HP OpenView Reporting Server Using Radia

for the Windows operating system

Software Version: 4.1

Installation and Configuration Guide

Manufacturing Part Number: T3424-90113

June 2005

Legal Notices

Warranty

Hewlett-Packard makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices

Trademark Notices

Linux is a registered trademark of Linus Torvalds.

Microsoft®, Windows®, and Windows® XP are U.S. registered trademarks of Microsoft Corporation.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Acknowledgements

PREBOOT EXECUTION ENVIRONMENT (PXE) SERVER
Copyright © 1996-1999 Intel Corporation.

Support

Please visit the HP OpenView web site at:

http://www.managementsoftware.hp.com/

This web site provides contact information and details about the products, services, and support that HP OpenView offers.

You can also go directly to the support web site at:

http://support.openview.hp.com/

HP OpenView online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valuable support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track progress on support cases
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and log in. Many also require a support contract.

To find more information about access levels, go to:

http://support.openview.hp.com/access_level.jsp

To register for an HP Passport ID, go to:

https://passport2.hp.com/hpp/newuser.do

Revisions

The version number on the title page of this document indicates the software version. The print date on the title page changes each time this document is updated.

You will receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Chapter 2: Creating the Reporting Server Environment

4.1.1 Page 18, Reporting Server Requirements: Updated Patch database requirements to 2.0 SP1.

Contents

1 Introduction

At the end of this chapter, you will:

- Be familiar with the HP OpenView Reporting Server Using Radia (Reporting Server) reports.
- Be able to create a Radia Reporting environment.
- Understand the contents of this guide.

Overview

As part of the Radia extended infrastructure, the web-based Reporting Server allows you to query the combined data in the existing databases of the HP OpenView Inventory Manager Using Radia (Inventory Manager), HP OpenView Patch Manager Using Radia (Patch Manager), and HP OpenView Usage Manager Using Radia (Usage Manager), and create detailed reports. In addition, you have the option of mounting an existing LDAP directory, which allows you to filter your data using your LDAP directory levels.

The Reporting Server interface provides a dynamic and intuitive way to use Radia SQL data for reporting and overall environmental assessment.

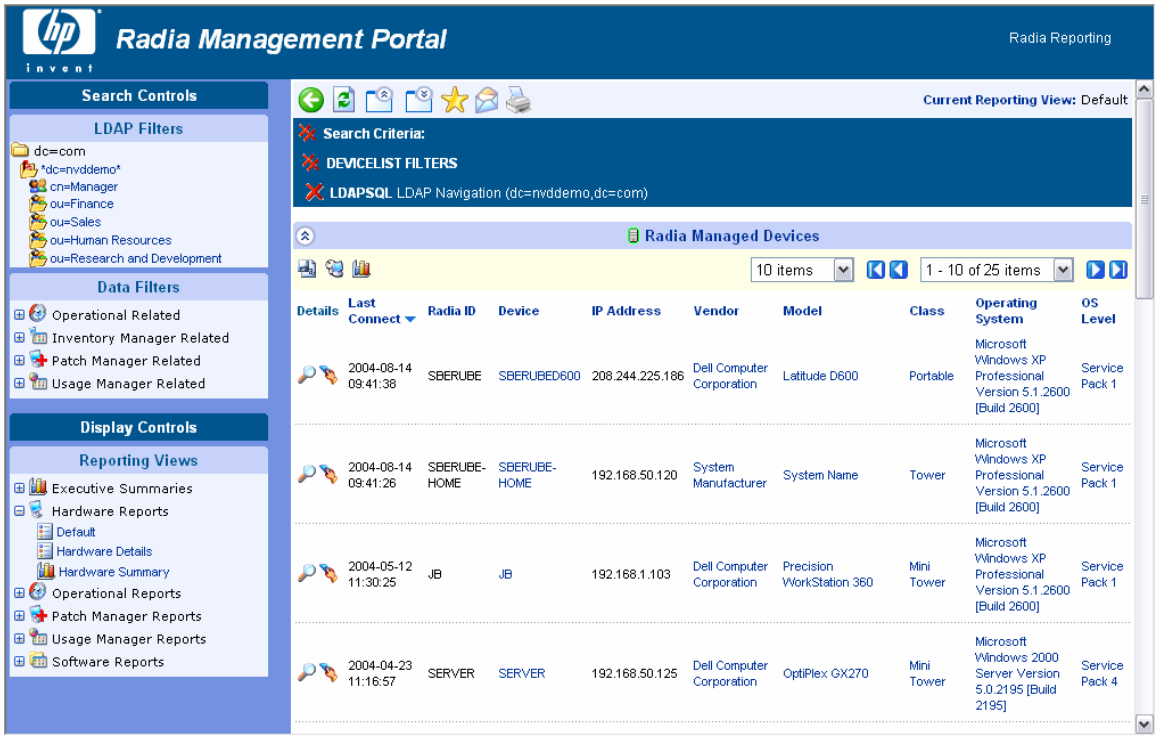


Figure 1: Reporting Server Web interface.

The Reporting Server Environment

A Reporting environment is illustrated in Figure 2 below.

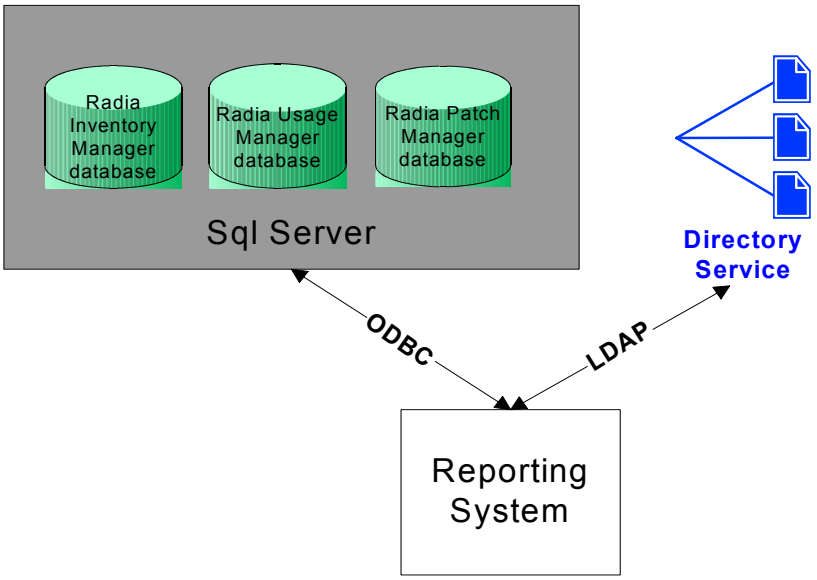


Figure 2: Reporting Environment.

The Radia Reporting environment includes the following required and optional components:

- **Reporting System Modules**
  The Reporting System modules do not require the Management Portal. They can be installed on any Win32 machine, which includes the required Web server, discussed below.

  The Management Portal is only required if you intend to use the Reporting Server to notify devices.

- **Web Server**
  The Reporting Server requires the Microsoft Internet Information Service (IIS) component for Web services, which is available as a component of all Win32 platforms and Windows Server 2003. For more information on IIS, refer to your Windows Operating System documentation.

- **Connections to Radia SQL Databases**
  Radia Reporting can access any Radia SQL database, such as those for Inventory Manager, Patch Manager, and Usage Manager. However, all SQL databases accessed by the Reporting Server must exist on a single SQL Server.
- **Connections to LDAP Directory** (optional)
  The Reporting Server supports optional access to an existing LDAP directory in your enterprise. Access to an LDAP directory allows you to filter report data according to the directory entries.

Terminology

Become familiar with the following terms used throughout this guide.

bulletin

A bulletin is a vulnerability reported by Microsoft about one of their products.

patch

The patch is the actual file to be deployed and executed to fix a vulnerability. A bulletin may have multiple patches depending on platform, number of bits, and language.

qnumber

A qnumber is equivalent to the ticket opened by Microsoft Support. One bulletin can have multiple qnumbers.

Summary

- The Radia Reporting Environment requires access to Radia SQL Databases, such as inventory, patch, and usage. All databases accessed by the Reporting Server must exist in the same SQL Server.
- Optionally, the Radia Reporting Environment can access an existing LDAP Directory in your enterprise.
- The Reporting Server must be installed on a Win32 platform with the Microsoft IIS component (for HTTP Web services).

2 Creating the Reporting Server Environment

At the end of this chapter, you will:

- Know the prerequisites for obtaining reports with the Reporting Server.
- Be able to apply the prerequisites required for using the Reporting Server.
- Be able to add all WBEM components needed for the Reporting Server to your Inventory Manager Reporting audit package.
- Be able to configure the Reporting Server to connect to your databases.
- Optionally, be able to configure the Reporting Server to connect to an existing LDAP directory.
- Be able to access the Reporting Server Web site.

Reporting Server Requirements

As always, be sure to test any recommended or required environment changes before they are implemented into production.

The Reporting Server provides unified access to your existing SQL Server or Oracle databases including the following Radia products:

- **Inventory Manager**
- **Patch Manager** - For patch reporting, a patch database created with the Patch Manager version 2.0 SP1 or higher is required.
- **Usage Manager** - For usage reporting, a usage database created with the Usage Manager version 1.9.2 or higher is required (version 2.0 is recommended).

The Reporting Server software is located on the Radia Infrastructure CD-ROM in the extended infrastructure directory.

The Reporting Server can be installed on any Windows computer system that includes the following:

- An Internet Information Services (IIS) Web server. IIS is provided as a component of Win32 Operating Systems.
- Access to either a SQL Server (version 7 or above) or Oracle server (version 8i or 9i) where your Radia extended infrastructure databases are defined for existing inventory, patch, and usage data.

  All databases accessed from a configured Reporting Server must exist on the same SQL Server or Oracle server.

- If desired, access to your enterprise's LDAP directory, such as Active Directory. LDAP access allows you to filter Reporting Server queries according to the directory information.
- If you will be notifying devices selected from the Reporting Server, the Management Portal is required. The Reporting Server sends the notify requests to the Management Portal. Monitor the status of notify jobs using the Management Portal.

  In order to view the Reporting Server graphical reports using Windows Server 2003, Java Runtime or Virtual Java Machine is required. For more information, go to **http://java.com/en/index.jsp.**

Radia Infrastructure Prerequisites

The list below displays the Radia Infrastructure modifications required to support the Reporting Server. Detailed instructions follow the list.

The following update procedures for the Integration Server and the Inventory Manager apply only to Configuration Servers and Inventory Managers prior to version 4.0. The Configuration Server update applies to all versions of the Configuration Server.

Radia Infrastructure Updates

- Apply the update for `taskend.tcl` for the HP OpenView Integration Server Using Radia (Integration Server) for pre-Radia 4.0 versions of the Integration Server.
- Apply the update for `taskend.tcl` for the HP OpenView Configuration Server Using Radia (Configuration Server).
- Apply the update for `device.config.sql` for Inventory Manager to pre-Radia 4.0 versions of the Inventory Manager.

Integration Server Update for pre-Radia 4.0 users

The `extended_infrastructure\reporting_server\win32\PreReq\RCS\LIB` directory contains a new `taskend.tcl` file to support Radia Reporting.

To apply the taskend.tcl file to your Configuration Server

1. Stop the Configuration Server Service.
2. Make a backup copy of your existing `taskend.tcl` (located, by default, in *`SystemDrive`*`:\Novadigm\IntegrationServer\etc\rim\lib`).
3. Copy the `taskend.tcl` file from the `\extended_infrastructure\reporting_server\win32\PreReq\RCS\LIB` directory to your Integration Server `\etc\rim\lib` folder (by default, *`SystemDrive`*`:\Novadigm\IntegrationServer\etc\rim\lib`).
4. Start the Configuration Server Service.

Configuration Server Update

The `\extended_infrastructure\reporting_server\win32\PreReq\RCS\LIB` directory contains a new `taskend.tcl` file to support Radia Reporting.

To apply the taskend.tcl file to your Configuration Server

1. Stop the Configuration Server Service.
2. Make a backup copy of your existing `taskend.tcl` (located, by default, in *`SystemDrive`*`:\Novadigm\ConfigurationServer\Lib`).
3. Copy the `taskend.tcl` file from the `\extended_infrastructure\reporting_server\win32\PreReq\RCS\LIB` directory to your Configuration Server `\Lib` folder (by default, *`SystemDrive`*`:\Novadigm\ConfigurationServer\Lib`).
4. Start the Configuration Server Service.

Inventory Manager Server Update for pre-Radia 4.0 users

The `\extended_infrastructure\reporting_server\win32\PreReq\RIM\ETC\SQL` directory contains a new `device.config.sql` file to support Radia Reporting.

To apply device.config.sql to your Radia Inventory Server

1. Stop the Configuration Server Service running the Radia Inventory Server.
2. Make a backup copy of your existing `device.config.sql` (located, by default, in *`SystemDrive`*`:\Novadigm\IntegrationServer\etc\sql`).
3. Copy the `device.config.sql` file from the `\extended_infrastructure\reporting_server\win32\PreReq\RIM\ETC\SQL` directory to your Inventory Manager Sever `\etc\sql` directory (by default, *`SystemDrive`*`:\Novadigm\IntegrationServer\etc\sql`).
4. Start the Configuration Server Service.

Radia Database Prerequisites

Modifications must be made to your Radia Databases to accommodate the Reporting Server. The following sections describe which modifications must be made whether you are using Oracle or SQL Server to store your databases.

Radia SQL Server Database Prerequisites

The following modifications must be applied to each SQL Server database the Reporting Server accesses.

A set of SQL scripts was supplied with your Reporting Server media, located in the `\Prereq\SQL\` directory. The scripts are located within the RIM, RUM, and RPM, subdirectories. Depending on the Radia databases you will be using, run the appropriate script using the Microsoft SQL Server Enterprise Manager Query Analyzer tool. If you need assistance running these scripts, see your database administrator.

SQL Creation Scripts are as follows:

- `Prereq\SQL\RIM\RIM PreReq Creation Script.sql`
- `Prereq\SQL\RUM\RUM PreReq Creation Script.sql`
- `Prereq\SQL\RPM\RPM PreReq Creation Script.sql`

Before running these scripts, first review them using a text editor and make sure the default table owner names are correct. Make any changes and save the files.

Inventory Manager Database Updates for SQL Server

Make the following changes to an existing Inventory Manager SQL database to support Radia Reporting.

To update your Inventory Manager SQL Server database

A new field is required for the DeviceConfig table. Modify the Table definition of DeviceConfig by inserting the **devicename** column below the **protocol** column with the following attributes.

1 Using the SQL Enterprise Manager, select **Tables** for the database containing RIM data.

2 Right-click the **DeviceConfig** table in the right-hand pane and select **Design Table** from the context menu.

3 Right-click the protocol column and select **Insert Column** from the context menu. Add the new column with the following information:

Column Name	Data Type	Length	Allow Nulls
devicename	varchar	128	√

4 Use the Microsoft SQL Server Enterprise Manager Query Analyzer and run the script `RIM PreReq Creation Script.sql` against your Inventory Manager database. In the SQL Server Enterprise Manager, select the **Tools** menu, and then select **SQL Server Query Analyzer**.

5 From within the Query Analyzer, open the file `\Prereq\SQL\RIM\RIM PreReq Creation Script.sql`.

6 Use **Query Execute** or press **F5** to run execute the script.

7 Verify that the script completes without error. Consult your SQL DB administrator if changes to the script are required regarding table ownership.

8 Close the SQL Query Analyzer.

Three views are created: **DevicesPrimaryWBEM**, **DevicesPrimaryCIM**, and **DevicesPrimary**. In addition, a **DataAlias** table is created and populated.

If you are using machine connects only to populate your Radia Inventory Database, a database administrator can update the **devicename** column using the following syntax:

```
update deviceconfig set devicename = device_id
```

This will allow the **devicename** column to be populated with the value in device_id.

Usage Manager Database Update for SQL Server

A Database Administrator needs to make the following change to an existing Usage Manager SQL database to support Radia Reporting.

- Use the Microsoft SQL Server Enterprise Manager Query Analyzer and run the script `RUM PreReq Creation Script.sql` against your Usage Manager database. This will add a User Defined Function object called **fn_USAGESTATUS**, to calculate usage status.

Patch Manager Database Update for SQL Server

A Database Administrator needs to make the following change to an existing Patch Manager SQL database to support Radia Reporting.

- Use the Microsoft SQL Server Enterprise Manager Query Analyzer and run the script `RPM PreReq Creation Script.sql` against your Patch Manager database. This will add a User Defined Function object called **fn_PATCHSTATUS**.

Radia Oracle Database Prerequisites

The following modifications must be applied to each Oracle database the Reporting Server accesses.

A set of scripts was supplied with your Reporting Server media, located in the `\Prereq\Oracle\` directory. The scripts are located within the RIM, RUM, and RPM, subdirectories. Depending on the Radia databases you will be using, run the appropriate script. If you need assistance running these scripts, see your database administrator.

SQL Creation Scripts are as follows:

- `Prereq\Oracle\RIM\RIM PreReq Creation Script.oracle`
- `Prereq\Oracle\RUM\RUM PreReq Creation Script.oracle`
- `Prereq\Oracle\RPM\RPM PreReq Creation Script.oracle`

Before running these scripts, first review them using a text editor and make sure the default schema names are correct. Make any changes and save the files. When finished, verify that the scripts completes without error. Consult your Oracle DB administrator if changes to the script are required regarding table ownership.

Inventory Manager Database Updates for Oracle

Make the following changes to an existing Inventory Manager Oracle database to support Radia Reporting.

To update your Inventory Manager Oracle database

1 Use the Oracle DBA Studio application (for Oracle version 8i and below) or the SQL Plus Worksheet application (for Oracle version 9i and above) and execute the script `RIM PreReq Creation Script.oracle`, making sure to include the correct path to the script locations.

Three views are created: **DevicesPrimaryWBEM**, **DevicesPrimaryCIM**, and **DevicesPrimary**. In addition, a **DataAlias** table is created and populated.

2 A new field is required for the **DeviceConfig** table. Modify the Table definition of **DeviceConfig** by inserting the **devicename** column below the **protocol** column with the following attribute:

Column Name	**Data Type**	**Length**	**Allow Nulls**
devicename	varchar	128	√

If you are using machine connects only to populate your Radia Inventory Database, a database administrator can update the **devicename** column using the following syntax:

```
update deviceconfig set devicename = device_id
```

This will allow the **devicename** column to be populated with the value in device_id.

Usage Manager Database Update for Oracle

A Database Administrator needs to make the following change to an existing Usage Manager Oracle database to support Radia Reporting.

- Use the Oracle DBA Studio application (for Oracle version 8i and below) or the SQL Plus Worksheet application (for Oracle version 9i and above) and execute the script `RUM PreReq Creation Script.oracle`, making sure to include the correct path to the script locations. This will add a User Defined Function object called **fn_USAGESTATUS**, to calculate usage status.

Patch Manager Database Update for Oracle

A Database Administrator needs to make the following change to an existing Patch Manager Oracle database to support Radia Reporting.

