# HP OpenView ServiceCenter

For supported Windows® and UNIX® operating systems

Upgrading Software Versions: SC4-SC6.2

# Upgrade Guide

Manufacturing Part Number: T4011-90006
Document Release Date: October 2006
Software Release Date: October 2006

# 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

## Trademark Notices

Java™and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.

# Documentation Updates

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

- Software version number, which indicates the software version
- 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 of a document, go to:

**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.

# Support

## Peregrine Product Support

You can obtain support information for products formerly produced by Peregrine as follows:

- If you work with an HP OpenView Services Integrator (SVI) partner (**www.hp.com/managementsoftware/svi_partner_list**), contact your SVI agent.
- If you have an active HP OpenView support contract, visit the HP OpenView Support site and use the Self-Solve Knowledge Search to find answers to technical questions.
- For the latest information about support processes and tools available for products formerly produced by Peregrine, we encourage you to visit the HP-Peregrine Software Support web site at: **www.hp.com/managementsoftware/peregrine_support** where you can download the Customer Support Handbook.
- Contact your HP Sales Representative if you have additional questions.

## HP OpenView Support

You can visit the HP OpenView Support web site at:
**www.hp.com/managementsoftware/support**

HP OpenView online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support 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 find more information about access levels, go to:
**www.hp.com/managementsoftware/access_level**

To register for an HP Passport ID, go to:
**www.managementsoftware.hp.com/passport-registration.html**

# Contents

# Application upgrade overview

The ServiceCenter Upgrade Utility upgrades the ServiceCenter applications SC4 or later version to a newer version of the applications. If you are running a ServiceCenter application version of A9902 or earlier, you should contact *HP OpenView Customer Support* for information about the best upgrade strategy for your version.

Topics in this section include:

## Upgrading the client and server

The ServiceCenter Upgrade Utility upgrades only ServiceCenter applications. It does not upgrade the ServiceCenter client or server. You can upgrade the server run-time environment (RTE) separately from the ServiceCenter applications. If you are only upgrading your server RTE, you do not need to create and apply a custom upgrade or do conflict resolution.

- To obtain the latest client, install the ServiceCenter 6.2 client from the installation CD-ROM by following the instructions in the *ServiceCenter Installation guide.*
- To obtain the latest server, upgrade the ServiceCenter 6.2 server from the installation CD-ROM by following the instructions in *Phase III: Upgrading the Server Runtime Environment on page 35 of this guide.*

# What are applications?

Applications are the ServiceCenter modules and their related configuration files. For example, Incident Management, Change Management, and Inventory Management are ServiceCenter applications.

Applications reside in the ServiceCenter file system in a series of tables described by the database dictionaries (dbdicts) named format, application, code, and enclapplication. Other database dictionaries, such as formatctrl, validity, and environment describe tables containing supporting data.

Because ServiceCenter enables you to modify existing database dictionaries or add new database dictionaries, the ServiceCenter Upgrade Utility identifies differences between your existing customized files and new files, and enables you to make decisions about your migration strategy.

Complete migration to a newer version of ServiceCenter applications requires you to:

- Install the appropriate clients by using the ServiceCenter installation CD-ROM and the *ServiceCenter Installation Guide*.
- Upgrade the server and applications by using the instructions in this guide.

# Version numbers

When you do an upgrade you must be aware of three version numbers, the software release version number, the application version number, and the Upgrade Utility version number.

## Application version numbers

In ServiceCenter 3 and earlier versions, application version numbers were the format *Ayyrr,* where *yy* represents the last two digits of the year of the release and *rr* represents the release number for that year. For example, the first application release of 1999 was A9901. Starting with ServiceCenter 4.0, applications are identified as SC*x*, where *x* is the version number. For example, ServiceCenter 4.0 applications are version SC4.

The application versions upgraded by this release include:

- SC4
- SC4.0.5
- SC5
- SC5.1
- SC6.0

SC61

## Upgrade Utility version numbers

Earlier versions of the Upgrade Utility matched the application version number. Beginning with SC6 applications, the utility has a version number in this format:

*application_version_range vn.n Build nnnn language*

For example, if the version number is SC4–SC6.2 v1.0 Build 0020 English,

- The Upgrade Utility upgrades SC4 (and later releases) to version SC6.2
- The Upgrade Utility version number is 1.0
- The Upgrade Utility build number for this version is 0020
- The language supported is English only

When the Upgrade Utility finishes, it displays the date, time, old application version, new application version, and Upgrade Utility version number on the upgrade screen:

```
07/16/06 23:12:46 Upgraded from SC4 to SC6.2
using SC40-SC6.2 v1.0 build 0020 E
```

# How does customization affect the upgrade process?

Most customers modify their systems to meet their business needs. Because of changes to the functionality and behavior of a customized system, upgrading ServiceCenter is not an automatic process.

## Protecting customized files

The ServiceCenter Upgrade Utility determines whether each ServiceCenter file is identical to the out-of-box file installed originally. If the existing file is unchanged, the Upgrade Utility replaces it with a new version. If the file on your system is different from the out-of-box version, the Upgrade Utility keeps your version and adds a prefix (NEW6.2<name>) to the name of the new version.

Resolving differences between your files and new application files, and integrating them to preserve functionality creates a *custom upgrade*.

## Interdependency

ServiceCenter has an inherent application interdependency. Files often rely on other files to complete tasks. If the Upgrade Utility upgrades one file and does not upgrade a dependent file, the two may not interact properly. The application upgrade process is designed to avoid that possibility.

In some cases, the interdependent files may work well when only one is upgraded. The testing process may identify interdependent files that will not work together unless you upgrade both of them.

# Comparing new files to old files

The ServiceCenter Upgrade Utility uses *digital signatures* to determine if an application matches the original HP OpenView ServiceCenter application. Digital Signature enables an entire object to be identified.

The ServiceCenter Upgrade Utility includes the digital signatures of every object from all ServiceCenter releases after SC4. It compares the signature of each object in your file system against the entire library of HP OpenView ServiceCenter signatures from the previous releases. A matching signature indicates that the object in your system is unchanged from the original object and that the utility can upgrade it safely.

If a signature does not match the object in your system, the Upgrade Utility copies the new object to your system, but does not overwrite the existing object. You can examine both versions during the resolution process later.

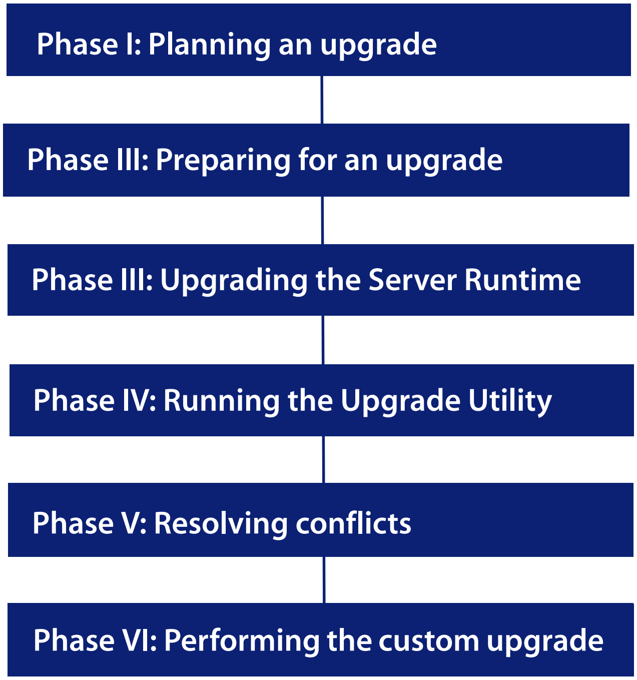
**Note:** The Upgrade Utility does not check the data written to non-system files, such as Incident, Contact, or Request records.

# Ongoing customization and upgrades

Do not modify your ServiceCenter production system during the upgrade process. Changes made to your production system after you create mirror image development and test environments are not captured by the custom upgrade. If you change your production system, you must restart the upgrade process.

# Upgrade phases

These are the main steps in the application upgrade process. If you are viewing this document online, you can click each step to link to a detailed explanation of that step.


Phase I: Planning an upgrade
Phase III: Preparing for an upgrade
Phase III: Upgrading the Server Runtime
Phase IV: Running the Upgrade Utility
Phase V: Resolving conflicts
Phase VI: Performing the custom upgrade

These steps, followed in sequence, help you complete a successful application upgrade:

- Read the Upgrade guide and use it to plan the upgrade carefully.
- Prepare a development environment that is a mirror image of your production environment, including operating system, database platform, and ServiceCenter system.
- Run the out-of-box upgrade in the development environment.
- Resolve all differences between your modified applications and new applications.
- Build a custom upgrade that integrates your customizations with the new applications.

- Prepare a test environment that is a mirror image of your production environment, including operating system, database platform, and ServiceCenter system.
- Apply the custom upgrade in the test environment.
- Test the new system independently. If you thoroughly test your custom upgrade, you will have little need to do cleanup.
- Apply the tested custom upgrade in the production environment.

CHAPTER 1

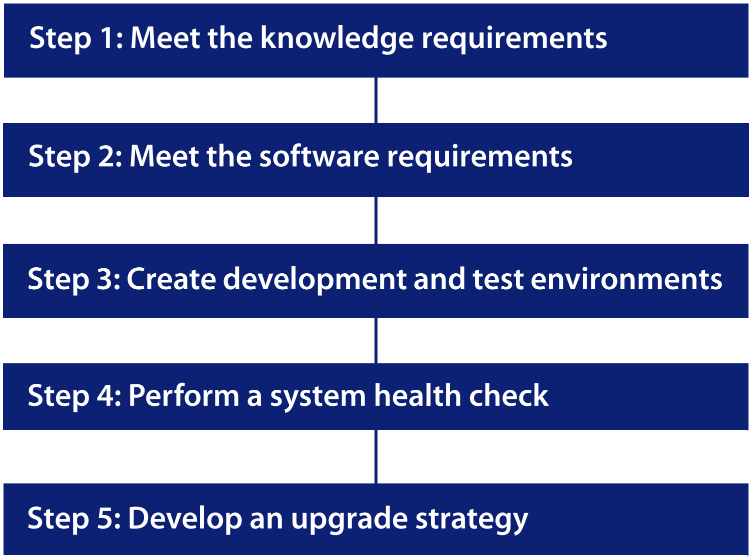
# Phase I: Planning an upgrade

Good planning allows your upgrade to run as smoothly and quickly as possible, and helps to avoid retracing your steps.

Topics in this section include:

# Planning overview

These are the main steps in planning an application upgrade. If you are viewing this document online, you can click each step to link to a detailed explanation of that step.



Step 1: Meet the knowledge requirements

Step 2: Meet the software requirements

Step 3: Create development and test environments

Step 4: Perform a system health check

Step 5: Develop an upgrade strategy

**Important:** Hewlett-Packard strongly recommends that you read the entire Upgrade guide before embarking on an application upgrade.

# Step 1: Meet the knowledge requirements

You should be an experienced ServiceCenter system administrator who is familiar with your installation to manage the upgrade. You should know the following:

- How the ServiceCenter file system operates
- How the application files function
- How to compare records within ServiceCenter
- How to use ServiceCenter's Rapid Application Development (RAD) environment
- How to program in RAD for systems with customized RAD applications

**Note:** If you are not familiar with RAD and administering a system with customized RAD applications, contact the RAD programmer who maintains your system modifications.

If you do not have the administrative experience necessary to manage the upgrade, you may need assistance from your local application developers and database administrators. You can also contact HP OpenView Customer Support.

## ServiceCenter tools

The utilities you will use most during the upgrade process include:

- Database Manager
- Database Dictionary
- Display application
- Forms Designer
- System Definition

If data resides in an RDBMS:

- You should be knowledgeable about that RDBMS. If not, ask for assistance from your database administrator.
- You should know how the ServiceCenter file system interacts with the database that stores your ServiceCenter data.

## Documentation resources

You need access to the following ServiceCenter documentation.

- See the ServiceCenter 6.2 on-line Help for information about database dictionaries, Database Manager, Forms Designer, Request Management and the Display application. You can access the Help when you install a client or the Help server.
- See the *ServiceCenter Database Conversion and RDBMS Support* guide if your data maps to an RDBMS. You can find a PDF version in the ServiceCenter 6.2 Help.

- See the *ServiceCenter Installation Guide* for client/server installation instructions. HP OpenView bundles this guide with the installation media.
- See the *ServiceCenter Release Notes* for information about Software Change Requests (SCRs), enhancements, and known issues. The release notes are part of the installation materials and also available on the Customer Support Web site.

**Note:** For an upgrade to earlier versions, review the documentation for that version.

## HP OpenView Customer Support Web site access

The HP OpenView Customer Support Web site has operating system and compatibility information, product documentation, and release notes. If you do not have an account for this Web site, contact HP OpenView Customer Support at: *www.hp.com/managementsoftware/support.*

## Authorization code

Your authorization code is in the `sc.ini` file, located in the ServiceCenter RUN directory. If you do not have a valid authorization code, contact HP OpenView Customer Support.

# Step 2: Meet the software requirements

Before starting your upgrade, meet the following ServiceCenter system requirements:

- Your RDBMS version, operating system, and client/server environment must meet all criteria listed in the Compatibility Matrix for the target version. See the HP OpenView Customer Support Web site to review the Compatibility Matrix.
- Your current ServiceCenter application release level must be SC4 or later. If your production system is on a pre-SC4 release, contact HP OpenView Customer Support for the best upgrade path.
- Your ServiceCenter application release must be English only. To upgrade localized versions of ServiceCenter, use the localized application upgrade.
- The ServiceCenter server process (scenter) must have read-write access to the database, or your database administrator must make all required database changes.

# Shared memory

HP OpenView recommends at least 96 MB of shared memory for the upgrade to run. However, if you have a large database, you may need to allocate more shared memory to accommodate it during the upgrade process. You can increase shared memory by using the shared memory parameter in the ServiceCenter initialization file, `sc.ini`.

# Disk space

You must have enough free disk space available to run the Upgrade Utility, build the custom upgrade, and store the upgrade files. If enough disk space is not available, the upgrade will fail.

## Running the Upgrade Utility

You need space for ServiceCenter files and enough work space to build the upgrade. The scdb.db1 file in your ServiceCenter file system increases to 200 MB during the upgrade process because of the amount of new information loaded by the utility. The scdb.asc file may increase up to 16 MB. Some of this space can be reclaimed after the upgrade is complete.

### Building a custom upgrade

The amount of customization determines the amount of space required to build the custom upgrade. You need:

- 75 MB of free space for light customization.
- 150 MB of free space for moderate customization.
- 200 MB of free space for heavy customization.

### Upgrade Utility files

You must have enough free space to accommodate the Upgrade Utility files. These files require 56 MB of free space.

| File | Size in KB | Contents |
|---|---|---|
| AppUpgVersion.txt | 2.1 | Contains Upgrade Utility version and build numbers to help you identify which application upgrade version you have available. |
| preupg.bin | 765.4 | Applications and data needed to make changes to the system before the upgrade. |

| File | Size in KB | Contents |
|---|---|---|
| self_service.unl | 8.3 | Profile, template, and user role to enable Self Service ticketing |
| sqlupgrade.unl | 53.7 | Applications and data needed to determine new fields that you must add to P4 and RDBMS databases. |
| transfer.bin | 3,152.3 | The ServiceCenter Upgrade Utility and all supporting objects. |
| upgdbdct.dta | 417.6 | Temporary dbdicts needed for the SQL Compare process. |
| upglang.unl | 303.3 | Applications and formats used to prepare a system for multi-language support. |
| upgrade.dta | 42,521.6 | Upgrade data for all information except dbdicts. |
| upgrade.inf | 16,098.9 | Upgrade definition file (includes a description of what to upgrade and lists acceptable old signatures). |
| upgrade.mak | 11.0 | `signature` file unload (needed on the target machine). |
| upgrade.str | 1,007.7 | Required database dictionary upgrades. |
| upgrade.ver | 1 | Version stamp for this upgrade. |

## Reclaimed space

You can reclaim some space at the end of the upgrade process by running the ServiceCenter Database Utility (LFMAP) on your scdb.db1 file (pool 3). For more information about LFMAP, see the ServiceCenter 6.2 Help.

