Oracle8™ ConText® Cartridge

Workbench User's Guide

Release 2.4

July 1998

Part No. A63822-01

ORACLE®

Enabling the Information Age™

Oracle8 ConText Cartridge Workbench User's Guide, Release 2.4

Part No. A63822-01

Release 2.4

Copyright © 1996, 1998, Oracle Corporation. All rights reserved.

Primary Authors: D. Yitzik Brenman, Graham Hoare

Contributors: Dave Allewell, John Barry, Roy Clarke, Franco Cravero, Colin McGregor

Contents

3   Document Viewing

4   Input/Output (I/O) Utility

5 CTXQUERY Forms Sample Application

6 TextServer3 Dictionary Migration Tool

A   Viewer Cartridge: Manual Configuration

B   Configuration Manager: Manual Configuration

Index

Send Us Your Comments

Oracle8 ConText Cartridge Workbench User's Guide, Release 2.4

Part No. A63822-01

Oracle Corporation 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 any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments to us in the following ways:

- FAX - (650) 506-7200.   Attn: Oracle8 ConText Cartridge Documentation
- Postal service:
  Oracle Corporation
  Oracle8 ConText Cartridge Documentation
  500 Oracle Parkway
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, and telephone number below.

If you have problems with the software, please contact your local Oracle World Wide Support Center.

x

Preface

This manual introduces the Oracle8 ConText Cartridge Workbench components and provides a description of each, as well as any setup and usage information that is needed for each component.

Audience

This manual is intended for system administrators responsible for installing and maintaining the ConText Cartridge Workbench Release 2.4.5 components. It is also intended for ConText administrators and application developers who use the ConText Cartridge Workbench components to administer ConText or develop applications.

Prerequisites

This document assumes that you have experience with the Oracle relational database management system, SQL, SQL*Plus, PL/SQL, and with Web-based software such as Oracle Web Request Broker and Oracle Web Application Server. See the other documentation provided with your hardware and software for additional information.

Related Publications

For more information about ConText, see:

- *Oracle8 ConText Cartridge QuickStart*
- *Oracle8 ConText Cartridge Application Developer's Guide*
- *Oracle8 ConText Cartridge Administrator's Guide*
- *Oracle8 Error Messages*

For more information about Oracle8, see:

- *Oracle8 Concepts*
- *Oracle8 Administrator's Guide*
- *Oracle8 Utilities*
- *Oracle8 Tuning*
- *Oracle8 SQL Reference*
- *Oracle8 Reference Manual*
- *Oracle8 Application Developer's Guide*

For more information about PL/SQL, see:

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

How This Manual Is Organized

The manual is divided into six chapters and two appendices:

Chapter 1: Introduction

This chapter introduces the ConText Workbench components and provides an overview of their features.

Chapter 2: Administration Tools

This chapter describes the administration tools that are contained in the ConText Workbench, and shows how they are used.

Chapter 3: Document Viewing

This chapter describes the document viewers that are contained in the ConText Workbench, and shows how they are used.

Chapter 4: Input/Output (I/O) Utility

This chapter describes the I/O utility and provides guidance for its use.

Chapter 5: CTXQUERY Forms Sample Application

This chapter presents the CTXQUERY Oracle Forms sample application.

Chapter 6: TextServer3 Dictionary Migration Tool

This chapter describes the TextServer3 Dictionary Migration Tool and provides guidance for its use.

Appendix A, "Viewer Cartridge: Manual Configuration"

This appendix details the manual configuration and deconfiguration procedures for the Viewer Cartridge.

Appendix B, "Configuration Manager: Manual Configuration"

This appendix details the manual configuration and deconfiguration procedures for the Configuration Manager.

Type Conventions

This manual adheres to the following type conventions:

Type	Meaning
UPPERCASE	Uppercase letters indicate Oracle commands, standard database objects and constants, and standard Oracle PL/SQL procedures.
lowercase italics	Italics indicate variable names, URLs, names of user objects (tables, views, preferences, policies, etc.), PL/SQL parameter/argument names, and names of example PL/SQL procedures, especially when these elements occur within normal text.
	Italics also indicate emphasis.
`monospace`	Monospace type indicates example commands, strings, codes etc. Type in the command, string, or code exactly as it appears.

Further notational conventions are described in Appendices A and B.

Customer Support

You can reach Oracle Worldwide Customer Support 24 hours a day.

In the USA: **1.650.506.1500**

In Europe: **+ 44.344.860.160**

Please be prepared to supply the following information:

- your CSI number (This helps Oracle Corporation track problems for each customer)

- the release numbers of the Oracle Server and associated products

- the operating system name and version number

- details of error numbers and descriptions (Write down the exact errors you encounter)

- a description of the problem

- a description of the changes made to the system

You can also access the Oracle Support web site at *http://www.oracle.com/support/*

Your Comments Are Welcome

Please use the "Send Us Your Comments" form at the front of this document to convey your comments to us. You can also contact us at:

Documentation Manager
Oracle8 ConText Cartridge
Oracle Corporation
500 Oracle Parkway
Redwood Shores, California 94065

1

Introduction

This chapter provides brief descriptions of the Oracle8 ConText Cartridge Workbench components, as well as general installation and configuration information.

The following topics are covered in this chapter:

- ConText Workbench Overview

- Installing the ConText Workbench

- Configuring the ConText Workbench

> **Note:** Detailed configuration information for the ConText Workbench components is located in the respective chapter for each component.

ConText Workbench Overview

The Oracle8 ConText Cartridge Workbench is included in the Oracle8 client, which is automatically provided with every Oracle8 (and Oracle8 Enterprise Edition) distribution. In a Windows NT environment, the Oracle8 client is part of the Oracle8 CD. In all other environments (i.e. UNIX-based environments), the Oracle8 client is provided as a separate CD.

The ConText Workbench consists of a set of tools and utilities that provide a client-side interface to ConText installed on a server host machine. The tools and utilities that comprise the ConText Workbench can be used to perform the following tasks from any client on which the ConText Workbench has been installed and configured:

- Administration
- Document Viewing
- Document Input/Output
- Sample Applications
- TextServer3 Migration

Administration

Administration can be carried out from the client-side via the ConText Workbench administration tools:

- Configuration Manager
- System Administration tool

The **Configuration Manager** is a platform-independent, HTML-based application that works with Oracle WebServer or Web Application Server. It offers administration facilities for the CTXSYS user, as well as querying facilities and a way of generating prototype applications. Once configured for a database, it can be used from multiple platforms to administer the database.

In contrast, the **System Administration tool** is a Windows-based application that connects to Oracle8 using Net8 and can be used by any users with the CTXADMIN or CTXAPP role to perform administration tasks and text setup for multiple databases.

> **See Also:** For more information, see Chapter 2, "Administration Tools"

Document Viewing

Viewing of documents can be carried out in a client/server environment or in a three-tier environment (via Oracle WebServer or Web Application Server). The ConText Workbench provides separate utilities for use in the two supported environments:

- Viewer Control

- Viewer Plugin

The **Viewer Control** is an OCX custom control that can be embedded either in Windows applications or in HTML pages where the browser supports Microsoft's ActiveX controls.

The **Viewer Plugin** is a Netscape client-side plugin that allows documents to be displayed in native format on the World Wide Web using browsers that support plugins.

In addition, the ConText Workbench provides the **Viewer Cartridge**, an Oracle WebServer/Web Application Server cartridge that provides the Viewer Plugin with documents for viewing. It assembles the highlighting information for a document, fetches the highlighting information and the document from the database, and delivers it to the browser for viewing either by the Viewer Plugin or by a specified helper application.

> **See Also:** For more information, see Chapter 3, "Document Viewing"

Document Input/Output

The **Input/Output (I/O) Utility** can be installed on any Windows NT or 95 client and is useful for moving data between an Oracle**8** database and client-side applications. The utility also offers scripting access for launching Windows programs and text processing facilities.

> **See Also:** For more information, see Chapter 4, "Input/Output (I/O) Utility"

Sample Applications

The **CTXQUERY sample application** is an Oracle Forms application that is installed on the client-side, but can be used on any machine that has Oracle Forms runtime installed and is connected to an Oracle8 database (either directly or using Net8).

It uses demonstration tables and data which are created on the server-side using SQL scripts supplied with ConText. It illustrates a way of developing Oracle Forms 4.5 applications with ConText.

> **See Also:** For more information, see Chapter 5, "CTXQUERY Forms Sample Application"

TextServer3 Migration

The **TextServer3 Dictionary Migration Tool** is a PL/SQL package that contains procedures for migrating from TextServer3 to ConText. The scripts for creating the package are installed as part of the ConText Workbench and then run to create the package on the server-side. While the tool does not provide facilities for migrating TextServer3 applications, it does provides procedures for migrating the TextServer3 data dictionary (e.g. text tables, indexing preferences, thesauri, synonyms, and section groups).

> **See Also:** For more information, see Chapter 6, "TextServer3 Dictionary Migration Tool"

Installing the ConText Workbench

Figure 1–1
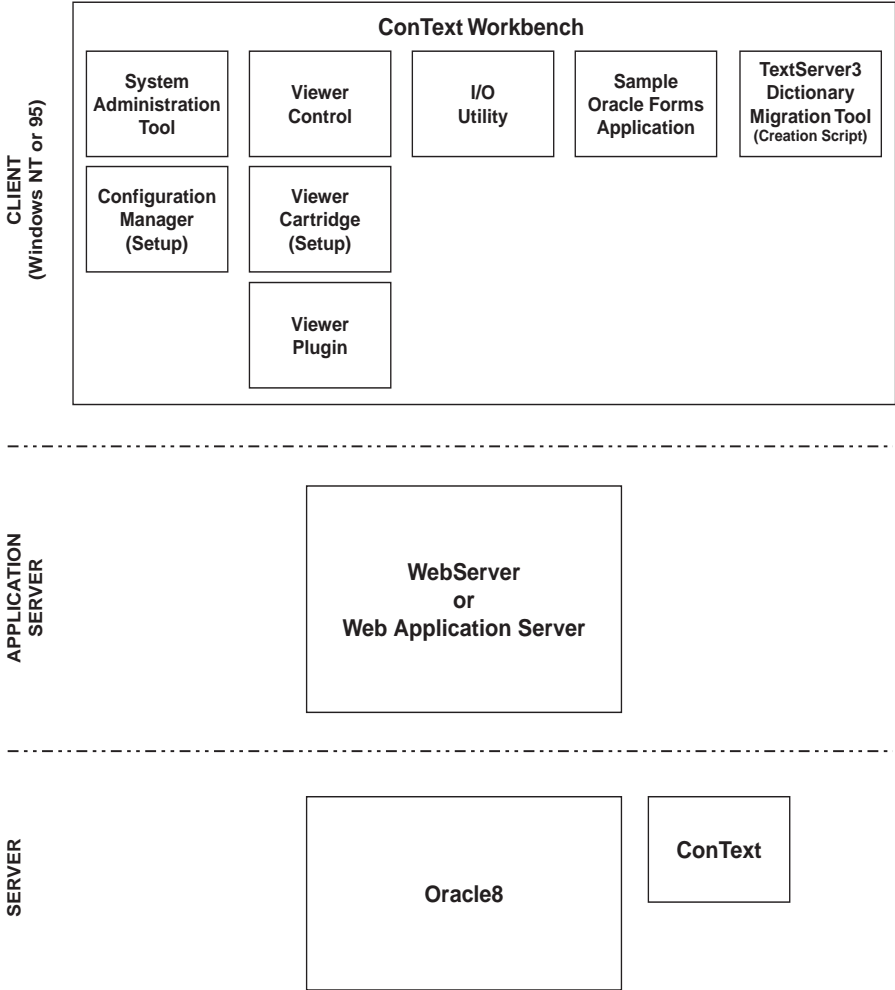
The ConText Workbench is installed using the Oracle Installer provided with the Oracle8 client. The ConText Workbench can be installed on any client machine running Windows NT or 95.

The Oracle Installer allows you to select the ConText Workbench, which automatically installs all of the ConText Workbench components, or to select individual components. Figure 1–1 illustrates the components that are installed, as well as their initial location, if you select the entire ConText Workbench.

The components you choose to install and the number of client machine(s) on which you install them are determined by how you want to use the components:

- Single Installation
- Multiple Installations

Single Installation

If you want to use the Configuration Manager and/or Viewer Cartridge, you only need to install them once, then configure them to work with Oracle WebServer or Web Application Server. Once they are configured, they can be accessed from any browser.

Similarly, if you want to use the TextServer3 Dictionary Migration Tool (a PL/SQL package), you only need to install it once, then create the PL/SQL package for the database.

> **See Also:** For more information, see "Configuring the ConText Workbench" in this chapter.

Multiple Installations

If you want to use the Viewer Control or Viewer Plugin (which relies on the Viewer Cartridge), they must be installed on *each* machine from which you want to provide document viewing.

Similarly, the System Administration tool, I/O Utility, and sample Oracle Forms application must be installed on each client from which you want to perform administration, move data, or use the sample form.

Configuring the ConText Workbench

Figure 1–2
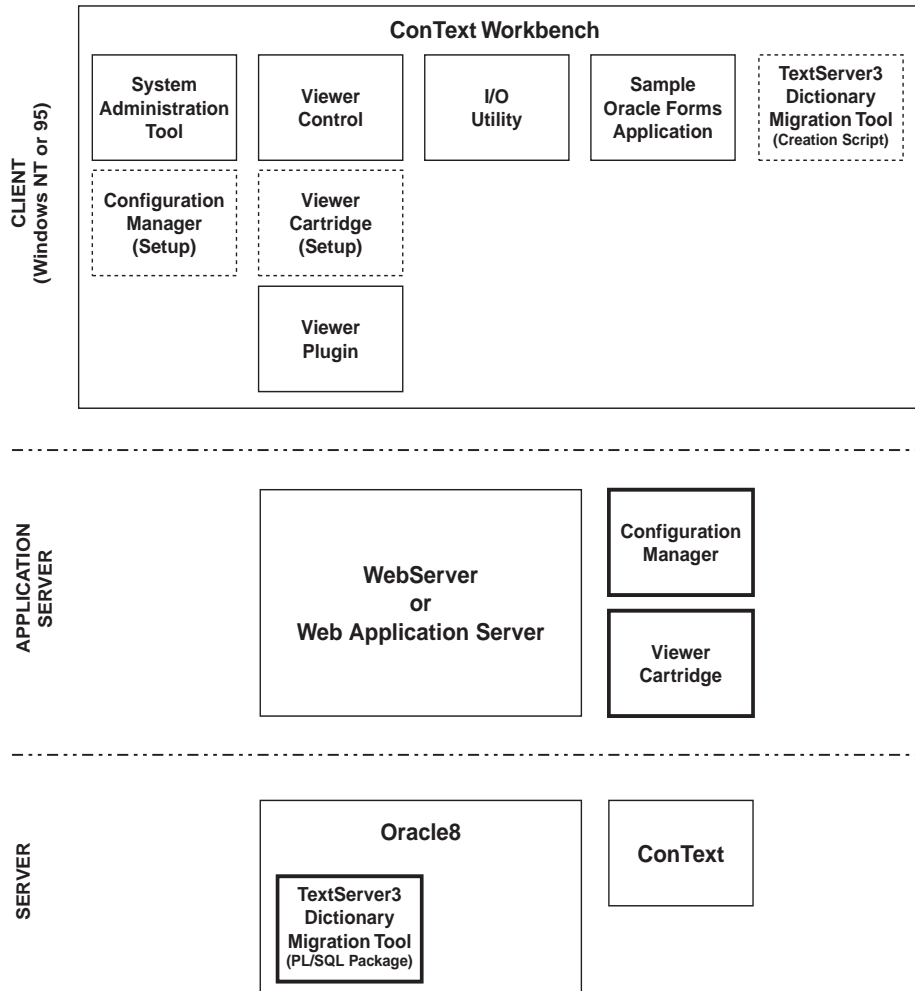
Configuration is performed after the ConText Workbench components are installed. Once installed, some components, such as the I/O utility and Viewer Control require no setup. Other components, such as the System Administration tool,

sample Oracle Form application, and Viewer Plugin require little or no setup, depending on whether the required setup tasks have already been performed.

In contrast, the Configuration Manager and Viewer Cartridge must be configured to work with Oracle WebServer or Web Application Server. Similarly, the PL/SQL package for the TextServer3 Dictionary Migration tool must be created in the Oracle8 database that contains the TextServer3 data dictionary to be migrated. Figure 1–2 illustrates the location of these components after configuration has been completed.

> **See Also:** For more information about each of these components, as well as detailed configuration instructions, see:
>
> - "Configuration Manager" in Chapter 2, "Administration Tools"
> - "Viewer Cartridge" in Chapter 3, "Document Viewing"
> - Chapter 6, "TextServer3 Dictionary Migration Tool"

2

Administration Tools

This chapter introduces and describes the administration tools provided by the ConText Workbench.

The following topics are covered in this chapter:

- System Administration Tool

- Configuration Manager

> **See Also:**   For more information about ConText administration, see *Oracle8 ConText Cartridge Administrator's Guide*

System Administration Tool

This section provides setup and usage information for the System Administration tool.

About the System Administration Tool

Figure 2–1
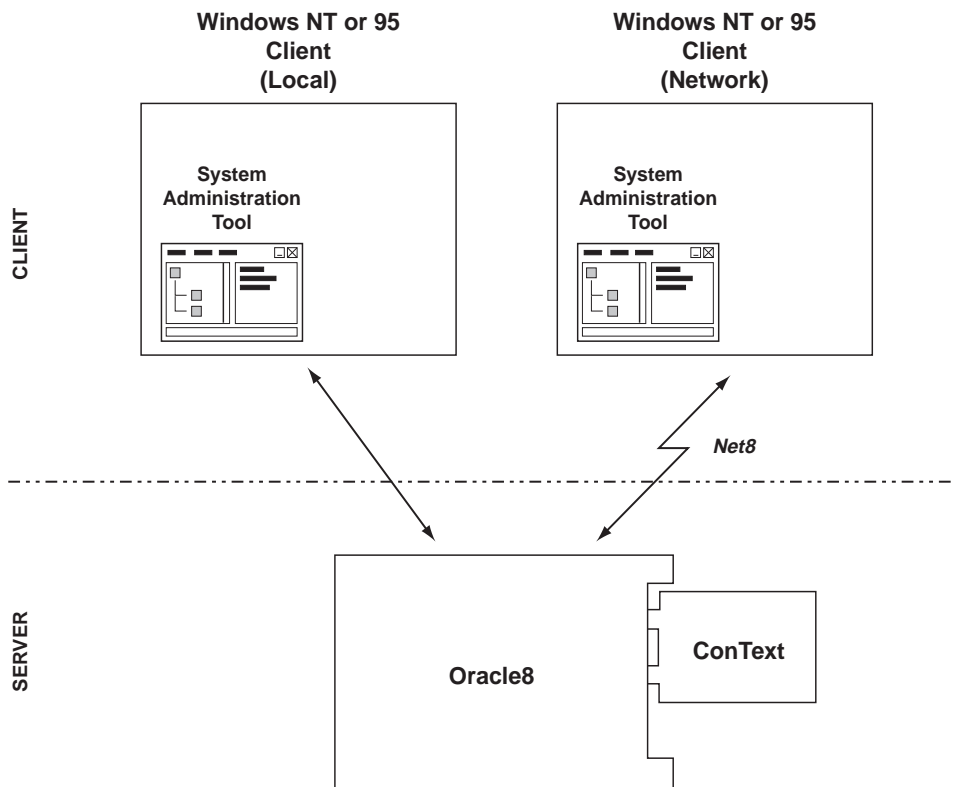
The System Administration Tool is a client-side Visual Basic application that provides a graphical user interface (GUI) for ConText administrators (i.e. users with the CTXADMIN role) to manage ConText servers, users, and queues for multiple Oracle8 databases.

It also can be used by ConText designers (i.e. users with the CTXAPP role) to perform the following tasks:

- manage indexes policies, preferences, and sections

- create setting labels for use with the ConText Linguistics.

- view and create thesauri in a graphical environment for use in text queries

- perform batch index operations, such as index creation and optimization, via a Batch Queue available only through the System Administration tool.

The System Administration Tool can be installed on any client PC running Microsoft Windows NT or Windows 95 and connects to Oracle8 using Net8.

Configuring the System Administration Tool

The following tasks may be required for using the System Administration tool.

Enable User Login

Before Oracle users with the CTXADMIN or CTXAPP role can connect to a database from the System Administration tool, the CTXSYS user may have to connect once to the database. If this is required, the System Administration tool will notify you.

> **Note:** CTXSYS needs to connect only once per database. Once CTXSYS connects to a database for the first time, the database is enabled thereafter for all users with the appropriate ConText roles.

Identify Databases

The TNSNAMES.ORA file lists all of the available databases to which an Oracle user can connect via Net8. If a user wants to connect to a database that does not appear in the list of known database in the System Administration tool, an alias for the database must be added to TNSNAMES.ORA and the System Administration tool restarted.

Using the System Administration Tool

To start the System Administration tool, simply click on the System Administration icon in the ConText Cartridge Workbench program group or choose Start > Programs > ConText Cartridge Workbench > System Administration.

Full usage information for the System Administration Tool is provided online as a WinHelp file.

Configuration Manager

This section provides set-up and usage information for the Configuration Manager.