- Use the Oracle DBA Studio application (for Oracle version 8i and below) or the SQL Plus Worksheet application (for Oracle version 9i and above) and execute the script `RPM PreReq Creation Script.oracle`, making sure to include the correct path to the script locations. This will add a User Defined Function object called **fn_PATCHSTATUS**.

Radia Inventory Reporting Auditing Requirements

When using the Reporting Server to view a Inventory Manager database, the WBEM instances listed in Table 1 below should be enabled for the Inventory Manager Reporting Package in the Audit class of your Radia Database. These fields are relied upon to produce the primary reporting table or detailed reports shown in later chapters.

For details on how to enable these options, see *Adding Components to Radia Inventory Audits* on page 44.

Table 1: Inventory Manager Reporting WBEM Instances to Enable Radia Reporting

WBEM Class Instance	WBEM Class Instance
Win32_Bios	Win32_PointingDevice
Win32_ComputerSystem	Win32_Printer
Win32_ComputerSystemProduct	Win32_Processor
Win32_DesktopMonitor	Win32_Product
Win32_DiskDrive	Win32_SerialPort
Win32_DiskPartition	Win32_Service
Win32_Environment	Win32_Share
Win32_Group*	Win32_SoundDevice
Win32_Keyboard	Win32_TimeZone
Win32_LogicalDisk	Win32_USBController
Win32_LogicalMemoryConfiguration	Win32_UserAccount*
Win32_MotherboardDevice	Win32_VideoController
Win32_NetworkAdapter	Win32_CDROMDrive
Win32_NetworkAdapterConfiguration	Win32_Process
Win32_OperatingSystem	Win32_SystemEnclosure
* Queries may require additional changes. See caution below.	

When auditing for Win32_UserAccount or Win32_Group, large amounts of data may be returned. Failure to limit the scan may result in high network traffic. In order to limit the amount of data returned by these queries, modify the class.

In order to restrict the results to LOCAL user accounts and LOCAL groups only, modify the CNDITION field of the Win32_UserAccount and Win32_Group classes by adding the following syntax:

```
CNDITION      Domain = "&(zconfig.zhdwcomp)"
```

Be sure to check the HP OpenView support web site for the most recent information on this topic.

Reporting Server System Implementation Tasks

The Reporting Server software and components is located on the Radia Infrastructure CD-ROM in the extended infrastructure directory (`\extended_infrastructure\reporting_server\win32\`).

To use Radia Reporting, complete the following system implementation tasks:

- ❑ Review the SQL Server ODBC connections and DSNs for each Radia Database.
- ❑ Install the Reporting Server.
- ❑ Configure Internet Information Services for a Radia Reporting `*.tcl` Extension and Web Sharing.
- ❑ Modify the Reporting Server configuration file.

Reviewing your SQL Server Access and Database DSNs

All SQL databases accessed by the Reporting Server must exist on the same SQL Server. In order to complete the Reporting Server configuration in the tasks that follow, you will need to know the DSNs and access credentials assigned to each database in the SQL Server.

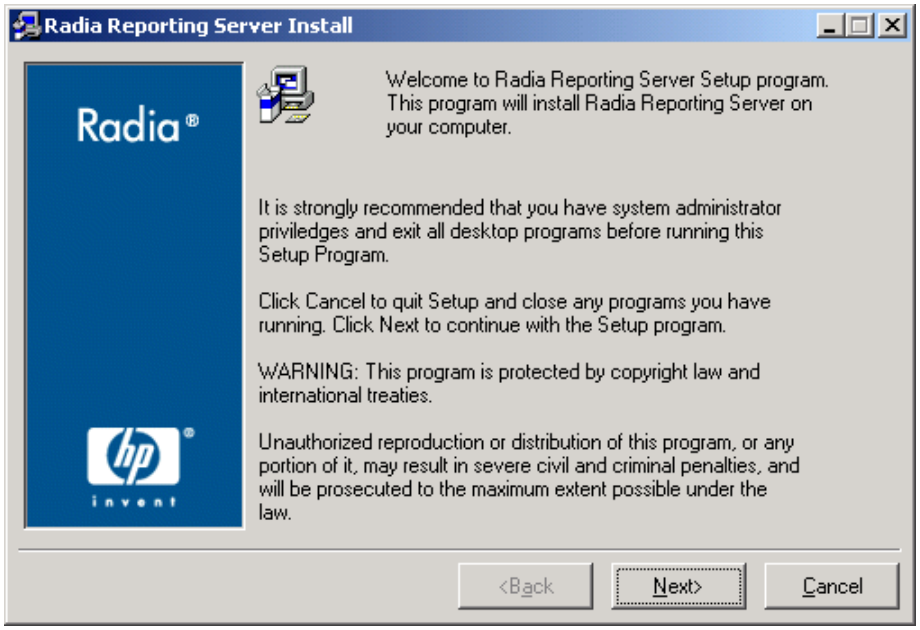
Installing the Reporting Server

Identify a Win32 computer with the Internet Information Services (IIS) component installed to act as your Reporting Web server. This computer must be able to communicate with your SQL Server where the Radia Databases are defined, the Internet, and your LDAP Directory, if desired.

The Reporting Server runs independently of the Management Portal. It may be installed on any Win32 machine whether or not that machine is running the Management Portal.

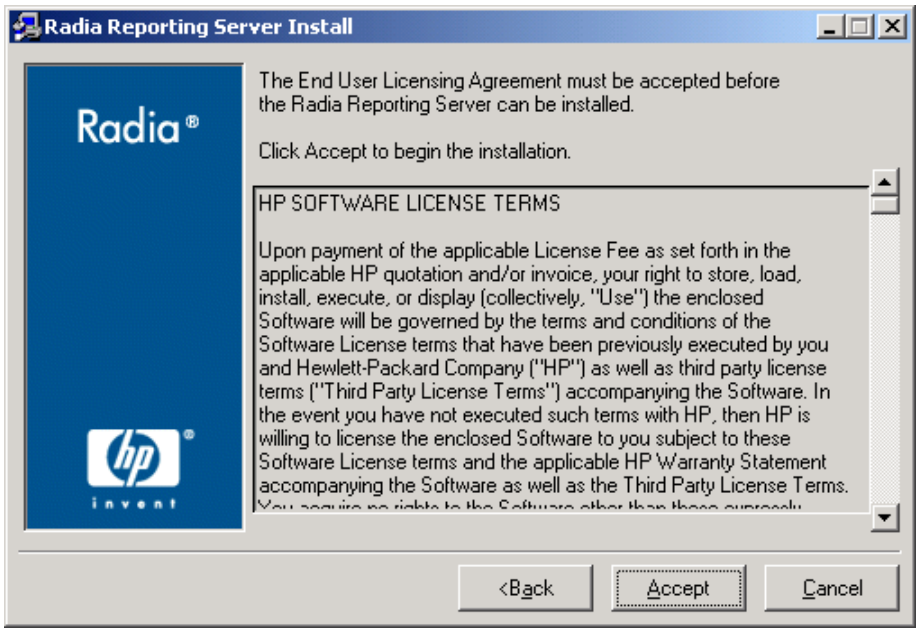
To install the Reporting Server

1 Double-click the Reporting Server installation executable, `setup.exe`. This file is located on your Radia infrastructure CD-ROM in the `\extended_infrastructure\reporting_server\win32\` directory. The Reporting Server Welcome window opens.



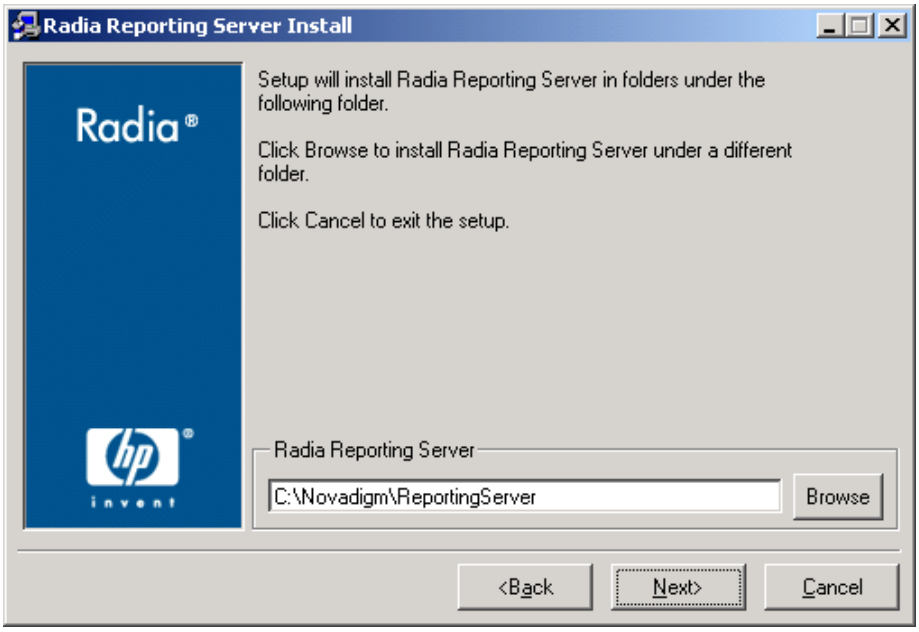
2 Click **Next**.

The HP Software License Terms window opens.



3 Read the end-user licensing agreement and click **Accept**.

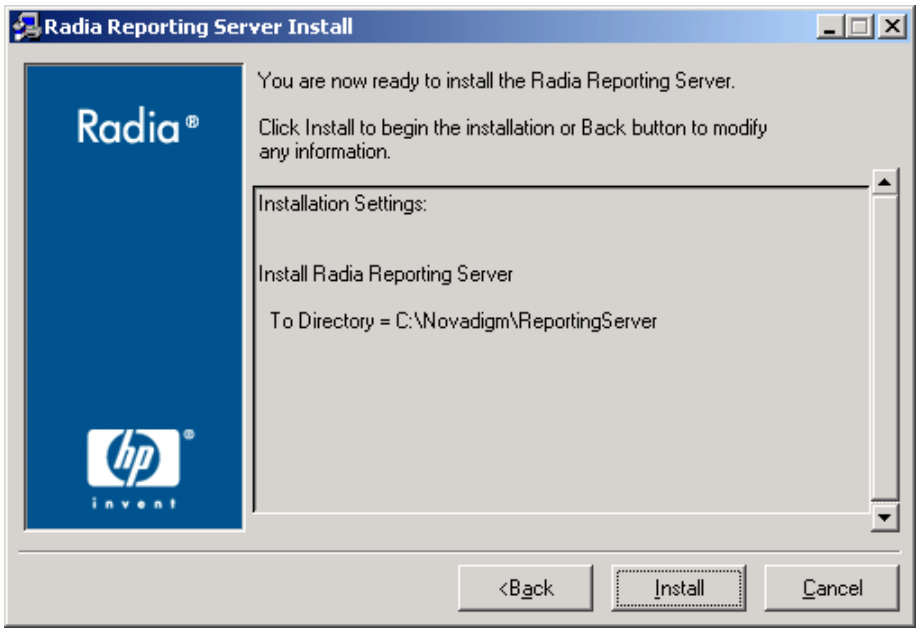
The Reporting Server installation location window opens.



4 Type a location to which to install the Reporting Server or click **Browse** to manually select a location.

5 Click **Next**.

The installation settings window opens.



6 Click **Install**.

The Reporting Server is installed.

The Reporting Server must now be configured for access to your SQL Server databases, and, optionally, an LDAP directory. Prior to completing this task, however, configure IIS for Radia Reporting.

Configuring Microsoft Internet Information Services for Radia Reporting

The Microsoft IIS Web server hosts the Reporting Server. Use these procedures to configure a `.tcl` extension in IIS for Radia Reporting, as well as create a Web share for easy access to the Reporting Server Web pages.

For additional details, refer to your Windows operating system documentation or help.

To configure IIS for the Reporting Server .tcl extension and web sharing

1 The Web Site for Reporting requires a `.tcl` extension configured in Internet Information Services to point to the `nvdkit.exe` executable.

For example, to add an application extension mapping in Windows XP, complete the following steps:

a Select **Start**, **Administrative Tools**, **Computer Management**.

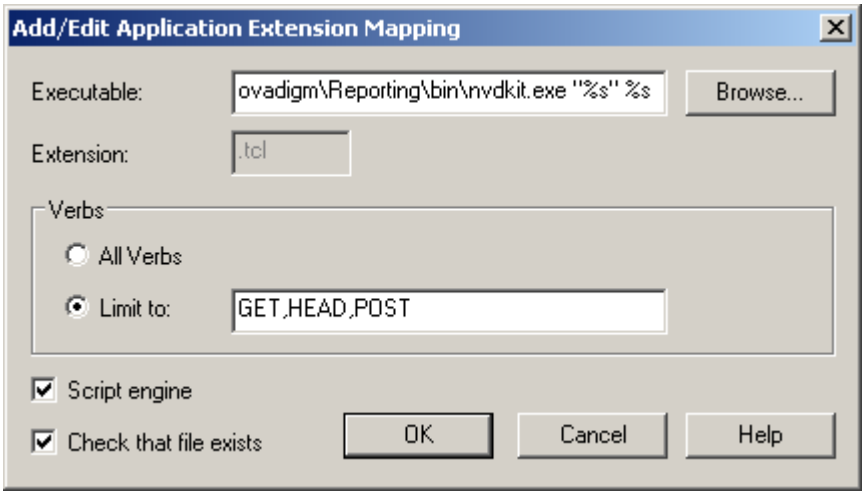
b Browse to **Services and Applications**, **Internet Information Services**, **Web Sites**, **Default Web Site**.

c Display the Default Web Site Properties page and click the **Home Directory** tab.

d Click the **Configuration** button to open the Application Configuration window.

e Click **Add** to add the Application Mapping for the `.tcl` extension. Complete the entries for the Add/Edit Application Extension Mapping dialog box as follows:

Executable	**`C:\novadigm\reportingServer\bin\nvdkit.exe "%s" %s`**
Extension	**`.tcl`**
Verbs	Limit to: GET,HEAD,POST
Script engine	Selected
Check that file exists	Selected



f Click **OK** when finished.

2 Enable web sharing from the root directory for Reporting, such as `C:\Novadigm\ReportingServer`, to the Internet Information Services web site. To do this:

   a Use Windows Explorer to browse to your Reporting folder, such as: `C:\Novadigm\ReportingServer`.

   b Right-click and select **Properties** from the shortcut menu. This displays the Reporting Properties dialog for your root Reporting folder.

   c Select the **Web Sharing** tab, and complete the entries to Share the folder and add an Alias for Reporting. You can use the defaults for Access and Application permissions.

   d Click **OK** to save the Reporting Alias for IIS web sharing.

The Internet Information Services component is now configured to support Radia Reporting for compatible Windows platforms with the exception of Windows Server 2003.

If you are using Windows Server 2003, an additional configuration step is required. See below.

Windows Server 2003 Additional Configuration for IIS

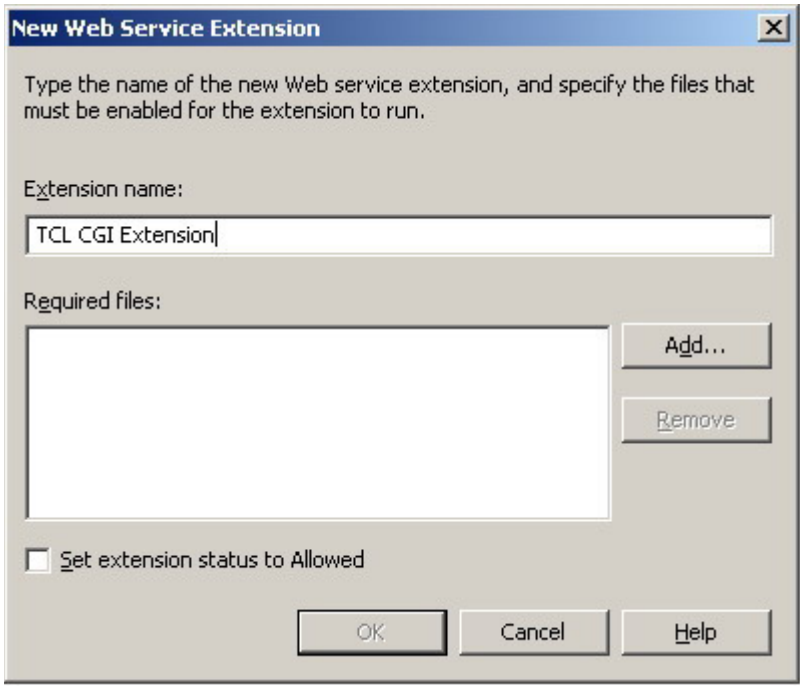
Some additional IIS configuration steps are required if you are using Windows Server 2003. Each step is configured within the IIS Manager.

First, create the TCL CGI Extension.

To create the TCL CGI Extension

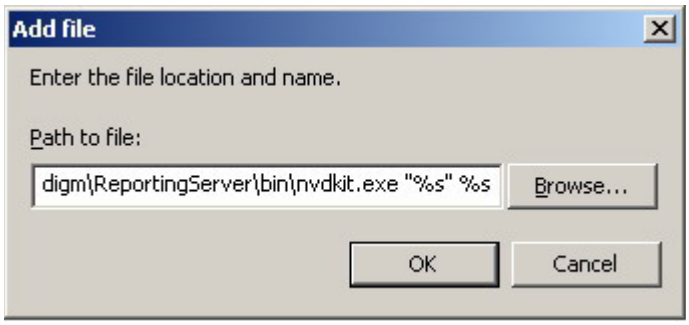
1 Within your IIS server, in the right-hand pane, click A**dd a new Web server extension.**

   The New Web Service Extension dialog window opens.



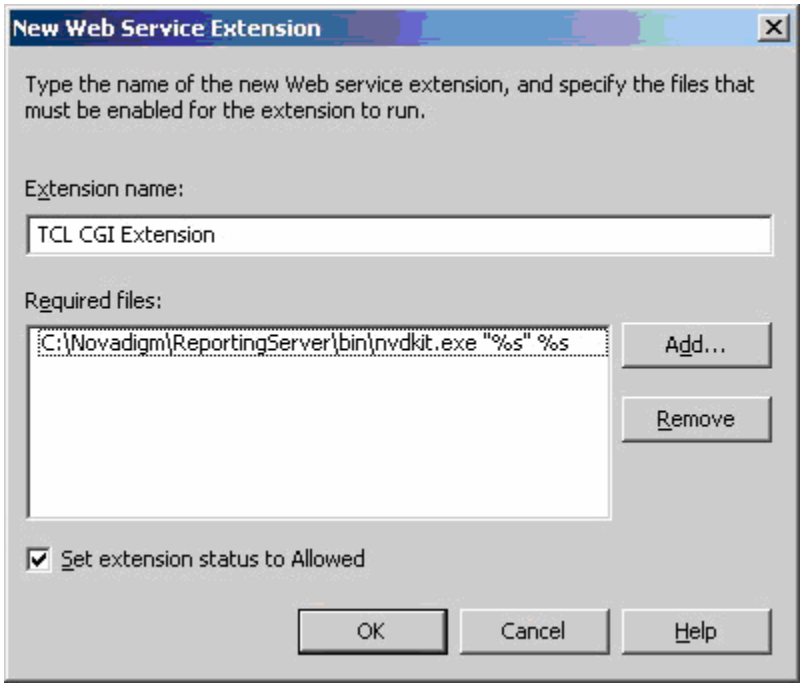
2 Enter the extension name and click **Add**.