After you successfully upgrade your production system, you can delete the Upgrade Utility files to reclaim the 56 MB that they require.

**Note:** It is not necessary to do this if you are fully mapped to an RDBMS.

# Backup requirements

You must keep backups of your original production environment, as well as the development and test environments. HP OpenView recommends that you back up files as you resolve differences and build the custom upgrade. If your development and testing activity is on a single system, frequent backups create checkpoints that you can go back to whenever necessary.

### NFS-mounted partitions

Do not install either ServiceCenter or the ServiceCenter Upgrade Utility on an NFS-mounted remote partition. This can cause serious performance degradation. The performance of an NFS-mounted partition drops significantly if it reads data in many small pieces instead of one large chunk. ServiceCenter generates a lot of database read/write activity. An NFS-mounted partition is significantly slower than a local drive when running the ServiceCenter Upgrade Utility process.

## Step 3: Create development and test environments

In addition to your production system, plan to have:

- A development system that mirrors your current production environment. Use the development system to run the Upgrade Utility and build a custom upgrade. This system should not be on the same machine as the production server.
- A test system that mirrors your current production environment. Run the custom upgrade on the test system and verify it there. This system may be on the same machine as the production server.

HP OpenView recommends that you export the custom upgrade to a test environment and run acceptance tests on this version before you move to live production. However, some users choose to run acceptance tests in their development environment. When testing is complete, you can export the custom upgrade to your production environment.

## Step 4: Perform a system health check

A well-maintained production system is the easiest to upgrade. Before starting the upgrade process, perform all regular maintenance on your production system. If necessary, contact HP OpenView Customer Support for recommended actions. Suspend all customization activity on the production system.

## Step 5: Develop an upgrade strategy

In standard ServiceCenter terminology, customization refers to changes to RAD applications; tailoring refers to changes made by using ServiceCenter tailoring tools, such as Forms Designer and Format Control. Configuration refers to local settings in your environment records, system information record, and so on.

The upgrade process affects different parts of the ServiceCenter system. Besides upgrading the standard ServiceCenter applications, an upgrade can affect the RDBMS where ServiceCenter is running as well as any customized files or RAD applications. For more information, see How does customization affect the upgrade process? on page 14.

## Tailored systems

A list of tailored files can help you resolve differences quickly between your existing files and new files. You can also use the SQL Compare utility to determine how files differ.

## RDBMS-mapped systems

If your ServiceCenter P4 data files are mapped to an RDBMS, you must choose one of the following options before beginning the upgrade:

- Allow the Upgrade Utility to modify your RDBMS tables for you. This is highly recommended, unless you are running DB2®.
- Use SQL Compare to update both the P4 database dictionaries and the RDBMS databases before beginning the upgrade process.
- Convert your database back to P4 before beginning the upgrade. When the upgrade is complete, follow the steps in the *ServiceCenter Database Conversion and RDBMS Support* guide to reconvert your data files to the RDBMS format.

The upgrade can affect certain mappings and tables. Contact your database administrator for assistance and to discuss the impact on the RDBMS.

## Customized RAD applications

A list of customized RAD applications and the extent of the customization is useful. If it is not available, the RAD programmer who made the changes may be able to supply information. Or, you may need to run a comparison between the existing application and the new version.

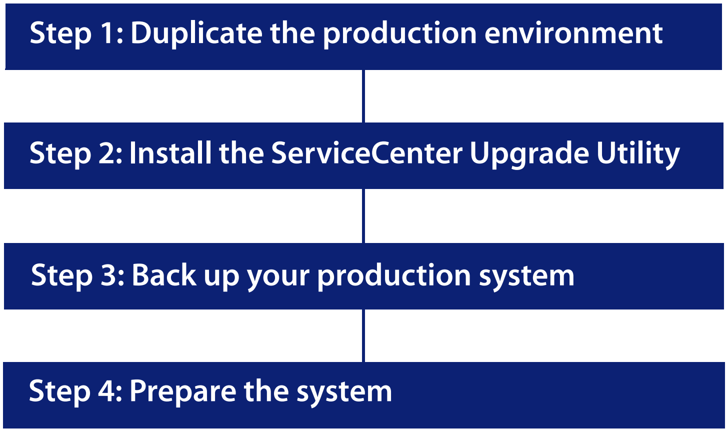
# 2 CHAPTER Phase III: Preparing for an upgrade

Before you can develop or test a custom upgrade, you need to create a mirror image of your current ServiceCenter production environment and prepare the data for the upgrade process.

Topics in this section include:

## Preparation overview

These are the main steps in preparing for an application upgrade. If you are viewing this document online, you can click each step to link to a detailed explanation of that step. These steps do not apply if you are doing a server upgrade only.



Step 1: Duplicate the production environment

Step 2: Install the ServiceCenter Upgrade Utility

Step 3: Back up your production system

Step 4: Prepare the system

# Step 1: Duplicate the production environment

To achieve the best results, develop and test the custom upgrade on a system that resembles your production environment as closely as possible.

### To duplicate the production environment

1 Identify a server to use for the development and test environments.

- **Unix:** You can copy the files to a new location on your production machine.
- **Microsoft Windows™:** You must create the development system on a different machine from your production system.

2 Ensure that adequate memory and disk space is available and accessible. Frequent backups are necessary.

3 Ensure that your development and test systems meet all upgrade requirements. For more information, see Step 2: Meet the software requirements on page 22.

- Upgrade your RDBMS to a version compatible with ServiceCenter 6.2. See the HP OpenView ServiceCenter 6.2 compatibility matrix.
- Convert your RDBMS codepage to Unicode. See your RDBMS vendor documentation.

4 Decide how you want to manage your RDBMS, as described in RDBMS-mapped systems on page 26.

5 Set up the environment of your development and test machines to match your production server exactly.

- The operating system version and service pack level should match.

6 Install the same version of the ServiceCenter client and server at the production system onto your development machine. You do not need to install the DATA files.

7 If you are doing an application upgrade, go to Step 2: Install the ServiceCenter Upgrade Utility on page 29.

8 If you are doing an RTE upgrade only, add a new Upgrade subdirectory to the primary directory in the development or test system. It should be on the same level as the RUN and DATA directories.

9 Go to Step 3: Back up your production system on page 30.

# Step 2: Install the ServiceCenter Upgrade Utility

Follow these steps when you set up your development and test systems, and when you prepare your production system for an application upgrade.

**Important:** Do not try to run the upgrade directly from a CD-ROM. The upgrade needs read and write access to its files.

### To install the ServiceCenter Upgrade Utility

1. Add new subdirectories to the primary directory in the development or test system. They should be on the same level as the RUN and DATA directories.

   For an application upgrade, add

   - Upgrade
   - Backups
   - CustomUpgrade

2. Ensure that ServiceCenter server process (scenter) has write and execute privileges for these directories.

3. Copy the ServiceCenter Upgrade Utility files to the Upgrade directory.

4. Remove read-only permissions from the files.

## Step 3: Back up your production system

After duplicating your production environment in a development system, you will make two baseline backup copies of your production system DATA directory. One is a permanent archive; the other becomes the DATA for your development or test system.

**Important:** If mapped to an RDBMS, you must back up your RDBMS database whenever you back up your data. Refer the documentation for your relational database management system (RDBMS) for RDBMS backup instructions.

There are two backup methods:

- Cold backup
- Hot backup

## Cold backup

### To make a cold backup of your production system

1 Stop the ServiceCenter production server.

- **Unix:** Run the scstop script in your ServiceCenter directory from the operating system command prompt.
- **Windows:** Open the ServiceCenter server console and click **Stop**.

2 Copy all files that follow the naming convention scdb.* in the production DATA directory to the Data directory in the development or test systems. This copy of the DATA directory files replaces the out-of-box version, if you installed it in Step 6 on page 29.

3 Archive a backup copy of the Data directory in case you must restore it later. Frequent checkpoint backups of the data enable you to retrace your steps if necessary. You can compress the backup and store it on a CD-ROM or other storage media, or store a copy in the Backups directory.

4 Restart your production system.

- **Unix:** Run the scstart script from the operating system command prompt in your ServiceCenter directory.
- **Windows:** Open the ServiceCenter Console and click **Start**.

## Hot backup

The difference between a hot backup and a cold backup is whether the ServiceCenter server is running when you copy the data. Do not use the hot backup method on a system mapped to an RDBMS.

### To make a hot backup of your production system

1 Start the logging process by issuing the following command at the operating system command prompt.

```
scenter -startlogging
```

2 Verify that logging is enabled by viewing the status report. Issue the following command at the operating system command prompt.

```
scenter -infologging
```

3 Copy all files that follow the naming convention scdb.* in the production DATA directory to the Data directory in the development or test systems. This copy of the DATA directory files replaces the out-of-box version, if you installed it in Step 6 on page 29.

4 Archive a backup copy of the Data directory in case you must restore it later. Frequent checkpoint backups of the data enable you to retrace your steps if necessary. You can compress the backup and store it on a CD-ROM or other storage media, or store a copy in the Backups directory.

5 Stop the logging process by issuing the following command at the operating system command prompt:

```
scenter -stoplogging
```

6 Verify that stoplogging was successful by viewing the status report.

# Step 4: Prepare the system

Follow these steps when you set up your development and test systems, and when you prepare your production system for an application upgrade.

**Note:** These steps do not apply if you are doing an RTE upgrade only. Skip ahead to Phase III: Upgrading the Server Runtime Environment on page 35.

### To prepare a system for an upgrade

1 Edit the `sc.ini` of the server you installed in Step 6 on page 29 to make it connect to the data you just copied to the development or test system.

2 Start that server.

3 Log into it using the client you installed in Step 6 on page 29.

4 If you are running your upgrade from the server machine, disable **Client-side Load/Unload**. If you are running your upgrade from a client on another machine, enable **Client-side Load/Unload.** To determine whether **Client-side Load/Unload** is enabled:

- For versions prior to SC 6.0, check **File** > **Client-side Load/Unload**.
- For version 6.0 and later, open **Window** > **Preferences** and check **ServiceCenter** > **Client-side Load/Unload**.

5. Disable development auditing (revision tracking), if necessary.

6. Purge any files left over from upgrades to previous versions.

   - Type `*aapm.upgrade.purge` on the ServiceCenter client command line. Press **Enter**.
   - Select **I'm done, and I want to remove the upgrade files completely.**
   - Click **OK**.
   - Click **OK** again to exit the Purge Routine Completed dialog box.

7. Run LFMAP, Option 4. Although the upgrade data has been purged, free space is not reclaimed until you run LFMAP. For help running LFMAP, see the documentation for your version of ServiceCenter.

8. Record all IR keys and then delete them. You can then manually restore them after the upgrade.

9. If you are upgrading from a version prior to 6.0, ensure that there are no sqlhints for the **Number** field for the Change Management files cm3r, cm3t, cm3rpage, and cm3tpage. If there are, delete them before running the upgrade. During the re-mapping, the data from these files exists in P4 temporarily.

   Also ensure that these files reside in data pools that will not exceed the 2-GB limit when combined with the Change Management data. You may need to extend them over multiple pools.

10. If you are using an RDBMS, ensure your RDBMS account has administrator rights to the database and that files have room for unrestricted growth.

11. Make a checkpoint backup of the data files to enable you to restore from this point, if necessary. See Step 3: Back up your production system on page 30.

# 3 CHAPTER Phase III: Upgrading the Server Runtime Environment

You can upgrade the server run-time environment (RTE) separately from the ServiceCenter applications. If you are only upgrading your server RTE, you do not need to create and apply a custom upgrade or do conflict resolution.

Topics in this section include:

## Upgrade options

Depending upon what version of ServiceCenter you are upgrading from you may be required to upgrade your applications to produce a supported production configuration. The following table lists the upgrade options available with ServiceCenter 6.2.

| Previous version of ServiceCenter | Upgrade options |
|---|---|
| 6.x | ■ Upgrade server RTE to 6.2 and run 6.x applications<br>■ Upgrade server RTE to 6.2 in preparation for upgrading applications to SC6.2 |
| 4.x to 5.x | Upgrade server RTE to 6.2 in preparation for upgrading applications to SC6.2 |

If you need to upgrade your applications, go to Phase I: Planning an upgrade on page 19.

# Tasks for upgrading your server to 6.2

The following tasks are for upgrading ServiceCenter 4x-6.x servers to a 6.2 server RTE. Contact Customer Support to upgrade servers from pre-4.x versions of ServiceCenter.

After you upgrade your server RTE, you can continue running your 6.x applications or upgrade your applications to 6.2 to take advantage of all the latest features.

**Important:** You cannot run a 6.2 server with pre-6.0 applications.

**Step 1** Ensure that you have completed Step 1: Duplicate the production environment on page 28.

**Step 2** Ensure that you have completed Step 3: Back up your production system on page 30.

**Step 3** Prepare the development environment. See Preparing the development environment on page 37.

**Step 4** Convert pre-SC6 data files to UTF-8. See Converting data files to UTF-8 in pre SC6 systems on page 38. Skip this step if you are upgrading a SC6 or SC61 system.

**Step 5** Update the ServiceCenter initialization file. See Updating the ServiceCenter ini and cfg files on page 42.

**Step 6** Mark SC6 or SC61 data as already converted to UTF-8. See Marking 6.0.x and 6.1.x files as converted to UTF-8 on page 43. Skip this step if you are upgrading a pre-SC6 system.

**Step 7** Convert pre-6.1.x attachments to the new table format. See Converting attachments on page 44. Skip this step if you are upgrading a 6.1system.

**Step 8** Convert pre-6.1.x favorites to inboxes. See Converting favorites on page 50. Skip this step if you are upgrading a 6.1.x system.

**Step 9** Update the system status and inactivity timer applications. See Updating the system status and inactivity applications on page 56.

**Step 10**Make a checkpoint backup of the data files to enable you to restore from this point, if necessary. See Step 3: Back up your production system on page 30.

If you want to continue using ServiceCenter 6.x applications in your production environment, then you can replace the ServiceCenter 6.x production server with the upgraded 6.2 development environment.

# Preparing the development environment

Hewlett-Packard recommends first deploying ServiceCenter 6.2 on a development environment to review new features and ensure system stability.

1 If you have mapped your ServiceCenter operator table to an RDBMS, you must update the SQL mapping for the password field to accept a larger character limit. Refer to your vendor documentation for instructions.

   The 6.2 server automatically encrypts operator passwords the first time it accesses each operator record. The 6.2 encryption scheme requires a larger character limit than previous versions of ServiceCenter. The new data policy settings are as follows.

| RDBMS object | Requirement |
|---|---|
| Table | `operatorm1` (default) |
| Field | `password` |
| Datatype | VARCHAR (VARCHAR  2) for Oracle |
| Size | 136 characters or greater |

You must update the SQL mapping **before** logging into the upgraded version of ServiceCenter. ServiceCenter will refuse to open the operator file if the password field is to small, which results in failure to login. There will be a message like this in the sc.log file: "!!! column PASSWORD is too short (60) to hold encrypted password, should be 136 characters."

2 Perform a custom install of the ServiceCenter 6.2 server on the development system. Do not install the sample database.

See the *ServiceCenter  Installation Guide* for instructions.

3 Install a ServiceCenter 6.2 client on the development system or an available desktop system.

See the *ServiceCenter  Installation Guide* for instructions.

4 Put a copy of the original data files in the development system.

See the your RDBMS documentation for restore instructions.

