

# Oracle9iAS Discoverer™ Plus and Viewer

Configuration Guide

Release 4.1 for UNIX

April 2001

Part No. A90288-01

**ORACLE®**

Part No. A90288-01

Copyright © 2001, Oracle Corporation. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

**Restricted Rights Notice** Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle and SQL\*Plus are registered trademarks of Oracle Corporation, Redwood City, California. Oracle Discoverer, Discoverer, Oracle Discoverer/2000, End User Layer, are trademarks of Oracle Corporation, Redwood City, California.

Perspective for Java licensed from Three D Graphics, Inc. Copyright 2000 Three D Graphics, Inc.

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

---

---

# Contents

<b>Send Us Your Comments .....</b>	<b>xi</b>
<b>Preface.....</b>	<b>xiii</b>
<b>1 Overview of Oracle9iAS Discoverer</b>	
1.1 Oracle9iAS Discoverer and Oracle9i Application Server .....	1-2
1.2 The Oracle9iAS Discoverer architecture .....	1-2
1.2.1 The three tier architecture of Discoverer Plus.....	1-3
1.2.2 The n-tier architecture of Discoverer Viewer.....	1-4
1.2.3 The Client tier (Discoverer Plus and Viewer) .....	1-5
1.2.3.1 The Client tier and Discoverer Plus .....	1-5
1.2.3.2 The Client tier and Discoverer Viewer .....	1-5
1.2.4 The HTTP Server tier (Discoverer Viewer only) .....	1-5
1.2.4.1 What is a servlet? .....	1-5
1.2.4.2 What is a servlet engine? .....	1-5
1.2.4.3 What is the Discoverer Viewer Servlet? .....	1-6
1.2.5 The Discoverer Services tier (Discoverer Plus and Viewer) .....	1-6
1.2.5.1 The Discoverer Java Applet and HTML Pages (Discoverer Plus only) .....	1-6
1.2.5.2 The Session Component (Discoverer Plus and Viewer).....	1-7
1.2.5.3 The Locator Component (Discoverer Plus and Viewer) .....	1-7
1.2.5.4 The Preferences Component (Discoverer Plus and Viewer) .....	1-8
1.2.6 The Database Tier (Discoverer Plus and Viewer).....	1-8
1.2.7 Which server machines to install the Discoverer Services components on?.....	1-8
1.2.7.1 The HTTP Server machine.....	1-9
1.2.7.2 The Application Server machine(s) .....	1-9
1.2.7.3 The Master Discoverer Server machine .....	1-9
1.3 How Oracle9iAS Discoverer works.....	1-10
1.3.1 The Discoverer Plus process.....	1-10
1.3.2 The Discoverer Viewer process.....	1-11
1.3.2.1 How the Discoverer Viewer Servlet interprets the HTTP request .....	1-12

1.4	Scalability and load balancing.....	1-12
-----	-------------------------------------	------

## 2 Running, Maintaining and Supporting Oracle9iAS Discoverer

2.1	Running Oracle9iAS Discoverer .....	2-1
2.1.1	Running Discoverer Plus .....	2-2
2.1.2	Running Discoverer Viewer .....	2-6
2.2	Maintaining Oracle9iAS Discoverer .....	2-7
2.3	Registering and Unregistering Discoverer Services Components .....	2-8
2.3.1	To make sure the OAD is running:.....	2-8
2.3.2	To see the registered components on one machine:.....	2-8
2.3.3	To see all the registered components on the network: .....	2-9
2.3.4	Registering and Unregistering from the Command Line .....	2-9
2.3.4.1	To Register the Preferences and Session components with the OAD.....	2-9
2.3.4.2	To Unregister the Preferences and Session components from the OAD .....	2-10
2.4	Editing the End User Preferences .....	2-10
2.4.1	What are the End User Preferences? .....	2-10
2.4.2	To set End User Preferences: .....	2-11
2.5	Shutting Down.....	2-12
2.5.1	Shutting Down the Entire Discoverer System .....	2-12
2.5.1.1	To Shut Down the Discoverer System Gradually .....	2-12
2.5.1.2	To Shut Down the Discoverer System Immediately.....	2-13
2.6	Restarting the Discoverer system.....	2-13
2.7	Adding and Removing Servers.....	2-14
2.7.1	Adding an Additional Discoverer Server machine.....	2-14
2.7.2	To Permanently Remove an Additional Discoverer Server Machine .....	2-14
2.8	Utilizing the Discoverer Viewer JavaScript Enhanced User Interface .....	2-16
2.9	Frequently Asked Questions .....	2-17

## 3 Installing Oracle9iAS Discoverer on multiple machines

3.1	What are the differences between a default installation and a distributed installation of Oracle9iAS Discoverer?.....	3-1
3.1.1	Default installation of Oracle9iAS Discoverer .....	3-1
3.1.2	Distributed Oracle9iAS Discoverer installation .....	3-2
3.2	Performing a distributed installation .....	3-4
3.2.1	Confirming connection to the default Master Discoverer Server machine .....	3-4

3.2.2	Confirming the name of the Master Discoverer Server machine.....	3-4
3.2.3	Installing Discoverer on Additional Discoverer Server machines.....	3-4
3.2.4	Editing the Discoverer Services pref.txt file (Discoverer Plus and Viewer).....	3-6
3.2.4.1	What is the pref.txt file? .....	3-6
3.2.4.2	Why edit the pref.txt file? .....	3-7
3.2.4.3	What is load balancing? .....	3-7
3.2.4.4	Examples of Load Balancing .....	3-8
3.2.4.5	How to edit the pref.txt file .....	3-9
3.2.5	Editing the tnsnames.ora file (Discoverer Plus and Viewer).....	3-11
3.2.5.1	What is the tnsnames.ora file?.....	3-11
3.2.5.2	Why edit the tnsnames.ora file?.....	3-11
3.2.5.3	How to copy the tnsnames.ora file.....	3-12
3.2.5.4	How to edit the tnsnames.ora file.....	3-12
3.2.6	Editing the disco4iv.xml file (Discoverer Viewer only) .....	3-13
3.2.6.1	What is the disco4iv.xml file?.....	3-13
3.2.6.2	Why edit the disco4iv.xml file?.....	3-13
3.2.6.3	How to edit the disco4iv.xml file and what to change.....	3-13
3.2.7	Confirming connection to the Additional Discoverer Server.....	3-14

## 4 Using a URL to start Oracle9iAS Automatically

4.1	What is the format of the URL to start Oracle9i AS Discoverer? .....	4-1
4.2	Discoverer Plus URL arguments.....	4-3
4.3	Discoverer Viewer URL arguments.....	4-6

## 5 Customizing Discoverer Viewer

5.1	Using Discoverer Viewer for Customized Web Applications.....	5-1
5.1.1	Customize the general appearance.....	5-1
5.1.2	Customize the application .....	5-1
5.1.3	How the HTML is produced .....	5-2
5.1.4	Steps to Generate the HTML needed to build a specific Discoverer Application	5-3
5.1.4.1	Step 1 Browser sends URL.....	5-3
5.1.4.2	Step 2 XML generation.....	5-3
5.1.4.3	Step 3 Apply XSL .....	5-4
5.1.4.4	Step 4 Generate HTML.....	5-5
5.2	Customization Examples.....	5-6

5.2.1	Files needed for customization .....	5-6
5.2.2	Quick Customization.....	5-6
5.2.2.1	Quick customization attributes.....	5-7
5.2.2.2	Quick customization steps.....	5-7
5.2.2.3	Example of editing the style.xml file.....	5-8
5.2.3	Full Customization.....	5-9
5.2.3.1	Background.....	5-9
5.2.3.2	Prerequisites .....	5-10
5.2.3.3	File Structure.....	5-10
5.2.3.3.1	disco4iv.xml.....	5-10
5.2.3.3.2	page_layouts.xml.....	5-11
5.2.3.3.3	gui_components.xml.....	5-11
5.2.3.3.4	style.xml.....	5-12
5.2.3.3.5	functions.xml.....	5-12
5.2.3.3.6	errors.xml.....	5-12
5.2.3.3.7	scripts.xml.....	5-12
5.2.3.3.8	render_table.xml .....	5-12
5.2.3.3.9	viewer_config.xml .....	5-12
5.2.3.4	Servlet Programming Interface.....	5-13
5.2.3.4.1	Receiving data from the Servlet.....	5-13
5.2.3.4.2	Sending data to the Servlet.....	5-13
5.2.3.5	Customizing Look and Feel - XML/XSL Example .....	5-13
5.2.3.5.1	Step 1 - The XML file .....	5-14
5.2.3.5.2	Step 2 - The XSL file.....	5-15
5.2.3.5.3	Step 3 - XML+XSL = HTML .....	5-15
5.2.3.5.4	Step 4 - Customize the XSL .....	5-17
5.2.4	Video Stores Demo - an example of an alternative user interface .....	5-20
5.2.4.1	What is installed.....	5-20
5.2.4.2	Set up steps .....	5-21
5.2.4.3	Set up Web Server.....	5-21
5.2.4.4	Run the Video Stores demo .....	5-21
5.2.4.5	Modify the video.xml file .....	5-22
5.2.5	Other customization examples.....	5-23
5.2.5.1	Logging on automatically.....	5-23
5.2.5.1.1	Step 1: Add a new template to page_layouts.xml .....	5-23

5.2.5.1.2	Step 2: Include username, password and database .....	5-23
5.2.5.1.3	Step 3: Modify disco4iv.xml .....	5-24
5.2.6	Customization Summary .....	5-24

## 6 SSL Support in Oracle9iAS Discoverer

6.1	Getting more information .....	6-1
6.2	SSL and Discoverer .....	6-2
6.3	What is SSL and why should I use it? .....	6-2
6.4	How does SSL work? .....	6-2
6.5	About Public Key Encryption Technology .....	6-3
6.6	SSL authentication and certificates .....	6-3
6.7	What steps do I need to follow? .....	6-4
6.7.1	To configure SSL with Discoverer Plus .....	6-4
6.7.2	To configure SSL with Discoverer Viewer .....	6-4
6.8	Configuring Discoverer Plus to use SSL .....	6-5
6.8.1	How does Discoverer Plus support SSL? .....	6-5
6.8.2	How do I configure Visibroker Gatekeeper to use SSL? .....	6-5
6.8.3	Installing Visibroker Gatekeeper to work with SSL .....	6-5
6.8.4	Getting SSL certificates .....	6-6
6.8.4.1	Specifying the certificate information .....	6-8
6.8.4.2	Specifying private key passwords .....	6-8
6.8.4.3	Specifying output file name .....	6-9
6.8.4.4	Specifying key size .....	6-9
6.8.4.5	Using an S/MIME certificate chain .....	6-9
6.8.5	Installing SSL certificates .....	6-10
6.8.6	Installing the x509cert.jar file .....	6-12
6.8.7	Installing the required shared library files .....	6-12
6.8.8	Enabling SSL in Discoverer Plus start pages .....	6-13
6.8.9	Using SSL inside firewalls .....	6-14
6.9	Configuring Discoverer Viewer using HTTPS .....	6-15
6.9.1	Editing the httpd.conf file to set up HTTPS .....	6-15
6.9.2	Editing the disco4iviewer.properties file to set up HTTPS .....	6-16
6.9.3	Editing the disco4iv.xml file to set up HTTPS .....	6-16
6.9.4	Editing the viewer_config.xml file to set up HTTPS .....	6-17
6.9.5	Enabling SSL in Discoverer Viewer .....	6-17

## 7 Configuring Discoverer Plus for Firewall Support

7.1	Terminology .....	7-2
7.2	Getting more information .....	7-2
7.3	Before you start.....	7-2
7.4	About Internet Firewalls .....	7-3
7.4.1	What is an Internet Firewall?.....	7-3
7.4.2	What is a Demilitarized Zone?.....	7-3
7.4.3	About secure ports.....	7-4
7.5	About Visibroker Gatekeeper.....	7-5
7.5.1	What is Visibroker Gatekeeper?.....	7-5
7.5.2	Why does Discoverer Plus need Visibroker Gatekeeper? .....	7-5
7.5.3	Required Visibroker Gatekeeper Versions.....	7-6
7.6	Choosing a Gatekeeper configuration for Discoverer .....	7-6
7.6.1	About firewall configurations.....	7-6
7.6.2	DMZ configuration examples .....	7-7
7.6.3	Which Gatekeeper configuration do I choose? .....	7-8
7.7	Using Visibroker Gatekeeper with an IIO Proxying Configuration.....	7-9
7.7.1	Overview .....	7-9
7.7.2	Configuring Gatekeeper with no firewall separating Discoverer Services .....	7-10
7.7.3	Configuring Gatekeeper with Discoverer Server components behind an internal firewall .....	7-11
7.8	Using Visibroker with a HTTP Tunnelling Configuration .....	7-12
7.8.1	Overview .....	7-12
7.8.2	Configuring Gatekeeper with no firewall separating Discoverer Services .....	7-14
7.8.3	Configuring Gatekeeper with Discoverer server components behind an internal firewall .....	7-14
7.8.4	Persistent HTTP Connections.....	7-16
7.9	Configuring the Discoverer client.....	7-16
7.9.1	About the Discoverer client connection sequence.....	7-17
7.9.2	How to use a specific connection method .....	7-17
7.10	Configuring Visibroker Gatekeeper .....	7-19
7.10.1	Where to run Visibroker Gatekeeper .....	7-20
7.10.2	Running Visibroker Gatekeeper on the HTTP Server .....	7-20
7.10.3	Running Visibroker Gatekeeper on a different Server .....	7-21
7.10.4	Configuring the Visibroker Gatekeeper port.....	7-21



7.10.5	Changing the default Visibroker Gatekeeper port.....	7-22
7.10.6	Getting Log Information from Visibroker Gatekeeper.....	7-23
7.10.7	Configuring NAT devices to work in front of the Gatekeeper.....	7-24

## **A Oracle9iAS Discoverer and Oracle Applications**

A.1	Specifying display of the Applications User Checkbox in the Discoverer Viewer Connect Dialog.....	A-1
A.2	Specifying display of the Applications User Checkbox in the Discoverer Plus Connect Dialog.....	A-2
A.3	Settings in pref.txt for Oracle Applications Users.....	A-3

## **B Discoverer Viewer Configuration Files**

B.1	Configuration settings in viewer_config.xml.....	B-1
B.2	Configuration settings in disco4iv.xml.....	B-3

## **C Oracle9iAS Discoverer Preference Settings**

C.1	Preference settings in pref.txt.....	C-1
-----	--------------------------------------	-----

## **Index**



---

---

# Send Us Your Comments

## **Oracle9iAS Discoverer Plus and Viewer Configuration Guide for UNIX**

**Part No. A90288-01**

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

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

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: [discdoc\\_us@oracle.com](mailto:discdoc_us@oracle.com)
- FAX: 650 506 7421 Attn: Oracle9iAS Discoverer
- Postal service:  
Oracle Corporation  
Oracle9iAS Discoverer Documentation  
500 Oracle Parkway, 2OP10  
Redwood Shores, CA 94065  
USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.



---

# Preface

Welcome to Oracle9iAS Discoverer!

This guide explains how to configure and customize Discoverer Plus and Discoverer Viewer after they have been installed as part of Oracle9i Application Server.

This guide does not explain how to install Oracle9iAS Discoverer. For information about installation, refer to the *Oracle9i Application Server Installation Guide*.

To use this guide effectively, we recommend you are already familiar with HTTP servers and database concepts.

For the latest information, please read the Oracle9iAS Discoverer Release Notes in conjunction with this guide.

## Intended Audience

This guide is intended for the Discoverer Administrator.

## Conventions

In this manual:

- EUL, the End User Layer™, refers to the metadata interface between the database and Discoverer.
- <iSUITES\_HOME> refers to the directory where the Oracle HTTP Server (powered by Apache) and the JServ servlet engine software are installed, along with the Oracle RDBMS. For example, <iSUITES\_HOME> might be /usr/home/oracle/ias10210.
- <ORACLE\_806\_HOME> refers to the directory where products (e.g. Oracle9iAS Discoverer) that depend on Oracle 8.0.6 RSFs (Required Support Files) are installed. If no default Oracle Home exists, <ORACLE\_806\_HOME> is created as a subdirectory of <iSUITES\_HOME>. For example, <ORACLE\_806\_HOME> might be /usr/home/oracle/ias10210/6iserver.

Other conventions used in this manual are shown in the table below:

Convention	Meaning
. . . ...	Vertical ellipses in examples indicate omitted information not necessary for the example.
...	Horizontal ellipses in statements or commands indicate omitted material not necessary for the statement or command.
<b>boldface text</b>	Boldface type in text indicates a command name, menu name, file name, keyboard keys, or other choice or selection.
<i>italic text</i>	Italic type indicates a term defined in the text, the glossary, or in both locations. It can also represent a variable, and be used to provide emphasis.
< >	Angle brackets enclose user-supplied names.
[ ]	Brackets enclose optional clauses from which you can choose one or none.
Menu name   Command	Text in this format conveys a sequence of choices, <i>i.e.</i> , choose the menu, then the command under that menu.
<code>Courier text</code>	Text in this format indicates a command line to be typed.

# Oracle9iAS Discoverer Navigation and Accessibility

## **Keyboard Navigation**

Oracle9iAS Discoverer supports standard keyboard navigation. Standard keyboard navigation includes the use of the tab key, mnemonics (using the Alt key and the underlined character), and accelerators (such as Alt+F4 to exit a window).

## **Documentation Accessibility**

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

## **Accessibility of Code Examples in Documentation**

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





---

---

# Overview of Oracle9iAS Discoverer

Oracle9iAS Discoverer is a business intelligence tool for analyzing data. Using Discoverer's award-winning user interface, users can access and analyze database data.

There are two Oracle9iAS Discoverer products:

- Discoverer Plus is the Internet version of the award-winning product, Discoverer Desktop Edition. With Discoverer, business professionals can get and analyze data in a company's database without having to understand difficult database concepts. Using Wizard dialogs and menus, Discoverer guides them through the steps to get and analyze data to support their business decisions.
- Discoverer Viewer is a tool for viewing workbooks created by users of Discoverer Plus and Discoverer Desktop Edition. Discoverer Viewer can also be used to integrate database output into a web site and portal and is easily customized to conform to a particular web site look and feel, or to build custom Discoverer applications for the web. Discoverer Viewer is also optimized for performance and designed to minimize network traffic.

## 1.1 Oracle9iAS Discoverer and Oracle9i Application Server

Oracle9iAS Discoverer is installed as part of Oracle9i Application Server. The default installation is to install all Discoverer components on a single machine. However, you can install different components on different machines to improve performance (for more information, refer to [Chapter 3, "Installing Oracle9iAS Discoverer on multiple machines"](#)).

## 1.2 The Oracle9iAS Discoverer architecture

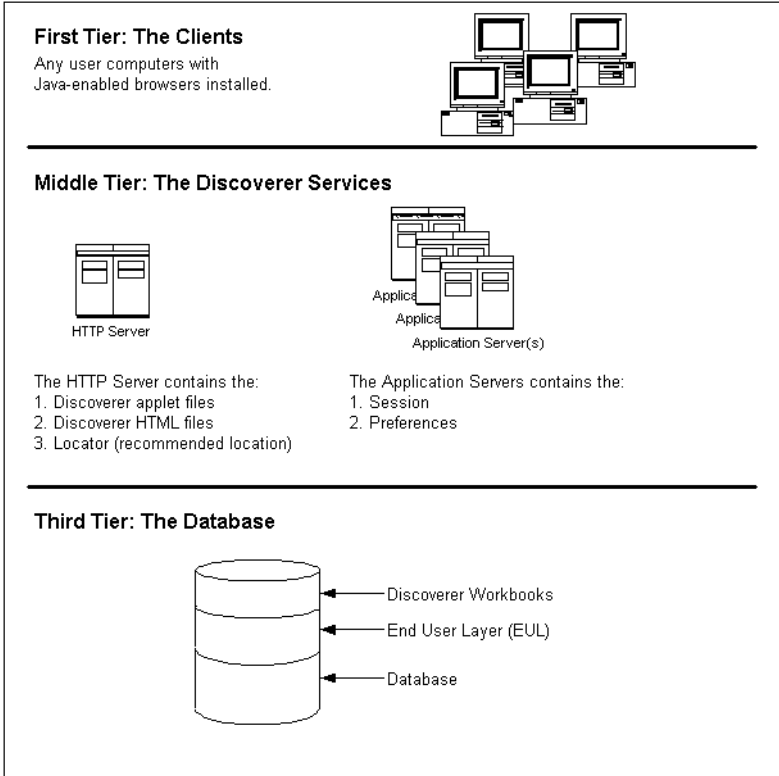
Oracle9iAS Discoverer has a multi-tiered architecture. This architecture takes advantage of the distributed nature of the Web environment. While it is possible to install all tiers of the Oracle9iAS Discoverer architecture on the same machine, we recommend distributing your installation over multiple machines to maximize performance.

### 1.2.1 The three tier architecture of Discoverer Plus

Discoverer Plus has a three tier architecture:

- Client
- Discoverer Services
- Database

Figure 1–1 Three tier architecture of Discoverer Plus

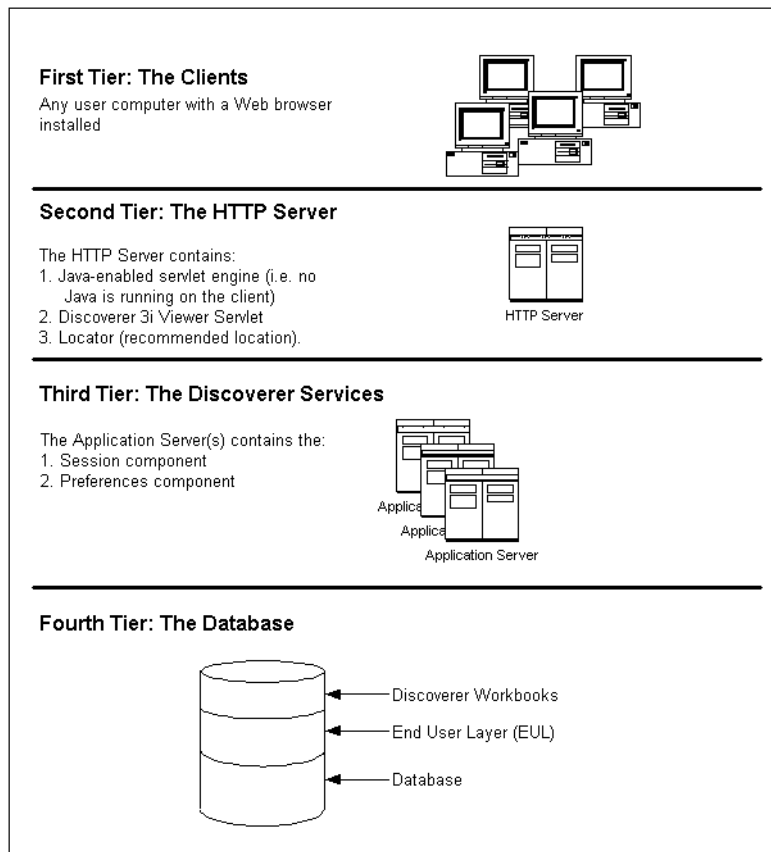


## 1.2.2 The n-tier architecture of Discoverer Viewer

Discoverer Viewer has an n-tier architecture:

- Client
- HTTP Server
- Discoverer Services
- Database

**Figure 1–2 N-tier architecture of Discoverer Viewer**



## 1.2.3 The Client tier (Discoverer Plus and Viewer)

The first tier of the Oracle9iAS Discoverer architecture is the client—a user's computer.

There is no setup or installation to do on the client machine, nor does this machine need to have access to any database.

### 1.2.3.1 The Client tier and Discoverer Plus

In the case of Discoverer Plus, the only requirement for the client machine is that it runs a Java-enabled Web browser (e.g. Microsoft Internet Explorer, or Netscape Navigator with Oracle JInitiator). By accessing the URL that you (the Discoverer administrator) provide, users can log on and run Discoverer to analyze data.

### 1.2.3.2 The Client tier and Discoverer Viewer

In the case of Discoverer Viewer, the only requirement for the client machine is that it can run HTML through a web browser (e.g. Microsoft Internet Explorer or Netscape Navigator). By accessing the URL that you (the Discoverer Administrator) provide, users can connect and run Discoverer Viewer to view data.

## 1.2.4 The HTTP Server tier (Discoverer Viewer only)

For Discoverer Viewer, the HTTP Server tier (and specifically the HTTP Server's servlet engine) is where the Discoverer Viewer servlet is installed. Running the Java-enabled servlet engine on the HTTP Server tier removes the need for Java to run on the client tier.

Note that in a multiple machine installation, a component from the Discoverer Services tier (the Locator component) is usually installed on the same machine as the HTTP Server tier.

### 1.2.4.1 What is a servlet?

A servlet comprises modules of Java code that run on a server machine to answer HTTP requests from client machines. Using a servlet minimizes client-side processing.

### 1.2.4.2 What is a servlet engine?

A servlet engine comes as part of (or as a plug-in to) the HTTP Server software and is the environment in which servlets run. The servlet engine incorporates a Java Virtual Machine (JVM) and implements the Java Servlet API.

### 1.2.4.3 What is the Discoverer Viewer Servlet?

The Discoverer Viewer Servlet uses an XML/XSL processor to generate HTML pages from HTTP requests.

The Discoverer Viewer Servlet and the XML/XSL processor are installed in the HTTP Server's servlet engine

## 1.2.5 The Discoverer Services tier (Discoverer Plus and Viewer)

The Discoverer Services tier is the portion of the Discoverer architecture that the Discoverer administrator installs and maintains.

The Discoverer Services tier consists of the following components:

- the Discoverer Plus Java applet and HTML pages (Discoverer Plus only)
- the Locator component (Discoverer Plus and Viewer)
- the Session component (Discoverer Plus and Viewer)
- the Preferences component (Discoverer Plus and Viewer)

The Discoverer Java applet and HTML pages are installed on an HTTP server machine. The Locator is also usually installed on the HTTP server. The remaining components are installed on one or more application servers. You can install Discoverer Services on one server machine or across many machines.

In addition, the following CORBA (Common Object Request Broker Architecture) components from VisiBroker are installed:

- Object Activation Daemon (OAD)
- SmartAgent

These CORBA components are responsible for activating new components and providing a location service when a user requests that a Discoverer session begin.

### 1.2.5.1 The Discoverer Java Applet and HTML Pages (Discoverer Plus only)

The Discoverer Java applet provides the Discoverer Plus user interface and functionality for analyzing data in the database. The applet runs in a Java-enabled browser on a client machine.

The HTML pages are the Web pages that the user sees when logging on to the URL location for Discoverer Plus.

