Oracle®Discoverer™ 3*i* Viewer

Oracle Discoverer 3*i* Viewer Configuration Guide for Oracle Internet Application Server

For Unix

September 2000

Part Number: A86662-01

Oracle® Discoverer™ 3*i Viewer*, Oracle Discoverer 3*i* Viewer Configuration Guide for Oracle Internet Application Server, Unix

Part Number: A86662-01

Primary Author: Anjana Suparna Sriram

Contributing Authors: Suresh Lakshman Mali, Kayalvizhi Prageeth, Nick Fry

Contents

2 Running, Maintaining and Supporting Discoverer 3i Viewer

3 Accessing Oracle Discoverer 3i Viewer From Other Web Applications

4 Customizing Oracle Discoverer 3i Viewer

Send Us Your Comments

Oracle Discoverer 3*i Viewer Configuration Guide for Oracle Internet Application Server* for Unix

Part Number: A86662-01

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

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

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

- Email: discdoc@us.oracle.com
- FAX - +44 (0) 117 946 2646 Attn: Discoverer 3*i* Viewer Documentation
- By Post:
  Oracle Corporation
  Oracle Discoverer 3*i* Viewer Documentation
  CastleGate
  Tower Hill
  Bristol, BS2 0JA
  UK

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

____

____

____

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

Preface

Welcome to Oracle Discoverer 3*i* Viewer!

This guide explains how to configure and customize Oracle Discoverer 3*i* Viewer after it has been installed as part of Oracle Internet Application Server.

This guide does not explain how to install Oracle Discoverer 3*i* Viewer. For information about installation, refer to the *Oracle Internet Application Server Installation Guide*.

To use Oracle Discoverer 3*i* Viewer and this guide effectively, you should be familiar with HTTP servers and database concepts.

For the latest information, please read the Oracle Discoverer 3*i* Viewer release notes in conjunction with this guide.

Intended Audience

This guide is intended for the person who is maintaining Oracle Discoverer 3*i* Viewer. Most likely this will be a server administrator.

Conventions

EUL, the End User Layer™, refers to the metadata interface between the database and Discoverer.

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.
boldface text	Boldface type in text indicates a command name, menu name, file name, keyboard keys, or other choice or selection.
italic text	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.e.*, choose the menu, then the command under that menu.
`Courier text`	Text in this format indicates a command line to be typed.

1

Overview of Oracle Discoverer 3*i* Viewer

1.1 Introduction

Oracle Discoverer 3*i* Viewer is a business intelligence tool for viewing workbooks created by users of the client/server or web releases of Discoverer Plus (previously known as Discoverer User Edition). Using Oracle Discoverer's easy to use interface via a web browser, users can access and analyze database data.

Oracle Discoverer 3*i* Viewer is straightforward to set up and maintain. Oracle Discoverer 3*i* Viewer can be used to integrate database output into your web site and portal and is easily customized to fit in with your web site look and feel, or to build custom Discoverer applications for the web. Oracle Discoverer 3*i* Viewer is also optimized for performance and designed to minimize network traffic.

This chapter provides an overview of the Oracle Discoverer 3*i* Viewer architecture, describes the components and illustrates how Oracle Discoverer 3*i* Viewer works to give data access to users.

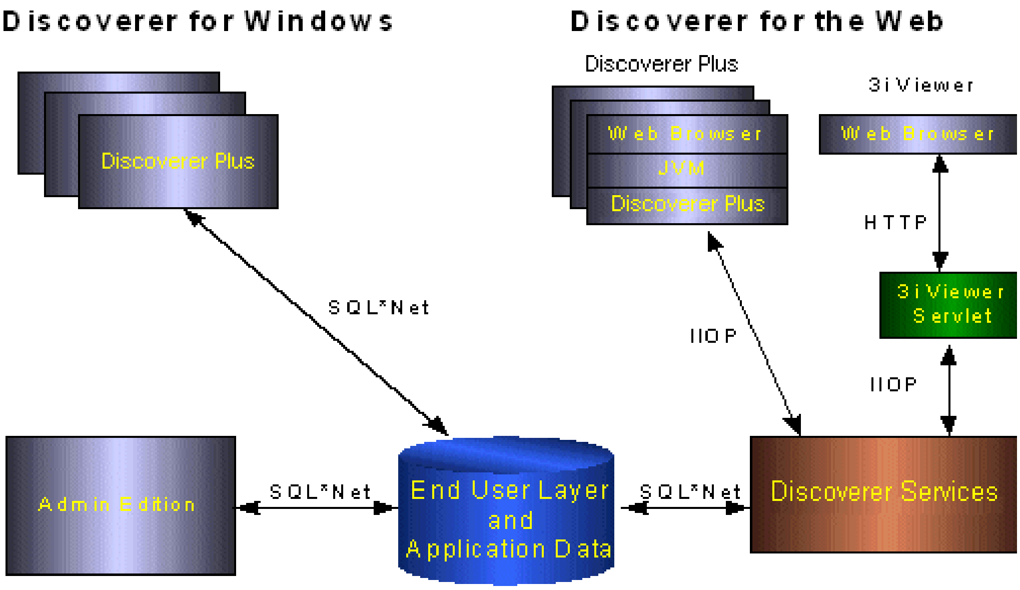
1.2 Oracle Discoverer 3*i* Viewer and Oracle Internet Application Server

Oracle Discoverer 3*i* Viewer is installed as part of Oracle Internet Application Server (iAS). The default installation is a single machine installation (i.e. all the Discoverer components are installed on a single machine).

1.3 Oracle Discoverer *3i* Viewer in context

Oracle Discoverer 3*i* Viewer can be used to view workbooks created by Discoverer Plus on client/server or the web. The deployment of Oracle Discoverer 3*i* Viewer is best understood by looking at its context (see Figure 1–1).

Figure 1–1 Oracle Discoverer 3i Viewer components in context



Oracle Discoverer *3i* Viewer is deployed using an Internet computing architecture that sits on top of the existing Discoverer Services.

Oracle Discoverer Services works with the Oracle iAS HTTP listener to provide a robust, scalable deployment platform for the Oracle Discoverer 3*i* Viewer.

1.4 The n- tiered architecture of Oracle Discoverer 3*i* Viewer

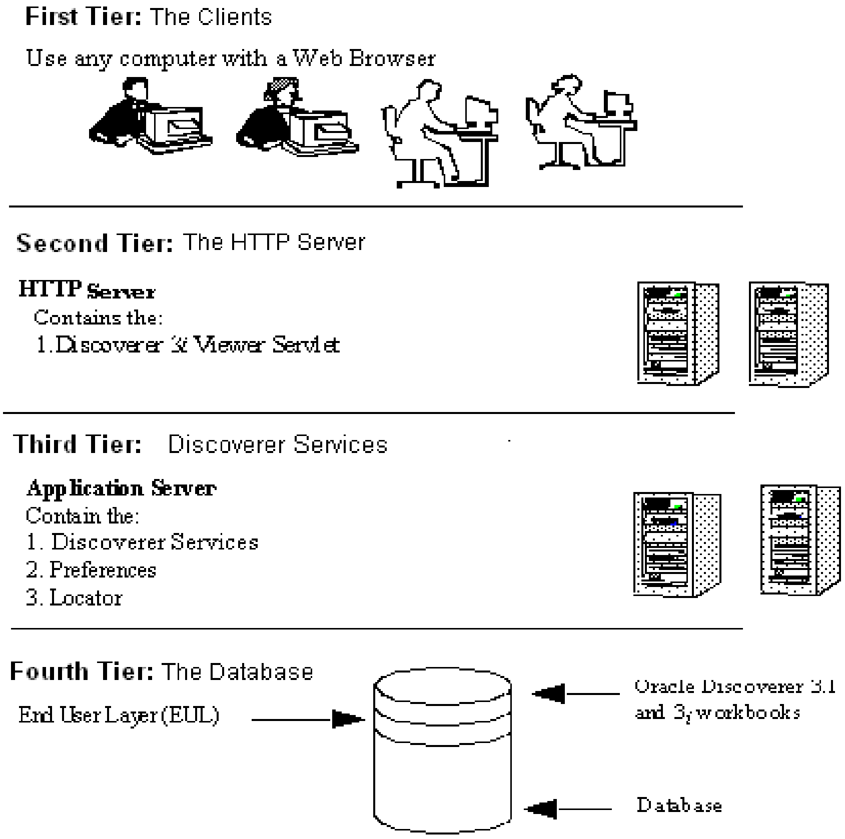
The n-tiers of Discoverer are as follows:

- Client
- HTTP Server
- Discoverer Services
- Database

This architecture takes advantage of the distributed nature of the web environment.