**Important:** Do not start the ServiceCenter server yet.

# Converting data files to UTF-8 in pre SC6 systems

If you are upgrading from a 6.0.x or 6.1.x release the data files are already converted to UTF-8. Skip ahead to Marking 6.0.x and 6.1.x files as converted to UTF-8 on page 43.

Beginning with ServiceCenter 6.0, the server stores all application and system files in UTF-8 character encoding. If you are upgrading from a pre-SC6 system, you must convert your existing data to UTF-8 format before you can run the ServiceCenter Upgrade Utility.

If your ServiceCenter data is on a P4 database, then the UTF-8 conversion process converts all of your data to UTF-8. If you have pushed your ServiceCenter data to a n RDBMS, the UTF-8 conversion process only converts data mapped to binary columns. The ServiceCenter data mapped to character-based columns remains in the codepage of your RDBMS (as defined by the `dblanguage` parameter). If you want to convert your user data to UTF-8, see your RDBMS documentation for information.

# Conversion process options

As of ServiceCenter 6.2, you can run the ServiceCenter server at the same time as the UTF-8 conversion process. This allows you to convert a production system to UTF-8 without having to take the server out of production. If during the conversion process a process requests an unconverted resource, the server automatically converts the requested resource to UTF-8.

**Note:** The server's response performance will be lower while the UTF-8 conversion process runs.

## Dedicated process with outage

You can run the UTF-8 conversion as one or more dedicated processes that use all available system resources for the conversion. Running dedicated processes requires a ServiceCenter system outage.

## Multiple background processes

If you have pushed your ServiceCenter data to an RDBMS, you can decrease the UTF-8 conversion time by running multiple background processes. If your ServiceCenter data is on P4, you must use a single process conversion.

1 Ensure that you have a backup copy of the data in case you must restore it later. See Step 3: Back up your production system on page 30.

**Important:** Conversion to UTF-8 is not reversible. Hewlett-Packard strongly recommends that you make a backup copy of your database files prior to UTF-8 conversion.

2 To run the conversion as a dedicated process, stop the ServiceCenter server.

To run the conversion using multiple background processes, do not stop the server.

3 Open an operating system command prompt, and navigate to the ServiceCenter RUN directory of the 6.2 server. For example on a default Windows installation the path is:

```
cd C:\Program Files\Peregrine Systems\ServiceCenter
6.2\Server\RUN
```

4 Type one of the following commands:

| Option | Command |
|---|---|
| Single process conversion | `scenter -converttoutf8` |
| Multiple process conversion | `scenter -converttoutf8:`*<number>* |

For *<number>*, type the number of background processes you want to use for UTF-8 conversion on an RDBMS.

**Tip:** Hewlett-Packard Development Company, L.P. recommends two background processors for each CPU available on the system.

The time it takes to complete the conversion process depends on the amount of data to be converted. Check the ServiceCenter system log for a confirmation message that the conversion is complete.

After the `converttoutf8` utility completes, you have a ServiceCenter 6.2 server that can run with your old applications.

5 Make a checkpoint backup of the data files to enable you to restore from this point, if necessary. See Step 3: Back up your production system on page 30.

# Verifying UTF-8 conversion from 6.0.x servers

If you are upgrading from ServiceCenter 6.0.x and you have been using Oracle or DB2 to store ServiceCenter data, you may need to verify the UTF-8 conversion. This step is not necessary for ServiceCenter versions prior to 6.0.x or for ServiceCenter 6.0.x servers where the data resided on either SQL Server or P4 databases. Likewise, it is not necessary to verify the conversion if your Oracle or DB2 tables were already using UTF-8 codepages.

1 Open an operating system command prompt, and navigate to the ServiceCenter RUN directory. For example on a default Windows installation:

```
cd C:\Program Files\Peregrine Systems\ServiceCenter
6.2\Server\RUN
```

2 Type the following command:

```
scenter -verifyutf8
```

The ServiceCenter server scans the tables mapped to your RDBMS and repairs certain character sequences that ServiceCenter 6.0.x did not convert properly. The command creates a report of any data that it repaired or that it could not repair.

If your RDBMS uses Windows 1252 codepages, ServiceCenter 6.2 repairs any incorrect character sequences, provided there are no truncation errors from storing the column data in the RDBMS tables. You can avoid truncation errors by defining CHAR or VARCHAR fields with adequate field lengths.

If your RDBMS uses ISO codepages, it is likely that the verify utility cannot repair some characters. Contact HP OpenView Customer Support with a list of all UTF-8 conversion errors.

3 Make a checkpoint backup of the data files to enable you to restore from this point, if necessary. See Step 3: Back up your production system on page 30.

# Updating the ServiceCenter ini and cfg files

The ServiceCenter The upgrade process requires several specific initialization file parameter values to run properly. Before running the upgrade utility add or update the following startup parameters:

| sc.ini setting | Description |
|---|---|
| path:<path to development system> | Required to connect to development system |
| substrclassic:1 | Required to update the load/unload utility |
| usethreading:0 | By default, ServiceCenter 6.2 server runs in multi-threaded mode. If you will be converting data to or from an RDBMS during the upgrade process, you need to use a single thread. |

1 To prevent the schedulers from starting, open the ServiceCenter 6.2 server's configuration file (`sc.cfg`) and insert a comment character at the beginning of all statements except one `scenter -listener` statement.

- Edit the sc.cfg file for both Windows and UNIX.

2 Update the ServiceCenter 6.2 server's initialization file (`sc.ini`) to use the data files on the development system.

If your data is in P4 files, update the `path` parameter to indicate the location of your copied P4 data files. If your data is on an RDBMS, update the `path`, `sqldb` and `sqllogin` parameters to indicate the connection information of your restored database.

3 Add the `substrclassic:1` parameter to `sc.ini`.

4 Add the `usethreading:0` parameter to `sc.ini`, if appropriate. Remove it when you are done moving data to or from your RDBMS.

5 Start the server.

6 Log into the server using the ServiceCenter 6.2 Client.

7 Open Database Manager, and load *<SC Installation Path>\serverupgrade\6.xfileload_list.unl*.

8 Log out of the client.

9 Stop the server.

10 Remove the `substrclassic:1` parameter from `sc.ini`.

11 Restart the server.

# Marking 6.0.x and 6.1.x files as converted to UTF-8

ServiceCenter 6.2 uses a new record value to indicate that you have already converted your RDBMS data files to UTF-8. If you converted your data files to UTF-8 from a ServiceCenter 6.0 or 6.1 system you must load a special utility to indicate that you have already converted your RDBMS data files to UTF-8. If you are migration from a preSC6 system, skip this step and go to Converting attachments on page 44.

**Note:** If your data is in P4 files you could skip this step, however, Hewlett-Packard recommends that you do it anyway. It will be necessary if you ever fully convert to an RDBMS.

1 Start the ServiceCenter 6.2 server.

2 Using a ServiceCenter 6.2 client, log on to ServiceCenter with an administrator account.

3 Click **Toolkit** > **Database Manager**.

4 Right-click the form and select **Import/Load**.

5 Type or select the following information

| Field | Description |
|---|---|
| **File Name** | Type the name and path of the attachment conversion utility:<br>*<ServiceCenter server installation path>*`\serverUpgrade\utf8_info_flag.unl` |
| **File Type** | Select the source operating system of the unload file. |
| **Messages Option** | Select **All Messages** to view all messages generated while loading the file. |

**Note:** You can view the contents of an unload file before importing it by clicking **List Contents**.

6 Click **Load FG**.

The server loads the UTF-8 information flag and supporting records.

7 Restart the server.

# Converting attachments

ServiceCenter 6.2 and later releases store attachments in a repository called SYSATTACHMENTS, which contains a separate record for each attachment. Earlier ServiceCenter versions stored all attachment files for a given record as a single binary unit in the SYSBLOB table. To display attachments stored in the SYSBLOB table, the ServiceCenter server had to retrieve the record, open the binary object, extract the desired attachment, and then display the attachment. The new method is simpler and faster.

To view existing attachments, you must migrate the contents of the SYSBLOB table to the SYSATTACHMENTS table. ServiceCenter has two ways of converting attachments.

- Run a background conversion process to convert all attachments over time. This is the preferred conversion method as administrators can schedule the conversion process and users can still access the system while the process is running. A system administrator can start and stop the background process as needed.

- Use on-demand conversion to convert only attachments in records that operators view or update. This secondary conversion method allows users access to all attachment records.

If your system contains OLE object formatted attachments, you have to manually convert them. See the knowledge article Reading OLE Attachments in ServiceCenter™ Versions 6.0 and 6.1 from the Customer Support Web site for more information.

## Loading the attachment conversion utility

Before you convert your attachments, load the new SYSATTACHMENTS table onto your system.

1 Start the ServiceCenter 6.2 server.

2 Using a ServiceCenter 6.2 client, log on to ServiceCenter with an administrator account.

3 Click **Toolkit** > **Database Manager**.

4 Right-click the form and select **Import/Load**.

5 Type or select the following information

| Field | Description |
|---|---|
| **File Name** | Type the name and path of the attachment conversion utility:<br>*<ServiceCenter server installation path>*`\serverUpgrade\sysattachments.unl` |
| **File Type** | Select the source operating system of the unload file. |
| **Messages Option** | Select **All Messages** to view all messages generated while loading the file. |

**Note:** You can view the contents of an unload file before importing it by clicking **List Contents**.

6 Click **Load FG**.

The server loads the attachment conversion utility and necessary tables.

7 If your system is mapped to an RDBMS, the server prompts you to indicate whether you want ServiceCenter to automatically update the RDBMS or to have your database administrator perform the update.

### Reviewing SQL mappings

If you store ServiceCenter data on an RDBMS, ensure that the SYSATTACHMENT table has the mapping properties listed below.

**Note:** You must have administrative access to the RDBMS to change these settings.

- The **data** column of the SYSATTACHMENT table must be able to store pure binary data. This column should have a data type of either BLOB or image.
- The sqlmapping record for the **data** column must have a value of true for the **SQL RC** column.
- The **filename** column of the SYSATTACHMENT table should have a character data type sufficiently large enough to contain the longest attachment file name in your system, such as for example, varchar(256).

### Reviewing P4 attachment pool size

If you store ServiceCenter data on a P4 database, the server stores the SYSATTACHMENT table in pool 3 after loading the new dbdict. You should verify that this pool has sufficient space to store all the attachments in your system. If needed, create additional pools to store the attachments in your system or move the table to its own pool.

See the *ServiceCenter  Database Conversion and RDBMS Support Guide* for more information about data pools.

**Note:** You must have administrative access to the ServiceCenter server operating system to change the pool settings.

## On-demand conversion

If an operator views the Attachment tab on a record containing an unconverted attachment, then the server converts all attachments in that record and saves them in the new SYSATTACHMENTS table. This on-demand conversion only occurs when using ServiceCenter clients and the HP OpenView Web Services API.

# Background conversion

You can start a dedicated ServiceCenter process to run the attachment conversion utility. This background process converts attachments while the server is running. While the server accepts client connections during the conversion process, users may notice decreased system performance. If the conversion process stops, it can resume where it left off when you restart the attachment conversion utility.

The time the conversion process takes to complete depends on the amount of data to be converted and on the system performance of your RDBMS and ServiceCenter server. Conversion times generally decrease if you run the background process during non-peak hours. A general performance metric is 25,000 attachments converted in an hour of run time. A million attachments in a large system could take 40 hours of run time.

Warning: Do not add, change, or delete records from the SYSBLOB or SYSATTACHMENTS tables with anything other than the attachment conversion utility or a ServiceCenter client during the background conversion process. You can lose data if any SQL tool or customer-written application changes the SYSBLOB or SYSATTACHMENTS tables.

## Setting attachment segment size

The ServiceCenter server divides each attachment into segments based on the value of the `attachmentsegmentsize` parameter. It is important that you determine the desired attachment segment size prior to conversion because once the server converts an attachment, the only way to change the segment size is to manually edit and save every converted attachment again. If you prefer to keep attachments in as few segments as possible, then set the attachment segment size to its 2-GB maximum value. See the ServiceCenter 6.2 Help for information about this parameter.

1 Stop the ServiceCenter 6.2 server.

2 Open an operating system command prompt, and navigate to the ServiceCenter RUN directory. For example on a default Windows installation:

```
cd C:\Program Files\Peregrine Systems\ServiceCenter
6.2\Server\RUN
```

3 Open the ServiceCenter initialization file (sc.ini) in a text editor.

4 Add the attachmentsegmentsize parameter. For example:

```
attachmentsegmentsize:2147483647
```

See the ServiceCenter 6.2 Help for information about this parameter.

5 Save the initialization file.

6 Restart the ServiceCenter 6.2 server.

## Starting the attachment conversion process

Hewlett-Packard Development Company, L.P. recommends that you start the attachment conversion process from an operating system command prompt so that you can include additional parameters such as logging settings.

1 Stop the ServiceCenter 6.2 server.

2 Open an operating system command prompt, and navigate to the ServiceCenter RUN directory. For example on a default Windows installation:

```
cd C:\Program Files\Peregrine Systems\ServiceCenter
6.2\Server\RUN
```

3 Type the following command to start the ServiceCenter server and the attachment conversion process with a dedicated log file:

```
scenter -convertsysblob -log:..\logs\convertsysblob.log
-numberoflogfiles:15 -maxlogsize:5242880
```

ServiceCenter converts attachments from the SYSBLOB table to the SYSATTACHMENTS table. During the conversion process, the server deletes records from the SYSBLOB table and adds records to the SYSATTACHMENTS table. If the background process cannot convert a record, it preserves the record in the SYSBLOB table and creates a log entry of the error.

**Note:** You can stop and restart the attachment conversion process as needed. If you stop the process, it resumes converting attachments from where it last stopped.

After completion, the server displays the message:

```
SYSBLOB file conversion to SYSATTACHMENTS completed
successfully.
```

## Verifying the attachment conversion process

You have two ways to verify and monitor the progress of the attachment conversion process:

- View the contents of the SYSBLOB and SYSATTACHMENTS tables
- Review the log file

You can use a ServiceCenter Database Manager to review the contents of the SYSBLOB and SYSATTACHMENT tables.

As the conversion progresses, the number of records in the SYSBLOB table should decrease, while the number of records in the SYSATTACHMENTS table should increase. Records in the SYSBLOB table with a segment number of zero indicate attachments still to be converted. Records in the SYSATTACHMENT table with a segment number of zero indicate attachments that have already been converted.

If you started a ServiceCenter process with a dedicated log file, you can review the log file for periodic updates. Review the log file you started to see what error messages the server produced. If your system contains any attachments with OLE objects, the log file will indicate the table and record containing these attachments. During a normal conversion process, you should see a series of informational messages regarding attachment conversions.

**Note:** Some earlier versions of ServiceCenter stored empty structures in the SYSBLOB table. The background process reports these empty structures and deletes them during conversion.

If the conversion is successful, you should have no remaining records in the SYSBLOB table. You may safely delete the SYSBLOB table after conversion. If the SYSBLOB table still contains records after the conversion, then note the values of the **application** and **topic** fields for these records. You can use these values to determine the associated table and record for the attachment.

If the associated records for these attachments no longer exist, then you can safely delete the SYSBLOB record. If the associated records still exist, then you can use the following steps to reconcile them.

1 Log on to the production server hosting your previous version of ServiceCenter.

2 Search for a record containing an unconverted attachment.

   The application name identifies the related record table name and the topic identifies the unique key of the related record containing the attachment.

3 Open the attachments in the record and save them on a local hard drive.

4 Delete the existing attachments from the record.

5 Add the attachments back to their proper records on your new ServiceCenter system.

   See the ServiceCenter 6.2 Help for information on adding attachments.

