Oracle9*i* Reports

Release Notes (Patch 4)

Release 4 (9.0.2)

Part Number B13551-01

February 2004

Oracle9*i* Reports Release Notes (Patch 4), Release 4 (9.0.2)

Part Number B13551-01

Primary Author: Parineetha Henry

Contributing Author: Frank Rovitto

Contributors: Robin Fisher, Shaun Lin, Paul Narth, Rohit Marwaha, Rajesh Ramachandran, Danny Richardson, Jeff Tang, Philipp Weckerle, Anil Sharma, Vinayak Hegde, Sripathy P Rao, Stewart Wilson, Ken Young, Sivakumar Balagopalan, Rajendra K Gokidi, Diego Iglesias, Sudha Thazhathaveetil, Srivas.N. Chennu, Puvanenthiran Subbaraj

Contents

Send Us Your Comments vii

Preface ix

Intended Audience ix
Documentation Accessibility ix
Structure x
Related Documents x
Conventions x

1 Patch Release 4

1.1 Enhancements in Patch Release 4 1-1
1.1.1 Resolving font alignment issues in R-T-L PDF font subsetting 1-1
1.2 Enhancements in Previous Patch Releases 1-2
1.2.1 Setting the Format Order of a Report 1-2
1.2.2 Turning on Component Tracing 1-4
1.2.3 Improving Image Resolution of Graphs Output to a PDF file or a Printer 1-4
1.2.4 Setting the JVM options in Oracle9*i* Reports 1-4
1.2.5 Specifying a BiDi-related Algorithm 1-5
1.2.6 Printing the Delimited Character at the End of the Report 1-5
1.2.7 Printing Collated / Non-Collated pages 1-6
1.2.8 Dynamic Environment Switching 1-6
1.2.9 Running a Report to Reports Server Asynchronously 1-11
1.3 Information About Important Bug Fixes and Workarounds in Patch Release 4 1-12
1.3.1 Running a multibyte report with the WE8ISO8859P15 character set 1-12
1.3.2 Standardizing the error message template character set 1-12
1.3.3 Specifying the character encoding for reports output using rwservlet 1-13
1.3.4 Specifying jobwise database connection information 1-13

2 General Issues and Workarounds

2.1 Migration 2-1
2.2 Deprecated Features from Earlier Versions 2-1
2.3 reports.sh (UNIX) 2-2
2.4 Using the Hotspot JVM Instead of Classic 2-2
2.5 Spaces in HTML Output 2-2
2.6 Printing Reports Without Hardware-based Left Margins on Windows 2-3

2.7 Inserting Multiple Report Blocks In Web Source 2-3
2.8 PDF Font Sub Setting 2-4
2.9 Supported Data Types in Pluggable Data Sources 2-4
2.10 Links Between Queries Using Pluggable Data Sources 2-4
2.11 Oracle9*i*AS Portal Portlet/Page Parameters 2-4
2.11.1 Customize Page 2-5
2.12 Report Portlet and Netscape Communicator 4.7 on UNIX 2-5
2.13 Setting the Size of the Report Portlet on Windows 2-5
2.14 Destination Parameter and Microsoft Internet Explorer 2-5
2.15 Order of Parameters in a URL 2-5
2.16 Accessing Web Commands With DIAGNOSTIC set to NO 2-6
2.17 Templates 2-6
2.18 Oracle9*i*AS Portal Security, Portal Destination, and Job Status Repository 2-6
2.19 Import Oracle9*i*AS Portal Component 2-6
2.20 JDeveloper Integration 2-7
2.20.1 Deadlock when debugging JSP reports in Oracle JDeveloper 2-7
2.20.2 Graph not appearing in Oracle JDeveloper 2-7
2.20.3 Reports In-Process Server 2-7
2.21 Oracle9iAS Forms Services Integration 2-7
2.22 URL Encoding 2-8
2.23 Command Line Arguments 2-8
2.23.1 ENVID 2-9
2.23.2 engineResponseTimeOut 2-9
2.23.3 SUPPRESSLAYOUT Keyword 2-9
2.23.4 UPGRADE_PLSQL 2-9
2.23.5 RECURSIVE_LOAD 2-10
2.23.6 SQLTRACE 2-10
2.24 Built-ins 2-11
2.24.1 SRW.GET_VALUE 2-11
2.24.2 SRW.GET_REPORT_NAME 2-11
2.25 Default Character Set for JSPs 2-11
2.26 REPORTS_NLS_XML_CHARSETS 2-11
2.27 Java Importer 2-12
2.28 Buttons 2-12
2.29 Parameter Forms 2-12
2.30 RTF Output in Microsoft Word 95 for Japanese 2-12
2.31 Bold Korean Font Not Appearing Correctly 2-12

3 Configuration Issues and Workarounds

3.1 Reports Configuration Assistant Fails during Installation 3-1
3.2 Reports CGI Not Starting on Windows 3-2
3.3 In-process Server Not Starting on Linux 3-2
3.4 Reports Server Installed with Reports Builder 3-2
3.5 Running JSP Reports from Oracle9*i*AS Portal 3-2
3.6 REPORTS_CLASSPATH 3-3
3.7 X-terminals and Graphical Terminals 3-3
3.8 REPORTS60_DEFAULT_PIXEL_SIZE 3-3

3.9 Oracle Reports 6i Access to Oracle9*i*AS Reports Services 3-3

4 UI Issues and Workarounds

4.1 Source Control (Windows only) 4-1
4.1.1 Check In/Check Out Restriction 4-1
4.2 Some Languages Not Appearing Correctly in Web Source View 4-1
4.3 Find/Replace in Web Source View for Asian Languages Causes Crash........ 4-1
4.4 Documentation Errata 4-2

5 Oracle Enterprise Manager

5.1 Oracle Enterprise Manager Configuration........ 5-1
5.2 Reports Server Main Page........ 5-2
5.3 Reports Server Performance Page 5-4
5.4 Reports Server Queue Page 5-5
5.5 Reports Server Scheduled Job Queue Page........ 5-5
5.6 Reports Server Finished Job Queue Page 5-6
5.7 Reports Server Configuration Page........ 5-7
5.8 Reports Server Failed Job Queue Page 5-7
5.9 Job Trace Page 5-8
5.10 Reports Server Trace Page 5-9
5.11 Reports Server Log Page........ 5-9

6 Oracle Express Pluggable Data Source

6.1 Before You Begin........ 6-1
6.2 Preparing for Express Connections........ 6-1
6.2.1 Creating connection files 6-1
6.2.2 Related information........ 6-2
6.2.3 Using an already installed version of the Express Connection Editor 6-2
6.2.4 Downloading a file from OTN to install just the Express Connection Editor 6-2
6.2.5 Install just the Express Connection Editor from the Express Client CD 6-3
6.2.6 Create the files manually using a text editor 6-3
6.2.7 XCF file settings 6-4
6.2.8 Sample connection files........ 6-6
6.3 Known Issues........ 6-7

A Fixes and Workarounds for 9.0.2.4

Send Us Your Comments

Oracle9*i* Reports Release Notes (Patch 4), Release 4 (9.0.2)

Part Number B13551-01

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

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

If you find any errors or have suggestions for improvement to this documentation, please send us your comments through the Oracle9*i* Reports discussion group forum: `http://otn.oracle.com/products/reports/index.html`

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

Preface

This Preface contains the following topics:

- Intended Audience
- Documentation Accessibility
- Structure
- Related Documents
- Conventions

Intended Audience

The Oracle9*i* Reports Release Notes (Patch 4) intended for anyone interested in Oracle9*i* Reports Developer and Oracle9*i*AS Reports Services.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at:

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Structure

This document contains the following chapters:

Chapter 1, "Patch Release 4"

This chapter describes the changes for this patch release.

Chapter 2, "General Issues and Workarounds"

his chapter describes general issues and their work arounds for Oracle9*i*AS Reports ServicesOracle9*i*AS Reports Services.

Chapter 3, "Configuration Issues and Workarounds"

This chapter describes configuration issues and their work arounds for Oracle9*i*AS Reports Services.

Chapter 4, "UI Issues and Workarounds"

This chapter describes user interface issues and their work arounds for Oracle9*i*AS Reports Services.

Chapter 5, "Oracle Enterprise Manager"

This chapter describes how to configure Oracle Enterprise Manager for your Reports Server as well as the Reports Server pages available in Oracle Enterprise Manager.