The Add file dialog box opens.



3 Enter the path information for `nvdkit.exe` and click **OK**.

4 Select the **Set extension status to Allowed** check box.



5 Click **OK**.

The TCL CGI Extension is created.

When the TCL CGI Extension is created, add a new MIME type to allow IIS to serve the `.tcl` extension.

To add a new MIME type for Windows Server 2003

1 Right-click your IIS server and from the shortcut menu that opens, select **Properties**.

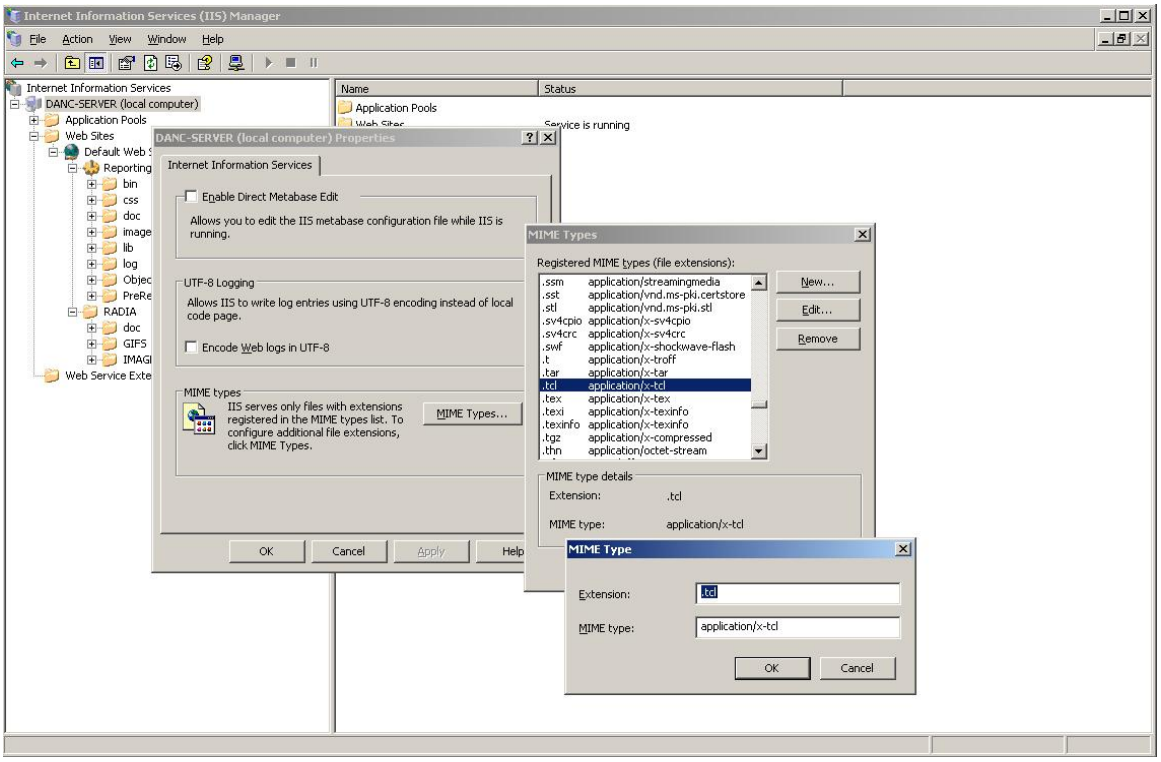
2 Click **MIME Types**.

3 Click **New** to add a new extension.

4 In the MIME Type window, add the following:

Extension **`.tcl`**
MIME Type **`application/x-tcl`**

5 Click **OK**.



The additional IIS configuration for Windows Server 2003 is complete.

Modifying the Reporting Server Configuration File

The Reporting Server configuration file, `config.tcl`, is located in the directory where you installed the Reporting component, such as `C:\HP\ReportingServer`. This file includes settings such as the path where you installed Reporting components, the ODBC DSN information and access credentials for each SQL database you are using, as well as LDAP Directory access root and credentials. Use the configuration file to enable or disable specific report types, including usage and patch and also to enable or disable individual features, such as caching.

You can edit this file manually using any text editor or you can edit this file using a web browser. We recommend that you make a backup copy of the config.tcl before you make any changes.

If you require encrypted passwords, use the web browser to edit the configuration file. Passwords are automatically encrypted when the configuration file is generated. Using a text editor to edit the file does not allow for creating encrypted passwords.

Using a Text Editor to Modify config.tcl

To set parameters for config.tcl using a text file

1 Open the `config.tcl` file with any text editor. It is located in the folder location where you installed Reporting, such as: `C:\HP\ReportingServer`.

2 Modify the values of the following entries near the top of the `config.tcl` file. Values must be enclosed in quotation marks and *all path values must use forward slashes*.

{HOMEPATH}	The exact path of your Reporting folder. Use forward slashes only. `set value "c:/hp/reportingserver"`
{DATABASE}	Requires SQL. `set value "sql"`

3 By default, the reporting `config.tcl` file is delivered with access to all of the following data or directories enabled: Radia Application Manager, Inventory Manager, and LDAP. If you are not accessing one of these databases, or an LDAP directory, change the appropriate {xxxENABLE} value from 1 to 0.

For example, to disable access to a Inventory Manager database, set the value for RIMENABLE to 0, as shown below:

```
{RIMENABLE} {
   # Enable Inventory Manager Support 1/0
   set value 0
   }
```

4 For each SQL database you are accessing (inventory, patch, or usage), set the values for any tablename PREFIX for synonyms, as well as the ODBC DSN, username and password required to access the database. All parameters for a given database follow the *ENABLE parameter which enables that data type. If necessary, contact your database administrator to obtain the required credentials.

For example, here are the parameters to fully configure access to the Inventory Manager database.

{RIMENABLE}	Enables access to Inventory Manager SQL database. `set value 1`
{RIMPREFIX}	#Tablename Prefix for Synonyms. Modify this value to the Tablename Prefix for synonyms used to access your RIM database. Enclose in quotation marks. `set value "dbo."`
{RIMDSN}	#ODBC DSN for Inventory Manager. Default is RIMDB. Set this value to the ODBC DSN for your RIM database. Enclose in quotation marks. `set value "RIMDB"`
{RIMDSN_USER}	# ODBC USER for Inventory Manager. Default is "sa". Set this value to the ODBC USER needed to access your RIM database. Enclose in quotations marks. `set value "sa"`
{RIMDSN_PASSWORD)	# ODBC PASSWORD for Inventory Manager. Default is null. Set this value to the ODBC PASSWORD for the ODBC USER for your Inventory Manager database. Enclose in quotation marks. `set value ""`

5 If you are accessing an existing LDAP directory, also set the values for the following parameters in `config.tcl`.

{LDAPENABLE}	Enables access to an LDAP directory. `set value 1`
{LDAPSERVER}	Enables access to an LDAP directory. Defaults to the local machine. Specify the IP address for a directory located on a remote machine. Enclose in quotation marks. `set value "127.0.0.1"`

{LDAPPORT}	# LDAP port number. Default port is 389. Enclose in quotation marks. `set value "389"`
{LDAPBASE}	# LDAP Base OU. The base organization unit to be mounted as the root of the LDAP directory. This becomes the highest level for filtering reports. Enclose in quotation marks. `set value "dc=nvddemo,dc=com"`
{LDAPUSER}	# LDAP User to Authenticate. Default is: has to be qualified such as: administrator@nvddemo.com value. Enclose in quotations marks. `set value ""` Note: value must be qualified, such as: administrator@hpreportdemo.com
{LDAPPASS)	# LDAP User Password. Default is null value. Enclose in quotation marks. `set value ""`

6 Save and exit the file. Use a web browser to update your `config.tcl` file. Once you click **Apply**, a new file, `config.new.tcl`, is created.

Using a Web Browser to Modify config.tcl

To modify the `config.tcl` file using a web browser, open the `setup.tcl` file located in your Reporting Server directory with any web browser. The `setup.tcl` file allows you to create a new file, `config.new.tcl`, which then must be renamed to `config.tcl` in order to apply any configuration changes.

Using the web interface to update your configuration file allows for encrypted passwords.

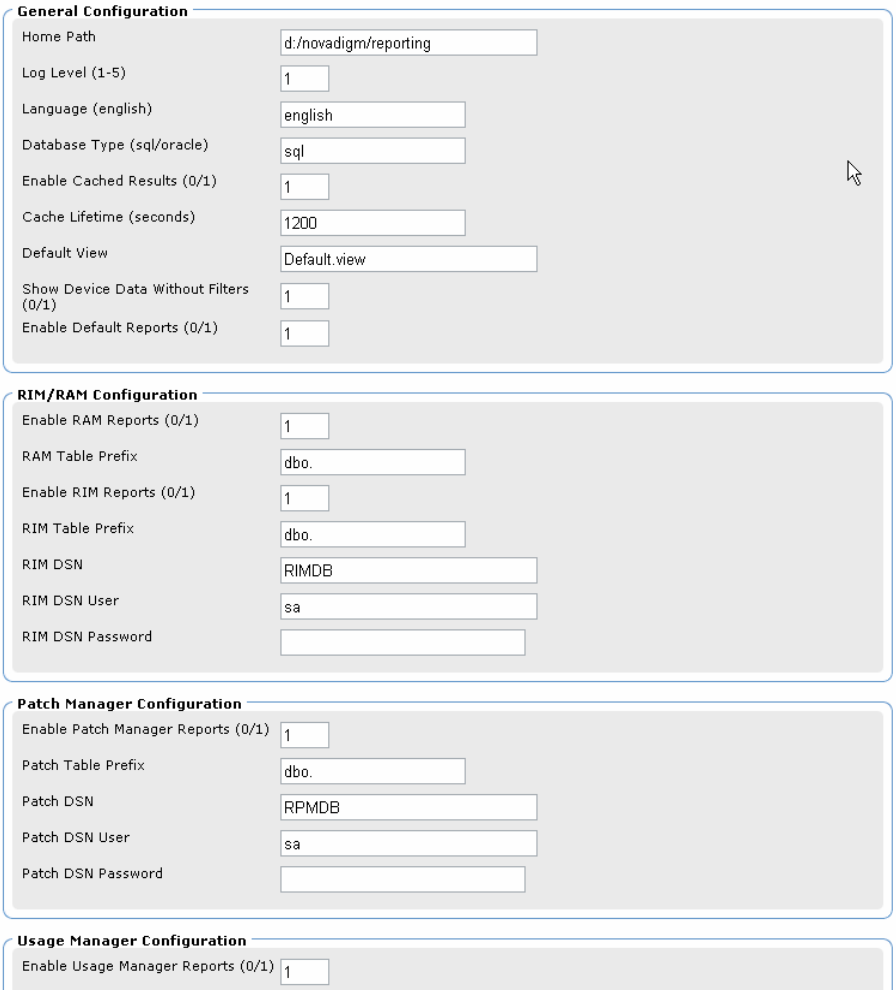
To set parameters for config.tcl using a web browser

1 Open a web browser and type:

`http://localhost/`*`reporting`*`/setup.tcl`

where *`reporting`* is the Alias specified in Step 2 c of *Configuring Microsoft Internet Information Services for Radia Reporting* on page 30.

The configuration file page opens.



2 To modify each configuration file section, follow the directions in Using a Text Editor to Modify config.tcl on page 36. The similarly named text boxes in the web interface represent the `config.tcl` file parameters.

3 After you are finished making modifications, click **Apply**.

4 A new file is created, `config.new.tcl`, and stored in your Reporting Server directory. In order to apply any changes, first backup your existing `config.tcl` file and rename `config.new.tcl` to `config.tcl`.

The Reporting Server is now fully configured.

The Reporting Server is set up to allow for Radia Notify by default. If you will not be using the Reporting Server to notify devices, you can turn off the notify-related options and icons by setting the {NOTIFYENABLE} value to 0.

Notify requires the Management Portal.

Configuring the Reporting Server for Notify

By default, the Reporting Server is configured for Radia Notify. Since the Management Portal is required to complete the notify process, the `config.tcl` file must be configured with your Management Portal settings.

The Management Portal version 2.0 or higher is required for Notify.

Before you define your Management Portal settings, ensure that the notify function is enabled by checking the `config.tcl` file, parameter value {NOTIFYENABLE} is set to value 1 if you are using a text editor. Alternatively, if you are using a web browser, make sure the Management Portal Configuration section parameter Notify Enabler, has a value of 1.

To configure a Management Portal for Notify

1. Open the `config.new.tcl` file with a text editor or use the `setup.tcl` web interface to create a new `config.new.tcl` file.
2. To define your Management Portal settings, modify the values of the following parameters.

   If using a text editor: (Values must be enclosed in quotation marks and all path values must use forward slashes.)

   ```
   {RMPIP}    #TCP/IP Address or Host Name for Management Portal
              set value "127.0.0.1"

   {RMPPORT} #TCP/IP Port for Management Portal
              set value "3466"
   ```

   If using a web browser:

   RMP Address `127.0.0.1`

   RMP Port `3466`
3. Enter your Management Portal IP address and port number.

4 Save and close the `config.tcl` file, or click **Apply** if using the web browser interface.

If you used a web browser to enable Notify, make sure to rename the newly created `config.new.tcl` file to `config.tcl`.

The Reporting Server has been configured for Notify.

Enabling the Reporting Server Cache Feature

Similar to an HTTP proxy, the Reporting Server has the ability to save report data in a cache file. When a user requests a report, that report data is saved in a file on the Reporting Server. Then, if any subsequent users request the same reports, the data is readily available in the cache, returning the report much faster by avoiding the processing time used to retrieve the data from the Oracle or SQL database. The cache file is saved in a folder within the Reporting Server installation directory, such as `C:\HP\ReportingServer\cache`.

Use the configuration file to enable caching and to determine how long a cache is available.

To configure caching

1 Open the `config.new.tcl` file with a text editor or use the `setup.tcl` web interface to create a new `config.new.tcl` file.

2 To enable and define your cache settings, modify the values of the following General Configuration section parameters.

If using a text editor: (Values must be enclosed in quotation marks and all path values must use forward slashes.)

```
{CACHEENABLE} {
          # 0 Disabled, 1 enabled
                set value "1"
          }
{CACHELIFE}   {
                # Cache Lifetime in seconds
                set value "1200"
 }
```

If using a web browser:

```
Enable Cached Results (0/1)   0
```

```
Cache Lifetime (seconds)      1200
```

3. Caching is disabled by default. Enable caching by setting CACHEENABLE or Enable Cached Results (0/1) to 1.
4. Define how long in seconds the cache will be available using CACHELIFE or Cache Lifetime (seconds).
5. Save and close the `config.tcl` file, or click **Apply** if using the web browser interface.

   Report caching has been enabled and configured.

Accessing the Reporting Server Web Site

After setting values in the `config.tcl` file, you are ready to access the Reporting Server web page.

To access the Reporting Server locally

- From the Win32 machine running IIS, open a Web browser and type:

  `http://localhost/reporting`

  Where *reporting* is the Alias specified in Step 2 c of *Configuring Microsoft Internet Information Services for Radia Reporting* on page 30.

  The Management Portal banner page opens to the Reporting Server home page, as shown in Figure 3 below.

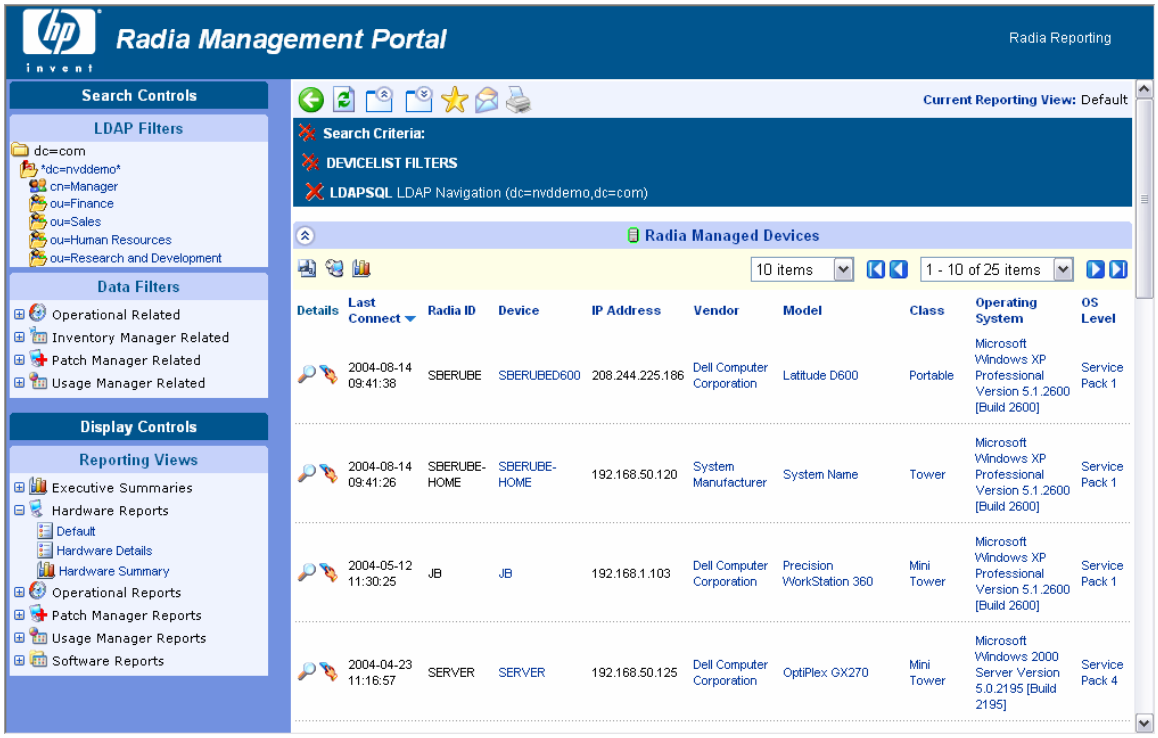


Figure 3: Sample Radia Reporting home page with LDAP access enabled.

Chapter 4, Customizing Reports explains how to use the Reporting Server interface, and Appendix A, Sample Reporting Scenario illustrates a sample reporting session in detail.

Adding Components to Radia Inventory Audits

To obtain the reports available through the Reporting Server features, it is highly recommended you audit your Radia Clients for the Win32 components previously listed in Table 1 on page 25. If you're missing some of the components, you can use these procedures to add them to your Inventory Manager Reporting audit package.