# Converting favorites

Earlier versions of ServiceCenter used inboxes to launch reusable queries. With ServiceCenter 6.0, the server also offered favorites that overlapped with the existing inbox functionality. With ServiceCenter 6.2, the server now stores all data used by inboxes, charts, dashboards, and favorites in the inbox table. You can access inboxes by expanding the Favorites folder, or you can continue to access them using traditional ServiceCenter navigation.

The inbox and favorites conversion process migrates the contents of the `systext` table to the inbox table. This migration ensures that favorites from ServiceCenter 6.0 and inboxes from all earlier releases work seamlessly in ServiceCenter 6.2.

ServiceCenter 6.2 and later releases store all data used by inboxes, charts, and favorites in the inbox table. Run the inbox and favorites conversion process to ensure favorites from ServiceCenter 6.0 and inboxes from all earlier releases work seamlessly in ServiceCenter 6.2.

## Backing up existing inboxes and favorites

Before converting your favorites and inboxes, you should create an unload file consisting of the data in the `systext` and inbox tables.

See the ServiceCenter 6.2 Help for information on creating unload files.

## Loading the favorites conversion utility

Before conversion you must load the favorites conversion utility.

1 Start the ServiceCenter 6.2 server.

2 Using a ServiceCenter 6.2 client, log on to ServiceCenter with an administrator account.

3 Click **Toolkit** > **Database Manager**.

4 Right-click the form and select **Import/Load**.

5 Type or select the following information

| Field | Description |
|---|---|
| **File Name** | Type the name and path of the favorites conversion utility: *<ServiceCenter server installation path>*`\serverUpgrade\favorites_conversion.unl` |
| **File Type** | Select the source operating system of the unload file. |
| **Messages Option** | Select **All Messages** to view all messages generated while loading the file. |

**Note:** You can see the contents of an unload file before importing it by clicking **List Contents**.

6 Click **Load FG**.

The server loads the favorites conversion utility and necessary tables.

7. If your system is mapped to an RDBMS, the server prompts you to indicate whether you want ServiceCenter to automatically update the RDBMS or to have your database administrator perform the update.

### Reviewing SQL mappings

If you store ServiceCenter data on an RDBMS, ensure that the inbox table has the mapping properties listed below.

**Note:** You must have administrative access to the RDBMS to change these settings.

- The **query** column of the inbox table should have a character data type sufficiently large to avoid truncating long queries, such as varchar(256).
- The **normalized_query** column of the inbox table should have a character data type sufficiently large to avoid truncating long queries, such as varchar(256).
- The **short_query** column of the inbox table should have a character data type sufficiently large to avoid truncating long queries, such as varchar(256).
- The **short_normalized_query** column of the inbox table should have a character data type sufficiently large to avoid truncating long queries, such as varchar(256).
- The **inbox_name** column of the inbox table should have a character data type sufficiently large to avoid truncating long inbox names, such as varchar(256).

## Updating the unique key of the inbox table

To run properly, the favorites conversion utility requires changes to the inbox database dictionary file.

### ServiceCenter P4 database

The following instructions are only for systems where ServiceCenter data is on the P4 database.

1. Start the ServiceCenter 6.2 server.

2. Using a ServiceCenter 6.2 client, log on to ServiceCenter with an administrator account.

3. In the System Navigator, expand **System Definition** > **Tables**.
4. Open the **inbox** table.
5. Double-click **Keys**.
6. Select the existing unique key.
7. In the General section, select **Add** and type the following field to be a unique key: `container.id`.
8. Use the up and down options to arrange the unique keys in the following order:

   - `container.id`
   - `inbox.name`
   - `inbox.type`
   - `operator.name`

9. Click **Save**.

   ServiceCenter prompts you to regenerate the key.

10. Click **OK**.

    ServiceCenter regenerates the key and displays messages to indicate that the regen has completed successfully.

## RDBMS

On systems where ServiceCenter data is pushed to an RDBMS, you must have a database administrator make the following changes to the inbox table.

- Change the unique key to contain these fields in the following order:
  - `container.id`
  - `inbox.name`
  - `inbox.type`
  - `operator.name`

# Running the favorites conversion utility

After updating the inbox database dictionary, you can run the favorites conversion utility.

1 Start the ServiceCenter 6.2 server.

2 Using a ServiceCenter 6.2 client, log on to ServiceCenter with an administrator account.

**Important:** The user who runs the favorites conversion utility becomes the owner of all inboxes on the system, and as such will be able to view and edit all inboxes. If necessary, you may want to create a special administrator user to run the favorites conversion and own inboxes.

3 In the ServiceCenter command line, type the following command:

```
rad
```

The RAD Editor opens.

4 In the Application field, type the following:

```
convert.old.favorites
```

5 Click **Search**.

6 Click **Test**.

7 Click **Proceed**.

ServiceCenter converts the existing inboxes and favorites to the new table properties. When complete, the server displays the following message:

```
Favorites conversion complete.
```

8 Check for error messages in the Windows client messages view and in the server log file.

9 Make a checkpoint backup of the data files to enable you to restore from this point, if necessary. See Step 3: Back up your production system on page 30.

# Adding additional keys to the inbox table

After running the favorites conversion utility, you need to add additional keys to the inbox database dictionary.

## ServiceCenter P4 database

The following instructions are only for systems where ServiceCenter data is on the P4 database.

1 Start the ServiceCenter 6.2 server.

2 Using a ServiceCenter 6.2 client, log on to ServiceCenter with an administrator account.

3 In the System Navigator, expand **System Definition** > **Tables.**

4 Open the **inbox** table.

5 Double-click **Keys**.

6 Click **New** in the Keys section of the Fields and Keys tab.

7 In the General section, select **No duplicates** from the drop-down list in the **Type** field.

8 Click Add and select the following fields to be the **No duplicates** key:

9 Type `inbox.id.`

10 Click **Save**.

ServiceCenter prompts you to regenerate the key.

11 Click **OK**.

ServiceCenter regenerates the key and displays messages to indicate that the regen has completed successfully.

12 Click **New** in the Keys section of the Fields and Keys tab.

13 In the General section, select **Nulls and duplicates** from the drop-down list in the **Type** field.

14 Click Add and select the following fields to be the **Nulls and duplicates** key:

- `inbox.type`
- `inbox.class`
- `operator.name`
- `groups`

15 Click **Save**.

ServiceCenter prompts you to regenerate the key.

16 Click **OK**.

ServiceCenter regenerates the key and displays messages to indicate that the regen has completed successfully.

### RDBMS

On systems where ServiceCenter data is pushed to an RDBMS, you must have a database administrator make the following changes to the inbox table.

- Change the No Duplicates key to contain the following field:
  - `inbox.id`
- Change the Nulls and Duplicates key to contain the following fields:
  - `inbox.type`
  - `inbox.class`
  - `operator.name`
  - `groups`

## Updating the system status and inactivity applications

The ServiceCenter 6.2 server run-time environment requires new versions of the system status and inactivity timer RAD applications to use the latest load balancer, to use one of the new servlet implementations, and to stop inactive processes running in the new implementation options.

**Note:** You can run the 6.2 load balancer with 6.1 applications by loading `62systemstatus.unl` into your 6.1 applications.

1 Start the ServiceCenter 6.2 server.

2 Using a ServiceCenter 6.2 client, log on to ServiceCenter with an administrator account.

3 Click **Toolkit** > **Database Manager**.

4 Right-click the form and select **Import/Load**.

5 Type or select the following information

| Field | Description |
|---|---|
| **File Name** | Type the name and path of the attachment conversion utility:<br>*<ServiceCenter server installation path>*`\serverUpgrade\sc6systemstatus.unl` |
| **File Type** | Select the source operating system of the unload file. |
| **Messages Option** | Select **All Messages** to view all messages generated while loading the file. |

**Note:** You can view the contents of an unload file before importing it by clicking **List Contents**.

6 Click **Load FG**.

The server loads the updated system status and inactivity timer applications. The server then prompts you to indicate whether you want ServiceCenter to automatically update the RDBMS or to have your database administrator perform the update.

# 4 CHAPTER Phase IV: Running the Upgrade Utility

Now that you have a functional environment, you are ready to run the Upgrade Utility. Follow the steps in this chapter to run the out-of-box upgrade against the data in your development system and to run your custom upgrade against your test and production systems.

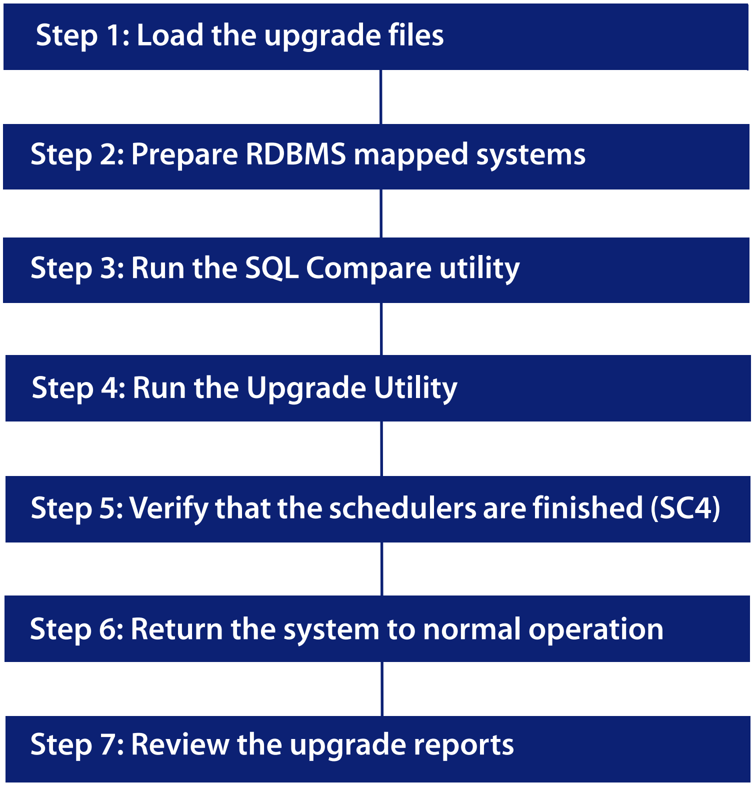
Topics in this section include:

## Overview

The ServiceCenter Upgrade Utility upgrades the Display components, the database dictionaries, and the application data. It also purges the upgrade files at the end of the process. If the upgrade process encounters any problems while upgrading the components of any of these sections, it stops automatically at the end of the affected section to alert you that it encountered problems.

Whenever the upgrade stops, you must exit the upgrade process and fix the identified problems. If the Upgrade Utility reports any problems with the database dictionary upgrade, resolve all issues before continuing the data phase of the upgrade. Whenever you restart ServiceCenter Upgrade Utility, it restarts where it stopped.

These are the main steps in running the Upgrade Utility. If you are viewing this document online, you can click each step to link to a detailed explanation of that step.



Step 1: Load the upgrade files

Step 2: Prepare RDBMS mapped systems

Step 3: Run the SQL Compare utility

Step 4: Run the Upgrade Utility

Step 5: Verify that the schedulers are finished (SC4)

Step 6: Return the system to normal operation

Step 7: Review the upgrade reports

When running an application upgrade:

- If you encounter problems that cause the upgrade process to stop unexpectedly and you cannot determine the cause, contact HP OpenView Customer Support. Be ready to answer these question for customer support if you call them.
  - What upgrade step was the upgrade was at when it stopped?
  - What messages showed up on the screen?

  - What messages showed up on the SC.LOG file?
  - What messages showed up on the upgrade logs?

- The system you are upgrading is not fully functional until the application upgrade is complete.
- Before beginning the upgrade process, turn off your screen saver and any power-saving options on your computer.
- If the upgrade process fails at any point, you can restart from that point.
- When you restart the upgrade process, do not run the upgrade purge routine again.
- After you complete Step 4: Run the Upgrade Utility on page 71, check the log files periodically during the remaining steps to monitor the progress of an upgrade.
- If new schedulers are displayed on the status window after the upgrade completes, do not stop the system! Let the schedulers finish the background data upgrade before stopping the system.
- During the upgrade process, the Windows Task Manager indicates that ServiceCenter is not responding. This is normal and does not indicate a problem with the upgrade.

# Step 1: Load the upgrade files

The upgrade files preupg.bin and transfer.bin are part of the ServiceCenter Upgrade Utility. These files contain the applications and data to run the application upgrade on your development system.

## Identify the correct files

Running the upgrade the first time on your development system is different from exporting the custom upgrade files to your test or production system. The Upgrade Utility manages both processes, depending on the option you choose and the files that you point to.

- If you are upgrading your development system, use `preupg.bin` and `transfer.bin` (included with the ServiceCenter Upgrade Utility.)
- If you are exporting the custom upgrade to your test or production system, point to the Custom Upgrade directory where the custom upgrade files reside.

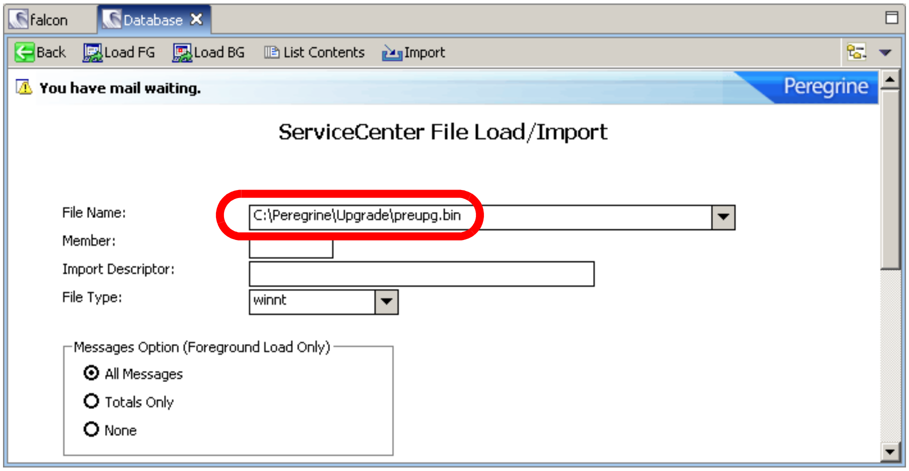
# Load the files

## To load preupg.bin

1. To prevent the schedulers from starting, ensure that you insert a comment character at the beginning of all statements except one `scenter -listener` statement.

   - Edit the sc.cfg file for both Windows and UNIX.

2. Start the ServiceCenter 6.2 server.
3. Start the ServiceCenter 6.2 client.
4. Log on as a user with SysAdmin permission.
5. Type db in the ServiceCenter client command line. Press **Enter**.
6. Click **Options** > **Import/load**.



You have mail waiting.

ServiceCenter File Load/Import

File Name: C:\Peregrine\Upgrade\preupg.bin
Member:
Import Descriptor:
File Type: winnt

Messages Option (Foreground Load Only)
● All Messages
○ Totals Only
○ None

7. Type the fully qualified path to `preupg.bin`, including the file name in the **File Name** field. For example, if you copied the files to a tmp directory, the path would be:

   - **Windows:** `c:\temp\upgrade\preupg.bin`
   - **Unix:** `/tmp/upgrade/preupg.bin`

8 Click **Load FG**.

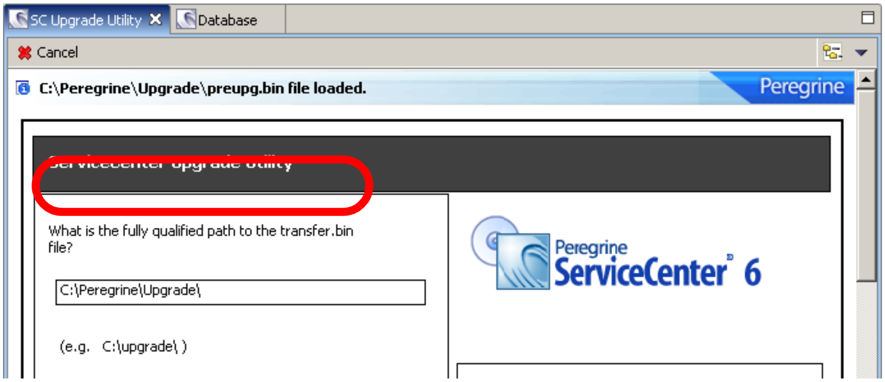
Depending on the speed of your system, loading this file should take from one to five minutes. When the operation is complete, ServiceCenter returns to the main Database Manager form and displays a confirmation message.