- About the Configuration Manager
- About the Application Generator
- Configuring the Configuration Manager
- Using the Configuration Manager

About the Configuration Manager

Figure 2–2
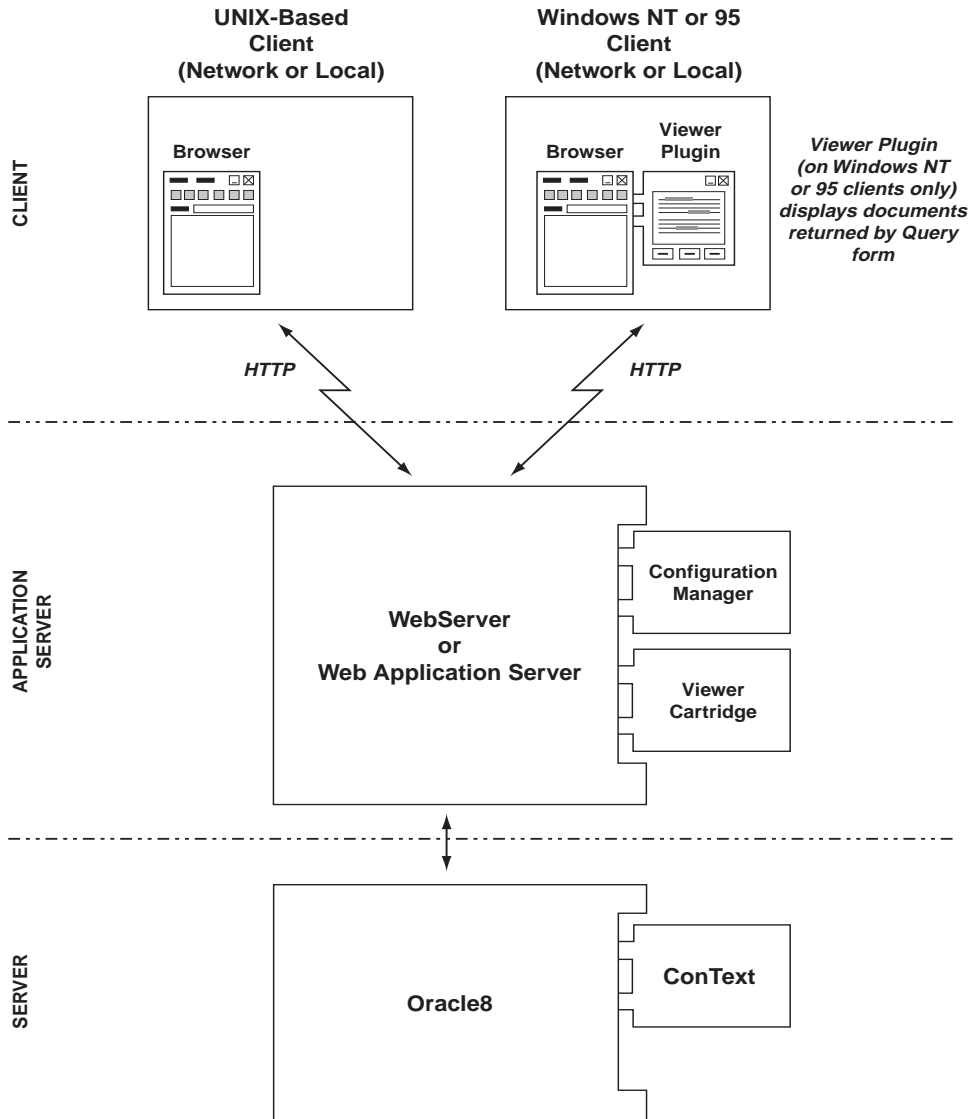
The Configuration Manager is a platform-independent, Web-based application that allows the CTXSYS user to quickly and easily manage ConText servers, users, and queues for a single database. It also provides a simple ad-hoc query interface and an application generator for quickly and easily creating prototype ConText applications

There is one installation of the Configuration Manager per database; each installation runs under the CTXSYS user.

The Configuration Manager utilizes a generic PL/SQL package for processing HTML pages extended with special tags that allow transparent interaction with a database.

> **Note:** The Configuration Manager is installed with the ConText Workbench via the Oracle Installer; however, the Installer does not automatically configure the Configuration Manager.
>
> Following the installation of the ConText Workbench, the Configuration Manager is configured by means of an icon which launches a separate Oracle Installer session.

About the Application Generator

The Configuration Manager includes an application generator that lets you build prototype ConText applications and then run them, using the querying facilities and theme and Gist viewing. You can access this tool by clicking *Applications* on the Configuration Manager Menu page.

> **See Also:** For more information about ConText administration, see *Oracle8 ConText Cartridge Administrator's Guide*

Configuration Requirements

Before you configure the Configuration Manager, ensure the following requirements are met:

- Oracle8 is installed and a database instance started

- if the Oracle8 database is installed on a different machine than the current machine (i.e. the machine from which you are performing the Configuration Manager configuration), a Net8 alias for the Oracle8 database must be created on the current machine

- Oracle8 ConText Cartridge is installed and at least one ConText server with the DDL (D) personality is started

- Oracle WebServer (or Web Application Server), release 2.1.0.3.2 or higher, is installed and at least one runtime listener and one admin listener is started

- if Oracle WebServer (or Web Application Server) is installed on a different machine than Oracle8, a Net8 alias for the Oracle8 database must be created on the WebServer machine

Configuring the Configuration Manager

Configuration of the Configuration Manager involves:

- creating database packages in the Oracle8 database (performed by Installer)

- configuring WebServer or Web Application Server and specifying database connection information (performed by Installer)

- copying required files from the current machine to the machine on which WebServer or Web Application Server is installed (performed either by you or the Installer)

> **Note:** If creation of the PL/SQL packages fails or configuration of WebServer/Web Application Server for the Configuration Manager fails, you can manually create the packages and configure WebServer/Web Application Server.
>
> In addition, deconfiguration procedures can be applied to manually remove the Configuration Manager from the appropriate release of WebServer/Web Application Server.
>
> For more information, see Appendix B, "Configuration Manager: Manual Configuration".

Local and Remote Configurations

If WebServer/Web Application Server is installed on the current machine (i.e. the machine from which Configuration Manager configuration is performed), this is considered a 'local' configuration.

If WebServer/Web Application Server is not installed on the current machine, this is considered a 'remote' configuration.

Configuration Procedure

Configuration is carried out via two Oracle Installer dialog boxes:

- *Oracle8 ConText Cartridge Configuration Manager Setup*

- *Remote Web Application Server* (appears only for remote configurations)

> **Note:** Configuration Manager configuration is identical regardless of whether Oracle WebServer or Web Application Server is installed. For convenience purposes, the following configuration procedure assumes Web Application Server is installed; however, *WebServer* can be inserted wherever *Web Application Server* appears.

To configure the ConText Configuration Manager:

1. Click on the 'Configuration Manager Setup' icon in the ConText Cartridge Workbench program group

or

Choose Start > Programs > ConText Cartridge Workbench - [*HOME_NAME*] > Configuration Manager Setup.

The Oracle Installer starts and the *Oracle8 ConText Cartridge Configuration Manager Setup* dialog box appears:

2. Enter the appropriate information in each of the fields, and click OK:

Field[1]	Description
Web Application Server	
Machine name	Enter the machine name on which Web Application Server is installed. By default, the name of the current machine appears.
	If you are performing a local configuration (i.e. Web Application Server is installed on the current machine), use the default.
	If you are performing a remote configuration, enter the name of the remote machine on which Web Application Server is installed.
Admin listener port	Enter the port number of the admin listener for Web Application Server. The default ('8888') is the default admin listener port for Oracle Web Application Server 3.0.
Admin listener user and password	Enter the username and password for the admin listener. The default username ('admin') is the default for Oracle Web Application Server 3.0.
Listener name : port	Enter the name and port of an existing runtime listener for Web Application Server. The defaults ('www' and '80') are the default runtime listener name and port for Oracle Web Application Server 3.0 on NT.
	Other runtime listener name defaults include:
	weblsn (WebServer 2.1 on Unix)
	websvr (WebServer 2.1 on Unix)
Database	
User/Password	Enter the password for CTXSYS, the owner of the Configuration Manager installation.

Field[1]	Description
Password Type	Specify the type ('database' or 'Operating system') for the CTXSYS password.
Database Connection	Select the connection type that will be used between Web Application Server and Oracle8:
	■ local - Web Application Server and Oracle8 are installed on the same machine.
	If you select this option, you must also provide the SID of the Oracle8 database.
	■ remote - Web Application Server and Oracle8 are installed on different machines
	If you select this option, you must also provide the Net8 alias (database connection) to the Oracle8 database for Web Application Server.
	Note: The alias that you enter here is *not* the alias between the current machine and Oracle8 machine.
NLS_LANG	Enter the NLS_LANG value for the Oracle8 database.

[1] Ask your web server administrator for assistance with these values.

3. If a local configuration is being performed (i.e. Web Application Server is installed on the current machine), go to Step 9. Otherwise, the *Remote Web Application Server* dialog box appears:

4. Select the platform of the remote machine on which Web Application Server is installed.

5. In a remote configuration, certain files required by the Configuration Manager must be copied from the current machine to the remote machine on which Web Application Server is installed. The files are initially located in the following directory on the current machine:

ORACLE_HOME\ctxw\middle\cfgmgr\remote

The files must be copied to the following directory on the remote machine:

ORACLE_HOME/ctxw/middle/cfgmgr

> **Note:** The 'remote' subdirectory is not required on the remote machine.

Select how you want to copy the required files to the Web Application Server machine:

- Automatically - the Oracle Installer copies the required files to the appropriate directory for you

  You can select 'Automatically' only if Web Application Server is installed on a machine that is mounted to the current machine. If you select this option, you must also specify the directory on the current machine to which the ORACLE_HOME on the Web Application Server remote machine is mounted.

- Manually - you must copy the files manually to the appropriate directory on the remote machine after configuration is completed

  ---

  Note: If the remote machine is not running Windows NT, all file and directory names *must* be in lowercase on the remote machine.

  ---

6. Enter the Net8 alias between the current machine and the machine on which Oracle8 is installed. Note that this alias is *not* the alias between the Web Application Server machine and the Oracle8 machine.

7. Click OK.

8. If you chose 'Manually' in Step 5, copy the required files from the current machine to the remote machine.

9. Shut down and restart all Web Application Server listeners and services.

Using the Configuration Manager

An HTML-based help system is delivered as part of the Configuration Manager. The help system utilizes ConText to generate a text index that enables you to search the help system; therefore, detailed usage information for the Configuration Manager is not included in this user's guide.

3

Document Viewing

This chapter introduces and describes the various document viewers provided in the ConText Workbench. In particular, this chapter provides configuration and usage information for the Viewer Plugin and Viewer Cartridge.

The following topics are covered in this chapter:

- Viewer Control
- Viewer Plugin
- Viewer Cartridge

Viewer Control

This section provides setup and usage information for the Viewer Control:

- About the Viewer Control
- Registering the Viewer Control
- Using the Viewer Control

About the Viewer Control

Figure 3–1
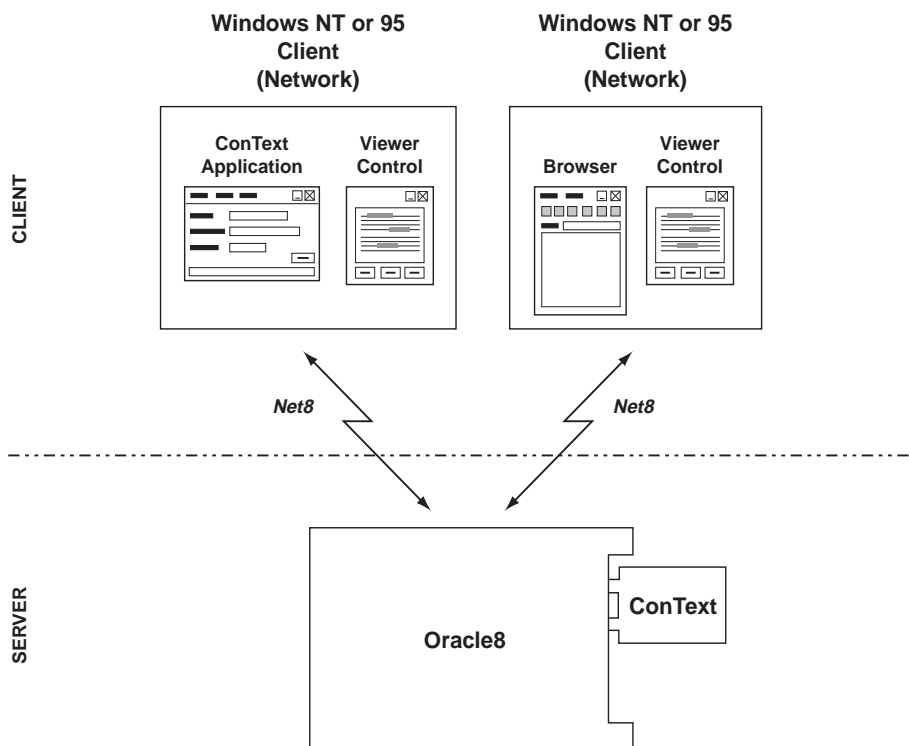
The ConText Viewer Control is a 32-bit Windows OCX custom control for use in two-tier client/server configurations. The Viewer Control uses Net8 to retrieve documents from an Oracle8 database and display the retrieved documents using the intended layout and with search terms highlighted. It is delivered with a sample application implemented as a stand-alone container.

A viewed document can be in any of the supported server-side document filter formats. The user can scroll through the document using the Next and Previous buttons to jump to subsequent/previous occurrences of the search term(s).

The Viewer Control currently supports WYSIWYG viewing and query term highlighting in the following formats:

- WordPerfect for Windows 5.x, 6.x

- WordPerfect for DOS 5.0, 5.1, 6.0

- Microsoft Word for Windows 2, 6.x

All other formats supported by ConText are displayed, by default, as plain (ASCII) text; that is, highlighting is retained, but WYSIWYG formatting is lost.

> **See Also:** For further information about document viewing, see *Oracle8 ConText Cartridge Application Developer's Guide*

Registering the Viewer Control

The Viewer Control is normally registered automatically during installation. If, for some reason, the Viewer Control has not been registered, the following message will appear when you try to use the ConText Cartridge Workbench Viewer Example:

```
Can't load (or register) custom control: 'CTXV245.OCX'
```

In this case, register it manually using the regsvr32 utility (located in *ORACLE_HOME\b*in). For example:

```
regsvr32 ctxv245.ocx
```

Note that the viewer is not unregistered during deinstallation. You must manually unregsiter it using the regsvr32 utility before deinstallation. For example:

```
regsvr32 /u ctxv32.ocx
```

Using the Viewer Control

The Viewer Control can be embedded in Windows applications or in HTML pages for display in Internet Explorer using the Object tag.

> **Note:** Full usage information for the Viewer Control is provided online as a WinHelp file.

Windows Applications

The Viewer Control can be embedded in any application environment that supports custom controls:

- as a client-side ActiveX control component in a Windows application, to view a document directly from the viewer's own database connection

- as a client/server component in an HTML page as an ActiveX control to view a document directly from the viewer's own database connection

- in a helper application to view documents that may be provided by means other than via the viewer's own database connection (see "URL to Invoke a Helper Application" in this chapter)

HTML Pages

To use the Viewer Control in an HTML environment, the browser must support ActiveX controls (e.g. Internet Explorer). In addition, the Viewer Control must be installed, along with the Required Support Files (RSFs), on the machine from which the browser is running.

Viewer Plugin

This section provides setup and usage information for the Viewer Plugin.

- About the Viewer Plugin
- Installing the Viewer Plugin
- Activating the Viewer Plugin
- When Is the Viewer Plugin Invoked?

About the Viewer Plugin

Figure 3–2

Windows NT or 95
Client
(Network or Local)

CLIENT

Browser **Viewer Plugin**

HTTP request

Highlighted document with mime type (displayed in Viewer Plugin)

APPLICATION SERVER

WebServer or Web Application Server

Viewer Cartridge

JAVA

Highlight call *Highlighted document*

SERVER

Oracle8
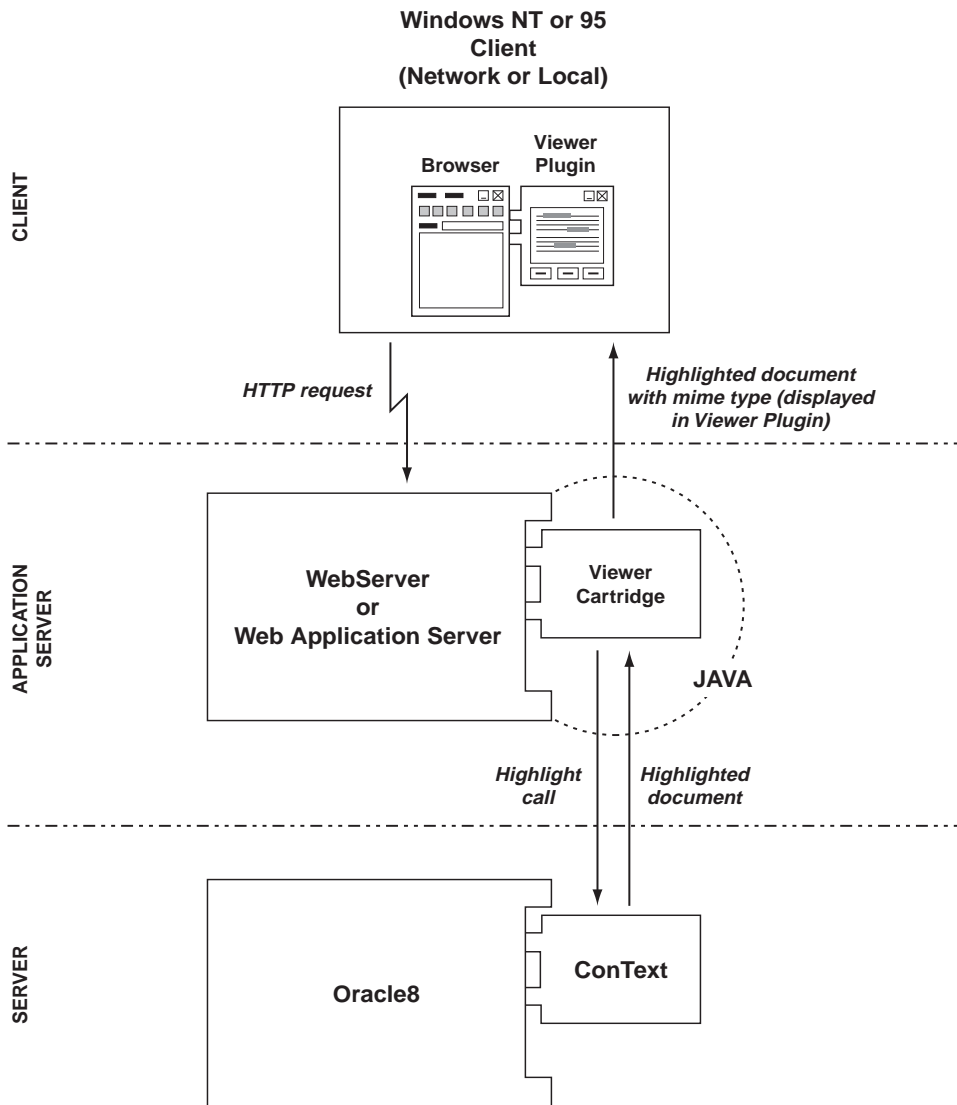
ConText

The ConText Viewer Plugin is a client-side Netscape plugin (provided by ConText) that can display many documents on the World Wide Web just as they would appear in their native format.

> **Note:** To use the Viewer Plugin in a Netscape browser, the Viewer Plugin must be installed on each machine on which the browser is installed.

Supported Formats

The Viewer Plugin actually consists of a set of libraries that provide the means for viewing the document. The plugin currently supports WYSIWYG viewing and query term highlighting in the following formats:

- WordPerfect for Windows 5.x, 6.x

- WordPerfect for DOS 5.0, 5.1, 6.0

- Microsoft Word for Windows 2, 6.x

All other formats supported by ConText are displayed, by default, as plain (ASCII) text; that is, highlighting is retained, but WYSIWYG formatting is lost.

> **See Also:** For further information about document viewing, see
> *Oracle8 ConText Cartridge Application Developer's Guide*

Viewer Cartridge

The Viewer Plugin requires the Viewer Cartridge, which is configured separately on the middle tier and runs under the Oracle Web Application Server Java cartridge. The Viewer Cartridge generates the highlight information for a document, fetches the document from the ConText database, and then sends the document as a file (MIME type *"application/x-ctxv"*) to the Web browser, which uses the Viewer Plugin (if available) to display the document.

However, for MIME types other than the ConText-specific *"application/x-ctxv"*, the Web browser can use its own mapping to invoke a helper application or plugin which can display the ConText-indexed documents natively. In such cases, suitable URLs should be specified for the Viewer Cartridge, thus removing the need for the Viewer Plugin.

> **Note:** For more information, see "Viewer Cartridge" in this chapter.

Installing the Viewer Plugin

The Viewer Plugin is installed automatically when the ConText Workbench is installed; however, you do not have to install the entire Workbench to install the Viewer Plugin. The Installer allows you to select the Viewer Cartridge as a sub-component of the ConText Workbench and install it individually.

Viewer Plugin Files

When you use the Installer to install the Viewer Plugin, the Viewer Plugin file - called *npctxvpi.dll* - is automatically placed in the Web browser *plugins* directory. The required support dlls for the Viewer Plugin are placed in the *<ORACLE_ HOME>\bin* directory.

If no Web browser is installed:

- install the Web browser
- copy the Viewer Plugin file (*npctxvpi.dll*) manually from the *<ORACLE_ HOME>\ctxw\viewer\plugin* directory to the Web browser *plugins* directory

Manual Installation on Multiple Machines

After you install the Viewer Plugin on a machine, if you want to use other machines for viewing, you can either use the Installer to install the Viewer Plugin on each machine or you can manually copy the Viewer Plugin file to the appropriate directory on each machine.

If you choose to manually copy the Viewer Plugin file to other machines, you must also copy the required support dlls to each machine. These support files are provided in a file named *ctxvpi.zip*, which is in the *<ORACLE_ HOME>\ctxw\middle\cfgmgr* directory on the machine where the Configuration Manager was initially installed.

Copy this file to the Web browser *plugins* directory on the viewing machine(s), then unzip the files to make the Viewer Plugin available.

> **Note:** If you require full NLS support on a viewing machine(s), you must use the Installer to perform a full installation on the machine(s).

Activating the Viewer Plugin

To activate the Viewer Plugin file:

1. Close down all open Web browser sessions (if any),

2. Start a new Web browser session.

    To ensure that the file has been registered, select the **Help** -> **About Plug-ins** menu item from Netscape. The Viewer Plugin should appear in the list.

When Is the Viewer Plugin Invoked?

Once installed, the Viewer Plugin is automatically invoked by the Web browser when:

■ a document with the MIME (Multipurpose Internet Mail Extensions) type *"application/x-ctxv"* is delivered to the Web browser via the Viewer Cartridge and Oracle Web Application Server, or

■ a local file with the extension *".ctxv"* is loaded into the Web browser

> **Note:**   Before using the Viewer Plugin, the Viewer Cartridge must be installed and configured, and at least one suitable URL must be specified.
>
> For more information, see "Viewer Cartridge" in this chapter.

Viewer Cartridge

This section provides setup and usage information for the ConText Viewer Cartridge:

- About the Viewer Cartridge

- Configuration Requirements

- Configuring the Viewer Cartridge

- Specifying URLs

- Using URLs

- Security Mechanisms

- Configuration Files

- Using an Alternative Oracle JDBC Driver

About the Viewer Cartridge

The Viewer Cartridge is installed as a middle-tier component and runs under the Oracle Web Application Server Java cartridge. Its purpose is to deliver ConText-indexed documents, and other data held in an Oracle database, to Web browsers. These documents can then be viewed in the Netscape browser, either by the supplied Viewer Plugin, or by a helper application.

> **Note:** The Viewer Cartridge is installed with the ConText Workbench via the Oracle Installer; however, the Installer does not automatically configure the Viewer Cartridge.
>
> Following the installation of the ConText Workbench, the Viewer Cartridge is configured by means of an icon which launches a separate Oracle Installer session.

Uniform Resource Locators (URLs)

URLs are used to tell the Viewer Cartridge (how) to view ConText-indexed documents that come from an Oracle database. When these URLs are specified, a MIME type determines how the documents will be delivered. With the default ConText MIME type, *"application/x-ctxv"*, the Web browser automatically tries to use the Viewer Plugin, if available.

URLs can be used to display documents in various ways. Documents can be viewed full-frame, in a separate window, or to predetermined dimensions.

> **See Also:**   For more information about URLs in the Viewer Cartridge, see "Specifying URLs" and "Using URLs" in this chapter.