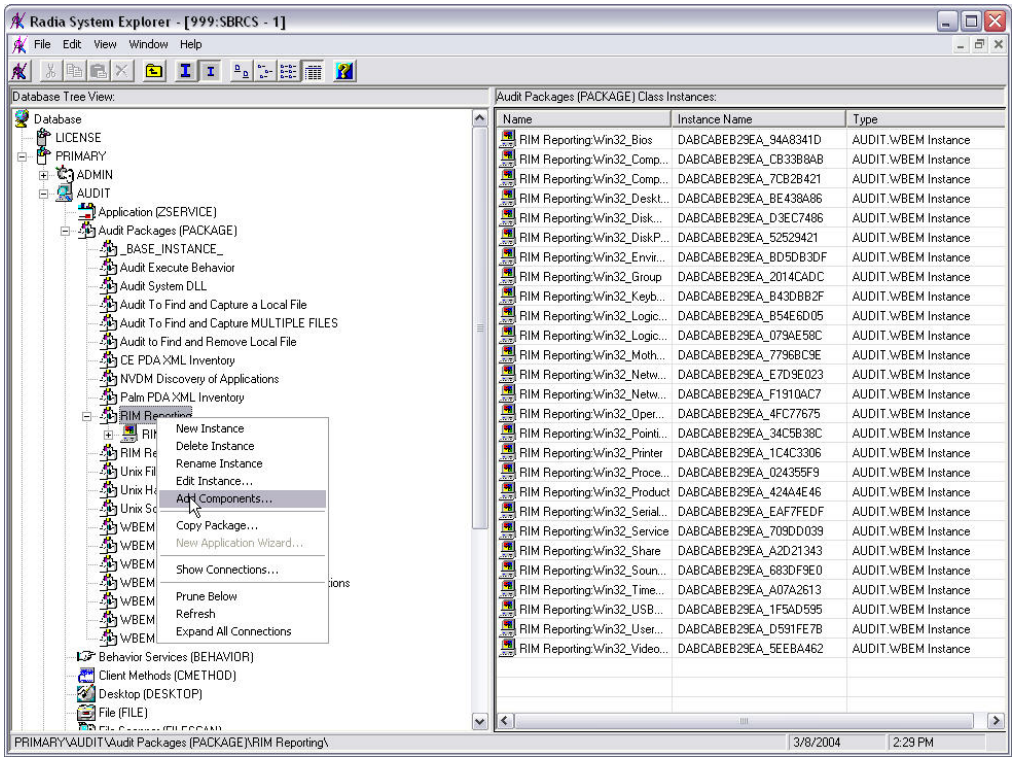
For additional information, refer to the *Installation and Configuration Guide for the HP OpenView Inventory Manager Using Radia (Inventory Manager Guide)*.

The following example adds the Win32_MemoryDevice component to the Inventory Manager Reporting Package. Use the same procedure to add any component to the Inventory Manager Reporting Package or to another audit package that is used to collect Inventory Manager data at your site.

Before adding any components, make sure they do not already exist within the reporting audit package. Adding duplicate components will cause errors.

To add components to Inventory Manager reporting audit packages

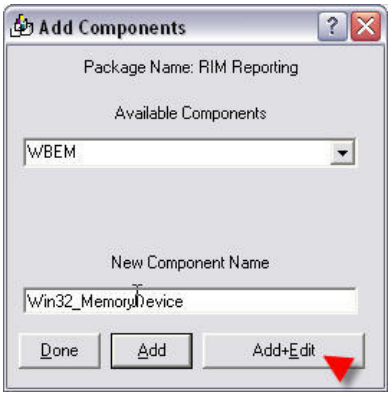
1. Use the System Explorer for the HP OpenView Administrator Workstation Using Radia (System Explorer) and browse to the PRIMARY.AUDIT.PACKAGE class.
2. Locate the Inventory Manager Reporting package and expand all connections to show all component instances.
3. Right-click on the Inventory Manager Reporting Package instance, and select **Add Components** from the shortcut menu.



4 The Add Components dialog box opens. Select **WBEM** from the **Available Components** drop-down list, and type the component name in the **New Component Name** text box. For this example, we will type:

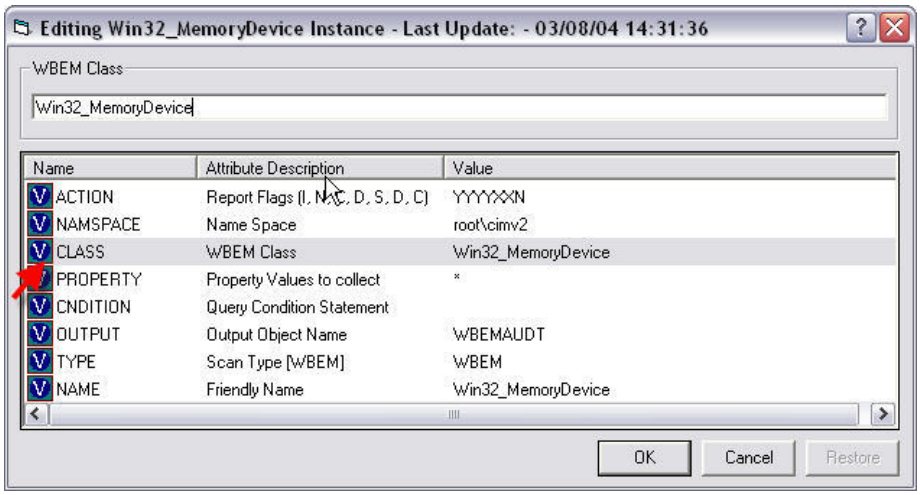
`Win32_MemoryDevice`.

5 Click **Add+Edit**.

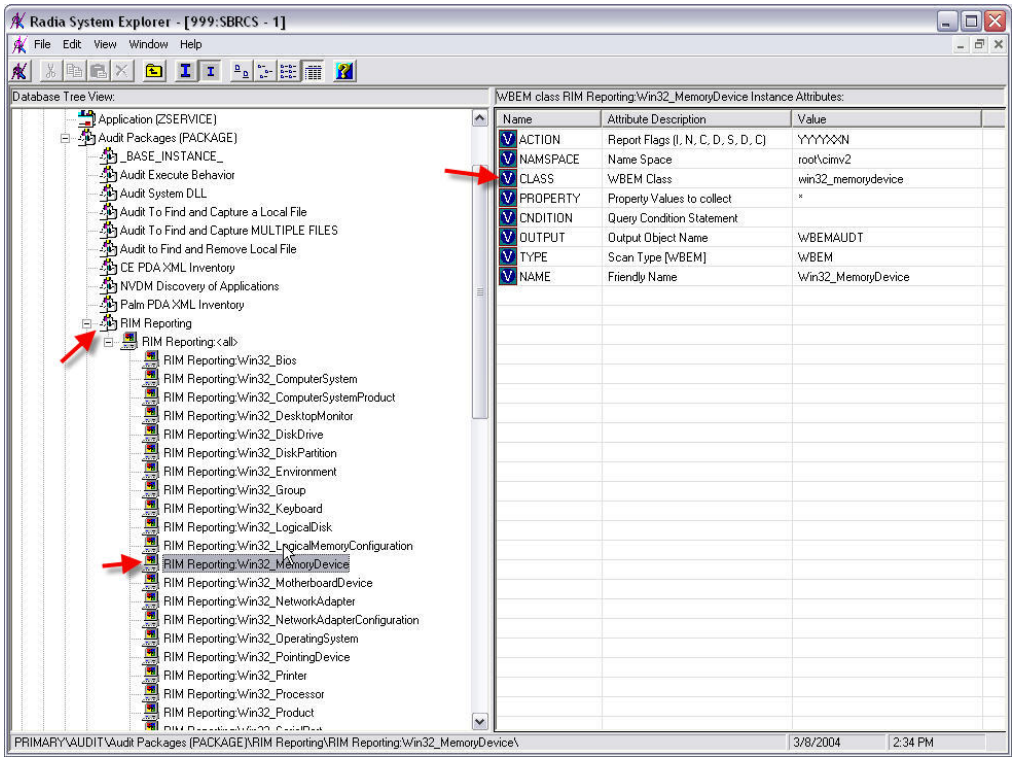


6 The Edit Instance dialog box opens. Click the **CLASS** entry, and type the value of the WBEM class. Use the same name as the component name, e.g., **`Win32_MemoryDevice`**.

7 Click **OK** to save your changes.



8 As shown in the next image, the Win32_MemoryDevice instance is added to the Inventory Manager Reporting Package.



9 Repeat this Add Component procedure for each entry in Table 1 on page 25 that is missing from your inventory auditing package.

When auditing for Win32_UserAccount or Win32_Group, large amounts of data may be returned. Failure to limit the scan may result in **high network traffic**. In order to limit the amount of data returned by these queries, modify the class.

In order to restrict the results to LOCAL user accounts and LOCAL groups only, modify the CNDITION field of the Win32_UserAccount and Win32_Group classes by adding the following syntax:

```
CNDITION      Domain = "&(zconfig.zhdwcomp)"
```

Be sure to check the HP Openview support web site for the most recent information on this topic.

Summary

- The Reporting Server must be installed onto a Win32 machine with the Microsoft Internet Information Services component.
- Create or locate Radia SQL databases so they are defined on a single SQL Server for access by the Reporting Server.
- Modify your Configuration Server and Inventory Manger Server to support Radia Reporting.
- Modify the SQL database definitions for Inventory Manager, Patch Manager, and Usage Manager to support Radia Reporting.
- If necessary, modify your Inventory Manager audits to include the set of Win32 components needed to take advantage of Radia Reporting.
- Install the Radia Reporting folders and files on the Win32 Web server machine.
- Configure IIS to point to NVDKIT for `.tcl` extensions, and enable an IIS web share to your Reporting folder.
- Modify the `config.tcl` file to point to all DSNs and any LDAP directory you are accessing for Reporting.

3 Reporting Server Features

At the end of this chapter, you will:

- Be familiar with the Reporting Server user interface.
- Know how to use the features of the Reporting Server.
- Know how to customize the Reporting Server interface.

Accessing the Reporting Server

The Reporting Server runs as an independent application hosted by a Microsoft Internet Information Services (ISS) Web service. Once installed and configured, users have access to the reports from any Web browser connected to the Internet.

To access the Reporting Server

- Open any Web browser and type the following address:

  `http://<hostname>/reporting`

  Where *<hostname>* is the host name for the IIS web server on which the Reporting Server was installed and where *reporting* is the Alias assigned to Radia Reporting during installation and configuration.

  Reporting is optimized for display screen area setting 1024 x 768 or greater.

About the Reporting Server Interface

The Reporting Server user interface contains several distinct areas, as shown in Figure 4 on page 52.

- The **Management Portal banner** runs across the top of the page. The current version of the Reporting Server and its components are available by using the help icon.
- **Search Controls**. Use the LDAP Filter or Data Filter area to apply one or more filters to the dataset being accessed by the Reporting Server for the current View. Any filters you apply are listed as Search Criteria above the above the reports.
  - **LDAP Filters**. The LDAP Filters are available if Radia Reporting was configured to access your LDAP Directory. Click on an LDAP entry to filter the current dataset to that level. The LDAP area is discussed on page 53.
  - **Data Filters**. Use this area to generate or select a filter to be applied to the current dataset. See Using Search Controls to Select Filters on page 53 for details on how to use this area.

- **Display Controls**. Use the Reporting Views area to control your current session and display.

  — **Reporting Views**. A Reporting View defines the set of reporting windows to display for the current dataset and initial settings related to each window (such as minimized or maximized, and the number of items per window). When you first access the Reporting Server, the Default View is applied. The current view is listed on the right of the Global Toolbar.

  Use the Reporting Views area to change or customize your Reporting View. For details, see Using Display Controls to Select Reporting Views on page 56.

- The **Search Criteria** above the report windows list the filters that have been applied to the dataset using one of the Search Controls.

  To remove a filter, click the ***X*** to the left of a filter name.

- **Report Windows** display the current View.

  Click minimize  on the Window title bar to collapse a report window.

  Click maximize  on the Window title bar to expand a report window.

  See About Reporting Windows on page 58 for details about using the Report Window Action Bar icons, as well as browsing, sorting, and viewing details for the items in a report.

- Each window contains an **Action Bar** that includes icons allowing you to create CSV files from current datasets, switch to graphical views, or to notify devices.

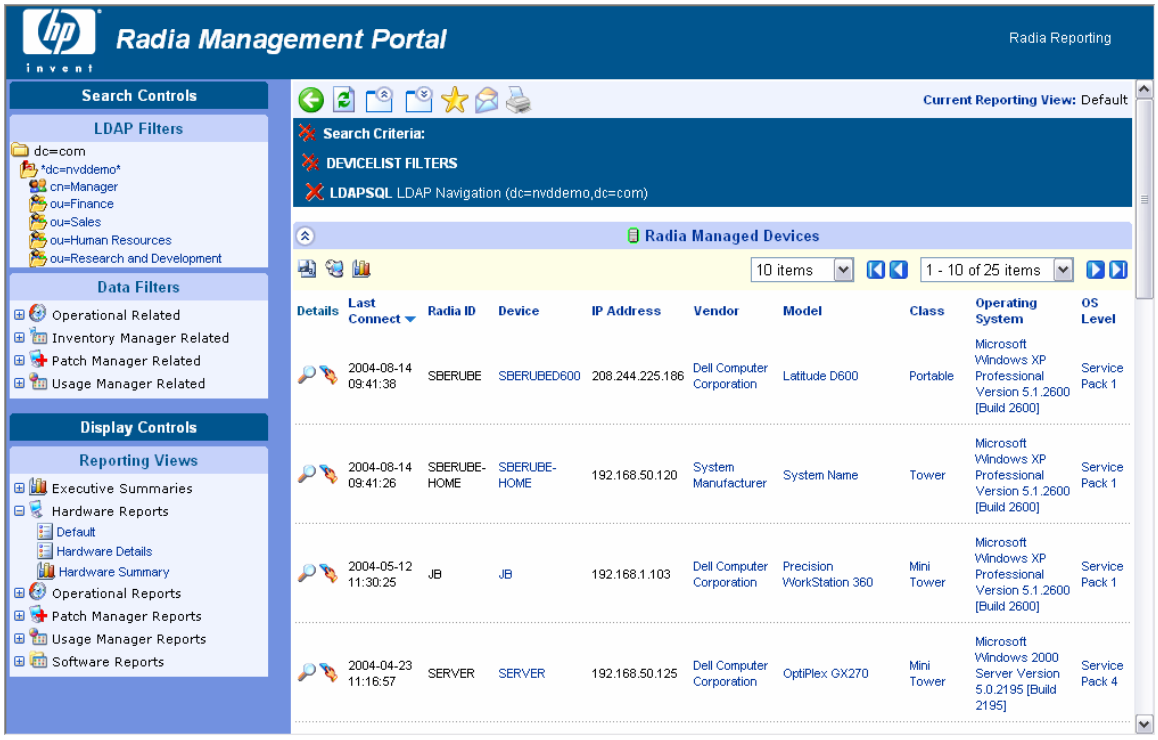


Figure 4: Reporting Server user interface

About the Banner Area

The banner area displays across the top of the Reporting Server Web page and contains descriptive information.

Figure 5: Banner area.

- To display the current version number of the Reporting Server and its components, mouse over the help icon in the upper right-hand corner of the banner section.

Using Search Controls to Select Filters

The Search Control areas give you two ways to filter datasets within the Reporting Server. You can:

- Select an LDAP directory entry from the LDAP Filter area. This limits the results to the LDAP entry level.
- Use the Data Filter area to create or apply a filter. This limits the results to the specific filter you applied.

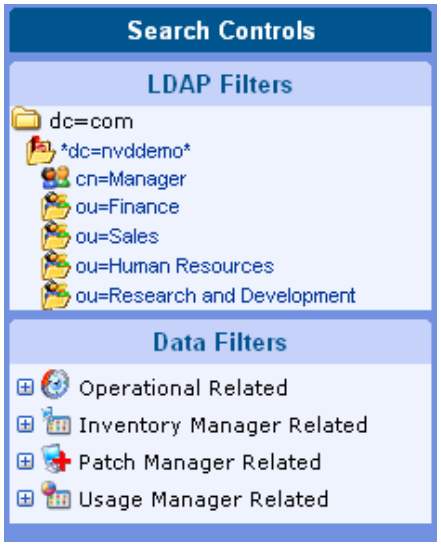


Figure 6: Search Controls area.

When you select an LDAP Filter or apply a Data Filter, your filter is automatically listed as a Search Criteria entry.

The LDAP Filters Area

The **LDAP Filter** appears as a Search Control if the Reporting Server is configured to access an LDAP directory.

Use the LDAP Filter to browse to an entry in your directory. As you click a directory entry, the Reporting Server automatically filters the reporting data displayed for that entry. For example, if you click the **Sales** department entry,

the reporting area limits the display to only the devices that are associated with the Sales department.

Navigating Within the LDAP Filters Area

Clicking any image within the LDAP Filters area allows you to drill down further into the LDAP tree. Clicking any text will apply the associated filter to your data.

Figure 7: LDAP Filters Area.

Once you expand the tree view in the LDAP Filters area, the expanded branch becomes the root branch.

The Data Filters Area

The **Data Filters Area** is always available as a Search Control (along the left side of the Reporting Server page). Use it to select a filter to apply to the current dataset. Once a filter is applied, you will see it added to the Search Criteria list above the report windows.

To select and apply a filter using the Data Filter area

1. From the Data Filter area, use the Filter Group tree-view and select a group. The example shown in Figure 8 on page 55, selected **Hardware Related Filters**.
2. Open the Filter tree-view and select a filter. The example shown in Figure 8 on page 55, selected **Device Classification**.
3. In the **Filter Value** text box, type a specific value. For example, **`*Notebook*`**. You can use wildcards, including * for multiple characters, or ? or _ (underscore) for single characters.

4 Click **Apply** to add this filter to the report. After applying the filter, you will see it added to the Search Criteria list above the report windows.

The **Reset** button clears the Filter Value field and resets the Filter Group and Filter selections to their default values.

Figure 8 below displays an example of the Data Filter entries used to limit the report to only Notebook devices.

