

# HP OpenView Messaging Server Using Radia

for the AIX, HP-UX, Linux, Solaris and Windows operating systems

Software Version: 3.2

---

## Migration Guide

## 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

© Copyright 2006 Hewlett-Packard Development Company, L.P.

No part of this document may be copied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard Company. The information contained in this material is subject to change without notice.

### 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.

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

TFTP SERVER  
Copyright © 1983, 1993  
The Regents of the University of California.

OpenLDAP  
Copyright 1999-2001 The OpenLDAP Foundation, Redwood City, California, USA.  
Portions Copyright © 1992-1996 Regents of the University of Michigan.

OpenSSL License  
Copyright © 1998-2001 The OpenSSLProject.

Original SSLeay License  
Copyright © 1995-1998 Eric Young (eay@cryptsoft.com)

DHTML Calendar  
Copyright Mihai Bazon, 2002, 2003

## 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](http://www.hp.com/managementsoftware/access_level)**

To register for an HP Passport ID, go to:

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

## Documentation Updates

This manual's title page contains the following identifying information:

- Version number, which indicates the software version
- Document release date, which changes each time the document is updated

To check for recent updates or to verify that you are using the most recent edition, visit the following URL:

**[http://ovweb.external.hp.com/lpe/doc\\_serv/](http://ovweb.external.hp.com/lpe/doc_serv/)**

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

# Contents

- 1 Introduction .....7
  - About this Guide.....7
  - Who this Guide is for .....7
  
- 2 Radia Inventory Database Prerequisites .....9
  - Introduction .....9
  - Updates for Inventory Manager Database Version 3.x .....9
  - Updates for Inventory Manager Database Version 4.x .....9
    - Implementation Notes for Tables 1 and 2.....12
    - Reasons for Modifications in Tables 1 and 2 .....12
  
- 3 Upgrading to Messaging Server 3.2 ..... 13
  - Upgrading from Messaging Server 3.x.....13
  - Upgrading from Messaging Server 2.x.....15
    - Upgrading your RCS ZTASKEND REXX Method .....16
      - About ZTASKEND Version 1.12.....16
    - Upgrading a Messaging Server from 2.x to 3.2.....17
    - Migrating Custom SQL Code from an Inventory Manager Server.....19
      - About the Scripts and SQL Queries used with the Data Delivery Agents .....19
    - Verifying the Patch Method Connection and Queue Name .....21
    - Enabling HTTPS Routing using SSL .....21
    - Using Store and Forward Configurations (to Place Objects Close to a SQL Compliant Database)....23



# 1 Introduction

## About this Guide

### Who this Guide is for

This migration guide is for Radia system administrators who want to upgrade:

- an *HP OpenView Messaging Server Using Radia (Messaging Server)* environment to Version 3.2.
- an existing *Inventory Manager Database schema* to support this version of the Messaging Server with Patch Manager 3.0 and Reporting Server 4.2.

This guide contains information for the Windows and UNIX platforms.

You should be familiar with Radia products, such as the Radia Configuration Server, the Configuration Server Database and Inventory Database, and Radia methods such as ZTASKEND.

If using Patch Manager and Reporting Server, you should be familiar with those products. For details, see the appropriate guides.





# 2 Radia Inventory Database Prerequisites

## Introduction

If you are using the Messaging Server 3.2 to process Inventory data, it is important that you update your Inventory Database schema to accommodate this release, which supports Patch Manager 3.0 and Reporting Server 4.2.

A summary of updates to the Inventory Manager Database schema that have been made since Inventory Manager 3.x are documented in this chapter.

### Updates for Inventory Manager Database Version 3.x

If not already completed, make the following changes to an existing Radia Inventory Manager database to support this release of Messaging Server, Patch Manager, and Reporting Server. Make these changes whether you are using an Oracle or SQL Server database.

These changes are required.

- 1 The **AppEvent** table requires two new columns: **product** and **errornum**. Either delete the table so it is created with the necessary columns by the new Messaging Server install, or modify the Table definition of **AppEvent** by adding the following columns:

Column Name	Data Type	Length	Allow Nulls
product	varchar	10	√
errornum	varchar	10	√

- 2 The **DeviceConfig** table requires a new column: **devicename**. Either delete the table so it is created with the new column by the Messaging Server install, or 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	√

- 2 Continue by reviewing the updates needed for Inventory Manager Database Version 4.x, below.