Realms and Configuration Files

Access to ConText-indexed documents can be restricted through the Oracle Web Application Server by the use of realms.

A Viewer Cartridge configuration file is used to determine how the Viewer Cartridge connects to the database. Multiple configuration files can be used to allow different database connection details to be protected by different realms. This allows varying levels of security to be applied to realms containing particular documents.

> **See Also:**   For more information about using realms, see "Security Mechanisms" in this chapter.
>
> For more information about configuration files, see "Configuration Files" in this chapter.

Configuration Requirements

Before you configure the Viewer Cartridge, ensure the following requirements are met:

- Oracle8 is installed and a database instance started

- if the Oracle8 database is installed on a different machine than the current machine (i.e. the machine from which you are performing the Viewer Cartridge configuration), a Net8 alias for the Oracle8 database must be created on the current machine

- Oracle8 ConText Cartridge is installed and at least one ConText server with the DDL (D) personality is started

- Oracle WebServer (or Web Application Server), release 2.1.0.3.2 or higher, is installed and at least one runtime listener and one admin listener is started

- if Oracle WebServer (or Web Application Server) is installed on a different machine than Oracle8, a Net8 alias for the Oracle8 database must be created on the WebServer machine

■ Oracle JDBC driver, release 7.3.3.1.3 or higher, is installed

If you need to install an Oracle JDBC driver, note the connection requirements during installation. These are used when writing additional configuration file entries.

To obtain a JDBC driver, either navigate to *www.oracle.com*, look for the free software page, and choose the Oracle JDBC option. Or navigate directly to the page; currently it is:

```
www.oracle.com/products/free_software/
```

Otherwise, you can use an alternative driver.

> **See Also:** For more information about configuration files, see "Configuration Files" in this chapter.
>
> For more information about alternative JDBC drivers, see "Using an Alternative Oracle JDBC Driver" in this chapter.

Configuring the Viewer Cartridge

Configuration of the Viewer Cartridge involves:

■ configuring Web Application Server and specifying database connection information (performed by the Installer)

■ copying required files from the current machine to the machine on which Web Application Server is installed (performed by you or the Installer)

> **Note:** If configuration of WebServer/Web Application Server for the Viewer Cartridge fails, you can manually configure WebServer/Web Application Server.
>
> In addition, deconfiguration procedures can be applied to manually remove the Viewer Cartridge from the appropriate release of WebServer/Web Application Server.
>
> For more information, see Appendix A, "Viewer Cartridge: Manual Configuration".

Local and Remote Configurations

If Web Application Server is installed on the current machine (i.e. the machine from which Viewer Cartridge configuration is performed), this is considered a 'local' configuration.

If Web Application Server is not installed on the current machine, this is considered a 'remote' configuration.

Configuration Procedure

Configuration is carried out via two Oracle Installer dialog boxes:

- *Oracle8 ConText Cartridge Viewer Cartridge Setup*

- *Remote Web Application Server* (appears only for remote configurations)

> **Note:** Viewer Cartridge configuration is identical regardless of whether Oracle WebServer or Web Application Server is installed. For convenience purposes, the following configuration procedure assumes Web Application Server is installed; however, *WebServer* can be inserted wherever *Web Application Server* appears.

To configure the ConText Viewer Cartridge:

1. Click on the 'Viewer Cartridge Setup' icon in the ConText Cartridge Workbench program group

   or

   Choose Start > Programs > ConText Cartridge Workbench - [*HOME_NAME*] > Viewer Cartridge Setup.

The Oracle Installer starts and the *Oracle8 ConText Cartridge Viewer Cartridge Setup* dialog box appears:

2. Enter the appropriate information in each of the fields, and click OK:

Field[1]	Description
Web Application Server	
Machine name	Enter the machine name on which Web Application Server is installed. By default, the name of the current machine appears.
	If you are performing a local configuration (i.e. Web Application Server is installed on the current machine), use the default.
	If you are performing a remote configuration, enter the name of the remote machine on which Web Application Server is installed.
Admin listener port	Enter the port number of the admin listener for Web Application Server. The default ('8888') is the default admin listener port for Oracle Web Application Server 3.0.
Admin listener user and password	Enter the username and password for the admin listener. The default username ('admin') is the default for Oracle Web Application Server 3.0.

Field[1]	Description
Listener name	Enter the name of an existing runtime listener for Web Application Server. The default ('www') is the default runtime listener name for Oracle Web Application Server 3.0 on NT.
	Other runtime listener name defaults include:
	weblsn (WebServer 2.1 on Unix)
	websvr (WebServer 2.1 on Unix)
Database	
User/Password	Enter the username and password for the owner of the Viewer Cartridge. The default username is CTXSYS.
Connection alias	Enter the Net8 alias (database connection) to the Oracle8 database for Web Application Server.
	There can be several database connections for the Viewer Cartridge. This field allows you to enter an alias for the default database connection. The username/password and connection alias are combined to define the default connection for access by the URL for the Viewer Cartridge.
	The alias that you enter here is used as the first entry in the list of connections in the configuration file. After configuration of the Viewer Cartridge, you can add further entries to this configuration file if you wish (see "Configuration Files" in this chapter).
	Note: The alias that you enter here is *not* the alias between the current machine and Oracle8 machine.

[1] Ask your web server administrator for assistance with these values.

3. If a local configuration is being performed (i.e. Web Application Server is installed on the current machine), go to Step 9. Otherwise, the *Remote Web Application Server* dialog box appears:
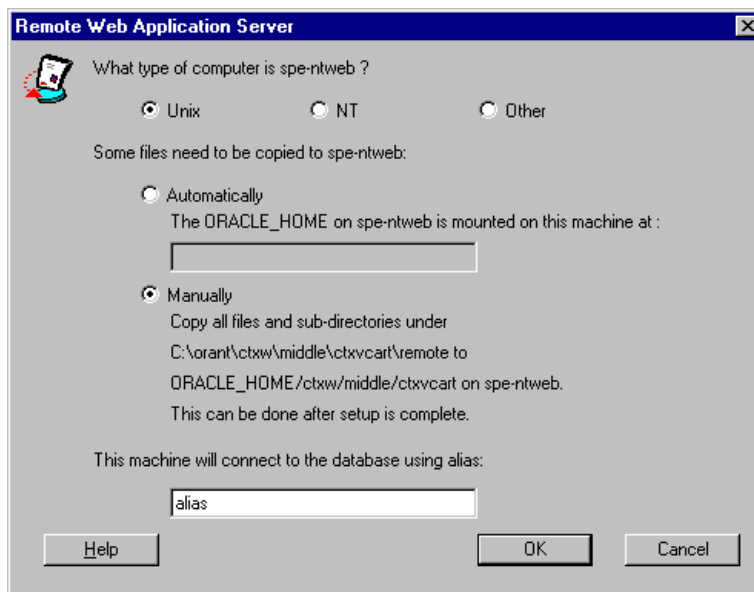
4. Select the platform of the remote machine on which Web Application Server is installed.

5. In a remote configuration, certain files required by the Viewer Cartridge must be copied to the remote machine on which Web Application Server is installed. The files are initially located in the following directory:

ORACLE_HOME\ctxw\middle\ctxvcart\remote

The files must be copied to the following directory structure on the remote machine:

ORACLE_HOME/ctxw/middle/ctxvcart

Note: The 'remote' subdirectory is not required on the remote machine.

Select how you want to copy the required files to the Web Application Server machine:

- Automatically - the Oracle Installer copies the required files to the appropriate directory for you

  You can select 'Automatically' only if Web Application Server is installed on a machine that is mounted to the current machine. If you select this option, you must also specify the directory on the current machine to which the ORACLE_HOME on the Web Application Server remote machine is mounted.

- Manually - you must copy the files manually to the appropriate directory on the remote machine after configuration is completed

  Note: If the remote machine is not running Windows NT, all file and directory names *must* be in lowercase on the remote machine.

6. Enter the Net8 alias between the current machine and the machine on which Oracle8 is installed. Note that this alias is *not* the alias between the Web Application Server machine and the Oracle8 machine.

7. Click OK.

8. If you chose 'Manually' in Step 5, copy the required files from the current machine to the remote machine.

9. Shut down and restart all Web Application Server listeners and services.

Specifying URLs

To be able to use the Viewer Cartridge for viewing ConText-indexed documents from an Oracle database, you need to specify URLs and embed them into HTML pages.

These URLs must contain sufficient information for the Web Request Broker to direct the request to the Viewer Cartridge. The Viewer Cartridge can then generate the following to service the request: a database connection, parameters that define what the document is, plus any highlight information that is necessary.

Parameters to the URL can be encoded within the URL (via the HTML GET attribute), or supplied through HTML form fields (via the HTML POST attribute).

URLs for Viewing Highlighted Documents

To get a representation of a highlighted document (in WYSIWYG, if supported, or in ASCII, if WYSIWYG is not supported), use the URL parameters as follows (no spaces allowed):

```
http://<host>/<virtual_path>/CtxwViewCart?colspec=<policy>&textkey=<textkey>[&conf=<config_
section_name>]&queryex=<query>
```

where:

<host>
is the Internet name of the host running the Oracle Web Application Server

<virtual_path>
specifies the Viewer Cartridge alias (virtual path). *ctxwview* is the installed virtual path that maps onto the Viewer Cartridge default realm (*ctxview_realm*). Note that multiple configuration files can be created to allow different database connection details to be protected by different realms. In that case, the appropriate virtual path to the Viewer Cartridge must be specified. See "Security Mechanisms" and "Configuration Files" in this chapter.

CtxwViewCart
is the name of the Java class that provides the entry point to the Viewer Cartridge; always *CtxwViewCart*

<policy>
specifies the ConText indexing policy to use

<textkey>
specifies the textkey column value identifying the document to be retrieved

<config_section_name>
optionally identifies a section name from the appropriate configuration file

<query>
optionally specifies a query expression

URLs for Viewing Unhighlighted Documents

To get a WYSIWYG display of an unhighlighted document, the correct MIME type must be mapped to the appropriate application in the Netscape web browser. The browser can then use this mapping to invoke a helper application or plugin to handle the data. The syntax is (no spaces allowed):

```
http://<host>/<virtual_
path>/CtxwViewCart?colspec=<policy>&textkey=<textkey>[&mimetype=<mimetype>][&conf=<conf_section_name>]
```

where:

<host>
is the Internet name of the host running the Oracle Web Application Server

<virtual_path>
specifies the Viewer Cartridge alias (virtual path). *ctxwview* is the installed virtual path that maps onto the Viewer Cartridge default realm (*ctxview_realm*). Note that multiple configuration files can be created to allow different database connection details to be protected by different realms. In that case, the appropriate virtual path to the Viewer Cartridge must be specified (See "Security Mechanisms" and "Configuration Files" in this chapter).

CtxwViewCart
is the name of the Java class that provides the entry point to the Viewer Cartridge; this is always *CtxwViewCart*

<policy>
specifies the ConText indexing policy to use

<textkey>
specifies the textkey column value identifying the document to be retrieved

<mimetype>
specifies the appropriate MIME type that has been set up in the web browser to enable the document to be viewed in its original, unhighlighted form. If *mimetype* is omitted, the default MIME type "application/x-ctxv" is assumed, and the browser tries to view the document via the Viewer Plugin, if available. If the document is in a non-supported format, the plugin cannot display it in WYSIWYG.

<conf_section_name>
optionally identifies a section name from the appropriate configuration file

Example Here, the MIME type for Word has been mapped in the web browser (as *"application/msword"*). A URL is then set up to retrieve and display an original Word document. No spaces are allowed:

```
http://<host>/ctxwview/CtxwViewCart?colspec=ctxcm_help_pol&textkey=HELP_
POL&mimetype=application/msword&conf=DEFAULT
```

URL for Viewing Original Documents

If the document in its original, unhighlighted form is required (without using ConText), and is stored internally in the database - that is, *not* URL or external data store - it is more efficient to go directly to the base table.

The following URL example displays a document from the table field defined by *<table_name>*, *<column_name>* and *<where_clause>* as these elements would appear in a SQL select statement. No spaces are allowed:

```
http://<host>/<virtual_path>/CtxwViewCart?table=<table_name>&column=<column_name>&where=<where_
clause>[&mimetype=<mimetype>][&conf=<config_section_name>]
```

where:

table=<table_name>
table specification

column=<column_name>
column to fetch

where=<where_clause>
row selection

[mimetype=<mime_type_string>]
specifies an appropriate MIME type that has been set up in the web browser to enable the document to be viewed in its original, unhighlighted form. If *mimetype* is omitted, the MIME type defaults to *"application/octet-stream"* for binary data and *"text/plain"* for textual data.

[conf=<config_section_name>]
optionally identifies a section name from the appropriate configuration file

Using URLs

Once suitable URLs have been specified, they can be used in HTML in various ways:

- URL as a Link Target
- URL in an Embedded Plugin Tag
- URL to Invoke a Helper Application

URL as a Link Target

When you use URLs as a link target, highlighted documents are viewed full-frame if the Viewer Plugin has been installed on the client. The frame is resizable, but plugin parameters cannot be specified.

Documents of other MIME types (such as jpeg files) are also supported. These are downloaded and viewed on the client by the best means available to the browser for that MIME type.

URL in an Embedded Plugin Tag

When you use a URL in an embedded plugin tag, you can specify EMBED tag parameters that can be passed to the plugin. These parameters are included within an EMBED tag in the appropriate HTML page and they determine how a document is viewed. The parameters fall into two categories:

- Standard HTML Parameters
- Private Viewer Plugin Parameters

> **Note:** Only documents of the MIME type *"application/x-ctxv"* are supported.

Standard HTML Parameters

For details of the many available standard HTML parameters, refer to your preferred HTML reference source. The most important of the standard HTML parameters are *src*, which specifies the URL string, and *pluginspage*, which for our purposes is used to provide the location of the Viewer Plugin archive.

The dimensions of the plugin display are pre-determined. However, the height and width of the document view can be reset by the *height* and *width* tag parameters.

Typical syntax within the EMBED tag would be:

```
src="http://<host>/<virtual_
path>/CtxwViewCart?colspec=<policy>&queryex=<query>&textkey=<tkey>[&conf=<sectio
n>]"
[conf=<section>]
height="nnn" width="nnn"
pluginspage="/ctxw_cm/ctxvpi.zip"
[<optional_private_params>]>
```

where:

src
specifies the string of standard parameters (as described in "Using URLs" in this chapter). The *src* attribute value (everything between the double-quotes) must not contain any spaces.

[conf=<section>]
Optional. *conf* is used to indicate the name of the section in the configuration file that is to be used for database connection data. This parameter defaults to use the *[DEFAULT]* section.

nnn
is a positive number, in pixels, that determines the height and width of the document view

pluginspage
states where the Viewer Plugin was placed by default when the Configuration Manager was installed; used to provide the location of the plugin archive for an 'install-on-demand' mechanism

<optional_private_params>
is where the private Viewer Plugin parameters are specified (See "Private Viewer Plugin Parameters" in this chapter).

Example: The following example from the Configuration Manager (View Data form) uses the Viewer Plugin to display a document from the table field implied by the policy and textkey, and uses the query to highlight hit words:

```
<table border="2" cellpadding="0" cellspacing="0">
<tr>
<td align="center"><EMBED
src="/ctxwview/CtxwViewCart?colSpec=CTXSYS.CTXCM_HELP_POL&queryEx=(++{policy})+
&textKey=HELP_POL&conf=bugdb"
align="baseline" border="0" width="556" height="330"
pluginspage="/ctxw_cm/ctxvpi.zip"><NOEMBED>[OCO Viewer Plugin]</NOEMBED></td>
</tr>
</table>
```

The *src* attribute value (everything between the double-quotes) must not contain any spaces.

The value *bugdb* refers to a section contained in the sample configuration file documented in this chapter.

> **See Also:** For more information, see "Configuration Files" in this chapter.

Private Viewer Plugin Parameters

You can specify optional private Viewer Plugin parameters to be included within the EMBED tag on the HTML page. Like the standard HTML parameters, these influence various aspects of how the document is viewed. These are:

```
toolbar=true|false
silenterrors=true|false
idleinterval=<n>
shortcutmenu=true|false
pageview=true|false
```

where:

toolbar
If *toolbar* is set to true, the toolbar is shown. The default is true.

silenterrors
If *silenterrors* is set to true, it suppresses error information dialog boxes. The default is false.

idleinterval

represents a value (in milliseconds) between 1 and 500 that sets the interval between document section reads. The default is 50. A value of 0 means that there is no idle time reading of document sections.

shortcutmenu

If *shortcutmenu* is set to true, you can click the right mouse button on the viewer window to obtain a short-cut menu. The default is true.

pageview

If *pageview* is set to true, the plugin will attempt to represent the document page as the page would appear when printed. The default is false.

Example:

```
<p>
<embed width=600 height=400 pluginspage="http://fred.uk.oracle.com/"
src="file:///C|/WINNT/Profiles/jbloggs/Desktop/test files/test.ctxv"
toolbar=true silenterrors=true idleinterval=5 shortcutmenu=false pageview=true
hidden=false> </embed>
</p>
```

Notice the use of a local file in the *src* parameter.

URL to Invoke a Helper Application

URLs can be used to invoke a helper application to display the document in a separate window as a separate process. However, the mapping between the helper application's MIME type and the program to be executed must have been defined in the Netscape browser. Only then can the helper application be used to display the document.

For example, the URL can be used as a link URL where a MIME type is mapped to the ConText Cartridge Viewer Example that uses the Viewer Control. The ConText Cartridge Viewer Example is then used as a helper application. The Viewer Example is delivered as part of the ConText Workbench, and is accessible via the installed icon called 'ConText Cartridge Viewer Example'.

Security Mechanisms

A realm is a group of users and (other) groups assigned by an authentication scheme to regulate access to specific documents and directories through the Oracle Web Application Server. Access to ConText-indexed documents can be restricted by using realms.

Authentication schemes allow you to define named groups of user name/password combinations, and named realms that are groups of these groups. You can then assign user, group, and realm names to virtual files and directories, requiring any client requesting access to input one of the specified username/password combinations.

This way, some groups can be granted access to ConText-indexed documents that are held in a variety of databases, while access can be denied to other groups.

To restrict access through realms:

1. Map a virtual path to a realm through the Oracle Web Application Server Administration pages.

2. Map that virtual path to the Viewer Cartridge through the Oracle Web Application Server Administration pages.

3. Create the configuration files to which the virtual paths are mapped.

   If the virtual path contains more than one element, the last element is used to map onto the name of the configuration file. In this manner, each realm can have a separate configuration file. This ensures that users in groups with access to one realm can be denied access to information available in other realms.

4. Enter the connection details into the configuration file (using the *conf* URL parameter) for each database connection that is within that realm.

5. Specify URLs using the appropriate virtual paths.

   See Also: For more information about using the Oracle Web Application Server Administration pages, see Appendix A, "Viewer Cartridge: Manual Configuration", and also the Oracle Web Application Server documentation.

   For more information about the configuration file, see "Configuration Files" in this chapter.

   For more information about specifying URLs, see "Specifying URLs" in this chapter.

Configuration Files

The following section describes Viewer Cartridge configuration files. The Viewer Cartridge uses configuration files to determine how the cartridge connects to a database. Each configuration file contains one or more sections that describe the database connection, including the username/password, database details, and JDBC driver to be used.

Default Configuration File

The default configuration file (CTXVCART.CFG) is installed when the Viewer Cartridge is configured. It is installed on the same tier as the Web Application Server, in the following directory:

`$ORACLE_HOME/ctxw/middle/ctxvcart`

You can edit this default file if you wish; however, if you need to allow different database connection details to be protected by different realms, you must create additional configuration files.

Multiple Configuration Files

Multiple configuration files can be created to allow different database connection details to be protected by different realms. These files can be used instead of, or in conjunction with, the default file CTXVCART.CFG.

All configuration files must reside on the same tier as the Web Application Server, in the following directory:

`$ORACLE_HOME/ctxw/middle/ctxvcart`

You create multiple configuration files as text files, giving each file the same name as the last element of the applicable virtual path of the Viewer Cartridge, plus a mandatory *.cfg* extension.

For example:

Virtual Path	Configuration File
/ctxview	ctxvcart.cfg (the default)
/ctxview/secure	secure.cfg
/ctxview/public	public.cfg

Configuration File Syntax

Each configuration file consists of one or more sections, each section representing a different database connection with its details.

Each section must start with a section name enclosed within square brackets and contains a number of parameters associated with that section.

A section name is specified in a URL by means of the *conf* parameter (See "Using URLs" in this chapter).

Lines that contain non-standard parameter names are ignored. If one or more parameters of the same name are found within one section, the value associated with last one is used. A value cannot be split over more than one line.

The section syntax is as follows:

```
[<section_name>]
username=<user_name>
password=<password>
database=<database_specification>
driver=<Java_class>
subprotocol=<subprotocol>
```

where:

[<section_name>]
names the section. The name is enclosed by square brackets. Duplicate section names are not allowed in the same configuration file. The default configuration file has the section name *[DEFAULT]*.

Each section ends when the next section name is met, or when the end of the file is reached.

> **See Also:** For information on how the *conf* parameter is used to point to a section name in a configuration file when using a URL, see "Specifying URLs" in this chapter.

username
specifies the name of the Oracle user that the database connection uses. Defaults to *CTXSYS*.

password
specifies the password of the Oracle user. Cannot be null.

database
depends on the value of the *subprotocol* parameter.

For any subprotocol, a full connect descriptor, as used in *tnsnames.ora* files, can be used. For example:

```
database=(DESCRIPTION=(ADDRESS=(COMMUNITY=DECCOM.FIN.HQ.ACME)(PROTOCOL=DECNET)(NODE=NY_
VAX.FIN.HQ.ACME)(OBJECT=LSNR))(CONNECT_DATA=(SID=DB1)(GLOBAL_NAME=NY_FIN.FIN.HQ.ACME)))
```

For the oci7 or oci8 subprotocol, a service name present in a *tnsnames.ora* file can be used.

For the thin protocol, a value in the form *<host>:<port>:<sid>* can be used. For example:

```
hq_server:1521:ORCL
```

If *database* is null, it defaults to the installation's default database.

driver
specifies the Java class name of the JDBC driver (of release 7.3.3.1.3 or later). It should always be:

```
oracle.jdbc.driver.OracleDrive
```

subprotocol
should be assigned one of the following values:

```
oci7 | oci8 | thin
```

If the *subprotocol* parameter name does not appear within the section, a default of oci7 is used.

> **Note:** For more information about the values you can specify for the *database*, *driver*, and *subprotocol* parameters, see the documentation for the JDBC driver that you are using. As this information is liable to change, you must look in the JDBC driver readme.txt file and modify the configuration file as appropriate.