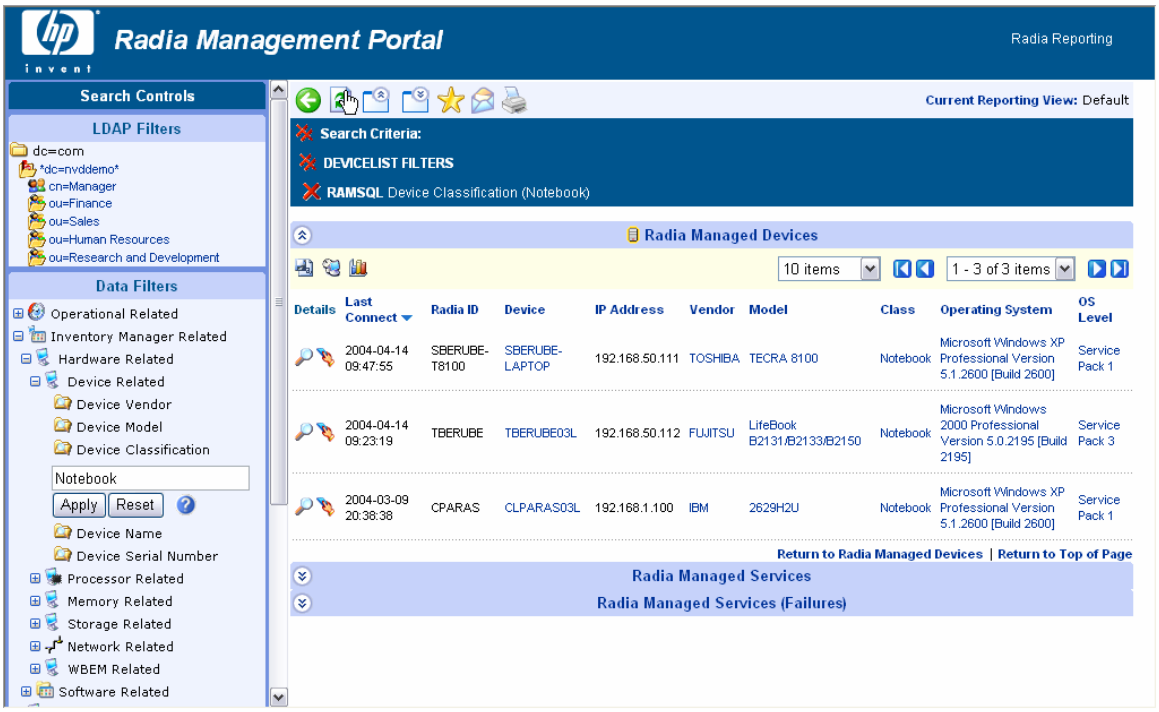


Figure 8: Applying a Search Criterion to limit report to Notebook Devices.

Special Filter Value Characters and Wildcards

Finding the right records can be made easier by using special characters and wildcards within your search strings. Use these special characters in conjunction with the text you enter into the **Filter Value** text box. Table 2 on page 56 explains each special character.

Table 2: Special Characters and Wildcards

Character	Description
* or %	Return all records of specific text string. Example: Device Vendor Filter HP* returns all HP records. %HP% returns all records including HP.
? or _	Return any single character Example: Device Classification Filter Not?book returns all records beginning with 'Not' and ending with 'book'. Note_ook returns all records beginning with 'Note' and ending with 'ook'.
!	Negates filter. The ! must be placed before the text string. Example: Device Vendor Filter !HP* will return all non-HP records.

Using Display Controls to Select Reporting Views

Within the **Display Controls** area, **Reporting Views** specify which windows are to be displayed on the report page, as well as their initial state (maximized or minimized).

View Groups and Views are stored as objects.

Figure 9: Display Controls area.

Applying a View from the Reporting View Area

To apply a View

1. From the **Reporting View** area, open the **View Group** drop-down list and select a group. The example shown in Figure 10 on page 58, has **Software Reports** views selected.
2. Next, select a view for that group. The example shown in Figure 10 on page 58, has **Service Details** selected.
3. Click **Apply** to apply this View to the dataset. After applying the view, you will see the appropriate report windows displayed for the selected View.

The **Reset** button resets the View Group and view selections to their default values.

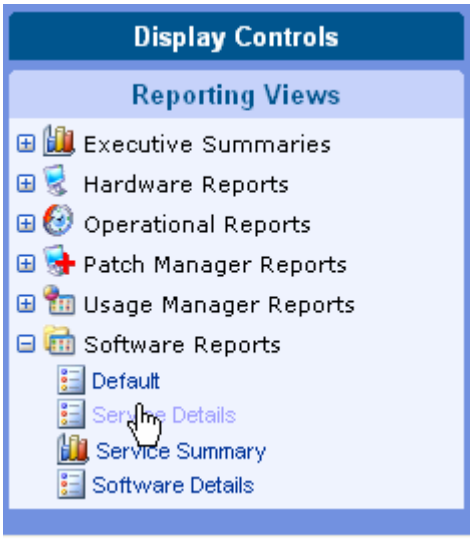


Figure 10: Sample Selections for Software Reports and associated Reports.

Use the back button  to return to any of the previous reporting windows. When you reach the top of the history, the back button disappears.

About Reporting Windows

The Report Page displays the Windows specified in the applied view. Figure 11 on page 59 shows an example of three Windows displayed on the Report Page. The **Radia Managed Devices** and **Radia Managed Services** windows are maximized, while the **Application Usage** window is minimized.

Using the Windows Action Bar Icons

Each Window contains an **Action Bar** with the following possible icons:

 **Notify Devices**

From the Radia Managed Devices window, click the Notify icon to notify all devices in the window. The Management Portal is required for notifying. The Reporting Server sends the notify request to the Management Portal.

 **Switch to Graphical View**

Click this icon to switch to a graphical view of the data.

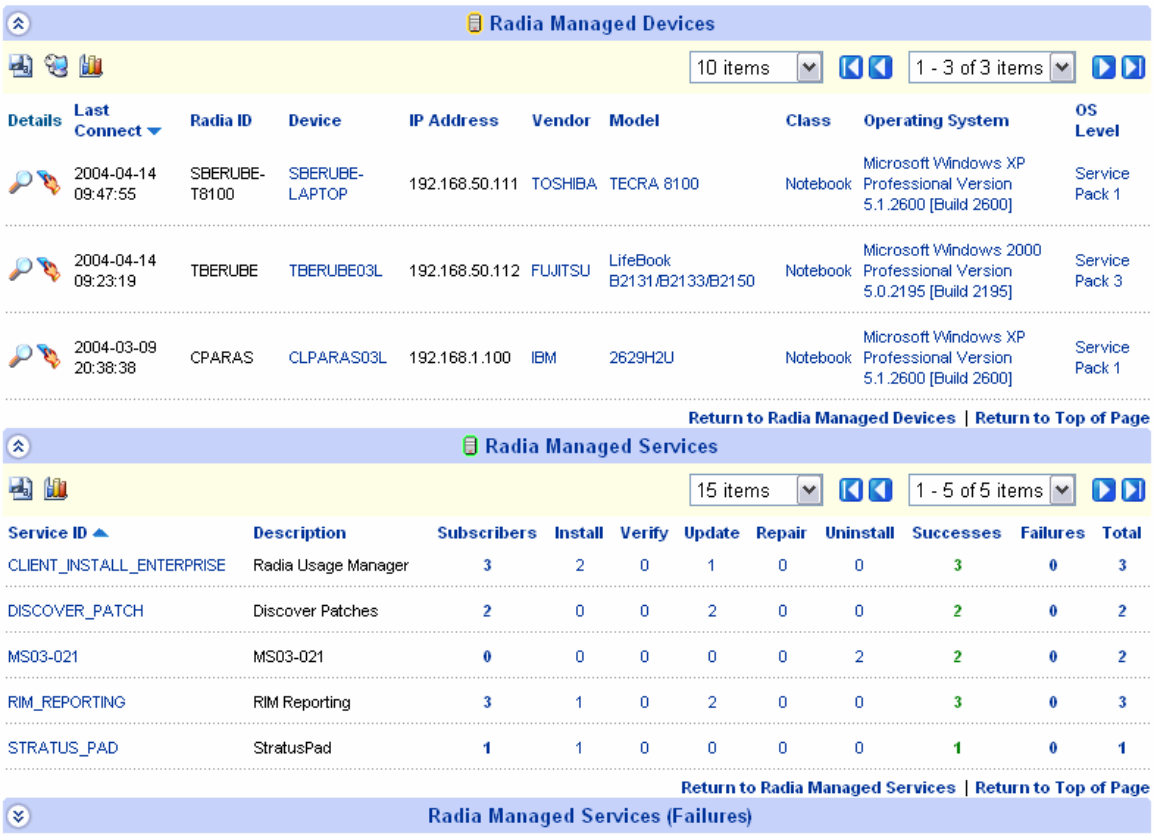


Figure 11: Sample Device and Services on a Reporting Page.

Browsing Items in a Report

There may be very large numbers of items in any report. The Action Bar lets you customize how many items to view in a given window area. To browse to records outside your current window area, use the Browse buttons or drop-down list, as illustrated in Figure 12 on page 60.

Maximum items per window.
Use this drop-down list box to limit how many items to display in the current window. For example, if you select a maximum of 30 items, you will be able to scroll 30 items in the current window.

Browse Back and Forward Buttons.
If you set the maximum items per window smaller than the total items in the report, you will have the ability to browse through multiple windows. Use the

browse buttons to go to the First, Previous, Next, or Last window for the current report.

Browse to a specific window.
Alternatively, select which set of items to view from the list of available windows. For example, select **1 - 15 of 46** items from drop-down list box to view that set of items.

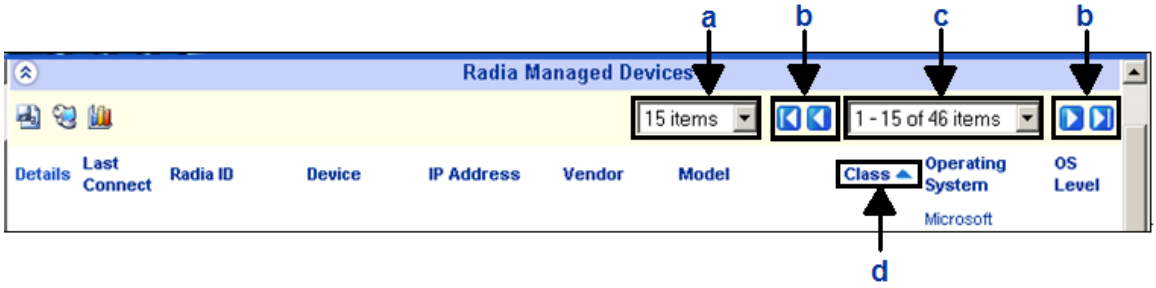


Figure 12: Report Display Settings: 15 Items per Window, Sort by Class in ascending order.

a Maximum items per window

b Browse buttons

c Current display and total

d Current sort field and order

Sorting Columns

Sort items in any report by any column either in ascending or descending order by clicking the column-heading name.

Clicking a column-heading name selects the column for the sort and displays the items in ascending order. An up arrow indicates the active sort column and ascending order.

To toggle between ascending and descending sorts, click a currently selected sort field. A down arrow indicates the items are displayed in descending order.

For example, Figure 12 on page 60 shows a report sorted on the **Class** column in ascending order. Notice the up-arrow to the right of the **Class** column heading.

Notifying Devices

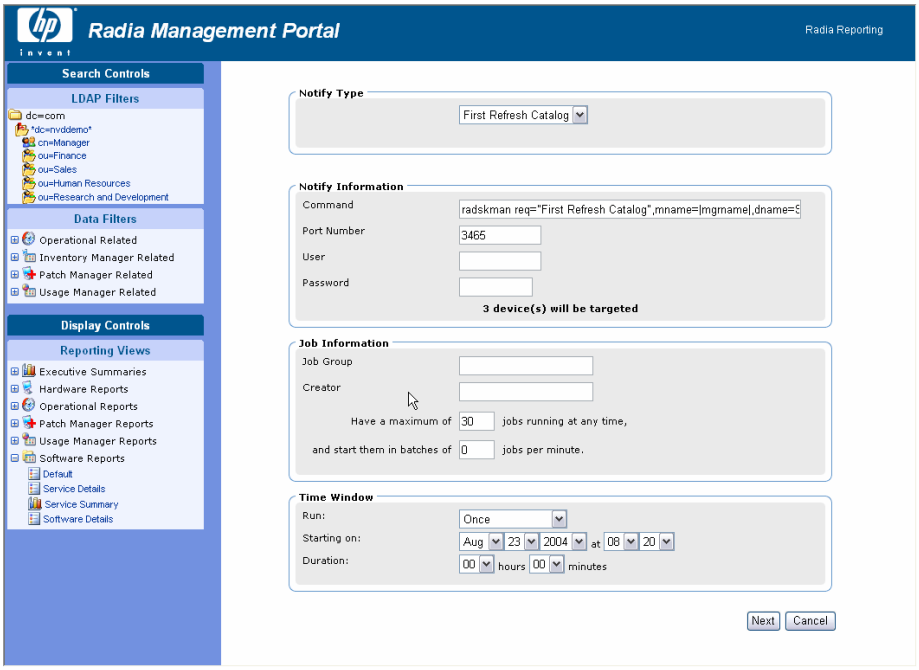
The Management Portal is required in order to use the Notify function.

If you would like to notify the devices displayed in the **Radia Managed Devices** window, click the **Notify** icon located in the action bar.

To Notify devices

1 After you've selected which devices you would like to notify, click **Notify**.

The Notify window opens.



2 Define your notify settings and click **Next**.

A request is sent to the Management Portal and the notify process is started. To view the status of the notify job, use the Management Portal.

For more information about using Radia Notify and the Management Portal, refer to the *Installation and Configuration Guide for the HP OpenView Application Manager Using Radia (Application Manager Guide)* and the *Installation and Configuration Guide for the Management Portal Using Radia (Management Portal Guide),* respectively.

Switching to a Graphical View of Reporting Data

Click **Switch to Graphical View** in the action bar area of any report window to switch to a graphical view of the report data. Figure 13 below shows a sample graphical view.

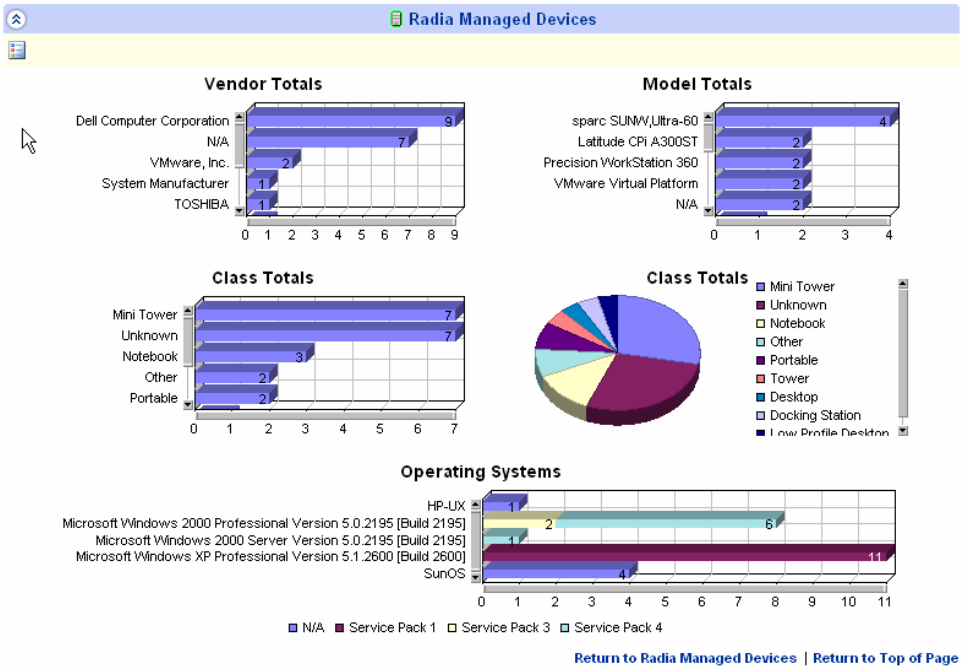


Figure 13: Sample Report in Graphical View.

Double-click any individual graph object to add a new Search Criteria. A new set of graphs is displayed based on the specific information you selected.

To return to the detailed view at any time, click **Switch to Detailed View** in the action bar.

Displaying Device Details

From the Radia Managed Devices report window, click **Show Details** next to any item to display the details for that device.

The Device Summary window opens, as shown in the following figure. Notice that in addition to the standard global icons, the green arrow icon allows you to return to the previous window.

Figure 14: Click Show Details to access the Device Summary Window.

Click any heading at the bottom of the page to expand its listing. For example, if you click **Services**, you'll see the list of Windows Services installed on the system.

The **Device Summary** contents will vary according to the starting Report Window. For example, the following figure displays the **Device Summary** for another device accessed from a Patch Manager Devices window.

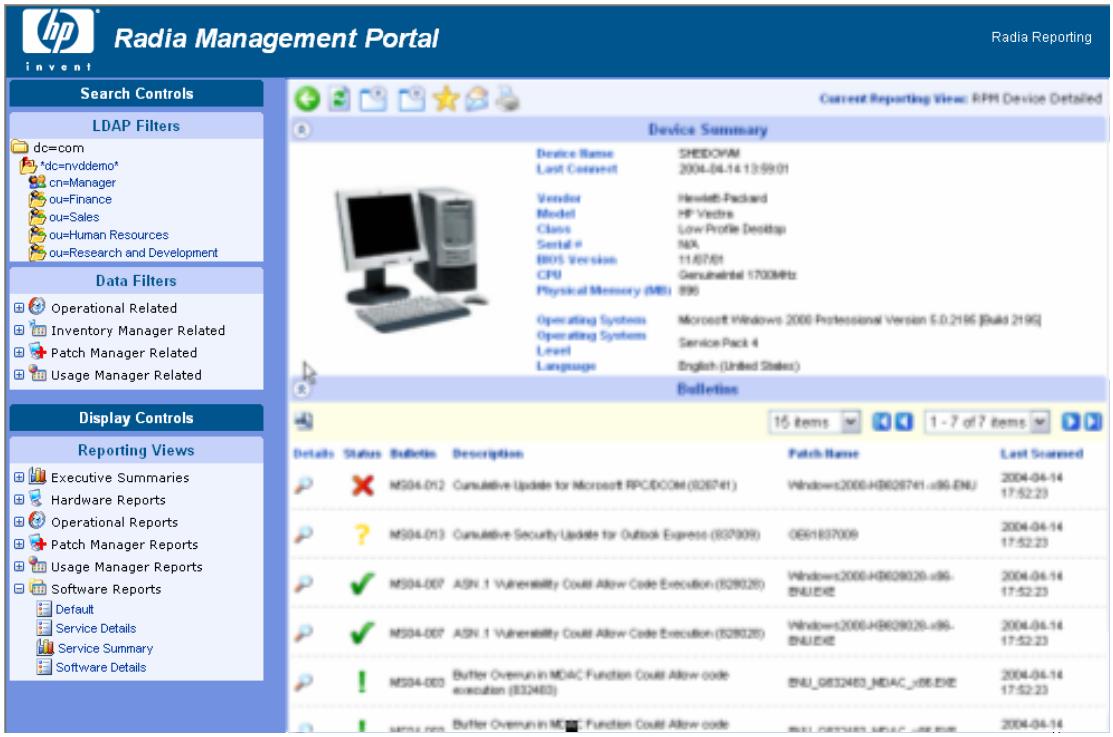


Figure 15: Sample Device Summary from Patch Manager Devices window.

Continue to view details of the Bulletin for the first item, MS04-012. Click **Show Details** in the **Details** column and the Microsoft page listing this patch is opened, as shown in the following figure.