Chapter 6, "Oracle Express Pluggable Data Source"

This chapter describes how to configure and use the Express Pluggable Data Source with Oracle9*i*AS Reports Services.

Appendix A, "Fixes and Workarounds for 9.0.2.4"

This appendix lists all the bugs that have been fixed for this release, that is version 9.0.2.4.

Related Documents

For more information, see the following manuals in the Oracle9*i* Reports documentation set:

- *Oracle9iAS Reports Services Publishing Reports to the Web, Part No. A92102-01*
- *Oracle9i Reports*
- *Oracle9i Reports Building Reports, Part No. B10310-01*
- *Getting Started with Oracle9i Reports*, available on the Oracle Technology Network:

  `http://otn.oracle.com/products/reports/`

Conventions

The following conventions are also used in this manual:

Convention	Meaning
. . .	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.

Convention	Meaning
. . .	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted
boldface text	Boldface type in text indicates a term defined in the text, the glossary, or in both locations.
variable	Monospace italic type indicates variables or user-supplied names.
[ ]	Brackets enclose optional clauses from which you can choose one or none.

1

Patch Release 4

This chapter describes the changes that occurred in Patch Release 4. As Patch Release 4 is a cumulative patch, installing it will enable newly incorporated features and fixes, including those that pertain to Patch Release 3.

This chapter contains the following sections:

- Enhancements in Patch Release 4
- Enhancements in Previous Patch Releases
- Information About Important Bug Fixes and Workarounds in Patch Release 4

1.1 Enhancements in Patch Release 4

This section outlines the newly incorporated enhancements pertaining to Patch Release 4.

1.1.1 Resolving font alignment issues in R-T-L PDF font subsetting

Bug Nos: 2968284, 2659642, 3017227

Applicable to All Platforms

Description

When you create R-T-L PDF reports with font subsetting enabled, you may see some font misalignment if the row contains a large number of characters. This issue occurs because of the variance between the actual value and the width calculated by Oracle Toolkit in the case of variable width fonts.

Fix

To fix this issue, you must download and install the patch located at: `http://metalink.oracle.com`.

The ARU patch number for Windows is 5143372.

The ARU patch number for HP-UX 11 is 4519711.

The ARU patch number for HP-UX 11.0/32 BIT is 4519711.

The ARU patch number for Solaris is 3039666.

The ARU patch number for Sun Solaris (SPARC) is 4650528.

Known Limitation

On UNIX platforms, you may see some font alignment issues even after applying the patch. This issue will be fixed in a later release.

To work around this issue, use fixed width fonts instead of variable width fonts. For example, Miriam Fixed True Type font (Hebrew), a fixed width font available on Windows 2000 can be used for font subsetting on UNIX platforms. This should fix any font alignment issues with Hebrew fonts.

1.2 Enhancements in Previous Patch Releases

This section outlines the newly incorporated enhancements pertaining to Patch Release 1, Patch Release 2, and Patch Release 3.

1.2.1 Setting the Format Order of a Report

Bug No: 2775230

In prior releases, Oracle9*i* Reports formatted the sections of a report in sequential order: Header section, followed by Main section, followed by Trailer section. This release introduces the capability to change the order in which the three sections of a report are formatted.

The format order can be set using the `SRW.SET_FORMAT_ORDER` built-in procedure.

This feature is useful for formatting any report section first to retrieve information that is known only at the time of formatting, such as page numbers, then using that information in the formatting of a previous section.

For example, to create a table of contents (TOC) for a report, you can format the Main section first and use report triggers to build a table containing the TOC entries. When the first element for the TOC is formatted, a trigger fires and creates a row in the TOC table containing the TOC entry and the page number. After the Main section has completed formatting, the format order setting can define that the Header section is formatted next. The Header section can contain a report block based on the TOC table. After formatting, you can output your report with a TOC (the Header section), followed by the report body (the Main section), followed by the Trailer section.

1.2.1.1 SRW.SET_FORMAT_ORDER

Description Use the `SRW.SET_FORMAT_ORDER` built-in procedure to specify the order in which the three sections of a report (Header, Main, and Trailer) are formatted.

Syntax

```
SRW.SET_FORMAT_ORDER (first_section, second_section, third_section);
```

where:

```
first_section,second_section,third_section
```

If `SRW.HEADER_SECTION,SRW.MAIN_SECTION,SRW.TRAILER_SECTION` specified in the order in which the report should be formatted

Usage Notes

- `SRW.SET_FORMAT_ORDER` can be called in either the Before Parameter Form trigger or the After Parameter Form trigger. If called in both triggers, the call from the After Parameter Form trigger takes precedence.

Note: Regardless of the order in which the report sections are formatted, the output order (that is, visual order) is unchanged: Header section, followed by Main section, followed by Trailer section.

Restrictions

- In certain circumstances involving the Paper Design view, Reports Builder does not execute the Before Parameter Form or After Parameter Form triggers; therefore, `SRW.SET_FORMAT_ORDER` is not executed. This occurs in the following cases:
  - If you switch view from the Paper Design view to, say, the Paper Layout view and then back to the Paper Design view.
  - If in the Paper Design view, you choose Print Preview, then close the Print Preview window.
  - If you perform a live refresh on the Paper Design view.

  To overcome this limitation, you can close the Paper Design view, then redisplay it.

  Note: This limitation applies only in Reports Builder; it does not occur in Reports Runtime and Reports Builder.

- If the `SRW.SET_FORMAT_ORDER` built-in procedure is called from any trigger other than the Before Parameter Form trigger or the After Parameter Form trigger Reports Builder issues an error message:

  ```
  REP-1426: <trigger name>: Running SRW.SET_FORMAT_ORDER from incorrect context.
  ```

Examples

Example 1:

```
function AfterPForm return boolean is
  begin
    SRW.SET_FORMAT_ORDER(SRW.MAIN_SECTION, SRW.HEADER_SECTION, SRW.TRAILER_
SECTION);
    return (TRUE);
  end;
```

Example 2:

This example produces the same result as the earlier example.

```
function AfterPForm return boolean is
  begin
    SRW.SET_FORMAT_ORDER(1, 0, 2);
    return (TRUE);
  end;
```

Note: The possible range of values that `SRW.SET_FORMAT_ORDER` takes are: 0,1 and 2. If you specify values outside this range, Reports Builder issues an error message:

```
REP-1417: <trigger name>: Invalid integer argument passed to
SRW.SWT_FORMAT_ORDER.
```

1.2.2 Turning on Component Tracing

Bug No: 2833652

Prior to Patch Release 3, turning on the trace option in the server configuration file turned tracing on for both the server and the engine. Performance suffers when you set the tracing option for the engine.

To specify the tracing option only for the server, you must set the new trace attribute, `traceModule`. The values for `traceModule` are: all,server,and engine.

For example:

```
<trace traceOpts="trace_all" traceModule="server"/>
```

If you do not specify a value for `traceModule`, both `server` and `engine` tracing are turned on.

1.2.3 Improving Image Resolution of Graphs Output to a PDF file or a Printer

Bug No: 2192297

Description The `REPORTS_GRAPH_IMAGE_DPI` environment variable specifies a dots per inch (DPI) value for graphs being output to a PDF file or a printer. This environment variable enables you to increase the earlier fixed default value of 72 DPI, in turn improving the image resolution of graphs output to a PDF file or a printer.

Valid Values 72 through 300

Default 250

> **Note:** With the default value of 250:
>
> - The time taken to generate a report with an `YES` graph increases 5 to 6 times when compared to the time taken to generate the same report with the value set to 72dpi.
> - The PDF file size also increases 5 to 6 times.

Usage Notes

- (Windows only) Use the registry to specify the value. (UNIX only) Set the value as an environment variable (shell script).
- When you set a DPI value greater than 250 and your chart is bigger than 5"x5" (approximately), you may also need to change the JVM heap size value using `REPORTS_JVM_OPTIONS` to avoid the Out Of Memory error for the JVM.

  See Also: Setting the JVM options in Oracle9i Reports. For more information on setting the JVM options.

- To revert to the old behavior, set the value of this environment variable to 72 DPI.
- This variable is currently not supported in YES distribution functionality as this is specific to PDF and printer outputs only.