Sample Configuration File

This sample file contains three sections: *DEFAULT*, *bugdb*, and *live*. Each section denotes a different database connection.

```
[DEFAULT]
username=ctxwvcart
password=ctxwvcart
database=(DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (COMMUNITY = tcp.uk)
(PROTOCOL = TCP) (Host = bris_nt_001) (Port = 1521)) (ADDRESS = (COMMUNITY =
tcp.world) (PROTOCOL = TCP) (Host = bris_nt_001) (Port = 1526))) (CONNECT_DATA
= (SID = ORCL)))
subprotocol=oci7
driver=oracle.jdbc.driver.OracleDriver
Note_1=  To connect to a non-default database :-
Note_2=    1. Add a new section with the new database details (ensure to use a
unique section name)
Note_3=    2. Connect to the database as ctxsys and type 'grant select on
ctxw_message to ctxuser'

[bugdb]
username=bu1203
password=hello
database=hg_bug:1527:orcl
subprotocol=thin

[live]
username=web_user
password=we8_user
database=live_quotes
subprotocol=oci8
driver=oracle.jdbc.driver.OracleDriver
```

Using an Alternative Oracle JDBC Driver

It is possible that your driver is not suitable for the platform you are using. To use an alternative Oracle JDBC driver:

1. Install the alternative driver into the Oracle WebServer's *<ORACLE_HOME>*.

2. Modify the Viewer Cartridge CLASSPATH to include a path to (the zip archive containing) the class file root directory.

3. Modify the Viewer Cartridge LD_LIBRARY_PATH to include a path to the directory containing any native library object files required by the driver.

> **See Also:** For more information about modifying the
> CLASSPATH and LD_LIBRARY_PATH, see Appendix A, "Viewer
> Cartridge: Manual Configuration".

4. Modify the configuration file that contains the section referred to by the *conf*
   parameter in the URL by tailoring the following configuration file parameters:

```
database=<instance_specification>
driver=<fully_qualified_driver_class_name>
subprotocol=<oci7|oci8|thin>
```

> **See Also:** For more information about the values you can specify
> for the *database*, *driver*, and *subprotocol* parameters, see the
> documentation for the JDBC driver that you are using. Also refer to
> "Configuration File Syntax" in this chapter.

4

Input/Output (I/O) Utility

This chapter describes how to use the Oracle8 ConText Cartridge Input/Output (I/O) utility. You can use this utility in Windows 32-bit environments such as Windows NT and Windows 95.

The following topics are covered in this chapter:

- About the I/O Utility
- Command-Line Syntax
- Parameter File Structure
- Examples
- The I/O Utility dll Interface

About the I/O Utility

The I/O Utility enables Windows applications to load and unload data between database tables and client-side files. It is a 32-bit, stand-alone executable that performs a series of operations carried out in sequence as defined by a parameter file.

Example parameter files are documented in this user's guide; in addition, one of the example parameter files is included in the ConText Workbench as a sample.

The I/O utility can be used to perform the following operations:

- write data from a (possibly RAW) database field to an operating system file

- read data from an operating system file and update a (possibly RAW) database field

- execute any piece of PL/SQL or SQL that can be executed in an anonymous PL/SQL block

- spawn an operating system command

- load comma-delimited data into the database

- load directory hierarchies into the database

- filter any ASCII text file before loading into the database

These operations are performed in a sequence defined by a parameter file that you specify on the command line.

Command-Line Syntax

The I/O utility has the following command-line syntax:

```
ctxio32 [-s] [-w] [-p file] [-l file] [-v editor] [-d connect_string]
```

where:

Parameter	Description
-s	Shows login box.
	This allows the user to confirm the connect string and the name of the parameter file.
-w	Displays status window.
	Messages in the parameter file are displayed in this window during processing.
-p file	Specifies the name of parameter file.
	For more information about the format of the file, see "Parameter File Structure" in this chapter.
-l file	Specifies the name of log file.
	Error and debug messages, as well as any messages in the parameter file, are output to this file. No messages are reported directly to the user.
-v editor	Displays log file, using given editor (a Windows executable, such as Notepad).
-d connect_string	Specifies an explicit database username/password connection string.
	If "-d none" is given, there is no log on and no prompt so no database session is started. This option would normally be used to spawn an operating system command.
	If -d is omitted, the user is prompted to enter the connection string.

Parameter File Structure

The general structure of a parameter file is similar to the tag structure used in HTML/SGML:

<START>
many lines of text
<END>
<START>
many lines of text
<END>

Start Tag	End Tag	Text	Description
<SQL>	</SQL>	Limited to a maximum of 32k of SQL split over many lines. Keep each line below 2k.	Encloses SQL commands.
		If the first character is @, the SQL is read from the file that follows; for example, @oco.sql loads SQL from the file oco.sql, and executes it.	
		SQL read from a file must not exceed 5Mb.	
<GET>	</GET>	Line 1: Destination filename	Writes a document from a database table to an operating system file.
		Line 2: Table containing source document	
		Line 3: Column containing source document	
		Line 4: WHERE clause (excluding WHERE keyword)	
<PUT>	</PUT>	Line 1: Source filename	Puts an operating system file into a database table.
		Line 2: Destination table name	
		Line 3: Destination column name	
		Line 4: WHERE clause (excluding WHERE keyword)	

Start Tag	End Tag	Text	Description
<REG>	</REG>	Reg_key, Reg_value [, Trunc_string]	Registration entries can be used within <EXE></EXE> pairs.
			Reg_key is the registration key, as it appears in the lefthand pane of the Registry Editor window.
			Reg_value is the registry entry name as it appears in the righthand pane of the Registry Editor window. This defaults to "(Default)".
			Trunc_string is optional and causes truncation of the fetched registry value from the given string. Many useful registry entries have '%1' or similar placeholders. See the parameter file examples further on.

Start Tag	End Tag	Text	Description
<LOAD>	</LOAD>	Line 1: Table_name [TAB_OPTIONS] Line2: Column_name[COL_ OPTIONS],Column_name[COL_ OPTIONS]... Line 3+: Column_data, column_data, ... **Note:** The backslash represents the escape character. Therefore, to obtain \, use \\. Or to obtain ", use \\'.	Line 1: TAB_ OPTIONS := (Replace) *Replace*: Data is updated rather than inserted into the database. This option is used if the row that satisfies the 'where' clause already exists. Line 2: COL_OPTIONS := (COL_OPTION, COL_ OPTION...) COL_ OPTION := ReadFromFile | Replace "X" "Y" ["C"] | Ignore "X" "Y" ["Z"] ["C"] | ExtractOne "X" "Y" ["Z"] ["C"] | ExtractAll "X" "Y" ["Z"] ["C"] | SaveDir "X" | TempFile "X" | Where | Directory "X" | UseFilter "X" | Sequence n | LoadDir "A" "B" "C" "D" "E" "F" | GeneratedFile *ReadFromFile* Reads the contents of the given file into the database column. *Replace* "X" "Y" ["C"] After reading the file replaces all occurrences of X with Y. There can be more than one Replace call per column. X and Y are case-sensitive if C is omitted or if C equals "N". X and Y are case-insensitive if C equals "Y". *Ignore "X" "Y" ["Z"] ["C"]* After reading the file removes all text between X and Y inclusive of X and Y. If Z is given, remove the text only if Z occurs between X and Y. X, Y and Z are case-sensitive if C is omitted or if C equals "N". X and Y are case-insensitive if C equals "Y".

Start Tag	End Tag	Text	Description
			ExtractOne "X" "Y" ["Z"] ["C"]
			ExtractAll "X" "Y" ["Z"] ["C"]
			After reading the file, extracts all text between X and Y, excluding X and Y. If Z is given, extracts the text only if Z occurs between X and Y. *ExtractOne* extracts only the first instance of the text; *ExtractAll* extracts all instances with a carriage return/line feed between instances. X, Y and Z are case-sensitive if C is omitted or if C equals "N". X and Y are case-insensitive if C equals "Y".
			SaveDir "X"
			This parameter copies the post-filtered versions of the input files into directory X. This is useful for checking that search/replace processing has worked as expected.
			TempFile "X"
			Any temporary files are created by default in the home directory; X can be used to specify another file name that is used instead. Typically, this is not required.
			Where
			This defines the columns to be used to define the where clause. This is used to insert the file data into the database and to perform the updates during *Replace* mode.
			Directory "X"
			The file names given refer to files in this directory. X must end in a double backslash ("\\").
			UseFilter "X"
			For each file, this spawns a user-defined filter program to pre-filter the files before any Extract, Replace or Ignore actions. The program must take an input file as the first argument, and an output file as the second argument. X is the program name (including the full path).

Start Tag	End Tag	Text	Description
			Sequence n
			This generates a number of the value of this column. This value is initially n (default = 1).
			LoadDir "A" "B" "C" "D" "E" "F"
			This recursively scans files below directory A (which must have a trailing slash) and looks for files of type B (containing wildcards and separated by ';').
			Use C as the directory separator. Prefix all files with D. Recursion ceases when more than E directory levels is reached (a value of 0 means no limit is set). Do not recurse into directories F (where multiple directories are separated by ';').
			See the LoadDir Example in this chapter.
			GeneratedFile
			This is related to LoadDir. The generated file from LoadDir is used as input for this column.
			Line 3: Column_data = Strings, dates and so on should be quoted as if they were entered in an INSERT SQL statement.
			Note:
			Only ASCII files can be filtered with Replace, Ignore, ExtractOne, ExtractAll.
<EXE>	</EXE>	OS command.	Operating system command. This can be split over many lines. If so, carriage returns are removed and replaced by spaces.
<QU>	</QU>	Line 1: Question Line 2: Yes_label Line 3: No_label	This prompts the user with the given question and resumes execution after the line starting *'Yes_label:'* if the user responds 'yes'. If the user responds 'no', execution is continued after the line starting *'No_label:'*.
			Alternatively, labels may take the form *'#<string>'*
<GOTO>	label	None.	Resumes execution after the line starting *'label:'*

Start Tag	End Tag	Text	Description
<MSG>	end of line	Message	Displays *message* in log file and, if -w option is chosen, displays in status window.
			Note:
			<MESS> is still supported as an alternative to <MSG>.
<ERROR CODE>	end of line	Error number	Sets the error code to the given number.
			If a command fails, and the current error code is non-zero, then that error code is returned by ctxio32.exe. If the -r command line option is used in this case, the error code is also placed in the registry at the following location:
			"HKEY_LOCAL_ MACHINE\SOFTWARE\ORACLE\ConText Workbench\ctxio_ret"
<AUTOC OMMIT>	end of line	On | Off	Sets the standard autocommit function. If On, there is a commit after every SQL statement until either it is turned Off, or the end of the session.
#	end of line	None	Parameter file comment - not used.

Examples

This section provides the following four examples for using the I/O utility:

- LoadDir Example
- GeneratedFile Example
- Simple Parameter File Example
- Advanced Parameter File Example

LoadDir Example

```
"y:\\webpages\\" "*.htm *.html" "/" "http://oracle.uk.oracle.com" 20 ""
```

This example loads all .htm and .html files into and below y:\webpages (up to a maximum of 20 levels). The path from y:\webpages\ is then built by using '/' as the separator, and this relative path is then prefixed by 'http://oracle.uk.oracle.com/'.

So, if y:\webpages contains a.htm, subdir\b.html and subdir\c.htm, then the following three strings are loaded into the database:

```
'http://oracle.uk.oracle.com/a.htm'
'http://oracle.uk.oracle.com/subdir/b.html'
'http://oracle.uk.oracle.com/subdir/c.htm'
```

GeneratedFile Example

```
Head(GeneratedFile,ExtractOne "<TITLE>" "</TITLE>" "" "Y")
```

This example loads the file found by LoadDir and extract the text between 'TITLE' and '/TITLE'. Case is ignored because of the "Y" parameter.

Simple Parameter File Example

The file below is a sample CTXIO32 parameter file. It is installed as part of the ConText Workbench.

The command to execute this parameter file is:

```
ctxio32 –v notepad –s –w –p pfile.txt –l log.txt –d ctxdemo/ctxdemo
```

```
#
# CTXIO sample script
# File : pfile.txt

#
# Script split into 2
#
# 1) First  section (#ReplaceAll)  replaces all  employees.
# 2) Second section (#ReplaceSome) replaces some employees.
#
# The <QU> .. </QU> section asks a question and jumps to the relevant tag
#

<QU>
Do you wish to replace ALL employees?
#ReplaceAll
#ReplaceSome
</QU>

#
# -----------   This section replaces all current employees -----------

#
#ReplaceAll


<MESS>Drop employee table
<SQL>
DROP TABLE CTXIO_EMP
</SQL>

<MESS>Create employee table
<SQL>
CREATE TABLE CTXIO_EMP (EMPNO NUMBER, EMPNAME VARCHAR2(100), RESUME LONG)
</SQL>
<SQL>
alter table ctxio_emp add constraint unique_ctxioemp unique(empno)
</SQL>

<MESS>Truncate employee table
```

```
<SQL>
truncate table CTXIO_EMP
</SQL>

<MESS>Loading ALL employees into database...
# Use SaveDir so we can check search and replace has required affect
<LOAD>
CTXIO_EMP
EMPNO(WHERE), EMPNAME, RESUME(ReadFromFile, Ignore "<Confidential>" "</Confidential>", Replace "brewing"
"chemistry", Replace "Brewing" "Chemistry", SaveDir "temp")
1,"Joe Bloggs","jbloggs.htm"
2,"Ray Smith","rsmith.htm"
3,"Lisa Turner","lturner.htm"
</LOAD>

# Skip next bit which is for single update only

<GOTO> #Common

#
# -------------   This section updates the resumes of some employees --------------
#
#ReplaceSome

<MESS>Replacing resumes in database...
# Use SaveDir so we can check search and replace has required affect
<LOAD>
CTXIO_EMP (REPLACE)
EMPNO(WHERE), EMPNAME, RESUME(ReadFromFile, Ignore "<Confidential>" "</Confidential>", Replace "brewing"
"chemistry", Replace "Brewing" "Chemistry", SaveDir "temp")
2,"Ray Smith","rsmith.htm"
</LOAD>

# Load this resume without search and replace
<PUT>
jbloggs.htm
ctxio_emp
resume
EMPNO=1
</PUT>

#
# ----------- This section common to both -----------

#
#Common

# Check Smith's CV was filtered and stored OK
<GET>
temp\rsmith.htm
```

```
ctxio_emp
resume
EMPNO=2
</GET>


#
# ----------- Create OCO index -----------
#
<MESS>Dropping policy and index (may produce errors but this is OK)...
<SQL>
begin
ctxsys.ctx_ddl.drop_index('CTXIO_EMPRES');
end;
</SQL>
<SQL>
begin
ctxsys.ctx_ddl.drop_policy('CTXIO_EMPRES');
end;
</SQL>

<MESS>Creating policy on test table and creating index...
<SQL>
begin
ctxsys.ctx_ddl.create_policy(policy_name=>'CTXIO_EMPRES',
colspec=>'CTXIO_EMP.RESUME',
textkey=>'EMPNO');
end;
</SQL>
<SQL>
begin
ctxsys.ctx_ddl.create_index('CTXIO_EMPRES');
end;
</SQL>


#
# ----------- Do some OCO querying -----------
#

<MESS>Create marked up table
<SQL>
drop table mutab
</SQL>
<SQL>
create table mutab (id number, document long)
</SQL>

<MESS>Get highlights...
<SQL>
```

```
begin
  ctxsys.ctx_query.highlight(CSPEC=>'CTXIO_EMPRES', TEXTKEY=>2,
   query=>'chemistry,Chemistry', mutab=>'MUTAB', ID=>999,starttag=>'<EM>',
   endtag=>'</EM>');
end;
</SQL>

<MESS>Write marked up text to file...
<GET>
mudoc.htm
mutab
document
id=999
</GET>

# This views the marked-up resume in notepad
#<EXE>
#notepad mudoc.htm
#</EXE>

# This views the HTML resume in Netscape
<EXE>
<REG>HKEY_CLASSES_ROOT\NetscapeMarkup\shell\open\command,,"</REG> file:///mudoc.htm
</EXE>

#Finished
```

Advanced Parameter File Example

This example demonstrates how to load a website. In particular, it shows how to use the LoadDir option of the I/O utility.

> **Note:** This example is not included as part of the ConText Workbench distribution

```
<MESS>Drop table
<SQL>
DROP TABLE WEBPAGE
</SQL>
<SQL>
DROP TABLE WEBPAGE_OPTIONS
</SQL>
<MESS>Create tables..
<SQL>
CREATE TABLE WEBPAGE(DOCNO NUMBER, DOC LONG, HEAD VARCHAR2(1000))
</SQL>
<SQL>
CREATE TABLE WEBPAGE_OPTIONS(NAME VARCHAR2(100), VALUE LONG, DES VARCHAR2(1000))
</SQL>
<SQL>
CREATE TABLE WEBPAGE_HL(ID NUMBER, DOCUMENT LONG)
</SQL>
<MESS>Create sequence
<SQL>
CREATE SEQUENCE HL_SEQ
</SQL>
<SQL>
alter table WEBPAGE add constraint unique_WEBPAGE unique(docno)
</SQL>
<MESS>Truncate table
<SQL>
truncate table WEBPAGE
</SQL>
<AUTOCOMMIT> On
<MESS>Scanning and loading roll ...
<LOAD>
WEBPAGE
Docno(Where, Sequence 1), Doc(LoadDir "j:\\www\\ocowsite\\" "*.htm;*.html" "/" "http://roll.uk.oracle.com/"
20 "www"), Head(GeneratedFile, ExtractOne "<TITLE>" "</TITLE>" "" "Y")
</LOAD>
<MESS>Scanning and loading shake ...
<LOAD>
WEBPAGE
Docno(Where, Sequence 1000), Doc(LoadDir "y:\\webpages\\" "*.htm;*.html" "/" "http://shake.uk.oracle.com/"
```

```
20), Head(GeneratedFile, ExtractOne "<TITLE>" "</TITLE>" "" "Y")
</LOAD>
<MESS>Scanning and loading uksn16 ...
<LOAD>
WEBPAGE
Docno(Where, Sequence 2000), Doc(LoadDir "h:\\etc\\httpd\\htdocs\\" "*.htm;*.html" "/"
"http://uksn16.uk.oracle.com/" 20), Head(GeneratedFile, ExtractOne "<TITLE>" "</TITLE>" "" "Y")
</LOAD>
<MESS>Dropping policy and index (may produce errors but this is OK)...
<SQL>
begin
ctxsys.ctx_ddl.drop_index('WEBPAGE');
end;
</SQL>
<SQL>
begin
ctxsys.ctx_ddl.drop_policy('WEBPAGE');
end;
</SQL>
<MESS>Creating policy on test table and creating index...
<SQL>
begin
ctxsys.ctx_ddl.create_policy(policy_name=>'WEBPAGE',
colspec=>'WEBPAGE.DOC',
filter_pref => 'CTXSYS.HTML_FILTER',
dstore_pref => 'CTXSYS.DEFAULT_URL',
textkey=>'DOCNO');
end;
</SQL>
<SQL>
begin
ctxsys.ctx_ddl.create_index('WEBPAGE');
end;
</SQL>
<SQL>
TRUNCATE TABLE WEBPAGE_OPTIONS
</SQL>
<MESS>Initializing search page options ...
<LOAD>
WEBPAGE_OPTIONS
NAME, VALUE, DES
'GenShortTitle','OCO Websearch','Short generic title'
'FrontTitle','Oracle8 ConText Cartridge Web Site Search','Title of front page'
'FrontHeading','<center><h1>Oracle8 ConText Cartridge Web Site Search</h1><center>','Heading of front page'
'FrontIncludeCount','Y','Include web pages count (Y/N)'
'FrontContentsPreQuery','','Contents of page after count and before query section (max 32k)'
'FrontContentsPostQuery','','Contents of page after query section (max 32k)'
'ResultTitle','Oracle8 ConText Cartridge Search Results','Title of results page'
'ResultHeading','<center><h1>Oracle8 ConText Cartridge Search Results</h1></center>','Heading of results
page'
```

```
'ResultShowExpression','Y','Show query expression (Y/N)'
'ResultLabelHeading','Page heading','Label to describe heading'
'ResultLabelScore','Score','Label to describe score'
'ResultLabelPage','Page','Label to describe page'
'ResultHighStart','<Font color=red size=+4>','Highlight on'
'ResultHighEnd','</font>','Highlight off'
'ResultTagStart','<OCOH>','Internal use'
'ResultTagEnd','</OCOH>','Internal use'
'ResultContentsPreTable','','Contents of page before the results table (max 32k)'
'ResultContentsPostTable','','Contents of page after the results table (max 32k)'
'ResultHighNextYN','Y','Show the [Next] link after each highlight (Y/N)'
'ResultHighNext','<font size=-1>[Next]</font>','If ResultHighNextYN=Y, this is  displayed after each
highlight'
'ResultHighFirst','[First Highlight]','If ResultHighNextYN=Y, this is  displayed at start and end of
document'
</LOAD>
<PUT>
frontpre.txt
webpage_options
value
name='FrontContentsPreQuery'
</PUT>
<PUT>
frontpos.txt
webpage_options
value
name='FrontContentsPostQuery'
</PUT>
<PUT>
respre.txt
webpage_options
value
name='ResultContentsPreTable'
</PUT>
<PUT>
respos.txt
webpage_options
value
name='ResultContentsPostTable'
</PUT>
```

The I/O Utility dll Interface

All the functionality of ctxio is incorporated in a single dll (ctxio32l.dll). Developers can call the C routines within this dll directly.

> **See Also:**   For more information, see the required header file, ctxexp.h, which is installed in $ORACLE_HOME/ctxw/io/public as part of the ConText Workbench distribution.

5

CTXQUERY Forms Sample Application

This chapter presents the CTXQUERY sample application, developed with Oracle Forms 4.5, and illustrates one method for developing applications using ConText.

The following topics are covered in this chapter:

- About CTXQUERY
- Setting Up the Sample
- CTXQUERY Architecture

About CTXQUERY

The CTXQUERY sample application (*ctcquery.fmb*) is a basic Oracle Forms 4.5 application which provides a graphical interface for performing text queries and DML operations with ConText. The application exists to show one way of developing Forms applications using ConText. It supports all the query expression operators and can provide query term highlighting using asterisks in the specified documents.

> **Note:** An icon is not created for CTXQUERY during installation of the ConText Workbench because the sample application requires some setup tasks to be performed on the server before the application can be used.
>
> For more information, see "Setting Up the Sample" in this chapter.

> **See Also:** For more information about developing ConText applications, see *Oracle8 ConText Cartridge Application Developer's Guide*

Concepts

The ConText query concepts demonstrated in this example are:

- query expression syntax
- boolean and statistical (logical) operators
- expansion operators
- score thresholds
- result limits
- two-step queries
- sharing result tables
- term highlighting

> **Note:** The CTXQUERY sample Forms application does *not* utilize
> any of the theme querying functionality provided by ConText.
> However, the application could be theme-query enabled for
> English-language documents by creating a theme indexing policy,
> indexing the policy, and referencing the policy in the query block.

Layout

The form is laid out in four blocks on three canvases. There is a block and a canvas
for each step of the query, hitlist, and view with one extra block for all the buttons.
All canvases are displayed in a single window, CTXQUERY.

Setting Up the Sample

This section provides information about the setup tasks that must be performed before using CTXQUERY:

- Demonstration Installation
- Oracle Forms Installation

Demonstration Installation

Before the sample Forms application can be used, the demonstration data must be imported into the required table and demonstration installation must be performed. The demonstration data is provided in a dump file (*demo.dmp*). Installation is performed by a set of SQL scripts (*demoinst.sql* and *genling.sql*).