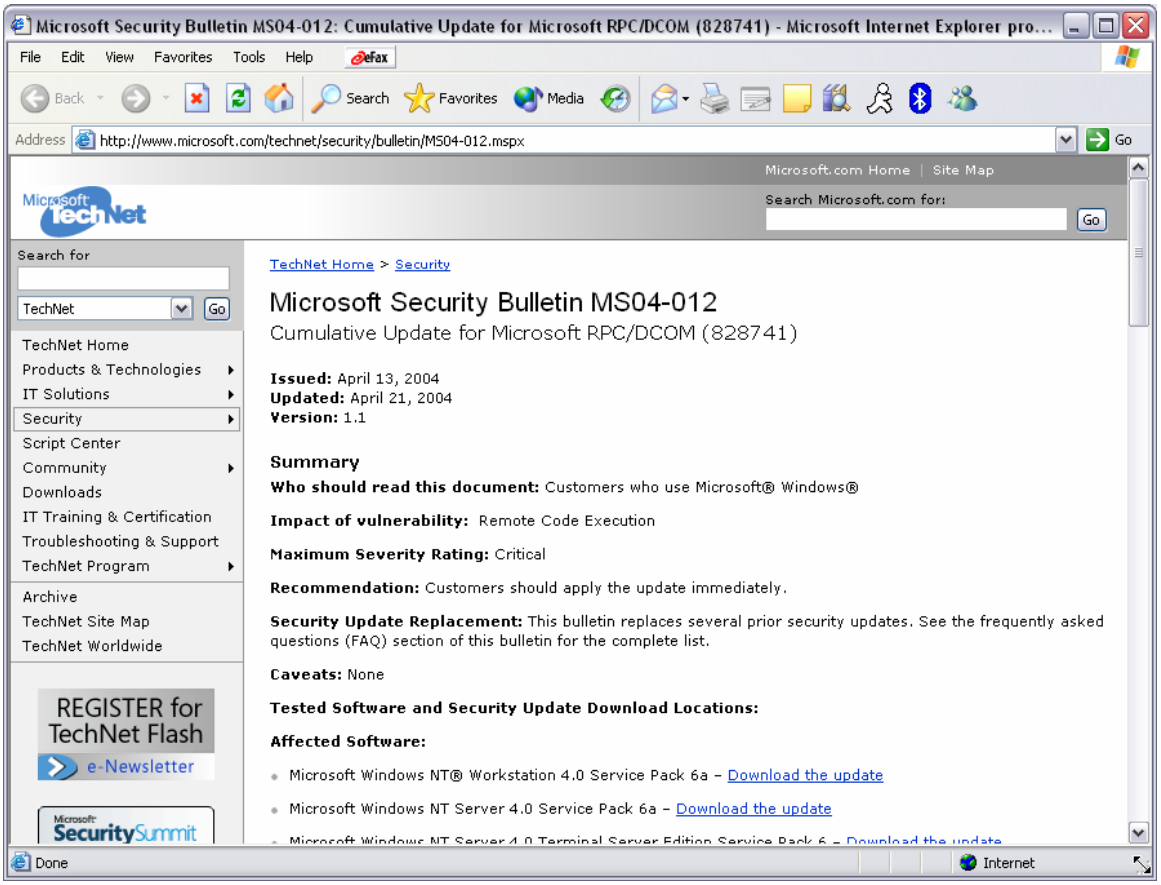


Figure 16: Details for a Bulletin Item links to the Microsoft Security page for it.

Using Remote Control (VNC)

If you'd like to begin a VNC session for an individual device, click **Remote Control (VNC)**  next to the appropriate device row.

Summary

- The Reporting Server user interface contains several distinct areas. Use these to generate reports based on criteria you supply.
- Use the Search Controls to apply one or more filters.
- Reporting Views define the set of reporting windows to display.
- The Search Criteria lists the filters that have been applied to the current dataset.
- Use the Action Bar to navigate throughout your report, generate a CSV file, switch to a graphical view or notify devices.

4 Customizing Reports

At the end of this chapter, you will:

- Understand the different reporting object types, including views, view groups, filters, filter groups, and windows.
- Understand how to modify the reporting object files to create custom reports.
- Be presented with an example scenario showing how to customize reporting objects.

The Reporting Server allows for extensive report customization by modifying any of the reporting object files. These files are located, by default, in the `Objects` directory after installing the Reporting Server. These reporting object files determine what data you will see and in what format it is presented on the Reporting Server Web page.

There are multiple reporting object file types that you can modify. Each type is located within a separate sub-directory within the appropriate language folder, as seen in Figure 17 below.

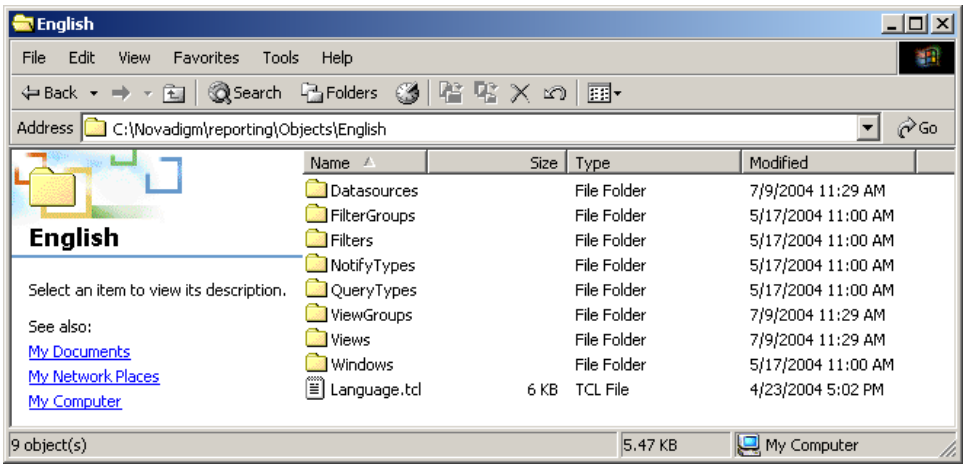


Figure 17: Reporting object file directories.

Customizing Reporting Server reporting object files requires a basic knowledge of XML. Familiarity with TCL and HTML is also beneficial.

When generating a reporting page, the Reporting Server looks for all files of a specific extension type, **`*.views`**, for example, within the directories mentioned above.

Backing up Your Reporting Object Files

Before you begin modifying any reporting object files, make sure to backup the original files. It is recommended that you rename any of the files you will modify with an easily identifiable designator, your company name or initials, for example. If your company initials were ABC, then you could copy and rename the View Group object, **Hardware Reports.viewgroup** to **ABC_HardwareReports.viewgroup**.

Since the Reporting Server only looks for specific extensions, renaming your modified files allows easy incorporation into your reporting environment. Also, any updates to the Reporting Server files will be easily incorporated into your existing configuration without undoing any of your report customizations.

Understanding the Reporting Object Files

Before you customize any reporting object files, become familiar with the purpose of each file and how they are constructed.

Reporting Object File Construction

Each reporting object file is designed using XML (Extendable Markup Language) as well as TCL (Tool-Command Language) and HTML. You should have a basic understanding of each of these programming languages before you attempt to make any modifications to your reporting object files. Use any text editor to modify and save new reporting object files.

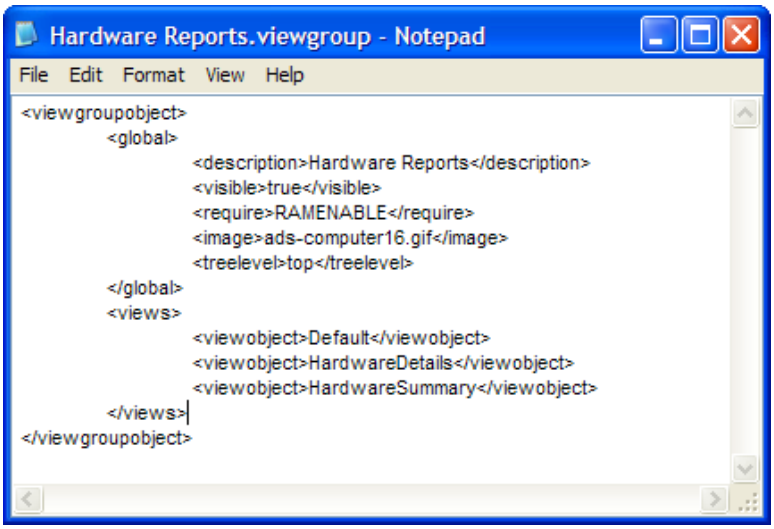


Figure 18: A sample View Group object file viewed using Notepad.

Reporting Object File Types

Several reporting object file types can be modified to generate custom report pages. These include:

- View Groups
- Views
- Filter Groups
- Filters
- Windows
- Datasources

View Group Objects and View Objects

View Group (`*.viewgroup`) objects contain a selectable list of View objects available for that category. Each View Group contains a specific set of available Views. A view group can include another view group in its list to allow for multiple levels in the tree view.

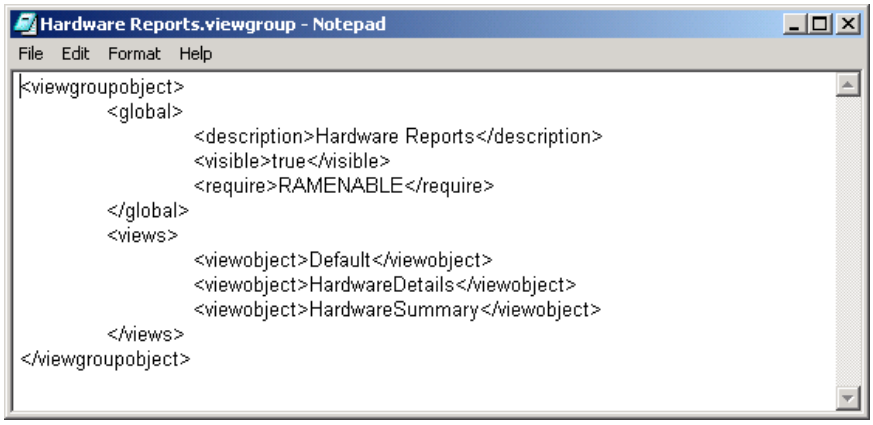


Figure 19: Sample View Group object file.

View (`*.view`) objects determine the currently active window object displayed on the right side of the Reporting Server web page.

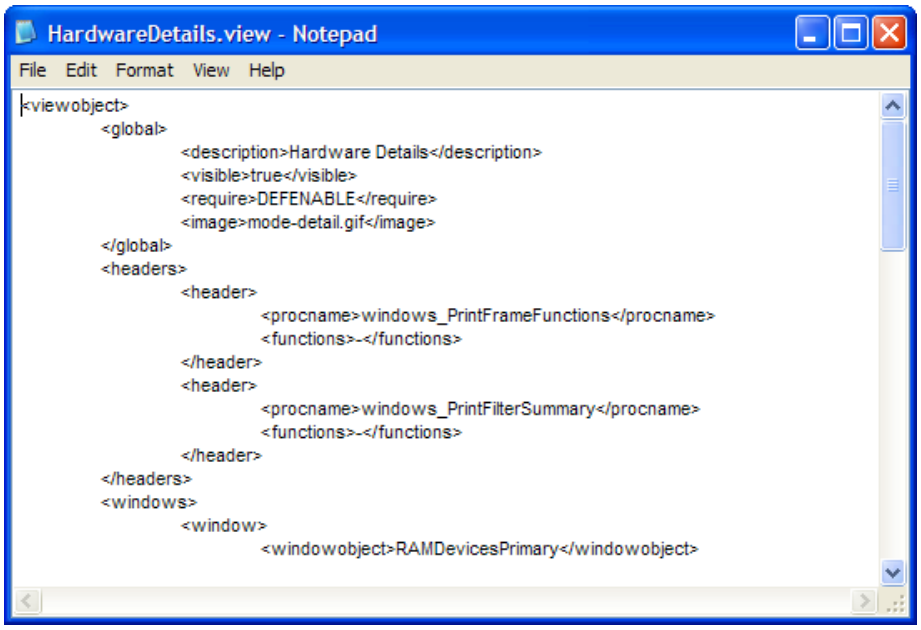


Figure 20: Part of a sample View object file.

Modifying View Group objects and View object files will determine what is displayed on the left-hand side of the Reporting Server Web page in the two drop-down list boxes within the Display Controls/Reporting Views section.

The currently active View object is displayed in the top-right corner of the page after **Current Reporting View**.

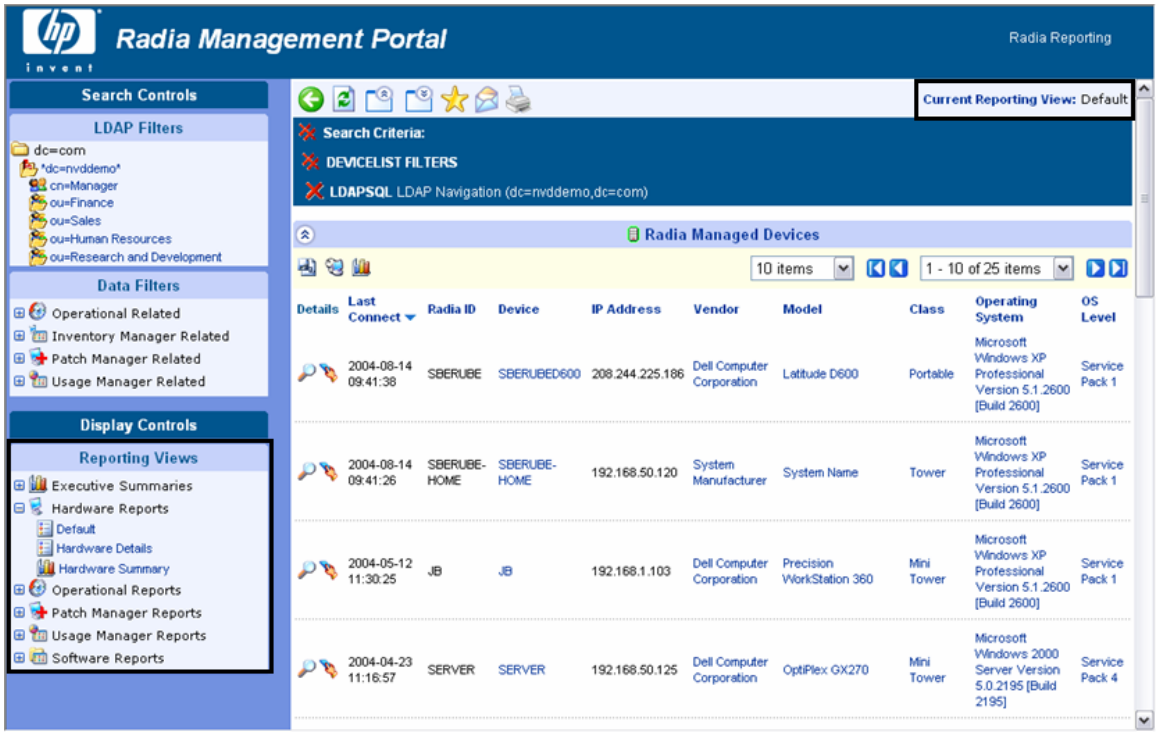


Figure 21: View Groups, Views and the Current Reporting View displayed on the Reporting Server Web page.

In addition to adding or altering Display Controls, View object files are used to determine which window objects are present. View objects can also determine specific Window object settings using the **Window Object Overrides** section. The parameter values within this section take precedence over any parameter values within a Window object file.

Filter Group Objects and Filter Objects

Filter Group (`*.filtergroup`) objects determine a selectable list of Filter objects. Like View Groups and Views, Filter Groups contain a specific set of available Filters. Depending on the Filter Groups selected, different Filter objects can be made available. Filter group objects can contain other Filter Group objects to allow for multiple levels in the tree view.

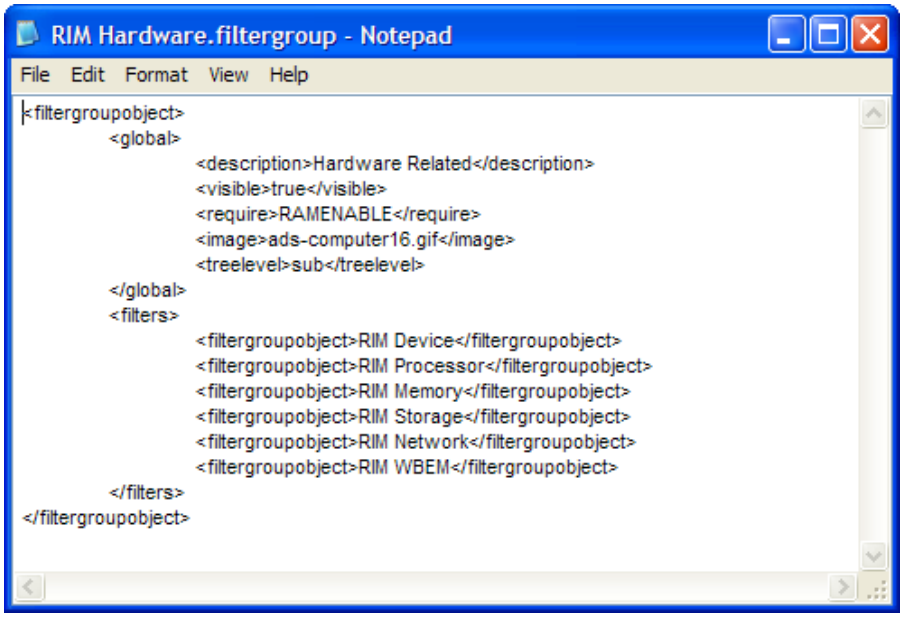


Figure 22: Sample Filter Group object file.

Filter (`*.filter`) object files define which data is displayed within each reporting page.

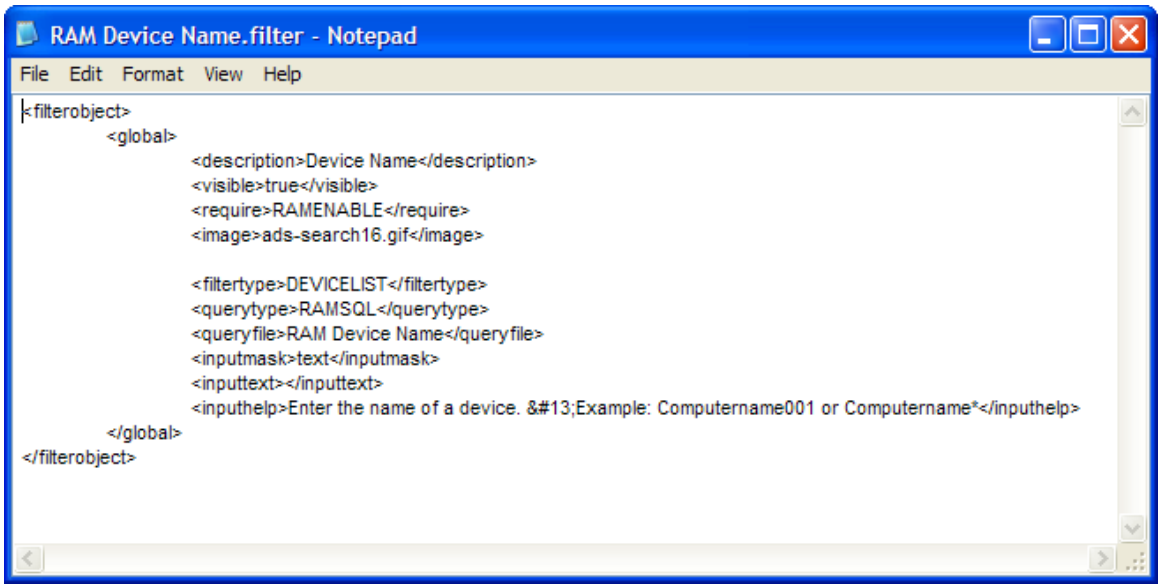


Figure 23: Sample Filter object file.