---

---

**Note:** The first time a user logs on, the Discoverer Java applet is initialized and cached on the user's machine. Later, when the user logs on again, the Discoverer Java applet runs from the local cache and does not need to be downloaded.

---

---

### 1.2.5.2 The Session Component (Discoverer Plus and Viewer)

The Session component provides the link between the client machine (running either Discoverer Plus or Discoverer Viewer) and the database. The Session component contains all of the application logic and performs all of the Discoverer operations such as connecting to the database or opening a workbook.

A machine that has a Session component installed can run multiple concurrent sessions for clients. To increase overall performance, you can install the Session component on many individual machines, and each machine can run multiple concurrent sessions. Distributing the Session component on several machines also improves availability. If one machine is not operating, other machines handle the session requests.

The Session component can run on any server machine (i.e. HTTP server machine or application server machine) in the Discoverer Services configuration that runs a version of UNIX to which Discoverer has been ported.

### 1.2.5.3 The Locator Component (Discoverer Plus and Viewer)

The purpose of the Locator component is to receive a request for a session from the client machine (running either Discoverer Plus or Discoverer Viewer), find the next available application server machine, start a new session, and return a reference for the new session back to the client. Once the client has received this reference, the client and the session communicate directly and the Locator waits for the next request. Install the Locator component on the HTTP Server machine (i.e. the machine running the HTTP Server software). You need only one Locator component on the network for the Discoverer Services.

This structure means that, after a session begins, the HTTP Server machine is no longer involved with the communication between the client and the session. A major advantage of this structure is that no additional traffic is routed through the HTTP Server machine after the session has started, so the performance of the HTTP Server machine is not affected as the number of sessions increases. Another benefit is that if the HTTP Server machine or the Locator component fails, the user session

is not interrupted because the client is communicating directly with the application server.

The Locator component also helps balance the load on the available application servers because it determines which application server will start each requested session. You can specify how the Locator allocates sessions to achieve the best performance.

#### 1.2.5.4 The Preferences Component (Discoverer Plus and Viewer)

The Preferences component provides a single location for preference settings for all users of Oracle9iAS Discoverer (both Discoverer Plus and Discoverer Viewer). The Discoverer Services rely on stored preference settings to dictate certain aspects of behavior. You install only one Preferences component for the Discoverer Services.

Having a single Preferences component is important in a distributed environment where different components can run on different machines. The Preferences component provides consistent preference settings for all components regardless of where they are running. The Preferences component must be installed on a server running a version of UNIX to which Discoverer has been ported.

### 1.2.6 The Database Tier (Discoverer Plus and Viewer)

The database tier of the Discoverer architecture contains:

- the data that users want
- the End User Layer (EUL) that provides an easy to understand view of the data
- the Discoverer workbooks that users can work with to view and analyze the data

You use the Discoverer Administration Edition to create and maintain the EUL.

---

---

**Note:** Before users can use Discoverer to analyze data, the database **must** contain a Discoverer Release 4.1 EUL. You must create or update the EUL with Oracle Discoverer Administration Edition Release 4.1 or later.

---

---

### 1.2.7 Which server machines to install the Discoverer Services components on?

The Discoverer Services software components must each be installed on a server machine. There are three types of server machine that implement the Discoverer architecture:



- the HTTP Server machine
- the Application Server machine(s)
- the Master Discoverer Server machine

These server machines need not be physically different machines. In fact, in a default Oracle9i Application Server installation the HTTP Server machine, the Application Server machine, and the Master Discoverer Server machine are all the same machine.

### 1.2.7.1 The HTTP Server machine

The HTTP Server machine is a machine running HTTP Server software on which you install the Locator component. You only need one HTTP Server machine as part of your Discoverer environment.

If you install the different Discoverer Services software components on a single machine, that machine is both the HTTP Server machine and the Application Server machine.

### 1.2.7.2 The Application Server machine(s)

The Application Server machine is the machine on which you install the Preferences component and the Session component. You can have one Application Server machine or many Application Server machines as part of your Discoverer environment.

If you install the different Discoverer Services software components on a single machine, that machine is both the Application Server machine and the HTTP Server machine.

If you distribute the different Discoverer Services software components across several machines, you must install the Session component (but not the Preferences component) on every machine. In this scenario, each machine is an Application Server and multiple sessions can be started on each machine. Note that the Preferences component is only installed on one Application Server machine (referred to as the Master Discoverer Server machine). The other Application Server machines are referred to as Additional Discoverer machines.

### 1.2.7.3 The Master Discoverer Server machine

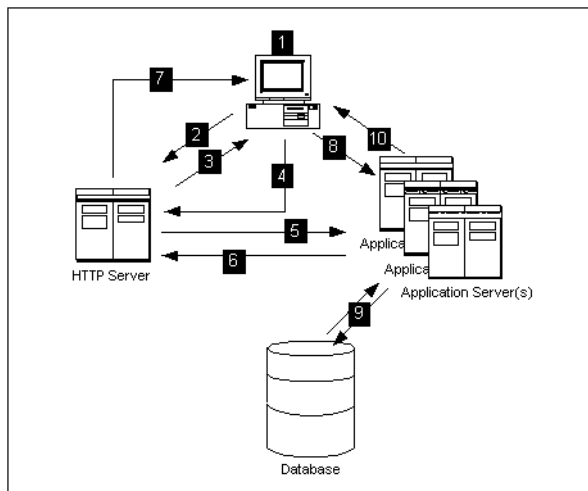
If you install the Session component on multiple Application Server machines, you must designate one of those Application Server machine as the Master Discoverer Server. In addition to the Session component, you must also install the Preferences

component on the Master Discoverer Server. Note that the Master Discoverer Server is the only machine on which you install the Preferences component.

## 1.3 How Oracle9iAS Discoverer works

### 1.3.1 The Discoverer Plus process

*Figure 1–3 The Discoverer Plus Process*

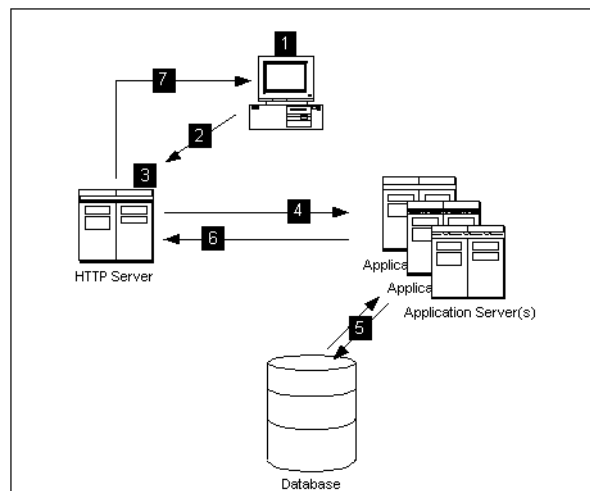


1. The user launches a Web browser on a client computer and types the URL of the HTTP Server.
2. The Web browser accesses the URL on the HTTP Server.
3. The HTTP Server retrieves the HTML pages and the Discoverer applet and transmits them to the client computer.
4. The Discoverer applet finds the Locator (the recommended location for the Locator is on the HTTP Server).
5. The Locator starts a Session component on an Application Server machine and a Discoverer session starts.

6. The Session component creates a reference and passes the reference to the Locator.
7. The Locator passes the reference to the client computer.
8. The client computer uses the reference to contact the Discoverer session directly. The HTTP server is no longer being accessed.
9. The Discoverer session requests and receives data from the database.
10. The Discoverer session transmits data to the Web browser on the client machine.

### 1.3.2 The Discoverer Viewer process

*Figure 1-4 The Discoverer Viewer Process*



1. The user launches a Web browser on a client computer and types the URL of the HTTP Server.
2. The Web browser accesses the URL on the HTTP Server.
3. The Discoverer Viewer Servlet interprets the HTTP request.
4. The Locator starts a Session component on an Application Server machine and a Discoverer session starts.

5. The Discoverer session requests and receives data from the database.
6. The Discoverer session transmits data to the Discoverer Viewer Servlet.
7. The Discoverer Viewer Servlet generates an HTML page and sends it to the browser.

### 1.3.2.1 How the Discoverer Viewer Servlet interprets the HTTP request

The Discoverer Viewer Servlet interprets the HTTP request from the client browser, and makes the necessary calls to the Discoverer Services.

The server response represented in XML is generated by the Servlet and sent to an XML/XSL processor.

The XML/XSL processor combines XML with an XSL configuration file that defines the representation of the User Interface and generates the output (e.g. HTML) to send back to the browser.

The Discoverer Viewer Servlet is composed of two primary layers - the application logic layer and the presentation logic layer.

The application logic layer manages the handling of the HTTP request, the state of the Discoverer Services, and generates the XML representing the logical state of the Discoverer server and the data for a given client.

The presentation logic layer takes the XML and uses XSL to generate a response in a given format (e.g. HTML, Microsoft Excel, XML etc.). The presentation logic can be customized to alter the appearance of the user interface (by editing XSL files). See [Chapter 5, "Customizing Discoverer Viewer"](#) for details.

## 1.4 Scalability and load balancing

Discoverer Plus and Discoverer Viewer are automatically installed to run on the Oracle9i Application Server as a single machine installation. However they are designed to be deployed on multiple machines (for more information, refer to [Chapter 3, "Installing Oracle9iAS Discoverer on multiple machines"](#)).

The Discoverer Services can be replicated so that different sessions can be started on alternate machines to balance the load.

In this way very large numbers of Discoverer users may be supported, since there is no single bottleneck. Multiple Discoverer sessions can run on a single machine, or can be spread across multiple machines as required.

The Discoverer Viewer Servlet can also be distributed.

---

---

# Running, Maintaining and Supporting Oracle9iAS Discoverer

This chapter explains how to run, maintain and support Oracle9iAS Discoverer.

The topics include:

- [Running Oracle9iAS Discoverer](#)
- [Maintaining Oracle9iAS Discoverer](#)
- [Registering and Unregistering Discoverer Services Components](#)
- [Editing the End User Preferences](#)
- [Shutting Down](#)
- [Restarting the Discoverer system](#)
- [Adding and Removing Servers](#)
- [Utilizing the Discoverer Viewer JavaScript Enhanced User Interface](#)
- [Frequently Asked Questions](#)

## 2.1 Running Oracle9iAS Discoverer

Once you have completed the Oracle9i Application Server installation (see the *Oracle9i Application Server Installation Guide*), you can run Discoverer Plus immediately.

Before you can run Discoverer Viewer, an XWindow display must be available to the Viewer servlet for its graphing and charting functions. After installation, you must grant XWindow permissions to the host named in the DISPLAY environment variable in the jserv.properties file in the <i>SUITES\_HOME>/Apache/JServ/etc

directory. By default, the DISPLAY variable is the hostname of the machine running Oracle9i Application Server. In order for Apache JServ to function properly, you must log in to the console of the host named in the DISPLAY environment variable and issue the following command:

```
> xhost +hostname.domain
```

where *hostname.domain* is the name and domain for the machine on which you installed Oracle9i Application Server.

For example, if you installed Oracle9i Application Server on a machine called myserver.oracle.com, you would type the following:

```
> xhost +myserver.oracle.com
```

Note that you must issue the xhost command each time the DISPLAY host system is restarted.

---

---

**Note:** Before users can use Discoverer to analyze data, the database **must** contain a Discoverer Release 4.1 EUL. You must create or update the EUL with Oracle Discoverer Administration Edition Release 4.1 or later.

---

---

## 2.1.1 Running Discoverer Plus

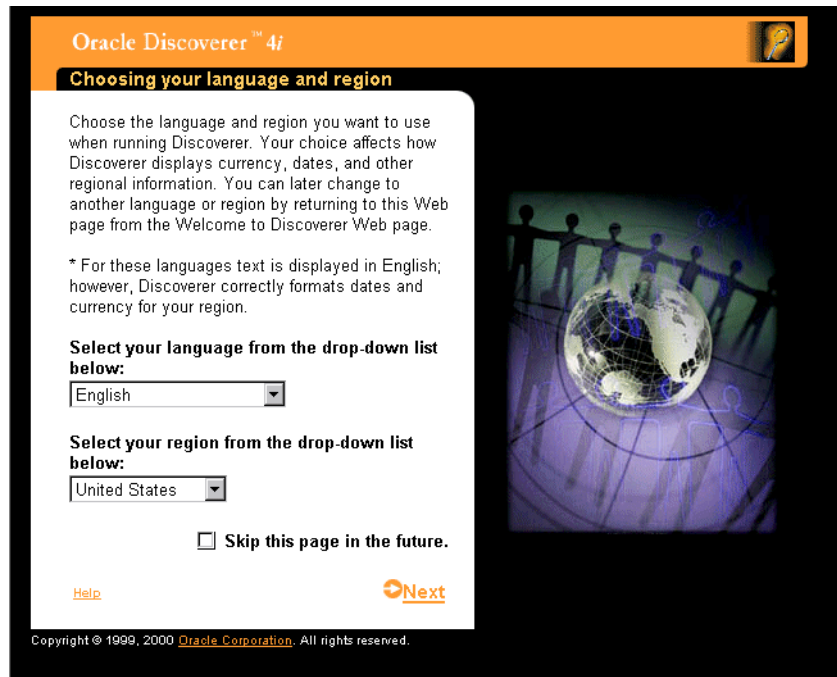
To run Discoverer Plus complete the following steps:

1. Start a browser window and type in your Discoverer Plus URL, using the following as a guide:

```
http://hostname.domain/discwb4/html/english/welcome.htm
```

Where *hostname.domain* represents the Web server machine that the HTTP server is installed on. When specifying a Web server name, you must use its fully qualified host name (including port number if necessary). For example:

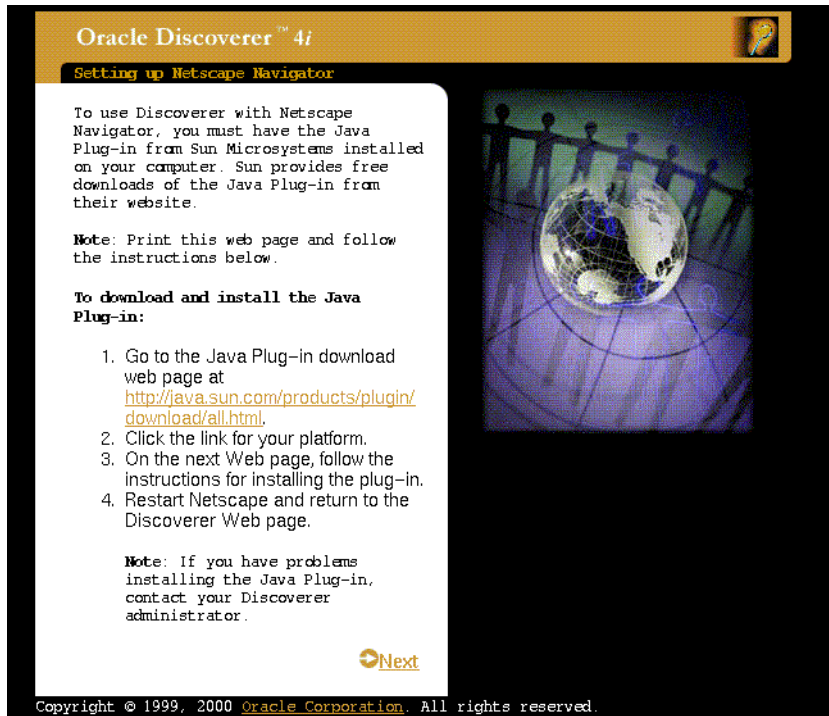
```
http://machinename.oracle.com:7777/discwb4/html/english/welcome.htm
```



## 2. Specify a language and region and click Next.

If you are using a Netscape browser and depending on the software already installed on the client machine, you might be prompted to download and install the Java Plug-in. If you are not prompted to download and install the Java Plug-in, skip the next step.

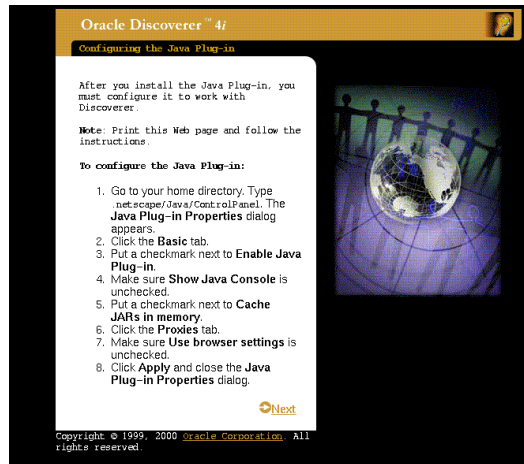
**Note:** The screen below only appears when using a Unix browser.



3. If you see the above screen, follow the instructions to download and install the Java Plug-in.
4. After you install the Java Plug-in, you must configure it to work with Discoverer. Follow the on-screen instructions.

**Note:** The screen below only appears when using a Unix browser.





When you have configured the Java Plug-in (or if you did not have to install and configure the Java Plug-in) the following screen appears:



5. Click on the large image to start Discoverer Plus.

When the Discoverer Plus applet has loaded, the Login Screen appears.



6. Enter your User Name, Password and database connect string.

7. Click Connect.

You are now running Discoverer Plus.

## 2.1.2 Running Discoverer Viewer

To run Discoverer Viewer complete the following steps:

1. Start a browser window and type in your own details using the following as a guide:

`http://hostname.domain/discoverer4i/viewer`

Where *hostname.domain* represents the Web server machine that the HTTP server is installed on. When specifying a Web server name, you must use its fully qualified host name (including port number if necessary). For example:

`http://machinename.oracle.com:7777/discoverer4i/viewer`

This will start Discoverer Viewer and you will see the following login screen:



The image shows the Oracle Discoverer 4i Viewer login interface. At the top, there is a decorative banner with various icons: a magnifying glass, a globe, a network diagram, and a CD-ROM. Below the banner, the Oracle logo is displayed in red, followed by the text "Discoverer™ 4i Viewer" in blue. To the right of the logo, there are four input fields: "User Name", "Password", "Database", and "Language". The "Language" field is a dropdown menu currently set to "English". A "Connect" button is located at the bottom right of the form area.

2. Enter your User Name, Password and database connect string.
3. (Optional) If you are connecting as an Oracle Applications user (to an Oracle Applications database) select the **Oracle Applications User** check box.

If the **Oracle Applications User** check box is not displayed in the connect screen above, refer to [Appendix A.1, "Specifying display of the Applications User Checkbox in the Discoverer Viewer Connect Dialog"](#).

4. Click Connect

You are now running Discoverer Viewer.

## 2.2 Maintaining Oracle9iAS Discoverer

To maintain Oracle9iAS Discoverer, you must first set the PATH and other environment variables. These settings enable you to:

- run Discoverer shell scripts (e.g. startoad.sh, startosagent.sh, startlocator.sh, stopoad.sh, stoposagent.sh, stopall.sh)
  - register and unregister Discoverer Services components
  - run other commands correctly (e.g. osfind, oadutil)
1. From the command prompt, navigate to the <ORACLE\_806\_HOME>/discwb4 directory and run the appropriate script:

- If you are using the Korn shell or the Bourne shell, type the following at the command prompt:

```
> . ./discwb.sh
```

Note that you must separate the periods with a space.

- If you are using the C Shell, type the following at the command prompt:

```
> source discwb.csh
```

## 2.3 Registering and Unregistering Discoverer Services Components

The Discoverer Services components must be registered with the Visibroker OAD (Object Activation Daemon). The components are registered during installation. When you shut down or remove a machine you must unregister Discoverer Services components.

You can view the registered components for one machine or for the entire network.

To register and unregister components (and to view registered components), the VisiBroker OAD must be running.

### 2.3.1 To make sure the OAD is running:

If you are not sure whether the OAD is running:

1. Type the following at the command prompt:

```
> ps -A | grep oad
```

If the OAD process is running, it appears in the list of processes. For example:

```
8634 pts/3 0:01 oad
```

If the OAD is not running:

1. Navigate to the <ORACLE\_806\_HOME>/discwb4/util directory and type the following at the command prompt:

```
> startoad.sh
```

### 2.3.2 To see the registered components on one machine:

1. Make sure the OAD is running.
2. Type the following at the command prompt:

```
> oadutil list
```

---

---

**Note:** Running the oadutil command displays only the list of components registered on the machine. It does not display all the components registered on the network of Discoverer machines.

---

---

### 2.3.3 To see all the registered components on the network:

1. Make sure the OAD is running.
2. Type the following at the command prompt

```
> osfind
```

A list shows all of the components running on the network.

### 2.3.4 Registering and Unregistering from the Command Line

You register and unregister components using shell scripts. The shell scripts are located in the <ORACLE\_806\_HOME>/discwb4/util directory of the machine where the components are installed.

#### 2.3.4.1 To Register the Preferences and Session components with the OAD

The Preferences and Session components must be registered with the OAD. The OAD will start a new session per connection of the client. For your convenience, scripts are available to help you with registering the Preference and Session components with the OAD.

The scripts are named:

- registersession.sh
- registerpreference.sh

To register the Session from the command line:

1. Navigate to the <ORACLE\_806\_HOME>/discwb4/util directory.
2. Type the following at the command prompt:

```
> registersession.sh
```

To register the Preferences from the command line:

1. Navigate to the <ORACLE\_806\_HOME>/discwb4/util directory.

2. Type the following at the command prompt:

```
> registerpreference.sh
```

#### 2.3.4.2 To Unregister the Preferences and Session components from the OAD

You can unregister the Preferences and Session from the OAD.

To unregister the Preferences component from the OAD:

1. Navigate to the <ORACLE\_806\_HOME>/discwb4/util directory.
2. Type the following at the command prompt:

```
> unregisterpreference.sh
```

To unregister the Session component from the OAD:

1. Navigate to the <ORACLE\_806\_HOME>/discwb4/util directory.
2. Type the following at the command prompt:

```
> unregistersession.sh
```

## 2.4 Editing the End User Preferences

### 2.4.1 What are the End User Preferences?

The End User Preferences control default Oracle9iAS Discoverer behavior.

The Preferences file (pref.txt) controls default settings that apply to all end users. When a new session starts, the settings in the pref.txt file take effect.

In some cases, end users can override these settings from the Discoverer Plus Options dialog. Individual preferences are stored for each user as a unique combination of database and userid, and loaded whenever the user starts a new session. Therefore, users can log on from different client machines and still have their individual settings take effect.

For more information about the settings in the pref.txt file, refer to [Appendix C.1, "Preference settings in pref.txt"](#).

---

---

**Note:** Editing Preferences, such as adding a machine IP address, does not require that you shut down the Preferences object. Thus, users can continue working uninterrupted while you edit.

---

---

## 2.4.2 To set End User Preferences:

1. Use a text editor to open the `pref.txt` file in the following location:

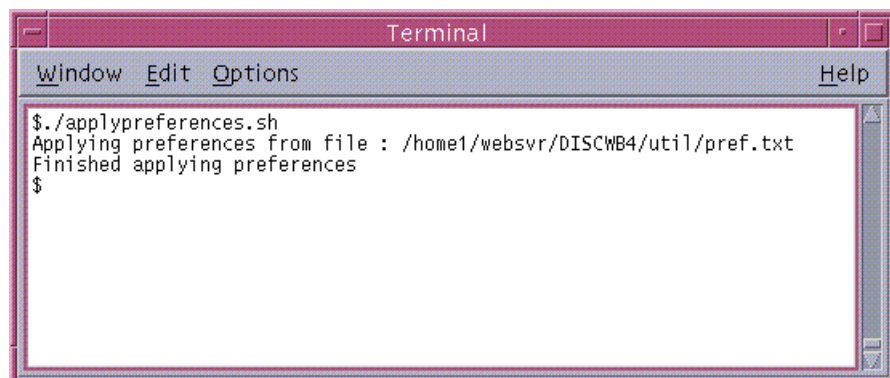
<ORACLE\_806\_HOME>/discwb4/util

The file **defaults.txt** (located in the above directory) is a copy of the original `prefs.txt` file and contains the default preference values for subsequent reference.

2. Edit the items in the `pref.txt` file.
3. Save the file and close the text editor.
4. Navigate to the <ORACLE\_806\_HOME>/discwb4/util directory.
5. Type the following at the command prompt to apply the preferences:

```
> applypreferences.sh
```

A confirmation message shows that the new preferences have been saved. If you suspect that the file contains errors, check the `error.txt` file in the current directory.



```
Terminal
Window Edit Options Help
$ ./applypreferences.sh
Applying preferences from file : /home1/websvr/DISCWB4/util/pref.txt
Finished applying preferences
$
```

The Preferences are updated in the Preferences component. Preferences are now set for the Discoverer Services.

## 2.5 Shutting Down

If you need to perform maintenance, change hardware settings, or perform any other tasks, you must shut down the entire Discoverer system. This prevents users from starting sessions on that server while you work.

*A reminder:* You do not need to shut down to edit Discoverer Preferences.

### 2.5.1 Shutting Down the Entire Discoverer System

You can shut down the entire Discoverer system in two ways:

- gradually, allowing current users to disconnect when they are ready
- immediately, forcing current users to end their Discoverer session

#### 2.5.1.1 To Shut Down the Discoverer System Gradually

Shutting down the Discoverer system gradually enables you to prevent new sessions from starting but allow current users to complete their sessions.

1. Navigate to the <ORACLE\_806\_HOME>/discwb4/util directory and type the following at the command prompt to stop the Locator:

```
> stoplocator.sh
```

Stopping the Locator prevents any new users from logging in.

2. Navigate to the <ORACLE\_806\_HOME>/discwb4/util directory and type the following at the command prompt to stop the OAD:

```
> stopoad.sh
```

3. Navigate to the <ORACLE\_806\_HOME>/discwb4/util directory and type the following at the command prompt to stop the SmartAgent:

```
> stoposagent.sh
```

4. Type the following at the command prompt to list the numeric identifier of every dis4pr Discoverer process that is currently running:

```
> ps -A|grep dis4pr
```

For example, the output might be:

```
18064 pts/3 0:01 dis4pr
18072 pts/4 0:08 dis4pr
```



5. Type the following at the command prompt to stop the first dis4pr process:

```
> kill -9 <process id>
```

where <process id> is the numeric identifier of the first dis4pr process.

For example, you might type the following:

```
> kill -9 18064
```

6. Repeat the previous step until you have stopped all the dis4pr processes.

When all of the current users have completed their sessions, the Discoverer server is shut down.

To check whether any Discoverer sessions are active, type the following at the command prompt to list the numeric identifier of every dis4ws Discoverer process that is currently running:

```
> ps -A|grep dis4ws
```

To stop any Discoverer dis4ws processes that are still active, type the following at the command prompt:

```
> kill -9 <process id>
```

where <process id> is the numeric identifier of the dis4ws process.