Importing the *demo.dmp* file performs the following tasks:

- creates the *articles* sample table
- populates the *articles* table with text

Running the SQL scripts performs the following tasks:

- creates a text index for the text column in the table (*demoinst.sql*)
- creates additional tables used for highlighting and Linguistic output (*demoinst.sql*)
- creates Gists and theme summaries for each document and stores them in the output tables (*genling.sql*)

Note: These files are *not* included in the ConText Workbench installation; they are provided in the ConText installation (server-side) and must be imported/run manually after ConText is installed.

In addition, the location of these files on the server-side is operating system dependent. For more information about the directory structure for ConText, see the Oracle8 documentation specific to your operating system.

See Also: For more information about setting up CTXQUERY, see *Oracle8 ConText Cartridge Application Developer's Guide*

Oracle Forms Installation

The sample application requires Oracle Forms (32-bit), version 4.5.7.4.0 or higher, to be installed.

Oracle Forms is not installed automatically during ConText Workbench installation and must be installed separately before using the sample application.

> **Note:** If a version higher than 4.5.7.4.0 is installed on your machine, to use the sample application, the *ctxquery.fmb* file may have to be regenerated after installation.

CTXQUERY Architecture

The CTXQUERY form is divided into eight distinct sections:

- Startup
- Enter the query screen
- Start the pre-query trigger
- Build the query expression
- Display an article in the view block
- Invoke the HIGHLIGHT procedure
- Perform DML Operations
- Cleanup

Startup

On startup, the When-New-Form-Instance trigger initializes the form.

```
When-New-Form-Instance
set_window_property
  (FORMS_MDI_WINDOW, WINDOW_STATE, MAXIMIZE);
set_window_property('CTXQUERY', WINDOW_STATE, MAXIMIZE);
select query_id.nextval
  into :global.query_id
  from dual;
```

This maximizes the windows for display, then selects a query ID from a sequence.

This form is designed so that it can be used by multiple concurrent users, which means that the results tables are shared. When result tables are shared, a unique query ID separates each user's results from other users. In this form the unique ID is created as an increasing sequence of numbers with the last value stored in a global variable.

Enter the query screen

The query screen is the first screen of the form displayed.

Each widget is part of the *query* block, which is a non-base-table block. There is one hidden field in this screen, *query_string*.

When the Query button is pressed, the When-Button-Pressed trigger goes to the *hitlist* block and executes a query:

```
When-Button-Pressed on BUTTONS.QUERY

go_block('hitlist');
clear_block(no_validate);
execute_query(all_records);
```

The *hitlist* block is based on the *article_hitlist* view. This view joins the query result table *query_temp* and the base table *articles* into a hitlist which has both score and article information. Although this block contains items for all the fields in this view, most are hidden. The visible page displays score, author, section, and title.

Start the pre-query trigger

Before the query on the hitlist block is executed, the *pre-query* trigger fires. This trigger executes the first step of a two-step query, then limits the relational fields to the query criteria:

```
Pre-Query on HITLIST

build_query_string;

ctx_query.contains('DEMO_POLICY',
                   :query.query_string,
                   'QUERY_TEMP',
                   1,
                   :global.query_id,
                   0,1,null);
--
--limit the relational fields to the query criteria
--

:hitlist.conid := :global.query_id;
:hitlist.author := :query.author;
:hitlist.section := :query.section;
copy(:query.pub_date,'hitlist.pub_date');
```

Build the query expression

The *build_query_string* procedure constructs the query expression by concatenating the term, weight, and operator fields from the form and putting them into the *query.query_string* hidden field.

build_query_string is divided into sections A through E.

Section A

Section A of the code clears the query string of any previous contents.

Section B

Section B takes each term from the user input form and builds the query expression as follows:

- attaches expansion operators (*fuzzy, stem, soundex*) to the front of each specified query term

- encases user-specified query arguments in curly brackets

    Query arguments are enclosed in curly brackets so that any special characters within the argument (e.g., the "&" in Q&A) will not be misinterpreted as ConText operators

- adds weighting factor

    If the user has filled in a weighting factor in the appropriate field of the input screen, this value is appended to the search argument.

- appends any specified logical operators to the end of the query expression

    When the concatenation is finished, a sample term looks like this:

Expansion	Search Criteria	Weight	Boolean
!	{cat}	*2	&

> **Note:** In CTXQUERY, the user can enter up to three separate query expression. These four steps are repeated to for each query expression.

Section C

Section C of the code strips off extraneous operators at the end of the final query string since the last term has no boolean operator.

Section D

Section D adds the score threshold value to the query expression.

Section E

Section E appends the result limit.

BUILD_QUERY_STRING Procedure - Sample Code

```
PROCEDURE build_query_string IS
BEGIN
--
--Section A
--
    :query.query_string := null;
--
--Section B
--
    if (:query.qterm1 is not null) then
      :query.query_string := :query.query_string ||
        :query.qexp1 ||
        '{' || :query.qterm1 || '}' ||
        '*' || to_char(nvl(:query.qwt1, 1)) ||
        substr(:query.qop1,2,1);
    end if;
if (:query.qterm2 is not null) then
    :query.query_string := :query.query_string ||
      :query.qexp2 ||
      '{' || :query.qterm2 || '}' ||
      '*' || to_char(nvl(:query.qwt2, 1)) ||
      substr(:query.qop2,2,1);
end if;
if (:query.qterm3 is not null) then
    :query.query_string := :query.query_string ||
      :query.qexp3 ||
      '{' || :query.qterm3 || '}' ||
      '*' || to_char(nvl(:query.qwt3, 1));
end if;
--
--Section C
```

```
--
:query.query_string :=
rtrim(:query.query_string, '&|,;-');
--
--Section D
--
    if (:query.qthresh is not null) then
        :query.query_string := '(' || :query.query_string ||
        ')>' || to_char(:query.qthresh);
    end if;
--
--Section E
--
    if (:query.qlimit is not null) then
        :query.query_string := '(' || :query.query_string ||
        '):' || to_char(:query.qlimit);
    end if;
END;
```

Display an article in the view block

When an article in the hitlist is double-clicked, the When-Mouse-Double-Click trigger on *hitlist.title* displays the article in the VIEW block:

```
When-Mouse-Double-Click on HITLIST.TITLE
GO_BLOCK('VIEW');
EXECUTE_QUERY;
```

In order to display the correct article, the Pre-Query trigger limits the *view* block to display the article highlighted in the *hitlist* block:

```
Pre-Query on VIEW:
:view.article_id := :hitlist.article_id;
```

The *view* block is based on *articles*, and displays the full text of the article.

Invoke the HIGHLIGHT procedure

When the highlight button on the *view* block is pressed, the When-Button-Pressed trigger on *buttons.highlight* invokes the CTX_QUERY.HIGHLIGHT procedure.

The *highlight* procedure is divided into sections A and B.

Section A

Section A call the HIGHLIGHT procedure and performs the following tasks:

- fetches the document

- parses the query

- identifies the matching terms

- inserts results into the *highlight_temp* table.

The highlight result table (*highlight_temp*) has the following structure:

Columns	TYPE
ID	NUMBER
OFFSET	NUMBER
LENGTH	NUMBER
STRENGTH	NUMBER

This procedure highlights matching query terms in the text window by surrounding them with <<< and >>>.

It is not possible to do an INSTR for the search terms because of the expansion operators; for example, *go=going=gone* in a stem expansion. Instead, use the highlight procedure in Section A to generate the highlights table.

This table holds the offset and the length of each word to be highlighted.

Section B

With the highlights generated, the cursor loop in Section B works through each offset, length pair backwards from last offset to first.

The asterisks insert won't change the other offsets. For each offset, length pair, the asterisks are inserted by reassigning the full text to: everything before the term (line 1), then the asterisks, then the term and some more asterisks (line 2) then everything after the term (line 3). Doing this repeatedly highlights all the terms in the document.

HIGHLIGHT Procedure -- Sample Code

```
PROCEDURE highlight IS
  cursor highcur is select offset, length
    from highlight_temp
    where id = :global.query_id
    order by offset desc;
--
--  Section A
--
BEGIN
  ctx_query.highlight(
    'DEMO_POLICY',              -- policy name
    to_char(:view.article_id),  -- textkey
    :query.query_string,        -- query string
    :global.query_id,           -- query_id
    null,
    null,
    'HIGHLIGHT_TEMP',           -- highlight table
    null,
    null,
    null,
    null);
--
--  Section B
--
  for hc in highcur loop
1.    :view.text := substr(:view.text,1,hc.offset - 1) ||
2.    '***'||substr(:view.text,hc.offset,hc.length)||'***'||
3.    substr(:view.text,hc.offset+hc.length);
  end loop;
END;
```

Perform DML Operations

When a document is changed it needs to be re-indexed and new linguistic information needs to be extracted. This can be performed in a table trigger or in the application.

The *view* block contains pre-update and pre-index triggers that perform the necessary re-indexing when there is a change to article data. Both triggers invoke the *reindex_article* procedure.

The *reindex_article* procedure performs the following tasks:

- submits the document for text re-index (Section A)

- clears existing linguistic information (Section B)

- extracts new linguistic information (Section C)

- submits the linguistic requests (Section D)

> **Note:** There is no COMMIT; it is performed by the COMMIT of the change to the VIEW block. In SUBMIT, you must explicitly ask for it not to COMMIT (the FALSE second parameter).

> **See Also:** For more information about DML and re-indexing, see *Oracle8 ConText Cartridge Administrator's Guide.*

REINDEX_ARTICLE Procedure -- Sample Code

```
PROCEDURE reindex_article IS
  handle number;
begin
#
# Section A
#
ctx_dml.reindex('demo_policy', TO_CHAR(:view.article_id));
#
# Section B
#
delete from article_themes
where pk = :view.article_id;

delete from article_gists
where pk = :view.article_id;
#
# Section C
#
ctx_ling.REQUEST_themes('demo_policy',
        TO_CHAR(:view.article_id),
           'article_themes');
ctx_ling.REQUEST_gist('demo_policy',
        TO_CHAR(:view.article_id),
           'article_gists');
#
```

```
Section D
#
  handle := ctx_ling.submit(0,FALSE,0);
end;
```

Cleanup

Before the session is ended, clean up any leftover rows in the hitlist result table (*query_temp*) and the highlight result table (*highlight_temp*) tables for the query ID:

```
Post-Form
delete from query_temp
where conid = to_number(:global.query_id);
delete from highlight_temp
where id = to_number(:global.query_id);
standard.commit;
```

The *standard.commit* statement executes an Oracle COMMIT without doing a Forms *commit_form*.

6

TextServer3 Dictionary Migration Tool

This chapter describes how to use the TextServer3 Dictionary Migration Tool.

The following topics are covered in this chapter:

- About the TextServer3 Dictionary Migration Tool
- Overview of the TextServer3 Migration Process
- Requirements and Limitations
- Installing the TextServer3 Dictionary Migration Tool
- Text Table Migration
- Thesaurus Migration
- Section Group Migration
- CTXWMG_MIGRATE: TextServer3/ConText Migration Tool Package
- Example Migration Using Supplied Script

About the TextServer3 Dictionary Migration Tool

The TextServer3 Dictionary Migration tool has been developed to assist users who are migrating from TextServer3 to ConText. It is not a complete migration tool, but it does allow you to:

- create a policy in ConText for each indexable column of a TextServer3 table

- migrate TextServer3 thesauri and synonym rings to ConText

- create a section group in ConText for each section group in TextServer3

The tool is delivered as a PL/SQL package called *CTXWMG_MIGRATE*. The package generates SQL scripts that you can customize to your requirements before running them.

This package must be created in the ConText database on the server after the ConText Workbench is installed on a client machine.

What is the Migration Tool?

The Migration Tool is a PL/SQL package, *CTXWMG_MIGRATE*, that is installed on the same database as ConText. It queries the TextServer3 dictionary to find as much useful information as possible that can be used in the creation of column policies. This is done using a database link to the database where the TextServer3 text dictionary resides. It also uses the database link to access any TextServer3 thesauri or section groups that are to be migrated.

The PL/SQL package that implements the tool makes use of the UTL_FILE package. This package adds File I/O capabilities to PL/SQL in Oracle8, release 8.0.

Who Performs a TextServer3 Migration

The Migration Tool is intended for use by application developers or TextServer3/ConText administrators. It is assumed that the user has an understanding of the following:

- how different settings for a TextServer3 text table affect indexing, filtering and querying

- the content and structure of any TextServer3 thesauri, synonym rings, and section groups to be migrated, and how they are used in the TextServer3 application

- ConText concepts such as policies, preferences, and attributes

Familiarity with the Oracle8 IMP/EXP utility is also assumed.

All references to migration refer to the TextServer3/ConText data dictionary migration process. All references to the Migration Tool refer to the TextServer3/ConText Dictionary Migration Tool.

> **See Also:** For more information about policies, preferences, attributes, and thesauri in ConText, see the *Oracle8 ConText Cartridge Administrator's Guide*

Overview of the TextServer3 Migration Process

Although TextServer3 and ConText share similar functionality, there are differences that prevent the automatic migration of text tables, thesauri and section groups. The user or application developer needs to make decisions about preferences and policies using their knowledge about the text application.

For this reason, the Migration Tool does not directly create column policies and thesauri in the ConText data dictionary. Instead, it generates:

- scripts which are run from SQL*Plus to create column policies and section groups

- load files to be used with ctxload to create thesauri and synonym rings

The generated scripts are edited by the user in order to customize the migration to suit the needs of the application. The user needs to alter the creation of some preferences because of differences in ConText functionality (eg. filters supported internally) or to take advantage of new functionality.

Requirements and Limitations

The Migration Tool is aimed at migrating from any version of TextServer3 to Oracle8 ConText Cartridge 2.4.5. The database used by each product must be accessible to the user carrying out the migration.

The table that is a TextServer3 text table must exist on the target database (where ConText is installed). It need not be populated, but must have a column that has a unique or primary key constraint on it - this column must be the TextServer3 textkey column. The exp/imp utilities can be used for this purpose, or the TextServer3 database can be upgraded to 8.0.5.

> **Note:** A separate 7.2 ORACLE_HOME must be maintained if the user wants to carry on using the TextServer3 application.

ConText 2.4.5 must be installed on a 8.0.5 database. The user who uses the Migration Tool to generate scripts and load files need not be the user who ultimately owns the column policies and thesauri. The Migration Tool user must have privileges that allow the creation and dropping of database links for each TextServer3 user whose text tables and thesauri are to be migrated.

SQEs

TextServer3 has areas of functionality that are available in ConText, but the ConText functionality does not exactly parallel the TextServer3 functionality. One of these areas is Stored Query Expressions (SQEs). In TextServer3, an SQE can be a query against multiple text columns. These text columns do not have to be in the same text table. In addition, the query also supports the full 'SELECT ... FROM ... WHERE ...' clauses in a SQL statement.

The ConText implementation binds an SQE to a single column policy and specifies only the query expression. For these reasons, no attempt is made to migrate TextServer3 SQEs to ConText.

Thesaurus

Another area of TextServer3 functionality that differs from the ConText implementation is the thesaurus. A term in a TextServer3 thesaurus or synonym ring can be a proper term, a reference to another thesaurus or synonym ring, or a reference to an SQE. In ConText, a term can only be a proper term or phrase. When doing the thesaurus migration, the user has control over what happens when a term is a thesaurus or synonym ring, but no attempt to process a term is made when that term is an SQE.

Section Groups

If section groups are to be migrated, the section groups must be created in the ConText data dictionary *before* policy creation and indexing is carried out.

Installing the TextServer3 Dictionary Migration Tool

This section provides details for installing the Migration Tool.

During installation of the ConText Workbench, the creation scripts for the *CTXWMG_MIGRATE* package are copied to the client machine and an icon is created for executing the scripts; however the scripts are not executed to create the package.

Installing the Migration Tool involves using the icon to create the *CTXWMG_MIGRATE* package in the database on the server machine.

Pre-Installation Tasks

Before installing the Migration Tool, you must perform the following tasks:

1. Choose target directory for generated files.

The Migration Tool, once installed on the ConText database, can generate files for the tables and thesauri that are to be migrated. These files are owned by the owner of the Oracle8 shadow process. Therefore, choose a directory that can be accessed by the Oracle Server user who runs SQL*Plus and ctxload. This is because the generated files may need to be customized.

2. Set the UTL_FILE_DIR initialization parameter.

Accessible directories must be specified in the INIT.ORA file for the 8.0 instance. Each accessible directory is specified by the following parameter:

```
UTL_FILE_DIR = <directory name>
UTL_FILE_DIR = <directory name> ...
```

The named director(ies) are then available to the PL/SQL File I/O package.

> **Note:** On UNIX-based systems, and possibly other operating systems, the shadow process, using the PL/SQL File I/O package, can write to *all* file locations specified by the UTL_FILE_DIR parameter. There are *no* user-level file permissions. As a result, the UTL_FILE_DIR parameter can override operating system file permissions.
>
> For this reason, create a new directory for the purposes of migration so there is no possibility of existing files being overwritten.

3. Set the GLOBAL_NAMES initialization parameter.

You may require that the names of database links are different from the name of the database where they are linking. To enforce this, add the following parameter to INIT.ORA:

```
GLOBAL_NAMES = FALSE
```

4. Restart the database.

The database instance where ConText is installed must be restarted so that the UTL_FILE_DIR and GLOBAL_NAMES parameters can take effect.

5. Drop and recreate database links.

The Migration Tool uses a database link to access the TextServer3 dictionary. The link is called *ts3db*.

For every TextServer3 user that owns a text table or thesaurus to be migrated, the link must be dropped, and then recreated.

Creating a Link

Links should be created by the Oracle user that runs the Migration Tool. From SQL*Plus, use the following syntax:

```
SQL>  create database link ts3db connect to <user>
   2  identified by <pw> using '<connect_string>';
```

Dropping a Link

When the database link must be created for another TextServer3 user, drop the existing ts3db link:

```
SQL>  drop database link ts3db;
```

Installing the Migration Tool

To install the *CTXWMG_MIGRATE* package, double-click on the TextServer3 Migration Tool Installation icon.

Text Table Migration

During text table migration, a script is generated that creates, as closely as possible, a policy that reflects an indexable column in a TextServer3 text table. The user must specify a prefix for the naming of the preferences in the policy. If a TextServer3 text table contains multiple indexable columns, then a script is created for each column.

Preferences and Policies

ConText has a concept of preferences and policies. A preference specifies one of the indexing options that are necessary for creating a text index for a column:

- data store (where is the text stored?)
- filter (what format is the text in?)
- lexer (how are tokens in the text identified for indexing?)
- engine (how is the text index stored?)
- stoplist (are there words that should not be indexed?)
- wordlist (are there special querying features that you want to use?)

The preferences are grouped into a policy and the policy is assigned to a column. Then, a text index is created for the column policy.

A preference consists of a Tile, which identifies the indexing option for the preference, and a number of attributes for the Tile. The attributes are set for a Tile using the CTX_DDL.SET_ATTRIBUTE procedure.

> **See Also:** For more information about policies, preferences, and attributes, see *Oracle8 ConText Cartridge Administrator's Guide*

How Attributes are Determined for Preferences

A brief outline of how the Migration Tool handles each type of preference is given here.

Data Store Preference

If the TextServer3 column is EXTERNAL, then the OSFILE Tile is used to create a Data Store preference. If an External File Path for EXTERNAL columns is specified, then the *path* attribute of OSFILE is set appropriately. Otherwise, the predefined CTXSYS.DEFAULT_DIRECT_DATASTORE preference is used.

Filter Preferences

If the TextServer3 text table contains documents of a single format supported internally by ConText, then the FILTER attribute is set appropriately. If multiple formats are used, that are all supported internally by ConText, the predefined CTXSYS.AUTOB preference is used. However, if unsupported formats are used, the tool generates a commented-out list of CTX_DDL.SET_ATTRIBUTE calls, for the *executable* attribute, to match the formats that are recognized by Autorec. The user should then uncomment the appropriate ones and edit the executable name.

Lexer Preference

The BASIC LEXER Tile is used to create the Lexer preference. If the TextServer3 text column has column-level options defined, then these are used when setting lexer attributes, otherwise table-level options are used. The options map as follows:

TextServer3 Option	BASIC LEXER Tile Attribute(s)
Alpha Join	printjoins
Numeric Join	numjoin, numgroup
Continuation	continuation
case Conversion = BASE	BASE_LETTER set to 1

If a particular TextServer3 option is not set, then the equivalent BASIC LEXER attribute value is not set.

Additionally, if the TextServer3 text column references a section group, then the following two BASIC LEXER Tile attributes are set:

Attribute Name	Value(s)
startjoin	'<' and '/'
endjoin	'>'

The user may wish to edit these.

Engine Preference

The GENERIC ENGINE Tile is used to create the Engine preference. The resulting script contains a call to CTX_DDL.SET_ATTRIBUTE for each attribute of GENERIC ENGINE, but they are commented out. The values need to be edited by the user as appropriate, and the line uncommented.

Wordlist

The GENERIC WORDLIST tile is used to create the Wordlist preference. For the *stclause* and *instclause* attributes, a commented-out call to CTX_DDL.SET_ATTRIBUTE() exists.

stemmer is set to 1 (English) and *fuzzy_match* is set to 1 (English).

If the TextServer3 text column references a section group, then the *section_group* attribute is set appropriately.

Stoplist

If there are no stop words in the appropriate TextServer3 wordlist, then the predefined CTXSYS.NO_STOPLIST preference is used.

If there are less than 4096 lowercase stop words in the TextServer3 table wordlist with a length of less than 65 characters, then these words are used to make up a stoplist.

If there are more than 4096 stop words, the CTXSYS.DEFAULT_STOPLIST preference is used.

Generating Scripts For Column Policies

The generation of a script is straightforward after the tool has been installed and the UTL_FILE_DIR initialization parameter has been set.

The user needs to decide which table to migrate and what the prefix for the preference names are going to be. Then, the user calls the CTXWMG_MIGRATE.MIGRATE procedure to create the necessary scripts for the migration.

The preference names that the tool generates in the scripts are a concatenation of the prefix supplied by the user, the name of the preference category, and the name of the column for which each script is created.

Editing the Generated Scripts

The generated scripts contain calls to CTX_DDL.SET_ATTRIBUTE that have been commented out with the SQL*Plus command rem. The user must examine the script to determine whether such calls need uncommenting. The comments are used for attributes where the Migration Tool is unable to determine a reasonable value.

An example of this is for the storage parameters used by various tables and indexes in the Engine preference.

Column Policy Script Example

This example uses the MIGRATE procedure to create column policy scripts.

In this example, a TextServer3 table, owned by *jbloggs*, is called *resume* and has two indexable columns called *cv* and *address*. The directory that has been chosen as the target directory for the scripts is called /usr/home/joebloggs/migrate. The prefix convention chosen is RESUME_APP. A database link called *ts3db* has been created for *jbloggs*.