Window Objects

Window object files (`*.window`) define the data layout, including column headings, icons, sort fields, default window mode, default window state, drill down elements and column data appearances.

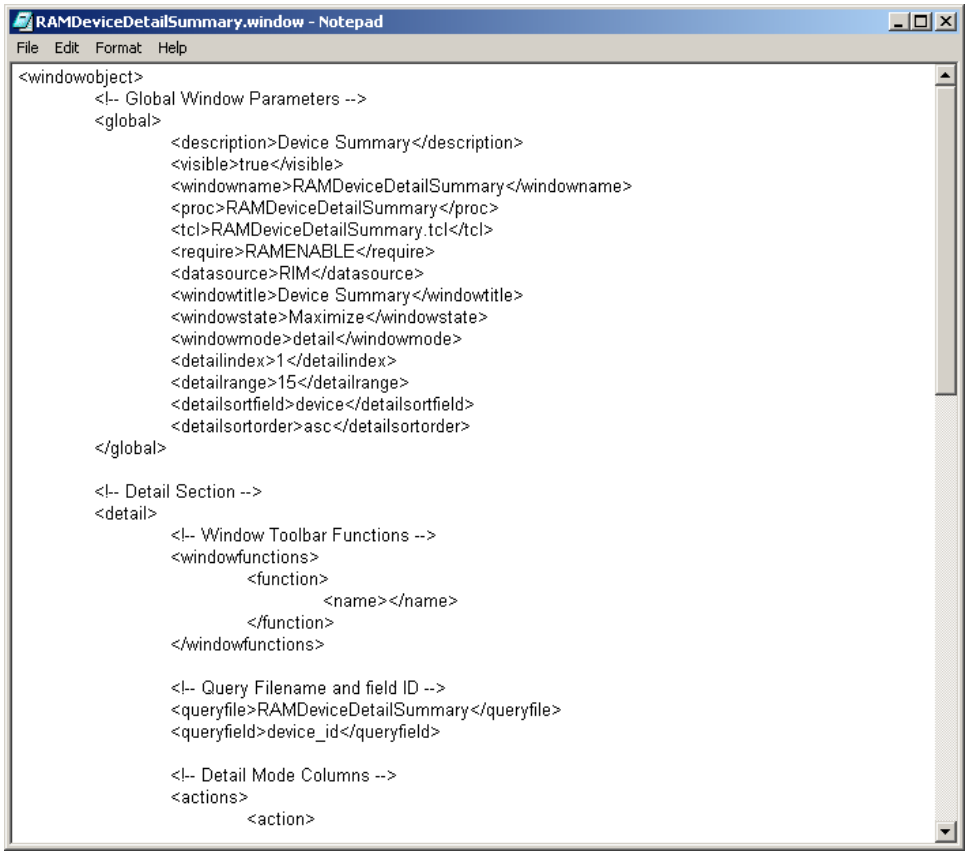


Figure 24: Part of a sample Window object file.

Each existing reporting object file type can be modified or copied to create new reporting pages.

Datasource Objects

Datasource object files (`*.datasource`) allow you to add multiple data sources for reporting. For example, you may want to add more SQL data sources or future DSML/LDAP data sources.

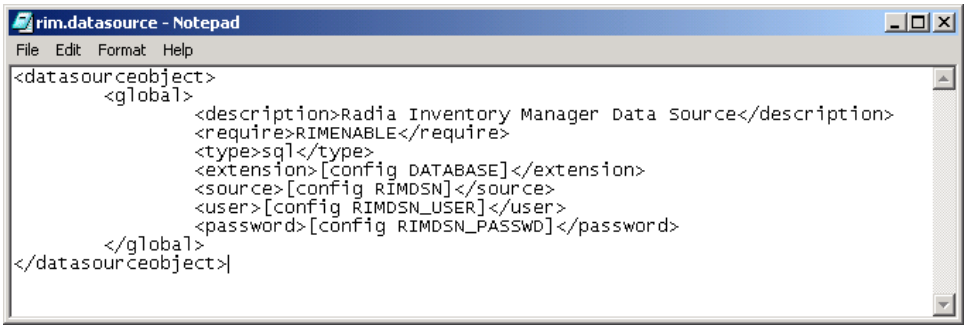


Figure 25: Sample Datasource object file.

Modifying Reporting Object Files

Use a text editor, like Notepad.exe, to modify your reporting object files. Each file is built using XML, TCL and HTML. Make sure you are familiar with these programming languages before you begin to modify these files.

Parameters within the reporting object files are defined using XML. For example, the global section description parameter within the `HardwareDetails.view` object file is defined as:

```
<description>Hardware Details</description>
```

Note the leading and trailing XML descriptor tags.

If you wanted to change the `HardwareDetails.view` object, global description to Hardware Specifics, you would need to change the text between the XML tags to:

```
<description>Hardware Specifics</description>
```

After the file is saved, the next time you view or refresh your Reporting Server Web page, the selectable entry within the **Views** drop-down list box, Hardware Details, will have been replaced with Hardware Specifics.

The following paragraphs and tables explain each object file section and their associated parameters and values. Modify any of these values to change or create reporting objects.

Modifying View Group Objects

View Group object files (`*.viewgroup`) contain two configurable sections: global and views. View groups can contain other View Groups to allow for multiple tree-view levels.

View Group Object Global Section

The global section is used to define the View Group name within the Display Controls drop-down list box, whether or not this View Group is visible and what settings are required for this View Group to be enabled. A sample <global> section follows:

```
<global>
   <description>Hardware Reports</description>
   <visible>true</visible>
   <require>RAMENABLE</require>
   <image>ads-computer16.gif</image>
   <treelevel>top</treelevel>
</global>
```

Table 3: View Group Object Global Section Parameters

Parameter	Description
<description>	What is displayed within the View Group drop-down list.
<visible>	True = This view group will be present in the View Group drop-down list. False = Hide this view group selection.
<require>	Prerequisite files or configuration file settings required for this View Group to be available.
<image>	Image used for tree view icon.
<treelevel>	top = load and show at top level. sub = load and show at sub level.

View Group Object Views Section

The views section determines which view objects are available when the view group is selected. A sample <views> section follows:

```
<views>
    <viewobject>Default</viewobject>
    <viewobject>HardwareDetails</viewobject>
    <viewobject>HardwareSummary</viewobject>
</views>
```

Table 4: View Group Object Views Section Parameters

Parameter	Description
<viewobject>	View object name. Define more than one view objects by adding additional viewobject lines.

Modifying View Objects

View object files (*.view) contain the following configurable sections: Global, Header and Windows.

View Object Global Section

The global section is used to define the View name within the Display Controls drop-down list box, whether or not this View is visible and what settings are required for this View to be enabled. A sample <global> section follows:

```
<global>
    <description>Hardware Details</description>
    <visible>true</visible>
    <require>DEFENABLE</require>
    <image>mode-detail.gif</image>
</global>
```

Table 5: View Object Global Section Parameters

Parameter	Description
<description>	What is displayed within the **Views** drop-down list.
<visible>	True = This view will be present in the **Views** drop-down list. False = Hide this view selection.
<require>	Prerequisite files or configuration file settings required for this View to be available.
<image>	Image used for tree-view icon.

View Object Header Section

The View object header section contains parameters used to define what items are available within each window heading. A sample `<header>` section follows:

```
<headers>
    <header>
        <procname>windows_PrintFrameFunctions</procname>
        <functions>-</functions>
    </header>
    <header>
        <procname>PrintFilterSummary</procname>
        <functions>-</functions>
    </header>
</headers>
```

Table 6: View Object Header Parameters

Parameter	Description
<procname>	Tcl procedure name to call for the header. The defaults are windows_printframefunctions and windows_printfiltersummary. The default functions draw the respective data displayed.

Parameter	Description
<functions>	List of parameters to pass to the function. Currently, most values are set to – (dash) as this is reserved for future use (with the exception of the sub-views, which have the value back to enable the back button).

View Object Windows Section

The windows section determines which window objects are available when the view is selected. The Window Object Overrides section allows you to alter the appearance of any window object by overriding parameter values set within the window object with new values. This allows for conformity between window objects available with a certain view object. A sample `<windows>` section follows:

```
<windows>
	<window>
		<windowobject>RAMDevicesPrimary</windowobject>

		<!-- Window Object Overrides -->
		<windowstate>Maximize</windowstate>
		<windowmode>detail</windowmode>
		<detailindex>1</detailindex>
		<detailrange>15</detailrange>
		<detailsortfield>-</detailsortfield>
		<detailsortorder>-</detailsortorder>
	</window>
</windows>
```

Table 7: View Object Windows Parameters

Parameter	Description
<windowobject>	Determines the window object to load.
Window Object Overrides	

Parameter	Description
<windowstate>	Determines the window object state.
<windowmode>	Determines window object mode.
<detaiindex>	Starting record number. Usually will be 1 (to start at record number 1).
<detailrange>	Number of records to display at one time.
<detailsortfield>	Default SQL field name by which to sort data results.
<detailsortorder>	Sort order. asc = sort ascending. des = sort descending.

Modifying Filter Group Objects

Filter Group objects (`*filtergroup`) contain two configurable sections, global and filters. Filter Groups can contain other Filter Groups to allow for multiple levels in the tree view.

Filter Group Object Global Section

The global section is used to define the Filter Group name within the Filter Group drop-down list box, whether or not this Filter Group is visible and what settings are required for this Filter Group to be enabled. A sample `<global>` section follows:

```
<global>
   <description>Hardware Related Filters</description>
   <visible>true</visible>
   <require>RAMENABLE</require>
   <image>ads-computer16.gif</image>
   <treelevel>sub</treelevel>
</global>
```

Table 8: Filter Group Object Global Parameters

Parameter	Description
<description>	What is displayed within the **Filter Group** drop-down list.
<visible>	True = This filter group will be present in the **Filter Group** drop-down list. False = Hide this filter group selection.
<require>	Prerequisite files or configuration file settings required for this filter group to be available.
<image>	Image used for tree view icon.
<treelevel>	top = load and show at top level. sub = object is a sub-level branch.

Filter Group Object Filters Section

The filters section determines which filter objects are available when the filter group is selected. A sample `<filters>` section follows:

```
<filters>
   <filterobject>RIM Device Vendor</filterobject>
   <filterobject>RIM Device Model</filterobject>
   <filterobject>RAM Device Name</filterobject>
   <filterobject>RIM Device Class</filterobject>
   <filterobject>RIM Device Serial Number</filterobject>
   <filterobject>RAM Device Memory</filterobject>
   <filterobject>RIM Device CPU Count</filterobject>
   <filterobject>RIM Device CPU Speed Less</filterobject>
   <filterobject>RIM Device CPU Speed More</filterobject>
   <filterobject>RAM Drive Space Free</filterobject>
</filters>
```

Table 9: Filter Group Object Filters Section Parameters

Parameter	Description
<filterobject>	Filter object name. Define more than one filter objects by adding additional filterobject lines.

Modifying Filter Objects

Filter object files (`*.filter`) contain a global configurable section.

Filter Object Global Section

The global section determines the filter name displayed, filter type, query type and parameters as well as any required input configurations. A sample `<global>` section follows:

```
<global>
   <description>Device ID</description>
   <visible>true</visible>
   <require>RAMENABLE</require>
   <image>ads-search16.gif</image>
   <querytype>RAMSQL</querytype>
   <queryfile>RAM Device ID</queryfile>
   <inputmask>text</inputmask>
   <inputtext></inputtext>
   <inputhelp>Enter the device_id of a device.</inputhelp>
</global>
```

Table 10: Filter Object Global Section Parameters

Parameter	Description
<description>	What is displayed within the **Filter** drop-down list.

Parameter	Description
<visible>	True = This view will be present in the **Filter** drop-down list. False = Hide this view selection.
<require>	Prerequisite files or configuration file settings required for this Filter to be available.
<image>	Image used for tree view icon.
<querytype>	Determines from where the data is be retrieved. Either, RAMSQL (for Radia Application Manager database), RIMSQL (for Inventory Manager database), RUMSQL (for Usage Manager database) or RPMSQL (for Patch Manager database).
<queryfile>	The SQL command used to obtain the data.
<inputmask>	date – inserts a calendar icon allowing the user to select a date. text – simple text input. dropdown - create a selectable dropdown list with predefined filters. Edit inputtext to enter values for the drop-down list. none - disables input field. Requires the filter to be hard coded (for example, Device last connect > 30 days).

<inputtext>	Input mask values.
	For text input mask, whatever is entered here will display as the default text input box value.
	For date input mask, will display default date. Must be a valid date format.
	For dropdown input mask value, use a space-delimited list to create the list values. For example:
	`<inputtext>a b c</inputtext>`
	Creates a dropdown list with the elements a b and c.
	Alternatively, a SQL query can be used to populate the dropdown list. For example:
	`<inputtext>[sql execute RIMDSN "select distinct os from deviceconfig"]</inputtext>`
	Creates a dropdown list of OS values.

<inputhelp>	Help icon text displayed on mouse-over.

Modifying Window Objects

Window object files (`*.window`) contain three main configurable sections, global, details and graph.

Window Object Global Section

The global section determines the window object description and initial settings. A view object may override some window object settings. A sample `<global>` section follows:

```
<global>
   <description>Device Summary</description>
   <visible>true</visible>
   <windowname>RAMDeviceDetailSummary</windowname>
   <proc>RAMDeviceDetailSummary</proc>
   <tcl>RAMDeviceDetailSummary.tcl</tcl>
   <require>RAMENABLE</require>
   <datasource>RIM</datasource>
   <windowtitle>Device Summary</windowtitle>
   <windowstate>Maximize</windowstate>
   <windowmode>detail</windowmode>
   <detailindex>1</detailindex>
   <detailrange>15</detailrange>
   <detailsortfield>device</detailsortfield>
   <detailsortorder>asc</detailsortorder>
</global>
```

Table 11: Window Object Global Section Parameters

Parameter	Description
<description>	Window object description.
<vsible>	True = This window will be visible. False = Hide this window object.

Parameter	Description
<windowname>	A unique descriptive name, same as the filename without the extension.
<proc>	For internal use only.
<tcl>	TCL script file to be used.
<require>	Prerequisite files or configuration file settings required for this Filter to be available.
<datasource>	Prerequisite data source setting required within the configuration file.
<windowtitle>	Window object title displayed.
<windowstate>	Window object state.
<windowmode>	Window object mode.
<detailindex>	Starting record number. Usually will be 1 (to start at record number 1).
<detailrange>	Number of records to display at one time.
<detailsortfield>	Default SQL field name by which to sort data results.
<detailsortorder>	asc = ascending sort order. des = descending sort order.

Window Object Detail Section

The detail section determines the window toolbar functions, query files and detail column settings. Within the details section are three sub-sections: Windows Toolbar Functions, Query Filename and Field ID and Detail Mode Columns. A sample `<detail>` section follows:

```
<detail>
    <!-- Window Toolbar Functions -->
    <windowfunctions>
          <function>
                <name></name>
          </function>
    </windowfunctions>
```

```
    <!-- Query Filename and field ID -->
    <queryfile>RAMDeviceDetailSummary</queryfile>
    <queryfield>device_id</queryfield>

    <!-- Detail Mode Columns -->
    <actions>
          <action>
                <enabled>0</enabled>
                <image></image>
                <columnname></columnname>
                <url></url>
                <urlalt></urlalt>
                <urltarget></urltarget>
          </action>
    </actions>
    <columns>
          <column>
                <fieldname></fieldname>
                <fieldtype></fieldtype>
                <fieldsettings></fieldsettings>
                <columnname></columnname>
                <columnsettings></columnsettings>
                <url></url>
                <urlalt></urlalt>
                <urltarget></urltarget>
          </column>
    </columns>
</detail>
```

Windows Toolbar Functions Sub-Section

This section allows for icons to be added to the window toolbar. In order to be useful, the icon must match the window mode (a graph cannot be exported, for example).

- **ExportCSV** adds the export to CSV Icon (Detail mode only).
- **ViewMode** adds the icon to change to graphical mode or when in graphical mode, adds the icon to change to detail mode. A corresponding mode must exist in the object.
- **Notify** adds notify support (used for the RAMDevicesPrimary window only).
- **NavBar** adds the right side navigation options (detail mode only).

Query Filename and Field ID Sub-Section

This section determines which query file should be used as well as which query field is required. The query filename used automatically appends an extension of `.sql` or .oracle.

Inventory Manager and Radia Application Manager data requires a queryfield of device_id. The Patch Manager and Usage Manager data requires a queryfield of device_name.

Detail Mode Columns Sub-Section

The Detail Mode Columns sub-section determines the default layout for each detail column (the first two columns to the left displayed for each window object). Two additional sub-sections, `<actions>` and `<columns>` are included within this section.

Table 12: Detail Mode Columns Section – Action Sub-Section Parameters

Parameter	Description
<enabled>	1 = Show detail column. 0 = Hide detail column.
<image>	Determines whether an image file is used to portray the data. Image files are stored within the `Images` folder.
<columnname>	The name displayed for the column.

Parameter	Description
<url>	Hyperlink value. Can be an internal reporting function or an external url. For example, **`Error! Hyperlink reference not valid.`**, will add a link using the value from row 2 from the SQL query as a symbolic.
<urlalt>	Mouse-over text.
<urltarget>	Determines whether or not a new window is opened. For example, clicking the Remote Control (VNC) icon opens a new window.

Table 13: Detail Mode Columns Section – Columns Sub-Section Parameters

Parameter	Description
<fieldname>	Must match SQL field name in the query 1 for 1 match.
<fieldtype>	text date numeric image dropdown - create a dropdown list based on space separated values.
<fieldsetting>	Manipulate output of data.
<columnname>	Friendly name to display on report. If column name is blank, no column will be displayed for that data.
<columnsetting>	Manipulate entire column.
<url>	Hyperlink value. Can be an internal reporting function or an external url. For example, **`Error! Hyperlink reference not valid.`**, will add a link using the value from row 2 from the SQL query as a symbolic.
<ulalt>	Mouse-over text.
<ultarget>	Determines whether a new window is opened. Set to _new to open a new window. For example, clicking the Remote Control (VNC) icon opens a new window.