9 Type `load transfer` in the ServiceCenter client command line. Press **Enter**.

10 Type the fully qualified path to `transfer.bin` including the final back slash (\) or forward slash (/), depending on your operating system. For example, if you copied the files to a tmp directory, the path would be:

- **Windows:** `c:\temp\upgrade\`
- **Unix:** `/tmp/upgrade/`

Do not include the file name (transfer.bin) in this path.



C:\Peregrine\Upgrade\preupg.bin file loaded.

ServiceCenter upgrade utility

What is the fully qualified path to the transfer.bin file?

C:\Peregrine\Upgrade\

(e.g. C:\upgrade\ )

11 Click **Next**. When the process is complete, the system displays a confirmation message.

12 Stop the server.

13 Make a checkpoint backup of the data files to enable you to restore from this point, if necessary. See Step 3: Back up your production system on page 30.

14 Restart the server.

# Step 2: Prepare RDBMS mapped systems

If you are not mapped to an RDBMS, go to Step 3: Run the SQL Compare utility on page 66.

## Verify SQL system table mappings

JavaScript support was added to ServiceCenter 6, and fields for holding JavaScript were added to a number of ServiceCenter applications tables (such as Process, formatctrl, link, etc.). When using an RDBMS, fields containing JavaScript must be mapped to BLOB data types or ServiceCenter will not store, retrieve or process the JavaScript properly. Most of these tables are defined as sqlsystemtables, meaning they are stored in their entirety as BLOBs, and the JavaScript processing is not an issue. However, you should verify that the following tables are defined as sqlsystemtables following the application of the unloads in the previous step.

If any of these tables are in your RDBMS and not defined or mapped as SQL system tables in ServiceCenter, you will need to manually verify the mappings of each of these fields to ensure that they are handled properly.

| TABLE | FIELD | REQUIREMENT |
|---|---|---|
| Process | javascript.pre | long blob (image for SQL servers) |
| Process | javascript.post | long blob (image for SQL servers) |
| cascadeupd | javascript | long blob (image for SQL servers) |
| displaycache | ds.javascript.pre | long blob (image for SQL servers) |
| displaycache | ds.javascript.post | long blob (image for SQL servers) |
| displayoption | javascript.pre | long blob (image for SQL servers) |
| displayoption | javascript.post | long blob (image for SQL servers) |
| displayscreen | javascript | long blob (image for SQL servers) |
| formatctrl | javascript | long blob (image for SQL servers) |
| ioaction | javascript | long blob (image for SQL servers) |
| link | javascript.pre | long blob (image for SQL servers) |
| link | javascript.post | long blob (image for SQL servers) |
| link.line | javascript.pre | long blob (image for SQL servers) |
| link.line | javascript.post | long blob (image for SQL servers) |
| schedule | javascript | long blob (image for SQL servers) |
| scripts | javascript.pre | long blob (image for SQL servers) |

| TABLE | FIELD | REQUIREMENT |
|---|---|---|
| scripts | javascript.post | long blob (image for SQL servers) |
| triggers | javascript | long blob (image for SQL servers) |
| upgraderesults | object.name | need length > 150 |
| wizard | javascript.actions | long blob (image for SQL servers) |
| wizard | javascript.init | long blob (image for SQL servers) |
| wizard | javascript.cancel | long blob (image for SQL servers) |
| wizard | query.for.records.query | need length > 150 |
| wizard | select.no.records | need length > 150 |

The steps to accomplish this are:

1 Update the existing RDBMS tables to ensure the above corresponding columns exist and are the correct data type and size.

2 Update the SQL mapping information in the dbdict records to match the RDBMS changes you made in Step 1.

3 If the table does not exist in the RDBMS, and there is no `sqlsystemtable` record, to enforce the special mappings described in the above table, create a `sqlhints` record for each field to ensure it will convert to the RDBMS properly.

# Add SQL base names

The ServiceCenter 6.2 upgrade removes most subtables. The data policy records of some ServiceCenter 5.1 Inventory Management subtables do not have SQL Base Name references. If you are upgrading SC5.1 data and have mapped the following tables to DB2 or Oracle, you must add a SQL Base Name value to the data policy record (datadict table).

- For sDB2, the length of the name must be <=13
- For Oracle, the length of the name must be <=25

| Filename | Suggested<br>SQL Base Name | RDBMS |
|---|---|---|
| computer | computer | DB2 |
| computerconnectionsattr | computercon | DB2 |
| computerconnectionsuniq | computercon1 | DB2 |

| Filename | Suggested<br>SQL Base Name | RDBMS |
|---|---|---|
| computerdriversattr | computerdri | DB2 |
| computerdriversuniq | computerdri1 | DB2 |
| computerkeyboardsattr | computerkey | DB2 |
| computerkeyboardsuniq | computerkey1 | DB2 |
| computerlogicaldrivesattr | computerlog | DB2 |
| computerlogicaldrivesuniq | computerlog1 | DB2 |
| computerphysicaldrivesattr | computerphy | DB2 |
| computerphysicaldrivesuniq | computerphy1 | DB2 |
| computerpointingdevicesattr | computerpoi | DB2 or Oracle |
| computerpointingdevicesuniq | computerpoi1 | DB2 or Oracle |
| computerportsattr | computerpor | DB2 |
| computerportsuniq | computerpor1 | DB2 |
| computerprintersattr | comppriattr | DB2 |
| computerprintersuniq | comppriuniq | DB2 |
| computerprocessorsattr | computerpro | DB2 |
| computerprocessorsuniq | computerpro1 | DB2 |
| computerusbattr | computerusb | DB2 |
| computerusbuniq | computerusb1 | DB2 |
| computervideoattr | computervid | DB2 |
| computervideouniq | computervid1 | DB2 |
| telecom | telecom | DB2 |
| telecomconnectionsattr | telecomconn | DB2 |
| telecomconnectionsuniq | telecomconn1 | DB2 |
| telecomdriversattr | telecomdriv | DB2 |
| telecomdriversuniq | telecomdriv1 | DB2 |
| telecomportsattr | telecomport | DB2 |
| telecomportsuniq | telecomport1 | DB2 |

## Step 3: Run the SQL Compare utility

The SQL Compare utility compares your existing table and field information with the tables and fields of the ServiceCenter version you are upgrading to. It will report on any new fields that are to be merged into the existing tables. You can:

- Use the list of the fields produced by SQL Compare to determine whether any fields in your current system differ from those in the new version.
- Use the report to determine which new fields you must add to RDBMS mapped files, if you chose to make the changes manually during the application upgrade.

**Note:** The application upgrade does not make key changes. You must make any key changes manually. See Application changes by release on page 115 for the necessary key changes.

## Non-RDBMS mapped systems

Use SQL Compare to see a list of the fields and tables that the upgrade will change.

## RDBMS mapped systems

When you run the upgrade utility (Phase IV), you will be asked if ServiceCenter should make the necessary SQL changes for you, or if you would like to make the changes manually. If you choose to make the changes manually, you can use the SQL Compare utility to identify the necessary changes. Then, after you apply the changes specified by SQL Compare, you can manually upgrade your ServiceCenter system while it is mapped to an RDBMS.

## Running SQL Compare

These SQL Compare files install when you install the ServiceCenter Upgrade Utility:

- sqlupgrade.unl
- upgdbdct.dta

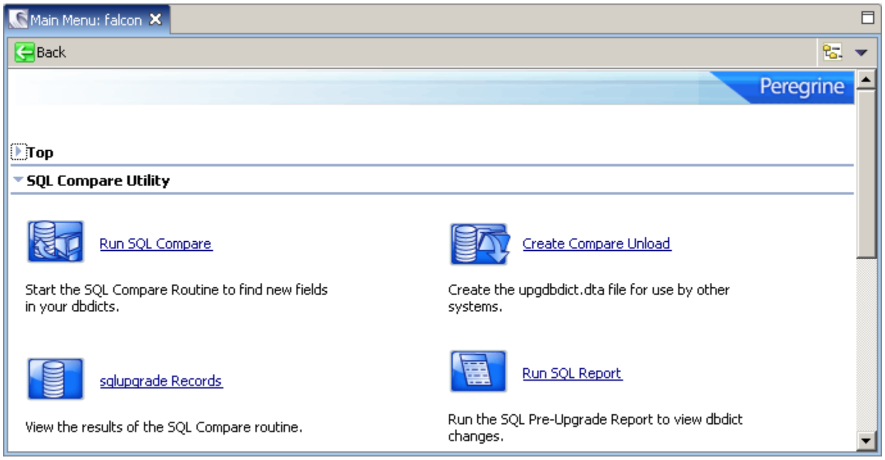
SQL Compare returns a message for every P4 database dictionary mapped to SQL that contains new fields. You can update these database dictionaries to contain the fields specified by the SQL Compare applications before you begin the application upgrade.

**Note:** Run SQL Compare on the development system.

## To compare applications

1. Load sqlupgrade.unl into your development system.

2. Type `*mSQL COMPARE` in the ServiceCenter client command line. Press **Enter**. The **SQL Compare Utility** menu opens.



Start the SQL Compare Routine to find new fields in your dbdicts.

Create the upgdbdict.dta file for use by other systems.

View the results of the SQL Compare routine.

Run the SQL Pre-Upgrade Report to view dbdict changes.

3. Click **Run SQL Compare**. A dialog box opens.

4. Type the fully qualified path to `upgdbdct.dta` including the final back slash (\) or forward slash (/), depending on your operating system. For example, if you copied the files to a tmp directory, the path would be:

   - **Windows:** `c:\temp\upgrade\`
   - **Unix:** `/tmp/upgrade/`

   Do not include the file name (upgdbdct.dta) in this path.

SQL Compare returns this message:

```
Process Complete. Please check for any additional messages.
```

The results of the SQL Compare process are in the sqlupgrade file. This file resets each time you run SQL Compare.

### To view the sqlupgrade file

1 Type `*mSQL COMPARE` in the ServiceCenter client command line. Press **Enter**. The **SQL Compare Utility** menu opens.

2 Click **sqlupgrade Records**.

3 Click **Search**. ServiceCenter displays the results.

Each file that requires changes appears as a separate record in the sqlupgrade file. This record also lists the new fields that you must add to the file, if you are updating your RDBMS mapped system manually.

The sqlupgrade record provides the following information for each field you must add, if you are updating your RDBMS mapped system manually.

| Field | Description |
|---|---|
| Field Name | The exact field name to add to the associated P4 database dictionary. |
| Type | The data type of the field. |
| Level | The level where this field resides. |
| Structure | The structure and array name that you should add to this field to. |
| Alias of | If this is an alias field, it contains the name of the primary field that it is an alias of. Otherwise this field is blank. |

## Add new fields

For the new fields to perform correctly, they must exist in the P4 database dictionary and the SQL database. If you are updating your RDBMS mapped system manually, you must add them to the database and update the existing ServiceCenter SQL mapping. When you update a table in the sqlsystemtables file, add fields only through the database dictionary. Modifying the SQL mapping damages the file structure of the table.

### Determine the correct structure

In most cases, you should add the new field to the descriptor structure. However, sometimes the **Structure** field contains something other than the word *descriptor*. When this occurs, add the new field to the appropriate location.

| In this instance | Add the field here |
|---|---|
| The field resides in another structure | If the field is not an array field, you must add the field to the structure listed in the Structure field. For example, if the **Structure** field reads *middle*, add the field to the middle structure of the dbdict. |
| The field is an array | If the field is an array, the field name appears twice in the new field list. The first entry has a data type of *array*; the second is the data type of the array, such as *character* or *logical*. Use the first entry to determine the structure where you should add the array. The **Structure** field in the second entry reflects both the structure for the array (unless it uses the descriptor structure) and the name of the array itself. |
| The field is part of an array of structures | If the **Structure** field lists multiple fields exclusive of an array name, you must add the field to a structured array. To determine the placement in the structured array, follow the list of field names in the Structure/Array from left to right. The first name is the array name and the second is the structure name.<br>**Note:** When adding fields to a structured array, add them in the same order as they appear in the sqlupgrade record. |

## Update field lengths

If you mapped the notification, signatures, upgradepseudolog, or upgradeobjects tables to SQL, check the field lengths of the fields in this table before you begin the upgrade. The standard default lengths for these fields may not be long enough. Change the field lengths for these columns in your SQL database to 255 characters.

| Table | Field |
|---|---|
| signatures | object.name |
| upgradepseudolog | message |
| upgradeobjects | object.name |
| notification[a] | str.condition |

a If the condition field in the notification file is not mapped as a BLOB type, then you must convert the notification file back to P4 and remap it with the condition field as a BLOB.

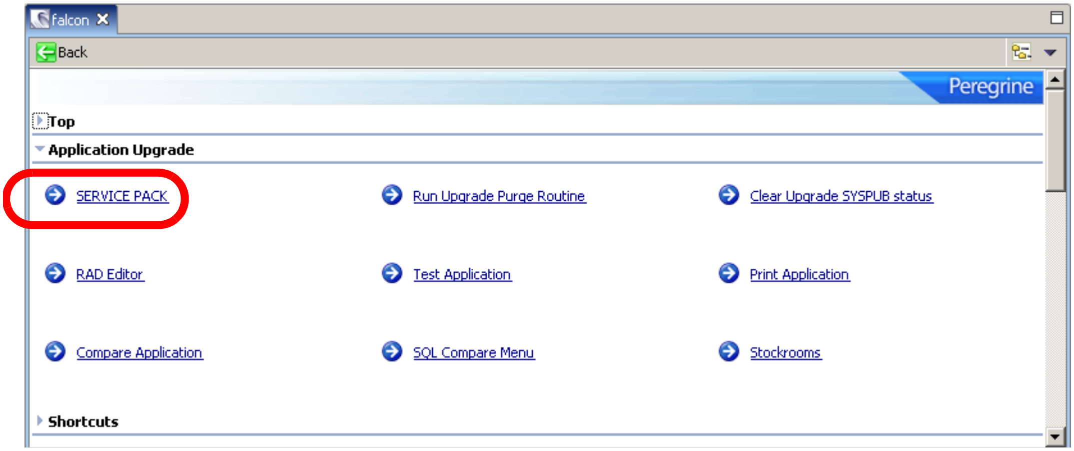
# Step 4: Run the Upgrade Utility

The Upgrade Utility installs application updates automatically. At any time during this procedure, you can click **Next** to continue, or click **Back** to return to previous screens.

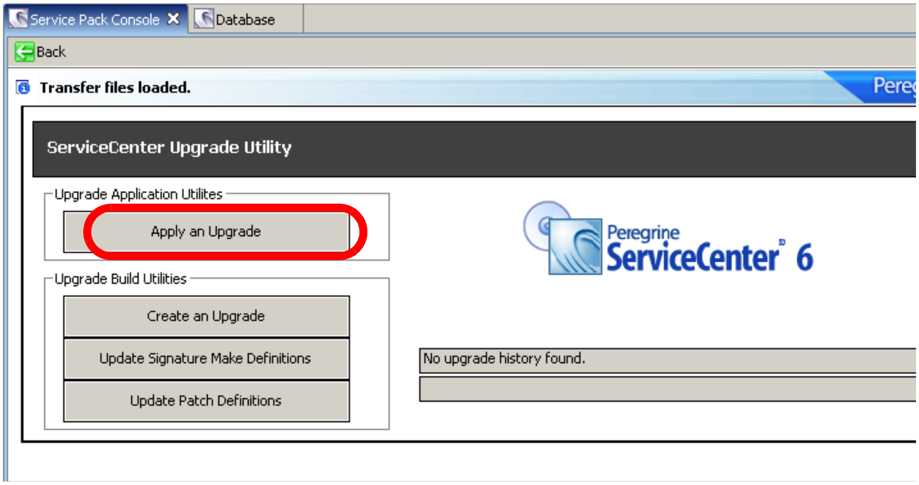
### To run the ServiceCenter Upgrade Utility

1 Type `sc62upgrade` in the ServiceCenter client command line. Press **Enter**.

2 Click **SERVICE PACK** in the **Application Upgrade** menu.



3 Click **Apply an Upgrade**.



Transfer files loaded.