### Updates for Inventory Manager Database Version 4.x

Tables 1 and 2 identify the updates to an existing Radia Inventory database that were implemented since Messaging Server 3.0 and Inventory Manager 4.0 to support this release of the Messaging Server.

*Implementation Notes and Reasons for Modifications* begin on page 12.

- ▶ Messaging Server v3.2 Build 100 incorporates all changes in Table 1 and those identified as incorporated in Table 2.

**Table 1 Inventory Database Changes from Messaging Server 3.0 and Inventory Manager 4.0/4.1**

<b>RMS 3.0 DDA and Build</b>	<b>RIM Builds</b>	<b>Table</b>	<b>Column</b>	<b>Modification (Add, Change or Delete)</b>	<b>Filename, Version and Date</b>
Wbem Build 1	4.0/404	rNVD_INSTALLED_UNINSTALL	wLanguage	(C) From int to varchar. See Implementation Note 1. See Reason 1.	nvd_installed_uninstall.sql v 1.4.8.2 2004/07/29
	4.0/404	FileAudit	fqsvcid	(A) To support Reporting Server See Implementation Note 2.	fileaudit.sql v 1.5 2004/04/08
	4.0/404		fqsvcid	(A) To support Reporting Server See Implementation Note 2.	filepost.tcl v 1.3 2004/04/13
Core Build 1	4.0/409	AppEvent	del_date	(C) Type from datetime to varchar(30). See Implementation Note 1. See Reason 2.	apps.services.sql v 1.9.2.1 2005/03/04
Core Build 1	4.0/409	HAppEvent	del_date	(C) Type from datetime to varchar(30). See Implementation Note 1. See Reason 2.	apps.services.sql v 1.9.2.1 2005/03/04
Inventory Build 2	4.0/411 4.1/501	FileAudit	name	(C) Size increased from 50 to 255. See Implementation Note 1. See Reason 3.	fileaudit.sql v 1.10 2005/06/16
Inventory Build 2	4.0/411 4.1/501	FileAudit	scanfor	(C) Size increased from 50 to 255. See Implementation Note 1. See Reason 3.	fileaudit.sql v 1.10 2005/06/16
Wbem Build 2	4.0/411 4.1/501	rRegistry		(A) New file to create a Registry table for the new Registry class	registry.sql v 1.2 2005/06/13

*Implementation Notes and Reasons for Modifications begin on page 12.*

**Table 2 Inventory Database Changes from Messaging Server 3.1 and Inventory Manager 4.0/4.1**

<b>RMS 3.1 DDA and Build</b>	<b>RIM Builds</b>	<b>Table</b>	<b>Column</b>	<b>Modification (Add, Change or Delete)</b>	<b>Filename, Version and Date</b>
Wbem Build 10	4.0/411 4.1/501	rNVD_INSTALLED_UNINSTALL	wInstallSource	(C ) Size increased from 128 to 255. Customer was receiving a string in excess of 128 bytes. See Implementation Note 1.	nvd_installed_uninstall.sql v 1.6.2.2 2005/09/12
Wbem Build 10	4.0/411 4.1/501	rNVD_INSTALLED_UNINSTALL	wVersionMajor	(C ) From int to varchar(10). Customer returned 3.0.0 for wVersionMajor See Implementation Note 1.	nvd_installed_uninstall.sql v 1.6.2.2 2005/09/12
Wbem Build 10	4.0/411 4.1/501	rNVD_INSTALLED_UNINSTALL	wVersionMinor	(C ) From int to varchar(10). Customer returned 3.0.0 for wVersionMajor. See Implementation Note 1.	nvd_installed_uninstall.sql v 1.6.2.2 2005/09/12
Core Build 11	4.0/412 4.1/502	DeviceConfig	memory	(C) From int to varchar(32). Needed to support a blank string value.	device.config.sql v 1.7.2.1 2005/11/03
RMS Build 64 Core Build 12	4.0/412 4.1/503	DeviceNotify		(C) Datatypes from INTEGER to INT. See Reason 4.	notify.sql v 1.5.2.2 2005/12/20
RMS Build 64 Core Build 12	4.0/412 4.1/503	DeviceZRState	behabi (now behai)	(C ) Column name from behabi to behai. Corrected misspelling. See Implementation Note 2.	device.zrstate.sql v 1.2.2.1 2005/12/21
RMS 3.2 Build 100 Data Delivery Agents Build 20 incorporate all changes above this line. RMS 3.2 Build 101 Data Delivery Agents Build 21 will incorporate all changes below this line.					
RMS Build 64 Core Build 12	4.0/412 4.1/503	DeviceZRStates	behabi	(C ) Column name from behabi to behai. Corrected misspelling. See Implementation Note 2.	device.zrstates.sql v 1.1.4.2 2006/01/03
RMS Build 64 Core Build 12	4.0/413 4.1/504	HJOBSTAT		(C ) Calls apps.jobstat.hupdate first before apps.jobstat.archive. See Reason 5.	apps.jobstat.sql v 1.4 2005/12/13
RMS Build 64 Core Build 12	4.0/413 4.1/504	HAppMSIEvent		(C ) Calls apps.msiservices.hupdate before apps.msiservices.archive. See Reason 5.	apps.msiservices.sql v 1.4 2005/12/13
RMS Build 64 Core Build 12	4.0/413 4.1/504	HAppRNPEvent		(C ) Calls apps.rnpservices.hupdate before apps.rnpservices.archive. See Reason 5.	apps.rnpservices.sql v 1.3 2005/12/13
RMS Build 64 Core Build 12	4.0/413 4.1/504	HAppEvent		(C ) Calls apps.services.hupdate before apps.services.archive. See Reason 5.	apps.services.sql v 1.3 2005/12/13