### 2.5.1.2 To Shut Down the Discoverer System Immediately

An immediate shut down stops new user sessions from starting and stops the current sessions. To perform an immediate shut down, you:

- prevent new sessions from being started
- stop the sessions currently running
- stop the Preferences component

1. Type the following at the command prompt:

```
> stopall.sh
```

## 2.6 Restarting the Discoverer system

You can restart the Discoverer system quickly and easily from the command line because all of the components are already registered with the OAD.

To restart the Discover system:

1. Navigate to the <ORACLE\_806\_HOME>/discwb4/util directory and type the following (in the order shown here) at the command prompt:

```
> startosagent.sh
```

```
> startoad.sh
```

```
> startlocator.sh
```

## 2.7 Adding and Removing Servers

You can add or remove server machines from your Discoverer system at any time.

### 2.7.1 Adding an Additional Discoverer Server machine

Follow the instructions in [Chapter 3, "Installing Oracle9iAS Discoverer on multiple machines"](#) to add a new server machine to the Discoverer system.

### 2.7.2 To Permanently Remove an Additional Discoverer Server Machine

Shutting down the OAD effectively removes an Additional Discoverer Server machine from the Discoverer system. You do not have to uninstall any software. However, you must edit the pref.txt file on the Master Discoverer Server machine to remove the Additional Discoverer Server machine's IP address from it.

1. Use a text editor to open the pref.txt file in the following location:

```
<ORACLE_806_HOME>/discwb4/util
```

2. Remove the machine's IP address from the pref.txt file.
3. Save the pref.txt file and close the text editor.
4. Type the following at the command prompt to apply the preferences:

```
> applypreferences.sh
```

5. Unregister all the Discoverer Services components installed on the Additional Discoverer Server machine by navigating to the <ORACLE\_806\_HOME>/discwb4/util directory and running the appropriate unregister scripts as follows:

- Type the following at the command prompt to unregister the Preferences component from the OAD:  
> unregisterpreference.sh
  - Type the following at the command prompt to unregister the Session component from the OAD:  
> unregistersession.sh
6. Navigate to the <ORACLE\_806\_HOME>/discwb4/util directory and type the following at the command prompt to stop the OAD:
- > stopoad.sh.

---

---

**Note:** To temporarily remove a machine, shut down the OAD.

---

---

## 2.8 Utilizing the Discoverer Viewer JavaScript Enhanced User Interface

Discoverer Viewer is supplied with two alternative user interfaces:

- a pure HTML user interface
- an JavaScript enhanced user interface

The JavaScript enhanced user interface provides:

- a popup menu for drill items
- drag-and-drop pivot functionality
- automatically applied page items

You can specify whether the JavaScript enhanced user interface is available to users using the `use_javascript_enhanced_ui` option in the `viewer_config.xml` file.

Make changes as follows:

1. Open the `viewer_config.xml` file for editing.

The `viewer_config.xml` file is located in the `<i>SUITES_HOME</i>/Apache/Apache/htdocs/disco4iv/html` directory.

2. Change the `use_javascript_enhanced_ui` option as required.

Below is an example of the change:

**viewer\_config.xml - Before change**

```
<option id="use_javascript_enhanced_ui" state="true"/>
```

**viewer\_config.xml - After change**

```
<option id="use_javascript_enhanced_ui" state="false"/>
```

Note that for a user to run the JavaScript enhanced user interface, both JavaScript and cookies must be enabled in the user's browser.

## 2.9 Frequently Asked Questions

### What environment variables are set?

During installation, four environment variables are set for the VisiBroker products:

```
OSAGENT_PORT= 14000
VBROKER_ADM = <ORACLE_806_HOME>/vbroker/adm
VBROKER_IMPL_DIR= <ORACLE_806_HOME>/vbroker/adm/impl_dir
VBROKER_JAVAVM = <ORACLE_806_HOME>/jre1162o/bin/jre
```

The CLASSPATH variable is not modified.

The PATH variable is updated with:

```
PATH = $<ORACLE_806_HOME>/vbroker/bin:$<ORACLE_806_HOME>/discwb4/bin:$<ORACLE_806_HOME>/bin:$PATH
```

This minimizes any conflict with other products.

### What happens if the Locator or Preferences components stop running?

If the Locator component goes down, no more users can log in to the system until it comes back up. Existing sessions are not affected.

The Locator component is designed so that it does not need to be shut down even when you edit the pref.txt file. By default the Locator component will pick up the edited values when the next client is connected.

Similarly, if the Preferences component goes down, the Locator component will continue to function and sessions already in progress are not affected. When the Preferences component is brought back up, the Locator component will bind to the Preferences component again.

### What should I do if I get a network error?

If Oracle 9iAS Discoverer receives a network exception during login, check the following:

- Are there any directories or jar files in the CLASSPATH or PATH that might interfere with Oracle9iAS Discoverer? Oracle9iAS Discoverer uses VisiBroker CORBA V3.4 and JDK 1.2.
- Make sure the Locator component is up and running. Did you see any error messages in the locator.log file? Make sure the locator.ior file is in the <iSUITES\_HOME>/Apache/Apache/htdocs/discwb4/applet directory

- Make sure the VisiBroker SmartAgent and OAD are up and running and that a new session can be started.
- Make sure the Preferences component is up and running. Use the command `ps-A|grep dis4pr` to confirm that the `dis4pr` process is running.

**What do I need to do if I get ORA-12154 TNS: Could not resolve service name?**

This is an Oracle error that appears when the session cannot connect to the database alias specified in the Connect field. Check that the:

- database alias is in the `tnsnames.ora` file
- database alias exists in the `tnsnames.ora` file on every machine that runs sessions

Hint: If you have SQL\*Plus (or any other Oracle product) running on that machine, try connecting to the database with that product.

**What do I need to do if the client gives the error: " Unable to connect to Oracle Discoverer Application Server because session failed to find locale en\_US. Please contact your system administrator"?**

1. Use a text editor to open the `startoad.sh` script in the following location:
2. Make sure that the `startoad.sh` script contains the following line:

```
unset ORA_NLS4
```

3. If the script does not contain this line, add it now.

**What do I need to do if `oad.log` contains "OAD: Fatal Error [Cross-device link] " while registering with OAD?**

The OAD gives a cross-device link error in the following circumstances:

- if the `VBROKER_ADM` directory and the `/tmp` directory are not on the same device

In this case, set the `TMPDIR` environment variable to a directory on the same device as the `VBROKER_ADM` directory before starting OAD.

- if you do not have write permission for the `VBROKER_ADM/impl_dir` directory or for the `impl_rep` and/or `impl_rep~` files.

In this case, enable write permission on the above directory and files.

**What do I need to do if I run `registerall.sh` and the message "oadutil: unable to bind to OAD" is displayed?**

Check the following:

- Make sure the OAD is properly running. To find out, use the `ps -A|grep oad` command. If the OAD is not running, check the `oad.log` file for any errors.
- Make sure there is at least one `osagent` running in the subnet (you can run one in the machine itself using `startosagent.sh`).
- Determine whether the machine is configured for multiple IP addresses. To find out, use the `ifconfig -a` command.

If the machine is configured for multiple IP addresses, follow these steps to include the machine IP address in the `oadutil` file:

1. From the command prompt, navigate to the `<ORACLE_806_HOME>/discwb4` directory and run the appropriate script:

- If you are using the Korn shell or the Bourne shell run the command

```
> . ./discwb.sh
```

Note that you must separate the periods with a space.

- If you are using the C Shell, type the following at the command prompt

```
> source discwb.csh
```

2. Type the following at the command prompt:

```
> oad -v
```

The first line of the output contains the machine IP address, as shown below:

```
Caught exception while checking for OAD [<IP_ADDR>, Exception:  
CORBA::NO_IMPLEMENT
```

where `<IP_ADDR>` is the machine IP address.

3. Use a text editor to open the `oadutil` file in the following location:

```
<ORACLE_806_HOME>/vbroker/bin
```

4. Add the machine IP address (preceded by -host) to the last line of the oadutil file as shown below:

<b>Before change:</b>	<code>\${VBJ} -VBJincppjar 1 -VBJaddJar vbjtools.jarcom.visigenic.vbroker.activationImpl.OADutil \$@</code>
<b>After change:</b>	<code>\${VBJ} -VBJincppjar 1 -VBJaddJar vbjtools.jarcom.visigenic.vbroker.activationImpl.OADutil \$@ -host &lt;IP_ADDR&gt;</code>
<b>where:</b>	<IP_ADDR> is the machine IP address identified by running the <code>oad -v</code> command.

5. Save the oadutil file and close the text editor.



---

---

## Installing Oracle9*i*AS Discoverer on multiple machines

Discoverer Plus and Discoverer Viewer are automatically installed to run with Oracle9i Application Server as a single machine installation. However Oracle9*i*AS Discoverer is designed to be deployed on multiple machines. The Discoverer Services can be replicated by specifying the machines available to Discoverer in a configuration file. Discoverer Services sessions are then started on alternate machines to balance the load.

In this way very large numbers of Discoverer users can be supported, since there is no single bottleneck. Multiple Discoverer sessions can run on a single machine, or can be spread across multiple machines as required.

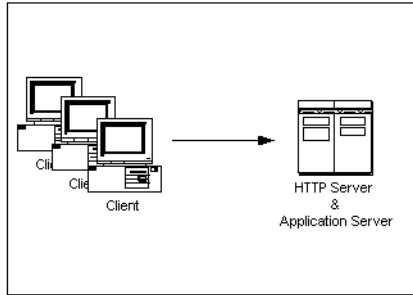
The topics include:

- What are the differences between a default installation and a distributed installation of Oracle9*i*AS Discoverer?
- Performing a distributed installation

### 3.1 What are the differences between a default installation and a distributed installation of Oracle9*i*AS Discoverer?

#### 3.1.1 Default installation of Oracle9*i*AS Discoverer

Oracle9*i*AS Discoverer is installed as part of Oracle9i Application Server. The default installation is a single machine installation (i.e. all the Discoverer components are installed on a single machine).



In this scenario, the Oracle9i Application Server installation has installed both of the following on the same machine:

- HTTP Server software
- all the Discoverer Services components (Locator, Session, Preferences, SmartAgent, and Object Activation Daemon)

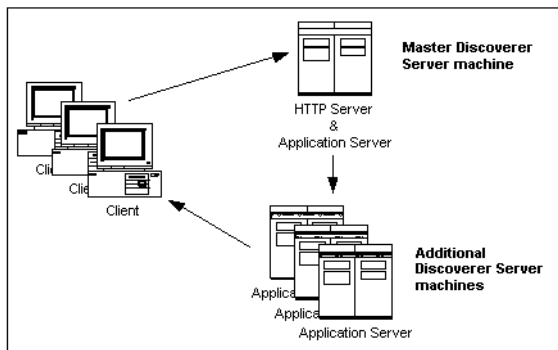
The machine is therefore both an HTTP Server machine and an Application Server machine.

When a user launches a web browser on a client machine to start Oracle9iAS Discoverer, the web browser sends a request to the HTTP Server software on the server machine. That request is processed and a Discoverer session is started on the server machine.

### 3.1.2 Distributed Oracle9iAS Discoverer installation

Having a single machine acting as both the HTTP Server machine and the Application Server machine can be inefficient when supporting large numbers of Discoverer users.

For this reason, the Discoverer Services components are designed to be deployed on multiple machines. By specifying the available machines in a configuration file, Discoverer Services sessions can be started on different machines to balance the load. Multiple Discoverer sessions can run on a single machine, or can be spread across multiple machines as required.



In this scenario (as with a default installation), the Oracle 9i Application Server installation has installed both of the following on the same machine:

- HTTP Server software
- all the Discoverer Services components (Locator, Session, Preferences, SmartAgent, and Object Activation Daemon)

The machine is therefore both an HTTP Server machine and an Application Server machine. This machine is referred to as the Master Discoverer Server machine.

As well as creating a Master Discoverer Server, you also install the Oracle 9i Application Server on additional machines. These machines are referred to as Additional Discoverer Server machines.

When a user launches a web browser on a client machine to start Oracle9iAS Discoverer, the web browser sends a request to the HTTP Server software on the Master Discoverer Server machine. That request is processed and a Discoverer session is started either on the Master Discoverer Server machine or on an Additional Discoverer Server machine.

Each Additional Discoverer Server machine can run multiple sessions; therefore, the number of machines you choose to use for sessions depends on how many concurrent sessions the users need, the amount of memory available, the speed of the hardware, and so forth.

## 3.2 Performing a distributed installation

### 3.2.1 Confirming connection to the default Master Discoverer Server machine

Before you start altering the default installation, we recommend that you connect to the default Master Discoverer Server machine to confirm that the default installation is functioning correctly.

1. Start a Web browser on a client machine.
2. Connect to the default Master Discoverer Server by typing the following URL:

```
http://hostname.domain/discoverer4i/viewer
```

where *hostname.domain* is the fully qualified name of the machine on which Oracle9i Application Server has been installed.

The Discoverer Viewer login screen appears when the default installation is functioning correctly.

### 3.2.2 Confirming the name of the Master Discoverer Server machine

During installation, the machine name is used as the Master Discoverer Server instance name. Make a note of the name of the Master Discoverer Server machine now because you will need it when installing on the machines that will become Additional Discoverer Server machines.

If you are not sure of the Master Discoverer Server machine name, follow the instructions below:

1. Use a text editor to open the `htmlvars.js` file in the following location:  
`<i>SUITES_HOME>/Apache/Apache/htdocs/discwb4/scripts`
2. Look for the string `ORAWEBDISCO_INSTANCE`.

The value of this string is the Master Discoverer Server machine name (and therefore the Master Discoverer Server instance name).

### 3.2.3 Installing Discoverer on Additional Discoverer Server machines

On every machine that you want to make an Additional Discoverer Server machine:

1. Follow the instructions in the *Oracle9i Application Server Installation Guide* to install Oracle9i Application Server on the machine you want to use as the Additional Discoverer Server machine.

2. On the Additional Discoverer Server machine, navigate to the <ORACLE\_806\_HOME>/discwb4 directory and set the Discoverer environment variables by running the appropriate script:
  - If you are using the Korn shell or the Bourne shell, type the following at the command prompt:

```
> . ./discwb.sh
```

Note that you must separate the periods with a space
  - If you are using the C shell, type the following at the command prompt:

```
> source discwb.csh
```
3. On the Additional Discoverer Server machine, navigate to the <ORACLE\_806\_HOME>/discwb4/util directory and type the following at the command prompt to unregister the corresponding Discoverer components:

```
> unregistersession.sh
```

```
> unregistercollector.sh
```

```
> unregisterpreference.sh
```
4. On the Additional Discoverer Server machine, navigate to the <ORACLE\_806\_HOME>/discwb4/util directory and type the following at the command prompt to stop all Discoverer processes:

```
> stopall.sh
```
5. On the Additional Discoverer Server machine, delete the following scripts from the <ORACLE\_806\_HOME>/discwb4/util directory:
  - registerpreference.sh
  - registercollector.sh
  - startlocator.sh
  - unregisterpreference.sh
  - unregistercollector.sh
6. On the Additional Discoverer Server machine, use a text editor to open the registersession.sh file and replace the Additional Discoverer Server machine name with the name of the Master Discoverer Server machine.

Note that there are three occurrences of the Additional Discoverer Server machine name in the `registersession.sh` file:

- two occurrences of the Additional Discoverer Server machine name are prepended to the string `OracleDiscovererSession4`
  - one occurrence of the Additional Discoverer Server machine name is prepended to the string `OracleDiscovererPreferences4`
7. On the Additional Discoverer Server machine, use a text editor to open the `unregistersession.sh` file and replace the Additional Discoverer Server machine name with the name of the Master Discoverer Server machine.

Note that there is one occurrence of the Additional Discoverer Server machine name in the `unregistersession.sh` file (prepended to the string `OracleDiscovererSession4`).

8. On the Additional Discoverer Server machine, navigate to the `<ORACLE_806_HOME>/discwb4/util` directory and type the following at the command prompt to start the OAD and Session component:

```
> startoad.sh  
  
> startosagent.sh  
  
> registersession.sh
```

When you have installed Discoverer on every machine that you want to be an Additional Discoverer Server machine, you must:

- edit the `pref.txt` file on the Master Discoverer Server machine (Discoverer Plus and Viewer - see Section 3.2.4)
- edit the `tnsnames.ora` file on all Discoverer Server machines (Discoverer Plus and Viewer - see Section 3.2.5)
- edit the `disco4iv.xml` file on the Master Discoverer Server machine (Discoverer Viewer only - see Section 3.2.6)

## 3.2.4 Editing the Discoverer Services `pref.txt` file (Discoverer Plus and Viewer)

### 3.2.4.1 What is the `pref.txt` file?

The `pref.txt` file located on the Master Discoverer Server machine contains a number of default settings that apply to all end users.

The only information required in the `pref.txt` file is the IP addresses (or machine names) of the machines used for Oracle9iAS Discoverer. The other items in the `pref.txt` file can be edited to suit your network or user requirements. See Section 3.2.4.5, “How to edit the `pref.txt` file” for more information.

A backup copy of `pref.txt` is located at `<ORACLE_806_HOME>/discwb4/util/defaults.txt` on the Master Discoverer Server machine. If you make a mistake when editing, lose, or corrupt the `pref.txt` file, you can restore the file to its default values from the backup copy.

#### 3.2.4.2 Why edit the `pref.txt` file?

In a distributed Discoverer installation, the `pref.txt` file located on the Master Discoverer Server must contain either the IP address or machine name of each Application Server machine that will run Oracle9iAS Discoverer sessions.

The order of the IP addresses listed in the `pref.txt` file determines the order in which sessions are started on each machine that has a Session component installed. Editing the `pref.txt` file enables you to add Additional Discoverer Server machines and to balance the load between them.

Entries in the `pref.txt` file are comma space delimited (i.e. entries are separated by a comma followed by a space). Enclose the entries in “double quotes.”

---

---

**Note:** Using IP addresses is somewhat faster because using machine names requires one extra lookup. See your System administrator for details about using IP addresses versus machine names.

---

---

#### 3.2.4.3 What is load balancing?

Load balancing is a way to improve the performance of your Discoverer installation by distributing sessions amongst different machines to make the most efficient use of each machine. You specify the machines to use by including their IP addresses (or machine names) in the `pref.txt` file.

The Discoverer Services Locator component uses the list of machines in the `pref.txt` file to determine which machine starts the next requested session. Using a round-robin method, the Locator requests a session on each machine in order, until it reaches the end of the list of IP addresses. When the Locator reaches the end of the list, it returns to the beginning of the list and starts over.

This round-robin method enables you to balance the load placed on the machines by:

- listing the machines in a particular order
- including the same machine several times

For example, you may want one machine to handle five sessions before the next machine is used. If so, include the first machine's IP address five times before including the next machine's address in the `pref.txt` file

Load balancing can greatly affect performance. We recommend that you plan a load balancing strategy before editing the `pref.txt` file. Consider the speed and performance of each server machine. For planning purposes, you can estimate that each session requires approximately 8 to 12 MB of memory.

### 3.2.4.4 Examples of Load Balancing

Example 1: Load balancing - equal machines

You have Discoverer Session components installed on two Application Server machines. Each machine has the same processor speed and RAM. The IP addresses for these two machines are:

**Table 3–1 Sample Server Names**

Server	Machine Name	IP address
App Server machine 1	disco1	123.456.789.1
App Server machine 2	disco2	123.456.789.2

You want App Server machine 1 to handle the first request, App Server machine 2 to handle the next, App Server machine 1 the next, and so on.

The `pref.txt` file entry would look like this:

```
Machine IPs="disco1, disco2"
```

or

```
Machine IPs="123.456.789.1, 123.456.789.2"
```

Example 2: Load balancing - one fast machine plus two other machines

You have Discoverer Sessions installed on three Application Server machines. The first Application Server machine has faster processors and more RAM. The second and third Application server machines are equal to each other in processor speed and RAM.



The IP addresses for these three machines are:

**Table 3–2 Sample Server Names**

<b>Server</b>	<b>Machine Name</b>	<b>IP address</b>
App Server machine 1	server1	123.456.789.1
App Server machine 2	server2	123.456.789.2
App Server machine 3	server3	123.456.789.3

You want App Server machine 1 to handle the first three requests, App Server machine 2 to handle next one, App Server machine 3 the next one, then back to App Server machine 1 for the next three requests.

The pref.txt file entry would look like this:

```
Machine IPs="server1, server1, server1, server2, server3"
```

or

```
Machine IPs="123.456.789.1, 123.456.789.1, 123.456.789.1, 123.456.789.2, 123.456.789.3"
```

### 3.2.4.5 How to edit the pref.txt file

You edit the pref.txt file on the machine where you installed the Preferences component (i.e. the Master Discoverer Server machine):

1. Use a text editor to open the pref.txt file in the following location:

```
<ORACLE_806_HOME>/discwb4/util
```

```

Terminal
Window Edit Options Help
# Discoverer Server Preferences File.
# To modify server preferences, edit this
# file, save it and then apply by executing
# the following:
# dis4pr -apply.

# Preferences must be under one of the following types:
# Application, Database, Crosstab, Table, Administrator, Debug, Report, WebCharts, Fonts, Locator,
# Server Machine, Session Manager

# Preference          Setting          Comment
[Locator]
# Administrator needs to set this value.
MachineIPs            = ""             # List of IP Addresses used by the Locator
# to spawn Sessions. List IP addresses of all machines where you installed Discoverer Sessions. For d
# etails see the Install and Admin Guide.

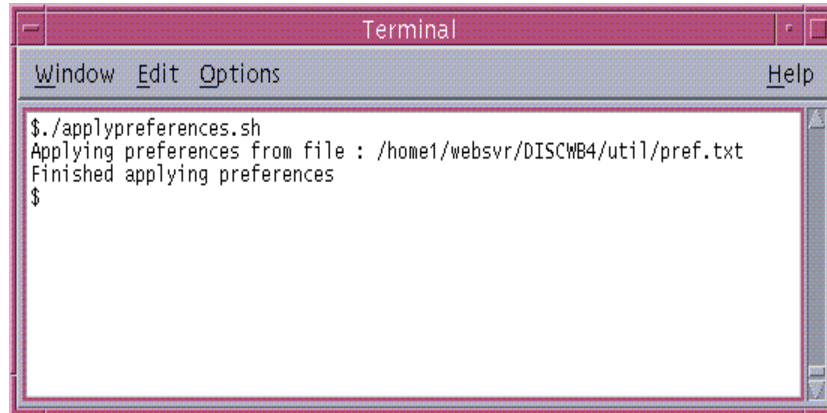
# User visible settings.

[Administrator]
ShowUserTypeChoice   = 0              # If set to "1", shows a checkbox on the co
# nnect string so the user can elect if they are an Apps user or not.
DefaultUserTypeIsApps = 0              # If set to "1", assume the username is an
# Apps username.

[Session Manager]
Timeout               = 600            # Timeout in seconds if there is no client
    
```

2. Type the IP address or machine name for each machine at the line labeled MachineIPs near the top of the file.
3. Separate each IP address or machine name with a comma followed by a space and enclose the entire list in “double quotes.”  
For example: “server1, server2, server3”.
4. When you’re finished editing the pref.txt file, save it and exit.  
For the changes you have made to pref.txt to take effect, you must apply preferences.
5. From the command prompt, navigate to the <ORACLE\_806\_HOME>/discwb4/util directory and run the following script to apply the preferences:  

```
> ./applypreferences.sh
```

A terminal window titled "Terminal" with a menu bar containing "Window", "Edit", "Options", and "Help". The terminal output shows the execution of a shell script: "\$ ./applypreferences.sh", followed by "Applying preferences from file : /home1/websvr/DISCWB4/util/pref.txt", then "Finished applying preferences", and finally a prompt "\$".

```
Terminal
Window Edit Options Help
$ ./applypreferences.sh
Applying preferences from file : /home1/websvr/DISCWB4/util/pref.txt
Finished applying preferences
$
```

If errors are detected in the `pref.txt` file, the confirmation window file includes an instruction to check the `error.txt` file. Correct any errors listed in the `error.txt` file and reapply preferences before continuing.

The preferences are updated and set for the Discoverer Server Instance.

## 3.2.5 Editing the `tnsnames.ora` file (Discoverer Plus and Viewer)

### 3.2.5.1 What is the `tnsnames.ora` file?

The `tnsnames.ora` file contains the names and aliases of all the databases that users can access using Oracle9iAS Discoverer or any other Oracle product.

### 3.2.5.2 Why edit the `tnsnames.ora` file?

Each server machine in a distributed Discoverer installation must have an identical `tnsnames.ora` file.

If you have installed Discoverer on an Additional Discoverer Server machine, you must make sure the `tnsnames.ora` files on the different machines are identical by doing one of the following

- copy the `tnsnames.ora` file from the Master Discoverer Server machine to each Additional Discoverer Server machine
- edit the `tnsnames.ora` file on each Additional Discoverer Server so that it is identical to the `tnsnames.ora` file on the Master Discoverer Server machine

Note that several versions of the `tnsnames.ora` file might exist on the same machine. It is very important that you copy or edit the correct file.

### 3.2.5.3 How to copy the `tnsnames.ora` file

1. Copy the `tnsnames.ora` file **from** the following location on the Master Discoverer Server machine **to** the same location on the Additional Discoverer Server machine and replacing the existing file:

```
<ORACLE_806_HOME>/network/admin
```

where `<ORACLE_806_HOME>` is the location in which the Discoverer Components are installed.

You can copy the `tnsnames.ora` file in a number of different ways. For example, you could copy the file having mounted the Master Discoverer Server machine on the Additional Discoverer Server machine as a network drive (e.g using NFS). Alternatively, you could use FTP software to copy the file.

### 3.2.5.4 How to edit the `tnsnames.ora` file

One way to edit the `tnsnames.ora` file:

1. Navigate to the `<ORACLE_806_HOME>/discwb4` directory and set the Discoverer environment variables by running the appropriate script:

- If you are using the Bourne shell or the Korn shell, type the following at the command prompt:

```
> . ./discwb.sh
```

Note that you must separate the periods with a space

- If you are using the C shell, type the following at the command prompt:

```
> source discwb.csh
```

2. Navigate to the `<ORACLE_806_HOME>/bin` directory and type the following at the command prompt:

```
> net8wiz.sh
```

A program runs, enabling you to edit the Net8 configuration.

To make sure that you edit the correct `tnsnames.ora` file, do **not** use the Net8 Configuration Assistant.

## 3.2.6 Editing the disco4iv.xml file (Discoverer Viewer only)

### 3.2.6.1 What is the disco4iv.xml file?

The disco4iv.xml file on the Master Discoverer Server machine provides Discoverer Viewer with configuration information. Among other things, Discoverer Viewer uses the disco4iv.xml file to identify the Master Discoverer Server machine.

