

HP Client Automation Enterprise

Messaging Server

for the HP-UX, Linux, Solaris, and Windows operating system

Software Version: 7.80

Migration Guide

Document Release Date: November 2009

Software Release Date: November 2009



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 1993-2009 Hewlett-Packard Development Company, L.P.

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.

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

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
 - The number before the period identifies the major release number.
 - The first number after the period identifies the minor release number.
 - The second number after the period represents the minor-minor release number.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

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

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport login page.

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

Table 1 Document Changes

Chapter	Version	Changes
All	7.80	Included information about migrating to version 7.80.
All	7.20	The HP Configuration Management products are newly named HP Client Automation for 7.20. Updated product names throughout the guide.
Documentation Updates	7.20	Updated URL for manuals.
Chapter 2	7.20	Page 13, renamed chapter Upgrading to Messaging Server 7.80 , and reorganized topics.
All	7.80	Updated for migration to HPCA version 7.80.
Chapter 2	7.50	Reorganized chapter to itemize tasks needed when migrating: <ul style="list-style-type: none">• from Versions 2.x or 3.x (Radia 4.x)• from Versions 5.x or 7.20 Classic• from Version 7.20 on Core and Satellites
Chapter 2	5.10	Page 21, Corrected Oracle syntax in the procedures to update Inventory Database from Version 5.10.
Appendix A	7.20	Page 27, Inventory Database Schema for Version 7.20 adds one script and table: win32_physicalmemory.sql creates the rWin32_PhysicalMemory table.
Appendix B	7.50	Page 35, Added Appendix with database migration topics that apply only when migrating from Radia 4.x.

Support

Visit the HP Software Support web site at:

www.hp.com/go/hpsoftwaresupport

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

HP Software online 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 valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into 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 sign in. Many also require a support contract. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Contents

1	System Requirements.....	7
2	Migration Overview and Prerequisites.....	8
	Using this Guide.....	8
	Prerequisites.....	8
	What Changed?.....	8
	How Do I Upgrade my Messaging Server and Database Schema?.....	8
	How Do I Determine the Current Version?.....	9
	Summary of Changes between Versions.....	10
	Changes Between Version 2.x or 3.x (Radia 4.x) and Version 5.00.....	10
	Changes Between Version 5.00 and 5.10.....	11
	Changes Between Version 5.10 and 7.20.....	11
	Changes Between Version 7.20 and 7.50.....	12
	Changes Between Version 7.5x and 7.80.....	12
3	Upgrading to Messaging Server 7.80.....	13
	Overview.....	13
	To Migrate from Versions 5.x or 7.x Classic.....	13
	To Migrate from Versions 2.x or 3.x (Radia 4.x).....	14
	Task Details.....	15
	Task 1: Upgrade your Configuration Server ZTASKEND REXX Method.....	15
	Task 2a: Convert Database to Unicode.....	15
	Task 2b: Run Database Script if Migrating from 5.00 Unicode.....	15
	Task 3: Apply Inventory Manager Database Schema Changes (Both SQL Server or Oracle-hosted Databases).....	15
	Schema changes from 7.50 to 7.80.....	16
	Schema changes from 7.20 to 7.50.....	16
	Procedures.....	16
	Schema changes from 5.10 to 7.20.....	17
	Procedures.....	18
	Schema Changes from 2.x, 3.x 5.00 to 5.10.....	20
	Procedures.....	21
	Task 4a: Upgrading from Messaging Server 2x or 3.x.....	22
	Task 4b: Upgrading from Messaging Server 5.x or 7.x.....	23
	Task 5: Review Post-Installation Configurations for Patch Manager.....	24
	Patch Manager: Set the DBTYPE for a Patch ODBC Database on Oracle.....	24
	Verify the Patch Method Connections and Queue Name.....	24
	Task 6: On UNIX platforms, Configure the HP-Supplied Datadirect Connect ODBC Drivers.....	26
	Task 7: Upgrade and Reconfigure Store and Forward Messaging Servers (to Place Objects Close to a SQL-compliant Database).....	26

A Inventory Manager Database – Tables and Scripts	27
B Additional Database Migration Topics	35
Database Conversion to Unicode Datatypes (Required)	35
Why convert the database for Unicode support?.....	35
About the SQL Server Migration Scripts	36
Possible Unicode Migration Issues and Resolution Options	36
SQL Server Conversion for Unicode	38
Oracle Conversion for Unicode.....	39
Migrating Custom SQL Code from an Inventory Manager Server	43
About the Scripts and SQL Queries used with the Data Delivery Agents.....	43

1 System Requirements

HP Client Automation (HPCA) version 7.80 supports those database servers listed in the *HPCA Enterprise Edition Release Notes*.

If your HPCA environment uses SQL Server, you must use either SQL Server 2005 or SQL Server 2008 with HPCA version 7.80.

If you are using an older version of SQL Server, be sure to upgrade your database to either SQL Server 2005 or SQL Server 2008 before you begin the HPCA migration process described in this document. Refer to your SQL Server documentation for instructions.

2 Migration Overview and Prerequisites

Using this Guide



This document pertains only to an HPCA Classic installation. If you are migrating from a version 7.x HPCA Core and Satellite environment, refer to the *HPCA Enterprise Edition Core and Satellite Migration Guide*.

Use the information in this guide to upgrade your 4.2x, 5.x, or 7.x HP Client Automation Messaging Server (HPCA Messaging Server) on Windows and Unix/Linux platforms to version 7.80.

The migration also covers the schema changes needed to update your SQL Server or Oracle database used by Inventory Manager, Enterprise Manager, and Reporting Server for reports.

Prerequisites

To perform an upgrade of your Messaging Server, you should be familiar with these HPCA infrastructure products or components:

- Configuration Server and Configuration Server Database and methods such as ZTASKEND.
- The SQL Server or Oracle database used by the Inventory Manager application and Reporting Server.
- If you are using Patch Manager, you should be familiar with that product and corresponding SQL Server or Oracle database.
- If you are using Enterprise Manager, you should be familiar with that product.

For details, see the appropriate guides for each product.

What Changed?

To review what changed between your existing version and the current one, see [Summary of Changes between Versions](#) on page 10.

How Do I Upgrade my Messaging Server and Database Schema?

To begin the tasks needed to upgrade your Messaging Server and database schema, verify your current Messaging Server version (see [How Do I Determine the Current Version?](#) on page 9), and select the topic that applies:

- [To Migrate from Versions 5.x or 7.x Classic](#) on page 13
- [To Migrate from Versions 2.x or 3.x \(Radia 4.x\)](#) on page 14

How Do I Determine the Current Version?

On the system hosting the HPCA Messaging Server, follow these steps to determine the version:

- 1 In a command line window, go to the Messaging Server installation directory. For example, the default installation directory for version 7.2x Classic on Windows is:

```
C:\Program Files\Hewlett-Packard\CM\MessagingServer
```

- 2 Run the following command:

```
nvdkit version rms.tkd
```

The output should look similar to this:

```
rms.tkd
Kit Version:      8.4.13
Kit Build:       481
Kit Encoding:    cpl252

Build Date:      20090818 04:59:20 UST
Build Platform: Windows NT 5.2
Build Machine:   intel
Build Host:      ftc-ca-winbldv7
Build User:     nvdkit

C:/Program Files/Hewlett-Packard/HPCA/MessagingServer/rms.tkd:
  module rms  version 7.80.000  build 7802  20090826 12:52:45 UST
```

Summary of Changes between Versions

Use the topics below to understand what changed between your existing Messaging Server version and version 7.80 for the server and database schema, scripts, and drivers. The changes are cumulative, and the tasks to upgrade your Messaging Server and schema will reflect this.

- [Changes Between Version 2.x or 3.x \(Radia 4.x\) and Version 5.00](#) below
- [Changes Between Version 5.00 and 5.10](#) on page 11
- [Changes Between Version 5.10 and 7.20](#) on page 11
- [Changes Between Version 7.20 and 7.50](#) on page 12
- [Changes Between Version 7.5x and 7.80](#) on page 12

Changes Between Version 2.x or 3.x (Radia 4.x) and Version 5.00

If you are upgrading from version 2.x or 3.x (Radia 4.x), these changes were introduced as of version 5:00:

- The default installation path changed to:

Windows: C:\Program Files\Hewlett-Packard\CM\MessagingServer

UNIX: /opt/HP/CM/MessagingServer

If desired, the default path can be changed during the installation process to point at an existing Messaging Server path.