> **Important:** The Discoverer End User Layer must be in the database before you can configure client access rights and privileges. You must also first install the Discoverer Administration Edition (v3.1.36 or later).

Figure 1–2 The four tiered architecture of Discoverer 3i Viewer


First Tier: The Clients
Use any computer with a Web Browser
Second Tier: The HTTP Server
HTTP Server
Contains the:
1.Discoverer 3i Viewer Servlet
Third Tier: Discoverer Services
Application Server
Contain the:
1. Discoverer Services
2. Preferences
3. Locator
Fourth Tier: The Database
End User Layer (EUL)
Oracle Discoverer 3.1 and 3i workbooks
Database

1.4.1 First tier - the Client

The first tier 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. The only requirement for this computer is that it can run HTML through a web browser, such as Microsoft Internet Explorer 4.01 or Netscape Navigator 4.0. By accessing the URL that you (the Discoverer administrator) provide, users can connect and run Oracle Discoverer 3*i* Viewer to view data.

1.4.2 Second tier - the HTTP Server

The second tier is the HTTP Server having the Discoverer 3*i* Viewer Servlet installed in the HTTP Server's Servlet engine. The Locator must be accessible from the HTTP server.

1.4.2.1 What is a Servlet?

A Servlet comprises modules of Java code that run as a server application to answer HTTP requests (hence the name "Servlets", similar to "Applets" which are used on the client side). The Servlet minimizes any client-side processing.

1.4.2.2 What is a Servlet Engine?

A Servlet Engine comes as part of (or a plug in to) an HTTP Server and is the environment in which the Servlet runs. The Servlet Engine incorporates a Java Virtual Machine (JVM) and implements the Java Servlet API.

1.4.2.3 HTTP Server Components

The HTTP server (Servlet engine) contains the following components:

- The Discoverer Servlet and XML/XSL Processor (generates output pages).

1.4.2.3.1 Discoverer Servlet The Discoverer Servlet generates HTML pages from HTTP requests using an XSL processor.

1.4.3 Third tier - Discoverer Services

The third tier is the Discoverer Services. The Server consists of three basic components: the Locator, the Session, and the Preferences.

The components are installed on one or more **application servers**. You can install Discoverer Services on one machine or across many machines. For more information on install Discoverer Services across multiple machines see the release notes.

In addition, two CORBA (Common Object Request Broker Architecture) components from VisiBroker—an Object Activation Daemon (OAD) and a SmartAgent component—are also installed. These CORBA components are responsible for activating new components and providing a location service whenever the user requests to begin a Discoverer session.

1.4.3.1 Discoverer Services Components

The Discoverer Services contains the following components:

- The Session component.
- The Locator component.
- The Preferences component.

1.4.3.1.1 The Session Component A Discoverer server session is created for each user and performs all the business intelligence of Discoverer 3*i* Viewer for a given user. Once a session is started, it is the user's instance of Discoverer 3*i* Viewer. 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.

Machines with a Session component installed can run multiple concurrent sessions for clients.

1.4.3.1.2 The Locator Component The purpose of the Locator is to receive a request for a session from the Servlet (the request is initiated after a URL is sent to the Servlet from the client browser), find the next application server, start a new session, and return a reference for the new session back to the Servlet. Once the Servlet has received this reference, the Servlet and the session communicate with each other and the Locator waits for the next request. The Locator component is installed on the HTTP server. You need only one Locator component on the network for the Discoverer Services.

1.4.3.1.3 The Preferences Component You should only have one Preferences component for a Discoverer Services environment. So if you have five machines running Discoverer Services, the Preferences component should only be running on one of them. Preference settings are used by Discoverer Services to dictate certain aspects of behavior.

1.4.4 Fourth tier - the database

The fourth tier of Discoverer is the database. The database contains the data users want, the End User Layer (EUL) that provides an easy to understand view of the data, and the Discoverer workbooks that users can work with to view and analyze the data. The EUL is created and maintained with the Discoverer Administration Edition.

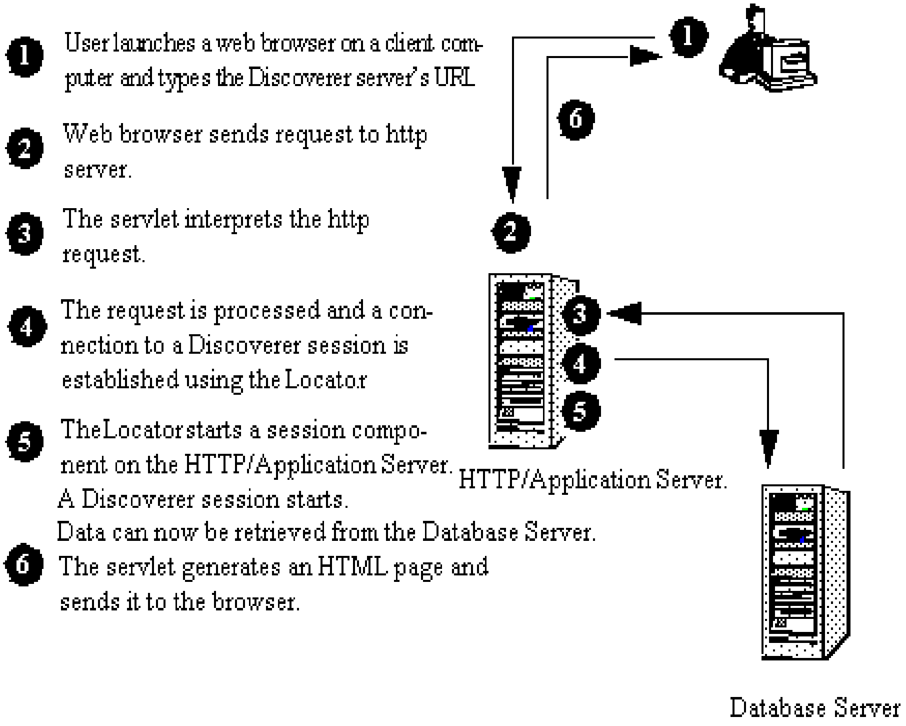
Note: Before users can use Discoverer 3*i* Viewer to view data, the database **must** contain a Discoverer 3.1 EUL. You must create or update the EUL with Oracle Discoverer 3.1.36 or later of the Administration Edition for Windows95/98/NT. If you are currently using Discoverer 3.0.8, you need to upgrade to Discoverer 3.1.36 or later. If you are using Discoverer 3.1.25, you need to install the Discoverer 3.1.36 patch. Discoverer 3.1.36 is Y2K compliant and assures that your EUL and software will work correctly in and beyond the year 2000.