### 3.2.6.2 Why edit the disco4iv.xml file?

When you install Discoverer Viewer on multiple machines, you need to modify the disco4iv.xml file to identify the Master Discoverer Server.

### 3.2.6.3 How to edit the disco4iv.xml file and what to change

Before editing the disco4iv.xml file, we strongly recommend that you take a copy of the file for future reference.

1. Open the disco4iv.xml file in the following location on the Master Discoverer Server machine:

```
<iSUITES_HOME>/Apache/Apache/htdocs/disco4iv/html
```

where `<iSUITES_HOME>` is the location in which Oracle9i Application Server components are installed.

2. Confirm that the session name entry in the disco4iv.xml file is as follows:

```
session name="instance_nameOracleDiscovererSession4"
```

where `instance_name` is the Master Discoverer Server instance name.

3. Confirm that the locator name entry in the disco4iv.xml file is as follows:

```
locator name="instance_nameOracleDiscovererLocator4"
```

where `instance_name` is the Master Discoverer Server instance name.

4. Change the path entry in the disco4iv.xml file

change from:	path="http://%LOCATOR_URL%"
change to:	path="http:// <i>locator_location</i> "
where:	<i>locator_location</i> is the location of the locator.ior file
example:	path="http://hostname.domain:PortNo/discwb4/applet/"

5. Remove the following two lines in the disco4iv.xml file:

<!-- REMOVE THIS COMMENT LINE (A) only if Discoverer 4i Viewer is installed on a different subnet than the Discoverer Services, or you are performing a distributed Discoverer Services installation

REMOVE THIS COMMENT LINE (B) only if Discoverer 4i Viewer is installed on a different subnet than the Discoverer Services, or you are performing a distributed Discoverer Services installation-->

6. Save the file and exit.
7. Navigate to the <i>SUITES\_HOME</i>/Apache/Apache/bin directory and verify the changes you have made by typing the following at the command prompt:

```
> apachectl configtest
```

If the 'Syntax OK' message does not appear, review the changes you have made, correct any mistakes, and run the above command again.

When the changes you have made are syntactically correct, the message 'Syntax OK' appears and you can go on to the next step.

8. Restart the Oracle HTTP Server by typing the following at the command prompt:

```
> apachectl restart
```

The message 'apachectl restart: httpd restarted' confirms that the Oracle HTTP Server has successfully restarted.

For the changes you have made to the `disco4iv.xml` file to take effect, you must restart the Discoverer Services components (for more information, refer to Section 2.7 "Restarting Discoverer 4i System").

### 3.2.7 Confirming connection to the Additional Discoverer Server

To confirm that the creation of Additional Discoverer Server machines has been successful, we recommend that you connect to each Additional Discoverer Server machine as follows:

1. Start a Web browser on a client machine.
2. Connect to the Master Discoverer Server machine by typing the following URL:

```
http://hostname.domain/discoverer4i/viewer
```

where *hostname.domain* is the fully qualified name of the Master Discoverer Server machine.

A process starts on the first Additional Discoverer Server machine specified in the `pref.txt` file and the Discoverer Viewer login screen appears.

3. To confirm that the installation was successful, type the following at the command prompt on the first Additional Discoverer Server machine to see details of the Discoverer process that has started:

```
> ps -A|grep dis4ws
```

4. Repeat the above steps for each Additional Discoverer Server machine specified in the `pref.txt` file.





---

---

## Using a URL to start Oracle9iAS Automatically

End users normally start Oracle9iAS Discoverer by logging in, choosing a workbook, and selecting parameter values. To speed up this process, you can create a URL to start Oracle9iAS Discoverer with specific settings (e.g. which workbook to open, which parameter values to use). Using a URL to automatically start Discoverer not only saves your end users time but also gives you control over the workbooks they can use.

Having created the URL, you can:

- give the URL to end users to type into their Web browser
- add the URL as a link on your company's Intranet site so that end users can start Discoverer with a single click

The topics in this chapter are:

- [What is the format of the URL to start Oracle9i AS Discoverer?](#)
- [Discoverer Plus URL arguments](#)
- [Discoverer Viewer URL arguments](#)

### 4.1 What is the format of the URL to start Oracle9i AS Discoverer?

The URL you create must adhere to the standard URL command line syntax as follows:

```
http://hostname.domain/prog_name?arg1=value1&arg2=value2&...&argN=valueN
```

where:

- *hostname.domain* is the fully qualified name of the Additional Discoverer Server machine
- *prog\_name* is one or other of the following (depending on whether you want to start Discoverer Plus or Discoverer Viewer):
  - the location and name of the Discoverer Plus Start page where end users click the **Start** button to launch the Discoverer Plus applet (e.g. `discwb4/html/english/welcome.htm`)
  - the location and name of the Discoverer Viewer servlet (e.g. `servlets/disco4iv`)
- `arg1=value1` is the first argument and the value specified for it

Note that the order of the arguments in the URL is not important.

If the following required arguments are not specified on the URL, end users are prompted for them:

- username
- password
- database
- workbook name
- worksheet name
- parameters

To include some special characters in a URL you must replace those characters with the equivalent ASCII (or in some cases UNICODE) codes. This process is known as URL encoding. For example, you must replace any spaces in the URL with a plus (+) symbol. Any characters not in the following list must be URL encoded:

- capital letters (i.e. A B C D E F G H I J K L M N O P Q R S T U V W X Y Z)
- lower case letters (i.e. a b c d e f g h i j k l m n o p q r s t u v w x y z)
- numerals (i.e. 0 1 2 3 4 5 6 7 8 9 0)
- certain special characters (i.e. - \_ . ! ~ ( ) \* ' )

## 4.2 Discoverer Plus URL arguments

**Table 4–1 Discoverer Plus URL arguments and values**

Argument and Values	Purpose	Example
BrandImage=<image>	Specifies the URL of a GIF file that replaces the Discoverer logo. For example, you can replace the Discoverer logo with your corporate logo.	BrandImage=http://.../logo.gif where logo.gif is your corporate logo
connect=[APPS_SECURE] <hostname_instancename>	For Oracle Applications end users, you can specify that they connect in secure mode. hostname_instancename refers to the database host name and the instance of the Applications server.	connect=[APPS_SECURE]genledger_payables
connect=<connection string>	Instructs Discoverer to connect to the database using the connection information supplied by connection string. This argument completely bypasses the Connection Information dialog for end users.	connect=username/password@demodb.
ConnectImage=<image>	Specifies the URL of a GIF file that replaces the “magnifying glass” image on the Connect to Discoverer dialog.	ConnectImage=http://.../logo.gif where logo.gif is your corporate logo
DefaultConnect=<database name or alias>	When Discoverer starts, the end user sees the Connection Information dialog. If you use this argument, the Database field is automatically filled in. The user can overwrite this database name or alias.	DefaultConnect=demodb
DefaultPassword=<string>	When Discoverer starts, the end user sees the Connection Information dialog. If you use this argument, the Password field is automatically filled in. The user can overwrite this password.	DefaultPassword=secret_code
DefaultUsername=<database username>	When Discoverer starts, the end user sees the Connection Information dialog. If you use this argument, the Username field is automatically filled in. The user can overwrite this username.	DefaultUsername=video_user

**Table 4–1 Discoverer Plus URL arguments and values**

Argument and Values	Purpose	Example
<code>eul=&lt;schema name&gt;</code>	Specifies the name of the database schema which contains the EUL that you want your end users to connect to by default. If you do not use this argument, Discoverer uses other methods to detect the default EUL (for example, from the end user's preference settings.)	<code>eul=VIDEOEUL</code>  Discoverer uses the EUL metadata that is found in the VIDEOEUL schema.
<code>FrameDisplayStyle=&lt;launched or separate&gt;</code>	Specifies the way in which the Discoverer application frame is displayed. <ul style="list-style-type: none"> <li>■ The value “launched” embeds the applet Start button into the calling HTML. The end user must click the Start button.</li> <li>■ The value “separate” automatically starts the Discoverer applet for the end user. The end user does not have to click the Start button.</li> </ul>	<code>FrameDisplayStyle=separate</code>  The Discoverer frame opens immediately in a separate window without requiring that the end user click the Start button.
<code>Locale=&lt;language[_country][_variant]&gt;</code>	Specifies the language (and optionally the country and variant) that end users see when they use Discoverer. This argument overrides the locale setting of the end user's client machine.  Use Java naming conventions to specify language, country, and variant.	<code>Locale=es_ES</code>  The language is Spanish, the country is Spain.
<code>NLSDateFormat=&lt;date format&gt;</code>	Formats dates for a specific region. For its value, type a SQL date format mask, such as DD-Mon-YY.	<code>NLSDateFormat=DD-Mon-YY</code>
<code>NLSDateLanguage=&lt;date language&gt;</code>	Selects the language for day and month names that end users see when they use Discoverer.	<code>NLSDateLanguage=Spanish</code>
<code>NLSLang=&lt;language&gt;</code>	Selects the language, territory, and character set end users see when they use Discoverer.  Use Oracle naming conventions to specify language, territory, and character set.	<code>NLSLang=Spanish</code>

**Table 4–1 Discoverer Plus URL arguments and values**

Argument and Values	Purpose	Example
NLSNumeric_Characters= <separator characters>	Selects decimal character and group separator for numbers.	NLSNumericCharacters=.,
NLSSort=<sort type>	Sets the character sort sequence.	NLSSort=XSpanish
opendb=<workbook name>	Gives Discoverer the name of the workbook you want to open and assumes that it is stored in the database (instead of scheduled). If you use the opendb argument more than once in the URL, Discoverer uses the last one.	opendb=Video Sales Workbook  Discoverer assumes the workbook is stored in the database and automatically opens it.
param_<parameter_name>= <parameter_value>	For workbooks that contain parameters, use this argument to select values. If the workbook does not contain a parameter by that name, Discoverer ignores the argument. Note that param_ must precede the parameter's name.	param_regionparam=East  Tells Discoverer to set the parameter named regionparam to the value East.
Responsibility= <responsibility name>	For Oracle Applications endusers (e.g. Oracle General Ledger), you can specify their responsibility.	Responsibility=Manager  Discoverer bypasses the Responsibility dialog and assigns the end user an Oracle Applications responsibility of Manager.
Sheet=<worksheetname>	Gives Discoverer the name of the worksheet you want to open by default. If you use the Sheet argument more than once in the URL, Discoverer opens the last one.	Sheet=Sales Detail Sheet  Discoverer runs the query for the worksheet named Sales Detail Sheet and displays the results to the end user.
WindowHeight=<pixel width>	Specifies the height in pixels of the Discoverer application frame. If you do not use this argument, Discoverer uses its own default value.	WindowHeight=600  The Discoverer frame is 600 pixels high.
WindowWidth=<pixel width>	Specifies the width in pixels of the Discoverer application frame. If you do not use this argument, Discoverer uses its own default value.	WindowWidth=800  The Discoverer frame is 800 pixels wide.
WorkbookName= <workbookname>	Gives Discoverer the name of the workbook you want users to open by default.	WorkbookName=Video Sales Workbook  Discoverer searches the database for the stored workbook called Video Sales Workbook.

**Table 4–1 Discoverer Plus URL arguments and values**

Argument and Values	Purpose	Example
WorkbookSource=<database or scheduled>	<p>Gives Discoverer the location of the workbook to open:</p> <ul style="list-style-type: none"> <li>■ The value “database” specifies the workbook is saved in the database.</li> <li>■ The value “scheduled” specifies the workbook is updated periodically.</li> </ul>	<p>WorkbookSource=database</p> <p>Discoverer searches the database for the stored workbook.</p>

### 4.3 Discoverer Viewer URL arguments

**Table 4–2 Discoverer Viewer URL arguments and values**

Argument and Values	Purpose	Example
ansdf=<date format>	For Oracle Applications end users, you can override the default date format for the session.	ansdf='MM/DD/YY'
ansdl=<date language>	For Oracle Applications end users, you can override the language to use for the spelling of day and month names and date abbreviations (AM, PM, AD, BC).	ansdl=fr
ansl=<language>	For Oracle Applications end users, you can override the session language	ansl=en-gb
ansnc=<separator characters>	<p>For Oracle Applications end users, you can override the default characters to use as the group separator and decimal.</p> <p>You must specify the decimal separator first, followed by the group separator.</p>	ansnc=',.'

**Table 4-2 Discoverer Viewer URL arguments and values**

Argument and Values	Purpose	Example
<code>anlss=&lt;sortname or binary&gt;</code>	For Oracle Applications end users, you can override the session collating sequence for ORDER BY queries and string comparisons. Values are: <ul style="list-style-type: none"> <li>■ the name of a linguistic sort sequence</li> <li>■ binary to specify a binary sort</li> </ul>	<code>anlss=binary</code>
<code>cp_height=&lt;number_of_pixels&gt;</code>	Specifies height of chart in pixels.	<code>cp_height=200</code>
<code>cp_show_legend=&lt;true or false&gt;</code>	Specifies whether a chart title is displayed.	<code>cp_show_legend=true</code>
<code>cp_width=&lt;number_of_pixels&gt;</code>	Specifies width of chart in pixels.	<code>cp_width=300</code>
<code>cs=[APPS_SECURE] &lt;hostname_instancename&gt;</code>	For Oracle Applications end users, you can specify that they connect in secure mode. <code>hostname_instancename</code> refers to the database host name and the instance of the Applications server.	<code>cs=[APPS_SECURE]genledger_payables</code>
<code>db=&lt;databasename&gt;</code>	Specifies the database to connect to	<code>db=video</code>
<code>eul=&lt;eulname&gt;</code>	Specifies the name of the EUL to connect as. You only need to specify this parameter if you want to override the default EUL.	<code>eul=myEUL</code>

**Table 4–2 Discoverer Viewer URL arguments and values**

Argument and Values	Purpose	Example
<code>fm=&lt;exportformat&gt;</code>	<p>Specifies one of the following export formats:</p> <ul style="list-style-type: none"> <li>▪ xml</li> <li>▪ xls (Microsoft Excel)</li> <li>▪ htm (HTML)</li> <li>▪ txt (Text)</li> <li>▪ csv (Comma Separated Values)</li> <li>▪ prn (Formatted Text)</li> <li>▪ dcs (Express)</li> <li>▪ dif (Data Interchange Format)</li> <li>▪ slk (Symbolic Link)</li> <li>▪ wks (Lotus 1-2-3)</li> <li>▪ sql (Structured Query Language)</li> </ul> <p>Note: You must run a query before you can use this parameter to export the data.</p>	<code>fm=xls</code>
<code>lm=&lt;applications or discoverer&gt;</code>	<p>Specifies the login method:</p> <ul style="list-style-type: none"> <li>▪ Set to "applications" to login in as an Oracle Applications user.</li> <li>▪ Set to "discoverer" to login in as a Discoverer user.</li> </ul> <p>If this parameter is not present, the default login method (as specified in the <code>viewer_config.xml</code> configuration file) is used.</p>	<code>lm=applications</code>
<code>nsl=&lt;language&gt;</code>	<p>Specifies the language code for the language you want.</p>	<code>nsl=en-gb</code>
<code>pg=&lt;page_number&gt;</code>	<p>Specifies the page of the worksheet to view.</p>	<code>pg=4</code>



**Table 4-2 Discoverer Viewer URL arguments and values**

Argument and Values	Purpose	Example
pi_ <pageitemname>=<pageitemvalue>	Specifies the name of a page item and the value to set it to.	pi_Region=West The above sets the "Region" page item to "West":
pw=<password>	Specifies the password corresponding to the specified user name for connecting to the specified database	pw=tiger
qp_<parameter_name>= <parameter_value>	Specifies the name of a query parameter and the value you want to set it to.	qp_City=Denver
qpd=<number_of_seconds>	Specifies the delay in seconds before a query progress page is displayed (or redisplayed). You might use this argument to avoid writing code to handle query progress pages when using the Viewer Servlet to return data to an application you have written.	qpd=60 The above specifies that a query progress page is to be displayed after the query has been running for 60 seconds. The query progress page will be re-displayed every 60 seconds until the query results are available.
rd=<number_of_seconds>	Specifies the delay in seconds before a browser is redirected to the original URL. Use this argument to prevent the browser timing out during export operations. Note that this argument only applies when a valid value has also been specified for the fm argument.	rd=90 The above specifies that the browser is redirected back to the original URL every 90 seconds until the export operation is complete.
rs=<responsibility>	Specifies the Oracle Applications Responsibility you want to connect with.	Rs=Manager
sg=<securitygroup>	Specifies the Oracle Applications Security Group you want to connect with.	Sg=securityGroup
us=<username>	Specifies the user name you want to use.	us=video4

**Table 4–2 Discoverer Viewer URL arguments and values**

Argument and Values	Purpose	Example
us=<connectString>	Specifies the username and database plus further optional connection information. The connect string uses the following syntax:  <username>[:<responsibility>[:<securityGroup>]][/<password>]@<database>	us=scott/tiger@disco
wb=<workbookname>	Specifies the name (or ID) of the workbook you want to open.	wb=My+Workbook
wbd=<workbookdate>	For opening a scheduled workbook. Specifies the date the scheduled workbook was run.	wbd=31-OCT-00
wbt=<workbooktime>	For opening a scheduled workbook. Specifies the time the scheduled workbook was run. If you specified the Scheduled Workbook's ID using the wb parameter, you don't need to use this parameter	wbt=4%3a00+PM The above specifies a time of 4:00 PM.
ws=<worksheetname>	Specifies the worksheet you want to open.	ws=My+Worksheet

---

---

# Customizing Discoverer Viewer

Discoverer Viewer is easily customized to fit in with your Web site look and feel, to incorporate your companies logo or other artwork, or to build custom Discoverer applications for the Web.

This chapter covers the following areas:

- [Using Discoverer Viewer for Customized Web Applications](#)
- [Customization Examples](#)

## 5.1 Using Discoverer Viewer for Customized Web Applications

### 5.1.1 Customize the general appearance

The appearance of Discoverer Viewer can be customized by specifying HTML formatting attributes in a single customization file. Fonts, colors and graphics are all easily changed by anyone familiar with HTML formatting.

### 5.1.2 Customize the application

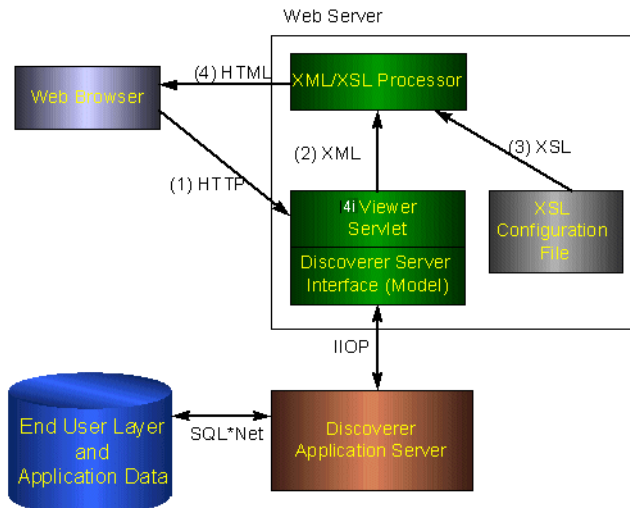
Discoverer Viewer can be used to build complete custom Web applications or deliver data to other middle tier Web systems. It uses industry standard XML to represent data and application state, and the XSL stylesheet language to format the User Interface. Standard XSL tools can be used to customize the User Interface or to produce a complete embedded Business Intelligence application.

### 5.1.3 How the HTML is produced

All the HTML generated by Discoverer Viewer is produced as a result of combining XML, which describes the information available, with XSL which defines how that information should be represented in HTML.

The XSL defines simple attributes, such as the fonts and colors to use, but it also defines the layout of each page, and the interactions with the user. By customizing the XSL, a specific Discoverer Application can be built and delivered on the Web.

**Figure 5–1 Discoverer Viewer HTML Generation using XSL/XML**



## 5.1.4 Steps to Generate the HTML needed to build a specific Discoverer Application

### 5.1.4.1 Step 1 Browser sends URL

After login, let's assume a Discoverer Viewer user has asked for a list of workbooks that they are allowed to open in order to do some analysis of their business. The URL issued is:

```
http://myserver.mycompany.com/Discoverer4i/Viewer?us=video&db=Disco
```

The URL specifies the machine the servlets are installed on, and the username and database connection string to use. The password is not normally shown on the URL for security reasons.

### 5.1.4.2 Step 2 XML generation

The URL is processed by the Discoverer Servlet, and the Discoverer Services is instructed to check the security setting for this user and return details of the workbooks that this user is allowed to access. The security settings are held in the End User Layer tables in the database. After this information is returned from the server, the Servlet generates the following XML:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="example1.xsl"?>
<discoverer version="4.1.x.x" login_method="discoverer">
  <request source="http://myserver.mycompany.com/servlets/viewer">
    <command name="db">disco</command>
    <command name="us">VIDEO</command>
  </request>
  <account>
    <user>VIDEO</user>
    <database>disco</database>
    <eul default="true" name="VIDEO">
      <workbook name="Annual Sales Report" ref="Annual~20Sales~20Report">
        <description>Shows yearly and quarterly sales of
products</description>
      </workbook>
      <workbook name="Store and Band Analysis"
ref="Store~20and~20Band~20Analysis">
        <description>Shows sales by Store, broken into sales
bands</description>
      </workbook>
      <workbook name="Video Sales Analysis" ref="Video~20Sales~20Analysis">
        <description>General purpose analysis of the Business </description>
      </workbook>
    </eul>
  </account>
</discoverer>
```

```
</eul>
<option name="qpw" enable="false">60</option>
<option name="qrl" enable="false">10000</option>
<option name="qtl" enable="false">1800</option>
<option name="msa" enable="false">60</option>
<option name="qif">250</option>
<option name="qll">15</option>
<option name="aq">true</option>
<option name="nv">NULL</option>
<option name="ftd">true</option>
<option name="rpp">25</option>
</account>
<export name="xls" format="application/vnd.ms-excel">Microsoft Excel Workbook
(*.xls)</export>
<export name="htm" format="text.html">Hyper-Text Markup Language
(*.htm)</export>
<export name="txt" format="text.plain">Text (Tab delimited) (*.txt)</export>
<export name="csv" format="text.plain">CSV (Comma delimited) (*.csv)</export>
<export name="prn" format="text.plain">Formatted Text (Space delimited)
(*.prn)</export>
<export name="dcs" format="text.plain">DCS (Express Format) (*.dcs)</export>
<export name="dif" format="application/vnd.ms-excel">DIF (Data Interchange
Format) (*.dif)</export>
<export name="slk" format="application/vnd.ms-excel">SYLK (Symbolic Link)
(*.slk)</export>
<export name="wks" format="application/vnd.ms-excel">WKS (Lotus 1-2-3)
(*.wks)</export>
</discoverer>
```

You can see information about three workbooks being returned - Store and Band Analysis, Video Sales Analysis and Annual Sales Report. Note there is NO information in the XML about how these workbooks names and descriptions should be displayed to the user, this is the function of the XSL file.

#### 5.1.4.3 Step 3 Apply XSL

XSL is the industry standard stylesheet language defined by the World Wide Web Consortium. It allows a selection of elements from an XML file to be combined with an HTML template to generate HTML output for a Web Browser. The Discoverer Viewer User Interface is entirely defined in XSL, which means that it can be customized or copied to define alternative User Interface styles using standard Web development tools such as HTML and XSL editors. The XSL and XML is combined in an XSL processor, and HTML is generated.

#### **5.1.4.4 Step 4 Generate HTML**

Given the XML above, the standard Discoverer Viewer XSL configuration file would result in HTML being generated, which is then sent back to the browser in response to the initial URL. In Discoverer Viewer, the generated HTML makes minimal demands on the browser or Internet device used. It also means that it is easy to integrate Discoverer Viewer with other Web Applications or Portals.

## 5.2 Customization Examples

This section will help you carry out a customization of the Discoverer Viewer user interface where you can either modify basic attributes such as fonts, colors and graphics or create your own look and feel with a custom user interface.

### 5.2.1 Files needed for customization

The files needed to carry out either a **Quick** or a **Full** customization can be found under the **html/** directory shown below. These directories are created during the installation of Discoverer Viewer in the root directory of the Web server.