- This version adds several scripts needed to generate standard tables for `wbem/cim`.
- The `nvdkit` for this release supports database character sets needed for multiple-language support, such as the `nvarchar` datatype for SQL Server and `nvarchar2` datatype for Oracle. To take advantage of these new datatypes the backend database must be converted. Scripts are included with this release to perform the database conversion.
- Datadirect Connect ODBC drivers for the supported Unix and Linux platforms will be installed by default. These ODBC drivers will allow data posting to a database directly from these platforms.

The Messaging Server 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 five available Data Delivery Agents: CORE, INVENTORY, WBEM, PATCH and USAGE The Data Delivery Agents can be used to post data to a SQL Server or Oracle database.
- Add the same scripts and `*.sql` code to the Messaging Server that is provided with the (now retired) Inventory Manager Server for creating the SQL tables and to modify the data in the Inventory database.



The migration procedures include a post-install task of relocating any custom SQL code from your previous Inventory Manager Server to your Messaging Server.

The Messaging Server 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 Appendix A of the *HPCA Messaging Server Installation and Configuration Guide*

Changes Between Version 5.00 and 5.10

These changes were introduced between version 5.00 and version 5.10:

- The Messaging Server install includes an optional selection for a Usage Manager DDA. This Data Delivery Agents offers support for the collation of Application Usage Manager files. Refer to the *Application Usage Manager User Guide* for more information.
- Two additional scripts are provided to migrate an Inventory Manager Database to Unicode. If you already modified your database to Unicode for version 5.00, you need to apply the last two steps.

Refer to [SQL Server Conversion for Unicode](#) on page 38 or [Oracle Conversion for Unicode](#) on page 39. These required database changes apply to all customers upgrading from version 5.00.

- Patch Manager 5.10 introduced an additional reporting object: PASTATUS. If your CM Messaging Server uses a 'Patch Message Directory to Scan' queue that is not named patch, refer to [Verify the Patch Method Connections and Queue Name](#) on page 24.

Changes Between Version 5.10 and 7.20

These changes were introduced between version 5.10 and version 7.20:

- **HP Client Automation Rebranding:** As of version 7.20, the Messaging Server has been renamed from the HP Configuration Management Messaging Server to the HP Client Automation (HPCA) Messaging Server. The default installation paths remain the same:

Windows: C:\Program Files\Hewlett-Packard\CM\MessagingServer

UNIX: /opt/HP/CM/MessagingServer

- **SQL Schema Changes:** Several tables in the Inventory SQL Database (Core and Inventory tables) require additional columns in order to support Client Automation 7.20 features, such as Vulnerability Management and Thin Client support. There are also new tables.

Prior to upgrading the Messaging Server, all customers must apply these table changes to the Inventory Manager Database. Refer to [Schema changes from 5.10 to 7.20](#) on page 17.

- **Vulnerability Support embedded in Core.DDA:** The Messaging Server and the core.dda from this Version 7.20 release are required for Vulnerability Management, as well as the latest ZTASKEND REXX version on the HPCA 7.20 media; Vulnerability Management uses the same ODBC DSN connections as the Core.DDA to post a device's OVAL-based security and vulnerability status to the appropriate Inventory Database tables. Refer to the *HP Client Automation Enterprise Manager User Guide* for more information on obtaining Vulnerability Reports for your managed-devices.

Changes Between Version 7.20 and 7.50

These changes were introduced between version 7.20 and version 7.50:

- **SQL Schema Changes:**

- Added new wbem table: rMSSD_AttributeData
 - Added new tables to support Remote Control Auditing (see [Inventory Manager Database – Tables and Scripts](#) on page 27). These are created automatically and do not require migration.
 - New columns:
 - scanner added to VM_ACQUISITION_FILE
 - cjobid added to APPEVENT, HAPPEVENT and DEVICESYNOPSIS
 - jobid added to APPEVENT, HAPPEVENT
 - Modified column:
 - jobid changed to length 64 in DEVICESYNOPSIS
- HPCA 7.50 now requires a Unicode database. If you have an existing database that was created prior to 5.00 and not previously converted to Unicode, it must be converted at this time.
 - SQL Server versions 2005 and 2008 are the only supported versions of SQL Server for HPCA version 7.50 and later.

Changes Between Version 7.5x and 7.80

No schema changes were introduced between version 7.5x Classic and 7.80 Classic.

SQL Server versions 2005 and 2008 are the only supported versions of SQL Server for HPCA version 7.80.

3 Upgrading to Messaging Server 7.80

Overview

Use the following procedures to upgrade from an existing Radia Messaging Server 2.x, 3.x, Configuration Management Messaging Server 5.x, or HP Client Automation 7.x environment to an HP Client Automation Messaging Server 7.80 environment.



This document pertains only to an HPCA Classic installation. If you are migrating from a version 7.x HPCA Core and Satellite environment, refer to the *HPCA Enterprise Edition Core and Satellite Migration Guide*.

The Messaging Server 7.80 release is a drop-in replacement for previously released versions of the Messaging Server that have Data Delivery Agent (.dda) support. The upgrade allows you to use existing configuration files and any customized scripts that you have for mapping client object data into backend databases.

Based on your current Messaging Server version, see the following:

- [To Migrate from Versions 5.x or 7.x Classic](#) on page 13
- [To Migrate from Versions 2.x or 3.x \(Radia 4.x\)](#) on page 14

To Migrate from Versions 5.x or 7.x Classic

Perform these tasks to migrate your Messaging Server that is co-located with the Configuration Server.



Use Tasks 2a, **2b** and **4b** as directed below.

Table 1 Task List for Migration from 5.x or 7.x

Task	Page	Description
Task 1	15	Upgrade the ZTASKEND REXX method.
Task 2a	15	Convert an existing non-Unicode database to Unicode.
Task 2b	15	<i>If you are migrating from version 5.00 Unicode</i> , run the database conversion script. This task applies only if you are migrating from version 5.00 and had previously converted your database to Unicode.
Task 3	15	Apply Database schema changes.
Task 4b	23	Upgrade from Messaging Server 5.x or 7.x.
Task 5	24	Review Post-Installation Configurations for Patch Manager.
Task 6	26	On UNIX platforms, configure the DataDirect ODBC Drivers.
Task 7	26	Upgrade any store and forward Messaging Servers in your environment.

To Migrate from Versions 2.x or 3.x (Radia 4.x)


 Use Tasks **2a** and **4a** as directed below.

Table 2 Task List for Migration from 2.x or 3.x (Radia 4.x)

Task	Page	Description
Task 1	15	Upgrade the ZTASKEND REXX method.
Task 2a	15	Convert an existing non-Unicode database to Unicode.
Task 3	15	Apply Database schema changes.
Task 4a	22	Upgrade Messaging Server 2.x or 3.x.
Task 5	24	Review Post-Installation Configurations for Patch Manager.
Task 6	26	On UNIX platforms, configure the DataDirect ODBC Drivers.
Task 7	26	Upgrade any store and forward Messaging Servers in your environment.

Task Details

Task 1: Upgrade your Configuration Server ZTASKEND REXX Method

Prior to upgrading the Messaging Server, it is a best practice to adopt the latest version of the ZTASKEND REXX delivered on the Client Automation media with the Configuration Server.

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



New ZTASKEND Method

New features introduced in HPCA 7.20, such as Vulnerability Management and Windows CE Thin Client management, require that you use the latest version of ZTASKEND that is supplied on the HPCA media.

Task 2a: Convert Database to Unicode

For HP Client Automation 7.80 support, all databases must be converted to Unicode.

If your database was created prior to 5.00, it must now be converted to Unicode. If you previously migrated a pre-5.00 database but did not convert your database to Unicode, you must do so now.

The steps to convert a database to Unicode are given in Appendix B, [Database Conversion to Unicode Datatypes](#) (Required) on page 35.

After converting a database to Unicode, continue with Task 3 below.

Task 2b: Run Database Script if Migrating from 5.00 Unicode

This task applies if:

- Your existing Version is 5.00
- Your Inventory Database was previously converted to Unicode as part of migrating from Messaging Server 2.x or 3.x to Version 5.00.

Run the final database migration script prior to upgrading the Messaging Server.

- For SQL Server, run Step6_rWin32_Service_Update_SQLServer.sql. For details, see Appendix B topic: [SQL Server Conversion for Unicode](#) on page 38.
- For Oracle, run Step5_rWin32_Service_Update_Oracle.sql. For details, see [Oracle Conversion for Unicode](#) on page 39.