1.2.4 Setting the JVM options in Oracle9*i* Reports

Bug number: 2192297

Description The `REPORTS_JVM_OPTIONS` environment variable specifies any JVM options that you want Reports Builder, Reports Runtime, or Oracle9*i*AS Reports Services to consider when it starts its JVM. For example, you can use this environment variable to specify the starting heap size and maximum heap size for the JVM, additional classpath entries, and so on.

Valid Values List of JVM options in the JVM command line syntax.

Default `-Xmx256M`

Usage Notes

- (Windows only) Use the registry to specify the value. (UNIX only) Set the value as an environment variable (shell script).
- The default value `-Xmx256M` specifies the JVM heap size of 256 to avoid the `Out Of Memory` error when running reports with large graphs or running big reports.
- In the server mode, you can also specify the jvmoptions in the *`servername`*`.conf` file in the `engine` attribute. If specified, the `jvmoptions` set in the server configuration file would take precedence over the `REPORTS_JVM_OPTIONS` environment variable. In this case any value set in the `REPORTS_JVM_OPTIONS` will be discarded.

1.2.5 Specifying a BiDi-related Algorithm

Bug No: 2926463

Description The `REPORTS_BIDI_ALGORITHM` environment variable specifies the bidirectional (BiDi) algorithm to be used to resolve NLS-related issues of BiDi languages (for example, Arabic or Hebrew) such as the period (.) for `NLS_CURRENCY`.

Set the value of this environment variable based on where you will invoke your report. For example, Microsoft's Internet Explorer, Netscape, and Java use the Unicode algorithm (`BIDI_ALGORITHM_UNICODE`) for Bidi handling. However, Oracle9iAS Forms Services uses the Oracle algorithm (`BIDI_ALGORITHM_PASTA200`)

Valid Values `ORACLE|UNICODE`

where

`ORACLE = BIDI_ALGORITHM_PASTA200`

`UNICODE = BIDI_ALGORITHM_UNICODE`

Default `ORACLE`

1.2.6 Printing the Delimited Character at the End of the Report

Description The `DELIMITED_LINE_END` environment variable specifies whether to print the delimited character at the end of the line for delimited output.

Valid Values `YES|NO`

Default `YES`

Usage Note

Set this environment variable to NO to ensure that the delimited character is not printed at the end of the line.

1.2.7 Printing Collated / Non-Collated pages

Table 1–1 indicates which commands can use the COLLATE keyword.

Table 1–1 Commands that can use the COLLATE keyword

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

Description: Use COLLATE to control the collating behavior when a report is sent to a printer. This behavior is applicable to both Postcript and PCL on UNIX platforms.

For example, printing three copies of a three page document with COLLATE set to YES would result in an output similar to the following:

```
1 2 3|1 2 3| 1 2 3
```

The order specified is the page numbers being printed. This behavior is similar to selecting the **Collate** check box in the Print dialog box.

Printing three copies of a three page document with COLLATE set to NO would result in an output similar to the following:

```
1 1 1| 2 2 2| 3 3 3
```

Syntax COLLATE={YES|NO}

Values

- YES Collates the pages when sent to a printer.
- NO Does not collate the pages when sent to a printer.

Default YES

1.2.8 Dynamic Environment Switching

Bug No: 2965782

In the past, Reports Server could only serve reports that were compatible with the environment in place when Reports Server was started. For example, reports had to be compatible with the value of the NLS_LANG parameter at the time Reports Server was started. This restriction meant that you needed to have one Reports Server running for each language you wanted to process. The new environment switching feature eliminates this restriction by enabling one instance of Reports Server to serve reports with any arbitrary environment settings, including language.

To use this new feature, you only need to add some elements to your Reports Server's configuration file.

1.2.8.1 environment Element

The environment element defines the characteristics (that is, environment variables) that you want to use to establish a particular runtime environment. By referencing the environment element's id, you invoke its settings. You can reference an environment element id from:

- The `defaultEnvId` attribute of the engine element in the Reports Server configuration file, to apply the corresponding environment settings to that engine when it starts up. Refer to defaultEnvId Attribute in this Release Note.
- The command line argument, `ENVID`, of your report's job request, which makes the environment settings only effective for that particular report job request. Refer to Examples in this Release Note.

The following is an example of an environment that you might want to invoke for the `defaultEnvId` element.

```
<environment id="JP">
  <envVariable name="NLS_LANG" value="Japanese_Japan.JA16SJIS"/>
  <envVariable name="NLS_CURRENCY" value="¥"/>
  <envVariable name="DISPLAY" value="MyServer.MyCompany.com:0.0"/>
</environment>
```

Usage Notes

- You may include as many environment elements as you need (for example, one for each language/territory you need to support). Inside an environment element, you can add as many `envVariable` elements as required.
- Each envVariable is specified as name–value pair. They can be either standard environment variables or user-defined environment variables.

1.2.8.2 defaultEnvId Attribute

`defaultEnvId` is an optional attribute of the engine element in your Reports Server configuration file. It specifies the default environment with which Reports Server starts an engine. The attribute takes an id that is associated an environment element in another part of the server configuration file (*`ORACLE_HOME`*`\reports\conf\server_name.conf`):

```
<engine id="rwEng" initEngine="1" minEngine="0" maxEngine="10" engLife="50"
maxIdle="30" defaultEnvId="JP"/>
```

The value `JP` identifies an environment element also specified in the server configuration file. The initial engines will be spawned with the environment settings specified in this environment element. For more information on the environment element, refer to environment Element.

Usage Notes

- `defaultEnvId` is optional. If you do not specify `defaultEnvId`, Reports Server spawns engines with the environment settings in force at startup time.
- Reports Server starts an engine with the environment variables specified in the referenced environment element plus whatever environment variables that Reports Server is running under.

1.2.8.3 Examples

The following examples illustrate the usage of the dynamic environment switching feature:

Example 1

Suppose that you need to run reports in Japanese from your Reports Server. An environment conducive to running reports in Japanese would include:

```
NLS_LANG = Japanese_Japan.JA16SJIS
```

The currency unit would be set to Yen (¥), the currency of Japan.

If the Server is running on UNIX, the `DISPLAY` variable would also need to be set.

To begin, you would have to add an environment element to your Reports Server configuration file that looks something like the following:

```
<environment id="JP">
  <envVariable name="NLS_LANG" value="Japanese_Japan.JA16SJIS"/>
  <envVariable name="NLS_CURRENCY" value="¥"/>
  <envVariable name="DISPLAY" value="MyServer.MyCompany.com:0.0"/>
</environment>
```

Once the environment element is in place, you could request a report with Japanese output using the following URL:

```
http://yourWebServer:port/reports/rwservlet?server=yourreportsserver
&REPORT=Japanese.rdf&userid=username/passwd@db&desformat=htmlcss
&DESTYPE=cache&ENVID=JP
```

When the URL is submitted to Reports Server, it detects the optional `ENVID` parameter and matches the specified id (in this case, JP) to the corresponding id in its configuration file. If Reports Server already has an engine running with these characteristics, it will reuse the existing engine to process the job. If not, then it spawns an engine using the current environment plus the three environment variables specified in the JP environment element. If spawning a new engine would cause Reports Server to exceed its `maxEngines` setting, then Reports Server shuts down an engine before starting a new one. An engine may be shut down even though it has not exceeded its engLife setting.

Once Reports Server has an engine with the correct environment running, the job is processed by that engine and the output is routed to the specified `DESTYPE`.

`envid` is an optional parameter. If you do not pass this parameter with the job, Reports Server processes the request using an engine started with the `defaultEnvId` environment. If `defaultEnvId` is not specified for the engine element in your Reports Server configuration file, then the engine will inherit the settings with which the Reports Server instance was started.

Reports Server may forward the request to another server on the cluster in cases where it cannot handle the request at that time. Even though the job may be routed to another server on the cluster, the target server will reuse or spawn an engine with the required environment to process this job.

Example 2

The following example illustrates how to use this environment switching feature to run an Arabic report on the same Reports Server that was used to run the Japanese report in Example 1.

Add another environment element to the Reports Server configuration file as shown below:

```
<environment id="AR">
```

```
 <envVariable name="NLS_LANG" value="Arabic_United Arab Emirates.AR8ISO8859P6"/>
 <envVariable name="NLS_CALENDAR" value="Arabic Hijrah "/>
</environment>
```

The Arabic report has to be submitted to Reports Server with the following command line:

```
http://yourWebServer:port/reports/rwservlet?server=yourreportsserver
&report=Arabic.rdf&userid=username/passwd@db&desformat=htmlcss
&destype=cache&envid=AR
```

Since the job is submitted with `envid=AR`, Reports Server finds or starts an engine with the environment specified by element AR in the Reports Server configuration file. The job is processed by the new engine and the output is distributed to the specified destination.

