HP OpenView Radia 4.1

HP OpenView Messaging Server Using Radia

Upgrade Procedures Guide

Software Version: 3.0

Windows Operating System

Software Version 3.0

Unix Operating Systems: AIX, HP-UX, Linux, Solaris, and Sunx86

June 2005

Legal Notices

Warranty

Restricted Rights Legend

Copyright Notices

Trademark Notices

Acknowledgements

Support

Please visit the HP OpenView support web site at:

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

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

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 enhancement requests online
- Download software patches
- 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://www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:

http://www.managementsoftware.hp.com/passport-registration.html

Preface

About this Guide

Who this Guide is for

This guide is for Radia system administrators who want to upgrade their HP-OpenView Messaging Server Using Radia environment to Version 3.0.

This guide contains information for the Windows and UNIX platforms.

You should be familiar with Radia products, such as the Radia Database, Radia Configuration Server, Radia methods such as ZTASKEND, the Radia Inventory Manager Server. If using Radia Patch Manager, you should be familiar with that product.

You should become familiar with newer Radia components, such as the Radia Reporting Server. For details, see the appropriate guides.

Conventions

You should be aware of the following conventions used in this book.

Table P.1 ~ Styles

Element	Style	Example
References	*Italic*	See the *Publishing Applications and Content* chapter in this book.
Dialog boxes and windows	**Bold**	The **Radia System Explorer Security Information** dialog box opens.
Code	`Andale Mono`	`radia_am.exe`
Selections	**Bold**	Open the **\Admin** directory on the installation CD-ROM.

Table P.2 ~ Usage

Element	Style	Example
Drives (system, mapped, CD)	Italicized placeholder	*SystemDrive*:\Program Files\Novadigm might refer to C:\Program Files\Novadigm on your computer. *CDDrive*:\client\radia_am.exe might refer to D:\client\radia_am.exe on your computer.
Files (in the Radia Database)	All uppercase	PRIMARY
Domains (in the Radia Database)	All uppercase	PRIMARY.SOFTWARE May also be referred to as the SOFTWARE domain in the PRIMARY file.
Classes (in the Radia Database)	All uppercase	PRIMARY.SOFTWARE.ZSERVICE May also be referred to as the ZSERVICE class in the SOFTWARE domain in the PRIMARY file.

Table P.4 ~ Radia Proxy Server Terminology

Term	Definition / Context
base installation directory	The location where your Radia Inventory Manager Server is installed. By default, the Radia Inventory Manager Server is installed into the following directory: • On a Windows machine, it is: <**SystemDrive**>**:\Novadigm\IntegrationServer** • On a UNIX machine, it is: /**opt/Novadigm/IntegrationServer**

Contents

1

Upgrading to HP-OpenView Messaging Server Using Radia 3.0

Use these procedures to upgrade an existing Radia Messaging Server 2.x environment to a Radia Messaging Server 3.0 environment. Use this document to:

■ Upgrade the RCS ZTASKEND REXX method to at least v 1.9.
   Optionally, port any custom code to the new ZTASKEND REXX.

■ Upgrade your Radia Messaging Server from 2.x to 3.0. Radia Messaging Server version 3.0 supports:

   ● Data Delivery Agents (DDA) and multiple data queues. DDAs provide the ability to post objects via ODBC directly to an SQL database or an Oracle database, for Inventory and Patch. This can eliminate the need for the Inventory Manager Server in most environments.

   ● Secure HTTPS routing using SSL.

■ Port any customized SQL code from RIM to RMS (when using DDAs to post objects to the SQL compliant database for Inventory using ODBC).

■ Review the Patch Manager migration requirements.

■ Optionally, enable secure HTTPS routing using SSL.

■ Use a store and forward server to place messages as close to the database as possible before posting using ODBC. (This is a best practice for both Windows and UNIX).

Upgrading your RCS ZTASKEND REXX Method

Prior to upgrading the Radia Messaging Server, it is a best practice to adopt the latest version of the ZTASKEND REXX delivered on the Radia 4.1 RCS CD.

It is also a best practice to upgrade the ZTASKEND REXX method before upgrading the Messaging Server.

