to a job with trace results you want to view, and click the View Trace button near the top of the page)▪ Rerunning the job (click the Select radio button next to a job you want to rerun, and click the Rerun Report button near the top of the page)
Id	<p>Displays a unique job identifier assigned to this report by the Reports Server. This number is strictly under the server's control and cannot be reset by a user.</p> <p>When the job includes the generation of a trace file, the value under Id is linked to the trace file for this job. Click Id to view this job's associated trace file.</p>

Table 8 Reports Server Finished Job Queue Page

Setting	Description
Job Name	If you specified a job name in the command line you used to run this report, that name is listed here. Otherwise, it is the name of the job provided for the "report=" or "module=" parameter of the job request. Job Name is linked to the output of this job. Click Job Name to see a Web version of this job's output (fetched from the Reports Server cache).
Owner	Displays the user ID under which this job was run.
Output Type	Displays the destination type (destype) specified for this job at runtime.
Output Format	Displays the destination format (desformat) specified for this job at runtime.
Queued At	Displays the date and time this job request was placed in the Job Queue.
Started At	Displays the date and time this job started running.
Finished At	Displays the date and time this job completed.
Status	Displays the finished status of the job. In the Finished Job Queue, Status is always Finished Successfully.

5.6 Reports Server Configuration Page

This page provides a window into the selected Reports Server's configuration file (<server_name>.conf). You can edit the configuration file here as well as check its syntax and save your changes. You must restart the server for your changes to take effect.

To edit the selected Reports Server configuration file, make your changes in the display window.

To check your syntax, click the **Check Syntax** button below the display window. Note that clicking this button does not validate the values you enter for configuration elements. For example, if an element requires that you specify a directory path, syntax checking does not validate the accuracy of your path. It just validates the XML syntax.

To save your changes, click the **Save Changes** button below the display window.

Note: You can use your browser's "Find in Page" functionality to search the content of the server configuration file. This is particularly useful if you must locate a syntax error in the file.

5.7 Reports Server Failed Job Queue Page

This page provides a detailed look at all failed jobs in the Job Queue on the selected Reports Server. Additionally, it provides a means of viewing a failed job's trace file or resubmitting a job request.

Note: Failed jobs are jobs that were cancelled by the user or that automatically terminated with error.

Click the **Previous** or **Next** button to page through the Failed Job Queue, or select a range of records to view from the drop-down list. To view a job's trace file, the Trace option must have been specified in the Reports Server configuration file or the runtime command line.

To view a job's trace file:

1. In the **Select** column, click the radio button next to the failed job whose trace file you want to view.
2. Click the **View Trace** button near the top of the page.

To resubmit a job:

1. In the **Select** column, click the radio button next to the job you want to resubmit.
2. Click the **Rerun Report** button near the top of the page.

Table 9 Reports Server Failed Job Queue

Setting	Description
Select	Use this radio button to select a particular job. On this page, this function is most useful for selecting a job and: <ul style="list-style-type: none">▪ Viewing the selected job's trace results (click the Select radio button next to a job with trace results you want to view, and click the View Trace button near the top of the page)▪ Rerunning the job (click the Select radio button next to a job you want to rerun, and click the Rerun Report button near the top of the page)
Id	Displays a unique job identifier assigned to this job by the Reports Server. This number is strictly under the server's control and cannot be reset by a user. When the job includes the generation of a trace file, the value under Id is linked to the trace file for this job. Click Id to view this report's associated trace file.
Job Name	If you specified a job name in the command line you used to run this report, that name is listed here. Otherwise, it is the name of the job provided for the "report=" or "module=" parameter of the report request.
Owner	Displays the user ID under which this job was run.

Table 9 Reports Server Failed Job Queue

Setting	Description
Output Type	Displays the destination type (destype) specified for this job at runtime.
Output Format	Displays the destination format (desformat) specified for this report at runtime.
Queued At	Displays the date and time this job request was placed in the Job Queue.
Started At	Displays the date and time this job started running.
Finished At	Displays the date and time this job was cancelled or terminated with error.
Status	Displays the status of the job. Status will either indicate that the job was cancelled by the user or provide some information on why the job was terminated with error.