Task 3: Apply Inventory Manager Database Schema Changes (Both SQL Server or Oracle-hosted Databases)

Make the following changes to an existing Inventory Manager database to support this release of Messaging Server and Reporting Server. Make these changes whether you are using an Oracle or SQL Server database. These changes are required.



Apply all schema changes that were made since the Messaging Server Version from which you are migrating. They are listed from most recent to least recent.

Schema changes from 7.50 to 7.80

There were no schema changes between version 7.50 and version 7.80.

Schema changes from 7.20 to 7.50

1. The VM_ACQUISITION_FILE table requires a new column: scanner

Either delete the existing VM_ACQUISITION_FILE table so it is created with the new column by the Messaging Server 7.80 installation, or modify the existing table definition by adding the column listed in the following table:

Column Name	Data Type	Length	Allow Nulls
scanner	nvarchar	1	√

2. The APPEVENT and HAPPEVENT tables require 2 new columns: jobid and cjobid

Either delete the existing APPEVENT and HAPPEVENT tables so they are created with the new column by the Messaging Server 7.80 installation, or, modify the existing table definition by adding the column listed in the following table:

Column Name	Data Type	Length	Allow Nulls
cjobid	nvarchar	64	√
jobid	nvarchar	64	√

3. The DEVICESYNOPSIS tables requires 1 new column (cjobid) and a modified length for column (jobid)

Either delete the existing APPEVENT and HAPPEVENT tables so they are created with the new and modified columns by the Messaging Server 7.80 installation, or, modify the existing table definition by adding or modifying the columns listed in the following table:

Column Name	Data Type	Length	Allow Nulls
cjobid	nvarchar	64	√
jobid	nvarchar	64 (modified)	√

Procedures

To modify the table definitions:

SQL Server: Run the following commands against a database running on SQL Server.

```
ALTER TABLE VM_ACQUISITION_FILE ADD scanner nvarchar(1) NULL; GO
ALTER TABLE APPEVENT ADD cjobid nvarchar(64) NULL; GO
```



```
ALTER TABLE APPEVENT ADD jobid nvarchar(64) NULL; GO
ALTER TABLE HAPPEVENT ADD cjobid nvarchar(64) NULL; GO
ALTER TABLE HAPPEVENT ADD jobid nvarchar(64) NULL; GO
ALTER TABLE DEVICESYNOPSIS ADD cjobid nvarchar(64) NULL; GO
ALTER TABLE DEVICESYNOPSIS MODIFY jobid nvarchar(64) NULL; GO
```

ORACLE: Run the following commands against a database running on Oracle to add the new column.

```
ALTER TABLE VM_ACQUISITION_FILE
ADD (
    scanner nvarchar2(1) NULL);
```

```
ALTER TABLE APPEVENT
ADD (
    cjobid nvarchar2(64) NULL,
    jobid nvarchar2(64) NULL);
```

```
ALTER TABLE HAPPEVENT
ADD (
    cjobid nvarchar2(64) NULL,
    jobid nvarchar2(64) NULL);
```

```
ALTER TABLE DEVICESYNOPSIS
ADD (cjobid nvarchar2(64) NULL)
MODIFY (jobid nvarchar2(64));
```

Schema changes from 5.10 to 7.20

Make the following changes to an existing Inventory Manager database to support this release of Messaging Server and Reporting Server. Make these changes whether you are using an Oracle or SQL Server database. These changes are required.

1. DeviceConfig and HDeviceConfig Table Changes: Add columns lastuser, subnetaddr and subnetmask

The DeviceConfig and HDeviceConfig tables require new columns: **lastuser**, **subnetaddr** and **subnetmask**. Either delete the existing DeviceConfig and HDeviceConfig tables so they are created with the necessary columns by the Messaging Server 7.80 installation, or, modify the existing Table definitions for DeviceConfig and HDeviceConfig by adding the columns listed in the following table:

Column Name	Data Type	Length	Allow Nulls
lastuser	nvarchar	128	√
subnetaddr	nvarchar	128	√

subnetmask	nvarchar	128	√
------------	----------	-----	---

2. SMBiosInfo Table Changes: Add columns for enc_model, enc_name, enc_slot, rackname

The SMBiosInfo table requires the new columns below. Either delete the existing SMBiosInfo table so it is created with the necessary columns by the Messaging Server 7.80 installation, or, modify the existing Table definition by adding the columns below:

Column Name	Data Type	Length	Allow Nulls
enc_model	nvarchar	128	√
enc_name	nvarchar	128	√
enc_slot	nvarchar	128	√
rackname	nvarchar	128	√

Use the task approach of your choice, below, to add the new columns.

3. rWin32_DiskDrive Table Changes: Add columns for Vista devices: wSerialNumber, wFirmwareRevision, wSignature and wCapabilityDescriptions2

The rWin32_DiskDrive table requires 4 new columns for Vista devices.

Either delete the existing rWin32_DiskDrive table so it is created with the necessary column by the Messaging Server installation, or, modify the existing Table definition for rWin32_DiskDrive by adding the following columns:

Column Name	Data Type	Length	Allow Nulls
wSerialNumber	nvarchar	128	√
wFirmwareRevision	nvarchar	128	√
wSignature	Int	4	√
wCapabilityDescriptions2	nvarchar	128	√

This completes the table changes required for Version 7.20.

Procedures

To modify the table definitions:

SQL Server: Run the following commands against a database running on SQL Server. If your database is not Unicode, substitute varchar for nvarchar:

```
ALTER TABLE DeviceConfig ADD subnetaddr nvarchar(128) NULL; GO
ALTER TABLE DeviceConfig ADD subnetmask nvarchar(128) NULL; GO
ALTER TABLE DeviceConfig ADD lastuser nvarchar(128) NULL; GO
```

```
ALTER TABLE HDeviceConfig ADD subnetaddr nvarchar(128) NULL; GO
ALTER TABLE HDeviceConfig ADD subnetmask nvarchar(128) NULL; GO
ALTER TABLE HDeviceConfig ADD lastuser nvarchar(128) NULL; GO
```

```
ALTER TABLE SMBiosInfo ADD enc_model nvarchar(128) NULL; GO
ALTER TABLE SMBiosInfo ADD enc_name nvarchar(128) NULL; GO
ALTER TABLE SMBiosInfo ADD enc_slot nvarchar(128) NULL; GO
ALTER TABLE SMBiosInfo ADD rackname nvarchar(128) NULL; GO
```

```

ALTER TABLE rWin32_DiskDrive ADD wSerialNumber nvarchar(128) NULL; GO
ALTER TABLE rWin32_DiskDrive ADD wFirmwareRevision nvarchar(128) NULL; GO
ALTER TABLE rWin32_DiskDrive ADD wSignature int NULL; GO
ALTER TABLE rWin32_DiskDrive ADD wCapabilityDescriptions2 nvarchar(128)

```

ORACLE: Run the following commands against a database running on Oracle to add the new columns. If your database is not Unicode, substitute varchar2 for nvarchar2:

```

ALTER TABLE DeviceConfig
ADD (
    subnetaddr nvarchar2(128) NULL,
    subnetmask nvarchar2(128) NULL,
    lastuser nvarchar2(128) NULL);

```

```

ALTER TABLE HDeviceConfig
ADD (
    subnetaddr nvarchar2(128) NULL,
    subnetmask nvarchar2(128) NULL,
    lastuser nvarchar2(128) NULL);

```

```

ALTER TABLE SMBiosInfo
ADD (
    enc_model nvarchar2(128) NULL,
    enc_name nvarchar2(128) NULL,
    enc_slot nvarchar2(128) NULL,
    rackname nvarchar2(128) NULL);

```

```

ALTER TABLE rWin32_DiskDrive
ADD (
    wSerialNumber nvarchar2(128) NULL,
    wFirmwareRevision nvarchar2(128) NULL,
    wSignature int NULL,
    wCapabilityDescriptions2 nvarchar2(128) NULL);

```

To delete and rebuild the existing tables (and reapply any customizations, if necessary):

- 1 Stop the Messaging Server service (RMS.TKD).
Stopping the RMS.TKD service automatically stops message processing for each Data Delivery Agent queue.