1.5 How Oracle Discoverer 3*i* Viewer Works

This section describes how Oracle Discoverer 3*i* Viewer works—from the point when the user launches the browser on the client computer to when data access becomes available. The following figure illustrates the basic process.

1.5.1 The Oracle Discoverer 3*i* Viewer process

Figure 1–3 The Discoverer 3i Viewer process



1 User launches a web browser on a client computer and types the Discoverer server's URL

2 Web browser sends request to http server.

3 The servlet interprets the http request.

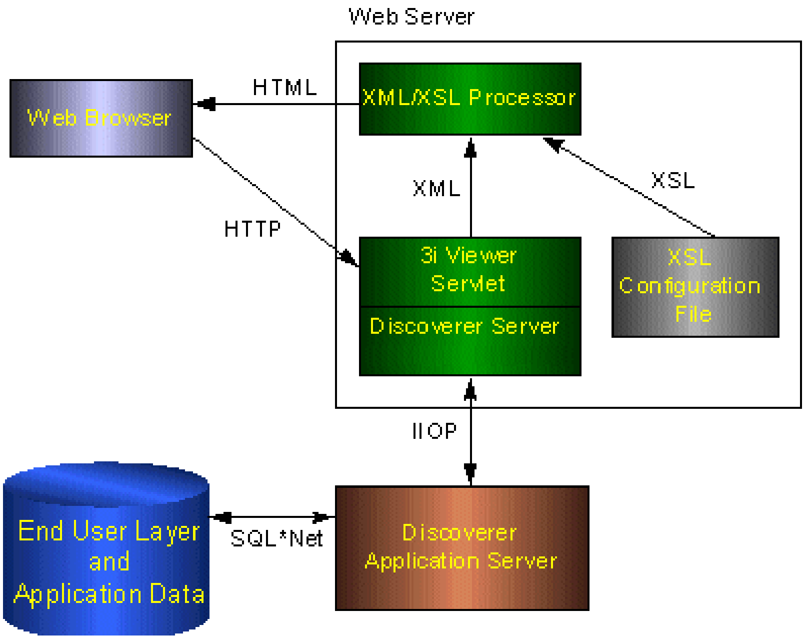
4 The request is processed and a connection to a Discoverer session is established using the Locator

5 TheLocatorstarts a session component on the HTTP/Application Server. A Discoverer session starts. Data can now be retrieved from the Database Server.

6 The servlet generates an HTML page and sends it to the browser.

1.5.2 Oracle Discoverer 3*i* Viewer Architecture

Figure 1–4 The Oracle Discoverer 3i Viewer Architecture



1.5.3 Invoking Oracle Discoverer 3*i* Viewer

Oracle Discoverer 3*i* Viewer is invoked via a URL from a standard web Browser.

The URL is processed by a Servlet running in the Servlet engine. The Servlet uses the Locator to communicate with the Discoverer Services.

Note: The Discoverer Services can be used by both the Oracle Discoverer Plus and Oracle Discoverer 3*i* Viewer, this means that the Discoverer Services need only be installed once for both applications.

1.5.4 Processing the HTTP request

The Oracle Discoverer 3*i* 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.

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.

1.5.4.1 Application and Presentation Logic

The Oracle Discoverer 3*i* Viewer is composed of two primary layers - the application logic and the presentation logic.

The application logic 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 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 user interface (by editing XSL files). See *Chapter 4, "Customizing Oracle Discoverer 3i Viewer"* for details.

1.5.5 Scalability and load balancing

The Oracle Discoverer 3*i* Viewer Servlet and Discoverer Services are automatically installed to run on the Oracle Internet Application Server as a single machine installation. However they are designed to be deployed on multiple machines (See the release notes for further information). 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 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 3*i* Viewer servlet can also be distributed.

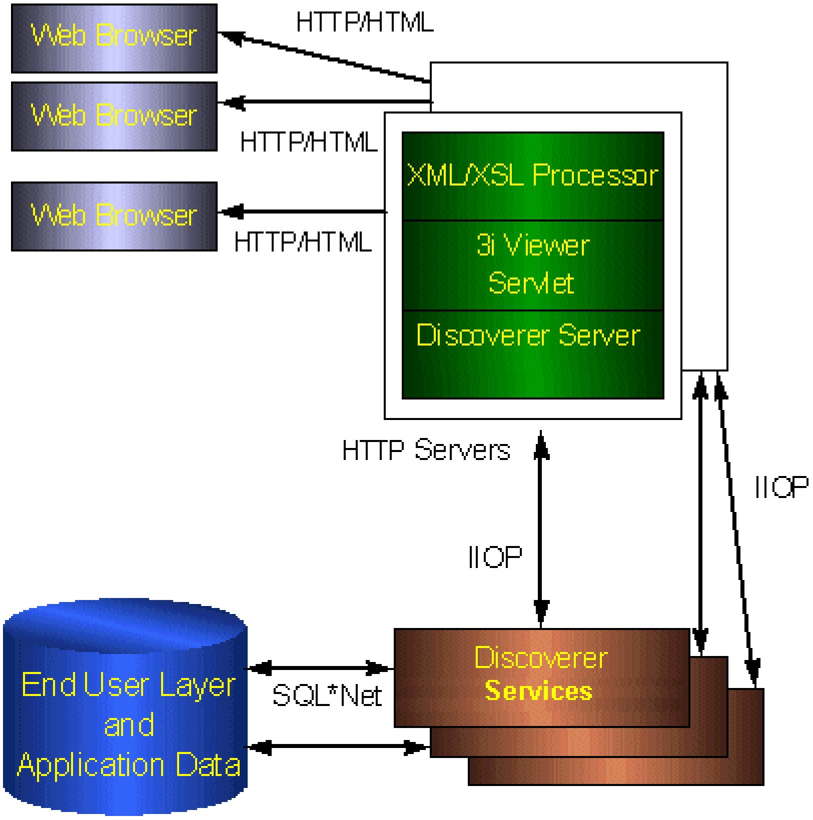


Figure 1–5 Using multiple Discoverer Application Servers for scalability

In Figure 1–5 two HTTP servers are used to spread the load for three browser sessions. There are three server machines, with Discoverer Services running on each. In a real system there would be many users using each HTTP and application server. Discoverer allows you to determine exactly how you want to spread the load across the available machines.

2

Running, Maintaining and Supporting Discoverer 3*i* Viewer

This chapter explains how to run and maintain Discoverer 3*i* Viewer.

The topics include:

- Running Discoverer 3i Viewer
- Maintaining Discoverer 3i Viewer
- Seeing Registered Discoverer Server Objects
- Editing End User Preferences
- Registering and Unregistering Objects
- Shutting Down
- Restarting Discoverer 3i System
- Setting up a secure link using HTTPS
- Frequently Asked Questions

2.1 Running Discoverer 3*i* Viewer

Once you have completed the Oracle Internet Application Server installation (see the *Oracle Internet Application Server Installation Guide*) you can run the Discoverer 3*i* Viewer. The Discoverer components should already be registered (if not see Section 2.5.1.1, "To Register the Preferences and Session with the OAD" for details).