Example 3

The following example illustrates how the environment switching feature could be used in conjunction with a JSP report, that is, without the Reports Servlet.

Suppose that you have the following environment elements in the Reports Server configuration file:

```
<environment id="UK">
 <envVariable name="NLS_LANG"  value="AMERICAN_UNITED KINGDOM.WE8ISO8859P1"/>
</environment>
<environment id="US">
 <envVariable name="NLS_LANG" value="AMERICAN_AMERICA.WE8ISO8859P1"/>
</environment>
```

If your JSP report uses a format mask such as the following, it means the currency, grouping, and decimal symbols can change according to the environment:

```
<rw:field id="sal" src="sal" formatMask="L999G999D999"/>
```

To run the report using the UK symbols for currency, grouping, and decimal, you would use the following URL:

```
http://myserver:port/test/myjsp?userid=scott/tiger@orcl&envid=uk
```

> **Note:** You could place `envid=uk` into a key in the `cgicmd.dat` file.

Usage Notes

- Although this feature is ideal for handling reports of various languages, its application can be much broader. You could use it in any situation where a report requires a particular environment to execute correctly.
- Reports Server will start one or more engines for each environment id as and when it gets requests requesting specific environments. The total number of engines, however, cannot exceed the `maxEngine` specified for that engine type. It is recommended that you set `maxEngine` to a value greater or equal to the number of environment elements specified in Reports Server configuration file.
- For engines used by the in-process server, the order of precedence for environment variables from highest to lowest is as follows:

  - `reports.sh` (UNIX only)
  - `environment` element in the Reports Server configuration file
  - In the *`ORACLE_HOME`*`/j2ee/OC4J_BI_Forms/config/oc4j.properties` file: the oracle.display property defines the `DISPLAY` setting, `oracle.home` property defines the *`ORACLE_HOME`* setting, and `oracle.path` defines the `PATH` setting.
  - In *`ORACLE_HOME`*`/opmn/conf/opmn.xml`, the `<environment>` element under `<oc4j instanceName="OC4J_BI_Forms" gid="OC4J_BI_Forms">`
  - The system settings and registry (Windows only)

- For engines used by the standalone server, the order of precedence for environment variables from highest to lowest is as follows:
  - `reports.sh` (UNIX only)
  - `environment` element in the Reports Server configuration file
  - The environment set in the console where you start `rwserver.sh`
  - The system settings and registry (Windows only)

1.2.8.4 engineResponseTimeOut Parameter

A new, optional attribute, `engineResponseTimeOut`, has been added to the `engine` element in the Reports Server configuration file. This attribute enables you to specify the maximum amount of time (in minutes) for an engine to update the status of the job while running a report in your environment. If it takes longer than this amount of time to update the job status for some reason (for example, due to the engine hanging or a long blocking SQL query), Reports Server terminates the job.

`engineResponseTimeOut` has also been added as a new command line parameter for `rwservlet`, `rwclient`, and `rwcgi`. From the command line, it overrides the value specified in the `engine` attribute for that particular job request only.

1.2.8.5 DelimitedData DESFORMAT

Bug number: 1211760

A new destination format (`DESFORMAT`) is available in this patch that resolves a bug (1211760) with the earlier delimited formatting functionality.

1.2.8.5.1 Using DelimitedData Customers who had problems running large volume reports with `DESFORMAT=DELIMITED` should now use `DESFORMAT=DELIMITEDDATA`.

To use the DelimitedData `DESFORMAT` for the Generate to File > XML menu option in Reports Builder, set the following environment variable in the operating system:

```
REPORTS_OVERRIDE_FILEGENXML = Yes
```

This setting overrides the XML generation functionality in Reports Builder with the new DelimitedData behavior. Next release onwards, both formats will be available natively inside Reports Builder, but in this patch, you must toggle this environment variable to switch between the two behaviors.

1.2.8.5.2 Modifying DelimitedData Output The new DelimitedData driver runs off of the data model and operates in much the same way as the XML driver, which is already available in the product. Since the driver runs off of the data model, any formatting changes defined in the layout are not reflected in the DelimitedData output. You can set the following column properties (Tools > Property Inspector) to alter column names and exclude columns from the DelimitedData output file:

XML Tag can be used to enter a column alias.

Exclude from XML output can be used to exclude the column from the DelimitedData output.

The DelimitedData functionality also honors the `DELIMITED`,`CELLWRAPPER`, `NUMBERFORMAT`, and `DATEFORMAT` command line arguments just as the original Delimited functionality did. Refer to the command line documentation in the *Reports Builder online help* for details on these arguments.

Known Issues

This feature has the following known issues:

- 2400121: DELIMITEDDATA: REPORT LEVEL SUMMARY INCLUDED WHILE CREATING COLUMN LEVEL SUMMARY
- 2400115: DELIMITEDDATA: COLUMN DATA REPEATED FOR GROUP LEVEL SUMMARY COLUMN
- 2397366: DELIMITEDDATA: REPORT LEVEL COLUMNS DISPLAYED JUST BELOW THE GROUP LEVEL COLUMNS

1.2.9 Running a Report to Reports Server Asynchronously

Table 1–2 indicates which commands can use the `BACKGROUND` keyword.

Table 1–2 Commands that can use the BACKGROUND keyword

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

Description

`BACKGROUND` specifies whether a report on the server should be run synchronously (`NO`) or asynchronously (`YES`).

Syntax `BACKGROUND={YES|NO}`

Values

- `YES` Runs the report asynchronously. The client sends the call to the server, then continues with other processes without waiting for the report job to complete. If the client process is killed, the job is canceled.
- `NO` Runs the report synchronously. The client waits for the report to queue, be assigned to a runtime engine, run, and finish.

Default `NO`

Usage Note

If `BACKGROUND=YES`is used with `rwbuilder` or `rwrun`, a warning is issued and the keyword is ignored.

1.3 Information About Important Bug Fixes and Workarounds in Patch Release 4

This section outlines the important bug fixes and workarounds pertaining to Patch Release 4:

See Also: Appendix A, "Fixes and Workarounds for 9.0.2.4"

For more information on all the bugs fixed and workarounds required.

1.3.1 Running a multibyte report with the WE8ISO8859P15 character set

Bug No: 2906401

Applicable to

Solaris only

Description

You may see garbled output when you run a multibyte report that is designed in Windows on a Solaris machine with the `we8iso8859-15` character set.

Workaround

To work around this issue, you must do the following:

1. Make an entry for the fonts used in your default printer's `.ppd` file. This file is located in the following directory:

   ```
   $ORACLE_HOME/guicommon9/tk90/admin
   ```

   > **Note:** The entries are for the fonts that appears garbled in the report output.

2. Set the encoding scheme in the font's AFM file to `FontSpecific` if it is `AdobeStandardEncoding`. Thus, the following entry in the AFM file:

   ```
   EncodingScheme AdobeStandardEncoding
   ```

   must change to:

   ```
   EncodingScheme FontSpecific
   ```

1.3.2 Standardizing the error message template character set

Bug No: 3056717

Applicable to

All Platforms

Description

The character set of the error message template (`rwerror.htm`) is now `iso-8859-1` to ensure consistency across all platforms. Earlier, it was `windows-1252`, which is a character set specific to the Windows platform.

Note: The error message template and all other templates are located under the following directory:

```
$ORACLE_HOME/reports/templates /
```

1.3.3 Specifying the character encoding for reports output using rwservlet

Bug No: 2855714

Applicable to

All Platforms

Description

When you pass a multibyte character with a URL escape to `rwservlet`, the NLS characters appear garbled.

Fix

After you have downloaded the 9.0.2.4 patch, you can now specify non-ASCII escaped characters in the request URL or in the parameter form input. You must specify the character encoding in the `rwservlet.properties` file before you can apply it. This is to ensure that `rwservlet` uses the required encoding when parsing the parameter value.