```
disco4iv/
  disco4iv.jar
  html/
    disco4iv.dtd
    disco4iv.xml
    disco4iv.xsl
    errors.xsl
    functions.xsl
    gui_components.xsl
    page_layouts.xsl
    render_table.xsl
    scripts.xsl
    style.xsl
    viewer_config.xml
  demos/
  help/
  images/
  resources/
doc/
```

Video Stores Demo  
HTML help files  
User interface and help  
Message files in different languages  
Configuration Guide

### 5.2.2 Quick Customization

Many users want to be able to simply modify fonts and colors to fit in with their corporate standards, or to display the company logo to add branding. These global changes can be made in a single XSL stylesheet file 'style.xsl' (found in the Servlet directory *disco4iv/*) that defines special 'tags' for each style that can be modified.

You can also create a presentation template in the *viewer\_config.xml* file to specify a set of pre-defined presentation options. These presentation options enable you to control what appears in the page header, worksheet area, and page footer. End-users can choose which presentation template to use on the Presentation Options page. A presentation template called "Corporate Look" is included in *viewer\_config.xml* as an example.



Note that to modify `style.xml` and `viewer_config.xml`, you must be able to access the XSL files on the server.

Many global style changes can be made by modifying `style.xml` and `viewer_config.xml`, but the overall operation of the User Interface will remain unchanged. Another way of customizing Discoverer Viewer is by using XSL, which allows a complete customized application to be made (for more information, refer to [Section 5.2.3, "Full Customization"](#)).

### 5.2.2.1 Quick customization attributes

The `style.xml` file contains the following customizable attributes:

- Company Logo
- Text Color
- Headings
- Fonts
- Styles
- Links
- Images

The `viewer_config.xml` file enables you to specify settings for the different choices available on the Presentation Options page.

### 5.2.2.2 Quick customization steps

To complete a Quick customization, take the following steps:

1. Open the `style.xml` file in a text editor (the file is located in the Servlet directory `/disco4iv/html`).
2. Edit text within the `>angle brackets<` for each variable you want to change. For further information, read the detailed comments inside the file itself.
3. Save your changes to `style.xml` and close the file.
4. Open the `viewer_config.xml` file in a text editor (the file is located in the Servlet directory `/disco4iv/html`).
5. Copy the following section:

```
<look name="Corporate Look">
<!-- Logo-->
<component name="log" state="false"/>
```

```
.  
. .  
.  
<!-- "Show XML" Link -->  
<component name="xml" state="false"/>  
</look>
```

6. Paste the copied section immediately before the `</layout_templates>` tag.
7. Change "Corporate Look" to your preferred template name (note that the name you specify will be visible to users).
8. Change the various component values to true or false as required.
9. Save your changes to `viewer_config.xml` and close the file.

### 5.2.2.3 Example of editing the style.xsl file

Further comments on how to edit this file are included within the file itself.

To insert a logo

```
<xsl:variable name="logo_src"> </xsl:variable name>
```

is changed to

```
<xsl:variable name="logo_src"> http://www.mycompany.com/images/mylogo.gif  
</xsl:variable name>
```

To change the color of the text:

```
<xsl:variable name="text_color">#000000</xsl:variable>
```

is changed to add the appropriate color code.

## 5.2.3 Full Customization

This section provides information to help you create a fully customized user interface to the Discoverer Viewer Servlet. Creating your own 'look and feel' with a custom user interface.

It consists of the following topics:

- [Background](#)
- [Prerequisites](#)
- [File Structure](#)
- [Servlet Programming Interface](#)
- [Customizing Look and Feel - XML/XSL Example](#)

### 5.2.3.1 Background

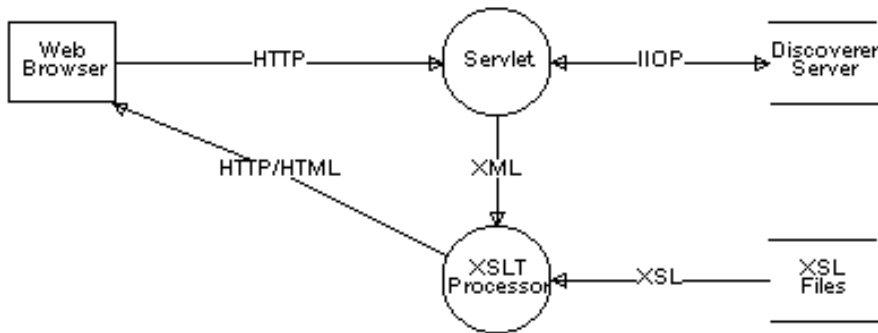
Before you start creating a custom interface to the Discoverer Viewer Servlet, it is useful to understand how it works.

The following steps describe the flow of data that occurs when a request is made:

- The request is sent from the client (Web browser) to the Servlet using HTTP. Usually the request is encoded on the URL but the Servlet also supports GET and POST actions.
- The Servlet interprets this request and retrieves the necessary data from the Discoverer Server.
- The Servlet reformats the data returned from the Discoverer Server as XML.
- This XML file is sent to the XSLT processor which transforms it into the final output (usually HTML) using XSL templates.
- The output is then streamed back to the client that originally made the request.

This model enables a clean separation between the application logic (handled by the Servlet) and the presentation logic (handled by the XSL files). It is therefore possible to completely change the appearance of the user interface by editing the XSL files.

[Figure 5-2](#) illustrates the data flow in Discoverer Viewer.

**Figure 5–2 Discoverer Viewer Data Flow**

### 5.2.3.2 Prerequisites

To be able to create your own custom user interface for Discoverer Viewer, you need the following prerequisites:

- A knowledge of HTML
- A knowledge of XML, XSLT & XPath
- Access to the XSL files on the server.
- Access to the Servlet Configuration File

### 5.2.3.3 File Structure

The Servlet Configuration File specifies the name and location of the XSL file that should be applied to the XML generated by the Servlet. This XSL file imports further XSL files which together contain all the templates that are required for the presentation logic. An XML file is also used to supply additional configuration options. These files are described below.

#### 5.2.3.3.1 disco4iv.xsl

This is the master XSL file. It performs the following functions:

- Gets the values of certain parameters set in the Servlet Configuration File.
- Gets the values of "Presentation" commands sent from the client.
- Imports the other XSL files.

- Examines the XML and chooses the most appropriate page layout for displaying the information.

#### 5.2.3.3.2 page\_layouts.xml

This file contains the definitions for each type of page that can be displayed. Each definition consists of a series of GUI components (in the order that they should appear on the page).

The following page layouts have been defined:

- Connect Page
- Reconnect Page
- Choose Oracle Applications Responsibility Page
- Options Page
- Choose Workbook Page
- Prepare to Run Query Page
- Modify Query Parameters Page
- Query Progress Page
- Data Available Page
- Export Worksheet Page
- Presentation Options Page
- Edit Layout Page

#### 5.2.3.3.3 gui\_components.xml

This file contains each of the GUI components (called by page\_layouts.xml) that are used to build a page. The following list provides a few examples of GUI components:

- Page Title
- Page Footer
- List of Worksheets
- Options Form
- Query Progress Meter

#### 5.2.3.3.4 style.xml

This file acts as a "stylesheet" by defining how certain text (such as headings) and graphics should appear. For more information, see the comments inside the file itself.

#### 5.2.3.3.5 functions.xml

This file contains a set of functions that perform common tasks. For example, building the href attribute when creating a hypertext link.

#### 5.2.3.3.6 errors.xml

This file interprets any errors, warnings or messages generated by the Discoverer Server and displays more user friendly versions.

#### 5.2.3.3.7 scripts.xml

This file contains JavaScript fragments for use by the JavaScript enhanced user interface. For more information, refer to [Section 2.8, "Utilizing the Discoverer Viewer JavaScript Enhanced User Interface"](#)).

#### 5.2.3.3.8 render\_table.xml

This file is used during the creation of tables and crosstabs. For more information, see the comments inside the file itself. Note that you will probably never need to edit this file.

#### 5.2.3.3.9 viewer\_config.xml

This file specifies:

- whether the following controls appear:
  - the Language dropdown list on the Connect dialog
  - the Applications User checkbox
  - the Show XML link at the bottom of every page
- whether to use the Javascript Enhanced user interface
- presentation templates that define a named set of presentation options (a presentation template called "Corporate Look" is included in the file as an example)

- which message files and help files to use for particular languages (the order of languages in this file determines the order in which languages appear in the Language dropdown list on the Connect dialog)

### 5.2.3.4 Servlet Programming Interface

#### 5.2.3.4.1 Receiving data from the Servlet

There are two methods that the Servlet can use to pass information to the XSL files:

- XML  
The XML contains all the information returned by the Discoverer Server as a result of the end user's request.
- XSL Parameters  
This method is used to pass entries in the disco4iv.xml file and "Presentation" commands from the URL to the XSL.

#### 5.2.3.4.2 Sending data to the Servlet

As mentioned above, the Servlet is driven by the commands it receives from the client (Web browser). These commands are sent using HTTP by calling the Servlet with parameters on the URL or by creating HTML forms that define the Servlet as the "action".

You can create your own "Presentation" commands by simply preceding it with an underscore ( \_ ) character. Any commands that begin with an underscore character are passed straight through to the XSL files as an XSL parameter. For example, specifying `_filter=sales` as a parameter on the URL results in the Servlet setting the XSL parameter "url\_filter" to "sales". You could use this parameter to filter a list of workbooks.

### 5.2.3.5 Customizing Look and Feel - XML/XSL Example

XSL customization is best shown with an example. You can use the XML and XSL fragments below to experiment with customization in a Web Browser.

### 5.2.3.5.1 Step 1 - The XML file

The data is a standard XML file, similar to the example below:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="example1.xsl"?>
<discoverer4iv version="4.1.x.x" state="login"
url="http://myserver.mycompany.com/disco/disco4iv" login_method="disco_
login">
  <account nv="NULL" aq="confirm">
    <user>video</user>
    <database>disco</database>
    <eul>
      <name>VIDEO</name>
      <workbook@name="Store and Band Analysis">
        <description>Shows sales by Store, broken into sales
bands</description>
      </workbook>
      <workbook@name="Video Sales Analysis">
        <description>General purpose analysis of the Business
</description>
      </workbook>
      <workbook@name="Annual Sales Report">
        <description>Shows yearly and quarterly sales of
products</description>
      </workbook>
    </eul>
  </account>
</discoverer4iv>
```

It starts by specifying the XML version. The second line specifies the XSL file to be applied to process the data, "example1.xsl" and the rest of the file is generated from the Discoverer Viewer. The first two lines have been added here so you can type the text into a file using a text editor and then open it in a Web Browser to see the results visually as the XSL is changed. Save the file with the extension "xml" if you want to try this (note that this is only possible with Internet Explorer 5 or above).



### 5.2.3.5.2 Step 2 - The XSL file

The XSL file “example1.xsl” looks like this:

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/W3C-xsl">
  <xsl:template match="/">
    <HTML>
      <BODY bgcolor="#FFFFFF" link="#663300" text="#000000">
        <b><i>Choose a Workbook :</i></b><P></P>
        <TABLE BORDER="2">
          <xsl:for-each select="/discoverer/account/eul/workbook">
            <TR>
              <TD width="242"><font face="Arial"><b><a href="link.htm">
                <xsl:value-of select="./@name"/></a></b></font>
              </TD>
              <TD>
                <xsl:value-of select="description"/>
              </TD>
            </TR>
          </xsl:for-each>
        </TABLE>
      </BODY>
    </HTML>
  </xsl:template>
</xsl:stylesheet>
```

### 5.2.3.5.3 Step 3 - XML+XSL = HTML

When the XML file is opened in a Web Browser (must be Internet Explorer 5 or above), it reads in the XSL stylesheet and generates HTML which looks like this:

**Figure 5–3 List of workbooks using XML + XSL example**

<i>Choose a Workbook :</i>	
<b><u>Store and Band Analysis</u></b>	Shows sales by Store, broken into sales bands
<b><u>Video Sales Analysis</u></b>	General purpose analysis of the Business
<b><u>Annual Sales Report</u></b>	Shows yearly and quarterly sales of products

Examining the XSL file shows how the HTML is generated. Again the file starts by specifying the XML version, and the second line says that this file is a stylesheet. The HTML template starts with the <HTML> tag on line 4.

```
<BODY bgcolor="#FFFFFF" link="#663300" text="#000000">
```

This line defines the colors to be used.

```
<b><i>Choose a Workbook:</i></b><P></P>
```

This is just HTML, it sets a bold italic font and inserts the text “Choose a workbook”

```
<TABLE BORDER="2">
```

```
<xsl:for-each select="/discoverer/account/eul/workbook">
```

Now an HTML table is started, with a 2 line border. The next line is the first real XSL to appear - it means:

*Go through the XML data file and for each workbook tag perform all the following steps until you reach the end tag: </xsl:for-each>*

So for every workbook that appears in the XML file the following XSL is processed, and a row is inserted into the HTML table for every workbook found:

```
<TR>
  <TD width="242"><font face="Arial"><b><a href="link.htm">
    <xsl:value-of select="@name" /></a></b></font>
  </TD>
  <TD>
    <xsl:value-of select="description" />
  </TD>
</TR>
```

<TR> starts a new row in the table, and <TD ... > defines the table data that is to be inserted for the first column. The width of the column is set to 242 pixels and the font is set to Arial. Next a hyperlink is defined that will go to the file “link.htm” when clicked. In fact in Discoverer Viewer this link target is generated dynamically, but it is shown as a static link here for simplicity.

```
<b><a href="link.htm"><xsl:value-of select="@name" /></a></b></font>
```

This XSL line inserts the text from the XML file for the name attribute for each workbook element.

```
<TD>
  <xsl:value-of select="description" />
</TD>
```

These lines define the second column in the HTML table and insert the text for the workbook description using the <description> element in the XML file. So each row in the HTML table will contain the workbook name, made into a link to click on,

and the workbook description as text. Since there are three workbooks in the XML file, there will be three rows in the table.

Note that this example is not exactly how the Discoverer Viewer shows the list of workbooks, it has been simplified here for clarity, but it illustrates how the XSL controls the appearance of the output. Also note that in Discoverer Viewer, the XML and XSL are combined in the XSL processor on the middle tier, and not in the Web Browser.

#### 5.2.3.5.4 Step 4 - Customize the XSL

We can modify the XSL file to appear as follows:

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <HTML>
      <BODY bgcolor="#FFFFFF" link="#663300" text="#000000">
        <b><i>Choose a Workbook :</i></b><P></P>
        <TABLE BORDER="0">
          <TR>
            <TD width="500" height="100" background="../../images/disco_
banner.gif">
              <font face="Arial"><b>Performance Reports</b></font>
            </TD>
          </TR>
        </TABLE>
        <TABLE border="0">
          <xsl:for-each select="/discoverer/account/eul/workbook">
            <TR>
              <TD width="242">
                <a href="link.htm">
                  
                    <xsl:attribute name="alt">
                      <xsl:value-of select="description"/>
                    </xsl:attribute>
                  </img>
                </a>
              </TD>
              <TD>
                <font face="Arial">
                  <xsl:value-of select="@name"/>
                </font>
              </TD>
              <TD>

```

```

        <xsl:value-of select="description" />
    </TD>
</TR>
</xsl:for-each>
</TABLE>
</BODY>
</HTML>
</xsl:template>
</xsl:stylesheet>

```

When this file is combined with the same XML, it appears as:

**Figure 5-4** List of workbooks using the same XML with a modified XSL stylesheet:



Now the appearance of the User Interface is completely different, as it takes on a more graphical look and feel. Instead of text links there are graphical buttons for running the reports, each with a dynamic 'tool tip' that pops up when you position the mouse over the button.

The XSL file is now different:

```

<TABLE BORDER="0">
  <TR>
    <TD width="500" height="100" background="../../images/disco_
    banner.gif">
      <font face="Arial"><b>Performance Reports</b></font>

```

```

    </TD>
  </TR>
</TABLE>

```

These lines create a table and insert a graphic and the heading “Performance Reports”

```

<TABLE BORDER="0">
  <xsl:for-each select="/discoverer/account/eul/workbook">
    <TR>

```

This starts the main table that the workbook names will be displayed in, as before, but now there is no border around the table and the rows are defined differently:

```

    <TD width="242">
      <a href="link.htm">
        
          <xsl:attribute name="alt">
            <xsl:value-of select="description"/>
          </xsl:attribute>
        </img>
      </a>
    </TD>

```

The first table data column is defined as a hyperlink again, but this time with the image “start1.gif” as an image, rather than a text link. To get a “tooltip” to appear over an image the HTML “ALT” attribute is used.

Normally the ALT attribute is used with a simple text string:

```



```

but since we want the tool tip to be dynamic we generate the ALT tag by getting the text from the <description> element in the XML file. The <xsl:attribute> tag is used to do this.

Finally:

```

<TD>
  <font face="Arial">
    <xsl:value-of select="@name"/></font>
  </TD>

```

The second column in the table simply selects the name of the workbook to display, by using XSL to get it from the XML file as before.

## 5.2.4 Video Stores Demo - an example of an alternative user interface

The Video Stores demo is installed with Discoverer Viewer and can be viewed following completion of a number of setup steps.

Once Discoverer Viewer has been fully installed, you can configure your Web server to run the Video Stores demo.

The Video Stores demo illustrates how the Discoverer Viewer user interface could be customized for an imaginary Video Store chain. You will be able to see the details which lie behind the user interface.

### 5.2.4.1 What is installed

During installation, the Video Stores demo directory **demos/video**, its sub directories and files are created. The **demos/video** directory is installed under the `<i>SUITES_HOME>/Apache/Apache/htdocs/disco4iv/html` directory.

In the Video Stores demo directory **demos/video** the following files are installed:

- *disco4iv.xml*  
You may need to edit this file to ensure that it refers to the correct session name.  
*The **disco4iv.xml** file contains configuration information (See **disco4iv.xml** for details).*
- *video.xsl*  
You can edit this file to create the new user interface.  
*This file contains all the definitions which control the 'look and feel' of the user interface, such as the fonts, images, background, and spatial details.*
- *render\_table.xsl*  
You will probably never need to edit this file.  
*This file is used during the creation of tables and crosstabs. For more information, see the comments inside the file itself.*
- Workbooks - These are used in the Video Stores Demo (See [Section 5.2.4.2, "Set up steps"](#))
  - **Food and Drink.dis**
  - **Games.dis**
  - **Movies.dis**

The **demos/video/images** directory contains all the images used in the Video Stores user interface. This is where you would place any new images the user interface needs to use.

#### 5.2.4.2 Set up steps

You need to carry out the following steps:

- Create a Video Stores Demo database user  
(See the *Oracle Discoverer Administration Guide* for details about how to create a new database user).
- Install the Video Stores Tutorial into the Video Stores Demo user created in the previous step.  
(See the *Oracle Discoverer Administration Guide*, section on installing the Tutorial for details).
- Using Discoverer Plus, save the three Video Stores Demo workbooks into the database as the Video Stores Demo user.
- Edit the **video.xml** file to point the **username**, **password** and **database name** to the newly created Video Stores Demo user. The **video.xml** file is currently set up to point to a username **us=videodemo**, password **ps=videodemo** and database **db=disco**.  
This is the only change that you need to make to **video.xml**.

#### 5.2.4.3 Set up Web Server

The Web server needs to be configured with the 'Zone' name. You need to set up a new zone for the Video Stores Demo to specify the location of the `/demos/video/disco4iv.xml` (Video Stores Demo configuration file)

#### 5.2.4.4 Run the Video Stores demo

Open a Web browser and enter the following URL:

- `http://yourmachineURL/'Your new Video Demo Zone'`

The Video Stores demo will be displayed:

Video Store Analysis		
Product Type = 'MOVIE'		
Page: 1 of 1		
	Profit SUM	Sales SUM
▶ Product Category		
▶ Action	\$7,006	\$10,616
▶ Comedy	\$2,731	\$4,193
▶ Horror	\$15,179	\$22,904
▶ Romantic Comedy	\$4,873	\$7,308
▶ Sci-Fi	\$2,767	\$4,177
▶ Thriller	\$9,076	\$13,775
Sum	\$41,633	\$62,973

#### 5.2.4.5 Modify the video.xsl file

The video.xsl file located in the <i>iSUITES\_</i>HOME>/Apache/Apache/htdocs/disco4iv/html/demos/video directory specifies the 'look and feel' for the Video Stores Customization of Discoverer Viewer.

For example, you can edit the *video.xsl* file to replace any of the images displayed, including the background. You can also specify the workbooks you want to display in the Demo.

The ability to alter almost everything displayed by editing the video.xsl file illustrates how you can create not only a look and feel for your organization, but also define unique views of your data.



## 5.2.5 Other customization examples

### 5.2.5.1 Logging on automatically

You can modify the `page_layouts.xml` file to enable Discoverer Viewer users to log on automatically (i.e. bypassing the login screen) without exposing the password as a URL argument. Note that users must have a JavaScript-enabled browser for this mechanism to work.

#### 5.2.5.1.1 Step 1: Add a new template to `page_layouts.xml`

Add the following lines to the `page_layouts.xml` file:

```
<xsl:template name="layout_auto_login">
  <html>
    <head><title>Auto Login</title></head>
    <body>
      <form method="post" name="autoLogin"
action="{/discoverer/request/@source}">
        <!-- Specify the User Name in the value field below -->
        <input type="hidden" name="us" value="username"/>
        <!-- Specify the Password in the value field below -->
        <input type="hidden" name="pw" value="password"/>
        <!-- Specify the Database in the value field below -->
        <input type="hidden" name="db" value="database"/>
        <!-- Specify the Language Code in the value field below (en is English)
-->
        <input type="hidden" name="nls1" value="en"/>
        <!-- To show a list of Database Workbooks, leave the following two value
fields set to "dwb" otherwise, set them to "swb" for Scheduled Workbooks. -->
        <input type="hidden" name="in" value="dwb"/>
        <input type="hidden" name="_in" value="dwb"/>
        <!-- The following line requests the user is logged in using the default
method (as set in viewer_config.xml) -->
        <input type="hidden" name="lc" value="false"/>
      </form>
      <script language="JavaScript">
        document.autoLogin.submit();
      </script>
    </body>
  </html>
</xsl:template>
```

#### 5.2.5.1.2 Step 2: Include username, password and database

Modify the `layout_auto_login` template to include the username, password and database information as indicated in the comments.

#### 5.2.5.1.3 Step 3: Modify `disco4iv.xsl`

Search for the following text in `disco4iv.xsl`:

```
<xsl:call-template name="layout_connected" />
```

Replace the above text with the following text:

```
<xsl:call-template name="layout_auto_login" />
```

## 5.2.6 Customization Summary

XML and XSL provides a powerful customization environment for Discoverer Viewer. Simple global formatting changes are made to a single style sheet file, while more sophisticated custom applications can be made by using completely alternative XSL processing. In this way Discoverer Viewer provides the way to integrate Business Intelligence directly into other Web applications.

---

---

## SSL Support in Oracle9iAS Discoverer

This chapter explains how to configure Discoverer Plus and Discoverer Viewer to work with the Secure Sockets Layer (SSL) protocol.

The topics include:

- [Getting more information](#)
- [SSL and Discoverer](#)
- [What is SSL and why should I use it?](#)
- [How does SSL work?](#)
- [About Public Key Encryption Technology](#)
- [SSL authentication and certificates](#)
- [What steps do I need to follow?](#)
- [Configuring Discoverer Plus to use SSL](#)
- [Configuring Discoverer Viewer using HTTPS](#)

### 6.1 Getting more information

This chapter contains edited extracts from the Visibroker SSL Pack 3.3 Programmer's Guide.

For more information about Visibroker's SSL features, refer to the Borland Inprise Corporation Internet site at:

<http://www.borland.com/techpubs/books/addons/vbssl/vbssl33/index.html>

## 6.2 SSL and Discoverer


Discoverer Plus and Discoverer Viewer can be configured to use the SSL protocol:

- In Discoverer Viewer, SSL support is provided using the SSL capabilities of the Oracle HTTP Server powered by Apache.
- In Discoverer Plus, SSL support is provided using the SSL capabilities of the HTTP Server and Visibroker Gatekeeper. With Discoverer Plus, you can encrypt the data between the client and Visibroker Gatekeeper when the client is outside the Server's firewall.

## 6.3 What is SSL and why should I use it?

The SSL (Secure Sockets Layer) protocol is an industry standard protocol to establish secure connections between clients and servers. The SSL protocol enables sensitive data to be transmitted over an insecure network, such as the Internet, by providing the following security features:

- **Authentication:** A client can determine a server's identity and be certain that the server is not an impostor. Optionally, a server can also authenticate the identity of its client applications.
- **Privacy:** Data passed between the client and server is encrypted so that if a third party intercepts their messages, the third party will not be able to unscramble the data.
- **Integrity:** The recipient of encrypted data will know if a third party has corrupted or modified that data.

When SSL is enabled, a closed padlock () or other equivalent symbol (browser dependent) appears in the browser's status bar, and the web page's URL starts with "https://" instead of "http://".

## 6.4 How does SSL work?

SSL provides a secure communications channel between two parties by providing several security services at the transport layer level using a combination of public key and symmetric key technology. Usually these parties are a web browser and a web (HTTP) server. However, since SSL operates at the transport layer, it is application independent. Protocols such as Telnet, FTP (File Transfer Protocol) and HTTP (HyperText Transfer Protocol) can therefore be layered on top of it.

## 6.5 About Public Key Encryption Technology

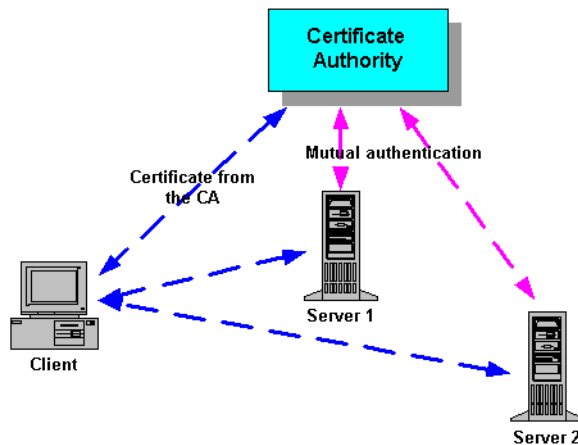
Public key encryption uses two different keys to encrypt and decrypt messages, a public key and a (secret) private key. The two keys are used such that a message encrypted with a private key can only be decrypted with the corresponding public key, and a message encrypted with a public key can only be decrypted with the corresponding private key.

## 6.6 SSL authentication and certificates

Before establishing a secure connection, the server (and optionally the client) is authenticated. This is to make sure that one is communicating with the correct server. Authentication is done through digital certificates, or public-key certificates. Certificates used in SSL are defined by an ITU (International Telecommunications Union) standard called X.509, (see figure below).

To prove that the server is the legitimate owner of the certificate, SSL requires that certificates are digitally signed by a trusted Certificate Authority (CA), for example VeriSign, Inc. You can think of a certificate as a passport and the Certificate Authority as a government passport office. You trust that the government passport office has verified the identity of the passport holder before issuing a passport to him or her.

**Figure 6–1** Certificate verification by a Certificate Authority



## 6.7 What steps do I need to follow?

### 6.7.1 To configure SSL with Discoverer Plus

1. Follow the steps in [Section 6.8, "Configuring Discoverer Plus to use SSL"](#).
2. Make sure that you have the jar file `x509cert.jar` installed on your client machines. Follow the steps in [Section 6.8.6, "Installing the x509cert.jar file"](#).
3. Make sure that you have the required shared library file, see [Section 6.8.7, "Installing the required shared library files"](#).
4. Enable SSL in your Discoverer start pages by following the steps in [Section 6.8.8, "Enabling SSL in Discoverer Plus start pages"](#).

### 6.7.2 To configure SSL with Discoverer Viewer

1. Follow the steps in [Section 6.9, "Configuring Discoverer Viewer using HTTPS"](#).
2. Enable SSL in your Discoverer pages using the HTTPS protocol, see [Section 6.9.5, "Enabling SSL in Discoverer Viewer"](#).

**Note:** Discoverer Viewer uses HTTP and HTTPS protocols that are firewall compliant and do not require Visibroker Gatekeeper.

## 6.8 Configuring Discoverer Plus to use SSL

### 6.8.1 How does Discoverer Plus support SSL?

To provide support for SSL, Discoverer Plus uses Visibroker Gatekeeper (for more information, refer to [Section 7.5, "About Visibroker Gatekeeper"](#)).

Discoverer Plus supports the SSL mode of communication only for clients that are outside the server's firewall and hence connect to Discoverer Server components through Visibroker Gatekeeper. Clients in the intranet (i.e. inside the Server's firewall) do not use SSL when talking to the Discoverer Server components.

You have to configure Visibroker Gatekeeper to use SSL with Discoverer Plus.

### 6.8.2 How do I configure Visibroker Gatekeeper to use SSL?

To configure Visibroker Gatekeeper to use SSL, follow these steps:

1. Install Visibroker Gatekeeper on either your HTTP server machine or a different machine, depending on your firewall configuration, (see [Section 6.8.3, "Installing Visibroker Gatekeeper to work with SSL"](#)).
2. You need to get SSL certificates from a Certificate Authority (see [Section 6.8.4, "Getting SSL certificates"](#)).

If you have already obtained certificates for your enterprise or organization, these can be reused with Visibroker Gatekeeper.