To run Discoverer 3*i* Viewer complete the following steps:

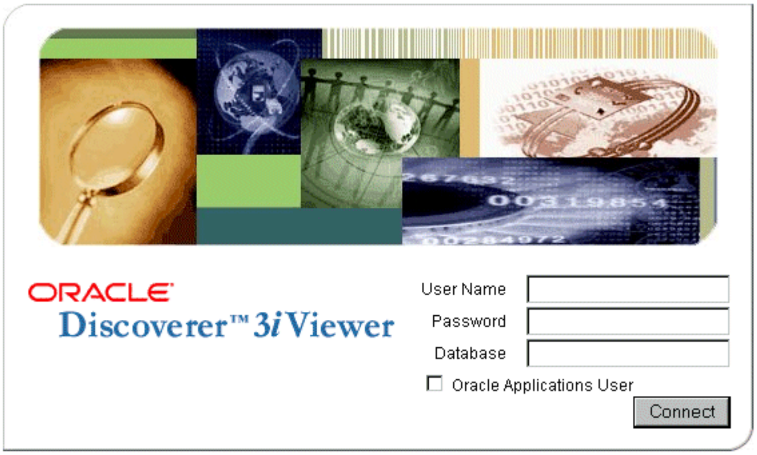
1. Start a browser window and type in your own details using the following as a guide:

http://mywebserver.company.com/servlets/discoservlet

Where **mywebserver.company.com** represents the Web server machine that the HTTP server is installed on. You must replace this with its fully qualified host name. For example:

http://machinename.oracle.com/servlets/discoservlet

This will start Oracle Discoverer 3*i* Viewer and you will see the following login screen:



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, details on how to display it are detailed in an Oracle Applications mode section in the Table 2–1, "Discoverer Server Preferences".
4. Click Connect

You are now running Oracle Discoverer 3*i* Viewer.

2.2 Maintaining Discoverer 3*i* Viewer

To maintain Discoverer 3*i*, first go to the directory **<ORACLE_HOME>/DISCWB33/util**

If you are using C Shell (csh) run the command

```
> source discwb.csh
```

If you are using Korne shell/Bourne shell (ksh/sh) run the command

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

Running this command is very important because this will set the required PATH and other environment variables. These settings are required to run the scripts startoad.sh, startosagent.sh, startlocator.sh, stopoad.sh, stoposagent.sh, stopall.sh, registering and unregistering scripts, plus commands like osfind and oadutil.

2.3 Seeing Registered Discoverer Server Objects

Installed objects are registered with the VisiBroker OAD (Object Activation Daemon). You can see the registered objects for one machine or for the entire network.

To see the registered objects on one machine:

1. Make sure the OAD is running. To do this perform Step 2 below:
   If it is not, run the script startoad.sh.
2.
3. At the command prompt, type:

   `oadutil list` and press return.

> **Note:** Running the oadutil command displays only the list of objects registered on the machine.

2.4 Editing End User Preferences

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

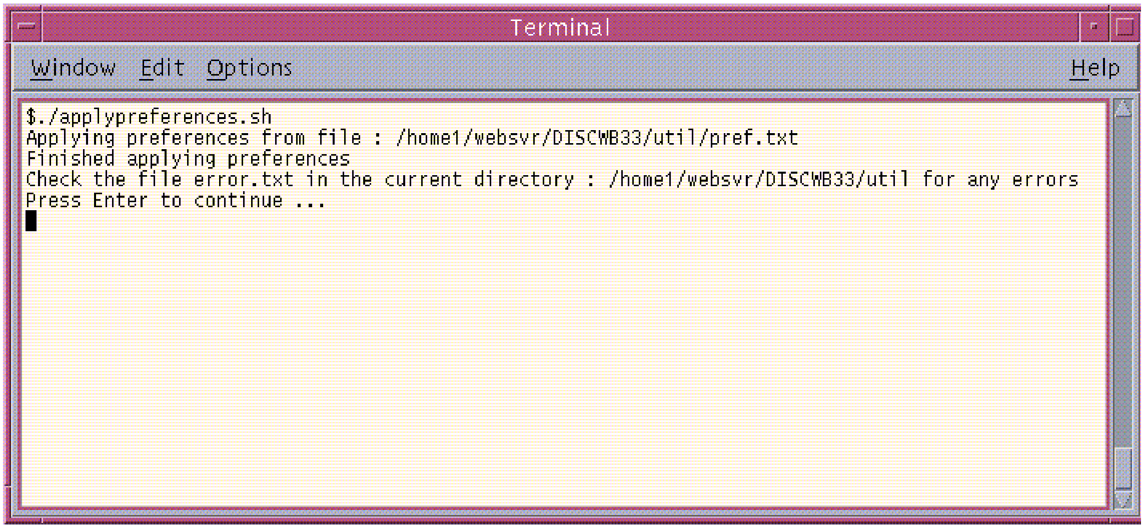
Some of these settings can be customized by users from the Discoverer 3*i* Options page. Individual preferences are stored for each user as a unique combination of database and userid, and loaded whenever the user starts a new session. Therefore, users can log on from different client machines and still have their individual settings take effect.

Also see the "*Discoverer Administration Edition User's Guide*" for more information.

> **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.

1. Open pref.txt, under the following directory in a text editor (vi or emacs):
   <ORACLE_HOME>/DISCWB33/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.
3. Save the file and quit.
4. Run the applypreferences.sh script, to apply preferences. This script is available under **<ORACLE_HOME>/DISCWB33/util** directory.

   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.



```
$./applypreferences.sh
Applying preferences from file : /home1/websvr/DISCWB33/util/pref.txt
Finished applying preferences
Check the file error.txt in the current directory : /home1/websvr/DISCWB33/util for any errors
Press Enter to continue ...
```

5. To continue, press any key.

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

The following table shows the items in the Preferences file:

Table 2–1 Discoverer Server Preferences

Affected Feature	Preference Item Key Name	Description	Default/Values
Locator	MachineIPs	IP address or machine name of all Discoverer Server Application server; **this is the only required item in the Preferences file**	
Session Manager	Timeout	Amount of time after which Discoverer 3*i* 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)
Application	CacheFlushPercentage	Percentage of cache flushed if the cache is full	25

Table 2–1 Discoverer Server Preferences

Affected Feature	Preference Item Key Name	Description	Default/Values
	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
	ShowDialogBitmaps	End users see the bitmap graphics on Discoverer Plus dialogs.	1
	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” halign=”right” 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” halign=”left” valign=”top”> </fontFormat>”

Table 2–1 Discoverer Server Preferences

Affected Feature	Preference Item Key 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” halign=”left” valign=”top”> </fontFormat>”
	NullValue	Specifies how null values are displayed in worksheets, for example, “NULL”, “N/A”, “0”	“NULL”
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)
	ItemClassDelay	Fetch list of values for this amount of time before timing out	15 (seconds)
	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

Table 2–1 Discoverer Server Preferences

Affected Feature	Preference Item Key Name	Description	Default/Values
	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
	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	Specified the schema in which certain vital Oracle Applications data is stored	"APPS"
	ShowUserTypeChoice	Activates an end-user checkbox that allows them to specify whether they want to run Discoverer against an Oracle Application	1
	DefaultUserTypeIsApps	Specifies that users run Discoverer by default against an Oracle Application	1
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)

Table 2–1 Discoverer Server Preferences

Affected Feature	Preference Item Key Name	Description	Default/Values
	SummaryObjectsUseCachedAccessibility	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
	QPPMinActCpuTime	Only records or uses statistics with a CPU time greater than this value	0

Table 2–1 Discoverer Server Preferences

Affected Feature	Preference Item Key Name	Description	Default/Values
	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)

2.5 Registering and Unregistering Objects

The VisiBroker OAD must be running to register and unregister components, using the command line.

2.5.1 Registering and Unregistering Individual Component(s)

You can register components using shell scripts. The shell scripts are located in the **<ORACLE_HOME>/DISCWB33/util** directory of the machine where the components are installed.

2.5.1.1 To Register the Preferences and Session with the OAD

The Preferences and Session must be registered with the OAD. The OAD will spawn 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`
- `registerpreferences.sh`

To register the Session from the command line:

1. Change the directory to **<ORACLE_HOME>/DISCWB33/util**.
2. Type

```
> ./registersession.sh
```

To register the Preferences from the command line:

1. Change the directory to **<ORACLE_HOME>/DISCWB33/util**.
2. Type

```
> ./registerpreferences.sh
```

2.5.1.2 To Unregister Objects from OAD

You can unregister the Preferences and Session from the OAD.