Window Object Graph Section

The graph section determines the settings for the graphical representation of your data. This includes two sub-sections, one for incorporating functions (Window Toolbar Functions) and another for manipulating the graphical representation of the data (Chart Objects For Graphical Mode). A sample `<graph>` section follows:

```
<graph>
   <!-- Window Toolbar Functions -->
   <windowfunctions>
          <function>
                 <name>ViewMode</name>
          </function>
   </windowfunctions>

   <!-- Chart Objects for Graphical Mode -->
   <charts>
          <chart>
<queryfile>RUMCoreProductSummary_Graph_UsageTotals</queryfile>
                 <queryfield>devicename</queryfield>
          <charttype>chart_drawBarChartStacked</charttype>
                 <chartwidth>700</chartwidth>
                 <chartheight>400</chartheight>
                 <charttitle>Usage Totals</charttitle>
                 <charturl></charturl>
                 <params>
                 </params>
          </chart>
   </charts>
</graph>
```

Summary

- Reporting objects are designed using the Extendable Markup Language (XML).
- Before you modify any reporting objects, make sure you are familiar with XML as well as TCL and HTML.
- Reporting object files are located within the Objects directory.
- Rename any reporting objects files you modify in order to allow for seamless product updates.

5 Troubleshooting

At the end of this chapter, you will:

- Be able to access the Reporting Server log file.
- Be able to resolve typical problems, such as Reporting toolbars not loading properly.
- Be able to adjust the CTI timeout value of IIS, if necessary.

About the Reporting Server Log

A `reporting.log` file is located in the `log` directory of the base Reporting Server directory. By default, the log file is located:

SystemDrive:\Novadigm\ReportingServer\log\reporting.log

Use the log file to review or troubleshoot Reporting Server session activity. The most recent entries are located at the bottom of the log.

- SQLLOGIN entries identify all SQL databases that have been configured for reporting access.
- SESSION START entries also give the specific session identifier, such as "radia9061259".
- SESSION END entries mark the end of the session.

Select `reporting.log` entries for a typical session. Here is a sample:

```
. . . SQLLOGIN: RIMDSN
. . . SQLLOGIN: RUMDSN
. . . SQLLOGIN: RPMDSN
. . . SESSION START: radia9061259
. . . RAMSQL: select device_id from dbo.DeviceConfig where device_id in
      ('sberube','server','sberube-home','sberubed600', 'sberube-d600','tberube','sberube-
      vm2k','administrator','device100','device101','device102','device103','device105','devic
      e104','device714','device435','device998',device200','device201','device202','device203'
      ,'sberube-t8100','sberubevmxp','device300','device301','device302','device400',
      'device401','device402','device403','device404','device405','device406','device407','dev
      ice408','device409','device410','device411','device412','device413','device414','UNIXPPL
      MHRURBZTU','UNIXEIQTUUJEAOWZ','UNIXHYWMOADCOVZL','UNIXTXBIRZQFVPHI','UNIXPMKAKGRGAJBN','
      device1000')
. . . RUMSQL:
. . . RPMSQL:
. . . TEMPTABLE CREATE BEGIN: Temptable ##radia9061259dn has been created.
. . . SQL FILTERS: 1 SQL filters type(s) used.
. . . WINDOW RAMDevicesPrimary DETAILDATA BEGIN: Processing query on RIMDSN.
. . . WINDOW RAMDevicesPrimary DETAILDATA END: Query processing complete on RIMDSN.
. . . SESSION END: radia9061259
```

Common Reporting Server Problems and Solutions

- **Problem: Left Toolbar not loading.**
  The left-side navigation toolbar of the Radia Reporting page isn't loading properly.

Solution: This has been resolved on Compaq machines by disabling TEAMING.

- **Problem: CGI Timeout.**
  The report window shows a "CGI timeout" when I run reports against the entire dataset. How do I change the CGI timeout?

  Solution: The default IIS value for CGI timeout is 5 minutes, or 200 seconds. Reporting Server queries against a large dataset can take longer. We recommend changing the CGI timeout to 1200 seconds, or 20 minutes. For details on how to do this, see Modifying the CGI Timeout Value in IIS below.

- **Problem: Browser hangs when opening Reporting Page -- navigation bar or reports are not displayed.**
  When browsing to the Reporting page, the page hangs when trying to display the navigation bar (`navigate.tcl`) or reports (`results.tcl`).

  Solution: Change the file permissions of `C:\WINNT\Temp` to allow everyone full control, including subfolders.

Modifying the CGI Timeout Value in IIS

The CGI timeout is documented by Microsoft at: **http://www.microsoft.com/windows2000/en/server/iis/htm/asp/apro6e9g.htm**.

If you receive a CGI Timeout error, you can use one of the following procedures to modify the CGI timeout value in IIS.

To change the CGI timeout value in IIS using a script

- From a command prompt, access the *SystemDrive*`:\inetpub\adminscripts` directory, and type:

  `cscript adsutil.vbs set w3svc/CGITimeout "1200"`

  then press **Enter**.

To change the CGI timeout value in IIS using Windows tools

1. On your Windows desktop, right-click **My Computer** and select **Manage**.
2. Browse to **Services and Applications**, **Internet Information Services**, **Web Sites**.
3. Display the Web Sites Properties page and select the **Home Directory** tab.

4 Click the **Configuration** button at the bottom of the page to open the Application Configuration window.

5 Select the **Process Options** tab.

6 In the **CGI script timeout** text box, type the desired value in seconds. We recommend 1200 seconds. Click **OK** to save the Process Option.

7 Click **OK** to save the Application Configuration entry.

Summary

- A `reporting.log` file is saved in the log directory of your Reporting Server base directory.
- Refer to the log entries to obtain Reporting Server session activity details.
- Modify the CTI timeout value for IIS, if necessary.

A Sample Reporting Scenario

This appendix presents a sample reporting scenario that will help you become familiar with the Reporting Server's features. The scenario demonstrates the use of filter options and view options as well as some of the functions available after a report is generated.

Scenario: Report for Sales Department on Devices Needing Service Pack Updates

The following scenario instructs how to use the Reporting Server to obtain a list of all Radia Managed Devices in the Sales Department that have the desired operating system for rolling out a new application, but which do not have the latest service pack level. The final report itemizes those devices that need a service pack update prior to installing the application.

The scenario assumes that the Reporting Server has been configured to access the Active Directory for your enterprise as well as your Inventory Manager SQL database. It has been installed on the IIS server with the hostname: myHostIIS.

Task 1 Access Radia Reporting

To access the Reporting Server, open a browser window and type:

`http://myhostIIS/reporting`

Where *`myhostIIS`* is the host name for the IIS web server on which Reporting Server was installed and where *`reporting`* is the Alias assigned to Reporting Server during installation and configuration.

The Reporting Server Web page opens with the default toolbars and windows.

At the top-left of the page, the **LDAP Filters** area includes the enterprise's Active Directory tree.

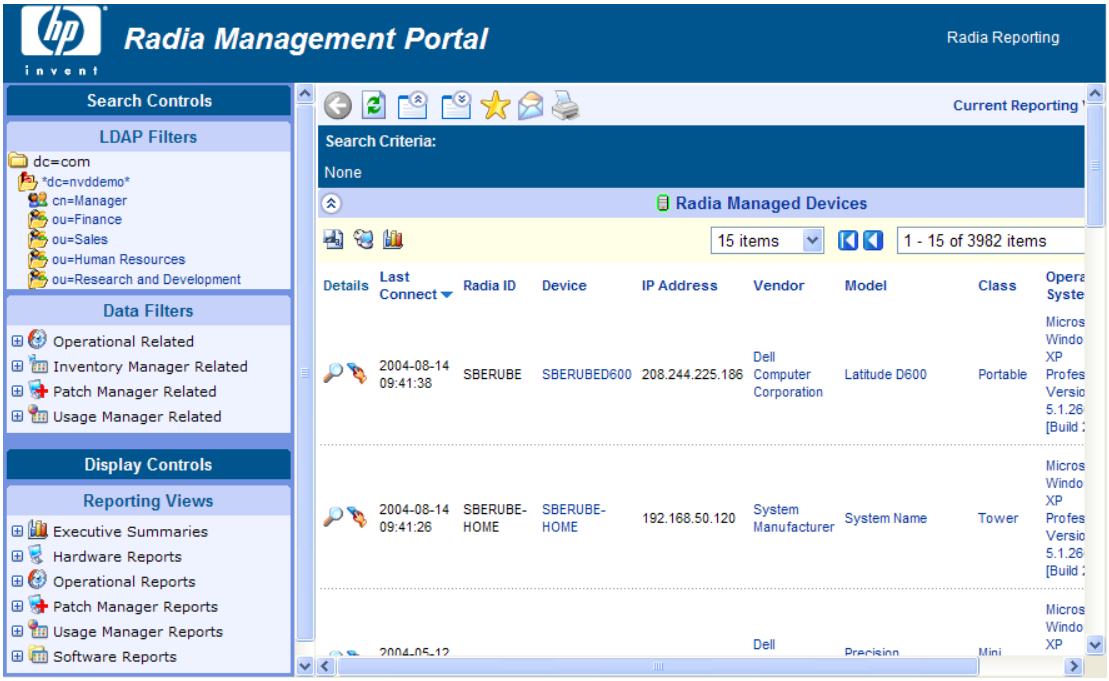


Figure 26: First Access to the Reporting Server.

We will select various filters from the Reporting Server interface to limit the report to only those Devices in the Sales Department that need service pack updates for our new application.

Task 2 Search for Sales Department Devices Only

Use the LDAP Filters area to select the Sales group from the Active Directory structure. Making a selection in this area adds an LDAP filter of Sales to the Search Criteria, and returns a list of all devices assigned to Sales.

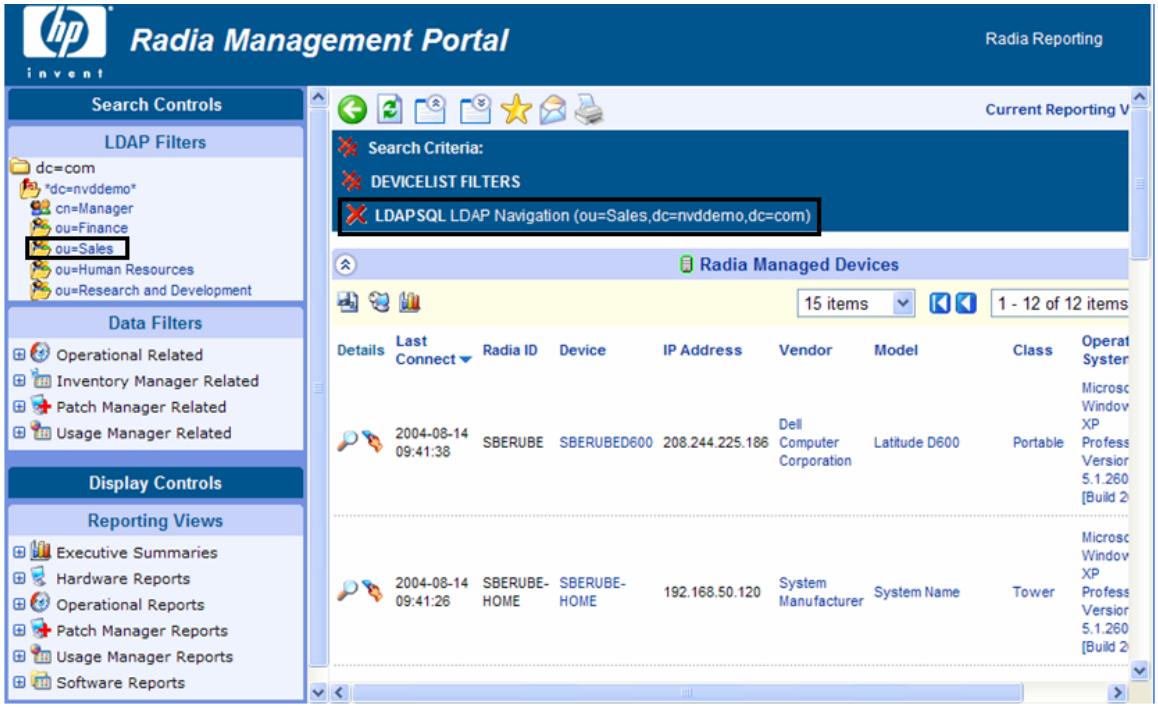


Figure 27: Devices filtered to the Sales Department.

Task 3 Limit Search to Targeted Operating System

From the list of devices in sales, we will click the following hyperlink entry (found in the Operating System column of the Radia Managed Devices Report) to add another search criterion:

Class Operating System

Microsoft Windows 2000

Professional Version 5.0.2195 [Build 2195]

Now our result set shows all devices in Sales that have the **Microsoft Windows 2000 Professional Version 5.0.2195 [Build 2195]** operating system.

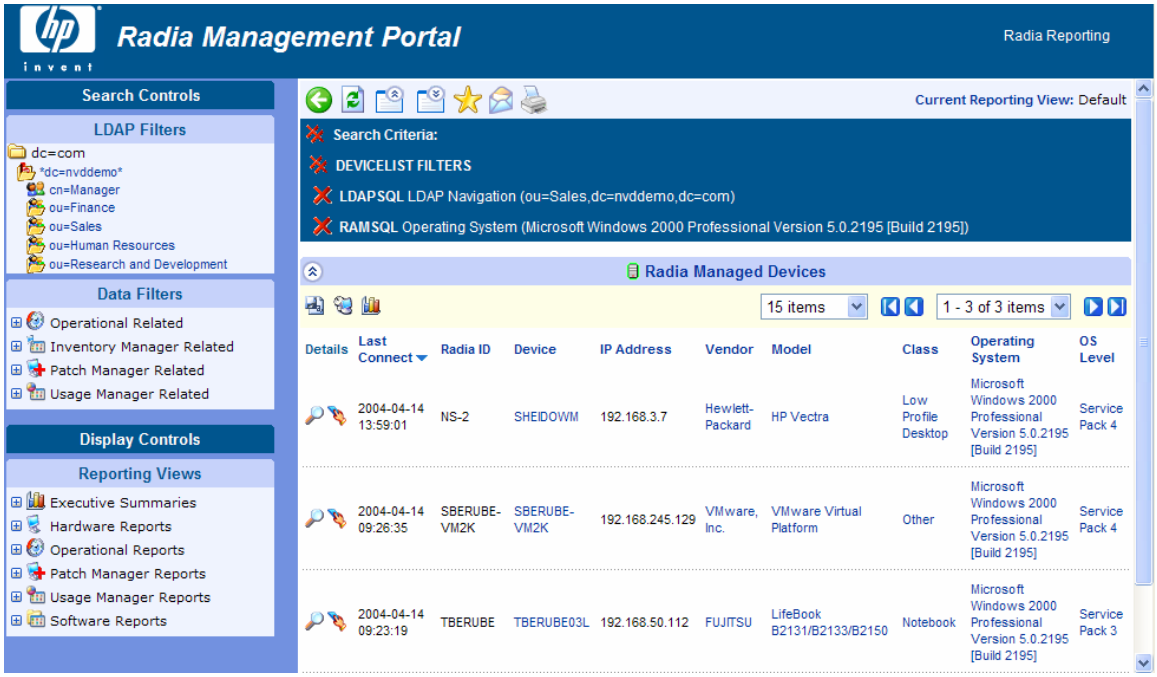


Figure 28: Sales Devices with the Required Operating System.

Task 4 Search for Operating Systems without Service Pack 4

Now we want to drill down further to identify devices from this list that do not have Service Pack 4. To do this, we're going to apply a data filter. Data Filters are easily applied from the Search Controls toolbar on the left side of the page.

In the **Data Filters** area, we will select a Filter Group and Filter. First, select the Filter Group **OS Related Filters**. Second, select the Filter **Operating System Level**.

In the **Filter Value** text box, type the value **`!Service Pack 4`** and click **Apply** (the ! denotes "not").

Our report now shows all Devices in Sales, which have an operating system of **Microsoft Windows 2000 Professional Version 5.0.2195 [Build 2195]** but which *do not* have **Service Pack 4**.

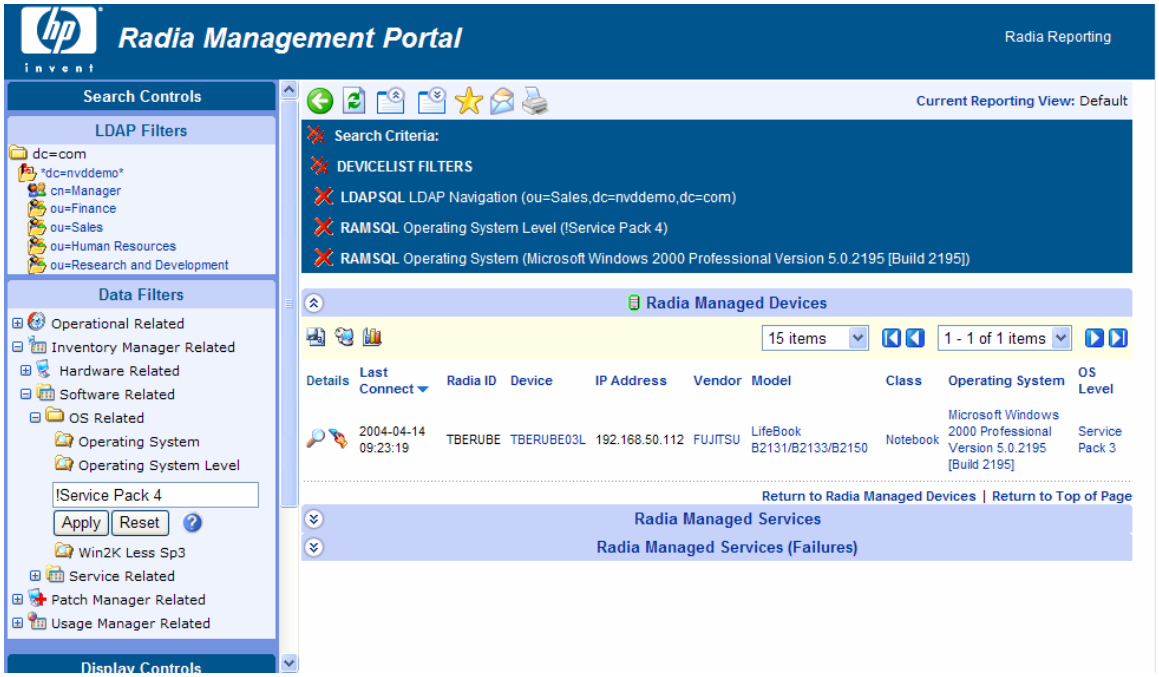


Figure 29: Sales Devices with desired Operating System, but missing required Service Pack Level.

Our report now lists those devices in the Sales department that need to have a Service Pack upgrade applied prior to rolling out the new application.

Task 5 Save or Print the Report

Print any report by clicking the printer icon above the Search Criteria.

Save the report by exporting it to a CSV file. Use **Export as CSV** in the Action bar within the report to export the data to a CSV file.

Index

S

T

U

V

W

X