- 3 Create a backup of the `device.config.sql`, `smbios.info.sql`, `win32_diskdrive.sql` and `taskend.tcl` files that are found in the following folders under where your Messaging Server is installed.

```

\etc\core\sql\hp\device.config.sql (hp-delivered version)
\etc\core\sql\hp\smbios.info.sql (hp-delivered version)
\etc\core\sql\hp\win32_diskdrive.sql (hp-delivered version)
\etc\core\lib\taskend.tcl

```

Note: If you are migrating from a later Messaging Server version, the files will be found in these locations:

```

\etc\core\hp\device.config.sql (hp-delivered version)
\etc\core\hp\smbios.info.sql (hp-delivered version)
\etc\core\hp\win32_diskdrive.sql (hp-delivered version)
\etc\core\taskend.tcl

```

- 4 Delete the existing `device.config.sql`, `smbios.info.sql`, `win32_diskdrive.sql` and `taskend.tcl` files from the above locations, so that the newest versions can be unpacked when the Messaging Server is installed and executed.
- 5 Following installation, reapply any customizations to the default versions of `device.config.sql`, `smbios.info.sql`, `win32_diskdrive.sql` and `taskend.tcl` files, which now contain the new columns. Place the customized versions of any `*.sql` file in the `\etc\core` directory. This allows your custom scripts to take precedence over the default scripts in the lower-level `\hp` subdirectory. Also reapply any customizations from your backup version of `taskend.tcl` to the default version of `taskend.tcl`, if appropriate, and place the customized file back in `\etc\core\lib\`.

Schema Changes from 2.x, 3.x 5.00 to 5.10

The changes below were introduced with Messaging Server 5.10. If you are upgrading from a pre-5.10 version of the Inventory Manager database, also make these changes whether you are using an Oracle or SQL Server database. These changes are required.

1. DeviceConfig and HDeviceConfig Table Changes: Add columns `clientrel` and `tpmchip`

The `DeviceConfig` and `HDeviceConfig` tables require new columns: **`clientrel`** and **`tpmchip`**. Either delete the existing `DeviceConfig` and `HDeviceConfig` tables so they are created with the necessary column by the Messaging Server installation, or, modify the existing Table definitions for `DeviceConfig` and `HDeviceConfig` by adding the following columns:

Column Name	Data Type	Length	Allow Nulls
<code>clientrel</code>	<code>nvarchar</code>	128	√
<code>tpmchip</code>	<code>nvarchar</code>	1	√

2. SMBiosInfo Table Changes: Add columns for `biosvend`, `biosdate`, `biosvers`, `flashmem`

The `SMBiosInfo` table requires the new columns below. Either delete the existing `SMBiosInfo` table so it is created with the necessary columns by the Messaging Server installation, or, modify the existing Table definition by adding the columns below:

Column Name	Data Type	Length	Allow Nulls
<code>biosvend</code>	<code>nvarchar</code>	128	√

biosdate	nvarchar	128	✓
biosvers	nvarchar	128	✓
flashmem	nvarchar	128	✓

Use the task approach of your choice, below, to add the new columns.

Procedures

To modify the table definitions:

SQL Server: Run the following commands against a database running on SQL Server.

```
ALTER TABLE DeviceConfig ADD clientrel nvarchar(128) NULL; GO
ALTER TABLE HDeviceConfig ADD clientrel nvarchar(128) NULL; GO
ALTER TABLE DeviceConfig ADD tpmchip nvarchar(1) NULL; GO
ALTER TABLE HDeviceConfig ADD tpmchip nvarchar(1) NULL; GO
ALTER TABLE SMBiosInfo ADD biosvend nvarchar(128) NULL; GO
ALTER TABLE SMBiosInfo ADD biosdate nvarchar(128) NULL; GO
ALTER TABLE SMBiosInfo ADD biosvers nvarchar(128) NULL; GO
ALTER TABLE SMBiosInfo ADD flashmem nvarchar(128) NULL; GO
```

ORACLE: Run the following commands against a database running on Oracle to add the new columns.

```
ALTER TABLE deviceconfig
ADD (
    clientrel nvarchar2(128) NULL,
    tpmchip nvarchar2(1) NULL);

ALTER TABLE hdeviceconfig
ADD (
    clientrel nvarchar2(128) NULL,
    tpmchip nvarchar2(1) NULL);

ALTER TABLE SMBiosInfo
ADD (
    bisovend nvarchar2(128) NULL,
    biosdate nvarchar2(128) NULL,
    biosvers nvarchar2(128) NULL,
    flashmem nvarchar2(128) NULL)
```

To delete and rebuild the existing tables (and reapply any customizations, if necessary):

Follow the procedures to delete and rebuild existing tables are given on page 19.

This completes the steps needed to update existing Inventory Database tables for Messaging Server 7.50.

Task 4a: Upgrading from Messaging Server 2x or 3.x

Messaging Server 7.80 requires the Data Delivery Agents delivered with Version 7.80, and vice versa, as well as the latest version of nvdkit. All of these modules are installed by default. 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 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 existing 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
```

- 4 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. Optionally, delete or rename 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
```

- 5 The Patch Manager object processing changed as of Version 5.00, and requires the new Patch DDA configuration file. Rename or delete an existing `patch.dda.cfg` file prior to running the install.



The `rms.cfg` section type of ODBC has been removed. (Prior to Version 5.00, the ODBC section was used to post PATCH messages to a SQL or ORACLE-hosted database.) The `patch.dda.cfg` section type of PATCHDDAODBC must be used to post PATCH messages.

- 6 Launch the installation program for the Messaging Server, available from the following platform-specific location on the HPCA 7.80 media:

```
Infrastructure\extended_infrastructure\messaging_server\<<platform>
```

— For Windows, click on **setup.exe** to launch the installation program.

— For a UNIX platform, enter the following command:

```
setup
```

and press **Enter**.

Follow the prompts to complete the installation, making sure to select the installation of all existing data delivery agents.

- 7 Following installation, reapply any customizations from your backup files to the *.SQL and *.TCL files located in the \etc\core and \etc\wbem directories.
- 8 If your existing Messaging Server 3.x environment used an Inventory Manager Server, the next step is to port any customizations you made from the existing Inventory Manager Server to the appropriate Messaging Server locations.

See Appendix B, [Migrating Custom SQL Code from an Inventory Manager Server](#) on page 43.

- 9 Continue with [Task 5](#) on page 24.

Task 4b: Upgrading from Messaging Server 5.x or 7.x



Inventory Database Migration and Updates

- If you previously converted your Inventory Database to Unicode as part of migrating your Messaging Server to Version 5.00, you should have performed [Task 2b](#) on page 15.
- All customers also need to apply changes to the Inventory Manager Database prior to upgrading the Messaging Server. These were covered in [Continue with Task 3](#) on page 15.

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



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

- 2 Create a backup of the directory where your existing Messaging Server is installed.
- 3 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. Delete or rename an existing configuration file prior to running the install to obtain the newest default configuration for it; this is required for pach.dda.cfg to obtain the new reporting tables:

```
rms.cfg
core.dda.cfg
inventory.dda.cfg
wbem.dda.cfg
patch.dda.cfg
usage.dda.cfg
```



The rms.cfg section type of ODBC no longer exists. As of Version 7.50, the patch.dda.cfg section type of PATCHDDAODBC must be used to post PATCH messages.

- 4 Launch the installation program for the Messaging Server, available from the following platform-specific location on the HPCA 7.80 media:

```
Infrastructure\extended_infrastructure\messaging_server\<<platform>
```

— For Windows, click on **setup.exe** to launch the installation program.

— For a UNIX platform, enter the following command:

```
setup
```

and press **Enter**.

Follow the prompts to complete the installation, making sure to select the installation of all existing data delivery agents.

This completes the steps to upgrade an existing Messaging Server 5.x or 7.x to Version 7.80 on the Configuration Server. Continue with [Task 5](#) below.

Task 5: Review Post-Installation Configurations for Patch Manager

If you are using Messaging Server with Patch Manager, upgrade your Messaging Server before upgrading your Patch Manager Server.

After upgrading both servers, check to see if these post-installation tasks apply to your environment. If necessary, perform the customizations to make sure the Messaging Server is setup to process your Patch data correctly.

Patch Manager: Set the DBTYPE for a Patch ODBC Database on Oracle

If you installed the `patch.dda` and the Patch ODBC Database is running on Oracle, you must change the `DBTYPE` parameter in the `patchddaodbc` section of the `patch.dda.cfg` file from `"MSSQL"` to `"ORACLE"`.