No upgrade history found.

The Upgrade Utility guides you through questions specific to the upgrade process. You must answer them before the upgrade can begin.

4 Click **Next** to start the upgrade.

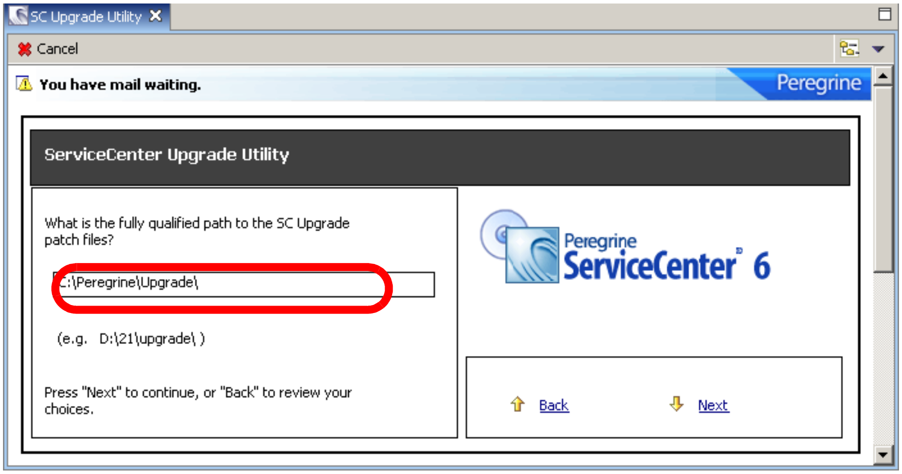
5 If this screen displays your correct version number, click **Next**.

If this screen does not display the correct version number, do not continue with the upgrade, instead contact HP OpenView Customer Support. This could happen because your version is older than SC4 or because your version number is corrupt and the Upgrade Utility does not recognize it.

6 Specify whether this is a development system. Click **Next**.

- Select **Yes** to apply the out-of-box upgrade to your development system. After this upgrade to your development system, you will perform conflict resolution and create a custom upgrade.
- Select **No** to apply a the custom upgrade to your test or production system. You must have already created a custom upgrade.

7 Type the fully qualified path to the Upgrade or Custom Upgrade directory. This path must contain a final forward slash (/) or backslash (\), depending on your operating system. Click **Next**.



You have mail waiting.

ServiceCenter Upgrade Utility

What is the fully qualified path to the SC Upgrade patch files?

C:\Peregrine\Upgrade\

(e.g. D:\21\upgrade\ )

Press "Next" to continue, or "Back" to review your choices.

8 Select how the Upgrade Utility should handle objects. Click **Next**.

- If you are applying the out-of-box upgrade to your development system, select **Install HP's Version of the Object Alongside Your Own.** This renames the new file whenever there is a difference between the existing file and the new file. You do not want to overwrite the existing object until you determine what the differences are. The file naming convention is `NEW<release><filename>`. For example, `NEWSC6.2pm.main`.
- If you are applying a custom upgrade to your test or production system, select **Replace your version of the object with HP's Version of the object.** This renames the old file if there is a difference between the existing file and the new file. The file naming convention is `OLD<release><filename>`. For example, `OLDSC4pm.main`. At this point, the custom upgrade should contain the correct file because you have already resolved the differences.

For a more complete explanation of these options, see Step 1: Resolve application data differences on page 87.

9 The Upgrade Utility always uses External logging. Click **Next**.

- Click **Yes** to also store the log files internally within ServiceCenter.
- Click **No** to use only external logging. External log files reside in the same directory as the upgrade files.

For more information, see Upgrade Utility logs and error messages on page 76.

10 Select the type of incident tickets to process for the SLA upgrade. Click **Next**.

- Select **All Incidents** for complete reporting.
- Select **Open Incidents** if you have a lot of tickets and want to reduce down time.
- Select **None** if you are not using SLA or if you plan to run the SLA upgrade separately.

**Note:** If you are running an RDBMS-mapped system, continue with Step 11, if not, proceed to Step 17 on page 75 to continue with the upgrade.

11 If any databases are shadowed to an RDBMS, the Upgrade Utility displays the message shown in the following dialog box. The Upgrade Utility does not make any modifications to the RDBMS for shadowed databases. You must make those changes separately. Click **Next**.

12 Select the **Target SQL Database Type**. If you are unfamiliar with the database connected to ServiceCenter, contact your database administrator.

13 Select the **Disposition of Array** fields. Click **Next**.

14 Type in the **Other RDBMS information**. Click **Next**.

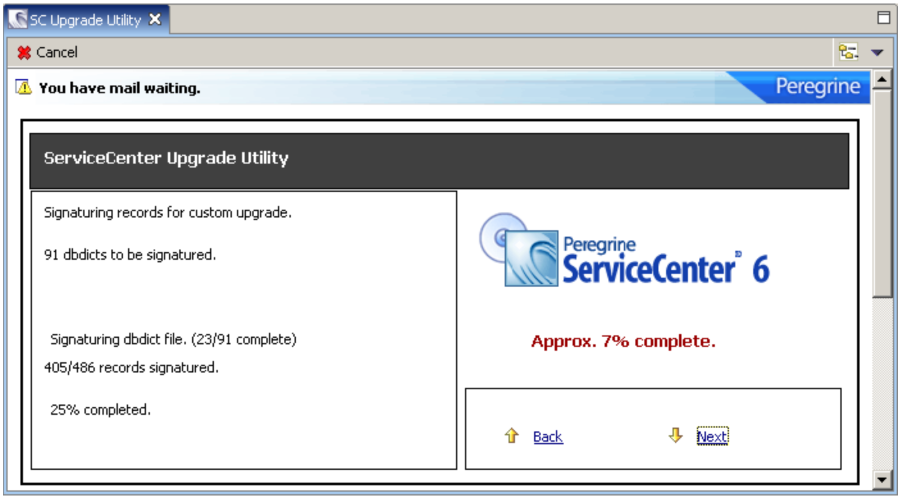
15 Choose whether you want the Upgrade Utility to update the RDBMS tables automatically or you want to view and modify the alter statements manually. Click **Next**. For more information, see Modifying the alter statements on page 76.

If you choose to view the alter statements, the upgrade stops at the end of the database dictionary phase, even if you specified the run-to-completion option. You will have the opportunity to modify the alter statements manually.

16 Click **Yes** to re-map the Change Management and system files automatically. Click **No** to re-map manually. Click **Next**.

17 Click **Next** to confirm your selections.

18 A window opens. Click **Yes** to start the upgrade. While the upgrade runs, the Upgrade Utility displays the progress of the upgrade process by indicating which process is running, the progress of record processing, and the completion percentage for the upgrade.



You have mail waiting.

ServiceCenter Upgrade Utility

Signaturing records for custom upgrade.

91 dbdicts to be signatured.

Signaturing dbdict file. (23/91 complete)

405/486 records signatured.

25% completed.

Approx. 7% complete.

If your system is mapped to an RDBMS, Upgrade Utility may display a dialog with the following message after the database dictionary upgrade:

```
Not all dbdicts have been upgraded. Do you want to continue?
```

If this message appears, log off and stop the server. This is necessary to clear the SQL cache. To continue, restart the server, log on, and start the upgrade application. The Upgrade Utility restarts from where it stopped. This message indicates that some field types may not have been as the application expected them and could have an impact on the upgrade or regular system operation. Review the upgrade logs for details.

19 When the upgrade process finishes, the Upgrade Utility displays a completion message. Click **OK** in the message dialog box.

20 Review the logs and error messages. For more information, see Upgrade Utility logs and error messages on page 76.

## Upgrade Utility logs and error messages

The Upgrade Utility creates a set of log files during the upgrade process. These files reside in the same directory as the upgrade files. You can also store a log internally within ServiceCenter.

| Log file | Contents |
|---|---|
| detail.log | Specific information about the upgrade, such as which files are being signatured at any time. |
| exception.log | Information about any exceptions reported by the upgrade.<br>The exception.log file may have important messages about data type mismatches that you should resolve, or database dictionaries that it cannot upgrade. If these error messages appear in the except.log file, do not continue the upgrade until you resolve the data type mismatches.<br>If there are exceptions in probsummary, problem, cm3r, cm3rpage, cm3t, or cm3tpage files, do not continue the upgrade until you repair them. |
| upgrade.log | Information about where the upgrade is at any point. This file contains only the main steps of the upgrade. |

You can access the post-upgrade reports through the Upgrade menu.

## Modifying the alter statements

If you are upgrading a system mapped to an RDBMS and you select the option to review the alter statements that the Upgrade Utility creates, the upgrade stops for each database dictionary that requires an SQL alter statement to be applied.

When the Upgrade Utility displays an alter statement, you can:

- Allow the Upgrade Utility to apply the alter statement to the RDBMS exactly as it appears.
- Modify the alter statement manually and allow the Upgrade Utility to apply that statement.
- Handle the altering of the RDBMS tables manually, and skip the alter statement.

After you view each alter statement and make any changes:

- Click **Proceed** to apply the alter statement.
- Click **Skip** to proceed to the next alter statement without taking any action.

  If the alter statement is unsuccessful, the alter window opens. You can edit the statement and try again, or quit the upgrade process.

**Note:** If you quit the upgrade process before successfully applying the alter statement, the database dictionary in question becomes unstable until you correct the SQL mapping.

# Files converted to system tables

If one of the following tables is mapped to an RDBMS on your system and is not a system table, the Upgrade Utility converts it into a system table during the upgrade process.

Application changes by release on page 115 lists some key (index) changes required. If any relate to a dbdict mapped as a system table, then the table needs to be converted back, the key added and the table remapped. That way a field required by the key is not part of the BLOB.

The ServiceCenter system tables include:

- application
- applicationrevision
- caldaily
- category
- cm3profile
- cm3ralerts
- cm3rcatphase
- cm3talerts
- cm3tcatphase
- datadict
- datamap
- dbdict
- displaycache
- displayevent
- displayeventrev

- displaymaster
- displayoption
- displayoptionrev
- displayscreen
- displayscreenrev
- dtqueue
- dtshad
- enclapplication
- enclapplrev
- enclapplrevision
- environment
- eventfilter
- eventin
- eventmap
- eventout
- eventregister
- format
- formatcontrolrevision
- formatctrl
- formatrevision
- globallists
- help
- icmenv
- info
- irqueue
- link
- linkrevision
- macro
- macrodef
- macroheader
- menu
- menucmdlist
- menurevision
- msglog
- object
- Objectrevision
- ocmalertlog
- ocmalertpool
- ocmapprlog
- ocmapprpool
- ocmcatselect
- ocmevents
- ocmlcat
- ocmocat
- ocmoptions
- ocmphaselog
- ocmprofile
- ocmqcat
- patches
- pmenv
- pmnotes
- Process
- Processrevision
- sc
- schedule
- scparms
- screlconfig
- scripts
- signatures
- slacontrol
- smenv
- sqldbdict
- sqlqueue
- Staterevision
- States
- status
- systemperform
- termtype
- tzfile
- upgdbdict
- upginfo
- upgrade
- upgradeapplication
- upgradeddbdict
- upgradeobjects
- upgradepseudolog
- upgradepsuedolog
- upgradestatus
- validity

**Note:** Not all tables on this list may exist in your system.

# Step 5: Verify that the schedulers are finished (SC4)

Skip this step unless you are upgrading from SC4 systems. Changes to the applications make it necessary to update the actual data within the system. For example:

- The Upgrade Utility upgrades active Incidents and Changes during the foreground upgrade process.
- The Upgrade Utility upgrades inactive (or closed) Incidents and Changes, and some cost information related to incidents that use background schedulers.

The Upgrade Utility starts the background processes automatically. When finished, they stop automatically.

| Background scheduler | Data upgraded |
|---|---|
| upgrade | Incidents |
| upgradecm3 | Changes |
| upgradecost | Costs related to incidents |

Warning: Do not shut down the system or build the custom upgrade until these three schedulers finish running.

To determine if the background schedulers are finished:

- Click **System Status** on the system administrator's home menu. When the processes are complete, ServiceCenter no longer displays the background schedulers in the System Status window.

# Step 6: Return the system to normal operation

After the upgrade, the system may exhibit abnormal behavior until you return it to its normal operating environment.

### To return to a normal operating environment

1 Log out.

2 Stop the server.

3 Remove the comment from the `system.start` entry from the `sc.cfg` file

4 Restart the ServiceCenter server.

5 Log on.

6 Wait for the background processes to finish.

7 Regen your IR keys.

# Step 7: Review the upgrade reports

The Upgrade Utility generates reports that provide feedback about its activities. After the upgrade finishes, you can view them to identify the files on your system that are different from the new files.

This table defines the Upgrade reports.

| Report Title | Report Name | Description |
|---|---|---|
| Service Pack Full Upgrade Report | apm.upgrade. results.full | A list of all objects that the Upgrade Utility processed, even if they upgraded smoothly.<br>This is a large report that typically contains more than 10,000 lines.<br>The ServiceCenter parameter maxreportpages default value is 1000. If you do not explicitly set it to a higher number in the `sc.ini` file, your report terminates after printing 1000 pages. |
| Service Pack Job Log (Large) | apm.upgrade. job.log | A dump of the upgrade job log.<br>This log is generally not necessary. Use it for debugging or informational purposes. |
| Service Pack Exception Report | apm.upgrade. results.exceptions | A list of the objects that the Upgrade Utility could not upgrade automatically.<br>It classifies those components as merged, renamed, or added. If it returns the message: "`No records selected by report.`"then all components upgraded successfully. There are no discrepancies and no records need special attention. If the Upgrade Utility displays any other messages, they describe upgrade exceptions you must resolve.<br>If there are exceptions, proceed to the next chapter, Phase V: Resolving conflicts on page 85. |

# Run the post-upgrade reports

The steps to view the reports are almost identical. The only difference is in the report you select.