3. When you have got the SSL certificates from the Certificate Authority, you need to install these certificates on Visibroker Gatekeeper (see [Section 6.8.5, "Installing SSL certificates"](#)).

### 6.8.3 Installing Visibroker Gatekeeper to work with SSL

SSL support in Discoverer works through single and multiple firewalls.

- When using single firewalls, you can run Visibroker Gatekeeper on any port. If you run Visibroker Gatekeeper on port 443 (the standard SSL port) you do not have to open your firewall (known as 'drilling a hole' in a firewall) for SSL communication. If you run Visibroker Gatekeeper on port other than 443, you must open your firewall on this port for SSL communication
- When using multiple firewalls, you should run Visibroker Gatekeeper on port 443.

NOTE: If you by running Visibroker Gatekeeper on port 443, (by convention, the standard SSL port), SSL proxies must only allow communication to port 443 also.

To enable SSL in Discoverer, you must configure Visibroker Gatekeeper to use SSL. When configuring Visibroker Gatekeeper, you have two options:

1. If your HTTP server is not using port 443 for SSL communication, you can install and run Visibroker Gatekeeper on the HTTP server machine. Then, configure the Visibroker Gatekeeper Exterior Port to 443.

See [Section 7.10.2, "Running Visibroker Gatekeeper on the HTTP Server"](#) and [Section 7.10.5, "Changing the default Visibroker Gatekeeper port"](#).

2. If your HTTP server is using port 443 for SSL communication, you must install Visibroker Gatekeeper on a different machine. Then, configure the Visibroker Gatekeeper Exterior Port to 443.

See [Section 7.10.3, "Running Visibroker Gatekeeper on a different Server"](#) and [Section 7.10.5, "Changing the default Visibroker Gatekeeper port"](#).

## 6.8.4 Getting SSL certificates

**NOTE: You can omit this step if you already have SSL certificates for your enterprise.**

Discoverer Plus provides the `vbcertrq` tool to automate the certificate request process. After specifying the required information, the `vbcertrq` tool generates three files:

- a certificate request file
- a file containing the values for the certificate request
- a file containing the private key.

After you have generated the files, you should submit the certificate request to a Certificate Authority (CA). The procedure for submitting your certificate request to a CA is determined by the certificate authority you are using. If you are using a CA that is internal to your organization, contact your system administrator for instructions. If you are using a commercial CA, contact them for instructions on submitting your certificate request.

To run the `vbcertrq` tool:

1. Go to `<ORACLE_806_HOME>/vbroker/bin` directory.
2. Type `vbcertrq` at command line prompt.



The Certificate Request Tool dialog appears.

**Figure 6–2** Certificate Request Tool dialog

The screenshot shows a dialog box titled "Certificate Request Tool" with a menu bar containing "File" and "Help". The dialog is organized into three main sections:

- Certificate Information:** This section contains several text input fields: "Common-Name:", "Organization:", "Country:" (with a dropdown menu currently showing "UNITED STATES"), "Email:", "Phone:", "Organization Unit:", "Locality/City:", and "State/Province:".
- Password:** This section contains two text input fields: "Type Password:" and "Confirm Password:".
- Output File Name:** This section contains a text input field and a "Browse" button.

At the bottom of the dialog, there is a "Key Size:" text input field and a "Generate Certificate" button.

3. Specify the certificate information (see [Section 6.8.4.1, "Specifying the certificate information"](#)).
4. Specify the password (see [Section 6.8.4.2, "Specifying private key passwords"](#)).
5. Specify the key size (see [Section 6.8.4.4, "Specifying key size"](#)).
6. Specify the output file name (see [Section 6.8.4.3, "Specifying output file name"](#)).
7. Click the Generate Certificate Request button to create the three files.

#### 6.8.4.1 Specifying the certificate information

The certificate information is used to create your unique name. Required information is shown in red in the vbcertq tool dialog. Specify the following information:

- **Common-Name:** Either the name of an individual or a hostname, depending on whether the certificate is intended for a person or a server. Required.
- **Organization:** The organization or company to which the certificate recipient belongs. Required.
- **Country:** The name of the country in which the certificate recipient resides. Required.
- **Email:** The email address of the certificate recipient or the person to contact for more information about this user. Optional.
- **Phone:** The user's phone number. Optional.
- **Organization Unit:** The user's department name. Optional.
- **Locality/City:** The city in which the user resides. Optional.
- **State/Province:** The state in which the user resides. Optional.

#### 6.8.4.2 Specifying private key passwords

You are required to specify a password that will be used to encrypt your private key. You will need to supply this password before you use your private key. The password prevents unauthorized use of your private key.

Your password is case-sensitive and can contain any character you can type on a keyboard. The password can be from six to 99 characters long. As with all passwords, avoid choosing common words that are susceptible to dictionary-style attacks. Passwords that contain numbers, blanks, and other characters are more secure than common words or phrases.

To specify the private key password, fill in the following fields:

- **Type Password:** A password to encrypt the private key. It can be any mix of alphanumeric characters.
- **Confirm Password:** Retype the password to confirm it.

### 6.8.4.3 Specifying output file name

The names of the three files generated by the `vbcertrq` tool are based on what you specify in the Output File Name field. For example, if you specify `mycertreq` in the file, the following files are created:

- `mycertreq` contains the certificate request
- `mycertreq#####`, where `#####` is a unique sequence of numbers, contains the values of the fields in the certificate request
- `mycertreqKEY` contains the private key in password encrypted PKCS #8 format

### 6.8.4.4 Specifying key size

You can specify the number of bits your key will contain in the Key Size field. The greater the number of bits in your private key, the greater the security of your encrypted data. You can specify any size between 360 and 2048 bits.

**NOTE: United States federal export regulations limit keys that are used in exported software to 512 bits.**

### 6.8.4.5 Using an S/MIME certificate chain

Certificate authorities are able to provide your certificate chain in a number of formats. Visibroker Gatekeeper expects the commonly used format of separate X.509 certificates. If you have a certificate chain in an S/MIME (PKCS #7) format, you must separate the certificates before they can be used with Visibroker Gatekeeper.

To separate S/MIME certificates:

1. Navigate to the `<ORACLE_806_HOME>/vbroker/ssleay` directory and type the following at the command prompt to break the certificate chain file into its certificates:

```
> ssleay pkcs7 -print_certs -inform DER -in cert-chain-file -out cert-file
```

The output file contains a list of one or more certificates and your private key. Each certificate is delimited with the following statements:

```
-----BEGIN CERTIFICATE-----
```

```
-----END CERTIFICATE-----
```

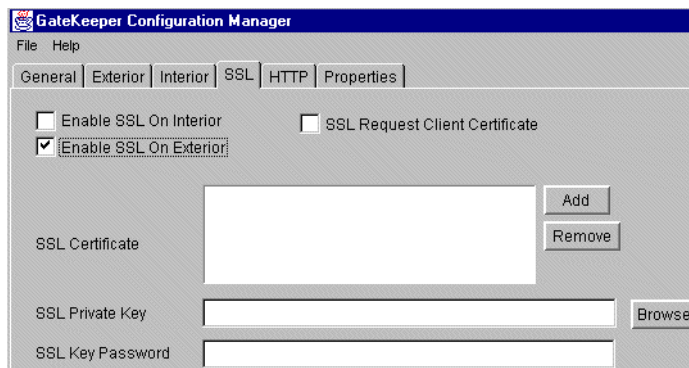
If there is more than one certificate, the first one is yours and each subsequent certificate represents a certificate authority up to a root (such as Thawte).

## 6.8.5 Installing SSL certificates

When you have all the necessary certificates, you need to install the certificates (starting with your certificate to the root certificate authority) on Visibroker Gatekeeper.

1. Run gkconfig from <ORACLE\_806\_HOME>/vbroker/bin.
2. Choose **File | Open** and open the file <iSUITES\_HOME>/Apache/Apache/htdocs/discwb4/applet/gatekeeper.properties.
3. Click on the **SSL** tab to display the SSL configuration details, (see [Figure 6–3](#) below).

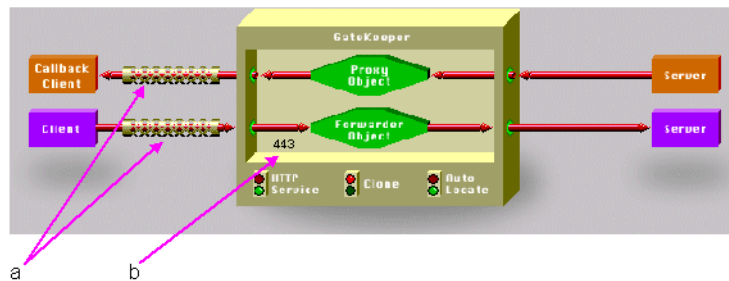
**Figure 6–3 Gatekeeper Configuration Manager SSL tab**



4. Check the Enable SSL on Exterior check box.

The SSL Certificate, SSL Private Key, and SSL Key Password properties become active, and SSL active indicators (also known as "mufflers") on the Visibroker Gatekeeper configuration model show where SSL is enabled, (see [Figure 6–4](#) below).

**Figure 6–4 Gatekeeper Configuration Manager with 'SSL enabled in Exterior'**



Key to [Figure 6–4](#).

- a - SSL active indicators on the exterior connection indicating where SSL is enabled.
- b - Note also that the Exterior Port (configured on the Exterior tab in the Gatekeeper Configuration Manager) has been set to 443, which is the standard SSL port.

**5. Click Add.**

An Open dialog box appears.

**6. Navigate to the file containing an SSL certificate.**

**7. Select the file, and then click Open.**

The full path name of the file appears in the SSL Certificate box.

**8. Repeat Steps 5 to 7, as necessary, to build a certificate chain.**

You can specify one or more certificate files. There is one certificate per file. However, you can specify a list of certificate files. The list acts as a chain of certificates. You must start the chain with Visibroker Gatekeeper's certificate followed by the issuer's certificate and continue, if there are more, up to the root.

**9. Click the Browse button beside the SSL Private Key field to specify an SSL private key (required for Visibroker Gatekeeper's SSL adapters to operate successfully).**

An Open dialog box appears.

**10. Navigate to the file containing the SSL private key.**

**11. Select the file and click Open.**

The full path name for the file appears in the SSL Private Key box, setting the private key for the exterior SSL adapters.

**12. Specify an SSL key password.**

If your SSL private key is encrypted, enter a password for the private key for the exterior and interior SSL adapters. If you used the vbcertreq tool to generate a certificate request, you specified this password there.

An asterisk appears for each character of the password you enter. The password itself does not appear in the SSL Key Password box.

**13. Choose File | Save.**

**14. Choose File | Exit.**

## 6.8.6 Installing the x509cert.jar file

To use SSL, Discoverer clients must have the jar file x509cert.jar installed.

To install x509cert.jar on UNIX clients, follow these steps:

1. Copy the x509cert.jar file from the <iSUITES\_HOME>/Apache/Apache/htdocs/discwb4/util directory on the HTTP server machine to the <Java-Plugin>/lib/ directory on the client machines (where <Java-Plugin> is the directory in which the Sun Java plugin is installed).

To install x509cert.jar on Windows NT clients, follow these steps:

1. Copy the x509cert.jar file from the <iSUITES\_HOME>/Apache/Apache/htdocs/discwb4/util directory on the HTTP server machine to the \Program Files\Oracle\JInitiator 1.1.7.30\lib directory on the client machines.

Note that you can download the above file from <http://server.com/discwb4/util/x509cert.jar>.

## 6.8.7 Installing the required shared library files

To use an SSL enabled ORB on UNIX clients, client machines must have the file libvbj30ssl.so. To install this file, follow these steps:

1. Copy the libvbj30ssl.so file from the <iSUITES\_HOME>/Apache/Apache/htdocs/discwb4/util directory on the HTTP server machine to the <Java-Plugin>/lib/ directory on the client machines (where <Java-Plugin> is the directory in which the Sun Java plugin is installed).

To use an SSL enabled ORB on Windows NT clients, client machines must have the DLL files vbj30ssl.dll and/or vbrnissl.dll installed. To install the DLL files, follow these steps:

1. Copy one or both of the following files from the specified directory on the HTTP server machine to the \winnt directory on the client machines:
  - <iSUITES\_HOME>/Apache/Apache/htdocs/discwb4/util/vbj30ssl.dll - for JInitiator users.
  - <iSUITES\_HOME>/Apache/Apache/htdocs/discwb4/util/vbrnissl.dll - for Internet Explorer users.

Note that you can download the above files from <http://server.com/discwb4/util/vbj30ssl.dll> and <http://server.com/discwb4/util/vbrnissl.dll> respectively.

## 6.8.8 Enabling SSL in Discoverer Plus start pages

To enable SSL in a Discoverer start page, add the following URL parameter to the URL for the start page:

```
ORBenableSSL=yes
```

For example:

```
http://server.com/discwb4/html/english/welcome.htm?ORBenableSSL=yes
```

**NOTE:** If you have multiple URL parameters in the same URL, separate the URL parameters with the ampersand character '&'.

For example:

```
http://server.com/discwb4/html/english/welcome.htm?ORBenableSSL=yes&ORBalwaysProxy=yes
```

**NOTE:** You cannot use the URL parameters ORBenableSSL and ORBalwaysTunnel at the same time.

## 6.8.9 Using SSL inside firewalls

By default, SSL support works only for clients outside your firewall. You can enable SSL inside firewalls by setting these URL parameters as follows:

- ORBalwaysProxy=yes - connect using IIOP Proxying (see also [Section 7.9.2, "How to use a specific connection method"](#)).
- ORBenableSSL=yes - enable SSL when requesting the URL.

**NOTE: When using SSL inside firewalls, there is no restriction on the SSL port.**



## 6.9 Configuring Discoverer Viewer using HTTPS

To make sure that your Discoverer Viewer documents are secure, and to get your browser to acknowledge a “secure” page, follow these steps:

- Install SSL certificates on your HTTP Server and enable SSL for Discoverer Viewer Servlets. For more information, refer to the *Oracle9i Application Server Installation Guide*.
- Edit the following files to set up HTTPS:
  - the httpd.conf file (see ["Editing the httpd.conf file to set up HTTPS"](#))
  - the disco4ivviewer.properties file (see ["Editing the disco4ivviewer.properties file to set up HTTPS"](#))
  - the disco4iv.xml file (see ["Editing the disco4iv.xml file to set up HTTPS"](#))
  - the viewer\_config.xml file (see ["Editing the viewer\\_config.xml file to set up HTTPS"](#))

### 6.9.1 Editing the httpd.conf file to set up HTTPS

Make changes to the httpd.conf file as follows:

1. Open the httpd.conf file for editing.

The httpd.conf file is located in <i>SUITES\_HOME>/Apache/Apache/conf.

2. Add the following line in the Aliases section of the file to specify the location of the disco4iv directory (if the line does not exist already):

```
Alias /disco4ivfiles/ "<i>SUITES_HOME>/Apache/Apache/htdocs/disco4iv/"
```

where:

- disco4ivfiles is any name you choose
- <i>SUITES\_HOME> is the location in which Oracle9i Application Server components are installed

For example:

#### (httpd.conf)

```
# Aliases: Add here as many aliases as you need (with no limit). The format is
# Alias fakename realname
#
# Note that if you include a trailing / on fakename then the server will
```

```
# require it to be present in the URL. So "/icons" isn't aliased in this
# example, only "/icons/"..
#
Alias /icons/ "/usr/home/iAS_Home/Apache/icons/"
Alias /disco4ivfiles/ "/usr/home/iAS_Home/Apache/Apache/htdocs/disco4iv/"
```

## 6.9.2 Editing the disco4iviewer.properties file to set up HTTPS

Make changes to the disco4iviewer.properties file as follows:

1. Open the disco4iviewer.properties file for editing.

The disco4iviewer.properties file is located in <iSUITES\_HOME>/Apache/Jserv/etc.

2. Change the path to the disco4iv.xml file from the HTTP protocol to the FILE protocol.

For example:

### (disco4iviewer.properties - Before change)

```
servlet.Viewer.initArgs=config=http://mywebserver.company.com:7777/disco4iv/html/disco4iv.xml
```

```
servlet.viewer.initArgs=config=http://mywebserver.company.com:7777/disco4iv/html/disco4iv.xml
```

### (disco4iviewer.properties - After change)

```
servlet.Viewer.initArgs=config=file:///usr/home/iAS_Home/Apache/Apache/htdocs/disco4iv/html/disco4iv.xml
```

```
servlet.viewer.initArgs=config=file:///usr/home/iAS_Home/Apache/Apache/htdocs/disco4iv/html/disco4iv.xml
```

## 6.9.3 Editing the disco4iv.xml file to set up HTTPS

Make changes to the disco4iv.xml file as follows:

1. Open the disco4iv.xml file for editing.

The disco4iv.xml file is located in <iSUITES\_HOME>/Apache/Apache/htdocs/disco4iv/html.

2. Change the xsl\_path argument to include the FILE protocol and the full path to the disco4iv.xsl file.

**NOTE: When determining the `xsl_path` argument, the FILE protocol and not the HTTPS protocol is used (see example below).**

For example:

**(disco4iv.xml - Before change)**

```
<argument name="xsl_path" type="href">disco4iv.xsl</argument>
```

**(disco4iv.xml - After change)**

```
<argument name="xsl_path" type="href">file:///usr/home/iAS_
Home/Apache/Apache/htdocs/disco4iv/html/disco4iv.xsl
</argument>
```

## 6.9.4 Editing the viewer\_config.xml file to set up HTTPS

Make changes to the viewer\_config.xml file as follows:

1. Open the viewer\_config.xml file for editing.

The viewer\_config.xml file is located in `<i>SUITES_HOME</i>/Apache/Apache/htdocs/disco4iv/html`.

2. Change the value of the `base_dir` attribute of the configuration element to the full secure path to the disco4iv.xsl file.

For example:

**(viewer\_config.xml - Before change)**

```
<configuration application="discoverer_viewer"
base_dir="." image_dir="images">
```

**(viewer\_config.xml - After change)**

```
<configuration application="discoverer_viewer"
base_dir="https://mywebserver.company.com:7777/disco4ivfiles/html/"
image_dir="images">
```

## 6.9.5 Enabling SSL in Discoverer Viewer

To enable SSL in Discoverer Viewer URLs, prefix URLs with **https**, rather than **http**.

For example:

```
https://company.com/Discoverer4i/Viewer/<Viewer parameters>
```



---

---

# Configuring Discoverer Plus for Firewall Support

This chapter explains how to configure Discoverer Plus to work with corporate firewalls, using Visibroker Gatekeeper version 3.4.

**Note:** This chapter does not apply to Discoverer Viewer. Discoverer Viewer uses HTTP and HTTPS protocols that are firewall compliant and do not require Visibroker Gatekeeper.

This chapter covers the following topics:

- [Terminology](#)
- [Getting more information](#)
- [Before you start](#)
- [About Internet Firewalls](#)
- [About Visibroker Gatekeeper](#)
- [Choosing a Gatekeeper configuration for Discoverer](#)
- [Using Visibroker Gatekeeper with an IIOP Proxying Configuration](#)
- [Using Visibroker with a HTTP Tunnelling Configuration](#)
- [Configuring the Discoverer client](#)
- [Configuring Visibroker Gatekeeper](#)

## 7.1 Terminology

- HTTP Server machine refers to the machine on which the HTTP Server software resides. This machine is commonly referred to as a Web Server.
- HTTP Server software refers to the HTTP software itself (e.g. Oracle HTTP Server).
- Discoverer Services refers to the Locator, Preference and Session components.
- Proxy Server refers to a machine that acts as an intermediary between a workstation and the Internet. Proxy servers provide security, administrative control, and a caching service.

## 7.2 Getting more information

For more information about Visibroker Gatekeeper, refer to the Borland Inprise Corporation Internet site at:

<http://www.inprise.com/techpubs/books/vbgate/vbgate33/gatekeeper/contents.html>

## 7.3 Before you start

Before you can decide how to deploy Discoverer Plus through your organization's firewall, you need to work with your Network Administrator to find out what the firewall policy is in your organization. You might also take into consideration the firewall policies in organizations that wish to access your Discoverer data across the Internet.

When discussing your organization's firewall policy with your Network Administrator, you need to be aware of the following basic concepts:

- Firewalls and secure TCP/IP ports - read [Section 7.4, "About Internet Firewalls"](#) to find out about firewalls, Demilitarized Zones (DMZs), and secure TCP/IP ports.
- Visibroker Gatekeeper - read [Section 7.5, "About Visibroker Gatekeeper"](#) to find out what Visibroker Gatekeeper is and why it is used.
- Firewall topologies - read [Section 7.6.1, "About firewall configurations"](#) and [Section 7.6.2, "DMZ configuration examples"](#) to find out how networks are physically organized.

- Visibroker Gatekeeper configurations - read [Section 7.6.3, "Which Gatekeeper configuration do I choose?"](#) to find out what network protocols are allowed by your firewall policy.

## 7.4 About Internet Firewalls

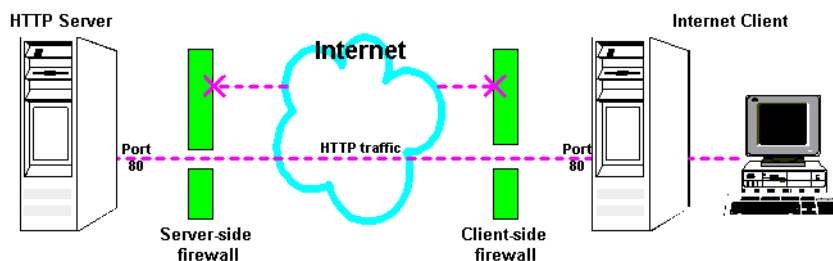
This section introduces the basic concepts of Internet firewalls and TCP/IP ports.

### 7.4.1 What is an Internet Firewall?

An Internet firewall is one system or a group of several systems put in place to enforce a security policy between the Internet and an organization's network.

In other words, an Internet firewall is an electronic 'fence' around a network to protect it from unauthorized access.

**Figure 7-1** A typical Internet connection with a Client-side and Server-side firewall



Typically, an organization using a Web Server machine that communicates across the Internet has a firewall between its HTTP Server machine and the Internet. This is known as a Server-side firewall. Other organizations (or remote parts of the same organization) connecting to this Web Server machine typically have their own firewall, known as a Client-side firewall. Information that conforms to the organization's firewall policy is allowed to pass through the firewalls enabling server machines and client machines to communicate.

### 7.4.2 What is a Demilitarized Zone?

A Demilitarized Zone (DMZ) is a firewall configuration that provides an additional level of security. In this configuration, the DMZ is an extra network placed between a protected network and the Internet. Resources residing within the DMZ are

visible on the public Internet, but are secure. DMZs typically hold servers that host a company's public web site, File Transfer Protocol (FTP) site, and Simple Mail Transfer Protocol (SMTP) server.

Refer to [Section 7.6.1, "About firewall configurations"](#) for more information about DMZs.

### 7.4.3 About secure ports

Internet servers and clients interconnect using Transmission Control Protocol/Internet Protocol (TCP/IP) ports, known as secure ports. Internet server machines run different services (for example HTTP or FTP) on different TCP/IP ports (see table below).

**Table 7-1 Commonly used TCP/IP ports and their services**

Port	Service
21	FTP
23	Telnet
79	Finger
80	HTTP
443	SSL

A port can only operate one service. For example, port 80 can only be used for HTTP traffic and port 443 can only be used for SSL traffic.

Firewalls provide secure networks by restricting traffic to secure TCP/IP ports, for example port 80 for HTTP.

A typical firewall blocks all communication to all ports except port 80 and port 443:

- Port 80 only allows HTTP traffic
- Port 443 only allows SSL traffic.

All other types of traffic on all other ports are blocked. Therefore, a person could not use the FTP on port 80 to access the Internet server and retrieve files because FTP communication is blocked on port 80.



## 7.5 About Visibroker Gatekeeper

This section explains what Visibroker Gatekeeper is and why it is used by Discoverer.

### 7.5.1 What is Visibroker Gatekeeper?

Visibroker Gatekeeper is a third-party software package developed by Inprise Corporation, which enables Visibroker CORBA clients and servers to communicate across networks while still conforming to security restrictions imposed by Internet browsers and firewalls.

### 7.5.2 Why does Discoverer Plus need Visibroker Gatekeeper?

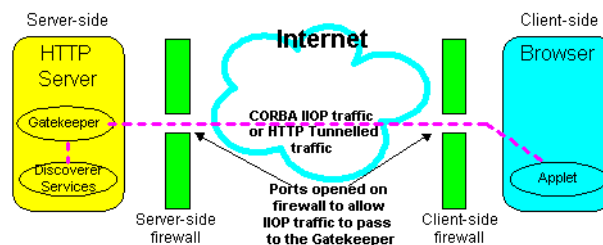
Discoverer Plus uses CORBA (Common Object Request Broker Architecture) to enable clients and servers to communicate. CORBA is an architecture that enables pieces of programs (called objects) to communicate with one another regardless of what programming language they were written in or what operating system they are running on.

CORBA is implemented over the internet using the IIOP (Internet Inter-ORB Protocol) protocol. The IIOP protocol enables the exchange of integers, arrays, and more complex objects between server and client machines. By contrast, the HTTP (HyperText Transfer Protocol) protocol that is the underlying protocol used by the internet only supports transmission of text.

For security reasons, some commercial firewalls do not allow IIOP traffic to pass directly through.

Visibroker Gatekeeper enables CORBA-based IIOP traffic to pass through firewalls.

**Figure 7-2** *Visibroker Gatekeeper being used to allow CORBA IIOP or HTTP Tunnelled traffic to pass through to a Server-side firewall*



Visibroker Gatekeeper typically runs behind the Server-side firewall on the HTTP Server machine. Visibroker Gatekeeper:

- 'listens' to (or monitors) all IIOP or HTTP requests from clients.
- forwards appropriate requests to Discoverer Services.
- listens for replies from the Discoverer Services and forwards them back to clients.

In this respect, Visibroker Gatekeeper works like a proxy server machine (a machine that acts as an intermediary between a workstation and the Internet).

### 7.5.3 Required Visibroker Gatekeeper Versions

Discoverer Plus uses Visibroker Gatekeeper version 3.4. Visibroker Gatekeeper is included as part of the Discoverer Plus installation.

## 7.6 Choosing a Gatekeeper configuration for Discoverer

This section describes typical firewall configurations and advises you on which Visibroker Gatekeeper configuration you need to deploy Discoverer Plus across public networks such as the Internet.