To change the `DBTYPE` for `ORACLE`:

- 1 Use a text editor to edit the `patch.dda.cfg` file located in the `\etc` folder of where the Messaging Server was installed.
- 2 Locate the `patchddaodbc` section, and set the `DBTYPE` to `"ORACLE"`. Enclose the value in quotes. An example is shown below—your values may differ:

```
msg::register patchddaodbc {
    TYPE          PATCHODBC
    DSN           "PATCHMGR"
    USER          "HPCAPATCH"
    PASS          "<encrypted password>{AES256}"
    DBTYPE        "ORACLE"
```

- 3 Save your changes, and restart the Messaging Server service.

Verify the Patch Method Connections and Queue Name

- Patch Manager requires five method connections in the Configuration Server Database. For details, refer to the *HPCA Patch Manager Installation and Configuration 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 following five `PRIMARY.SYSTEM.ZMETHOD` instances to match the Patch Directory to Scan value:

```
PATCH_DEERROR
PATCH_BUSTATUS
PATCH_DESTATUS
```

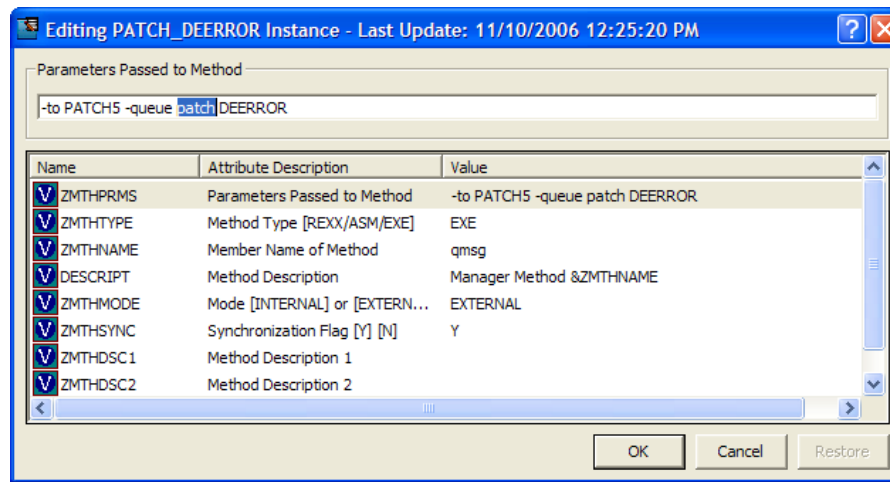

PATCH_RESTATUS

PATCH_PASTATUS

To modify the queue name in the five PATCH_* methods

- 1 Use the HPCA Administrator CSDB Editor to edit the ZMTHPRMS attribute of the PRIMARY.SYSTEM.ZMETHOD.PATCH_DEERROR 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".

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_DEERROR instance from:

```
-to PATCH5 -queue patch DEERROR
```

to

```
-to PATCH5 -queue mypatch DEERROR
```

- 3 Save your changes.
- 4 Repeat the above change you made to the ZMTHPRMS -queue value in the PATCH_DEERROR instance to these PRIMARY.SYSTEM methods:

PATCH_BUSTATUS

PATCH_DESTATUS

PATCH_RESTATUS

PATCH_PASTATUS

- 5 Save your changes.

Task 6: On UNIX platforms, Configure the HP-Supplied Datadirect Connect ODBC Drivers

Messaging Servers installed on UNIX platforms require the configuration of the HP-provided Datadirect Connect drivers in order to post data to a backend database. For information on how to configure these drivers, refer to the topics in the *Messaging Server Installation and Configuration Guide*, located in the \documentation folder of the HP Client Automation 7.20 media.

Task 7: Upgrade and Reconfigure Store and Forward Messaging Servers (to Place Objects Close to a SQL-compliant Database)

As previously mentioned, the installation program does not install a Messaging Server fully configured for store and forward capabilities in an HPCA classic environment.

► The HPCA Satellites support Messaging Servers configured for store and forward activities. The upgrades of the Messaging Servers on your Satellites are handled automatically when the Satellite Servers are upgraded. Refer to the *HPCA Core and Satellite Migration Guide* for details.

To upgrade your classic Messaging Servers configured for store and forward activities:

- 1 Perform the same steps named in Task 4 that are used to upgrade the Messaging Server on the Configuration Server. HP recommends renaming the existing *.cfg files so you have your store and forward configuration information available.
- 2 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 *HPCA Messaging Server Installation and Configuration Guide*.

A Inventory Manager Database – Tables and Scripts

Presently there are 142 Inventory Manager Database tables created by default from the .sql scripts in the etc\core, etc\wbem and etc\inventory directories of the Messaging Server.

No new .sql scripts were added for Messaging Server version 7.80.

Scripts added as of Messaging Server version 7.50 include:

- audit.attr.names.sql
- audit.cat.sql
- audit.event.s--ql
- audit.evt.attrs.sql
- audit.type.sql
- msstoragedriver_attributedata.sql

Scripts added as of Messaging Server V 7.20 include:

- acqprocess.sql
- aquisitions.sql
- cve.sql
- cve_url.sql
- devicescan.sql
- devicevulnerability.sql
- oval.sql
- ovalplatform.sql
- ovalsoftware.sql
- platform.sql
- severityhist.sql
- software.sql
- win32_physicalmemory.sql

Scripts added as of Messaging Server V 5.0 include:

- hpprov_biosenumeration.sql
- hpprov_biospassword.sql
- hpprov_biosstring.sql
- hpprov_biosorderedlist.sql

- win32_portablebattery
- win32_baseboard.sql
- win32_quickfixengineering.sql

Tables created with the .sql files included in the HPCA Messaging Server 7.80 release are listed below.

Table 3 SQL Filenames and Tables created by Messaging Server 7.80

SQL Filename	Table in Inventory Manager Database
apps.jobparm.sql	JOBPARM
apps.jobstat.sql	JOBSTAT
apps.jobstat.sql	HJOBSTAT
apps.jobtask.sql	JOBTASK
apps.msiseservices.sql	AppMSIEvent
apps.msiseservices.sql	HAppMSIEvent
apps.rnpservices.sql	AppRNPEvent
apps.rnpservices.sql	HAppRNPEvent
apps.services.sql	AppEvent
apps.services.sql	HAppEvent
device.config.sql	DeviceConfig
device.config.sql	HDeviceConfig
device.errors.sql	DeviceErrors
device.errors.sql	HDeviceErrors
device.map.sql	DeviceMap
device.services.sql	DeviceServices
device.state.sql	DeviceState
device.state.sql	HDeviceState
device.status.sql	DeviceStatus
device.status.sql	HDeviceStatus
device.synopsis.sql	DeviceSynopsis
device.zrstate.sql	DeviceZRState

device.zrstates.sql	DeviceZRStates
fileaudit.sql	FileAudit
notify.sql	DeviceNotify
query.sql	Query
smbios.info.sql	SMBiosInfo
usergroup.sql	DeviceUserGroup
fileaudit.sql	FileAudit

cim_cdromdrive.sql	rCIM_CDROMDrive
cim_computersystem.sql	rCIM_ComputerSystem
cim_directory.sql	rCIM_Directory
cim_diskdrive.sql	rCIM_DiskDrive
cim_dvddrive.sql	rCIM_DVDDrive
cim_ethernetadapter.sql	rCIM_EthernetAdapter
cim_export.sql	rCIM_Export
cim_hpux_swbundles.sql	rCIM_HPUX_SwBundles
cim_idecontroller.sql	rCIM_IDEController
cim_logicaldisk.sql	rCIM_LogicalDisk
cim_logicaldiskbasedonvolume.sql	rCIM_LogicalDiskBasedOnVolume
cim_mediapresent.sql	rCIM_MediaPresent
cim_nfs.sql	rCIM_NFS
cim_operatingsystem.sql	rCIM_OperatingSystem
cim_parallelcontroller.sql	rCIM_ParallelController
cim_process.sql	rCIM_Process
cim_processor.sql	rCIM_Processor
cim_product.sql	rCIM_Product
cim_productsoftwarefeatures.sql	rCIM_ProductSoftwareFeatures

cim_residesonextent.sql	rCIM_ResidesOnExtent
cim_scsicontroller.sql	rCIM_SCSIController
cim_scsiinterface.sql	rCIM_SCSIInterface
cim_service.sql	rCIM_Service
cim_softwareelement.sql	rCIM_SoftwareElement
cim_softwarefeature.sql	rCIM_SoftwareFeature
cim_softwarefeaturesoftwareelements.sql	rCIM_SoftwareFeatureElements
cim_storagevolume.sql	rCIM_StorageVolume
cim_unixcomputersystem.sql	rCIM_UnixComputerSystem
cim_unixlocalfilesystem.sql	rCIM_UnixLocalFileSystem
cim_unixoperatingsystem.sql	rCIM_UnixOperatingSystem