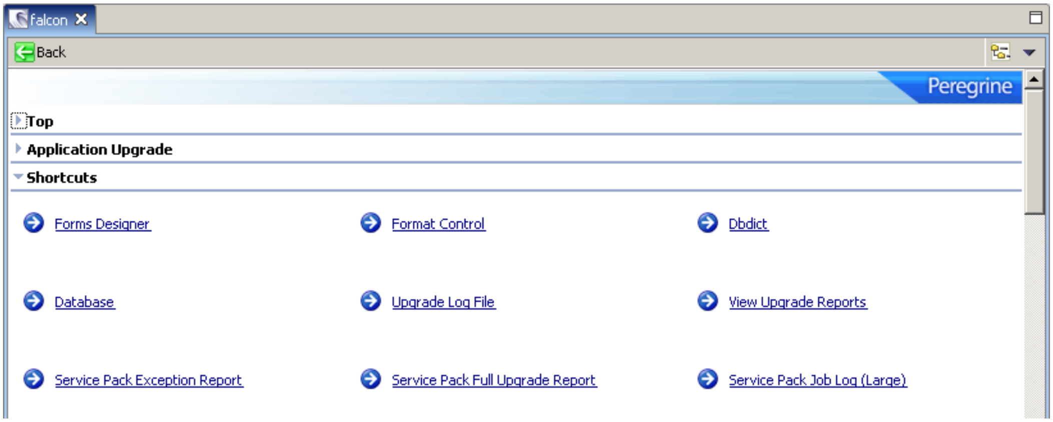
### To run a post-upgrade report

1 To print the post upgrade reports, you must enable Server printing.

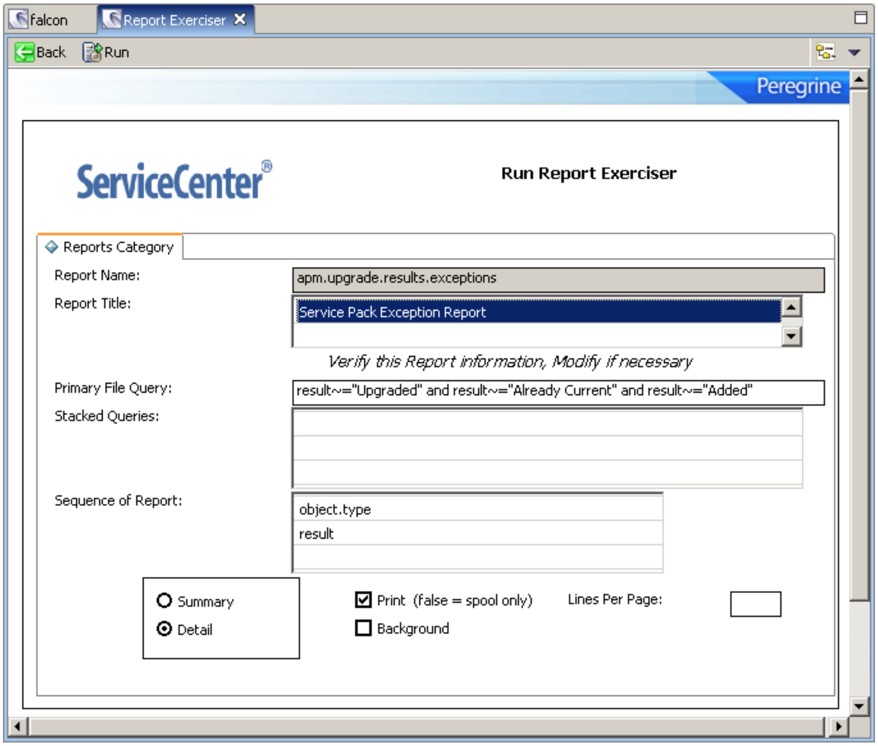
- For versions prior to SC 6.0, select the **File** > **Printing** > **Server Printing** option.
- For version 6.0 and later, go to **Window** > **Preferences**, and confirm that the **ServiceCenter** > **Client Printing** option is not selected.

2 Type `sc62upgrade` on the ServiceCenter client command line. Press **Enter**.

3 Open the **Shortcuts** menu and click the report you want to run.



The utility displays the query form for that report.



Report Name: apm.upgrade.results.exceptions

Report Title: Service Pack Exception Report

*Verify this Report information, Modify if necessary*

Primary File Query: result~="Upgraded" and result~="Already Current" and result~="Added"

Stacked Queries:

Sequence of Report: object.type<br>result

○ Summary
● Detail

☑ Print (false = spool only)
□ Background

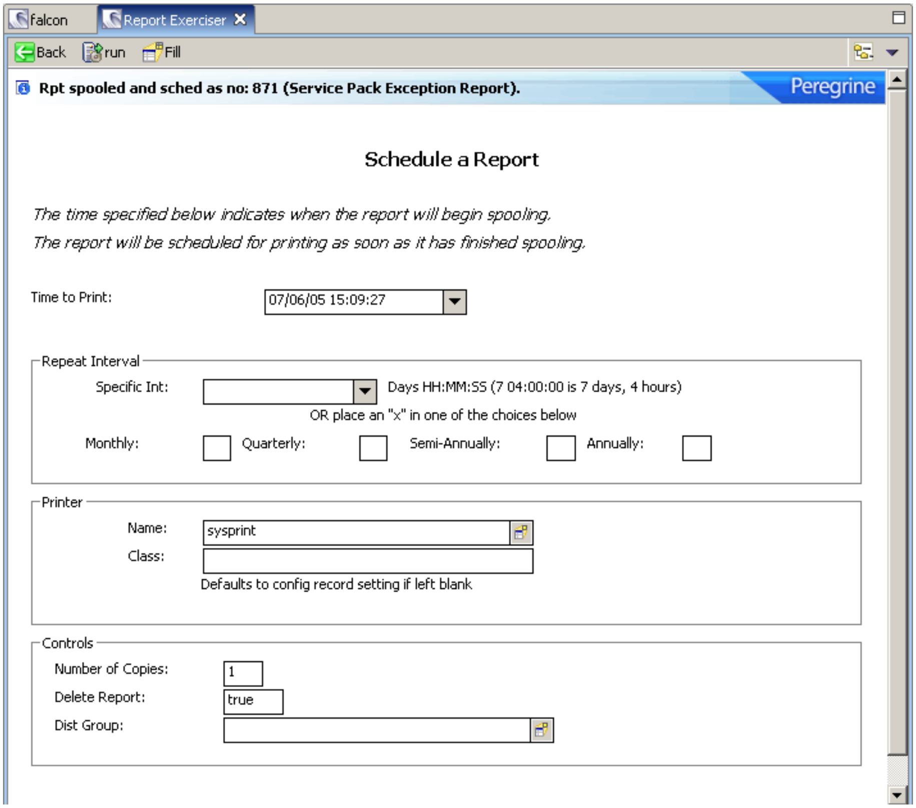
Lines Per Page:

4 Choose the options for generating the report.

| Option | Definition |
|---|---|
| Print | If you select the Print option, you can select the printer and schedule when the report prints.<br>If you clear the Print option, ServiceCenter displays the report.<br>The form only displays this option if you enable server printing. |
| Background | If you select the Background option, you can select the printer and schedule when the report prints. The report spools in the background, enabling you to return to the client immediately. The scheduler starts the spool process.<br>If you clear the Background option, the report spools in the foreground. The ServiceCenter client is disabled until the report finishes spooling. |

5 Click **Run**.

If you selected **Background**, The utility displays the report schedule form.



Rpt spooled and sched as no: 871 (Service Pack Exception Report).

Schedule a Report

*The time specified below indicates when the report will begin spooling.*
*The report will be scheduled for printing as soon as it has finished spooling.*

Time to Print: 07/06/05 15:09:27

Repeat Interval

Specific Int: Days HH:MM:SS (7 04:00:00 is 7 days, 4 hours)

OR place an "x" in one of the choices below

Monthly: □ Quarterly: □ Semi-Annually: □ Annually: □

Printer

Name: sysprint

Class:

Defaults to config record setting if left blank

Controls

Number of Copies: 1

Delete Report: true

Dist Group:

6 Choose the options for scheduling the report.

7 Click **Run**. The utility displays the Report Maintenance form.

8 Click **View** on the Report Maintenance form. The utility displays the spoolheader form.

9 Click **View Report** to open the report.

# Retrieve reports

You can run a report once and view it later by selecting it from the print spool.

### To retrieve an existing report

1. Type `spool` in the ServiceCenter command line. Press **Enter**.
2. When ServiceCenter displays the Spool Header record, press **Enter**. (Do not click **Find**.) ServiceCenter displays a record list of recent reports in the print spool.
3. Select the report to view.

# 5 CHAPTER Phase V: Resolving conflicts

After you complete the initial application upgrade on your development system, you are ready to resolve the differences between your customized files and the new files. You may need to resolve differences in the display components, the database dictionaries, and the applications. Until you resolve these differences, your upgraded system will not work properly.

Topics in this section include:

## Resolving conflicts overview

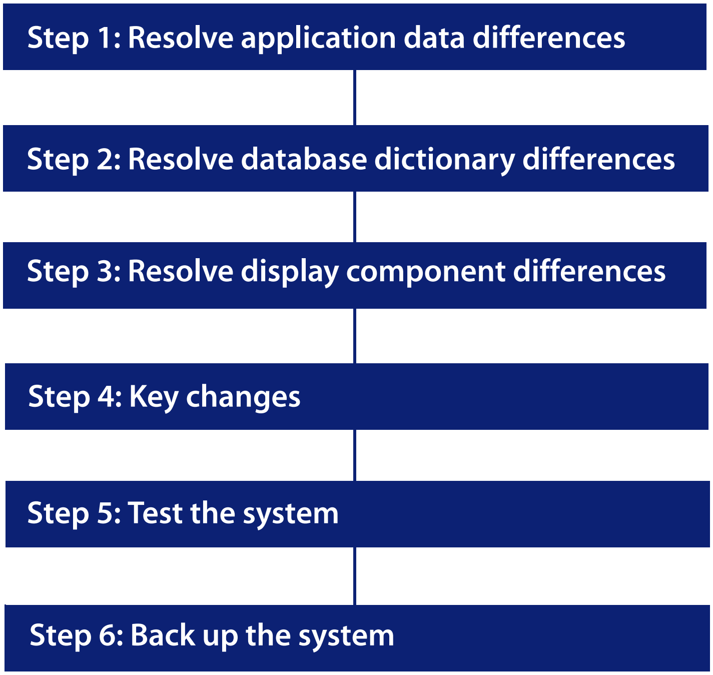
For each file identified in the exception reports, you have these choices:

- Use your custom file instead of the new file.
- Use the new file instead of your customized file.
- Make the same modifications in the new file and discard the old custom file.

When making your decision, consider that:

- It is not always desirable to overwrite existing files. For example, if you modified an application to add features for your site, you may not want to discard your modifications.
- New application files often have more functions and features that are more attractive than the existing files. Using new files makes your application more robust.

These are the main steps in resolving conflicts. If you are viewing this document online, you can click each step to link to a detailed explanation of that step.



Step 1: Resolve application data differences

Step 2: Resolve database dictionary differences

Step 3: Resolve display component differences

Step 4: Key changes

Step 5: Test the system

Step 6: Back up the system

# Resources

When you begin the resolution process, assemble your resources and save them as historical data for the next upgrade. You need:

- A list of customized files.
- Work notes and historical unload archives.
- The Service Pack Exception and Display reports.

You can use revision tracking to make unload archives for future use.

Be sure to make frequent checkpoint backups of the data files. See Step 3: Back up your production system on page 30.

# Object Types

The exception reports list ServiceCenter components as object types. For example, ServiceCenter 6 had 76 signatured objects. The Upgrade Utility identifies any object with a mis-matched signature as changed. You must decide if the change is significant, or if you can safely accept the difference without detailed investigation of the differences.

# Step 1: Resolve application data differences

The Upgrade Utility upgrades any objects in your file system that are unchanged from the out-of-box version for that release. When the Upgrade Utility encounters an object that is not identical to the original out-of-box version, the upgrade installs the new version as NEW<vvvnnn>, where vvv is the release name and nnn is the name of the object, or the upgrade renames your version as OLD<vvvnnn>.

The method depends upon the choice you made when starting the upgrade process.

- If you chose **Install HP's Version of the Object Alongside Your Own**:

  When the Upgrade Utility does not recognize an object as the original out-of-box version, it leaves that object alone. The upgrade names the new version of the object NEWxxxyyy where xxx is the ServiceCenter Version you are upgrading to and yyy is the name of the object. The new object has the status "Renamed" in the upgraderesults file.
- If you chose **Replace your version of the object with HP's Version**:

  When the Upgrade Utility does not recognize an object as the original out-of-box version, it renames that object as OLDxxxyyy where xxx is the ServiceCenter version you are upgrading from and yyy is the name of the object. The upgrade installs the new version of the object. The new object has the status "Forced" in the upgraderesults file.

You must examine the old object and the new one and decide whether to:

- Retain the old version of the object in question
- Accept the new object
- Create a combined object

Most objects, including forms, Format Control records, and validity records, are relatively easy to check. For example, adding a single user-defined field to a form causes a signature mis-match. Examine the old object and the new one, side-by-side, if possible.

## Application differences

Customized applications are more complicated. Application differences occur when you customize the RAD code or when you apply a patch to your existing version. The apm.upgrade.results report identifies these application differences. After you produce the report, check each application listed for recommended action. The utility assigns a status to each application. The following table lists the possible status and the recommended action.

| Upgrade status | Definition | Action |
|---|---|---|
| Added | The Upgrade Utility added the new object to your system because you did not have an existing object. | No action required. |
| Error | The Upgrade Utility could not upgrade the object. | Resolve the error. Contact Customer Support for assistance if necessary |

| Upgrade status | Definition | Action |
|---|---|---|
| Forced | The Upgrade Utility renamed your object OLDSC6.2<object.name> and added the new object as <object.name>.<br>The Upgrade Utility used this option because you chose to replace your version of the object with HP's version of the object. (See Step 8 on page 73.) | Select one:<br>**Keep the old version** — No action required.<br>**Keep the new version** — Rename the old version and give the new version the original name.<br>**Merge new and old versions** — Add the new features to the old version.<br>Always make your changes in the old file by manually adding the new features. If you rename the new file to the name of the old file, you will lose data because the data is linked to the logical file number, not the file name. |
| Renamed | The Upgrade Utility renamed the new object NEWSC6.2<object.name> and added it to your system.<br>The Upgrade Utility used this option because you chose to install HP's version of the object alongside your own. (See Step 8 on page 73.) | Select one:<br>**Keep the old version** — No action required.<br>**Keep the new version** — Back up the old version and give the new version the original name.<br>**Merge new and old versions** — Add the new features to the old version.<br>Always make your changes in the old file by manually adding the new features. If you rename the new file to the name of the old file, you will lose data because the data is linked to the logical file number, not the file name. |

## Guidelines for resolving differences

Use these guidelines to make conflict resolution decisions.

- If your company does not have a RAD license, always select the new version.
- If your company has a RAD license, identify the differences between the old version and the new version by using the RAD Comparison Utility. Choose one of the following options:
  - Keep the existing version.
  - Keep the new version.
  - Merge the new version with the existing version.

**Note:** If your company has RAD license and the application has been modified for you specifically, then you should merge changes. A SCR unload can cause applications to show up on the conflict resolution exception report that no longer apply with an upgraded version of the application.

For more information, see Using the RAD Comparison Utility on page 109.

- If the name of the application starts with apm.upgrade, keep the existing application and delete the NEWvvvnnn version of the application. RAD applications named apm.upgrade.* are HP-reserved applications. Do not change them.
- Avoid changing macro.build.field.array. If macro.build.field.array is not identical to the new version, keep the new version.
- Never rename the applications se.view.engine, display or se.get.object when resolving conflicts.
- Changes to the **Regen Every** field prevents a global list from upgrading. If comparison to the new version of the global list shows no other changes, keep the new version.

# Application dependencies

HP OpenView ServiceCenter applications often use the RAD language to call other applications. For example, the se.view.engine application calls a second application, se.get.object, to update change records. Consequently, the functionality of se.view.engine depends on the functionality of se.get.object.

It is possible to upgrade se.view.engine but not se.get.object. Upgrading only one application can cause parameter differences in a subordinate application and logical dependency differences.

## Logical dependences

Logical dependencies usually occur when there is a new feature in a child application that the parent application requires. If you upgrade the parent application but the not the child application, the new feature is not available to the parent application. Resolving logical dependencies usually requires reverting the parent application to the previous version, or upgrading the child application to the latest version.

## Parameter differences in subordinate applications

The most common dependency problem arises when the number of parameters in a subordinate application changes. For example, if an application had four parameters but now has five, the old parent application passes four parameters, but the new child application now expects five. If you upgrade the child application but not the parent, the missing parameter can cause errors.