About ZTASKEND Version 1.9

Version 1.9 of ZTASKEND REXX offers out-of-the-box support for both the RMS 2.x routing option (which uses a single, data default queue) as well as the new RMS 3.0 routing options available with the RMS Data Delivery Agents (which use multiple data queues).

Since ZTASKEND REXX is frequently customized, the comments in the ZTASKEND v1.9 code as well as the ZTASKEND discussion in the *Installation and Configuration Guide for the HP-OpenView Messaging Server Using Radia, v 3.0, (Messaging Server Guide)* point out where and how to customize it.

To upgrade your version of ZTASKEND REXX to version 1.9

1. Hewlett-Packard recommends stopping the RCS service to make these changes. However, if your RCS machine is operating close to 100% CPU usage, it will be necessary to stop the RCS service in order to replace ZTASKEND.

2. Copy the delivered version 1.9 of ZTASKEND REXX from the following platform-specific location on the **Radia 4.1 RCS CD** to the **rexx/Novadigm** directory of the RCS:

   \management_infrastructure\configuration_server\<platform>\media\rexx\NOVADIGM

Note
ZTASKEND rexx is frequently customized. If your current ZTASKEND rexx is customized, you'll need to port the custom code to the new ZTASKEND rexx version 1.9. You can place the copy of the ZTASKEND rexx (prior to customizations) in the **rexx/Novadigm** directory of the RCS. After porting your customizations, place the customized ZTASKEND rexx in the **rexx** directory of the RCS.

3. If your current ZTASKEND is not customized, go to the next step.

   If your current ZTASKEND is customized, review the comments throughout the ZTASKEND file itself before porting any customizations. Also refer to the discussion on ZTASKEND found in the *Messaging Server Guide*, which highlights the code sections most frequently used for customizations.

   Port any customizations to ZTASKEND version 1.9, as necessary. Document your changes.

4. Copy ZTASKEND version 1.9 (with any custom code, if any) to the **rexx** directory of the RCS.

5. Restart the RCS service, and continue with the Messaging Service upgrade.

After performing these steps, you can install the new Radia Messaging Server to the same location as the existing one. See *Upgrading a Radia Messaging Server* to 3.0 below.

Upgrading a Radia Messaging Server to 3.0

Use these steps to install version 3.0 of the Radia Messaging Server to the same location as an existing one. The install media is provided on the Radia 4.1 Infrastructure CD.

> **Note**
>
> Rename your existing rms.cfg file before you begin. This allows the install program to create a configuration file that includes support for new features, including:
>
> - Secure HTTPS routing using SSL
> - Load statement for Data Delivery Agents
> - Log configuration options

The RMS install program will:

- Create a new rms.cfg file (as long as you rename your existing one).
- Create *.dda.cfg configuration files for each Data Delivery Agent that is selected during the install. There are four available Data Delivery Agents: CORE, INVENTORY, WBEM and PATCH. The Data Delivery Agents can be used to post data to a SQL Database or Oracle Database.
- Add the same scripts and *.sql code to the Messaging Server that is provided with the Inventory Manager for creating the SQL tables and to modify the data in the Inventory database.
- The procedures include a post-install task of relocating custom SQL code from your Radia Inventory Manager Server to your Radia Messaging Server.

> **Note**
>
> The procedures include a post-install task of relocating any custom SQL code from your Radia Inventory Manager Server to your Radia Messaging Server.

The RMS install program will not:

- Install a Messaging Server fully configured for store and forward capabilities. Following the installation, you need to edit the appropriate configuration files and switch the routing options to forward messages to another Messaging Server. For details, refer to the Store and Forward Configuration topics in the *Messaging Server Guide*.

■ When a Messaging Server is co-located with an RCS on a UNIX platform, HP recommends a Store and Forward solution using a Windows based Messaging Server for the ODBC posting of data. The Messaging Server on the Windows platform should be located close to the SQL-compliant database. When configured this way, the ODBC drivers supplied with Windows are used for posting of messages to the backend SQL compliant database. For details, refer to the Store and Forward Configuration topics in the *Messaging Server Guide*.

To use the Messaging Server to post via ODBC from a Unix platform, third-party ODBC drivers must be installed. Currently, these are not supplied by the Messaging Server installation. Install the necessary drivers using installation instructions from the supplier. Include the DSN information in the appropriate Messaging Server or DDA configuration file.