To generate scripts, the procedure MIGRATE is called:

```
begin
   ctxwmg_migrate.migrate (tabname=>'RESUME',
                           pref_conv=>'RESUME_APP',
                           scriptdir=>'/usr/home/joebloggs/migrate');
end;
```

From this example, two scripts would be generated, one for each of the indexable columns in *resume*:

- JBLOGGS_CV.sql, which creates a column policy called *jbloggs_cv*
- JBLOGGS_ADDRESS.sql, which creates a column policy called *jbloggs_address*

Thesaurus Migration

The migration of TextServer3 thesauri and synonym rings to ConText is not as straightforward as the migration of TextServer3 tables. This is because in ConText each term in a thesaurus must be a phrase. In TextServer3, narrower terms can be another thesaurus or a synonym ring. A related term can also be a thesaurus or synonym ring. This means that the functionality of a TextServer3 thesaurus is determined just as much by its structure and its relationship with other thesauri/synonym rings, as by the query operators made available.

Another difference is that a thesaurus in TextServer3 can contain only one hierarchy. In ConText, a thesaurus can contain multiple hierarchies. A ConText thesaurus can also contain synonym rings. The load files create Standard Narrower Term hierarchies.

A term in a ConText synonym ring must be unique to that synonym ring within a particular thesaurus. All synonym ring terms are given a qualifier so that conflicts do not arise. The qualifier is the name of the TextServer3 synonym ring.

The Migration Tool provides four procedures for use in the migration of thesauri and synonym rings. Each of these procedures generates a load file that can be used with ctxload. The ctxload utility can be used for loading thesauri into the ConText dictionary.

> **See Also:** For more information about thesauri and synonym rings, see *Oracle8 ConText Cartridge Administrator's Guide*

Section Group Migration

Section groups can be migrated from TextServer3 to ConText. The generated script should be run from SQL*Plus to create the section group and add sections to it.

> **See Also:** For more information about sections and section groups, see *Oracle8 ConText Cartridge Administrator's Guide*

CTXWMG_MIGRATE: TextServer3/ConText Migration Tool Package

The CTXWMG_MIGRATE PL/SQL package contains the following stored procedures:

Name	Description
MIGRATE	Generates a script for each indexable column in a TextServer3 text table. Each script is used to create a column policy.
MIGRATE_ALL_SYNS	Generates a load file containing all TextServer3 synonym rings owned by a TextServer3 user.
MIGRATE_ALL_THES	Generates a load file containing all TextServer3 thesauri owned by a TextServer3 user.
MIGRATE_SG	Generates a script for a section group.
MIGRATE_SYN	Generates a load file containing a named TextServer3 synonym ring.
MIGRATE_THES	Generates a load file containing a named TextServer3 thesaurus and, optionally, any synonym rings owned by the TextServer3 user.

MIGRATE

The MIGRATE procedure creates a SQL script for a TextServer3 text table. When the script is run, it generates a ConText policy for an existing table in a ConText database.

Syntax

```
CTXWMG_MIGRATE.MIGRATE_(tabname   IN VARCHAR2,
                        pref_conv IN VARCHAR2,
                        scriptdir IN VARCHAR2);
```

tabname
Specify the name of the TextServer3 text table that contains one or more indexable columns for which you want to create a ConText policy.

pref_conv
Specify the string of characters used to generate a prefix for the preference names in the script.

scriptdir
Specify the directory for the script generated by MIGRATE.

Notes

The preference names that the tool generates in the script are a concatenation of the prefix supplied by the user, the name of the preference category, and the name of the column for which the script is being created.

For example, if the tool is creating a script for the *doc* column, and the user specifies a prefix of *myapp*, then the Engine preference is called *myapp_engine_doc*.

If the preference is being created for an internal filter, then the name of the preference category is substituted with four letters representing the format (e.g. WP51 for WordPerfect 5.1).

MIGRATE_ALL_SYNS

The MIGRATE_ALL_SYNS procedure creates a single load file for all the synonym rings owned by the user.

Syntax

```
CTXWMG_MIGRATE.MIGRATE_ALL_SYNS(loaddir IN VARCHAR2);
```

loaddir
Specify the directory for load file generated by MIGRATE_ALL_SYNS.

Notes

No 'in-place' expansion occurs.

When the load file generated by MIGRATE_ALL_SYNS is used to generate a ConText thesaurus, all the synonym rings from TextServer3 are created within one ConText thesaurus.

The name of the generated load file is based on the user ID for the TextServer3 user. For example, for a user with a TextServer3 user ID of 3, the generated file is called ALL_SYNONYMS_3.syn.

MIGRATE_ALL_THES

The MIGRATE_ALL_THES procedure creates a single load file for all the thesauri owned by the user.

Syntax

```
CTXWMG_MIGRATE.MIGRATE_ALL_THES(loaddir IN VARCHAR2);
```

loaddir
Specify the directory for load file generated by MIGRATE_ALL_THES.

Notes

No 'in-place' expansion occurs.

When the load file generated by MIGRATE_ALL_THES is used to generate a ConText thesaurus, all the hierarchies from the TextServer3 thesauri are created within one ConText thesaurus.

The name of the generated load file is based on the user ID for the TextServer3 user. For example, for a user with a TextServer3 user ID of 3, the generated file is called ALL_THESAURI_3.ths.

MIGRATE_SG

The MIGRATE_SG procedure generates a script for a named section group.

Syntax

```
CTXWMG_MIGRATE.MIGRATE_SG(group_name IN VARCHAR2,
                          loaddir    IN VARCHAR2);
```

group_name
Specify the name of the section group to be migrated.

loaddir
Specify the directory for script generated by MIGRATE_SG.

Notes

The name of the generated script is based on the name of the TextServer3 section group. For example, for a TextServer3 section group called MYSECTIONGROUP, the generated script is called SG_MYSECTIONGROUP.sql.

MIGRATE_SYN

The MIGRATE_SYN procedure generates a load file for a named TextServer3 synonym ring.

Syntax

```
CTXWMG_MIGRATE.MIGRATE_SYN(syn_sname IN VARCHAR2,
                           loaddir   IN VARCHAR2);
```

syn_name
Specify the name of the synonym ring to be migrated.

loaddir
Specify the directory for load file generated by MIGRATE_SYN.

Notes

No 'in-place' expansion occurs.

The name of the TextServer3 synonym ring is used as a qualifier for each term.

The name of the generated load file is based on the thesaurus name and the user ID for the TextServer3 user. For example, for a TextServer3 synonym ring called MYSYNRING belonging to a user with a TextServer3 user ID of 3, the generated file is called MYSYNRING_3.ths.

MIGRATE_THES

The MIGRATE_THES procedure generates a load file for a named TextServer3 thesaurus.

Syntax

```
CTXWMG_MIGRATE.MIGRATE_THES(thesname       IN VARCHAR2,
                            t_expand_mode  IN INTEGER DEFAULT 0,
                            s_expand_mode  IN INTEGER DEFAULT 0,
                            dump_syns      IN INTEGER DEFAULT 0,
                            loaddir        IN VARCHAR2);
```

thesname
Specify the name of the thesaurus to be migrated.

t_expand_mode
Specify the expansion mode for referenced thesauri:

   0 No thesaurus expansion

   1 Expand user referenced thesauri in-place

   2 Expand user and public referenced thesauri in-place

s_expand_mode
Specify the expansion for referenced synonym rings:

   0 No synonym ring expansion

   1 Expand single user synonym ring in-place

   2 Expand multiple user synonym rings in-place

   3 Expand single user or public synonym ring in-place

   4 Expand multiple user or public synonym rings in-place

dump_syns
Specify to dump all synonym rings owned by current user.

loaddir
Specify the directory for load file generated by MIGRATE_THES.

Notes

The *t_expand_mode* and *s_expand_mode* parameters allow users to control what happens when the thesaurus hierarchy contains a narrower term that is a synonym ring.

If thesaurus expansion is used, a referenced thesaurus is expanded 'in-place'. This means that instead of the term being a reference to a thesaurus, it becomes the top-term of the referenced thesaurus. This effectively makes the referenced hierarchy part of the main hierarchy. If a thesaurus that has been made part of the main hierarchy references other thesauri itself, then those thesauri also become part of the main hierarchy. Referenced thesauri may be owned by the user or may be public. Expansion of referenced thesauri can be limited to those owned by the user if required.

In-place synonym ring expansion is similar to thesaurus expansion, but the terms in the referenced ring become narrower terms of the referencing term. The user can limit expansion to those synonym rings owned by the user. It is possible for a synonym ring term to be a reference to another synonym ring. Expansion of such a synonym ring would result in those terms also being narrower terms of the term in the thesaurus. The user can control this, as well as being able to limit the expansion to synonym rings owned by the user.

If a value of 1 is specified for the *dump_syns* parameter, MIGRATE_THES dumps all synonym rings into the same load file. When the load file is used to create a thesaurus in ConText, all the dumped TextServer3 synonym rings are part of that ConText thesaurus.

The name of the generated load file is based on the thesaurus name and the user ID for the TextServer3 user. For example, for a TextServer3 thesaurus called MYTHES belonging to a user with a TextServer3 user ID of 3, the generated file is called MYTHES_3.ths.

Example Migration Using Supplied Script

A sample script is supplied with the Migration Tool. It can be altered to suit the user's requirements. It prompts the user for values.

The following example makes use of the sample script to demonstrate how a user's text table and thesaurus can be migrated from TextServer3 to ConText.

Example Description

In this example, the TextServer3 user is *jbloggs* on a TextServer3 database using a connect string 'prod_db'. The *jbloggs* user has a TextServer3 user ID of 3 and one text table called *odc_papers:*

- the textkey column is *paper_id*

- the text columns are *summary* and *paper*

- the table has a format ID column called *format_id*

The *summary* column and the documents in the *paper* column are in the same format, which is either Word for Windows 6 or WordPerfect 6 format. Both of these formats are supported internally by ConText. Neither of these two columns are external.

The application uses a single thesaurus, called *MY_THESAURUS*, but it does contain terms that are references to other thesauri and synonym rings. Some of these are owned by the user, others are public. The user requires the same terms to be returned from operations on the thesaurus using ConText.

The directory chosen for creating the column policy script and load files is */usr/home/joebloggs/migrate*. The script creates the database link, and drops it after finishing the script generation.

Using the Example

The steps for migrating the text table and thesaurus are:

1. Install the Migration Tool on the ConText database. For the purpose of this exercise, it is installed for the *jbloggs* user.

2. In the ConText database, create the TextServer3 table to be migrated. It need not be populated at this stage.

3. Add the UTL_FILE_DIR parameter to the init.ora file used by the ConText database and restart this database.

4. As the Oracle user for whom the migration package was installed, run the script samp_mig.sql:

```
% sqlplus jbloggs/jbloggs @samp_mig
```

Enter the following information at the prompts:

```
jbloggs
jbloggs
prod_db
/usr/home/joebloggs/migrate
odc_papers
joes_app
2
4
1
```

Example Results

The Migration Tool creates three script files in the directory */usr/home/joebloggs/migrate*:

- *JBLOGGS_SUMMARY.sql* - creates a policy called JBLOGGS_SUMMARY

- *JBLOGGS_PAPER.sql* - creates a policy called JBLOGGS_PAPER

- *MY_THESAURUS_3.ths*

If these scripts are then run from SQL*Plus, they create column policies for each of the text columns. Both column policies use the Autorec functionality to invoke the internal filters since the Migration Tool determines that all the formats used are supported by ConText. Before running the scripts, the user may choose to edit the preference name or any of the attribute values.

The thesaurus is migrated to ConText by using the ctxload utility. Assuming that the thesaurus should keep the name 'MY_THESAURUS', the following command would import the thesaurus:

```
% ctxload -user jbloggs/jbloggs -name MY_THESAURUS -file MY_THESAURUS_3.ths -thes
```

This completes the migration of the TextServer3 Dictionary information for the text table, ODC_PAPERS. The user should now import the data from the text table in the TextServer3 database and create a ConText index for the data.

A

Viewer Cartridge: Manual Configuration

This appendix documents the various procedures for manually configuring and deconfiguring Oracle WebServer 2.1 and Oracle Web Application Server 3.0 for use with the ConText Viewer Cartridge.

> **Note:** You only need to refer to this appendix if configuration of the Viewer Cartridge via the Oracle Installer failed.
>
> For more information, see "Configuring the Viewer Cartridge" in Chapter 3, "Document Viewing".

The following topics are covered in this appendix:

- Notational Conventions for Configuration/Deconfiguration
- Configuring Oracle WebServer 2.1 for the Viewer Cartridge
- Deconfiguring Oracle WebServer 2.1
- Configuring Oracle Web Application Server 3.0 for the Viewer Cartridge
- Deconfiguring Oracle Web Application Server 3.0

Notational Conventions for Configuration/Deconfiguration

The section provides details for the notational conventions that apply to all of the manual configuration and deconfiguration procedures for Oracle WebServer 2.1 and Oracle Web Application Server 3.0.

Buttons

Buttons referenced appear within square brackets, with capitalization the same as it appears in the browser.

For example: [Modify Listener]

Links

Links appear within angle brackets.

For example: <Web Request Broker>

Pathnames

All paths mentioned in the configuration steps follow the UNIX convention of directory names separated by forward slashes (i.e. '/'). If a path is a physical path (as opposed to a virtual path), and the platform is Windows NT, instead of a UNIX-based platform), then use the backslash character (i.e. '\') to separate directory names.

Configuring Oracle WebServer 2.1 for the Viewer Cartridge

Configuration of Oracle WebServer 2.1 for use with the ConText Viewer Cartridge is normally performed through a separate Oracle Installer session after installation of the ConText Workbench.

In the event that the WebServer portion of the configuration fails, manual configuration is necessary to set up WebServer for use with the Viewer Cartridge.

> **Note:** These manual procedures are different from those that apply for Oracle Web Application Server 3.0.
>
> If Web Application Server is installed, see "Configuring Oracle Web Application Server 3.0 for the Viewer Cartridge".

Manual configuration is performed through a browser, such as Netscape Navigator or Microsoft Internet Explorer. The browser should support forms and frame capabilities.

Deconfiguration procedures can also be applied to remove the Viewer Cartridge from WebServer.

> **See Also:** For more information about deconfiguring WebServer to remove the Viewer Cartridge, see "Deconfiguring Oracle WebServer 2.1" in this chapter.

Configuration Requirements

The following information is required during configuration:

- the name of the machine on which WebServer is running
- the URL of the WebServer Administration Page; usually:

  ```
  http://<hostname>:8888/ows-adoc/Intro.html
  ```

- the name of the listener used to access the Viewer Cartridge
- the type of platform that WebServer is running on (UNIX-based or Windows NT)
- the Oracle home directory for the WebServer installation

> **Note:** In certain steps of the configuration process for WebServer, it is necessary to enter file system paths. Part of the path may be the Oracle home directory, which is represented in the configuration steps as %ORACLE_HOME%. The actual value of ORACLE_HOME should be used instead.
>
> The following example assumes an Oracle home directory of D:\ORANT on Windows NT. If the step asks you to enter the following value in a field:
>
> ```
> %ORACLE_HOME%/ctxw/middle/ctxvcart/
> ```
>
> then enter the value:
>
> ```
> D:\ORANT\ctxw\middle\ctxvcart\
> ```

Configuration Tasks

There are three major tasks in the configuration procedure:

- Configuring Oracle Web Listener
- Creating the Web Request Broker Cartridge Configuration
- Configuring Oracle Web Request Broker

Configuring Oracle Web Listener

This section describes how to configure a Web Listener for use with the Viewer Cartridge. A MIME type is mapped to a File Extension for use with the Viewer Cartridge. Security for the virtual directory is also configured.

1. Access the WebServer Administration page.

2. On the WebServer Administration page:

   Select the <Oracle Web Listener> link.

   The Oracle Web Listener Administration page appears.

3. On the Oracle Web Listener (OWL) Administration page:

   Select <CONFIGURE> for the listener used to access the Viewer Cartridge.

   The OWL Administration - Server Advanced Configuration page appears.

4. On the OWL Administration - Server Advanced Configuration page:

Scroll down to the MIME Types block of fields to insert values into the first empty row.

In the Mime Type column, enter:

```
application/x-ctxv
```

In the File Extension(s) column, enter:

```
ctxv
```

Click [Modify Listener].

The Oracle Web Listener Administration page appears.

5. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener used to access the Viewer Cartridge.

The OWL Administration - Server Advanced Configuration page appears.

6. On the OWL Administration - Server Advanced Configuration page, select <Security: Access Control and Encryption>.

The Oracle Web Listener Security page appears.

7. On the Oracle Web Listener Security page, scroll to the Basic Users block of fields. In the User Name column of a blank row, enter:

```
ctxquery
```

In the Password field, enter:

```
ctxquery
```

Scroll to the Basic Groups block of fields. In the Group Name column, enter:

```
ctxview_group
```

In the User(s) column, enter:

```
ctxquery
```

Scroll to the Basic Realms block of fields. In the Realms column, enter:

```
ctxview_realm
```

In the Group(s) column, enter:

`ctxview_group`

Scroll to the Protection block of fields. In the Virtual Path column, enter:

`/ctxwview/`

In the Basic/Digest column, select Basic.

In the Realm column, enter:

`ctxview_realm`

Click [Modify Listener].

The Oracle Web Listener Security page reappears.

Scroll to the end of the page and click [Administration].

Creating the Web Request Broker Cartridge Configuration

This section describes how to create the Web Request Broker cartridge configuration for the ConText Viewer Cartridge. The configuration provides the Viewer Cartridge with information it requires when processing requests.

1. Access the WebServer Administration page.

2. On the WebServer Administration page, select the <Web Request Broker> link.

   The Web Request Broker Administration page appears.

3. On the Web Request Broker Administration page, select <Modify> for the listener used to access the Viewer Cartridge.

   The WRB Cartridge Administration Page appears.

4. On the WRB Cartridge Administration Page:

5. In the Cartridges field, enter:

   `CTXWVIEW`

   Click [Create Cartridge].

   The WRB Cartridge Configuration Page appears.

6. On the WRB Cartridge Configuration Page, in row 1, column 1 of the Cartridge Parameters block of fields, enter:

```
CLASSPATH
```

If the WebServer platform is UNIX-based, in row 1, column 2, enter:

```
%ORACLE_HOME%/ows21/java/classes.zip:%ORACLE_HOME%/ows21/javaoracle.zip:%ORACLE_
HOME%/jdbc/lib/classes102.zip:%ORACLE_HOME%/ctxw/middlectxvcart/ctxvcart.zip
```

If the WebServer platform is NT, in row 1, column 2, enter:

```
%ORACLE_HOME%\ows21\java\classes.zip;%ORACLE_HOME%\ows21\java\oracle.zip;%ORACLE_
HOME%\jdbc\lib\classes102.zip;%ORACLE_HOME%\ctxw\middle\ctxvcart\ctxvcart.zip
```

In row 2, column 1, enter:

```
LD_LIBRARY_PATH
```

If the WebServer platform is UNIX-based, in row 2, column 2, enter:

```
%ORACLE_HOME%/ows21/bin:%ORACLE_HOME%/ows21/lib:%ORACLE_HOME%/ows21/java/lib:%ORACLE_HOME%/jdbc/lib
```

If the WebServer platform is NT, in row 2, column 2, enter:

```
%ORACLE_HOME%\ows21\bin
```

In row 3, column 1, enter:

```
JAVA_HOME
```

In row 3, column 2, enter the physical path:

```
%ORACLE_HOME%/ows21/java
```

Click the [Modify Cartridge Configuration] button.

The WRB Cartridge Configuration page reappears.

Configuring Oracle Web Request Broker

This section describes how to set up the ConText Viewer Cartridge so that the Web Request Broker knows how to invoke it. The physical cartridge is mapped to the

virtual path of the Viewer Cartridge, and security for the Viewer Cartridge is configured.

1. Access the WebServer Administration page.

2. On the WebServer Administration page, select the <Web Request Broker> link.

    The Web Request Broker Administration page appears.

3. On the Web Request Broker Administration page, select <Modify> for the listener used to access the Viewer Cartridge.

    The WRB Cartridge Administration Page appears.

4. On the WRB Cartridge Administration Page, scroll to the Application and Objects block of fields.

    In the App. column, enter:

    ```
    CTXWVIEW
    ```

    If the WebServer platform is UNIX-based, in the Object Path column, enter:

    ```
    %ORACLE_HOME%/ows21/lib/libjava.so
    ```

    If the WebServer platform is NT, in the Object Path column, enter:

    ```
    %ORACLE_HOME%\ows21\bin\javai.dll
    ```

    In the Entry Point column, enter:

    ```
    ojsdinit
    ```

    In the Min column, enter:

    ```
    0
    ```

    In the Max column, enter:

    ```
    30
    ```

    Scroll to the Applications and Directories block of fields. In the Virtual Path column, enter:

    ```
    /ctxwview
    ```

In the App. column, enter:

`CTXWVIEW`

In the Physical Path column, enter:

`%ORACLE_HOME%/ctxw/middle/ctxvcart/ctxvcart.zip`

Scroll to the Protecting Applications block of fields. In the Virtual Path column, enter:

`/ctxwview/`

In the Basic/Digest column, select Basic.

In the Realm column, enter:

`ctxview_realm`

Click [Modify WRB Configuration].

The WRB Cartridge Administration page reappears.

The configuration of Oracle WebServer 2.1 for use with the ConText Viewer Cartridge is now complete.

To use WebServer and the Viewer Cartridge, stop and restart the associated WebServer.

Deconfiguring Oracle WebServer 2.1

This section provides detailed instructions for deconfiguring WebServer for the Viewer Cartridge.

Deconfiguration Requirements

The following information is required before deconfiguration starts:

- the name of the listener configured to access the Viewer Cartridge
- the URL of the WebServer Manager Home Page

> **Note:** In certain steps of the deconfiguration process for WebServer, it is necessary to remove file system paths. A portion of the path may be the Oracle home directory, which is represented in the configuration steps as %ORACLE_HOME%. The actual value of ORACLE_HOME should be used instead.
>
> The following example assumes an Oracle home directory of D:\ORANT on Windows NT. If the step asks you to remove the following value from a field:
>
> ```
> %ORACLE_HOME%/ctxw/middle/ctxvcart/
> ```
>
> then remove the value:
>
> ```
> D:\ORANT\ctxw\middle\ctxvcart\
> ```

Deconfiguration Tasks

There are three major tasks in the deconfiguration procedure:

- Deconfiguring Oracle Web Listener
- Deleting the Web Request Broker Cartridge Configuration
- Deconfiguring Oracle Web Request Broker

> **Note:** WebServer must be restarted once deconfiguration has completed.

Deconfiguring Oracle Web Listener

This section describes how to remove the Viewer Cartridge configuration from the Web Listener used to access the Viewer Cartridge.

1. Access the WebServer Administration page.

2. On the WebServer Administration page, select the <Oracle Web Listener> link.

   The Oracle Web Listener Administration page appears.

3. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener which is used to access the Viewer Cartridge.

   The OWL Administration - Server Advanced Configuration page appears.

4. On the OWL Administration - Server Advanced Configuration page, scroll down to the MIME Types block of fields.

   Remove all values from the row that has the following value in the Mime Type column:

   ```
   application/x-ctxv
   ```

   Click [Modify Listener].