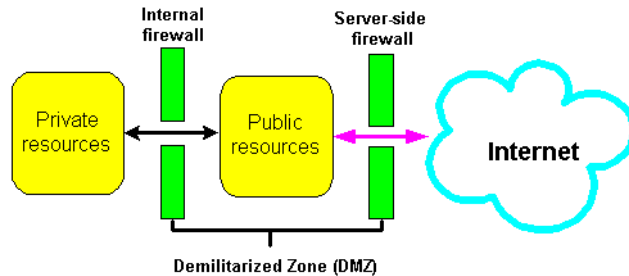
### 7.6.1 About firewall configurations

Firewall policies vary across organization, and there are a wide variety of bespoke and off-the-shelf firewall packages in use. This guide cannot cover every firewall scenario. Therefore, only the most common firewall configurations are covered.

A typical DMZ configuration comprises two firewalls, (see figure below):

- A server-side firewall between the Internet and your public resources.
- An internal firewall between your public resources and your private resources.

**Figure 7–3 A Demilitarized Zone (DMZ)**



A good firewall configuration assumes that resources in the DMZ will be breached, and should minimize damage to the internal network and any sensitive data residing on the network when this happens. This involves two steps:

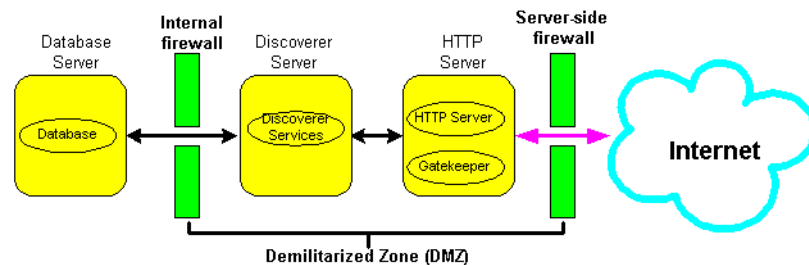
- Move sensitive private resources (at a minimum, databases and application logic) from the DMZ to the internal network behind the internal firewall.
- Restrict access to sensitive private resources from the DMZ itself, as well as from internal networks.

## 7.6.2 DMZ configuration examples

When deploying the Discoverer Server, two typical security configurations are:

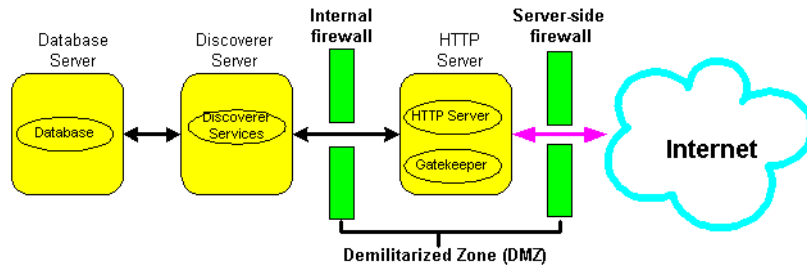
1. Deploy the Discoverer Server and HTTP Server in the DMZ. Deploy the database behind the Internal firewall, (see figure below). This provides a good level of security and is recommended for Discoverer Plus.

**Figure 7–4 Discoverer Server and HTTP deployed in a DMZ**



2. Deploy the HTTP Server in the DMZ. Deploy the Discoverer Server and the database behind the Internal firewall, (see figure below). This provides a higher level of security, but is more difficult to implement.

**Figure 7–5 Discoverer Server deployed behind an internal firewall**



### 7.6.3 Which Gatekeeper configuration do I choose?

Visibroker Gatekeeper typically handles IIOP or HTTP protocol traffic from clients and forwards them to Discoverer Services. This means that Visibroker Gatekeeper uses either an IIOP Proxying configuration or a HTTP Tunnelling configuration. Which configuration you choose depends on the restrictions imposed by your firewall policy.

#### 1. IIOP Proxying Configuration

- In this configuration, you open your Server-side firewall to allow TCP/IP traffic to pass through at a specific port. Client-side firewalls must also allow TCP/IP traffic to pass from the client to outside of the firewall.
- This configuration is recommended as it allows you to get the best performance from Discoverer Plus.
- However, because you often do not control your clients' firewalls, an IIOP configuration is more difficult to implement than a HTTP Tunnelling Configuration.

Refer to [Section 7.7, "Using Visibroker Gatekeeper with an IIOP Proxying Configuration"](#).

#### 2. HTTP Tunnelling Configuration

- In this configuration, you embed IIOProxy traffic inside HTTP packets to allow them to pass through your firewall.
- This method has the advantage that you do not need to open up the Server or Client-side firewalls for TCP/IP traffic.
- Therefore, the HTTP Tunnelling Configuration is more secure than the IIOProxying Configuration.
- This configuration has a significant performance overhead and is not as efficient as the IIOProxying Configuration.

Refer to [Section 7.8, "Using Visibroker with a HTTP Tunnelling Configuration"](#).

## 7.7 Using Visibroker Gatekeeper with an IIOProxying Configuration

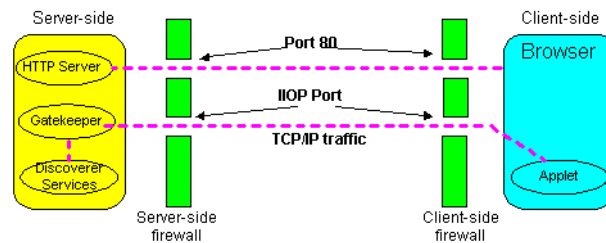
This section explains how to configure Discoverer Plus to use an IIOProxying configuration.

### 7.7.1 Overview

In an IIOProxying configuration, the Server-side and Client-side firewalls are opened to allow TCP/IP traffic.

When Discoverer Plus starts, the HTTP Server sends the Discoverer Applet to the Browser via Port 80. Once initialized, the Discoverer Applet communicates with the Gatekeeper via the specified IIOProxy port.

**Figure 7-6 Discoverer using an IIOProxying Configuration**



Visibroker Gatekeeper runs behind the Server-side firewall on the HTTP Server machine and helps Clients communicate with Discoverer Services. To do this, the Gatekeeper intercepts all requests from Clients and forwards them to Discoverer Services on behalf of the Clients. Replies from Discoverer Services to the Client are also passed through the Gatekeeper.

You can configure the Gatekeeper to use an IIOProxying configuration at two different security levels:

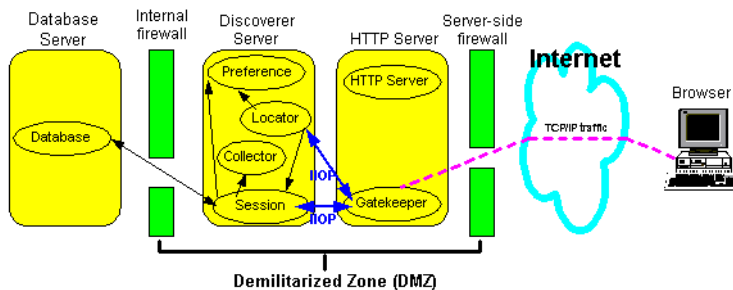
- [Configuring Gatekeeper with no firewall separating Discoverer Services](#) - Here, the Discoverer Server is deployed in the DMZ.
- [Configuring Gatekeeper with Discoverer Server components behind an internal firewall](#) - Here, the Discoverer Server is not deployed in the DMZ.

## 7.7.2 Configuring Gatekeeper with no firewall separating Discoverer Services

In this configuration, the Discoverer Server and the HTTP Server are deployed in the DMZ.

IIOProxy packets are not restricted to Discoverer Services - they can be sent to any location. The Discoverer Services and Locator are on the same sub-net (see figure below).

**Figure 7-7 IIOProxying with Discoverer deployed behind an internal firewall**



To configure Visibroker Gatekeeper:

- When you install Discoverer Plus, Visibroker Gatekeeper is installed on the HTTP Server at:
 

```
<ORACLE_806_HOME>/vbroker/bin/gatekeeper
```
- Navigate to the `<iSUITES_HOME>/Apache/Apache/htdocs/discwb4/applet` directory and type the following at the command prompt:
 

```
> <ORACLE_806_HOME>/vbroker/bin/gatekeeper/gatekeeper
```
- By default, Visibroker Gatekeeper listens to port 15000. To change the default port number, refer to [Section 7.10.5, "Changing the default Visibroker Gatekeeper port"](#).

- Once you have configured Visibroker Gatekeeper to run at a particular port, you must configure your Server firewall to allow TCP/IP traffic to go to Visibroker Gatekeeper's host and port.

If you do **not** control the Client-side firewall, there are three possible solutions for configuring the Server firewall:

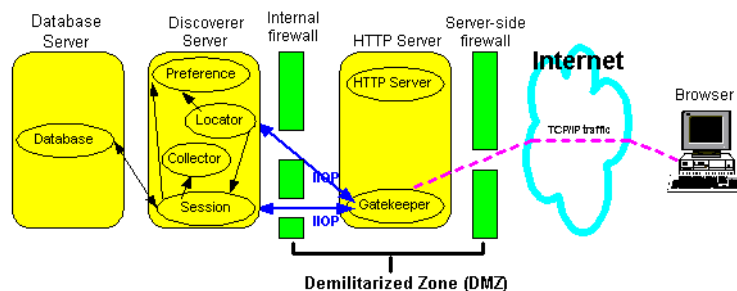
- If the Client's Firewall Administrator has configured the Client firewall to allow TCP/IP traffic to go out at all ports, the IIOProxying configuration will work correctly.
- If the Client's Firewall Administrator has configured some ports for TCP/IP (e.g. for Real Audio), you can use these opened ports to run the Gatekeeper.
- If there are clients that are not behind a client-side firewall, the IIOProxying configuration will work for them.

**Note:** Firewall configuration is specific to the firewall type used by your site. Contact your Firewall Administrator for details.

### 7.7.3 Configuring Gatekeeper with Discoverer Server components behind an internal firewall

In this configuration, the HTTP Server is deployed in the DMZ. The Discoverer Server and the database are deployed behind the Internal firewall (see figure below).

**Figure 7–8 IIOProxying with Discoverer deployed behind an internal firewall**



To configure Visibroker Gatekeeper:

- Follow the steps in [Section 7.7.2, "Configuring Gatekeeper with no firewall separating Discoverer Services"](#).

- Open the firewall between your HTTP Server and your Discoverer Servers to allow all TCP/IP traffic to go to Discoverer Server machines on all ports. Then selectively block ports to which you want to restrict access, (for example, the HTTP or telnet port).
- On your HTTP Server machine, prevent the Locator from running. To do this:
  - navigate to the <ORACLE\_806\_HOME>/discwb4/util directory
  - stop and restart Discoverer by typing the following at the command prompt:

```
> stopall.sh
> startosagent.sh
> startoad.sh
```
- On the Discoverer Server, run the Locator. To do this:
  - navigate to the <ORACLE\_806\_HOME>DISCWB4/util directory
  - stop and restart Discoverer by typing the following at the command prompt:

```
> stopall.sh
> startosagent.sh
> startoad.sh
> startlocator.sh
```
- Make sure that the Locator and Discoverer Session and Preferences are on the same subnet.
- Each time the Locator is started, the locator.ior file generated by the Locator must be copied to the <iSUITES\_HOME>/Apache/Apache/htdocs/discwb4/applet directory on the HTTP Server machine.

## 7.8 Using Visibroker with a HTTP Tunnelling Configuration

This sections explains how to configure Discoverer to use a HTTP Tunnelling configuration.

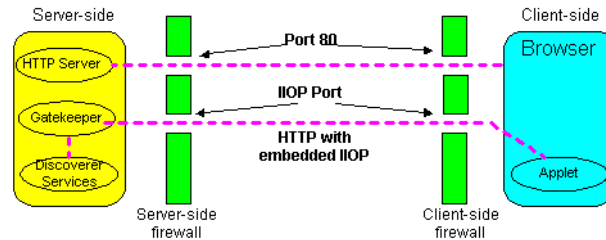
### 7.8.1 Overview

If the Client is running behind a firewall that only allows HTTP traffic, the client and server communicate by embedding IIOp data into HTTP packets, which allows them to pass through the firewall.



In an HTTP Tunnelling Configuration, Visibroker Gatekeeper extracts the IIOP requests from the HTTP packets and passes them on to the Discoverer Server components. Replies are sent back to the Client as HTTP response packets, which the Client extracts to get the IIOP packets.

**Figure 7–9 Discoverer using an HTTP Tunnelling Configuration**



The main advantage of using the HTTP Tunnelling configuration is that you do not have to open up your firewall for TCP/IP traffic. However, you must ensure that the firewall allows HTTP traffic to the host and port that your Gatekeeper is running on, see [Section 7.10.1, "Where to run Visibroker Gatekeeper"](#).

NOTE: There is an alternative method of running Discoverer through multiple firewalls without having to open up the firewalls. This involves running Visibroker Gatekeeper on port 443, (refer to [Section 6.8, "Configuring Discoverer Plus to use SSL"](#)).

Because the port 80 is the default port used by HTTP Servers, Visibroker Gatekeeper must use a different port number when running Visibroker Gatekeeper on the HTTP Server. The default port number that Visibroker Gatekeeper uses is port 15000. You can also run Visibroker Gatekeeper on a different machine from the HTTP Server, see [Section 7.10.3, "Running Visibroker Gatekeeper on a different Server"](#).

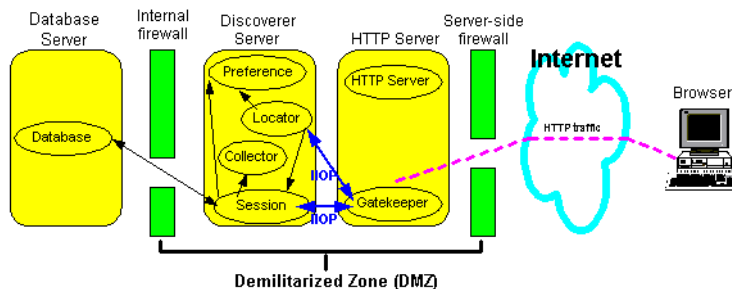
You can configure the Gatekeeper to use an HTTP Tunnelling configuration at two different security levels:

- [Configuring Gatekeeper with no firewall separating Discoverer Services](#) - Here, the Discoverer Server is placed in the DMZ
- [Configuring Gatekeeper with Discoverer server components behind an internal firewall](#) - Here, the Discoverer Server is not in the DMZ.

## 7.8.2 Configuring Gatekeeper with no firewall separating Discoverer Services

In this configuration, the Discoverer Server and the HTTP Server are deployed in the DMZ (see figure below).

**Figure 7–10 HTTP Tunnelling with Discoverer deployed behind an internal firewall**



To configure Visibroker Gatekeeper:

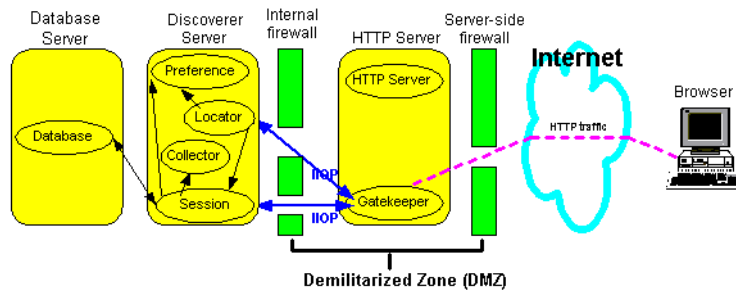
- When you install Discoverer Plus, Visibroker Gatekeeper is installed on the HTTP Server at:
 

```
<ORACLE_806_HOME>/vbroker/bin/gatekeeper
```
- Check that Gatekeeper is running. Gatekeeper is started by the Discoverer script that starts OAD, OSAgent, and Locator processes.
- By default, Visibroker Gatekeeper listens to port 15000. To change the default port number, refer to [Section 7.10.5, "Changing the default Visibroker Gatekeeper port"](#).
- Once you have configured Visibroker Gatekeeper to run at a particular port, you must ensure that the firewall allows HTTP traffic to the host and port that Visibroker Gatekeeper is running on.

**Note:** Firewall configuration is specific to the firewall type used by your site. Contact your Firewall Administrator for details.

## 7.8.3 Configuring Gatekeeper with Discoverer server components behind an internal firewall

In this configuration, the HTTP Server is deployed in the DMZ. The Discoverer Server and the database are deployed behind the Internal firewall (see figure below).

**Figure 7–11 HTTP Tunnelling with Discoverer deployed behind an internal firewall**

To configure Visibroker Gatekeeper:

- Follow the steps in [Section 7.8.2, "Configuring Gatekeeper with no firewall separating Discoverer Services"](#).
- Open the firewall between your HTTP Server and your Discoverer Servers to allow all TCP/IP traffic to go to Discoverer Server machines on all ports. Then selectively block ports to which you want to restrict access, (for example, the HTTP or Telnet port).
- On your HTTP Server machine, prevent the Locator from running. To do this:
  - navigate to the `<ORACLE_806_HOME>/discwb4/util` directory
  - stop and restart Discoverer by typing the following at the command prompt:
 

```
> stopall.sh
> startosagent.sh
> startoad.sh
```
- On the Discoverer Server, run the Locator. To do this:
  - navigate to the `<ORACLE_806_HOME>DISCW44/util` directory
  - stop and restart Discoverer by typing the following at the command prompt:
 

```
> stopall.sh
> startosagent.sh
> startoad.sh
> startlocator.sh
```
- Make sure that the Locator and Discoverer Session and Preferences are on the same subnet.

- Each time the Locator is started, the locator.ior file generated by the Locator must be copied to the <i>SUITES\_HOME>/Apache/Apache/htdocs/discwb4/applet directory on the HTTP Server machine.

## 7.8.4 Persistent HTTP Connections

Because HTTP is a connectionless protocol, a new HTTP connection must be established every time a Discoverer Client wants to communicate via the gatekeeper. This connection is made, a response is received, and the connection is closed.

The Discoverer Client and the gatekeeper can communicate via a connection that is established and used for all requests until the Discoverer Client exits. This is known as a Persistent Connection. When using HTTP Tunnelling, Persistent Connections are recommended for Discoverer Plus.

To use a Persistent Connection, the HTTP proxies running on the firewalls between the client and gatekeeper must support Persistent Connections.

**Note:** Firewall configuration is specific to the firewall type used by your site. Contact your Firewall Administrator for details.

## 7.9 Configuring the Discoverer client

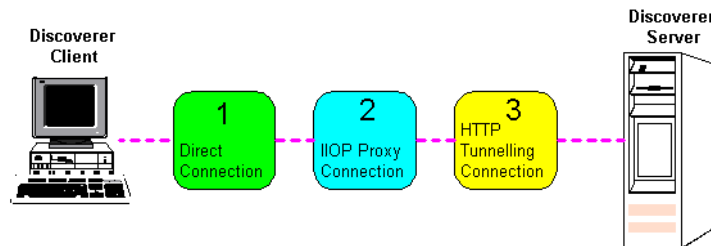
This section explains how to configure the Discoverer client to work with Visibroker Gatekeeper.

## 7.9.1 About the Discoverer client connection sequence

By default, a new Discoverer client session tries to connect to the Discoverer server in the following three-step connection sequence.

1. The Discoverer client tries to connect directly to the Locator (i.e. not using the Gatekeeper).
2. If the Discoverer client cannot make a direct connection, it tries to connect to the Discoverer server using IIOP Proxying.
3. If IIOP Proxying is not configured, the Discoverer client uses HTTP Tunnelling to communicate with the Discoverer server.

**Figure 7–12** *Default Discoverer client default three-step connection sequence*



To improve the performance of Discoverer, you can specify an alternative connection sequence (for more information refer to [Section 7.9.2, "How to use a specific connection method"](#)).

## 7.9.2 How to use a specific connection method

You can specify a connection method for a Discoverer client using one of the following URL parameters:

- `ORBalwaysProxy` - connect using IIOP Proxying (see table below).
- `ORBalwaysTunnel` - connect using HTTP Tunelling (see table below).

**Note:** Do not use both URL parameters in the same connect string.

**Table 7–2 Discoverer URL Parameters for specifying a connection method**

Argument and Values	Purpose	Example
ORBalwaysProxy=yes	Set this parameter to yes to bypass the direct connection and try to connect via IIOProxying. If this fails, the client tries to connect using HTTP Tunnelling.	http://server.com/discwb4/html/english/ms_ie/start_ie.htm?ORBalwaysProxy=yes
ORBalwaysTunnel=yes	set this parameter to yes to bypass both the direct connection and the IIOProxying and try to connect using HTTP Tunnelling	http://server.com/discwb4/html/english/ms_ie/start_ie.htm?ORBalwaysTunnel=yes

**Note:** Use these parameters to set up separate URLs for clients who you know will always be connecting from behind firewalls.

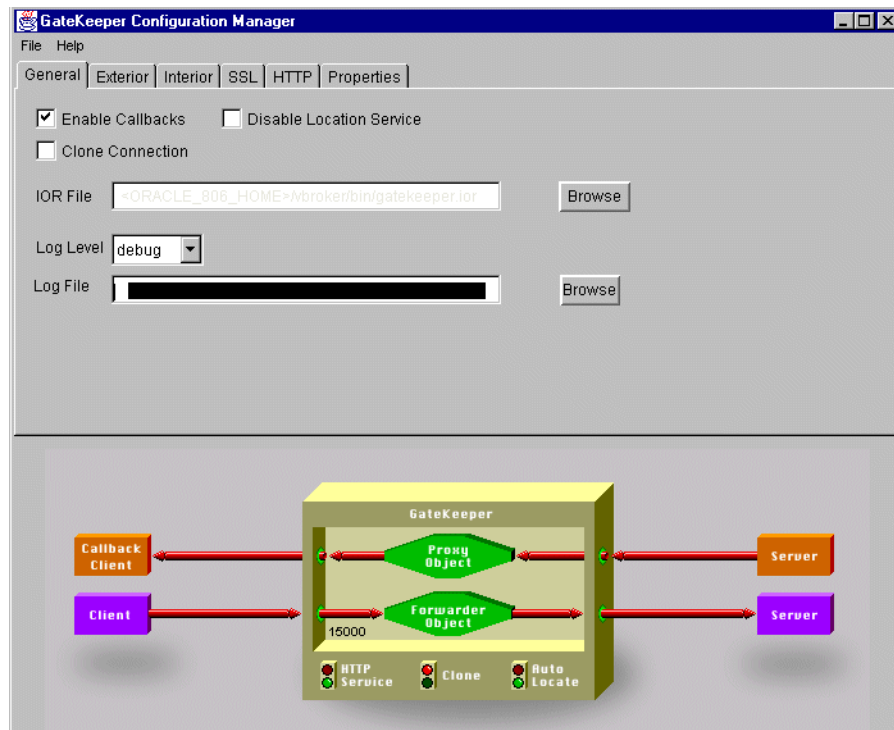
See also "[Enabling SSL in Discoverer Plus start pages](#)".

## 7.10 Configuring Visibroker Gatekeeper

This section explains how to configure Visibroker Gatekeeper to work with Discoverer Plus.

You configure Visibroker Gatekeeper using the Gatekeeper Configuration Manager (see figure below).

**Figure 7–13 GateKeeper Configuration Manager**



## 7.10.1 Where to run Visibroker Gatekeeper

When configuring Visibroker Gatekeeper, you have two options:

1. If your firewall policy allows the HTTP Server to receive HTTP traffic at a port other than port 80, install and run Visibroker Gatekeeper on the HTTP server machine.

**NOTE: This configuration is strongly recommended for Discoverer Plus.**

The default Visibroker Gatekeeper port is 15000, though this can be changed to any port number (except 80). See [Section 7.10.2, "Running Visibroker Gatekeeper on the HTTP Server"](#).

2. If your firewall policy only allows the HTTP Server to receive HTTP traffic at port 80, you must install Visibroker Gatekeeper on a different machine.

**NOTE: This configuration has a greater set up time and maintenance overhead and is not recommended for Discoverer Plus, unless option 1 (above) is not viable.**

The Visibroker Gatekeeper file `gatekeeper.ior` must be copied to the HTTP Server every time Visibroker Gatekeeper is re-started. See [Section 7.10.3, "Running Visibroker Gatekeeper on a different Server"](#).

**NOTE: If you wish to use Secure Sockets Layer (SSL), you must run Visibroker Gatekeeper on port 443, (the standard SSL port). Refer to ["Installing Visibroker Gatekeeper to work with SSL"](#).**

## 7.10.2 Running Visibroker Gatekeeper on the HTTP Server

This section explains how to run Visibroker Gatekeeper on the HTTP Server (for more information about choosing a configuration, see [Section 7.10.1, "Where to run Visibroker Gatekeeper"](#)).

- Visibroker Gatekeeper is installed on the HTTP server in the following location:  
`<ORACLE_806_HOME>/vbroker/bin/gatekeeper`
- Navigate to the `<iSUITES_HOME>/Apache/Apache/htdocs/discwb4/applet` directory and type the following at the command prompt:  

```
> <ORACLE_806_HOME>/vbroker/bin/gatekeeper/gatekeeper
```
- Visibroker Gatekeeper is started in the `<iSUITES_HOME>/Apache/Apache/htdocs/discwb4/applet` directory in the HTTP Server's documents directory. The Gatekeeper writes out a file `gatekeeper.ior` in



the directory in which it is started. The Discoverer client expects to find this file in the <iSUITES\_HOME>/Apache/Apache/htdocs/discwb4/applet directory.

- Make sure that the gatekeeper is started by checking the gatekeeper.ior file is present in the <iSUITES\_HOME>/Apache/Apache/htdocs/discwb4/applet directory in the HTTP Server's documents directory. You must also ensure that the gatekeeper.ior file has a time stamp corresponding to the time that the Discoverer 4i service was started. If not, users outside your firewall will not be able to connect to Discoverer.

### 7.10.3 Running Visibroker Gatekeeper on a different Server

This section explains how to run Visibroker Gatekeeper on a machine other than the HTTP Server machine (for more information about choosing a configuration, see [Section 7.10.1, "Where to run Visibroker Gatekeeper"](#)).

- To install Visibroker Gatekeeper on a different Server machine, you have two options:
  1. Install *iAS* on the separate Server machine, which installs Visibroker Gatekeeper.
  2. If you do not want to do a full *iAS* install on the separate Server machine just to install the Gatekeeper, we recommend that you consider purchasing a stand alone copy of Visibroker for Java 3.4 from Inprise Corp. For more information, visit the Visibroker Web site at:  
<http://www.inprise.com/visibroker/>.
- The **gatekeeper.ior** file generated by Visibroker Gatekeeper must be copied to the <iSUITES\_HOME>/Apache/Apache/htdocs/discwb4/applet directory in the HTTP Server's documents directory. The file must be copied every time you restart Visibroker Gatekeeper.
- Visibroker Gatekeeper must only run on the other machine and not on the HTTP Server machine.

### 7.10.4 Configuring the Visibroker Gatekeeper port

To work with Discoverer, Visibroker Gatekeeper is configured as follows:

- Visibroker Gatekeeper is installed on the HTTP Server, or a separate server.
- By default, Visibroker Gatekeeper monitors (or 'listens') to port 15000. To change the default port, see [Section 7.10.5, "Changing the default Visibroker Gatekeeper port"](#).