To install Messaging Server 3.0 (Windows and UNIX)

HP recommends performing this upgrade after you have upgraded the ZTASKEND method on the RCS to version 1.9. See the earlier procedures on page 10.

1. Launch the installation program for the Radia Messaging Server, available from the following platform-specific location on the Radia Infrastructure CD:

> \extended_infrastructure\messaging_server\<platform>

- For Windows, click on **setup.exe** to launch the installation program.
- For a UNIX platform, enter the following command:

    ```
    ./install
    ```

    and press **Enter.**

2. Follow the prompts to complete the installation. Choose the same install location as your existing Radia Messaging Server.

3. If you installed the CORE, INVENTORY or WBEM Data Delivery Agents for direct posting of objects to the Inventory Database using ODBC, you will want to port any existing customizations you have on the Inventory Manager Server to the appropriate Messaging Server locations. See Migrating Custom SQL Code from the Inventory Manager Server to the Messaging Server on page 13.

HP-Recommended Best Practices for RMS 3.0

1. The default WORKER setting for each configuration file is 1 worker, and HP recommends using this setting with Messaging Server 3.0 (for both the RMS.CFG file and the individual DDA.CFG files. If multiple workers were configured in the past, they should be changed back to WORKERS 1.

2. The introduction of the new Data Delivery Agent modules for CORE, INVENTORY and WBEM objects allows for posting of these objects directly into a back-end SQL compliant database using ODBC. HP recommends that the Messaging Server used to post these objects via ODBC be placed as close to the SQL compliant database as possible to minimize the network response time. Often this means using the Messaging Server co-located with the RCS as a forwarding messaging server, and installing a downstream Messaging Server close to the

SQL compliant database that is configured to do the actual ODBC posting. This is illustrated in the following figure.
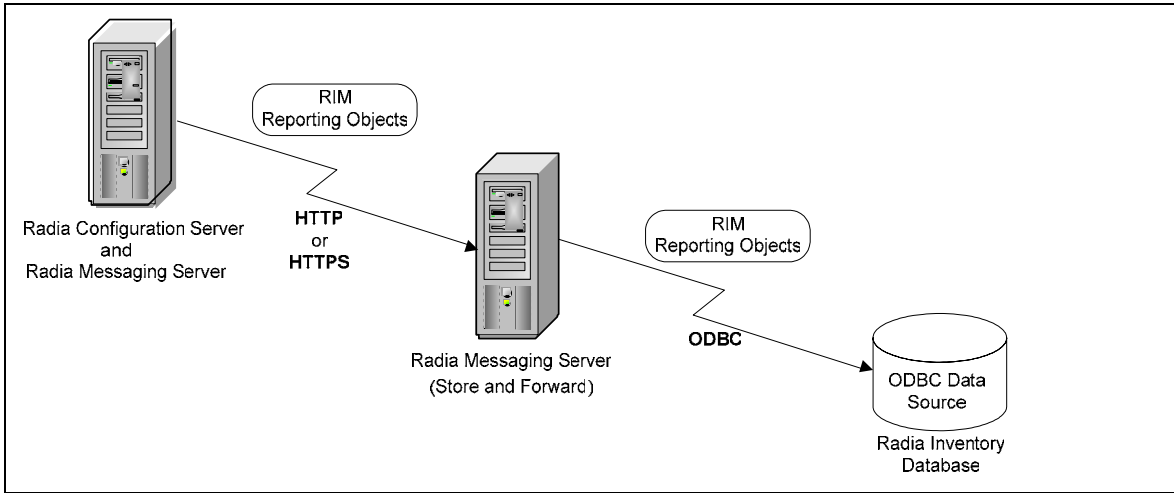

Figure 1: Radia Messaging Servers configured to pass data using a Store and Forward "Hop".

Refer to the *Messaging Server Guide* for details on how to configure store and forward servers in your environment.

Migrating Custom SQL Code from the Inventory Manager Server to the Messaging Server

If you elected to install any of the Data Delivery Agents for posting CORE, INVENTORY, and WBEM objects to the Inventory Manager using ODBC, the customized versions of any files listed in Table 1 on page 14 can be copied from their locations on your Inventory Manager Server to the equivalent locations on the Radia Messaging Server.

The Data Delivery Agents can be used to post data to a SQL Database or Oracle Database.