   The Oracle Web Listener Administration page appears.

5. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener which is used to access the Viewer Cartridge.

   The OWL Administration - Server Advanced Configuration page appears.

6. On the OWL Administration - Server Advanced Configuration page, select <Security: Access Control and Encryption>.

   The Oracle Web Listener Security page appears.

7. On the Oracle Web Listener Security page, scroll to the Basic Users block of fields.

   Remove all values from the row that has the following value in the User Name column:

   ```
   ctxquery
   ```

   Scroll to the Basic Groups block of fields. Remove all values from the row that has the following value in the Group Name column:

```
ctxview_group
```

Scroll to the Basic Realms block of fields. Remove all values from the row that has the following value in the Realms column:

```
ctxview_realm
```

Scroll to the Protection block of fields. Remove all values from the row that has the following value in the Virtual Path column:

```
/ctxwview/
```

Click [Modify Listener].

The Oracle Web Listener Security page reappears.

Scroll to the end of the page and click [Administration].

Deleting the Web Request Broker Cartridge Configuration

This section describes how to remove the Web Request Broker cartridge configuration for the ConText Viewer Cartridge.

1. Access the WebServer Administration page.

2. On the WebServer Administration page, select the <Web Request Broker> link.

   The Web Request Broker Administration page appears.

3. On the Web Request Broker Administration page, select <Modify> for the listener which is used to access the Viewer Cartridge.

   The WRB Cartridge Administration Page appears.

4. On the WRB Cartridge Administration Page, in the Cartridges section, select <Delete> for the CTXWVIEW cartridge.

   The Confirm Cartridge Deletion page appears.

   Click [Confirm Delete].

   The resulting page shows that the CTXWVIEW cartridge configuration has been deleted.

Deconfiguring Oracle Web Request Broker

This section describes how to remove the ConText Viewer Cartridge configuration from the Web Request Broker of the Web Listener used to access the Viewer Cartridge.

1. Access the WebServer Administration page.

2. On the WebServer Administration page, select the <Web Request Broker> link.

   The Web Request Broker Administration page appears.

3. On the Web Request Broker Administration page, select <Modify> for the listener which is used to access the Viewer Cartridge.

   The WRB Cartridge Administration Page appears.

4. On the WRB Cartridge Administration Page, scroll to the Application and Objects block of fields.

   Remove all values from the row that has the following value in the App. column:

   `CTXWVIEW`

   Scroll to the Applications and Directories block of fields. Remove all values from the row that has the following value in the Virtual Path column:

   `/ctxwview`

   Scroll to the Protecting Applications block of fields. Remove all values from the row that has the following value in the Virtual Path column:

   `/ctxwview/`

   Click [Modify WRB Configuration].

   The WRB Cartridge Administration page reappears.

The deconfiguration of Oracle WebServer 2.1 for use with the ConText Viewer Cartridge is now complete.

Configuring Oracle Web Application Server 3.0 for the Viewer Cartridge

Configuration of Oracle Web Application Server 3.0 for use with the ConText Viewer Cartridge is normally performed through a separate Oracle Installer session after installation of the ConText Workbench.

In the event that the Web Application Server portion of the configuration fails, manual configuration is necessary to set up Web Application Server for use with the Viewer Cartridge.

> **Note:** These manual procedures are different from those that apply for Oracle WebServer 2.1.
>
> If WebServer is installed, see "Configuring Oracle WebServer 2.1 for the Viewer Cartridge" in this chapter.

Manual configuration is performed through a browser, such as Netscape Navigator or Microsoft Internet Explorer. The browser should support forms and frame capabilities.

Deconfiguration procedures can also be applied to remove the Viewer Cartridge from Web Application Server.

> **See Also:** For more information about deconfiguring Web Application Server to remove the Viewer Cartridge, see "Deconfiguring Oracle Web Application Server 3.0" in this chapter.

Configuration Requirements

The following information is required during configuration:

- the name of the machine on which Web Application Server is running
- the URL of the Web Application Server Manager Home Page; usually:

  ```
  http://<hostname>:8888/ows-adoc/Intro.html
  ```

- the name of the listener used to access the Viewer Cartridge
- the type of platform that Web Application Server is running on (UNIX-based or Windows NT)

> **Note:** In certain steps of the configuration process for Web Application Server, it is necessary to enter file system paths. Part of the path may be the Oracle or Web Application Server home directory, which is represented in the configuration steps as either %ORACLE_HOME% or %ORAWEB_HOME%.
>
> *Do not* substitute the real values unless explicitly instructed to do so.

Configuration Tasks

There are four major tasks in the configuration procedure:

- Configuring the Oracle Web Listener
- Creating the Web Request Broker Cartridge
- Configuring Viewer Cartridge Specific Parameters
- Configuring Authorization Server

> **Note:** Web Application Server must be restarted once configuration has completed.

Configuring the Oracle Web Listener

This section describes how to configure a Web Listener for use with the Viewer Cartridge. A MIME type is mapped to a File Extension for use with the Viewer Cartridge. Security for the virtual directory is also configured.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Listener> link.

   The Oracle Web Listener Administration page appears.

3. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener used to access the Viewer Cartridge.

   The OWL Administration - Server Advanced Configuration page appears.

4. On the OWL Administration - Server Advanced Configuration page, scroll down to the MIME Types block of fields to insert values into the first empty row.

In the Mime Type column, enter:

`application/x-ctxv`

In the File Extension(s) column, enter:

`ctxv`

Click [Modify Listener].

The Oracle Web Listener Administration page appears.

5. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener used to access the Viewer Cartridge.

   The OWL Administration - Server Advanced Configuration page appears.

6. On the OWL Administration - Server Advanced Configuration page, select the <Security> link.

   The Oracle Web Listener Security page appears.

7. On the Oracle Web Listener Security page, scroll to the Basic Users block of fields.

   In the User Name column of a blank row, enter:

   `ctxquery`

   In the Password field, enter:

   `ctxquery`

   Scroll to the Basic Groups block of fields. In the Group Name column, enter:

   `ctxview_group`

   In the User(s) column, enter:

   `ctxquery`

   Scroll to the Basic Realms block of fields. In the Realms column, enter:

   `ctxview_realm`

In the Group(s) column, enter:

`ctxview_group`

Scroll to the Protection block of fields. In the Virtual Path column, enter:

`/ctxwview/`

In the Basic/Digest column, select Basic.

In the Realm column, enter:

`ctxview_realm`

Click [Modify Listener].

The Oracle Web Listener Security page reappears.

Scroll to the end of the page and click [Admin].

Creating the Web Request Broker Cartridge

This section describes how to create a configuration for the ConText Viewer Cartridge. The configuration provides the Web Request Broker with information required to access the Viewer Cartridge; it also defines and protects the virtual path used to access the cartridge.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Application Server> link.

   The Web Application Server Administration page appears.

3. On the Web Application Server Administration page, select the <Cartridge Administration> link.

   The Cartridge Administration Page appears.

4. On the Cartridge Administration Page, select the <Add New Cartridge> link.

   The New Cartridge Configuration Page appears.

5. On the New Cartridge Configuration Page, select the <Add New Cartridge with Manual Configuration> link.

   The New Cartridge Configuration Page appears.

6. On the New Cartridge Configuration Page, in the Cartridge Name field, enter:

```
CTXWVIEW
```

If the Web Application Server platform is UNIX-based, in the Object Path column, enter:

```
%ORAWEB_HOME%/lib/libjava.so
```

If the Web Application Server platform is Windows NT, in the Object Path column, enter:

```
%ORAWEB_HOME%\bin\javai.dll
```

In the Entry Point field, enter:

```
ojsdinit
```

In the Minimum # of Instances field, enter:

```
0
```

In the Maximum # of Instances field, enter:

```
100
```

In the Cartridge Description field, enter:

```
ConText Viewer Cartridge
```

In the Cartridge Icon field, enter:

```
/ows-img/32java.gif
```

In the Client Certificate field, select:

```
Disabled
```

In the Client Sessions field, select:

```
Disabled
```

In the Max session idle time field, select:

```
0
```

In the Error Page field, enter

```
%ORAWEB_HOME%/admdoc/wrberr.html
```

Scroll to the Virtual Paths block of fields. In the Virtual Path column, enter:

```
/ctxwview
```

In the Physical Path column, enter:

```
%ORACLE_HOME%/ctxw/middle/ctxvcart/ctxvcart.zip
```

Scroll to the Virtual Path Protection block of fields. In the Virtual Path column, enter:

```
/ctxwview/
```

In the Scheme field, select:

```
Basic
```

In the Realm field, enter:

```
ctxview_realm
```

Click [Register New Cartridge].

The Cartridge Administration page reappears.

Configuring Viewer Cartridge Specific Parameters

This section describes how to create specific parameters used by the ConText Viewer Cartridge when processing requests.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Application Server> link.

   The Web Application Server Administration page appears.

3. On the Web Application Server Administration page, select the <Cartridge Administration> link.

The Cartridge Administration Page appears.

4. On the Cartridge Administration Page, select the <ConText Viewer Cartridge> link.

The CTXWVIEW Cartridge Configuration page appears.

Select the <CTXWVIEW Cartridge specific parameters> link.

The Cartridge Configuration Page appears.

5. On the Cartridge Configuration Page, in row 1, column 1 of the Cartridge Parameters block of fields, enter:

```
CLASSPATH
```

If the Web Application Server platform is UNIX-based, in row 1, column 2, enter:

```
%ORAWEB_HOME%/java/classes.zip:%ORAWEB_HOME%/java/oracle.zip:%ORACLE_HOME%/jdbc/lib/classes102.zip:%ORACLE_
HOME%/ctxw/middle/ctxvcart/ctxvcart.zip
```

If the Web Application Server platform is NT, in row 1, column 2, enter:

```
%ORAWEB_HOME%\java\classes.zip:%ORAWEB_HOME%\java\oracle.zip:%ORACLE_HOME%\jdbc\lib\classes102.zip:%ORACLE_
HOME%\ctxw\middle\ctxvcart\ctxvcart.zip
```

In row 2, column 1, enter:

```
LD_LIBRARY_PATH
```

If the Web Application Server platform is UNIX-based, in row 2, column 2, enter:

```
%ORAWEB_HOME%/bin:%ORAWEB_HOME%/lib:%ORAWEB_HOME%/java/lib:%ORACLE_HOME%/jdbc/lib
```

If the Web Application Server platform is NT, in row 2, column 2, enter:

```
%ORAWEB_HOME%\bin
```

In row 3, column 1, enter:

```
JAVA_HOME
```

In row 3, column 2, enter the physical path:

`%ORAWEB_HOME%/java`

Click [Modify Cartridge Configuration].

The Cartridge Configuration page reappears.

Configuring Authorization Server

This section describes how to configure the Authorization Server for use with ConText Viewer Cartridge. It provides authentication for Web Application Server cartridges to authenticate users before allowing them to use the cartridge.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Application Server> link.

    The Web Application Server Administration page appears.

3. On the Web Application Server Administration page, select the <Authorization Server> link.

    The Authentication Configuration page appears.

4. On the Authentication Configuration page, select the <Basic> link.

    The Basic Authentication page appears.

5. On the Basic Authentication page, scroll to the Basic Users block of fields.

    In the User Name column, enter:

    `ctxquery`

    In the Password column, enter:

    `ctxquery`

    Scroll to the Basic Groups block of fields. In the Group Name column, enter:

    `ctxview_group`

    In the User(s) column, enter:

```
ctxquery
```

Scroll to the Basic Realms block of fields. In the Realms column, enter:

```
ctxview_realm
```

In the Group(s) column, enter:

```
ctxview_group
```

Click [Modify].

The resulting page shows that the Basic Authentication was updated.

The configuration of Oracle Web Application Server 3.0 for use with the ConText Viewer Cartridge is now complete. Scroll to the end of the page and click [Administration].

Deconfiguring Oracle Web Application Server 3.0

This section describes how to remove the ConText Virtual Cartridge configuration from the Web Listener used to access the Viewer Cartridge.

Web Application Server needs to be restarted after the deconfiguration has been completed.

Deconfiguration Requirements

The following information is required before deconfiguration starts:

- the name of the listener configured to access the Viewer Cartridge
- the URL of the Web Application Server Manager Home Page

Deconfiguration Tasks

There are four major tasks in the deconfiguration procedure:

- Deconfiguring the Oracle Web Listener
- Deleting Web Request Broker Cartridge
- Deconfiguring Viewer Cartridge Specific Parameters
- Deconfiguring Authorization Server

Deconfiguring the Oracle Web Listener

This section describes how to remove the ConText Viewer Cartridge configuration from the Web Listener used to access the cartridge.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Listener> link.

   The Oracle Web Listener Administration page appears.

3. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener which is used to access the Viewer Cartridge.

   The OWL Administration - Server Advanced Configuration page appears.

4. On the OWL Administration - Server Advanced Configuration page, scroll down to the MIME Types block of fields.

Remove all values from the row that has the following value in the Mime Type column:

```
application/x-ctxv
```

Click [Modify Listener].

The Oracle Web Listener Administration page appears.

5. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener which is used to access the Viewer Cartridge.

    The OWL Administration - Server Advanced Configuration page appears.

6. On the OWL Administration - Server Advanced Configuration page, select <Security>.

    The Oracle Web Listener Security page appears.

7. On the Oracle Web Listener Security page, scroll to the Basic Users block of fields.

    Remove all values from the row that has the following value in the User Name column:

```
ctxquery
```

    Scroll to the Basic Groups block of fields. Remove all values from the row that has the following value in the Group Name column:

```
ctxview_group
```

    Scroll to the Basic Realms block of fields. Remove all values from the row that has the following value in the Realms column:

```
ctxview_realm
```

    Scroll to the Protection block of fields. Remove all values from the row that has the following value in the Virtual Path column:

```
/ctxwview/
```

    Click [Modify Listener].

    The Oracle Web Listener Security page reappears.

Scroll to the end of the page and click [Administration].

Deleting Web Request Broker Cartridge

This section describes how to delete the Viewer Cartridge from the Web Request Broker.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Application Server> link.

    The Web Application Server Administration page appears.

3. On the Web Application Server Administration page, select the <Cartridge Administration> link.

    The Cartridge Administration Page appears.

4. On the Cartridge Administration Page, select <Delete Cartridge>.

    The Delete Cartridge Configuration Page appears.

5. On the Delete Cartridge Configuration Page, in the Select a Cartridge field, select:

    `CTXWVIEW`

    Click [Delete].

    The Confirm Cartridge Deletion page appears.

    Click [Delete].

    The resulting page shows that the CTXWVIEW cartridge configuration has been deleted.

Deconfiguring Viewer Cartridge Specific Parameters

This section describes how to remove the ConText Viewer Cartridge configuration from the Web Request Broker.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Application Server> link.

    The Web Application Server Administration page appears.

3. On the Web Application Server Administration page, select the <Cartridge Administration> link.

   The Cartridge Administration Page appears.

4. On the Cartridge Administration Page, select the <Cartridge Summary (Web Request Broker)> link.

   The Web Request Broker Administration page appears.

5. On the Web Request Broker Administration Page, scroll to the Protecting Applications block of fields.

   Remove all values from the rows that have the following value in the Virtual Path column:

   /ctxwview/

   Click [Modify WRB Configuration].

   The WRB Cartridge Administration page reappears.

Deconfiguring Authorization Server

This section describes how to remove the ConText Viewer Cartridge configuration from the Authorization Server.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Application Server> link.

   The Web Application Server Administration page appears.

3. On the Web Application Server Administration page, select the <Authorization Server> link.

   The Authentication Configuration page appears.

4. On the Authentication Configuration page, select the <Basic> link.

   The Basic Authentication page appears.

5. On the Basic Authentication page, scroll to the Basic Users block of fields.

   Remove all values from the row that has the following value in the User Name column:

   ctxquery

Scroll to the Basic Groups block of fields. Remove all values from the row that has the following value in the Group Name column:

`ctxview_group`

Scroll to the Basic Realms block of fields. Remove all values from the row that has the following value in the Realms column:

`ctxview_realm`

Click [Modify].

The resulting page shows that the Basic Authentication was updated.

The Web Application Server Authentication Configuration reappears.

The deconfiguration of Oracle Web Application Server 3.0 for use with the ConText Viewer Cartridge is now complete.

B

Configuration Manager: Manual Configuration

This appendix documents the various manual procedures for creating the Configuration Manager PL/SQL packages, as well as configuring and deconfiguring Oracle WebServer 2.1 and Oracle Web Application Server 3.0 for use with the Configuration Manager.

> **Note:** You only need to refer to this appendix if configuration of the Configuration Manager via the Oracle Installer failed.
>
> For more information, see "Configuring the Configuration Manager" in Chapter 2, "Administration Tools".

The following topics are covered in this appendix:

- Notational Conventions for Configuration/Deconfiguration
- Manually Creating the PL/SQL Packages
- Configuring Oracle WebServer 2.1 for the Configuration Manager
- Deconfiguring Oracle WebServer 2.1
- Configuring Oracle Web Application Server 3.0 for the Configuration Manager
- Deconfiguring Oracle Web Application Server 3.0

Notational Conventions for Configuration/Deconfiguration

The section provides details for the notational conventions that apply to all of the manual configuration and deconfiguration procedures for Oracle WebServer 2.1 and Oracle Web Application Server 3.0.

Buttons

Buttons referenced appear within square brackets, with capitalization the same as it appears in the browser.

For example: [Modify Listener]

Links

Links appear within angle brackets.

For example: <Web Request Broker>

Pathnames

All paths mentioned in the configuration steps follow the UNIX convention of directory names separated by forward slashes (i.e. '/'). If a path is a physical path (as opposed to a virtual path), and the platform is Windows NT, instead of a UNIX-based platform), then use the backslash character (i.e. '\') to separate directory names.

Manually Creating the PL/SQL Packages

Configuration of the Configuration Manager, which includes creation of PL/SQL packages used by the Configuration Manager, is normally performed through a separate Oracle Installer session after installation of the ConText Workbench.

If the I/O utility (ctxio32), which is called by the Oracle Installer to create the PL/SQL packages, fails to create the packages, you may need to manually create the packages.

A SQL script (CFGPACK.SQL) is provided for this purpose. The script is installed during configuration of the Configuration Manager; however, the script only needs to be run if the normal configuration procedure fails.

The location of the script depends on whether the installation was attempted against WebServer (or Web Application Server) installed on the current (i.e. local) machine or a remote machine.

Location of Files - Local Installation

If the configuration was attempted against WebServer (or Web Application Server) installed on the current machine, the files are located in the following directory:

```
%ORACLE_HOME%\ctxw\middle\cfgmgr\install
```

Location of Files - Remote Installation

If the configuration was attempted against WebServer (or Web Application Server) installed on a remote machine, the files are located in the following directory:

```
$ORACLE_HOME\ctxw\middle\cfgmgr\remote\install
```

Creating the Packages

To create the packages, change directories to the directory in which the packages are installed and use the following command from the command-line prompt:

```
> plus80 ctxsys/passwd@connect_string @cfgpack
```

Configuring Oracle WebServer 2.1 for the Configuration Manager

Configuration of Oracle WebServer 2.1 for use with the ConText Configuration Manager is normally performed through a separate Oracle Installer session after installation of the ConText Workbench.

In the event that the WebServer portion of the configuration fails, manual configuration is necessary to set up WebServer for use with the Configuration Manager.

> **Note:** These manual procedures are different from those that apply for Oracle Web Application Server 3.0.
>
> If Web Application Server is installed, see "Configuring Oracle Web Application Server 3.0 for the Configuration Manager" in this chapter.

Manual configuration is performed through a browser, such as Netscape Navigator or Microsoft Internet Explorer. The browser should support forms and frame capabilities.

Deconfiguration procedures can also be applied to remove the Configuration Manager from WebServer.

> **See Also:** For more information about deconfiguring WebServer to remove the Configuration Manager, see "Deconfiguring Oracle WebServer 2.1" in this chapter.

Configuration Requirements

The following information is required during configuration steps:

- the name of the machine on which WebServer is running

- either the SID or Connect String of the database on which ConText is installed

> **Note:** SQL*Net V2 is required by WebServer to communicate with Oracle8. Ensure that the correct alias has been set up in the appropriate *tnsnames.ora* file (that is, the one under *<ORACLE_HOME>\Network\admin*).

- the password of the CTXSYS user

- the URL of the WebServer Administration Page, usually

  `http://<hostname>:8888/ows-adoc/Intro.html`

- the name of the listener used to access the Configuration Manager

- the port number of the listener used to access the Configuration Manager

- the type of platform that WebServer is installed on (UNIX-based or Windows NT)

- Oracle home directory of the WebServer installation

  > **Note:** In certain steps of the configuration process for WebServer, it is necessary to enter file system paths. Part of the path may be the Oracle home directory, which is represented in the configuration steps as %ORACLE_HOME%. The actual value of ORACLE_ HOME should be used instead.
  >
  > The following example assumes an Oracle home directory of D:\ORANT on Windows NT. If the step asks you to enter the following value in a field:
  >
  > `%ORACLE_HOME%/ctxw/middle/cfgmgr/`
  >
  > then enter the value:
  >
  > `D:\ORANT\ctxw\middle\cfgmgr\`

Configuration Tasks

There are three major tasks in the configuration procedure:

- Creating a DCD

- Configuring Oracle Web Listener

- Configuring Oracle Web Request Broker

  > **Note:** WebServer must be restarted once configuration has completed.

Creating a DCD

This section describes the creation of a new Database Connection Descriptor (DCD) which provides PL/SQL Agent with the required information when connecting to a database. It also installs the WebServer Developer's Toolkit packages to the database. The DCD is used by the ConText Configuration Manager to process administration requests from a browser.

1. Access the WebServer Administration page.

2. On the WebServer Administration page, select the <PL/SQL Agent> link.

   The PL/SQL Agent Administration page appears.

3. On the PL/SQL Agent Administration page, select the <Create New DCD> link.

   The PL/SQL Agent Administration - Create New DCD page appears.

4. On the Create New DCD page, in the DCD Name field, enter:

   ```
   ctx_dba
   ```

   In the PL/SQL Agent Database User, enter:

   ```
   CTXSYS
   ```

   Click the Identified by Password radio button.

   In the PL/SQL Agent User Password and Confirm Password fields, enter the password of the CTXSYS user.

   If the database on which ConText is installed is a local database, enter the ORACLE_SID in the ORACLE_SID field; otherwise, enter a connect string in the SQL*Net V2 Service field.

   In the Authorized Ports field, enter the port number of the listener. This port number is used to access the Configuration Manager.

   Select the Install WebServer Developer's Toolkit PL/SQL Packages checkbox.

   In the DBA Username field, enter:

   ```
   CTXSYS
   ```

   In the Password field, enter the password of the CTXSYS user.

   Click the [Submit New Service] button.

The resulting page shows that the DCD has been created. This page may take some time to appear since packages are installed in the database.

Configuring Oracle Web Listener

This section describes how to configure a Web Listener for use with the Configuration Manager. A physical file-system directory is mapped to a virtual directory to specify a root directory for the Configuration Manager. Security for the virtual directory is also configured.