You can set the value of `DEFAULTCHARSET` in the `rwservlet.properties` file to either:

- The NLS characterset (for example, `JA16EUC`)
- The IANA charset (for example, `EUC-JP`)

Example

```
DEFAULTCHARSET=JA16EUC
```

1.3.4 Specifying jobwise database connection information

Bug No: 3146008

Applicable to

All Platforms

Description

Reports Engine may stop responding after a long running report is executed multiple times. This is because the engine always maintains a connection with the database.

Fix

Patch Release 4 onwards, the server configuration file, *`server_name`*`.conf` includes a new engine property, `keepConnection`, for the default runtime engine implementation. This property tells the default runtime engine whether to retain the existing connection (`YES`) or discard it and reconnect (`NO`).

To enable the `keepConnection` property:

1. Uncomment the entry in the engine node of the *`server_name`*`.conf`, file.
2. Specify the appropriate value.

Values

- `YES` Retain the existing database connection information.
- `NO` Discard the existing database connection information and reconnect with the userid specified for the job.

Default

`YES`

Usage Notes

- The `keepConnection` property does not affect reports deployed using either `rwbuilder` or `rwrun`.
- The `keepConnection` property will be migrated if a *`server_name`*`.conf` file used in previous versions (for example, 9.0.2.x) runs in the current environment.

2

General Issues and Workarounds

This chapter describes general issues and their work arounds pertaining to previous patch releases, that is, Patch Release 1 and Patch Release 2 for Oracle9*i* Reports.

2.1 Migration

For information about migration, please refer to Oracle9*i* Application Server *Migrating from Oracle9iAS Release 1 (1.0.2.2.x) to Release 2 (9.0.2)*, part number A96157-01.

2.2 Deprecated Features from Earlier Versions

Some features available in Oracle Reports 6*i* that have been deprecated or removed from Oracle9*i* Reports.

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
- 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 removed from Oracle9*i* Reports:

- `RWRUNC` character mode runtime
- client/server GUI report previewer in rwrun
- 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 Oracle9*i* Reports Statement of Direction white paper available from the Oracle Technology Network, (`http://otn.oracle.com/products/reports/`).

2.3 reports.sh (UNIX)

On UNIX, this patch will overwrite `reports.sh`. If you have modified your current `reports.sh` file, you should save it and, after the patch is installed, merge your modifications into the version of `reports.sh` installed with the patch. The version installed with the patch contains some required changes.

2.4 Using the Hotspot JVM Instead of Classic

Reports 9.0.2 Patch 1 uses the Classic JVM instead of the Hotspot performance engine. This proved to be an issue when Reports Builder handled large xml/jsp reports. To work around this issue you need to specify the path (`ORACLE_HOME\jdk\jre\bin\hotspot`) to the Hotspot performance engine using the `PATH` environment variable.

On Windows : Alter the `PATH` environment variable to point to the Hotspot performance engine, instead of the Classic JVM.

2.5 Spaces in HTML Output

Bug number: 2441615

Description

The `REPORTS_NO_HTML_SPACE_REPLACE` environment variable specifies whether spaces should not be replaced with `&nbsp;` in `HTML` or `HTMLCSS` output.

Oracle9*i* Reports will map HTML metadata characters in the data retrieved for a field to the appropriate encoding. That is, Oracle9*i* Reports automatically maps: < to `&lt;` and " to `&quot;`. In most cases, the browser produces the correct results and handles the spaces correctly. In some cases, the browser's handling of spaces does not produce the required output. This happens in such cases as where the user has padded the front of the data to produce indentation. Since the browser will treat multiple spaces as single space, the indentation is lost.

Valid Values `YES |NO`

Default `NO`

Usage Notes

- (Windows only) Use the registry to specify the value. (UNIX only) Set the value as an environment variable (shell script).
- If the value is set `NO`, all spaces are replaced by `&nbsp;`. This could cause problems in your output where you want the browsers to handle line breaks on spaces. It will also increase the size of the generated HTML file.
- If a field's Contains HTML Tags property is set to `YES`, then no encoding will take place since Oracle9*i* Reports just passes the field's value through to the output.

Example

If `REPORTS_NO_HTML_SPACE_REPLACE` is set to `YES`, then the output for the sentence `[ Typical data output]` will be:

```
[ Typical data output]
```

and display as (ignoring preceding spaces):

```
[Typical data output]
```

Setting the environment variable to NO will cause the output to change to:

```
[&nbsp;&nbsp;Typical&nbsp;data&nbsp;output
```

and display as (maintaining preceding spaces):

```
[ Typical data output]
```

> **Note:** Brackets in the earlier example are used to show preceding spaces; they are not part of the sentence.

As part of this change, the previous functionality of the DELIMITER command line argument for HTML and HTMLCSS output is removed. DELIMITER is still used with the DELIMITED DESFORMAT as in previous releases.

2.6 Printing Reports Without Hardware-based Left Margins on Windows

Bug No: 387196

Description

(Windows only) From Patch Release 2 onwards, the REPORTS_ADD_HWMARGIN environment variable specifies whether to include the printer hardware-based left margin. By default, this margin is ignored. The printing origin starts from the top-left corner (0,0) of the physical paper and not the printable area. This is to facilitate the design of printer-independent independent reports.

Valid Values YES | NO

Default NO

Usage Notes

- (Windows only) Use the registry to specify the value. (UNIX only) Set the value as an environment variable (shell script).
- You must ensure that your report's layout contains enough margin spacing such that your data falls within the printable area. Margin fields in the Page Setup dialog box of Reports Builder have been disabled to ensure consistency with Oracle9*i*AS Reports Services
- To revert to the old behavior of including the hardware margin, set the value to YES.

2.7 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.

2.8 PDF Font Sub Setting

In Oracle9*i*AS Reports Services, you can subset the TrueType fonts for multibyte, Unicode, and single byte PDF output. This functionality and more information about PDF report output can be found in the white paper, Using PDF with Oracle9*i* Reports, available from the Oracle Technology Network (`http://otn.oracle.com/products/reports/`).

2.9 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`.

2.10 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.NumberExpection:G
```

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

2.11 Oracle9*i*AS Portal Portlet/Page Parameters

Within Oracle9*i*AS Portal, you can choose whether to make a report's parameters visible to users through the Customization page of a Reports Definition File Access component.

To make a report's parameters visible to users:

1. Click **Customize** at the bottom of the Manage Component page for the report
2. Click **Customize** at the bottom of the Manage Component page for the report

Note: Click **Customize** at the bottom of the Manage Component page for the report.

If the parameter you are exposing has a corresponding Oracle9*i*AS 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 Oracle9*i*AS Portal, refer to the Oracle Technology Network (`http://portalcenter.oracle.com`)

2.11.1 Customize Page

The SSOCONN and CGI/servlet key parameters are missing from the JSP report Customize page in Oracle9*i*AS Portal The workaround for this issue is to add these missing parameters in the Additional User Parameters field.

2.12 Report Portlet and Netscape Communicator 4.7 on UNIX

In Oracle9*i*AS Portal with Netscape Communicator 4.7 on UNIX, 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 on Windows or in Netscape Communicator 6.0.

2.13 Setting the Size of the Report Portlet on Windows

If you run a report portlet on Windows through the servlet, you can control its size by specifying the Portlet Width and Portlet Height parameters in Additional User Parameters on the Customize page for the Reports Definition File object. The value of these parameters may be a percentage (%) or a number of pixels.

For example, you can enter:

Portlet Width: 90%

Portlet Height: **480**

If no value is specified, Oracle9*i*AS Reports Services uses its default value (640 pixels wide and 320 pixels high). This functionality is not available on UNIX platforms yet.

2.14 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 Oracle9*i*AS Reports Services 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 earlier example, you could switch the positions of the distribute and destination parameters in your URL.

2.15 Order of Parameters in a URL

When using key mapping, the order in which the parameters are substituted from the URL into the key is determined by the placement of CMDKEY in the URL. For example, suppose you have a key such as the following one in the cgicmd.dat file:

```
mykeys: DEPTNO=%1 MYPARAM=%2
```

Now, you execute a JSP report that references this key as follows:

```
http://neptune.world.com:80/jsp/myreport.jsp?userid=scott/tiger@hrdb&cmdkey=mykeys
&10&test
```

Because of the placement of CMDKEY in this URL, the 10 corresponds to %1 and test corresponds to %2. Even though they are not the first and second parameters in the URL, 10 and test are the first and second parameters to follow CMDKEY in the URL.

2.16 Accessing Web Commands With DIAGNOSTIC set to NO

Setting DIAGNOSTIC=NO causes the Web commands display, to be suppressed. This posed a problem for EM as it obtained Reports Server information from the Web commands. The remedy is to conditionally enable administrators whose username and password matches the values stored in the <identifier> element to access Web commands, even when DIAGNOSTIC is set to NO. If the values do not match then an error message similar to the following is displayed:

```
REP-52262: Diagnostic output is disabled.
```

The <identifier> element is set in the Reports Server configuration file by the first Web command call if it is not defined in the server configuration file and the values are automatically encrypted when Reports Server is started. The format to preset the <identifier> element in the Reports Server configuration file is:

```
<identifier confidential="yes" encrypted="no">username/password</identifier>
```

The clear text of username and password will be encrypted by Reports Server once it starts up, and the encrypted atttribute is changed to yes to indicate the content is encrypted.

Refer to *Oracle9iAS Reports Services Publishing Reports to the Web*, for more information on how to change the values of the <identifier> element.

2.17 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.

2.18 Oracle9*i*AS Portal Security, Portal Destination, and Job Status Repository

If you use these features, the JDBC database connections made by Oracle9*i*AS 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.

You can also use the new environment switching functionality to dynamically set the environment for reports. Refer to Dynamic Environment Switching.

2.19 Import Oracle9*i*AS Portal Component

In order import a report component into Oracle9*i*AS Portal, you must update the import_portal_config.xml file to point to the correct *ORACLE_HOME*. import_portal_config.xml is located in:

```
ORACLE_HOME/reports/templates
```

In the file, you need to find the following line and replace `orawin90` with your *ORACLE_HOME*.

```
<template_base_dir>/orawin90/reports/templates</template_base_dir>
```

2.20 JDeveloper Integration

This sections discusses the various issues that occur with JDeveloper integration.

2.20.1 Deadlock when debugging JSP reports in Oracle JDeveloper

When debugging an Oracle9*i*AS Reports Services JSP in Oracle JDeveloper, you must disable any breakpoints that are not of source type. If you do not disable these breakpoints, a deadlock is detected. You can disable breakpoints that are not of source type in the Breakpoints window.

2.20.2 Graph not appearing in Oracle JDeveloper

When running a JSP report with the rw:graph tag in Oracle JDeveloper, the image is not visible in the browser. To fix this problem, set the IMAGEURL in the rwservlet.properties file in *ORACLE_HOME*`/reports/conf`:

```
IMAGEURL=http://host:port/Workspace-Project-context-root/servlet/
oracle.reports.rwclient.RWClient
```

2.20.3 Reports In-Process Server

If the Java Virtual Machine (JVM) is set to ojvm, minimal, or vanilla in Oracle JDeveloper's Virtual Machine settings (Project Settings > Runner > Virtual Machine), the rwservlet's in-process server is not killed when JDeveloper's embedded OC4J server is terminated. As a result, a port conflict occurs the next time OC4J is started. To avoid this problem, do one of the following:

- Set the JVM to Java Virtual Machine to hotspot.
- Disable the in-process server in *ORACLE_HOME*`/reports/conf/rwservlet.properties` and use the standalone Reports Server instead.

2.21 Oracle9iAS Forms Services Integration

The default configuration for Oracle9iAS Forms Services does not run in Single Sign-On (SSO) mode. The default configuration for Oracle9*i*AS Reports Services does run in SSO mode.

Forms applications calling integrated Oracle9*i*AS Reports Services using the Run_Report_Object built-in will not experience any problems when Oracle9iAS Forms Services is running in non-SSO mode and Oracle9*i*AS Reports Services is running in SSO mode as long as the Reports Server and the requested report are not registered in Oracle9*i*AS Portal.

Other Requirements

- The property, Reports Server, must be set explicitly for all report objects in the Oracle9*i* Forms module.
- If a Reports Server other than the default is being used, that server must be started from the command line as follows:

```
rwserver server=Reports Server name
```

- The system variable, `REPORTS_PATH`, must be modified in the file *`ORACLE_HOME`*`/bin/reports.sh` to reference the path of the reports to be run.
- The first time a Reports Server is started, it creates a configuration file called *`ORACLE_HOME`*`/server/conf/`*`server_name`*.conf.
- The default status of a Reports Server is secure. To change the Reports Server status to non-secure, modify *`ORACLE_HOME`*`/server/conf/`*`reports_server_name`*`.conf` by commenting out the `<security>` tag and removing securityId from the `<job>` tags.
- After making these modifications, the Reports Server must be stopped and restarted.
- If Oracle9*i*AS Forms Services is configured to run in SSO mode, then report requests are sent with the authid provided, based on the SSO user login.
- Protected reports and Reports Servers can be registered in Oracle9*i*AS Portal.

The Table 2–1 lists the possible Forms/Reports combinations and expected results:

Table 2–1 Outcome of Forms/Reports Integration when Forms is Running in SSO Mode or Non-SSO Mode

Report Type	Registered, Secure Reports Server (runs only registered reports)	Registered, Secure Reports Server (runs any reports)	Non-SecureReports Server
Reports with public access	report generated	report generated	report generated
Reports with specific user access	report generated	report generated	report generated
Reports with no specific user access	report not generated	report not generated	report generated
Non-registered reports	report not generated	report not generated	report generated

2.22 URL Encoding

To ensure that spaces and control characters are passed correctly, you may need to turn URL encoding on or off for the fields in your report. You can turn URL encoding on or off with the `RW:FIELD` tag in a report:

```
<rw:field
...
urlEncode=yes|no
...
/>
```

The default value for `urlEncode` is no.

2.23 Command Line Arguments

This section describes the various command line options that are currently applicable.

2.23.1 ENVID

Refer to Section 1.2.8, "Dynamic Environment Switching" for more information.

2.23.2 engineResponseTimeOut

Refer to Section 1.2.8.4, "engineResponseTimeOut Parameter" for more information.

2.23.3 SUPPRESSLAYOUT Keyword

Table 2–2 indicates which commands can use the SUPPRESSLAYOUT keyword.

Table 2–2 Commands that can use SUPPRESSLAYOUT

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

Description Use SUPPRESSLAYOUT to to specify whether to suppress the formatting of the paper layout at runtime. The keyword allows users to control whether the paper layout in a report is executed at runtime. The most common use of this keyword is to increase the performance of JSP reports. Since a JSP report may have a paper layout and reference objects in it through an <rw:include> tag, Oracle9*i* 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 Paper layout objects will not be formatted at runtime

NO Paper layout objects will be formatted at runtime.

Default NO

2.23.4 UPGRADE_PLSQL

Table 2–3 indicates which commands can use the UPGRADE_PLSQL keyword.

Table 2–3 Commands that can use UPGRADE_PLSQL

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

Description Use UPGRADE_PLSQL to specify whether to upgrade any PL/SQL code in the report to the latest version required by Oracle9*i* Reports Developer.

Syntax `UPGRADE_PLSQL=[YES|NO]`

Values

YES PL/SQL code will be upgraded automatically if necessary.

NO PL/SQL code will not be updated.

Default YES

2.23.5 RECURSIVE_LOAD

Table 2–4 indicates which commands can use the RECURSIVE_LOAD keyword.

Table 2–4 Commands that can use RECURSIVE_LOAD

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

Description Use RECURSIVE_LOAD to specify whether to validate external references in program units when running a report. If any of the references become invalid, the program unit is 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 External references will be validated.

NO External references will not be validated.

Default YES

2.23.6 SQLTRACE

Table 2–5 indicates which commands can use the SQLTRACE keyword.

Table 2–5 Commands that can use SQL_TRACE

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

Description Use SQLTRACE to specify whether to perform SQL tracing on your report without modifying the report definition.

Syntax SQLTRACE=[YES|NO]

Values

YES SQL tracing will be performed on the report.

NO SQL tracing will not be performed on the report.

Default NO

2.24 Built-ins

This section outlines the various SRW functions and procedures that are currently applicable.

2.24.1 SRW.GET_VALUE

SRW.GET_VALUE is equivalent to the Oracle9*i* 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;
```

2.24.2 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;
```

2.25 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.

2.26 REPORTS_NLS_XML_CHARSETS

Description This environment variable provides an override option to enable you to define the character set encoding used when saving a report in XML format. This is only required where the required character set mapping for Oracle9*i* Reports NLS_ LANG to XML IANA character sets do not produce the required results.

Valid Values Set of mapping pairs separated by semicolons. The first value is the encoding that is being produced and the second mapped value is the value that should be used for these cases.

Default Not Defined

Usage Note This environment variable should not be required and is currently only used as a workaround to Bug 2224413:

```
NLS:NEED STABLE MAPPING FROM ORACLE'S NLS_CHARACTER_SET TO IANA ENCODING
```

Example `WINDOWS-936=WINDOWS-1252;CSEUCKR=EUC-KR;....`

2.27 Java Importer

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

- Add the necessary jar files to the `REPORTS_CLASSPATH` (for 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\*server_name*.conf`).

2.28 Buttons

If you open an existing report that contains buttons in Oracle9*i*AS Reports Services, the buttons will be converted into text items. You cannot add new buttons in Oracle9*i*AS Reports Services.

2.29 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 Reports Builder, but you have to create your own parameter form for runtime.

2.30 RTF Output in Microsoft Word 95 for Japanese

When you open the RTF output from Oracle9*i*AS Reports Services 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 for Japanese and do not occur in Microsoft Word 97 for Japanese.

2.31 Bold Korean Font Not Appearing Correctly

In JDK 1.3.1, a bug causes the bold Korean font to appear incorrectly. Oracle9*i*AS Reports Services uses JRE 1.3.1 and therefore all bold Korean strings in graphs within reports show up incorrectly

3

Configuration Issues and Workarounds

This chapter describes configuration issues and their work arounds pertaining to previous patch releases, that is, Patch Release 1 and Patch Release 2 of Oracle9*i*AS Reports Services.

3.1 Reports Configuration Assistant Fails during Installation

During installation, if the Oracle9*i* Reports Configuration Assistant fails and gives the following message:

```
Process destroyed exception (in the installer configuration palette)
```

then perform the following procedures:

Remove Security Page:

1. Log on to Oracle9*i*AS Portal.
2. Click Builder.
3. Click Navigator.
4. Click Contents for the Portal Design-Time Pages page group.
5. Click Pages.
6. Click Delete for the Oracle Reports Security page.

Remove Security Provider:

1. Log on to Oracle9*i*AS Portal.
2. Click Builder.
3. Go to the Build tab by clicking Build.
4. In the Providers portlet, type ORACLE REPORTS SECURITY in the name field.
5. Click Delete.

Run rwaddpag.sql:

1. Log on to the Oracle Internet Directory and obtain the Oracle9*i*AS Portal user database password.
2. Log on to the metadata repository as the Oracle9*i*AS Portal user and run the following script:

   ```
   ORACLE_HOME/portal/admin/plsql/wwd/rwaddpag.sql
   ```

This creates the Oracle9*i* Reports portlet in Oracle9*i*AS Portal.

Alternatively, you can create a tnsnames entry and execute the script as the Oracle9*i*AS Portal user with the user password obtained from the Oracle Internet Directory in Step 1.

3.2 Reports CGI Not Starting on Windows

After installation, if you are planning to use the Oracle9*i* Reports CGI, you should test it with the following URL:

```
http://server_name.domain:port_number/cgi-bin/rwcgi.exe/help?
```

This URL may fail with the following error:

```
Premature end of script headers: $ORACLE_HOME/bin/rwcgi.exe
```

If it does, you need to add the bin directory and the Java Virtual Machine's .dll path to the Oracle HTTP Server's path. You can do this by adding the following entry to `$ORACLE_HOME\opmn\conf\opmn.xml`:

```
<environment>
  <prop name="PATH"
     value="%ORACLE_HOME%\bin;%ORACLE_HOME%\jdk\jre\bin\classic"/>
</environment>
```

3.3 In-process Server Not Starting on Linux

On Linux, the in-process server does not start. To resolve this issue, you need to get the current value of the `$HOME` environment variable from your linux terminal session and then add the `$HOME` variable to the reports.sh script in `$ORACLE_HOME/bin`:

```
HOME=env_value_of_HOME; export HOME
```

3.4 Reports Server Installed with Reports Builder

Reports Builder requires an instance of the Reports Server. Hence, when you install 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.

3.5 Running JSP Reports from Oracle9*i*AS Portal

Bug number: 2224816

When you run JSP reports from Oracle9*i*AS Portal, you need to have a complete path for the CSS file because Oracle9*i*AS Reports Services and Oracle9*i*AS Portal have different virtual directories. For example, if you use css/my.css for a JSP report in Oracle9*i*AS Portal, the browser will construct the URL as follows:

```
http://server:port/portal-path/css/foo.css
```

This URL will not work. The URL needs to be as follows, which requires you to enter a fully qualified path in the JSP report:

```
http://server:port/reports/css/foo.css.
```

3.6 REPORTS_CLASSPATH

Oracle9*i*AS Reports Services 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 Oracle9*i*AS Reports Services 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.

3.7 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 Oracle9*i* Application Server.

3.8 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, Oracle9*i*AS Reports Services 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.

3.9 Oracle Reports 6i Access to Oracle9*i*AS Reports Services

Oracle9*i*AS Reports Services enables you to run Oracle6i Graphics for backward compatibility purposes. Therefore, the Oracle6i Graphics charts in a Reports6i report should continue to run correctly when the report is opened and run in the Oracle9*i*AS Reports Services environment. To ensure that this functionality works, you must:

On All Platforms:

Make sure the tnsname entry used by Oracle9*i* Reports exists in both the Oracle9i and Oracle6i tnsnames.ora file in *`ORACLE_HOME`*`/network/admin`.

On Solaris and other Unix platforms:

If you encounter the following error:

```
SSL fatal error: cannot execute g90runm
```

You must do the following:

1. Specify the path of your `ORACLE6i_HOME` in the `ORACLE_GRAPHICS6I_HOME` variable in g90runm.sh and ensure that your `ORACLE6i_HOME/bin` directory is in that path.
2. In `$ORACLE_HOME/bin/g90runm.sh`, create a link to `g90runm` as:

   ```
   ln -s g90runm.sh g90runm
   ```

On Windows

1. Ensure that, in your registry, `ORACLE_GRAPHICS6I_HOME` points to the `ORACLE6i_HOME\bin` directory.
2. Add `ORACLE6i_HOME\bin` to the `PATH` environment variable.

4

UI Issues and Workarounds

This chapter describes user interface issues and their work arounds pertaining to previous patch releases, that is, Patch Release 1 and Patch Release 2 of Oracle9*i*AS Reports Services.

4.1 Source Control (Windows only)

To enable the use of Oracle Source Control Management with Oracle9*i*AS Reports Services, 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 9*i* 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 Oracle9*i*AS Reports Services.

4.1.1 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.

4.2 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]
.....ja16sjis="MS Gothic"
.....ar8mswin1256="Courier New"
```

4.3 Find/Replace in Web Source View for Asian Languages Causes Crash

When performing a Find/Replace operation in the Web Source view of the Report Editor and running in an Asian language (for example, Japanese) on Windows, Reports Builder crashes. This problem does not occur on UNIX.

4.4 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 with Oracle9*i* Reports 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 (for example, rwclient).
- The PAGESIZE parameter's value will be applied to all sections in the report (header, main, and trailer).
- In order to run the barcode example report that is referenced in the Oracle9*i*AS Reports Services Getting Started with Oracle9*i* Reports on the Oracle Technology Network (http://otn.oracle.com/products/reports/) and the Oracle9*i* Reports Building Reports manual, you must:
  - Add oraclebarcode.jar to the REPORTS_CLASSPATH (for 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 multibyte report for multibyte 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* with Oracle9*i*AS Reports Services, 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* with Oracle9*i*AS Reports Services , 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. This chapter describes how to configure Oracle Enterprise Manager for your Reports Server as well as the Reports Server pages available in Oracle Enterprise Manager.

You will find more information on the Reports Server in Publishing Reports to the Web with Oracle9*i*AS Reports Services, included on the Oracle9*i*AS documentation CD.

5.1 Oracle Enterprise Manager Configuration

After installation, the Reports Server instance is not accessible from Oracle Enterprise Manager. The reason for this problem is that the password set for the reports target is not the Oracle9*i* Application Server administrator password.