5.8 Job Trace Page

This page provides a view of the trace results of a particular job, rather than all the jobs run on a particular server. This information can be useful in the event you must call for technical support.

Table 10 Job Trace Page

Setting	Description
Job Id n	Identifies the job by the number the Reports Server assigned to it at runtime.
Previous/Next/Drop-down list	Use these buttons and the list of values to navigate through the list of trace events.
Originating Time	The date and time the event occurred.
Module Name	The section in the underlying code where the event occurred.
Error Number	The error code assigned to this event. To look up the meaning associated with this number, see the Reports Builder online help.
Type	The type of event that occurred.
Message	If the developer who wrote the code included a message with this event type, it will appear here.

5.9 Reports Server Trace Page

This page provides a view of the trace results for the selected Reports Server. Depending on whether jobs results are appended to the existing

trace file or replace its content, this file shows the trace results for all jobs run on this server (append) or the last job run on the server (replace).

If this page is empty, it means you did not specify any trace options in the server configuration file (<server_name>.conf) or in the runtime command line.

Trace information is useful in the event you must call for technical support.

5.10 Reports Server Log Page

This page provides a view of the Reports Server log file. If the server configuration file contains a <log option="x"/> element, then a log file is created and maintained for the selected Reports Server.

If this page is empty, it means you did not specify any log options in the server configuration file (<server_name>.conf).

Logging is maintained for backward compatibility. Current practice is to capture processing information in a trace file.

6 Express Pluggable Data Source

This section describes how to configure and use the Express Pluggable Data Source with Oracle9i Reports Developer.

6.1 Before You Begin

To use the Express Pluggable Data Source within Oracle9i Reports Developer, you must connect to a supported Express Server version, which are versions 6.2.x and 6.3.x. When you install Reports9i, SNAPI 9.0.1 is automatically installed to enable connections to these Express Server versions.

Before creating reports using Express data, verify that you have completed all the necessary configuration steps. See the configuration topics in the Help system by searching for "configuring the Express data source."

In the Help topic called "About Configuring the Express Data Source," it mentions that the Express Connection Editor is installed with Reports. This is no longer the case, and you must take the appropriate steps to ensure that you can connect to Express Server, either by using the Express Connection Editor to create connection files or by creating them manually. See the section "Preparing for Express Connections" later in this document.

6.2 Preparing for Express Connections

6.2.1 Creating connection files

To create reports of Express data, you must be able to connect to an appropriate Express Server instance. To connect to an instance of Express Server, you use a connection file, which is a simple text file with the XCF extension. Each connection file defines a single connection to Express Server. A file called `xconnect.ini` specifies where connection files are located.

You can create connection files manually or you can use a utility called the Express Connection Editor to assist you in creating these files. The following list outlines the ways to create and use connection files, which are explained in detail below.

- Use an already installed version of the Express Connection Editor.
- Download a file from the Oracle Technology Network with which you can install just the Express Connection Editor.
- Install just the Express Connection Editor from the Express Client CD.
- Create the files manually using a text editor.

6.2.2 Related information

This document includes samples of three types of connection files in a later section.

For complete information on connecting to Express Server and on the Express Connection Editor, consult the following sources:

- The *Oracle Express Database Administration Guide* (part number A82800_02).
- The Help system for the Express Connection Editor.

6.2.3 Using an already installed version of the Express Connection Editor

If you already have the Express Connection Editor installed, then you can simply use that version to create an XCF with which you can connect to Express Server. You might not have to create any XCF files, if the appropriate ones have already been created. Use the following procedure to use the already installed Express Connection Editor with Oracle9i Reports.