2.5.1.2.1 Unregistering the Preferences

A script is available to help you unregister the Preferences from the OAD.

To unregister the Preferences from the OAD:

1. Change the directory to **<ORACLE_HOME>/DISCWB33/util**.
2. Type

   ```
   > ./unregisterpreferences.sh
   ```

2.5.1.2.2 Unregistering the Session

A script is available to help you unregister the Session from the OAD.

To unregister the Session from the OAD:

1. Change the directory to **<ORACLE_HOME>/DISCWB33/util**.
2. Type

   ```
   > ./unregistersession.sh
   ```

2.6 Shutting Down

If you need to perform maintenance, change hardware settings, or perform any other tasks, you should take the server machine off-line while working. This prevents users from starting sessions on that server while you work.

You will need to shut down the entire Discoverer 3*i* system.

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

2.6.1 Shutting down the entire Discoverer 3*i* system

1. Run the script:

   ```
   stoplocator.sh.
   ```

   available under **<ORACLE_HOME>/DISCWB33/util** directory.
2. This stops the locator. Stopping the Locator prevents any new users from logging in.
3. Run the script:

```
stopoad.sh.
```

available under **<ORACLE_HOME>/DISCWB33/util** directory. This stops the OAD.

4. At Command Prompt, type `ps -A|grep dis33ws` to get the process ID of all the running **dis33ws**. For example the output will be

   ```
   18064 pts/3   0:01   dis33ws
   18072 pts/4   0:08   dis33ws
   ```

5. Run the kill command to kill the process **dis33ws**:

   Example:

   ```
   > kill -9 18064.
   ```

   where 18064 is the ProcessID of dis33ws which is shown in step 4 example's output.

6. Run the script:

   ```
   stoposagent.sh.
   ```

   available under **<ORACLE_HOME>/DISCWB33/util** directory. This stops the SmartAgent.

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

2.7 Restarting Discoverer 3*i* System

You can restart the Discoverer 3*i* System quickly and easily from the command line because all of the components are already registered with the OAD.

2.7.1 Restarting Discoverer 3*i* system using shell files

Use the following process on your Discoverer 3*i* system.

- From the **<ORACLE_HOME>/DISCWB33/util** directory, type:

  ```
  > ./startosagent.sh

  > ./startoad.sh

  > ./startlocator.sh
  ```

2.8 Setting up a secure link using HTTPS

To ensure that your entire discoverer document is secure and to get your browser to acknowledge a “secure” page, it is necessary to make a minor adjustment to your **disco3iv.xml** file.

Make changes as follows:

1. Open your **disco3iv.xml** file for editing
2. Change the **image_path** so to includes ***https*** and the ***full path*** to the image directory.

<argument name="image_path" type="href"> images</argument>

<argument name="image_path" type="href">https://mywebserver.company.com/disco3ivhome/ images</argument>

Below is an example of the change:

disco3iv.xml - Before change

<argument name="help_path" type="href">help</argument>**<argument name="image_path" type="href"> images</argument>**<argument name="show_login_method_control">true</argument></document></disco_config>

disco3iv.xml - After change

<argument name="help_path" type="href">help</argument>**<argument name="image_path" type="href">https://mywebserver.company.com/disco3ivfiles/images</argument>**<argument name="show_login_method_control">true</argument></document></disco_config>

2.9 Frequently Asked Questions

What environment variables are set?

The Discoverer Server sets four environment variables for the VisiBroker products when installed:

VBROKER_ADM = <ORACLE_HOME>/vbroker/adm

VBROKER_JAVAVM = <ORACLE_HOME>/vbroker/bin

OSAGENT_PORT= 14000

VBROKER_IMPL_DIR= <ORACLE_HOME>/vbroker/adm/impl_dir

The Discoverer Server does not modify the CLASSPATH. The PATH variable is updated with:

PATH = $<ORACLE_HOME>/bin:$<ORACLE_HOME>/vbroker/bin:$PATH

This minimizes any conflict with other products.

What happens when the Locator or Preferences objects stop running?

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

The Locator is designed so it does not need to be shut down even when you edit Preferences. By default the Locator will pick up the edited values when the next client is connected. If the preferences object stops running it will automatically be brought up again when either the Locator or any Session tries to connect to it, since it is registered with OAD. Similarly, if the Preferences goes down, the Locator will continue to function. When the Preferences is brought back up, the Locator will bind to the Preferences again.

What do I need to do when I get a network error?

When Discoverer 3*i* Viewer receives a network exception during login, the server is most likely having the problem. Check the following:

- Are there any directories or jar files in the CLASSPATH or PATH that might interfere with Discoverer 3*i*? Discoverer 3*i* uses VisiBroker CORBA V3.4 and JDK 1.2.
- Make sure the Locator is up and running. Did you see any error messages in the locator.log file? Make sure the locator.ior file is in the:

  [HTTP_DIR]/discwb33/applet directory
- Make sure the VisiBroker SmartAgent and OAD are up and running and that a new session can be spawned.

- Make sure the Preferences object is up and running. To see that the Preferences object is up and running use the command `ps-a` and make sure that dis31pr is running.

What do I need to do when 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 aliases are 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 another Oracle product running on that machine, try connecting to the database with that product or SQL*Plus.

What do I need to do when 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"?

Check your startoad.sh script. It should contain the following line.

unset ORA_NLS33

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

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

OAD gives cross-device link error:

- If your VBROKER_ADM directory and /tmp directory are not on the same device. In this case you should set the TMPDIR environment variable to a directory on the same device of VBROKER_ADM, before starting OAD.
- If you don't have write permission for VBROKER_ADM/impl_dir directory also for files under the directory viz. impl_rep and impl_rep~. In this case make sure that you have enabled the write permission.

What do I need to do, when I run registerall.sh it gives me "oadutil: unable to bind to OAD"?

Check the following:

- See whether oad is properly running (using ps command). If not running see oad.log for any errors.
- Make sure there's at least one osagent running in the subnet (You can run one in your machine itself from startosagent.sh).

- If your machine is configured for multiple IP addresses, (to find out run "ifconfig -a" and see if you have more than one network interface configured).

In last case i.e., for multiple IP machines, follow these steps:

Get the IP address recognized by OAD.

```
$. discwb.sh (or in case of c shell "source discwb.csh")
$oad -v
```

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

```
Minor: 0
Completion Status: NO
]
^C
```

Note down this <**IP_ADDR**>

Edit the oadutil file which will be present in ORACLE_HOME/vbroker/bin, change the last line. i.e.

```
${VBJ}-VBJincppjar1-VBJaddJarvbjtools.jarcom.visigenic.vbroker.activ
ationImpl.OADutil $@
         to
${VBJ}-VBJincppjar1-VBJaddJarvbjtools.jarcom.visigenic.vbroker.activ
ationImpl.OADutil $@ -host <IP_ADDR>
```

(Note the addition of **-host** <**ip_addr**>)

3

Accessing Oracle Discoverer 3*i* Viewer From Other Web Applications

Oracle Discoverer 3*i* Viewer can be accessed from other Web applications by specifying a combination of parameters in the URL. Rather than having your end users log in, choose a workbook, and worksheet, you can specify a URL that starts the Oracle Discoverer 3*i* Viewer for them. You can explicitly specify the settings that you want; for example, you can specify which workbook and worksheet you want to open and which parameter values to use.

This means you can embed live Discoverer reports directly onto your Web site pages simply by specifying the details in a URL. When the link is clicked the database is queried and the latest data is displayed in HTML.