hp_biosenumeration.sql	rhp_biosenumeration
hp_biosevent.sql	rhp_biosevent
hp_biosinteger.sql	rhp_biosinteger
hp_biosorderedlist.sql	rhp_biosorderedlist
hp_biospassword.sql	rhp_biospassword
hp_biossensor.sql	rhp_biossensor
hp_biosstring.sql	rhp_biosstring

msstoragedriver_attributedata.sql	rMSSD_AttributeData
nvd_downloadstatistics.sql	rNVD_DownloadStatistics
nvd_groupaccount.sql	rNVD_GroupAccount
nvd_groupmember.sql	rNVD_GroupMember
nvd_installed_apps.sql	rNVD_INSTALLED_APPS
nvd_installed_uninstall.sql	rNVD_INSTALLED_UNINSTALL
nvd_multicaststatistics.sql	rNVD_MulticastStatistics

nvd_nisgroupaccount.sql	rNVD_NISGroupAccount
nvd_nisuseraccount.sql	rNVD_NISUserAccount
nvd_pdasystem.sql	rNVD_PDASystem
nvd_product.sql	rNVD_Product
nvd_solarispatch.sql	rNVD_SolarisPatch
nvd_useraccount.sql	rNVD_UserAccount
nvd_wbemstatus.sql	rNVD_WBEMStatus

registry.sql	rRegistry
wifi_networkadapter.sql	rWiFi_NetworkAdapter

win32_baseboard.sql	rWin32_baseboard
win32_bios.sql	rWin32_BIOS
win32_bootconfiguration.sql	rWin32_BootConf
win32_bus.sql	rWin32_Bus
win32_cachememory.sql	rWin32_CacheMemory
win32_cdromdrive.sql	rWin32_CDROMDrive
win32_computersystem.sql	rWin32_ComputerSystem
win32_computersystemproduct.sql	rWin32_ComputerSystemProduct
win32_desktop.sql	rWin32_Desktop
win32_desktopmonitor.sql	rWin32_DesktopMonitor
win32_devicememoryaddress.sql	rWin32_DeviceMemoryAddress
win32_diskdrive.sql	rWin32_DiskDrive
win32_diskpartition.sql	rWin32_DiskPartition
win32_displayconfiguration.sql	rWin32_DisplayConf
win32_displaycontrollerconfiguration.sql	rWin32_DisplayControllerConf
win32_dmachannel.sql	rWin32_DMACHannel

win32_environment.sql	rWin32_Environment
win32_floppycontroller.sql	rWin32_FloppyController
win32_floppydrive.sql	rWin32_FloppyDrive
win32_group.sql	rWin32_Group
win32_idecontroller.sql	rWin32_IDEController
win32_irqresource.sql	rWin32_IRQResource
win32_keyboard.sql	rWin32_Keyboard
win32_loadordergroup.sql	rWin32_LoadOrderGroup
win32_logicaldisk.sql	rWin32_LogicalDisk
win32_logicalmemoryconfiguration.sql	rWin32_LogicalMemoryConf
win32_logicalprogramgroup.sql	rWin32_LogicalProgramGroup
win32_memoryarray.sql	rWin32_MemoryArray
win32_memorydevice.sql	rWin32_MemoryDevice
win32_motherboarddevice.sql	rWin32_MotherboardDevice
win32_networkadapter.sql	rWin32_NetworkAdapter
win32_networkadapterconfiguration.sql	rWin32_NetworkAdapterConf
win32_networkconnection.sql	rWin32_NetworkConnection
win32_networkloginprofile.sql	rWin32_NetworkLoginProfile
win32_operatingsystem.sql	rWin32_OperatingSystem
win32_pagefile.sql	rWin32_PageFile
win32_pagefilesetting.sql	rWin32_PageFileSetting
win32_pagefileusage.sql	rWin32_PageFileUsage
win32_parallelport.sql	rWin32_ParallelPort
win32_physicalmemory.sql	rWin32_PhysicalMemory
win32_pnpentity.sql	rWin32_PnPEntity
win32_pointingdevice.sql	rWin32_PointingDevice
win32_portablebattery.sql	rwin32_portablebattery

win32_portresource.sql	rWin32_PortResource
win32_printer.sql	rWin32_Printer
win32_process.sql	rWin32_Process
win32_processor.sql	rWin32_Processor
win32_product.sql	rWin32_Product
win32_quickfixengineering.sql	rwin32_quickfixengineering
win32_serialport.sql	rWin32_SerialPort
win32_service.sql	rWin32_Service
win32_share.sql	rWin32_Share
win32_softwareelement.sql	rWin32_SoftwareElement
win32_softwarefeature.sql	rWin32_SoftwareFeature
win32_sounddevice.sql	rWin32_SoundDevice
win32_startupcommand.sql	rWin32_StartupCommand
win32_systemdriver.sql	rWin32_SystemDriver
win32_systemenclosure.sql	rWin32_SystemEnclosure
win32_timezone.sql	rWin32_TimeZone
win32_usbcontroller.sql	rWin32_USBController
win32_useraccount.sql	rWin32_UserAccount
win32_videocontroller.sql	rWin32_VideoController
audit.event.attrs.sql	Audit_Attrs
audit.attr.names.sql	Audit_AttrNames
audit.cat.sql	Audit_Cat
audit.event.sql	Audit_Event
audit.type.sql	Audit_Type
acquisitions.sql	VM_ACQUISITION_FILE
acqprocess.sql	VM_ACQUISITION_PROCESS

platform.sql	VM_AFFECTED_PLATFORM
software.sql	VM_AFFECTED_SOFTWARE
cve.sql	VM_CVE
cve_url.sql	VM_CVE_URL
devicescan.sql	VM_DEVICE_SCAN
devicevulnerability.sql	VM_DEVICE_VULNERABILITY
oval.sql	VM_OVAL
ovalplatform.sql	VM_OVAL_PLATFORM
ovalsoftware.sql	VM_OVAL_SOFTWARE
severityhist.sql	VM_SEVERITY_HISTORY

B Additional Database Migration Topics

This Appendix discusses additional migration topics that may apply if you are migrating from Radia 4.x and Messaging Server 2.x or 3.x. These topics were referenced from specific Tasks in previous chapters.

- [Database Conversion to Unicode Datatypes \(Required\)](#) below. This is required for all HPCA 7.80 customers who do not currently have a Unicode database.
- [Migrating Custom SQL Code from an Inventory Manager Server](#) on page 43. Some Radia 4.x customers may have custom code that needs to be ported from their Inventory Manager Server to their Messaging Server. After porting the custom code, the Inventory Manager Server can be retired.

Database Conversion to Unicode Datatypes (Required)

As of Version 5.00, Client Automation offered multilingual support as an option and existing customers needed to convert their existing Inventory Databases to Unicode to support this feature.

As of Version 7.80, all customers must convert an Inventory or Core database to Unicode, regardless of whether or not they want to take advantage of multilingual support. HPCA 7.80 is now designed to work with Unicode databases across the board.

The conversion process takes existing Inventory Database created with varchar datatypes and converts them to support nvarchar datatypes.

Why convert the database for Unicode support?

Storing data in multiple languages within one database is difficult to manage when you use only character data and code pages. It is also difficult to find one code page for the database that can store all the required language-specific characters. Additionally, it is difficult to guarantee the correct translation of special characters when being read or updated by different clients running various code pages. Databases that support international clients should always use Unicode data types instead of non-Unicode data types.

For example, consider a database of customers in North America that must handle three major languages:

- Spanish names and addresses for Mexico
- French names and addresses for Quebec
- English names and addresses for the rest of Canada and the United States

When you use only character columns and code pages, you must take care to make sure the database is installed with a code page that will handle the characters of all three languages. You must also take care to guarantee the correct translation of characters from one of the languages when read by clients running a code page for another language.

With the growth of the Internet, it is even more important to support many client computers that are running different locales. Selecting a code page for character data types that will support all the characters required by a worldwide audience would be difficult.

The easiest way to manage character data in international databases is to always use the Unicode nchar, nvarchar, and nvarchar(max) data types, instead of their non-Unicode equivalents, char, varchar, and text.