Note
Port the custom code to a Messaging Server being used to post data using ODBC to an Inventory database. It is not necessary to port any customizations to a Messaging Server that is forwarding data to another Messaging Server.

About the Scripts and SQL Queries used with the Data Delivery Agents

The Data Delivery Agents for CORE, WBEM and INVENTORY data post their message data into the same SQL tables created by the Inventory Server. These Data Delivery Agents use the exact same table definitions used by the Inventory Server to create tables, update and delete data. If the SQL tables have not been already created by an instance of the Inventory Server, when the Data Delivery Agent that uses the SQL table is started, the table will be created.

The definitions for these tables and associated SQL queries are contained in the /etc/<module name>/sql directory. The script necessary to map the CORE object data to the related SQL table column is taskend.tcl. This script is identical to the version of taskend.tcl on the Inventory Server. The script necessary to map the INVENTORY object data (FILEPOST object) is called filepost.tcl. Both these scripts are found in the /etc/<module name>/lib directory of the Messaging Server. Using the identical scripts found on the Inventory Server allows previous users of this Infrastructure service to migrate any customized scripts directly into the directory for the associated Data Delivery Agent module

To migrate custom code from Inventory Manager to Messaging Server

1. If necessary, stop the service for the Radia Messaging Server.

2. Create a backup of the Messaging Server \etc directory before porting any customized code.

3. Stop the [httpd] service for the Inventory Manager server.

4. Use Table 1 to locate the appropriate Messaging Server directory and file for each object type. Copy any customized versions of the code on your Inventory Server to the appropriate location on the Messaging Server.

Table 1: Directory Locations for Migrating Custom Code from RIM to RMS

Data Directory Agent: and Files	Inventory Server Directory Location	Messaging Server Directory Location
core.dda support:		
taskend.tcl	*<RIS>*\etc\rim\lib	*<RMS>*\etc\core\lib
*.sql files	*<RIS>*\etc\sql	*<RMS>*\etc\core\sql
inventory.dda support		
filepost.tcl	*<RIS>*\etc\rim\lib	*<RMS>*\etc\inventory\lib
*.sql files	*<RIS>*\etc\sql	*<RMS>*\etc\inventory\sql
wbem.dda support		
*.sql files	*<RIS>*\etc\sql\wbem	*<RMS>*\etc\wbem\sql

For example, if you have a customized version of taskend.tcl on your Inventory Server, copy it to the <RMS>\etc\core\lib location on the Messaging Server.

5. Restart the Radia Messaging Server service or process.

6. If you have used Data Delivery Agents with the ODBC routing options to post the CORE, INVENTORY, and WBEM inventory objects directly to an Inventory database, you do not need to restart the Inventory Manager Server.

Verifying the Patch Method Connection and Queue Name

■ Patch Manager requires a method connection in the Radia Database. For details, refer to the *Patch Manager Guide.*

■ If you installed the patch.dda and changed the name of the *Patch Message Directory to Scan* value during the Messaging Server installation (the expected value is **patch)**, you must change the -queue patch value in the ZMTHPRMS attribute of the PATCH_STATUS instance to match the Patch Directory to Scan value.

To modify the queue name in the PATCH_STATUS method

1. Use System Explorer to edit the ZMTHPRMS attribute of the PRIMARY.SYSTEM. ZMETHOD.PATCH_STATUS instance, as shown in Figure 2 below.