After you create this URL, you can give it to end users to type into their Web browser. Or you can add the URL as a link on your company's Intranet site so that end users can make a specific database query with a single click.

In addition, you can also specify an end user's Applications Responsibility in this same URL.

3.1 Using Parameters in URLs

Using parameters in a URL can be used to specify specific factors when accessing Oracle Discoverer 3*i* Viewer from other applications it saves your end users time and gives you control over the data they use.

You can specify username, database, EUL, workbooks, worksheets and worksheet pages through parameters placed in the URL. Passwords are generally not displayed in the URL for security reasons.

3.1.1 Specifying username, database

Format for each parameter is:

&us=<username>
&db=<databasename>

http:mywebserver/disco3iv?us=<username>&db=<databasename>

3.1.2 Specifying display of the Applications User Checkbox in Connect Dialog

You can configure the disco3iv.xml file so that the connect dialog enables or disables display of the Applications User checkbox:

To set up the Connect dialog to display (or hide) the Applications User checkbox, access the configuration file located in **D:\disco3ivhome\**disco3iv\html\disco3iv.xml and set the "**show_login_method_control**" argument to true or false as required.

3.1.3 Specifying a workbook in a URL:

Format for parameter is:

&wb=<wkbkname>

You MUST also specify the username and database name (see above). The order used is not important:

http://myserver/disco3iv?us=<username>&db=<databasename>&wb=<workbookname>

An Example is:

http://myserver/disco/disco3iv?us=qstats&db=disco&wb=nobugs

3.1.4 Specifying a worksheet

This is similar to workbook above.

Format for parameter is:

&ws=<worksheetname>

You must also specify the workbook name, username and database name. These parameters do not have to be in a specific order but they must be entered after the question mark?:

http://mywebserver/disco3iv?us=<username>&db=<databasename>&wb=<workbookname
>&ws=<worksheetname>

An Example is:

http://myserver/disco/disco3iv?us=qstats&db=disco&wb=nobugs&ws=Sheet+1

3.1.5 Specifying worksheet parameters

Format for parameter is:

&qp_<parameter_name>=<parameter_value>

You must also specify the workbook, worksheet name, username and database name. These parameters do not have to be in a specific order but they must be entered after the question mark?:

http://myserver/disco3iv?us=<username>&db=<dbname>&wb=<workbookname>&ws=<worksheetname>&qp_<parameter_name>=<parameter_value>

An Example is:

http://myserver/disco/disco3iv?us=video33&db=video33&wb=wb1&ws=ws2&qp_City=Denver&qp_City=Los+Angeles&qp_Year=1995

The query string specifies Denver and Los Angeles as values for the City parameter and 1995 as the value for the Year parameter for the worksheet with id ws2, in the workbook with the id wb1, in the database - name video33 with user name video33.

3.1.6 Specifying a page number

This is similar to worksheet above.

Format for parameter is:

&pg=<pagenumber>

You must also specify the workbook, worksheet name, username and database name. These parameters do not have to be in a specific order but they must be entered after the question mark?:

http://myserver/disco3iv?us=<username>&db=<dbname>&wb=<workbookname>&ws=<worksheetname>&pg=<pagenumber>

An Example is:

http://myserver/disco/disco3iv?us=video33&db=video33&wb=wb1&ws=ws2&pg=2

The query string specifies the second page in worksheet with id ws2, in the workbook with the id wb1, in the database - name video33 with user name video33.

Note: If the Servlet was supplied this query string the user would be challenged for a password, before opening the worksheet.

4

Customizing Oracle Discoverer 3*i* Viewer

Oracle Discoverer 3*i* 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 Oracle Discoverer 3i Viewer for Customized Web Applications
- Customization Examples

4.1 Using Oracle Discoverer 3*i* Viewer for Customized Web Applications

Customize the general appearance

The appearance of Oracle Discoverer 3*i* 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.

Customize the application

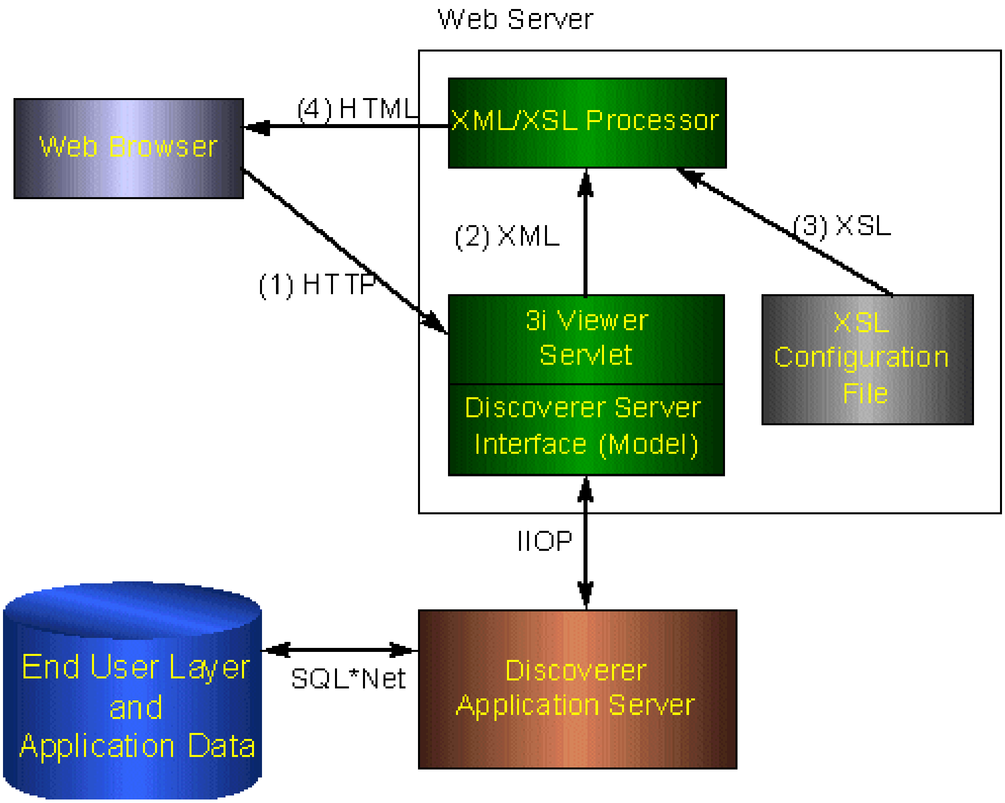
Oracle Discoverer 3i Viewer can be used to build complete custom Web applications or deliver data to other middle tier Web systems. It uses industry standard XML to represent data and application state, and the XSL stylesheet language to format the User Interface. Standard XSL tools can be used to customize the User Interface or to produce a complete embedded Business Intelligence application.

How the HTML is produced

All the HTML generated by Oracle Discoverer 3*i* 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 configuration file 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 4–1 - Oracle Discoverer 3I Viewer HTML Generation using XSL/HTML



4.1.1 Steps to Generate the HTML needed to build a specific Discoverer Application

4.1.1.1 Step 1 Browser sends URL

After login, let’s assume a Discoverer Viewer 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/disco/disco3iv?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.

4.1.1.2 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="3.3.x.x" login_method="discoverer">
   <request source="http://myserver.mycompany.com/servlets/viewer">
      <command name="_act">Connect</command>
      <command name="db">disco</command>
      <command name="in">dwb</command>
      <command name="_in">dwb</command>
      <command name="lc">true</command>
      <command name="us">VIDEO</command>
      <command name="JServSessionId">3fb75a688d4d3c68.97.951319520796</command>
   </request>
   <account>
      <user>VIDEO</user>
      <database>disco</database>
      <eul default="true" name="VIDEO">
         <workbook name="Annual Sales Report" ref="Annual+Sales+Report">
            <description>Shows yearly and quarterly sales of
products</description>
         </workbook>
         <workbook name="Store and Band Analysis" ref="Store+and+Band+Analysis">
            <description>Shows sales by Store, broken into sales
bands</description>
         </workbook>
         <workbook name="Video Sales Analysis" ref="Video+Sales+Analysis">
            <description>General purpose analysis of the Business </description>
         </workbook>
      </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.