Unicode is a standard for mapping code points to characters. Because it is designed to cover all the characters of all the languages of the world, there is no need for different code pages to handle different sets of characters. SQL Server 2005 supports the Unicode Standard, Version 3.2.

If all the applications that work with international databases also use Unicode variables instead of non-Unicode variables, character translations do not have to be performed anywhere in the system. Clients will see the same characters in the data as all other clients.

SQL Server 2005 stores all textual system catalog data in columns having Unicode data types. The names of database objects, such as tables, views, and stored procedures, are stored in Unicode columns. This enables applications to be developed by using only Unicode, and helps avoid all issues with code page conversions.

About the SQL Server Migration Scripts

There are scripts included in a migrate directory with the Messaging Server install for converting SQL Server and Oracle databases to convert the varchar datatype to nvarchar. All default conversion scripts will address the standard tables created by the Messaging Server and previously by RIM Server. Additional custom tables must be converted separately. See the Appendix on page 27 for the listing of the standard tables created by the included scripts.

All scripts must be reviewed by a Data Base Administrator familiar with the specific custom environment. The scripts are given as guidelines and in some cases must be edited prior to execution.

Always backup your existing database prior to performing this type of conversion.

Possible Unicode Migration Issues and Resolution Options

Expected Warning: Table Index Exceeds the Maximum Number of Bytes

During the migration process, you will see database warnings when running the migration scripts against your Inventory tables. You will be warned whenever the total bytes for a table's composite index, after conversion to the Unicode acceptable datatype, are greater than the permitted maximum size.

Using a SQL Server database as an example, the maximum size allowed for an index is 900 bytes. Several of the Inventory tables, when converted to nvarchar datatypes, will show a warning that the maximum size of the index has been exceeded.

In most cases these warnings do not present a problem because the column sizes were created with default values that were much larger than needed. The warning includes an alert that if subsequent insert or update actions on the variable-type columns result in a total size greater than 900 bytes, the action will fail and the user will get a run-time error. Likewise, if the index definition is composed of variable-type columns only, and the maximum total size of these columns is greater than 900 bytes, SQL Server will create the index, but return a warning.

The tables that can be expected to show this warning are listed in [Table 4](#) on page 37.

Table 4 Tables that may show Warnings during database migration

rCIM_CDROMDrive	rCIM_SoftwareElement
rCIM_Directory	rCIM_SoftwareFeatureElements
rCIM_DiskDrive	rCIM_StorageVolume
rCIM_DVDDrive	rCIM_UnixLocalFileSystem
rCIM_EthernetAdapter	rCIM_UnixOperatingSystem
rCIM_Export	rNVD_DownloadStatistics
rCIM_HPUNIX_SwBundles	rNVD_GroupAccount
rCIM_IDEController	rNVD_GroupMember
rCIM_LogicalDisk	rNVD_MulticastStatistics
rCIM_LogicalDiskBasedOnVolume	rNVD_NISGroupAccount
rCIM_MediaPresent	rNVD_NISUserAccount
rCIM_NFS	rNVD_Product
rCIM_OperatingSystem	rNVD_SolarisPatch
rCIM_ParallelController	rNVD_UserAccount
rCIM_Process	rRegistry
rCIM_Processor	rWin32_BIOS
rCIM_ProductSoftwareFeatures	rWin32_ComputerSystemProduct
rCIM_ResidesOnExtent	rWin32_Product
rCIM_SCSIController	rWin32_SoftwareElement
rCIM_SCSIInterface	rWin32_SoftwareFeature
rCIM_ServiceI	rWin32_StartupCommand

Possible Error and Resolution

However, if the actual data in an existing database table does exceed the composite index maximum limit during migration, the Messaging Server will show an error when it is being restarted. The error will look like this:

```
{[Microsoft][ODBC SQL Server Driver][SQL Server]Operation failed. The index entry of length 906 bytes for the index 'rCIM_SoftwareFeatureElementsI' exceeds the maximum length of 900 bytes.}
```

This error identifies the Inventory table whose data exceeds this maximum index value. The database error must be resolved in one of the following ways.

- If the data does not need to be retained, the table can be dropped from your database and will be recreated upon Messaging Server startup. Note, however, that this option does not prevent against subsequent entries or updates to the database table resulting in the composite index exceeding the maximum.
- If the data is valid and needs to be retained, the table can be converted back to a varchar datatype to resolve the error. To do this, create a script to convert the table back to varchar, run the script against your database, and then restart the Messaging Server. As an example, the following sample script was created to convert a table back to the

varchar datatype. To generate this sample script, we performed a search on each of the migration scripts for the table name "rCIM_SoftwareFeatureElements".

Sample script to revert a table from the datatype nvarchar to varchar

```
DROP INDEX
[dbo].[rCIM_SoftwareFeatureElements].[rCIM_SoftwareFeatureElementsI]
GO

ALTER TABLE [dbo].[rCIM_SoftwareFeatureElements]
    ALTER COLUMN [userid] [varchar] (32) Collate Latin1_General_CI_AS
GO

ALTER TABLE [dbo].[rCIM_SoftwareFeatureElements]
    ALTER COLUMN [wGroupComponent] [varchar] (255) Collate
Latin1_General_CI_AS
GO

ALTER TABLE [dbo].[rCIM_SoftwareFeatureElements]
    ALTER COLUMN [wNamespace] [varchar] (128) Collate Latin1_General_CI_AS
GO

ALTER TABLE [dbo].[rCIM_SoftwareFeatureElements]
    ALTER COLUMN [wPartComponent] [varchar] (255) Collate
Latin1_General_CI_AS
GO

CREATE UNIQUE INDEX [rCIM_SoftwareFeatureElementsI] ON
[dbo].[rCIM_SoftwareFeatureElements]([userid], [wNamespace],
[wGroupComponent], [wPartComponent]) ON [PRIMARY]
GO
```

After you run the script(s) on the database to revert the affected table(s) to a varchar datatype, restart the Messaging Server service and check the logs to validate that the module started without error.

SQL Server Conversion for Unicode

There are six scripts for the conversion of a SQL Server database. They are located on the HPCA media at:

```
Infrastructure\extended_infrastructure\messaging_server\
migrate\SQL_Server
```

Run the scripts using the Microsoft SQL Server Enterprise Manager Query Analyzer tool.



If your SQL Server database was migrated to Unicode for Messaging Server Version 5.00, just run the two scripts: Step5_Modify_Indexes_MSSQL.sql and Step6_rWin32_Service_Update_SQLServer.sql.

- Step1_Drop_Indexes_MSSQL.sql
The first script will drop the indexes from the standard table in the database
- Step2_Alter_Database_MSSQL.sql
Follow these instructions before executing a script:
 - a Replace RIMDB by your Inventory manager database Name.
 - b Replace [NEW Collation Name] by your new desired Name for example French_CI_AS etc.
 - c To run the script, the program needs to set the database into single user mode.
You should therefore ensure that there are no open connections on the database before running the script (use the stored -- procedure SP_WHO to identify any open connections).
You may want to use Kill command to Force logout the connections to the database.
 - d If you are running these commands from SQL Query Analyzer, its preferable Choose a different database eg Master and Run -- the below commands.
- Step3_Alter_Inventory_Tables_MSSQL.sql
This script alters the Table columns to support Unicode strings with the Collate of your choice.
Replace Latin1_General_CI_AS with the Desired Collate of your choice
- Step4_Add_Indexes_MSSQL.sql
This script adds the indexes back to the standard tables.
- Step5_Modify_Indexes_MSSQL.sql
This script corrects any UNIQUE indexes that were previously created as NON-UNIQUE.
- Step6_rWin32_Service_Update_SQLServer.sql
This step modifies the rWin32_Service table to set the wDescription column length to 512.

Oracle Conversion for Unicode



If your Oracle database was migrated to Unicode for Messaging Server Version 5.00, just run the last two scripts: Step4_Alter_Tables_Oracle.sql and Step5_rWin32_Service_Update_Oracle.sql.

There are 5 scripts for the conversion of an Oracle database. They are located on the HPCA media at:

```
Infrastructure\extended_infrastructure\messaging_server\
migrate\Oracle
```

Run these scripts as directed in Task 8, below, as the last task in converting an Oracle database to Unicode.

Task 1 Export Data

Use the Oracle export utility (EXP.exe found in the Oracle BIN directory) and export the data using the sample command below.