2. Adjust the **-queue patch** value to reflect the directory named as the "Patch Message Directory to Scan" during the Messaging Server installation.
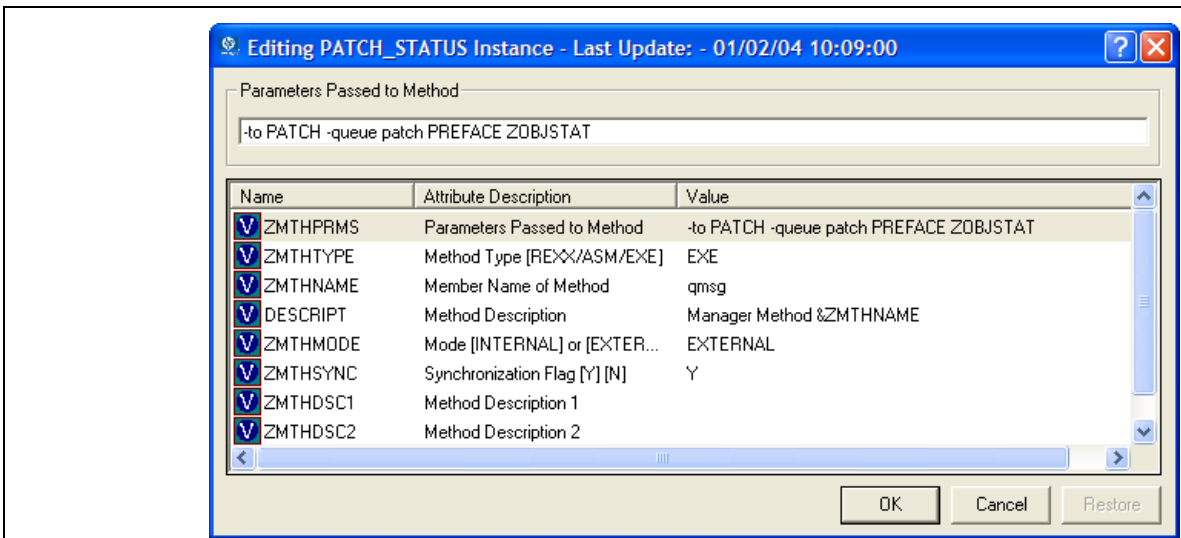

Figure 2: Specify the Patch queue name in ZMTHPRMS.

For example: if you entered `"..\ConfigurationServer\data\mypatch"` as the Patch Directory to Scan for the patch.dda, change the value of ZMTHPRMS in the PATCH_STATUS instance from:

```
-to PATCH -queue patch PREFACE ZOBJSTAT
```

to

```
-to PATCH -queue mypatch PREFACE ZOBJSTAT
```

3. Save your changes.

Enabling HTTPS Routing using SSL

The HP-OpenView Adapter for SSL updates the RMS SSL Configuration Parameters in a Messaging Server RMS.CFG file. Once these configuration parameters are available, you can modify the sections in the RMS.CFG and the various DDA.CFG files that route data using HTTP to route data using HTTPS.

To enable secure HTTPS routing using SSL

1. First update your Messaging Server to version 3.0 and then run the HP-OpenView Adapter for SSL using Radia (Adapter for SSL) to install SSL support for it. Refer to *Adapter for SSL Installation and Configuration Guide* for more information.

The Adapter for SSL will update the following section of the RMS.CFG file with the appropriate certificate and HTTPS port information:

```
#-------------------------------------------------------------
#        RMS SSL Configuration Parameters
#-------------------------------------------------------------
Overrides Config {
   SSL_CERTFILE  "   "
   SSL_KEYFILE   "   "
   HTTPS_PORT    "   "
   }
```

The HTTPS_PORT is the secure port that the Messaging Server uses to receive messages.

2. Edit the sections of the RMS or DDA configuration files currently defined to route data with a TYPE of HTTP. Change the TYPE from HTTP to **HTTPS**, and modify the URL address to include **https:** as well as the **SSL_port_number** for the server receiving the message**.**

For example, the following entry in the CORE.DDA.CFG file routes data to the RMP using HTTP:

```
msg::register rmp {
    TYPE          HTTP
```

```
    ADDRESS     {
        PRI     10
        URL     http://RMP_host:3466/proc/xml/obj
    }
}
```

After enabling SSL, the following modifications to the same section allow for HTTPS routing of RMP objects to an RMP server using a secure port of 443:

```
msg::register rmp {
    TYPE        HTTPS

    ADDRESS     {
        PRI     10
        URL     https://RMP_host:443/proc/xml/obj
    }
}
```

3. Apply the same modifications to any section of the RMS or DDA configuration files being used to route data using HTTP.

4. Save the changes, and restart the Messaging Server service.

Using Store and Forward Configurations (to Place Objects Close to a SQL Compliant Database)

As previously mentioned, the RMS install program will not install a Messaging Server fully configured for store and forward capabilities. Following the installation, you need to edit the appropriate configuration files and switch the routing options to forward messages to another Messaging Server.

For details, refer to the Store and Forward Configuration topics in the *Messaging Server Guide*.