4.1.1.3 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 Oracle Discoverer 3*i* 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.

4.1.1.4 Step 4 Generate HTML

Given the XML above, the standard Oracle Discoverer 3*i* 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 Oracle Discoverer 3*i* Viewer, the HTML generated does not use frames or javascript, and therefore makes minimal demands on the browser or Internet device used. It also means that it is easy to integrate Oracle Discoverer 3*i* Viewer with other Web Applications or Portals.

4.2 Customization Examples

This section will help you carry out a customization of the Oracle Discoverer 3*i* 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.

4.2.1 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 Oracle Discoverer 3*i* Viewer in the root directory of the Web server.

```
disco3iv/
        disco3iv.jar
        html/
            disco3iv.xml
            disco3iv.xsl
            errors.xsl
            functions.xsl
            gui_components.xsl
            page_layouts.xsl
            style.xsl
            demos/          Video Stores Demo
            help/           HTML help files
            images/         User inteface and help
        doc/                Installation/Administration Doc
```

4.2.2 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 *disco3iv/)* that defines special ‘tags’ for each style that can be modified.

You will need to be able to access the XSL files on the server.

4.2.2.1 Quick customization attributes

Customizable attributes in the *style.xsl* file

- Company Logo
- Text Color
- Headings
- Fonts

- Styles
- Links
- Images

4.2.2.2 Quick customization steps

To complete a Quick customization, take the following steps:

- Locate the file *style.xsl* in the Servlet directory \disco3iv\html.
- Open *style.xsl* in a text editor.
- Edit text within the >angle brackets< for each variable you want to change. For further information, read the detailed comments inside the file itself.
- Save your changes and exit.

4.2.2.3 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.

Many global style changes can be made in this way, but the overall operation of the User Interface will remain unchanged.

Another way of customizing Oracle Discoverer 3*i* Viewer is by using XSL, which allows a complete customized application to be made, as the next section explains.

4.2.3 Full Customization

This section provides information to help you create a fully customized user interface to the Oracle Discoverer 3*i* 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
- Customization Summary

4.2.3.1 Background

Before you start creating a custom interface to the Oracle Discoverer 3*i* Viewer Servlet, it is useful to understand how it works.

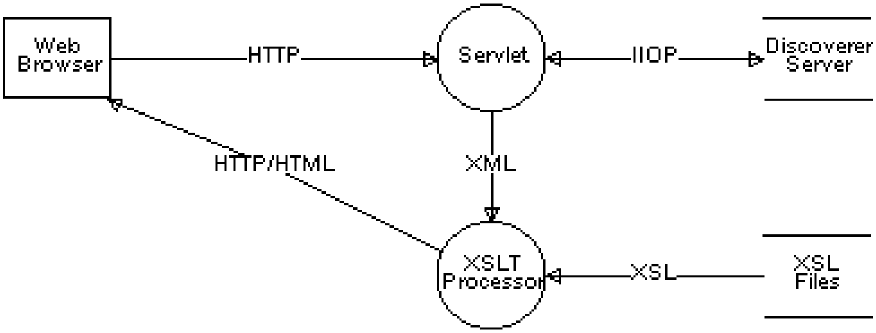
The following steps describe the flow of data that occurs when a request is made:

- 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 4–2 illustrates the data flow in Oracle Discoverer 3*i* Viewer.

Figure 4–2 Oracle Discoverer 3i Viewer Data Flow



4.2.3.2 Prerequisites

To be able to create your own custom user interface for Oracle Discoverer 3*i* 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

4.2.3.3 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 required for the presentation logic. These files are described below.

4.2.3.3.1 disco3iv.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.

4.2.3.3.2 page_layouts.xsl

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

4.2.3.3.3 gui_components.xsl

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

- Page Title
- Page FooterList of Worksheets
- Options Form
- Query Progress Meter

4.2.3.3.4 style.xsl

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.

4.2.3.3.5 functions.xsl

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

4.2.3.3.6 errors.xsl

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

4.2.3.4 Servlet Programming Interface

4.2.3.4.1 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 Servlet Configuration File and "Presentation" commands from the URL to the XSL.

4.2.3.4.2 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.

4.2.3.5 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.

4.2.3.5.1 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"?>
<discoverer3iv version="3.3.x.x" state="login"
url="http://myserver.mycompany.com/disco/disco3iv" login_method="disco_
login">
   <account nv="NULL" aq="confirm">
      <user>video</user>
      <database>disco</database>
      <eul>
         <name>VIDEO</name>
         <workbookinfo>
            <name>Store and Band Analysis</name>
            <description>Shows sales by Store, broken into sales
bands</description>
         </workbookinfo>
         <workbookinfo>
            <name>Video Sales Analysis</name>
            <description>General purpose analysis of the Business
</description>
         </workbookinfo>
         <workbookinfo>
            <name>Annual Sales Report</name>
            <description>Shows yearly and quarterly sales of
products</description>
         </workbookinfo>
      </eul>
   </account>
</discoverer3iv>
```

It starts by specifying the XML version. The 2nd line specifies the XSL file to be applied to process the data, “example1.xsl” and the rest of the file is generated from the Oracle Discoverer 3*i* 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.

4.2.3.5.2 Step 2 - The XSL file

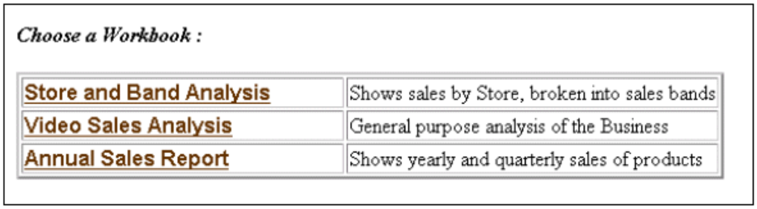
The XSL file “example1.xsl” looks like this:

```
<?xml version=’1.0’?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-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>
```

4.2.3.5.3 Step 3 - XML+XSL = HTML

When the XML file is opened in a Web Browser, it reads in the XSL stylesheet and generates HTML which looks like this:

Figure 4–3 List of workbooks using XML + XSL example



Choose a Workbook :

Store and Band Analysis	Shows sales by Store, broken into sales bands
Video Sales Analysis	General purpose analysis of the Business
Annual Sales Report	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 2nd 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 3*i* this link target is generated dynamically, but it is shown as a static link here for simplicity.

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

This XSL line inserts the text from the XML file for the <NAME> tag under each workbook section.

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

These lines define the 2nd column in the HTML table and insert the text for the workbook description using the <DESCRIPTION> tab 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 Oracle Discoverer 3*i* 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 Oracle Discoverer 3*i* Viewer, the XML and XSL are combined in the XSL processor on the middle tier, and not in the Web Browser.

4.2.3.5.4 Step 4 - Customize the XSL

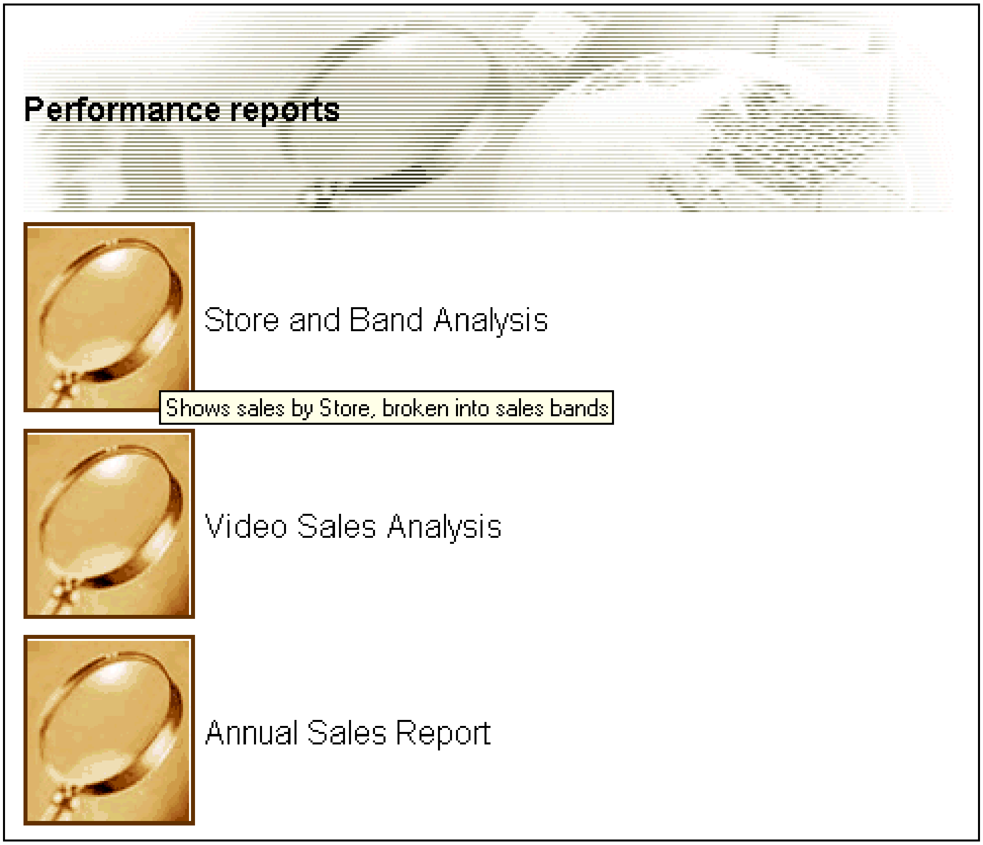
We can modify the XSL file to appear as follows:

```
<?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">
                  <img src="../../images/start1.gif">
                    <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, is appears as:

Figure 4–4 List of workbooks using the same XML with a modified XSL stylesheet



Performance reports

Store and Band Analysis

Shows sales by Store, broken into sales bands

Video Sales Analysis

Annual Sales Report

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>
          <TABLE border="0">
```

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">
    </TABLE>
```

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">
              <img src="../../images/start1.gif">
                <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:

```
    <img src="start1.gif" alt=”Tooltip text to appear when a mouse is
    over the image”>
```

but since we want the tool tip to be dynamic we generate the ALT tag by getting the text from the <DESCRIPTION> tab 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.

4.2.4 Video Stores Demo - an example of an alternative user interface

The Video Stores demo is installed with Oracle Discoverer 3*i* Viewer and can be viewed following completion of a number of setup steps.

Once Oracle Discoverer 3*i* Viewer has been fully installed you can configure your Web server to run the Video Stores demo.

The Video Stores demo illustrates how the Oracle Discoverer 3*i* 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.

4.2.4.1 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 [Disco3iv_ Home]/**disco3iv/html** directory.

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

- *disco3iv.xml*
  You may need to edit this file to ensure that it refers to the correct session name.
  *The **disco3iv.xml** file contains configuration information such as where to find images for the user interface (See disco3iv.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 spacial details.
- Workbooks - These are used in the Video Stores Demo (See Section 4.2.4.2, "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.

4.2.4.2 Set up steps

You need to carry out the following steps:

- Create a Video Stores Demo database user
  (See the *Oracle Discoverer 3.1 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 3.1 Administration Guide*, section on installing the Tutorial for details).
- Using Oracle Discoverer Plus, save the three Video Stores Demo workbooks into the database as the Video Stores Demo user.
- Edit the **video.xsl** file to point the **username**, **password** and **database name** to the newly created Video Stores Demo user. The ***video.xsl*** 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.xsl***.

4.2.4.3 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\disco3iv.xml (Video Stores Demo configuration file)

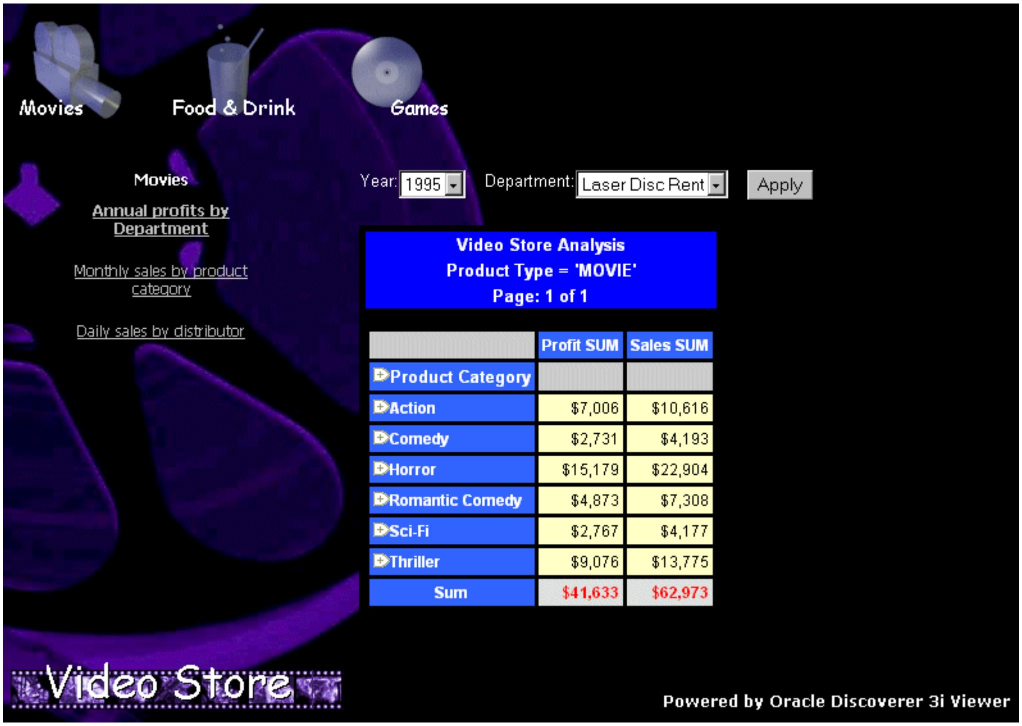
4.2.4.4 Run the Video Stores demo

Open a Web browser and enter the following URL:

- http://yourmachineURL/’Your new Video Demo Zone’

The Video Stores demo will be displayed:

Figure 4–5 The Video Store demo


Movies
Food & Drink
Games
Movies
Annual profits by Department
Monthly sales by product category
Daily sales by distributor
Apply
Video Store
Powered by Oracle Discoverer 3i Viewer

Year: 1995 Department: Laser Disc Rent

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

4.2.4.5 Modify the video.xsl file

The video.xsl file located in *[Disco3iv_Home]/disco3iv/html/demos/video/* specifies the ‘look and feel’ for the Video Stores Customization of Oracle Discoverer 3*i* 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.

4.2.5 Customization Summary

XML and XSL provides a powerful customization environment for Oracle Discoverer 3*i* 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 Oracle Discoverer 3*i* Viewer provides the way to integrate Business Intelligence directly into other Web applications.

Index