To use an existing Express Connection Editor installation:

1. Navigate to the `/olap` subdirectory in the Oracle home directory for Oracle9i Reports Developer.

2. Within the `/olap` subdirectory, create a subdirectory called `ecf901`.
3. To the newly created `ecf901` subdirectory, copy the `xconnect.ini` file from the installation directory of the Express Connection Editor.
4. Open the `xconnect.ini` from the `ecf901` subdirectory in a text editor. Ensure that the `ConnectionPath` setting points to whatever directory or directories hold the connection files that you plan to use for connecting to Express Server from Oracle9i Reports Developer. Use these files or use the Express Connection Editor to create other connection files.

Tip: Use semicolons to separate multiple directory specifications for `ConnectionPath`, if you want to store connection files in multiple locations.

6.2.4 Downloading a file from OTN with which you can install just the Express Connection Editor

If you do not already have the Express Connection Editor installed, then you can download the necessary files. Use the following procedure to download the file and install the Express Connection Editor.

To download a file and install the Express Connection Editor:

1. In a Web browser, access the Oracle Technology Network (<http://otn.oracle.com>).
2. Navigate to the Oracle9i Reports Developer area.
3. Download the file with which you can install the Express Connection Editor.
4. Unzip the downloaded file and use the `setup.exe` file to run the installation program to install just the Express Connection Editor.

Important: Ensure that you install the Express Connection Editor in the same Oracle home directory into which Oracle9i Reports Developer is installed.

Once the Express Connection Editor is installed, you can use it to create connection files.

6.2.5 Install just the Express Connection Editor from the Express Client CD

If you do not already have the Express Connection Editor installed, then you can use the Express Connection Editor that is supplied with the Express Client products. You can obtain a CD for the Express Client products or download a file that contains them. Use the following procedure to install just the Express Connection Editor.

To install just the Express Connection Editor:

1. Run the installation program by using the Express Client CD or by accessing the Express Server area of the Oracle Technology Network (http://otn.oracle.com/software/products/exp_server/htdocs/winsoft.html) and following the directions there.
2. In the appropriate location in the installation program, specify the same Oracle home directory into which Oracle9i Reports is installed.
3. Select a custom installation.
4. Select the Oracle Express Connection Editor in the Available Product Components page.
5. Complete the other pages of the installation as appropriate.

Once the Express Connection Editor is installed, you can use it to create connection files.

6.2.6 Create the files manually using a text editor

If you do not already have the Express Connection Editor installed and you do not want to install it, then you can create XCF files manually using a text editor. You must also create the `xconnect.ini` file, which specifies the location of the XCF files.

To create an XCF file manually:

1. Navigate to the `/olap` subdirectory in the Oracle home directory for Oracle9i Reports Developer.
2. Within the `/olap` subdirectory, create a subdirectory called `ecf901`.
3. In this document, locate the sample file that is closest to the XCF file that you need to create.
4. Open a text editor such as Microsoft Notepad.
5. Either cut and paste the text from this document or type the text of the XCF file into the text editor.
6. Edit the settings of the file as appropriate, using the information that is provided in the section "XCF file settings," later in this document.
7. Save your changes, giving the file any name that you want and including the XCF extension. Ensure that you save the file into the `/olap/ecf901` subdirectory in the Oracle home directory for Oracle9i Reports Developer.
8. Begin the process of creating the `xconnect.ini` file by creating an empty document in a text editor such as Microsoft Notepad.

9. Edit the file to contain the following contents, substituting the name of the Oracle home directory for Oracle9i Reports Developer for ORACLE_HOME:

[General]

```
ConnectionPath=d:\ORACLE_HOME\olap\ecf901;
```

Tip: Use semicolons to separate multiple directory specifications for ConnectionPath, if you want to store connection files in multiple locations.

10. Save your changes, naming the file `xconnect.ini`. Ensure that you save the file into the `/olap/ecf901` subdirectory in the Oracle home directory for Oracle9i Reports Developer.