Note: If you are running Visibroker Gatekeeper on the HTTP Server, you cannot set the Visibroker Gatekeeper port to port 80. This is because port 80 is the default HTTP Server port on which most HTTP Servers install themselves.

### 7.10.5 Changing the default Visibroker Gatekeeper port

To change the default Visibroker Gatekeeper port, follow these steps:

1. Run `<ORACLE_806_HOME>/vbroker/bin/gkconfig` to start the Gatekeeper Configuration Tool.
2. Choose **File | Open** and open the Visibroker Gatekeeper configuration file **gatekeeper.properties** in the HTTP Server (or separate server) document root directory (e.g. `<ISUITES_HOME>/Apache/Apache/htdocs/discwb4/applet`).
3. Click the **Exterior** tab.

**Figure 7–14** *Configuring the default Visibroker Gatekeeper port*

General	Exterior	Interior	SSL	HTTP	Properties
Exterior Address					
Exterior Port					
Exterior Proxy Address					
Exterior Proxy Port					
Exterior Callback Port					
Exterior Callback Proxy Port					

4. Type a new port number in the **Exterior Port** field.
5. Choose **File | Save**.
6. Close the Visibroker Gatekeeper Configuration Tool.
7. Restart Discoverer. For restarting Discoverer refer to [Section 2.6, "Restarting the Discoverer system"](#).

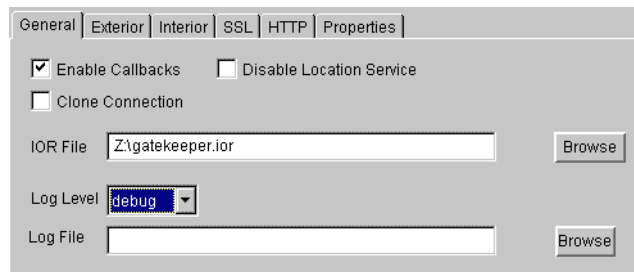
Visibroker Gatekeeper is now configured to use the new port number.

## 7.10.6 Getting Log Information from Visibroker Gatekeeper

To change the amount of information generated by the Visibroker Gatekeeper log, follow these steps:

1. Run `<ORACLE_806_HOME>/vbroker/bin/gkconfig` to start the Gatekeeper Configuration Tool.
2. Choose **File | Open** and open the Visibroker Gatekeeper configuration file **gatekeeper.properties** in the HTTP Server (or separate server) document root directory (e.g. `<i>SUITES_HOME</i>/Apache/Apache/htdocs/discwb4/applet`).
3. Click the **General** tab.

**Figure 7–15** *Configuring your Visibroker Gatekeeper debug level*



4. Select a log level from the options:
  - **Quiet** - do not print any messages.
  - **Warning** - print messages when an error occurs.
  - **Normal** - print error messages and information messages.
  - **Debug** - print all messages (the default value).
5. Click **File | Save**.
6. Close the Visibroker Gatekeeper Configuration Tool.
7. Restart Discoverer. For restarting Discoverer, refer to [Section 2.6, "Restarting the Discoverer system"](#).

Visibroker Gatekeeper is now configured to use the log level selected.

## 7.10.7 Configuring NAT devices to work in front of the Gatekeeper

To configure the Gatekeeper to work with your Network Address Translation (NAT) device *in front of* the Gatekeeper, follow these steps:

1. Run `<ORACLE_806_HOME>/vbroker/bin/gkconfig` to start the Gatekeeper Configuration Tool.
2. Choose **File | Open** and open the Visibroker Gatekeeper configuration file **gatekeeper.properties** in the HTTP Server (or separate server) document root directory (e.g. `<ISUITES_HOME>/Apache/Apache/htdocs/discwb4/applet`).
3. Click the **Exterior** tab.

**Figure 7–16** Configuring a NAT device to work in front of the Gatekeeper

The screenshot shows the Gatekeeper Configuration Tool interface with the 'Exterior' tab selected. The configuration fields are as follows:

Field	Value
Exterior Address	
Exterior Port	15000
Exterior Proxy Address	105.23.45.6
Exterior Proxy Port	235
Exterior Callback Port	
Exterior Callback Proxy Port	

4. In the **Exterior Proxy Address** field, enter the IP address that your NAT device translates your Gatekeeper's IP address to.

For example, if your Gatekeeper's IP Address is 101.20.34.6 and your NAT device translates that to 105.23.45.6, enter 105.23.45.6 in this field.

5. In the **Exterior Proxy Port** field, enter the port that your NAT device translates your Gatekeeper's port to.

For example, if your Gatekeeper's default port is 15000 and the NAT device translates that to 235, enter 235 in this field.

6. Click **File | Save**.
7. Close the Visibroker Gatekeeper Configuration Tool.

8. Restart Discoverer. For restarting Discoverer refer to [Section 2.6, "Restarting the Discoverer system"](#).

Visibroker Gatekeeper is now configured to use your NAT device in front of the Gatekeeper.

NOTE: Discoverer does not support the use of NAT devices behind the Gatekeeper.



---

---

# Oracle9iAS Discoverer and Oracle Applications

Oracle9iAS Discoverer provides additional capabilities for working seamlessly with Oracle Applications.

## A.1 Specifying display of the Applications User Checkbox in the Discoverer Viewer Connect Dialog

You can set up the Discoverer Viewer Connect dialog to enable an Oracle Applications user to log on using an Applications 'Responsibility' by displaying the Applications User checkbox and/or by specifying that Discoverer runs against Oracle Applications by default.

To set up the Connect dialog to display (or hide) the Applications User checkbox, access the configuration file located in: <iSUITES\_HOME>apache/apache/htdocs/disco4iv/html/**viewer\_config.xml** and set the "display\_login\_method\_control" option to "true" or "false" as required.

To specify whether users run Discoverer Viewer by default against an Oracle Application (if the checkbox is displayed, it will be checked or unchecked as appropriate), change the following line in the disco4iv.xml file:

### **disco4iv.xml - Before change**

```
<application xml_version="viewer41"/>
```

### **disco4iv.xml - After change**

```
<application xml_version="viewer41" apps_logon="yourdefault"/>
```

where "yourdefault" is either "true" or "false".

For more information about editing the disco4iv.xml file, refer to Section 3.2.6, “Editing the disco4iv.xml file (Discoverer Viewer only)”.

## A.2 Specifying display of the Applications User Checkbox in the Discoverer Plus Connect Dialog

You can set up the Discoverer Plus Connect dialog to enable an Oracle Applications user to log on using an Oracle Applications ‘Responsibility’ by displaying the Applications User checkbox and/or by specifying that Discoverer runs against Oracle Applications by default.

To set up the Connect dialog to display (or hide) the Applications User checkbox:

1. Change the value of the following variables in the pref.txt file:

Variable	Settings
ShowUserTypeChoice	Set to: <ul style="list-style-type: none"> <li>■ 0 to hide checkbox</li> <li>■ 1 to show checkbox</li> </ul>
DefaultUserTypeIsApps	Set to: <ul style="list-style-type: none"> <li>■ 0 to specify that users do not run Discoverer Plus by default against an Oracle Application (if the checkbox is displayed, it is unchecked)</li> <li>■ 1 to specify that users run Discoverer Plus by default against an Oracle Application (if the checkbox is displayed, it is checked)</li> </ul>

For more information about editing the pref.txt file, refer to Section 2.4, “Editing the End User Preferences”.

2. Run script: apply\_preferences.sh available under <ORACLE\_806\_HOME>/discwb4/util directory.
3. Stop and restart the OracleDiscoverer4i service. For restarting the Discoverer system, refer to Section 2.6, “Restarting the Discoverer system”.



## A.3 Settings in pref.txt for Oracle Applications Users

There are four settings in the pref.txt file specifically for Oracle Applications end users.

- AppsGWYUID
- AppsFNDNAM
- ShowUserTypeChoice
- DefaultUserTypeIsApps

For more information:

- about the settings in pref.txt, refer to Appendix C.1, “Preference settings in pref.txt”
- about editing the pref.txt file, refer to Section 2.4, “Editing the End User Preferences”



---



---

## Discoverer Viewer Configuration Files

### B.1 Configuration settings in viewer\_config.xml

Area	Setting	Description
configuration	base_dir="."	specifies the location of directories for images and help if different from the default or if the xsl_path attribute in disco4iv.xml has been set to use a protocol other than http: (e.g. if xsl_path=file:/// <path>)
	image_dir="images"	specifies the path to the directory containing the Viewer images (either an absolute path or a path relative to the directory specified by the base_dir attribute of the configuration element)
appearance	option id="display_login_method_control" state="false"	whether to display the "Applications User" checkbox
	option id="display_show_xml_link" state="true"	whether to display the "Show XML" link at the bottom of every page
	option id="display_language_control" state="true"	whether to display the Language dropdown list on the Connect dialog

Area	Setting	Description
	option id="use_javascript_enhanced_ui" state="true"	whether to use the JavaScript enhanced user interface
presentation_templates	look name="Corporate Look"	the name of the presentation template
	component name="xxx" state="true   false"	specifies whether a particular UI component is displayed or hidden for the current presentation template (refer to the comments in viewer_config.xml for more details about component names)
nls	resource id="xxx" href="yyy" help="zzz"	specifies the message file yyy and help file zzz to use for a particular language xxx.
help	base_dir="help"	specifies the path to the directory containing the help directories (either an absolute path or a path relative to the directory specified by the base_dir attribute of the configuration element)
	page name="reconnect" file="connect.htm"	filename for the "reconnect" online help file
	page name="roles" file="connect.htm"	filename for the "roles" online help file
	page name="workbooks" file="list_of_workbooks.htm"	filename for the "workbooks" online help file
	page name="worksheets" file="list_of_worksheets.htm"	filename for the "worksheets" online help file
	page name="run_query" file="run_query.htm"	filename for the "start query" online help file
	page name="running_query" file="query_progress.htm"	filename for the "running_query" online help file
	page name="data" file="worksheet_display.htm"	filename for the "data" online help file
	page name="parameters" file="edit_parameters.htm"	filename for the "parameters" online help file

Area	Setting	Description
	page name="options" file="options.htm"	filename for the "options" online help file
	page name="edit_layout" file="edit_layout.htm"	filename for the "edit_layout" online help file
	page name="presentation" file="presentation_options.htm"	filename for the "presentation" online help file
	page name="export" file="export.htm"	filename for the "export"

## B.2 Configuration settings in disco4iv.xml

Setting	Description
apps_logon	<p>Specifies whether users run Discoverer Viewer by default against an Oracle Application:</p> <ul style="list-style-type: none"> <li>set to true if users run Discoverer Viewer by default against an Oracle Application</li> <li>set to false if users do not run Discoverer Viewer by default against an Oracle Application</li> </ul> <p>For more information refer to Appendix A.1, "Specifying display of the Applications User Checkbox in the Discoverer Viewer Connect Dialog".</p>
query_progress_delay	Specifies the delay in seconds before a query progress page is displayed (or redisplayed). You might use this argument to avoid writing code to handle query progress pages when using the Viewer Servlet to return data to an application you have written.
redirect_delay	Specifies the delay in seconds before a browser is redirected to the original URL. Use this argument to prevent the browser timing out during export operations. Note that this setting only applies when a valid value has also been specified for the fm argument.
show_graphs	<p>Specifies whether to display charts that have been saved with workbooks:</p> <ul style="list-style-type: none"> <li>set to true to display charts (default)</li> <li>set to false to prevent the display of charts (improves performance where charts are not required)</li> </ul>

<b>Setting</b>	<b>Description</b>
xml_version	Specifies the version of the Discoverer Viewer DTD that XML documents are to conform to: <ul style="list-style-type: none"><li>▪ set to viewer33 to produce XML documents that conform to the Discoverer 3i Viewer DTD</li><li>▪ set to viewer41 (the default) to produce XML documents that conform to the Oracle9iAS Discoverer Viewer DTD</li></ul>
xsl_path	Specifies the path to the disco4iv.xsl file.
xsl_cache	Specifies whether the servlet caches the xsl or not.

---



---

# Oracle9iAS Discoverer Preference Settings

## C.1 Preference settings in pref.txt

For more information about changing settings in the pref.txt file, refer to Section 2.4, “Editing the End User Preferences”.

Affected Feature	Preference Item Name	Description	Default/Values
Locator	MachineIPs	IP addresses or machine names of all Discoverer Server Application server. <b>This is the only required item in the pref.txt file.</b>	
Administrator	ShowUserTypeChoice	Activates an end-user checkbox that allows users to specify whether they want to run Discoverer Plus against an Oracle Application. Not used by Discoverer Viewer (specified in viewer_config.xml instead).	0 (0=hide checkbox, 1=show checkbox)
	DefaultUserTypeIsApps	Specifies that users run Discoverer Plus by default against an Oracle Application. Not used by Discoverer Viewer (specified in disco4iv.xml instead).	0 (0=No, 1=Yes)
Session Manager	Timeout	Amount of time after which Discoverer disconnects an idle session from the database. Minimum time is 180 seconds.	1800 (seconds)
	RowsPerHTML	Number of rows to display per HTML page.	25 (any whole number)

Affected Feature	Preference Item Name	Description	Default/Values
Application	CacheFlushPercentage	Percentage of cache flushed if the cache is full.	25
	MaxVirtualDiskMem	Maximum amount of disk memory allowed for the data cache.	1024000000
	MaxVirtualHeapMem	Maximum amount of heap memory allowed for the data cache.	1024000000
	QueryBehavior	Action to take after opening a workbook.	0 (0 = Run Query Automatically, 1 = Don't Run Query, 2 = Ask for Confirmation)
	ShowDialogBitmaps	End users see the bitmap graphics on Discoverer Plus dialogs.	1 (0 = off, 1 = on)
	DataFormat	Applies HTML formatting to data cells in worksheets.	"<fontFormat fontName="Dialog" pitch="10" bold="false" italic="false" underline="false" strikethrough="false" foreground="0,0,0" background="255, 255, 255" valign="top"></ fontFormat">
	HeadingFormat	Applies HTML formatting to heading cells in worksheets.	"<fontFormat fontName="Dialog" pitch="10" bold="false" italic="false" underline="false" strikethrough="false" foreground="0,0,0" background="204, 204, 204" valign="top"></ fontFormat">



Affected Feature	Preference Item Name	Description	Default/Values
	TotalsFormat	Applies HTML formatting to cells that contain totals in worksheets.	<fontFormat fontName="Dialog" pitch="10" bold="false" italic="false" underline="false" strikethrough="false" foreground="0,0,0" background="255, 255, 255" valign="left" valign="top"></ fontFormat>
	NullValue	Specifies how null values are displayed in worksheets, for example, "NULL", "N/A", "0".	"NULL"
	ExcelVersion	Specifies the version of Excel to export to.  Different versions of Excel impose different limits on the number of rows per sheet: <ul style="list-style-type: none"> <li>■ the Excel95 limit is 16384 rows per sheet</li> <li>■ the Excel97 limit is 65536 rows per sheet</li> </ul> If the limit is exceeded, Discoverer places the exported rows on a new sheet.	"Excel95" ( "Excel97")
Database	DisableFanTrapDetection	Disables detection for fan trap in user queries.	0 (0 = on, 1 = off)
	DisableMultiJoinDetection	Disables multiple join detection.	1 (0 = on, 1 = off)
	DisableAutoQuery	If set to 0, queries do not automatically run when a worksheet opens.	0 (0 = on, 1 = off)
	DisableAutoOuterJoinsOnFilters	Turn off Automatic Outer Joins on filters.	0 (0 = on, 1 = off)
	ItemClassDelay	Fetch list of values for this amount of time before timing out.	15 (seconds)

Affected Feature	Preference Item Name	Description	Default/Values
	PredictionThresholdSeconds	Warn user if the predicted query time will exceed the number of seconds.	60 (seconds)
	PredictionThresholdSeconds Enabled	Query prediction threshold disabled (0) or enabled (1).	1
	QueryTimeLimit	Limit on query time in seconds.	1800 (seconds)
	QueryTimeLimitEnabled	Query time limit disabled (0) or enabled (1).	1
	RowFetchLimit	The maximum number of rows fetched.	10000 (rows)
	RowFetchLimitEnabled	RowFetchLimit parameter is disabled (0) or enabled (1).	1
	RowsPerFetch	The number of rows to fetch at once.	100 (rows)
	SummaryThreshold	Use summary table only if it is no older than this number of days.	30 (days) 0 do not use summary at all
	SummaryThresholdEnabled	If enabled (1), will use the value specified by SummaryThreshold. Otherwise, Discoverer will always use summary tables if they are available.	1
	MaterializedViewRedirection Behavior	8.1.6 Materialized View Redirection.	0 (0=always when available, 1=always when summary data not stale, 2=never)
	DefaultEUL	Sets the EUL you want all users to connect to. Each user can override this default from the Discoverer Plus Options dialog.	Any EUL you have created.
Oracle Applications Mode	AppsGWYUID	Specifies the public username and password for the AOL security DLL.	"APPLSYSPUB/PUB"
	AppsFNDNAM	Specifies the schema in which certain vital Oracle Applications data is stored.	"APPS"

Affected Feature	Preference Item Name	Description	Default/Values
Internal settings	BusinessAreaFastFetchLevel	Controls the amount of pre-caching that occurs for components and items whenever a Business Area is fetched.	1
	ObjectsAlwaysAccessible	Verify that Business Area objects and items exist in the database.	0 (0=verify; 1=don't verify and assume that the objects and items exist)
	SummaryObjectsUseCached Accessibility	Controls whether to access the summary-derived objects in the cache.	0
	AvoidServerWildcardBug	Set to 1 if running against RDBMS 7.3.2 with NLS_LANG set to Japanese_Japan.JA16SJIS to avoid certain list box appearing empty.	0
	RdbFastSQLOff		0
Query Prediction and Performance	QPPEnable	Uses query prediction/performance (QPP) if set to 1.	1 (0 = false, 1 = true)
	QPPCreateNewStats	Records new statistics if set to 1.	1 (0 = false, 1 = true)
	QPPLoadStatsByObjectUseKey	Records statistics for same objects first if set to 1.	1 (0 = false, 1 = true)
	QPPUseCpuTime	Query prediction uses CPU time within its algorithm.	1 (0 = false, 1 = true)
	QPPAvgCostTimeAlgorithm		2
	QPPMaxObjectUseKey	Affects the amount of statistics to cache in memory for query prediction.	30
	QPPCBOEnforced	Will use cost-based optimizer if set to 1. Will use default optimizer if set to 0.	1
	QPPObtainCostMethod		1
	QPPMinCost	Only records or uses statistics with a cost greater than this value.	0
	QPPMaxStats	Only loads this many previous statistics.	500

Affected Feature	Preference Item Name	Description	Default/Values
	QPPMinActCpuTime	Only records or uses statistics with a CPU time greater than this value.	0
	QPPMinActElapsedTime	Only records or uses statistics with an actual elapsed time greater than this value.	0
	QPPMinEstElapsedTime	Only records or uses statistics with an estimated elapsed time greater than this value.	0
	UseOptimizerHints	Will add optimized hints to SQL if set 1.	0
	QuerySQLFastFetchLevel		1
	SQLTrace	False (off).	0
Crosstab Layout	Title	Displays titles that were created in Discoverer 3.1 worksheets.	1 (0 = no, 1 = yes)
	Cell XGridline	Show horizontal gridlines.	0 (0 = no, 1 = yes)
	Cell YGridline	Show vertical gridlines.	0 (0 = no, 1 = yes)
	Axis Style	Crosstab axis position.	2 (1 = inline, 2 = outline)
Table Layout	Title	Displays titles that were created in Discoverer 3.1 worksheets.	1 (0 = no, 1 = yes)
	Cell XGridline	Show horizontal gridlines.	0 (0 = no, 1 = yes)
	Cell YGridline	Show vertical gridlines.	0 (0 = no, 1 = yes)
	Row Headings	Display row numbers on table worksheets.	0 (0 = no, 1 = yes)

---

---

# Index

## A

---

Additional Discoverer Server machine, 3-2, 3-4

## C

---

certificates and SSL, 6-3

configuration files for Discoverer Viewer, B-1

CORBA (Common Object Request Broker Architecture), 7-5

customizing Discoverer Viewer, 5-1

examples, 5-6, 5-9

## D

---

demilitarized zone (DMZ), 7-3

configuration examples, 7-7

disco4ivviewer.properties file, 6-16

disco4iv.xml file, 6-16, B-3

about, 3-13

editing, 3-13

Discoverer

and Oracle Applications, A-1

and Oracle9i Application Server, 1-2

architecture, 1-2

installing on multiple machines, 3-1, 3-2

process, 1-10

restarting, 2-13

running, 2-1

URL arguments, 4-1

Discoverer Plus

architecture, 1-3

firewall support, 7-1

process, 1-10

running, 2-2

URL arguments, 4-3

Discoverer Services components

Discoverer Java applet and HTML pages, 1-6

Locator component, 1-7

Preferences component, 1-8

registering, 2-8

Session component, 1-7

unregistering, 2-8

which machines to install on, 1-8

Discoverer Viewer

architecture, 1-4

configuration files, B-1

customizing, 5-1

enabling SSL, 6-15

installing SSL certificates, 6-15

JavaScript enhanced user interface, 2-16

running, 2-6

XML/XSL, 5-2

distributed installation, 3-1

DMZ, see demilitarized zone, 7-3

## E

---

End User Preferences, 2-10, 3-6

about, 2-10

pref.txt, C-1

setting, 2-11

## F

---

firewalls, 7-1

about, 7-3

Client-side, 7-3

- configuring Visibroker Gatekeeper, 7-19
- demilitarized zone, 7-3
- deploying the Discoverer Server, 7-7
- different configurations, 7-6
- DMZ, 7-3
- FTP, 7-3
- HTTP Tunnelling configuration, 7-8, 7-12
- IIOProxying configuration, 7-8, 7-9
  - policies, 7-6
  - port 443, 7-4
  - port 80, 7-4
  - secure ports, 7-4
  - Server-side, 7-3
  - SMTP, 7-3
  - SSL, 6-14
  - Visibroker Gatekeeper, 7-5

## G

---

Gatekeeper, see Visibroker Gatekeeper, 7-5

## H

---

- HTML generation by Discoverer Viewer, 5-2
- HTTP proxies, 7-16
- HTTP Tunnelling Gatekeeper configuration, 7-8, 7-12
- httpd.conf file, 6-15
- HTTPS and Discoverer Viewer, 6-15

## I

---

- iAS, see Oracle9i Application Server, 1-2
- IIOProxying (Internet Inter-ORB Protocol) protocol, 7-5
- IIOProxying Gatekeeper configuration, 7-8, 7-9
- installation of Discoverer
  - as part of Oracle9i Application Server, 1-2, 3-1
  - on multiple machines, 3-1, 3-2

## J

---

JavaScript enhanced user interface for Discoverer Viewer, 2-16

## L

---

- libvbj30ssl.so file, 6-12
- load balancing and Discoverer, 1-12, 3-7
  - examples, 3-8
- Locator component of Discoverer Services, 1-7
- logging on automatically using Discoverer Viewer, 5-22

## M

---

Master Discoverer Server machine, 3-2, 3-4

## N

---

- NAT devices
  - configuring to work in front of the Gatekeeper, 7-24
- Network Address Translation devices, see NAT devices, 7-24

## O

---

- OAD (Object Activation Daemon), 2-8
- Oracle Applications
  - and Oracle9iAS Discoverer, A-1
  - displaying the Applications User checkbox, A-1, A-2
  - settings in pref.txt, A-3
- Oracle9i Application Server
  - and Oracle9iAS Discoverer, 1-2
- Oracle9iAS Discoverer, see Discoverer, 1-1
- ORB - SSL enabled, 6-12
- ORBBalwaysProxy, 6-14, 7-17
- ORBBalwaysTunnel, 7-17
- ORBBenableSSL, 6-14

## P

---

- persistent HTTP connection, 7-16
- Plus, see Discoverer Plus, 1-1
- ports, 7-4
  - port 443, 7-4
  - port 80, 7-4
  - using, 7-4
- Preferences component of Discoverer Services, 1-8

- pref.txt file
  - about, 3-6
  - editing, 3-9
  - settings contained in, C-1
- private key encryption and SSL, 6-3
- public key encryption and SSL, 6-3

## R

---

- registering Discoverer Services components, 2-8
  - from the command line, 2-9

## S

---

- scalability and Discoverer, 1-12
- secure ports, 7-4
- security
  - firewalls, 7-1
  - SSL, 6-1
- Session component of Discoverer Services, 1-7
- shutting down
  - entire Discoverer system, 2-12
- SSL
  - about, 6-2
  - Certificate Authority, 6-3
  - certificates, 6-3
  - Discoverer Plus, 6-5
  - Discoverer Viewer, 6-15
  - enabling in Discoverer Plus Start pages, 6-13
  - enabling in Discoverer Viewer, 6-17
  - installing required files, 6-12
  - obtaining SSL certificates, 6-6
  - private keys, 6-3
  - public keys, 6-3
  - setting up secure links, 6-15
  - using inside firewalls, 6-14
  - X.509, 6-3
- starting Discoverer from a URL, 4-1

## T

---

- TCP/IP ports, 7-4
- tiered architecture of Discoverer
  - Client tier (Plus and Viewer), 1-5
  - Database tier (Plus and Viewer), 1-8

- Discoverer Services tier (Plus and Viewer), 1-6
- HTTP Server tier (Viewer only), 1-5
- tnsnames.ora file
  - about, 3-11
  - editing, 3-11
- troubleshooting, 2-17

## U

---

- unregistering Discoverer Services components, 2-8
  - from the command line, 2-9
- URL parameters - using multiple, 6-13
- URLs and Discoverer, 4-1
  - format of URL for Discoverer, 4-1
  - URL arguments for Discoverer Plus, 4-3
  - URL arguments for Discoverer Viewer, 4-6

## V

---

- vbj30ssl.dll file, 6-12
- vbrnissl.dll file, 6-12
- Video Stores Demo example, 5-20
- video.xml file, 5-22
- Viewer, see Discoverer Viewer, 1-1
- viewer\_config.xml file, 6-17, B-1
- Visibroker Gatekeeper
  - about, 7-5
  - and SSL, 6-5
  - configuring, 7-19
  - configuring the Discoverer client, 7-16
  - Gatekeeper Configuration Manager, 7-19
  - log information, 7-23
  - port, 7-21, 7-22
  - URL for more information about, 7-2
  - versions, 7-6
  - which machine to run on, 7-20

## X

---

- X.509 certificates, 6-3
- x509cert.jar, 6-12
- XSL
  - stylesheet, 5-1
  - tools, 5-1
  - using to produce HTML, 5-2