<b>RMS 3.1 DDA and Build</b>	<b>RIM Builds</b>	<b>Table</b>	<b>Column</b>	<b>Modification (Add, Change or Delete)</b>	<b>Filename, Version and Date</b>
RMS Build 64 Core Build 12	4.0/413 4.1/504	HDeviceConfig		(C ) Calls device.config.hupdate before device.config.archive. See Reason 5.	device.config.sql v 1.9 2005/12/12
RMS Build 64 Core Build 12	4.0/413 4.1/504	HDeviceErrors		( C ) Calls device.errors.hupdate before device.errors.archive. See Reason 5.	device.errors.sql v 1.9 2005/12/12
RMS Build 64 Core Build 12	4.0/413 4.1/504	HDeviceState		(C) Calls device.status.hupdate before device.status.archive. See Reason 5.	device.state.sql v 1.4 2005/12/12
RMS Build 64 Core Build 12	4.0/413 4.1/504	HDeviceStatus		(C ) Calls device.status.hupdate before device.status.archive. See Reason 5.	device.status.sql v 1.6 2005/12/13

## Implementation Notes for Tables 1 and 2

- 1 If implementing this change is not possible without a loss of data in your pre-existing table, HP recommends holding off the implementation until you experience the error identified as the reason for modification.
- 2 If implementing this change is not possible without a loss of data in your pre-existing table, HP knows of no alternative.

## Reasons for Modifications in Tables 1 and 2

- 1 rNVD\_INSTALLED\_UNINSTALL wLanguage  
Customer had Language returned as en\_US.
- 2 AppEvent and HAppEvent del\_date  
For SQL database, error writing a blank to the del\_date column (SQL inserted blank as 1900-01-01 00:00:00). Setting date type to varchar keeps the value as a blank. Oracle databases do not require this change.
- 3 FileAudit name and scanfor columns  
File Audit functions changed from Radia 3.x to Radia 4.x. Radia 4.x functionality was changed to insert the scanfor value when a file is NOT FOUND in the name column appended to a string "No File found for scan" This increases the length of the string and could exceed the current default length of 50 characters. The previous version of the Radia client (2.x, 3.x) inserted the friendly name in the name column, but that was problematic because the friendly name was not always intuitive.
- 4 DeviceNotify table change  
This new notify.sql file only needs to be implemented when creating the DeviceNotify table for the first time using NVDKIT built with TCL 8.4. If the DeviceNotify table already exists there is no need to implement this fix.
- 5 These changes were made to correct unique constraint violation errors. Implementing the changes is optional. Failure to implement will retain previous behavior. To implement these changes:
  - Prior to running the Messaging Server install, delete the sql files identified in the filename column of Table 2. This forces the new sql files to be unpacked.

## 3 Upgrading to Messaging Server 3.2

Use the following procedures to upgrade an existing Messaging Server 3.x or 2.x environment to a Messaging Server 3.2 environment.

▶ Patch Manager 3.0 requires you to delete an existing `patch.dda.cfg` file, as discussed in [Upgrading from Messaging Server 3.x](#) below.