6.2.7 XCF file settings

The following table briefly describes the main settings in any XCF file. The next table describes the settings that apply only to connections through Oracle Express Relational Access Manager. All these settings are described in more detail in the Help system for the Express Connection Editor and in the *Oracle Express Database Administration Guide*.

Table 11 XCF File Settings

Setting	Description
ConnectionType	Enter the type of connection: <ul style="list-style-type: none"> ▪ 1 for connections for Oracle Express Relational Access Manager. ▪ 0 for all other connections.
ServerDescription	Enter the description for this connection. The description is visible to users when they choose which connection to make to Express Server. Therefore, you should make the description no more than approximately 40 characters.
ServerVersion	Enter 1 for Express Server 6.x. No other values are applicable to connections from Oracle9i Reports Developer.
ServerType	Enter 1 for Express Server 6.x. No other values are applicable to connections from Oracle9i Reports Developer.
ServerLogin	See the list that follows this table for a description of this setting.
ServerString	Enter the name of the server machine on which Express Server 6.x is running. Note: The Express Connection Editor includes more than just the server machine name in the connection file, which is unnecessary for connections from Oracle9i Reports Developer.

The following list describes the values for the ServerLogin setting. For this setting, you enter the authentication type for this connection:

- 0 for None, no authentication is required.
- -1 for Host (Server Login), which specifies host authentication with a machine login. Use host authentication when the client machine is not on the Microsoft network. Use this setting when the server machine is not in the domain (and/or if there is no domain account) and the user has a server machine account. Only the user ID and password are required for logging into the server machine.
- -2 for Host (Domain Login), which specifies host authentication with a domain login. Use host authentication when the client machine is not on the Microsoft network. Use this setting when the server machine is within the domain and the user has a domain account. The domain name, user ID, and password are required for logging into the server machine.
- The following settings apply to domain authentication with a domain login. Domain authentication is the preferred way of authenticating clients for Windows NT Server and for Express Server 6.x. If you want to use domain authentication, then the client and the server must be on the Microsoft network. Also, the user has to have a domain account within the same domain that the server machine is on. The domain name, user ID, and password are required for logging into the server machine.
 - 1 for domain authentication with connection, which specifies that a user's identification is checked when connecting to Express Server. A user must have logged into the domain with a valid user name and password.
 - 2 for domain authentication with call, which specifies that a user's identification is checked on every call to Express Server.
 - 3 for domain authentication with packet, which specifies that a user's identification is checked on every packet sent to Express Server.
 - 4 for domain authentication with integrity, which specifies that a calculated cryptographic checksum is attached to every packet so that any tampering with the packet can be detected. A user's identification and the checksum is checked on every packet sent to Express Server. If the checksum received does not match the checksum sent, then the packet is discarded.
 - 5 for domain authentication with privacy, which specifies that the packet is encrypted. This option also uses the security measures that are available with the integrity option.

The following table briefly describes the Oracle Express Relational Access Manager settings in a connection file, which appear after the main settings that are described in the previous table.

Table 12 Oracle Express Relational Access Manager Settings

Setting	Description
ConnectionType	Enter 0. No other values are applicable to connections from Oracle9i Reports.
MasterDB	<p>Enter the name of the Relational Access Manager database to attach initially. You must specify only the database file name. You can get the database name in either of the following ways:</p> <ul style="list-style-type: none"> ▪ In the Express Relational Access Administrator, it is the Express Database Name that is displayed in the Database tab. ▪ In the .RDC file, it is specified as the DBName in the [DBInfo] section. <p>This database must reside in a folder that is included in the path list in ServerDBPath. You can check the ServerDBPath in the I/O Management sheet of the Express Instance Manager.</p>
PromptforExpressID	Enter 1 to prompt for an Express user ID before making the connection or 0 to not prompt. This setting applies only when PersonalConfig is set to 1.
ServerScript	Enter the complete file name (including the full path) of the remote database configuration file on the server. This file specifies information such as the location of code and data databases. Using UNC (Universal Naming Convention) syntax allows multiple users to use the same connection to access the data without having to map the same drive letter to that location. UNC syntax is \\ServerName\ShareName\ followed by any subfolders and/or files.
PersonalConfig	Enter 1 to create and attach a personal database with read/write access so that you can use many Oracle Sales Analyzer features such as custom measures. Enter 0 to not create this database. To enable this setting, you must have an account on the Express Server system. If this setting is 0, then you log in as a guest. This setting applies only with direct connections to Express Server 6.x. This setting does not apply when ServerLogin is set to 0.