To fix this problem, do the following:

1. Open the file `$ORACLE_HOME/sysman/emd/targets.xml`.
2. Search for the reports target. The target type for reports is oracle_repserv. The target definition will contain a property for UserName, for example:

   ```
   <Property NAME="UserName" VALUE="repadmin"/>
   ```

3. Change the `VALUE` to be the Oracle9*i* Application Server admin name, for example:

   ```
   <Property NAME="UserName" VALUE="orcladmin"/>
   ```

4. Search for the Password property of the Reports Server, for example:

   ```
   <Property NAME="Password" VALUE="3f769c1e7cfd7411"  ENCRYPTED="TRUE"/>
   ```

5. Change the `VALUE` to be the Oracle9*i* Application Server admin password and set `ENCRYPTED = ”FALSE”` for example:

   ```
   <Property NAME="Password" VALUE="adminpasswd"  ENCRYPTED="FALSE"/>
   ```

6. Search for the server name property of the Reports Server. For example, if the Reports Server is installed on the machine prod1-sun in domain us.oracle.com, it would have the following server property:

   ```
   <Property NAME="Server" VALUE="rep_prod1-sun.us.oracle.com"/>
   ```

7. Delete the domain portion of the server name. For example, in the preceeding case, you would remove us.oracle.com:

   ```
   <Property NAME="Server" VALUE="rep_prod1-sun"/>
   ```

8. Restart Oracle Enterprise Manager:

```
# $ORACLE_HOME/bin/emctl stop
# $ORACLE_HOME/bin/emctl start
```

Oracle Enterprise Manager will automatically encrypt the password and set the `ENCRYPTED` flag to `TRUE`.

5.2 Reports Server Main Page

This page summarizes the status of the selected Reports Server.

Table 5–1 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.
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 5–1 (Cont.) 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 TimeLists the average number of milliseconds it takes for the listed Reports Server to process a request from the client.
Performance Section	
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	

Table 5–1 (Cont.) Reports Server Main Page

Setting	Description
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.

5.3 Reports Server Performance Page

This page provides performance details about the selected Reports Server

Table 5–2 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.
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.

Table 5–2 (Cont.) Reports Server Performance Page

Setting	Description
Number of Running Engines	Provides the total number of this type of engine that is currently running on the selected Reports Server.

5.4 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 5–3 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.

5.5 Reports Server Scheduled Job Queue Page

This page functions very much like Reports Server Queue page. Refer to the Section 5.4, "Reports Server Queue Page" for more information.

5.6 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.
3. 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 5–4 Reports Server Finished Job Queue Page

Setting	Description
Select	Use this radio button to select a particular job. On this page, this function is most useful for selecting a report and: 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	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. 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.
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).

Table 5–4 (Cont.) Reports Server Finished Job Queue Page

Setting	Description
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.

5.7 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.

5.8 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.

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 5–5 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: 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.
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.

5.9 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 5–6 Job Trace Page

Setting	Description
Job Id n	Identifies the job by the number the Reports Server assigned to it at runtime.

Table 5–6 (Cont.) Job Trace Page

Setting	Description
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.

5.10 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.

5.11 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

Oracle Express Pluggable Data Source

This chapter describes how to configure and use the Express Pluggable Data Source with Oracle9*i*AS Reports Services.

6.1 Before You Begin

To use the Express Pluggable Data Source within Oracle9*i*AS Reports Services, you must connect to a supported Express Server version, which are versions 6.2.x and 6.3.x. When you install Reports9*i*, 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

This section outlines the necessary information that you need to keep in mind when you go about preparing the 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 Oracle9*i* Reports.

To use an existing Express Connection Editor installation:

1. Navigate to the `/olap` subdirectory in the Oracle home directory for Oracle9*i* 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 Oracle9*i* 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 to 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 Oracle9*i* 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 Oracle9*i* 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 Oracle9*i* 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 Oracle9*i* 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 Oracle9*i* 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 Oracle9*i* Reports Developer for *`ORACLE_HOME`*:

```
[General]
ConnectionPath=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 Oracle9*i* 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 6–1 XCF File Settings

Setting	Description
ConnectionType	Enter the type of connection: ■ 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 Oracle9*i* Reports Developer.
ServerType	Enter 1 for Express Server 6.x. No other values are applicable to connections from Oracle9*i* 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. The Express Connection Editor includes more than just the server machine name in the connection file, which is unnecessary for connections from Oracle9*i* 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 6–2 Oracle Express Relational Access Manager Settings

Setting	Description
ConnectionType	Enter 0. No other values are applicable to connections from Oracle9*i* Reports.
MasterDB	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: ■ 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. 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.

Table 6–2 (Cont.) Oracle Express Relational Access Manager Settings

Setting	Description
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 a 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 applies 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 *ORACLE_HOME*`/reports/jlib` directory and add the following line:

  ```
  options.snapi.FetchAlloc=size
  ```

  where size is a value greater that 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 Oracle9*i*AS Reports Services 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 Oracle9*i*AS Reports Services.
  - 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.

A

Fixes and Workarounds for 9.0.2.4

This appendix outlines the following:

- Bugs that were fixed in Oracle9i Reports Release 4 (9.0.2) (Table A–1).
- Bugs that require a workaround in Oracle9i Reports Release 4 (9.0.2) (Table A–3).

> **Note:** You must have a Metalink user id and password to access the bug information provided.
>
> To access bug information from Metalink, do the following:
>
> 1. Navigate to the following link: `http://metalink.oracle.com`
> 2. Log in with your Metalink user id and password.
> 3. Click Bug on the left pane.
> 4. Enter the Bug Number.

Table A–1 Bugs fixed in 9.0.2.4

Bug Number	Bug Description
3200306	POSTSCRIPT OUTPUT DOES NOT PRINT CORRECTLY WHEN PARAGRAPH IS FLUSH JUSTIFIED
3170571	WRONG CONTENT TYPE SET WHEN EMAILING PDF
3162793	RETURN STATUS FORMAT IS WRONG OCCATIONALLY IN CLUSTER ENV
3153558	OUTPUT COURIER NEW/ARIAL CHANGES TO TIMES NEW ROMAN IN RTF FORMAT
3146008	REPORTS SERVER REQUIRES ENGINE TO ALWAYS RECONNECT
3122650	FORMS NEED API TO GET REPORTS SERVER NAME AFTER CONNECTION USING CLUSTER NAME
3104655	FLUSH JUSTIFY DOES NOT WORK IN POSTSCRIPT IF PARAM EMBEDED IN BOILERPLATE TEXT
3103660	PASSING DOUBLE QUOTES FOR CELLWRAPPER WHEN USING RWSERVLET GIVES REP-50003
3069222	UNEXPECTED LINE BREAK IN RTF OUTPUT
3056717	RWERROR.HTM CONTAINS CHARSET AS WINDOWS-1252
3033165	FONT SUBSETTING IN REPORTS WITH MULTINATIONAL FONT MAPPING INCORRECTLY

Table A–1 (Cont.) Bugs fixed in 9.0.2.4

Bug Number	Bug Description
3017452	ENGINE CRASHES ITH ERROR FOR JDBC REPORTS IN IAS90201 + IAS PATCH 9023
3017227	MISALIGNMENT HAPPENS IN RTL PDF SUBSETTED REPORTS
2968284	START JUSTIFY DOESN'T WORK WITH FONT SUBSETTING WHEN DIRECTION IS RTL
2934541	KILLJOBID FAILS TO KILL THE JOB , EVENTUALLY HANGS THE SERVER
2906401	FONT ALIASING NOT WORKING FOR ALL FIELDS WITH WE8ISO8859P15 CHARACTERSET
2869632	SOLID LINES SHOWN AS DASHED LINES FOR SOME PDF REPORTS
2855714	RWSERVLET: URL ESCAPE FOR PARAMETERS
2762567	REPORTS ENGINE HANGS WHEN RUNNING REPORT WHICH USES SUBSETTING PDF
2713695	NLS CHARACTERS FOR PARAMETERS CHANGE TO QUESTION MARKS WHEN PASSED ON URL BAR
2382720	PROPERTY CONTAINS HTML FOR PAPER OBJECTS IS NOT SAVED IN JSP FORMAT

Table A–2 Bugs fixed in 9.0.2.4 on LINUX

Bug Number	Bug Description
2873939	ENGINE CRASHES (FREEZES) SOON AFTER STARTUP ON LINUX

Table A–3 Bugs that require workarounds in 9.0.2.4

Bug Number	Bug Description
2659642	TEXT NOT ALIGNED TO COLUMN WITH HEBREW R-T-L REPORT WITH PDF SUBSET