⚠ If your Messaging Server is processing Inventory Manager data and you do not have an existing Reporting Server installed, you must also upgrade your Inventory Manager Database schema for this release. See [page x](#) for details.

### Upgrading from Messaging Server 3.x

To apply [Messaging Server 3.2](#) to an existing [Messaging Server 3.x](#) installation

Messaging Server 3.2 requires the Data Delivery Agents delivered with Version 3.2, and vice versa. Be sure to re-install each Data Delivery Agent that was previously installed so that the Data Delivery Agent modules are updated as well as the Messaging Service module.

- 1 Stop the Radia Messaging Server (RMS) service (RMS.TKD).

▶ Stopping the RMS service automatically stops message processing for each Data Delivery Agent queue.

- 2 Create a backup of the directory where your Messaging Server is installed.
- 3 Delete the following subdirectories from the `etc` directory of where the Messaging Server is installed:

```
/etc/core/sql  
/etc/core/lib  
/etc/inventory/sql  
/etc/inventory/lib  
/etc/wbem/sql
```

The installation program will upgrade the Messaging Server and Data Delivery Agent modules, but does not replace the associated configuration files with the new ones.

- 4 To support Patch Manager version 3.0, delete the following file from the `etc` folder of where the Messaging Server was installed:

```
patch.dda.cfg
```

This is required to obtain the new `patch.dda.cfg` file, which contains a newly-defined column for ZOBJMSGI.

- 5 Optionally, delete an existing configuration file prior to running the `install` to obtain the newest default configuration for it:

```
rms.cfg  
core.dda.cfg  
inventory.dda.cfg
```

wbem.dda.cfg

Run the install for the Messaging Server, making sure to select the installation of all existing data delivery agents.

- 6 Following installation, reapply any customizations to the \*.SQL and \*.TCL files located in the \etc\core, \etc\inventory and \etc\wbem directories.



The Data Delivery Agents for Messaging Server 3.2 unpack the default versions of the .sql files into subdirectories named \etc\

- 7 If the Core Data Delivery Agent has been configured to post objects to the Management Portal, an additional configuration is recommended to re-queue the CORE.RMP messages into a specialized queue named rmpq, and then route them to the Management Portal. This change will allow throttling of the flow of messages to the Management Portal so that it doesn't get overloaded.

If you did not delete your existing *core.dda.cfg* file prior to running the installation, use the steps below to modify it to match the *core.dda* configuration delivered with this release. This configuration re-queues only the Management Portal data into an rmpq queue.

- a Use a text-editor to open the *core.dda.cfg* file. Check the `msg::register corerouter` section for the routing of CORE.RMP messages.
- b Change the routing of the CORE.RMP messages to use the rmpq identifier as specified in the following configuration:

```
ROUTE      {
    TO      CORE.RMP
    USE     rmpq
}
```

- c Add the following configuration sections to register a new QUEUE section (rmpq), ROUTER section (rmpqrouter) and HTTP section (rmpqhttp) to the bottom of the *core.dda.cfg* file.

```
msg::register rmpq {

    TYPE      QUEUE

    DIR      ../ConfigurationServer/data/rmp
    USE      rmpqrouter

    POLL     10
    COUNT    30
    DELAY    3600
    ATTEMPTS 200
}
```

```

msg::register rmpqrouter {
    TYPE          ROUTER

    ROUTE        {
        TO        CORE.RMP
        USE       rmpqhttp
    }
}

msg::register rmpqhttp {
    TYPE          HTTP

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

```

- d In the URL configuration parameter in the HTTP section, replace the given value of `localhost` with the IP address of the server hosting the Management Portal.

#### 8 Restart the RMS service.

This completes the steps to apply Messaging Server Version 3.2 to an existing 3.x installation.

## Upgrading from Messaging Server 2.x

This release contains a complete install of the Messaging Server service (`RMS.TKD`). The `setup.exe` file used to install the product can also be used to upgrade it from a Messaging Server 2.x environment—without additional configuration steps.

Use the topics that follow to:

- Upgrade the RCS ZTASKEND REXX method to version 1.12. Optionally, port any custom code to the new ZTASKEND REXX.
- Upgrade a Messaging Server from 2.x to 3.2. Messaging Server version 3.2 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.
  - DDA posting of objects using ODBC to multiple DSNs. For details, refer to the appendices of the *Messaging Server Guide*.