6.2.8 Sample connection files

This section provides samples of three types of connection files. You can use these samples as the basis for creating connection files manually.

Sample for connecting without using authentication

The following text shows a sample XCF file for connecting to Express Server without using authentication.

```
[Express]
ConnectionType=0
ServerDescription=expservername ExpSrv6.3.0.2 without authentication
ServerVersion=1
ServerType=1
ServerLogin=0
ServerString=expservername
```

Sample for connecting with using authentication

The following text shows a sample XCF file for connecting to Express Server using authentication.

```
[Express]
ConnectionType=0
ServerDescription= expservername ExpSrv6.3.0.2 with authentication
ServerVersion=1
ServerType=1
ServerLogin=-1
ServerString=expservername
```

Sample for connecting with Oracle Express Relational Access Manager

The following text shows a sample XCF file for connecting through Oracle Express Relational Access Manager (RAM).

```
[Express]
ConnectionType=1
ServerDescription=expservername ExpSrv6.3.0 with RAM
ServerVersion=1
ServerType=1
ServerLogin=-1
ServerString= expservername
[Relational Access Manager]
ConnectionType=0
MasterDB=d:\RAM\dram.db
PromptForExpressID=0
ServerScript=d:\RAM\ram.rdc
PersonalConfig=0
```

6.3 Known Issues

- The Express Pluggable Data Source does not support using password-protected Express databases files.
- The maximum number of dimensions for an Express query is 10.

- You might see the following message: **"XR-2005: Express Server cannot execute a command that is specified by the query. The command is invalid."** This message can be generated for very large queries (row counts in the millions) if the SNAPI buffer is not sufficiently large. To increase the buffer size from the default of 2MB, create or edit the `xrpdsprefs.ora` file in the `<oraclehome>/reports/jlib` directory and add the following line:

```
options.snapi.FetchAlloc=<size>
```

where `<size>` is a value greater than 2M (or 2000K, or 2,000,000), as shown in the following example:

```
options.snapi.FetchAlloc=10M
```

See the "Specifying a buffer size for Express data" topic in the online help system for complete information on this option.

- When Oracle8 and Express Server are installed on the same Solaris machine and the Express connection specifies an Authentication Type of "None", you may see the following error when running a report that contains Express data:

```
ORA-20004: Failed to attach to an Express database.
```

This error indicates that there is an access problem with the Express database. When the Authentication type is set to "None" and Oracle8 and Express Server are on the same machine, the SNAPI connection uses the user of the process that is establishing the connection. In this configuration, it is the user that started the Oracle8 database. Refer to the *Oracle Express Server Installation and Configuration Guide for Solaris* for information on accessing Express databases.

- If you suspect language incompatibilities between Oracle9i Reports Developer and the data from Express Server, then whenever possible, ensure that the following settings all specify the same language/character set:
 - The operating system's regional or locale setting for the machine on which you are running Oracle9i Reports Developer.
 - The actual data in the Express database.

If your environment does not permit these settings to match, then you should add the `options.data.XPCharSet` line to the `xrpdsprefs.ora` file and specify the language/character set for the data in the Express database. The format of this setting matches that of the `NLS_LANG` setting and a sample setting is shown here:

```
options.data.XPCharSet=JAPANESE_JAPAN.JA16SJIS
```

See the Help system for more information on the `xrpdsprefs.ora` file.