The Oracle export utility is located in these locations:

- For Oracle 9i:
`Install Drive\oracle\ora92\bin\EXP.EXE`
- For Oracle 10g:
`Install Drive\oracle\product\10.2.0\db_1\BIN\EXP.EXE`
- For Oracle 11g:
`Install Drive\app\Administrator\product\11.1.0\db_1\BIN\EXP.EXE`

Command to export data:

From a command prompt, go to the bin folder and type the following command; change any parameters if you wish and make sure the folder names that are used do exist.

```
exp CMDATA/CMDATA FILE=c:\RMSDATA\CMDATA.dmp OWNER=CMDATA GRANTS=y  
ROWS=y indexes=n constraints=y COMPRESS=y log=c:\RMSDATA\CMDATA.log
```

Here is a sample export command for Oracle 10g:

```
C:\oracle\product\10.2.0\db_1\BIN>exp CMDATA/CMDATA  
FILE=c:\RMSDATA\CMDATA.dmp OWNER=CMDATA GRANTS=y ROWS=y indexes=n  
constraints=y COMPRESS=y log=c:\RMSDATA\CMDATA.log  
Export: Release 10.2.0.1.0 - Production on Tue Jun 27 17:17:29 2008  
Copyright © 1982, 2002, Oracle Corporation. All rights reserved.
```

- ▶ If you are a database administrator, HP recommends that you change the settings of the export commands to suit your business needs, based on the environmental settings, as required.
- ▶ If you are a database administrator, HP also recommends that you review the database character set and NLS settings in order to rule out possible character conversions during import. For more information, refer to the *Oracle Database Globalization Support Guide*.

Task 2 Create New Unicode Database

Create a new Oracle 10g or 11g database with a database character set that supports Unicode data, preferably the Database Character Set AL32UTF8 or National Character Set UTF8.

- ▶ If you are a database administrator, HP recommends that you change the settings of the database character set to suit your needs. You may also consider changing the Block Size.

Task 3 Define Tablespaces

Using the Oracle Enterprise Manager console or SQL commands, create the tablespaces. Log in as system or SysDBA and execute the commands below:

For Oracle 10g:

```
create tablespace CMDATA datafile  
'c:\oracle\product\10.2.0\oradata\orcl\CMDATA.dbf' SIZE 10M AUTOEXTEND  
ON NEXT 10M MAXSIZE 100M;
```



```
create temporary tablespace CMDATAtmp tempfile
'c:\oracle\product\10.2.0\oradata\orcl\CMDATAtmp.dbf' SIZE 10M
AUTOEXTEND ON NEXT 10M MAXSIZE 100M;
```

For Oracle 11g:

```
create tablespace CMDATA datafile
'c:\app\Administrator\oradata\orcl1\CMDATA.dbf' SIZE 10M AUTOEXTEND ON
NEXT 10M MAXSIZE 100M;
```

```
create temporary tablespace CMDATAtmp tempfile
'c:\app\Administrator\oradata\orcl1\CMDATAtmp.dbf' SIZE 10M
AUTOEXTEND ON NEXT 10M MAXSIZE 100M;
```

Task 4 Define Common Roles

Use SQL Plus or Sql Worksheet, log in as system or **SysDBA**, and execute the following to define privileges to the user.

For Oracle 10g:

```
create user CMDATA identified by CMDATA default tablespace CMDATA
temporary tablespace CMDATAtmp;
grant create session to CMDATA;
grant create table to CMDATA;
grant create view to CMDATA;
grant create sequence to CMDATA;
alter user CMDATA quota unlimited on CMDATA;
grant connect,resource to cmdata;
```

For Oracle 11g:

```
create user CMDATA identified by CMDATA default tablespace CMDATA
temporary tablespace CMDATAtmp;
grant create session to CMDATA;
grant create table to CMDATA;
grant create view to CMDATA;
grant create sequence to CMDATA;
alter user CMDATA quota unlimited on CMDATA;
grant connect,resource to cmdata;
drop user cmdata cascade;
drop tablespace cmdata including contents and datafiles;
drop tablespace cmdatatmp including contents and datafiles;

select table_name from user_tables where table_name like 'VM_%';
```

Task 5 Import Data

Use the Oracle import utility (IMP.exe) to import the data. The import command and sample instructions are given below.

For Oracle 10g:

For 10g, the import utility can be found here:

Install Drive\oracle\product\10.2.0\db_1\BIN\EXP.EXE

Here is a sample import command:

```
C:\ oracle\product\10.2.0\db_1\BIN\>imp cmdata\cmdata
FILE=c:\RMSDATA\CMDATA.dmp SHOW=n FROMUSER=CMDATA ROWS=Y IGNORE=y
GRANTS=y Constraints=y log=c:\rmsdata\cmdataimp.log
```

- If you are a database administrator, HP recommends that you change the settings of the database character set to suit your needs. You may also consider changing the Block Size.
- If you are a database administrator, HP also recommends that you change the settings of the import commands to suit your business needs.

For Oracle 11g:

For 11g, the import utility can be found here:

Install Drive\app\Administrator\product\11.1.0\db_1\BIN.

After the import is successful, log in to Oracle Enterprise Manager and go to the views of the database user (often named **CMDATA**). Right-click on each view and compile.

Task 6 Create DSN for new database

If necessary, create a new DSN to point to the new database and reconfigure the Messaging Server to use this database for import.

Task 7 Run the scripts to complete the conversion of an Oracle database.

There are 5 scripts for the conversion of an Oracle database. They are located on the HPCA media at:

Infrastructure\extended_infrastructure\messaging_server\migrate\Oracle

- Step1_Drop_Indexes_Oracle.sql
- Step2_Alter_Tables_Oracle.sql
- Step3_Create_Indexes_Oracle.sql
- Step4_Alter_Tables_Oracle.sql
- Step5_rWin32_Service_Update_Oracle.sql

Migrating Custom SQL Code from an Inventory Manager Server

When migrating from Radia 4.x to HPCA Version 7.80, the Inventory Manager Server is no longer supported. Instead, the Messaging Server now creates the same tables using the same scripts that were previously created by the Inventory Manager server. The Messaging Server's CORE, WBEM and INVENTORY Data Delivery Agents are now used to post data directly to a SQL Database or Oracle Database.

If you have customized your Inventory Database, this topic discusses how you can port any customized versions of the files listed in [Table 5](#) on page 44 from their locations on your Inventory Manager Server to their equivalent locations on the Messaging Server.

The Data Delivery Agents are now used to post data directly to a SQL Database or Oracle Database.

- ▶ You only need to port the custom code to a Messaging Server that is 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 previous Inventory Manager Server. These Data Delivery Agents use the exact same table definitions used by the legacy Inventory Manager Server to create tables, update and delete data. If the SQL tables were not previously 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>/hp` directories. However, custom versions of these `.sql` files are to be placed in the `/etc/<module name>` directories; this means the customized versions will be executed instead of the HP-delivered versions placed in the lower level `hp` subdirectories.

- ▶ Both CORE and INVENTORY SQL queries are placed the `/etc/CORE/hp` directory.

WBEM SQL queries are placed in `/etc/WBEM/hp`.

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` previously used on the Inventory Manager Server. The script necessary to map the INVENTORY object data (FILEPOST object) is called `filepost.tcl`. Both these scripts are found in the `/etc/CORE/hp` directory of the Messaging Server. Using the identical scripts found on the Inventory Manager 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 an Inventory Manager Server to a Messaging Server](#)

- 1 If necessary, stop the service for the 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 5 below to locate the appropriate Messaging Server directory and file for each object type. Copy any customized versions of the code on your Inventory Manager Server to the appropriate location on the Messaging Server.

Table 5 Directory Locations for Migrating Custom Code from RIM to RMS

Data Directory Agent: and Files	Inventory Server Directory Location for Custom Code	Messaging Server 3.x, 5.x Directory Location for Custom Code	Messaging Server 7.x Directory Location for Custom Code
core.dda support:			
taskend.tcl	<RIS>\etc\rim\lib	<RMS>\etc\core\lib	<RMS>\etc\core
*.sql files	<RIS>\etc\sql	<RMS>\etc\core	<RMS>\etc\core
inventory.dda support			
filepost.tcl	<RIS>\etc\rim\lib	<RMS>\etc\inventory\lib	<RMS>\etc\core
*.sql files	<RIS>\etc\sql	<RMS>\etc\inventory	<RMS>\etc\core
wbem.dda support			
*.sql files	<RIS>\etc\sql\wbem	<RMS>\etc\wbem	<RMS>\etc\wbem

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

- 5 Restart the Messaging Server service or process.

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. The Inventory Manager Server can be removed.