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

### About ZTASKEND Version 1.12

Version 1.12 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.x routing options available with the RMS Data Delivery Agents (which use multiple data queues).

Since ZTASKEND REXX is frequently customized, there are comments in the ZTASKEND v1.12 code as well as in the ZTASKEND discussion in the *Installation and Configuration Guide for the HP-OpenView Messaging Server Using Radia, v 3.2, (Messaging Server Guide)* that point out where and how to customize it.



The ZTASKEND REXX version 1.12 is delivered on the Radia Infrastructure 4.2 CD with a temporary file extension of `.rex` (ZTASKEND.`rex`). You must remove the file extension in order to execute it; this step is included in the upgrade procedures below.

#### To upgrade your version of ZTASKEND REXX to version 1.12

- 1 Hewlett-Packard recommends stopping the RCS service prior 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 Access the Radia 4.2 Infrastructure CD and locate the new ZTASKEND, version 1.12. Its location and filename are given below.

```
\extended_infrastructure\messaging_server\migrate\ZTASKEND.rex
```

- 3 Place a copy of the delivered version of the new ZTASKEND, ZTASKEND.`rex`, on your RCS platform in the `\rex\Novadigm` directory, and then remove the file extension of `.rex`. Additional details on performing this step are given below.
  - a Rename the current ZTASKEND script in the `\rex\Novadigm` directory to **ZTASKEND.old**.
  - b Copy ZTASKEND.`rex` from the `\migrate` folder listed in Step 2 to the `\rex\Novadigm` directory of your RCS.
  - c Rename ZTASKEND.`rex` in the `\rex\Novadigm` directory to **ZTASKEND** with no extension.
  - d If you do *not* use a customized version of ZTASKEND, go to Step 5.



- 4 If you are using a customized ZTASKEND, it will be located in the `\rexx` directory on your RCS. This step guides you in porting your customizations from a current ZTASKEND to the new ZTASKEND, and then enabling the new, customized ZTASKEND for execution by removing the file extension.
  - a Rename your current ZTASKEND script in the `\rexx` directory to **ZTASKEND.old**.
  - b Copy `ZTASKEND.rex` from the `\migrate` folder listed in Step 2 to the `\rexx` directory on your RCS.
  - c Review the comments throughout the new `ZTASKEND.rex` file and compare it with the customizations in your `ZTASKEND.old` file. Also refer to the discussion on ZTASKEND found in the *Messaging Server Guide*, which highlights the code sections most frequently used for customizations.
  - d Port any customizations to the new ZTASKEND, `ZTASKEND.rex`, in the `\rexx` directory. Document your changes.
  - e Rename `ZTASKEND.rex` to **ZTASKEND** *with no extension*.
- 5 Restart the RCS service, and continue with the Messaging Service upgrade. See *Upgrading a Messaging Server from 2.x to 3.2* below.

## Upgrading a Messaging Server from 2.x to 3.2

Use these steps to install version 3.2 of the Messaging Server to the same location as an existing Messaging Server Version 2.x. The install media is provided on the Radia 4.2 Infrastructure CD.



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.

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 3.2 (Windows and UNIX)

HP recommends performing this upgrade after you have upgraded the ZTASKEND method on the RCS to version 1.12. See the earlier procedures on page 16.

- 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:

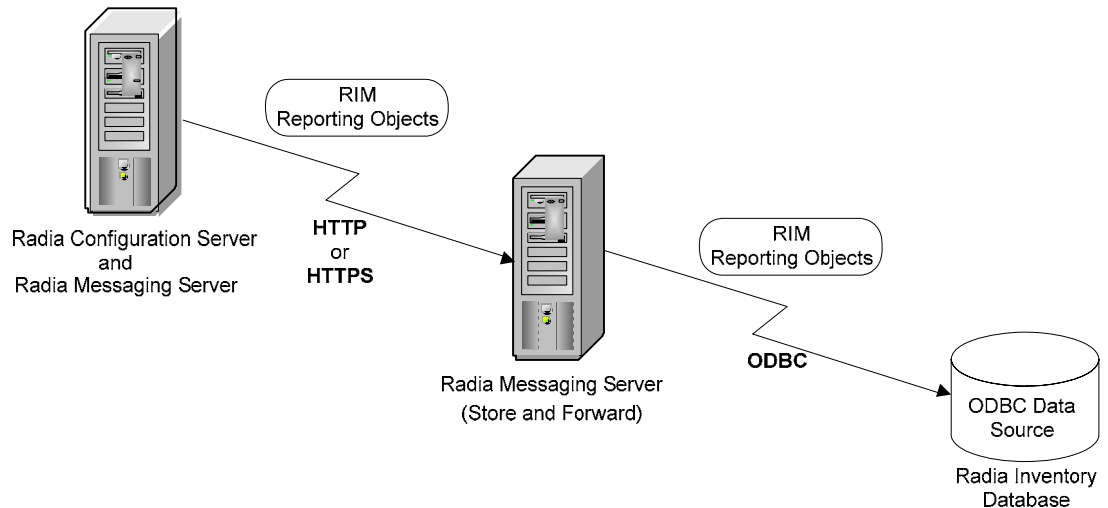
```
./install
```

and press **Enter**.

- 2 Follow the prompts to complete the installation. Choose the same install location as your existing 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 an Inventory Manager Server on page 19.

### HP-Recommended Best Practices for RMS 3.x

- 1 The default WORKER setting for each configuration file is 1 worker, and HP recommends using this setting with Messaging Server 3.x (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.



- 3 Whenever possible, HP recommends using the default configuration for STARTUPLOAD (set to 0) in the COREODBC and WBEMODBC sections of the CORE, INVENTORY and WBEM Data Delivery Agents. STARTUPLOAD controls when SQL tables are created and the defined SQL commands are loaded into memory. The STARTUPLOAD default value of 0 performs these SQL activities *upon the first message delivery*. This configuration allows only the commands needed by each message type to be loaded into memory and represents a more efficient use of resources. Only set STARTUPLOAD to 1 when it is necessary to create the SQL tables *upon Messaging Server and Data Delivery Agent startup* due to table-creation limitations set by a Database Administrator.
- 4 This allows SQL table creation and SQL commands to be  
Refer to the *Messaging Server Guide* for details on how to configure store and forward servers in your environment.

## Migrating Custom SQL Code from an Inventory Manager 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 20 can be copied from their locations on your Inventory Manager Server to the equivalent locations on the Messaging Server.

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

- ▶ 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 (as delivered from HP) are contained in the `/etc/<module name>/sql/hp` directories. However, custom versions of these `.sql` files are to be placed in the `/etc/<module name>/sql` directories; this means the customized versions will be executed instead of the HP-delivered versions placed in the lower level `hp` subdirectories.

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 below 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**

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

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.

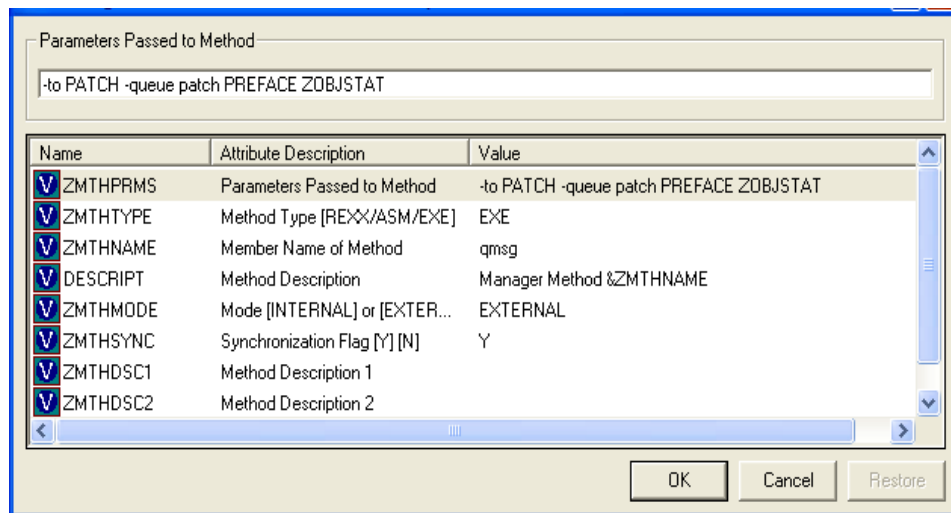
## Verifying the Patch Method Connection and Queue Name

- 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 the System Explorer to edit the ZMTHPRMS attribute of the PRIMARY.SYSTEM.ZMETHOD.PATCH\_STATUS instance, as shown in Figure 1 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 1 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 3.2 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*.