1. Access the WebServer Administration page.

2. On the WebServer Administration page, select the <Oracle Web Listener> link.

   The Oracle Web Listener Administration page appears.

3. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener used to access the Configuration Manager.

   The OWL Administration - Server Advanced Configuration page appears.

4. On the OWL Administration - Server Advanced Configuration page, scroll down to the Directory Mappings block of fields to insert values into the first empty row.

   In the File-System Directory column, enter the physical path:

   ```
   %ORACLE_HOME%/ctxw/middle/cfgmgr/
   ```

   In the Virtual Directory column, enter:

   ```
   /ctxw_cm/
   ```

   Click [Modify Listener].

   The Oracle Web Listener Administration page appears.

5. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener used to access the Configuration Manager.

   The OWL Administration - Server Advanced Configuration page appears.

6. On the OWL Administration - Server Advanced Configuration page, select <Security: Access Control and Encryption>.

   The Oracle Web Listener Security page appears.

7. On the Oracle Web Listener Security page, scroll to the Basic Users block of fields.

In the User Name column of a blank row, enter:

`ctxsys`

In the Password field, enter the CTXSYS password, scroll to the Basic Groups block of fields.

In the Group Name column, enter:

`ctxdba_group`

In the User(s) column, enter:

`ctxsys`

Scroll to the Basic Realms block of fields. In the Realms column, enter:

`ctxdba_realm`

In the Group(s) column, enter:

`ctxdba_group`

Scroll to the Protection block of fields. In the Virtual Path column, enter:

`/ctx_dba/owa/`

In the Basic/Digest column, select Basic.

In the Realm column, enter:

`ctxdba_realm`

Click [Modify Listener].

The Oracle Web Listener Security page reappears.

Scroll to the end of the page and click [Administration].

Configuring Oracle Web Request Broker

This section describes how to configure the Web Request Broker for a particular Web Listener for use with the Configuration Manager. The PL/SQL Agent is mapped to the virtual path of the Configuration Manager, and security for the PL/SQL Agent is configured.

1. Access the WebServer Administration page.

2. On the WebServer Administration page, select the <Web Request Broker> link.

The Web Request Broker Administration page appears.

3. On the Web Request Broker Administration page, select <Modify> for the listener used to access the Configuration Manager.

The WRB Cartridge Administration Page appears.

4. On the WRB Cartridge Administration Page, scroll to the Applications and Directories block of fields.

In the Virtual Path column, enter:

`/ctx_dba/owa`

In the App. column, enter:

`OWA`

In the Physical Path column, enter:

`%ORACLE_HOME%/ows21/bin`

Scroll to the Protecting Applications block of fields. In the Virtual Path column, enter:

`/ctx_dba/owa/`

In the Basic/Digest column, select Basic.

In the Realm column, enter:

`ctxdba_realm`

Click the [Modify WRB Configuration] button.

The configuration of Oracle WebServer 2.1 for use with the ConText Configuration Manager is now complete.

Before using the Configuration Manager, restart WebServer.

Deconfiguring Oracle WebServer 2.1

This section describes how to remove the Configuration Manager from Oracle WebServer 2.1.

Note: Some steps instruct the user to remove a particular value from a field. This can be done by placing the cursor at the start of the data in the field and holding down the Delete key until all the data has been removed. Alternatively, it may be possible to double-click on a value then press Delete.

For any pull-down menu fields, select a blank entry from the pull-down. If there is no blank entry, select the top entry.

Other steps instruct the user to remove a certain row from a block of fields. This means removing the value from each field in the row.

Deconfiguration Requirements

The following information is required during deconfiguration steps:

- the URL of the WebServer Manager Home Page, usually:

  ```
  http://<hostname>:8888/ows-adoc/Intro.html
  ```

- the name of the listener configured to access the Configuration Manager

---

Note: In certain steps of the deconfiguration process for WebServer, it is necessary to remove file system paths. Part of the path may be the Oracle home directory, which is represented in the configuration steps as %ORACLE_HOME%. The actual value of ORACLE_HOME should be used instead.

The following example assumes an Oracle home directory of D:\ORANT on Windows NT. If the step asks you to remove the following value from a field:

```
%ORACLE_HOME%/ctxw/middle/cfgmgr/
```

then remove the value:

```
D:\ORANT\ctxw\middle\cfgmgr\
```

Deconfiguration Tasks

There are three major tasks in the deconfiguration procedure:

- Removing the DCD
- Deconfiguring Oracle Web Listener
- Deconfiguring Oracle Web Request Broker

> **Note:** WebServer must be restarted once deconfiguration has completed.

Removing the DCD

This section describes how to remove the Database Connection Descriptor (DCD) called CTX_DBA. It does not remove packages from the database account associated with the DCD.

1. Access the WebServer Administration page.

2. On the WebServer Administration page, select the <PL/SQL Agent> link.

The PL/SQL Agent Administration page appears.

3. On the PL/SQL Agent Administration page, select the <CTX_DBA> link.

The Modify DCD (Database Connection Descriptor) page appears.

4. On the Modify DCD (Database Connection Descriptor) page, click the [Delete Service] button.

The Delete DCD (Database Connection Descriptor) page appears.

5. On the Delete DCD (Database Connection Descriptor) page, click the [Confirm Delete] button.

The resulting page shows that the CTX_DBA DCD has been deleted.

Deconfiguring Oracle Web Listener

This section describes how to remove the ConText Configuration Manager configuration from the Web Listener used to access the Configuration Manager.

1. Access the WebServer Administration page.

2. On the WebServer Administration page, select the <Oracle Web Listener> link.

The Oracle Web Listener Administration page appears.

3. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener which is used to access the Configuration Manager.

The OWL Administration - Server Advanced Configuration page appears.

4. On the OWL Administration - Server Advanced Configuration page, scroll down to the Directory Mappings block of fields.

Remove all values from the row that has the following value in the File-System Directory column:

```
%ORACLE_HOME%/ctxw/middle/cfgmgr/
```

Click [Modify Listener].

The Oracle Web Listener Administration page appears.

5. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener which is used to access the Configuration Manager.

The OWL Administration - Server Advanced Configuration page appears.

6. On the OWL Administration - Server Advanced Configuration page, select <Security: Access Control and Encryption>.

The Oracle Web Listener Security page appears.

7. On the Oracle Web Listener Security page, scroll to the Basic Users block of fields.

Remove all values from the row that has the following value in the User Name column:

```
ctxsys
```

Scroll to the Basic Groups block of fields.

Remove all values from the row that has the following value in the Group Name column:

```
ctxdba_group
```

Scroll to the Basic Realms block of fields.

Remove all values from the row that has the following value in the Realms column:

```
ctxdba_realm
```

Scroll to the Protection block of fields.

Remove all values from the row that has the following value in the Virtual Path column:

`/ctx_dba/owa/`

Click [Modify Listener].

The Oracle Web Listener Security page reappears.

Scroll to the end of the page and click [Administration].

Deconfiguring Oracle Web Request Broker

This section describes how to remove the ConText Configuration Manager configuration from the Web Request Broker of the particular Web Listener used to access the Configuration Manager.

1. Access the WebServer Administration page.

2. On the WebServer Administration page, select the <Web Request Broker> link.

The Web Request Broker Administration page appears.

3. On the Web Request Broker Administration page, select <Modify> for the listener which is used to access the Configuration Manager.

The WRB Cartridge Administration Page appears.

4. On the WRB Cartridge Administration Page, scroll to the Applications and Directories block of fields.

Remove all values from the row that has the following value in the Virtual Path column:

`/ctx_dba/owa`

Scroll to the Protecting Applications block of fields. Remove all values from the row that has the following value in the Virtual Path column:

`/ctx_dba/owa/`

Click the [Modify WRB Configuration] button.

The deconfiguration of Oracle WebServer 2.1 for use with the ConText Configuration Manager is now complete.

Configuring Oracle Web Application Server 3.0 for the Configuration Manager

Configuration of Oracle Web Application Server 3.0 for use with the ConText Configuration Manager is normally performed through a separate Oracle Installer session after installation of the ConText Workbench.

In the event that the Web Application Server portion of the configuration fails, manual configuration is necessary to set up Web Application Server for use with the Configuration Manager.

> **Note:** These manual procedures are different from those that apply for Oracle WebServer 2.1.
>
> If WebServer is installed, see "Configuring Oracle WebServer 2.1 for the Configuration Manager" in this chapter.

Manual configuration is performed through a browser, such as Netscape Navigator or Microsoft Internet Explorer. The browser should support forms and frame capabilities.

Deconfiguration procedures can also be applied to remove the Configuration Manager from WebServer.

> **See Also:** For more information about deconfiguring Web Application Server to remove the Configuration Manager, see "Deconfiguring Oracle Web Application Server 3.0" in this chapter.

Configuration Requirements

The following information is required during configuration steps:

- the name of the machine on which Web Application Server is running

- either the SID or Connect String of the database on which ConText is installed

    - SQL*Net V2 is required by Web Application Server to communicate with Oracle8. Ensure that the correct alias has been set up in the appropriate *tnsnames.ora* file (that is, the one under *<ORACLE_ HOME>\Network\admin*).

- the password of the CTXSYS user

- the URL of the Web Application Server Manager Home Page, usually:

  ```
  http://<hostname>:8888/ows-adoc/Intro.html
  ```

- the name of the listener that used to access the Configuration Manager

- the port number of the listener used to access the Configuration Manager

- The type of platform that Web Application Server is installed on (UNIX-based or Windows NT)

- the owner and group of the admin listener

- the Oracle home directory of the Web Application Server installation

> **Note:** In certain steps of the configuration process for Web Application Server, it is necessary to enter file system paths. A portion of the path may be the Oracle or Web Application Server home directory, which is represented in the configuration steps as either %ORACLE_HOME% or %ORAWEB_HOME%.
>
> *Do not* substitute the real values unless explicitly instructed to do so.

Configuration Tasks

There are five major tasks in the configuration procedure:

- Creating a Database Access Descriptor

- Creating a PL/SQL Agent

- Configuring the Oracle Web Listener

- Configuring the Oracle Web Request Broker

- Configuring Authorization Server

> **Note:** Web Application Server must be restarted once configuration has completed.

Creating a Database Access Descriptor

This section describes the creation of a new Database Access Descriptor (DAD) which provides the PL/SQL Agent with information used when connecting to a database. The DAD is used by the Configuration Manager to process administration requests from a browser.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Application Server> link.

   The Web Application Server Administration page appears.

   Select the <DAD Administration> link.

   The Database Access Descriptor Administration page appears.

3. On the Database Access Descriptor Administration page, select the <Create New DAD> link.

   The Database Access Descriptor Creation page appears.

4. On the Database Access Descriptor Creation page, in the DAD Name field, enter:

   ```
   ctx_dba
   ```

   In the Database User, enter:

   ```
   CTXSYS
   ```

   Click the Identified by Password radio button.

   In the Database User Password and Confirm Password fields, enter the password of the CTXSYS user.

   If the database on which ConText is installed is a local database, enter the ORACLE_SID in the ORACLE_SID field; otherwise, enter a connect string in the SQL*Net V2 Service field.

   Click the Store The User Name And Password in the DAD checkbox.

   In the DBA Username field, enter:

   ```
   CTXSYS
   ```

   In the Password field, enter the password of the CTXSYS user.

Click the [Submit New Service] button.

The resulting page shows that the DAD was created.

Creating a PL/SQL Agent

This section describes the creation of a new PL/SQL Agent configuration for use with the Configuration Manager. It also installs the Web Application Server Developer's Toolkit packages to the database.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Application Server> link.

   The Web Application Server Administration page appears.

3. On the Web Application Server Administration page, select the <Cartridge Administration> link.

   The Cartridge Administration page appears.

4. On the Cartridge Administration page, select the <PLSQL Cartridge> link.

   The PL/SQL Agent Administration page appears.

5. On the PL/SQL Agent Administration page, select the <Create New PL/SQL Agent> link.

   The PL/SQL Agent Administration Service Creation page appears.

6. On the PL/SQL Agent Administration Service Creation page, in the Name Of PL/SQL Agent field, enter:

   ```
   ctx_dba
   ```

   In the Name Of DAD field, select:

   ```
   ctx_dba
   ```

   In the Authorized Ports field, enter the port number of the listener used to access the Configuration Manager.

   In the DAD Username field, enter:

   ```
   CTXSYS
   ```

   In the DAD Password field, enter the password of the CTXSYS user.

Select the Install Web Application Server Developer's Toolkit PL/SQL Packages checkbox.

If the database is remote to the Web Application Server host OR if the database is local and admin listener owner is NOT a DBA then:

Enter the CTXSYS and password in the DBA Username and Password fields.

Click the [Submit New Agent] button.

The resulting page shows that the PL/SQL Agent was created. This may take time as packages are installed in the database.

Configuring the Oracle Web Listener

This section describes how to configure a Web Listener for use with the Configuration Manager. A physical filesystem directory is mapped to a virtual directory to specify a root directory for the Configuration Manager. Security for the virtual directory is also configured.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Listener> link.

   The Oracle Web Listener Administration page appears.

3. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener used to access the Configuration Manager.

   The OWL Administration - Server Advanced Configuration page appears.

4. On the OWL Administration - Server Advanced Configuration page, scroll down to the Directory Mappings block of fields to insert values into the first empty row.

   In the File-System Directory column, enter the physical path (substitute real value of the Oracle home directory):

   ```
   %ORACLE_HOME%/ctxw/middle/cfgmgr/
   ```

   In the Virtual Directory column, enter:

   ```
   /ctxw_cm/
   ```

   Click [Modify Listener].

   The Oracle Web Listener Administration page appears.

5. On the OWL Administration page, select <CONFIGURE> for the listener used to access the Configuration Manager.

The OWL Administration - Server Advanced Configuration page appears.

6. On the OWL Administration - Server Advanced Configuration page, select the <Security> link.

The Oracle Web Listener Security page appears.

7. On the Oracle Web Listener Security page, scroll to the Basic Users block of fields.

In the User Name column of a blank row, enter:

ctxsys

In the Password field, enter the CTXSYS password.

Scroll to the Basic Groups block of fields. In the Group Name column, enter:

ctxdba_group

In the User(s) column, enter:

ctxsys

Scroll to the Basic Realms block of fields. In the Realms column, enter:

ctxdba_realm

In the Group(s) column, enter:

ctxdba_group

Scroll to the Protection block of fields. In the Virtual Path column, enter:

/ctx_dba/owa/

In the Basic/Digest column, select Basic.

In the Realm column, enter:

ctxdba_realm

Click [Modify Listener].

The Oracle Web Listener Security page reappears.

Scroll to the end of the page and click [Administration].

Configuring the Oracle Web Request Broker

This section describes how to configure the Web Request Broker for a particular Web Listener for use with the Configuration Manager. The PL/SQL Agent is mapped to the virtual path of the Configuration Manager, and security for the PL/SQL Agent is configured.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Application Server> link.

   The Web Application Server Administration page appears.

3. On the Web Application Server Administration page, select the <Cartridge Administration> link.

   The Cartridge Administration Page appears.

   Select the <Cartridge Summary (Web Request Broker)> link.

   The Web Request Broker Administration page appears.

4. On the Web Request Broker Administration page, scroll to the Applications and Directories block of fields.

   In the Virtual Path column, enter:

   ```
   /ctx_dba/owa
   ```

   In the App. column, enter:

   ```
   PLSQL
   ```

   In the Physical Path column, enter:

   ```
   %ORAWEB_HOME%/bin
   ```

   Scroll to the Protecting Applications block of fields. In the Virtual Path column, enter:

   ```
   /ctx_dba/owa/
   ```

   In the Scheme column, select Basic.

   In the Realm column, enter:

   ```
   ctxdba_realm
   ```

Click the [Modify WRB Configuration] button.

The resulting page shows that the update of Web Request Broker was successful.

Configuring Authorization Server

This section describes how to configure the Authorization Server for use with the Configuration Manager. It provides authentication for Web Application Server cartridges to authenticate users before allowing them to use the Configuration Manager.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Application Server> link.

   The Web Application Server Administration page appears.

   Select the <Authorization Server> link.

   The Authentication Configuration page appears.

3. On the Authentication Configuration page, select the <Basic> link.

   The Basic Authentication page appears.

4. On the Basic Authentication page, scroll to the Basic Users block of fields.

   In the User Name column, enter:

   ```
   ctxsys
   ```

   In the Password column, enter the password of the CTXSYS user.

   Scroll to the Basic Groups block of fields. In the Group Name column, enter:

   ```
   ctxdba_group
   ```

   In the User(s) column, enter:

   ```
   ctxsys
   ```

   Scroll to the Basic Realms block of fields. In the Realms column, enter:

   ```
   ctxdba_realm
   ```

In the Group(s) column, enter:

```
ctxdba_group
```

Click the [Modify] button.

The resulting page shows that the Basic Authentication was updated.

The configuration of Oracle Web Application Server 3.0 for use with the Configuration Manager is now complete.

Deconfiguring Oracle Web Application Server 3.0

This section describes how to remove the Configuration Manager configurations from Oracle Web Application Server 3.0.

> **Note:** Some steps instruct the user to remove a particular value from a field. This can be done by placing the cursor at the start of the data in the field and holding down the Delete key until all the data has been removed. Alternatively, it may be possible to double-click on a value then press Delete.
>
> For any pull-down menu fields, select a blank entry from the pull-down. If there is no blank entry, select the top entry.
>
> Other steps instruct the user to remove a certain row from a block of fields. This means removing the value from each field in the row.

Deconfiguration Requirements

The following information is required during deconfiguration steps:

- the URL of the Web Application Server Manager Home Page, usually:

  ```
  http://<hostname>:8888/ows-adoc/Intro.html
  ```

- the name of the listener configured to access the Configuration Manager.

Deconfiguration Tasks

There are five major tasks in the deconfiguration procedure:

- Deleting the Database Access Descriptor
- Deleting the PL/SQL Agent
- Deconfiguring the Oracle Web Listener
- Deconfiguring the Oracle Web Request Broker
- Deconfiguring Authorization Server

> **Note:** Web Application Server must be restarted once configuration has completed.

Deleting the Database Access Descriptor

This section describes how to remove the Database Access Descriptor (DAD) called ctx_dba.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Application Server> link.

   The Web Application Server Administration page appears.

   Select the <DAD Administration> link.

   The Database Access Descriptor Administration page appears.

3. On the Database Access Descriptor Administration page, select the <ctx_dba> link.

   The Modify DAD (Database Access Descriptor) page appears.

4. On the Modify DAD (Database Access Descriptor) page, click the [Delete DAD] button.

   The Delete DAD (Database Access Descriptor) page appears.

   Click the [Confirm Delete] button.

   The resulting page shows that the ctx_dba DAD was deleted.

Deleting the PL/SQL Agent

This section describes how to delete the PL/SQL Cartridge Agent used by the Configuration Manager.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Application Server> link.

   The Web Application Server Administration page appears.

3. On the Web Application Server Administration page, select the <Cartridge Administration> link.

   The Cartridge Administration page appears.

4. On the Cartridge Administration page, select the <PLSQL Cartridge> link.

   The PL/SQL Agent Administration page appears.

5. On the PL/SQL Agent Administration page, select the <Create New PL/SQL Agent> link.

   The PL/SQL Agent Administration Service Creation page appears.

6. On the PL/SQL Agent Administration Service Creation page, select the <ctx_dba> link.

   The Modify PL/SQL Cartridge Agent (PLSQL) page appears.

7. On the Modify PL/SQL Cartridge Agent (PLSQL) page, click the [Delete Agent] button.

   The Delete PL/SQL Agent (PLSQL) page appears.

8. On the Delete PL/SQL Agent (PLSQL) page, click [Confirm Delete].

   The resulting page shows that the ctx_dba PL/SQL Agent has been deleted.

Deconfiguring the Oracle Web Listener

This section describes how to remove the ConText Configuration Manager configuration from the Web Listener used to access the Configuration Manager.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Listener> link.

   The Oracle Web Listener Administration page appears.

3. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener which is used to access the Configuration Manager.

   The OWL Administration - Server Advanced Configuration page appears.

4. On the OWL Administration - Server Advanced Configuration page, scroll down to the Directory Mappings block of fields.

   Remove all values from the row that has the following value in the File-System Directory column:

   ```
   %ORACLE_HOME%/ctxw/middle/cfgmgr/
   ```

   Click [Modify Listener].

   The Oracle Web Listener Administration page appears.

5. On the Oracle Web Listener (OWL) Administration page, select <CONFIGURE> for the listener which is used to access the Configuration Manager.

   The OWL Administration - Server Advanced Configuration page appears.

6. On the OWL Administration - Server Advanced Configuration page, select <Security>.

   The Oracle Web Listener Security page appears.

7. On the Oracle Web Listener Security page, scroll to the Basic Users block of fields.

   Remove all values from the row that has the following value in the User Name column:

   ctxsys

   Scroll to the Basic Groups block of fields.

   Remove all values from the row that has the following value in the Group Name column:

   ctxdba_group

   Scroll to the Basic Realms block of fields.

   Remove all values from the row that has the following value in the Realms column:

   ctxdba_realm

   Scroll to the Protection block of fields.

   Remove all values from the row that has the following value in the Virtual Path column:

   /ctx_dba/owa/

   Click [Modify Listener].

   The Oracle Web Listener Security page reappears.

   Scroll to the end of the page and click [Administration].

Deconfiguring the Oracle Web Request Broker

This section describes how to remove the ConText Configuration Manager configuration from the Web Request Broker.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Cartridge Administration> link.

    The Cartridge Administration page appears.

3. On the Cartridge Administration page, select the <Cartridge Summary (Web Request Broker)> link.

    The Web Request Broker Administration page appears.

4. On the Web Request Broker Administration page, scroll to the Applications and Directories block of fields.

    Remove all values from the row that has the following value in the Virtual Path column:

    `/ctx_dba/owa`

    Scroll to the Protecting Applications block of fields.

    Remove all values from the row that has the following value in the Virtual Path column:

    `/ctx_dba/owa/`

    Click the [Modify WRB Configuration] button.

    The WRB Cartridge Administration page reappears.

Deconfiguring Authorization Server

This section describes how to remove the Configuration Manager configuration from the Authorization Server.

1. Access the Web Application Server Administration page.

2. On the Web Application Server Administration page, select the <Oracle Web Application Server> link.

    The Web Application Server Administration page appears.

    Select the <Authorization Server> link.

    The Authentication Configuration page appears.

3. On the Authentication Configuration page, select the <Basic> link.

The Basic Authentication page appears.

4. On the Basic Authentication page, scroll to the Basic Users block of fields.

   Remove all values from the row that has the following value in the User Name column:

   ```
   ctxsys
   ```

   Scroll to the Basic Groups block of fields.

   Remove all values from the row that has the following value in the Group Name column:

   ```
   ctxdba_group
   ```

   Scroll to the Basic Realms block of fields.

   Remove all values from the row that has the following value in the Realms column:

   ```
   ctxdba_realm
   ```

   Click the [Modify] button.

   The Web Application Server Authentication Configuration appears.

   The deconfiguration of Oracle Web Application Server 3.0 for use with the ConText Configuration Manager is now complete.

Index