# Altered validity table entries

ServiceCenter processes validity table entries in entry sequence number order, such as 1, 2, ...n. Most users who implement multiple validity entries add sequence numbers to their entries. Typically, several validity entries exist for a given format with sequence numbers running from 1 to n. Many of the ServiceCenter default validity entries have sequence numbers of NULL.

When the Upgrade Utility updates your validity entries, it adds ServiceCenter's default validity lookup entries if it does not find an entry with the same key. For example, if your system does not have a validity entry with a sequence number of NULL, and the default validity entry has the NULL sequence number, the Upgrade Utility adds the new entry to your file system.

Although the Upgrade Utility does not remove any of your code, it adds a validity entry that it processes first. As a result, the first validity lookup you see is a default validity entry, rather than one of your modified entries.

### To remove invalid validity table entries added by the Upgrade Utility

1. Type `validity` on the ServiceCenter client command line. Press **Enter**.
2. Type the name of the file or format in question in the **Files/Formats** field.
3. Click **Search**.
4. Select a field name from the record list.
5. Look for an entry with a sequence number of NULL. Verify that the Upgrade Utility added this entry and that it is not an important part of your validity processing.
6. To remove the invalid validity record, click **Delete**.

   Validity processing now starts with your lowest sequence numbered validity entry.

# Step 2: Resolve database dictionary differences

When the Upgrade Utility upgrades your database dictionaries, it often finds that your database dictionary version does not match any previous ServiceCenter versions. For example, this occurs if you add or delete fields.

When the Upgrade Utility discovers these discrepancies, it merges your database dictionary with the new database dictionary. The resulting database dictionary contains all the fields in your original database dictionary, as well as all new fields.

The Upgrade Utility gives these combined database dictionaries a status of Merged. The Merged status does not indicate an error, but shows that the Upgrade Utility found differences in that database dictionary. Database dictionaries mapped to an RDBMS always have a Merged status because the mapping data within each table is unique and cannot be compared to basic ServiceCenter data.

## Data types

When the Upgrade Utility merges database dictionaries, the data type and the field name must match for success. If the new version of a field is a different type than the type in your system, it chooses your field type. HP OpenView recommends that you modify the field to the expected field type.

Duplicate field names are problematic. For example, if you create a scalar field with a user-defined name and the Upgrade Utility attempts to add a structure field with the same name, this discrepancy appears in the except.log file. The Upgrade Utility cannot add the structure and its subordinate fields until you resolve the name duplication. The except.log file lists the names of any fields that are not added because of a duplicate field name.

**Best practice:** Create a naming convention with a recognizable unique prefix for all user-defined fields. The Upgrade Utility action will always merge these fields successfully.

## Database dictionary differences

After you produce the apm.upgrade.results report, check it for recommended action. The following table lists the possible results and the recommended action.

| Upgrade result | Definition | Action |
|---|---|---|
| Merged | The Upgrade Utility merged the new database dictionary with your current version. | No action required. |
| Added | The Upgrade Utility added the new database dictionary to your system because you did not have an existing version. | No action required. |
| Error | The Upgrade Utility could not upgrade the database dictionary. | Resolve the error. Contact Customer Support for assistance if necessary. |

## Data policy differences

If you modified a data policy record, the Upgrade Utility usually cannot upgrade the associated data policy record. Keep the current version of the database dictionary because it reflects the current state. Investigate the new version of the database dictionary for any changes to data policy that you may want to move to the current database dictionary record.

# Step 3: Resolve display component differences

The ServiceCenter Upgrade Utility upgrades the display components. It reports discrepancies if any of your components differ from the new versions. You must be familiar with the Display application and how it functions to resolve the differences.

## Display application

The Display application is a ServiceCenter RAD application that provides access to RAD features without requiring RAD programming skills or RAD licensing.

If the Display application in your ServiceCenter system is not identical to the new version, rename the current version of the application to `display.old` and rename the `NEWSCxdisplay` version to `display`.

If you access this application through the RAD editor, use the equal sign (=) operator to select only the application in question, such as: `=display or =NEWSCxdisplay.`

# Display screen records

Display screens are individual records identified by a unique screen ID. The displayscreen records define the attributes of a screen and provide access to the individual records for options and events. A display screen is not the same as a form.

**Important:** There are triggers attached to the displayscreen file. Changes to the records in this file affect their associated display options and events.

When making changes to these records:

- Incorporate any changes to the new version of the displayscreen file into the old version of the file.
- Incorporate any changes from the new version of the record (prefixed with `NEWSC6.2`) into the original displayscreen record.

# Duplicate function keys

If you complete the upgrade and find that you have duplicate function keys, you most likely have added display options. Perform conflict resolution on your display options to resolve these conflicts.

# Changes to the display component upgrade

Prior to SC6.2, the Upgrade Utility upgraded the display options and display events differently than other components, such as formats or links. In release SC6.2 and later, the Upgrade Utility upgrades the display components in the same manner as all other components.

The options table and the events table have a unique identifier, stored in the “id” field. The upgrade process assigns an id to every display option following the pattern: `<screen id>_<action>_<number>`, where the screen id and action are from the display option (or event), and the number is an optional field added when multiple options have the same screen id and action.

If options that have the same action as others in the same display screen have been added to your system, the upgrade process assigns <number> in the order of the options' GUI option number.

Therefore, if the added option was not the last option in terms of GUI option number, the upgrade process does not add the additional numbers in the id field in the same order as they would have for an out-of-box system. The upgrade process renames the added option and any option after it (in GUI option order), it does not upgrade them automatically.

To ensure that this type of renaming does not happen in future upgrades, when performing conflict resolution on these options, use the id of the renamed option, NEWSC6.2<screen id>_<action>_<number> and manually change the identifier of the added options. Rename all of the other options to match the id of the renamed ones.

When renaming your option, use an identifier to specify that this is a customized option, added for your installation. For an example, an id might look like: “apm.edit.problem_do nothing_ACME1”.

This table gives an example of part of the display screen conflict resolution for apm.edit.problem.

| Screen Id | GUI Action | Upgrade Action | Display Screen option |
|---|---|---|---|
| 300 | do nothing | update | **Name:** apm.edit.problem_do_nothing_1<br>**Result:** This item was updated correctly.<br>**User Action:** No action necessary. |
| 400 | do nothing | update | **Name:** apm.edit.problem_do_nothing_2<br>**Result:** This item was updated correctly.<br>**User Action:** No action necessary. |

| Screen Id | GUI Action | Upgrade Action | Display Screen option |
|---|---|---|---|
| 450<br>This is an option you added. | do nothing | rename | **Name:** apm.edit.problem_do_nothing_3<br>**Result:** This item was renamed. It is your customized option.<br>**User Action:** Rename this object to give it a unique new name, such as: apm.edit.problem_do_nothing_ACME1<br>**Name:** NEWSC6.2apm.edit.problem_do_nothing_3<br>**Result:** This is the new SC6.2option.<br>**User Action**: Perform conflict resolution.<br>To perform conflict resolution, open apm.edit.problem and look at the options. Compare this option with apm.edit.problem_do_nothing_3 and NEWSC6.2apm.edit.problem_do_nothing_3. |
| 500 | do nothing | The upgrade ignores this option. | **Result:** This option does not appear in the reports.<br>**User Action**: Perform conflict resolution.<br>To perform conflict resolution, open apm.edit.problem and look at the options. Compare this option with apm.edit.problem_do_nothing_3 and NEWSC6.2apm.edit.problem_do_nothing_3. |

# Step 4: Key changes

During the application development process, HP OpenView ServiceCenter development may change, add, or delete database dictionary keys. The upgrade does not change any keys automatically because changing the keys requires a regen. The amount of time for the regen depends on how many records are in the table. You will need to change the keys manually after the upgrade is complete. Refer to Appendix B, Application changes by release for the key changes for this release.

# Step 5: Test the system

After you resolve all conflicts, test the upgraded system and verify that it functions properly. If there are problems that you cannot resolve, contact HP OpenView Customer Support.

# Step 6: Back up the system

Make a checkpoint backup of the data files to enable you to restore from this point, if necessary. See Step 3: Back up your production system on page 30.

**Important:** If mapped to an RDBMS, you must back up your RDBMS database whenever you back up your data. Refer the documentation for your relational database management system (RDBMS) for RDBMS backup instructions.

6 CHAPTER

# Phase VI: Performing the custom upgrade

When you begin this phase, you start with an upgraded development system that has no unresolved conflicts. You are ready to create a custom application upgrade on the development system that you can export to your test or production system.

Topics in this section include:

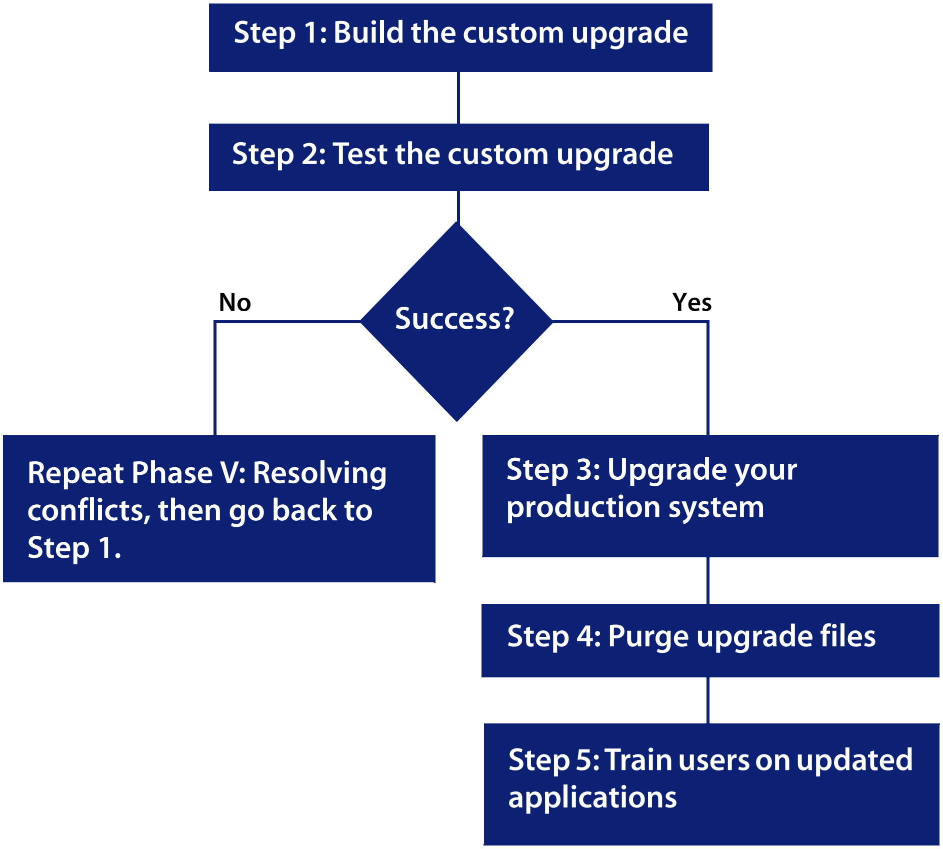
## Custom Upgrade process overview

The custom upgrade consists of three types of files:

- New SCSC6.2 application files that replaced old application files
- Customized application files that you retained
- Merged files that combine prior customization with new application functionality

HP OpenView recommends that you export to a test environment and run acceptance tests on this version before you move to live production. However, you can run acceptance tests in the same development environment where you ran the upgrade and resolved differences.

Make frequent checkpoint backups of the Data directory to enable you to retrace your steps if necessary. See Step 3: Back up your production system on page 30.These are the main steps in performing your custom upgrade. If you are viewing this document online, you can click each step to link to a detailed explanation of that step.


Step 1: Build the custom upgrade
Step 2: Test the custom upgrade
Success?
No
Yes
Repeat Phase V: Resolving conflicts, then go back to Step 1.
Step 3: Upgrade your production system
Step 4: Purge upgrade files
Step 5: Train users on updated applications

# Step 1: Build the custom upgrade

The ServiceCenter Upgrade Utility automatically exports the custom upgrade for you; however, you must specify a destination with adequate disk space.

### To build the custom upgrade

1 Use the new client to log on to the development system.

2 Use the database dictionary to delete the file scirexpert, if it exists. The upgrade recreates this file automatically.

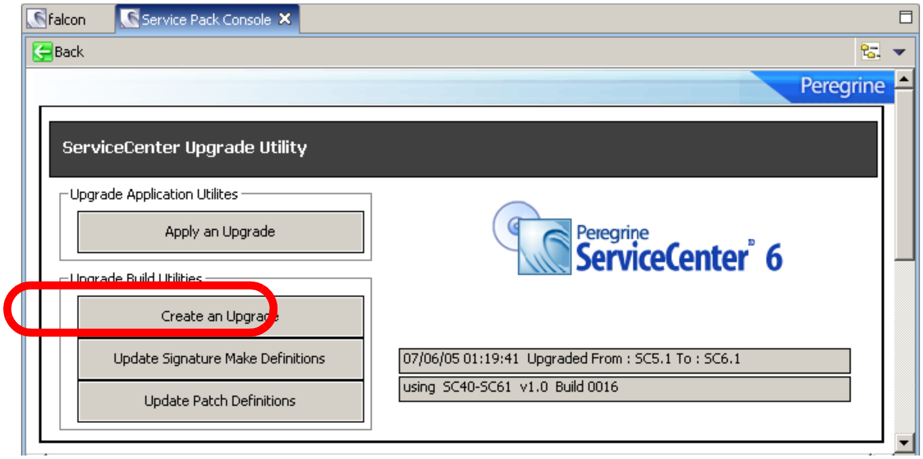
3 Verify that you have created a CustomUpgrade directory as described in Step 1 on page 30. The Upgrade directory contains the out-of-box upgrade files. You must create the custom upgrade files in a separate directory.

- Ensure that the CustomUpgrade directory is empty. If you build another custom upgrade, the Upgrade Utility appends to the files existing in the CustomUpgrade directory. It does not overwrite them.
- Your User ID must have read and write access to the CustomUpgrade directory.

4 Type `sc62upgrade` on the ServiceCenter client command line.

5 Click **SERVICE PACK** in the **Application Upgrade** menu.

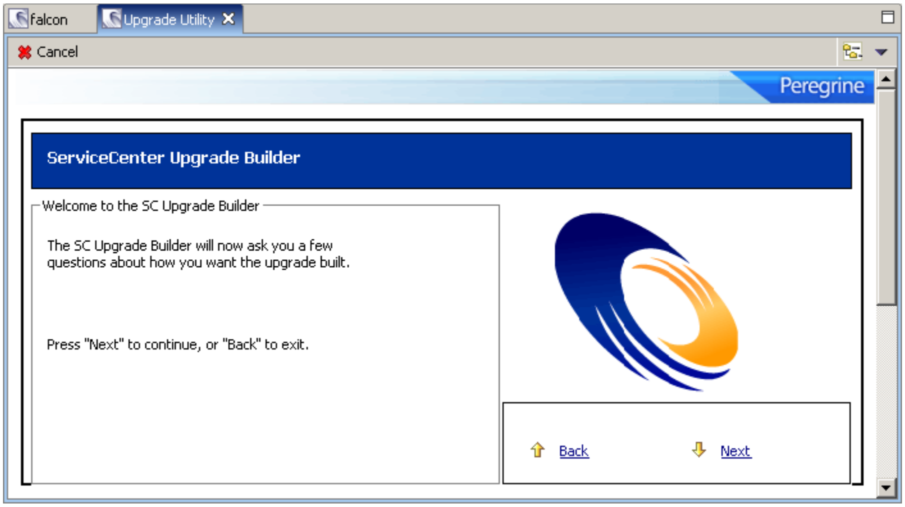
6 Click **Create an Upgrade**.



07/06/05 01:19:41 Upgraded From : SC5.1 To : SC6.1

using SC40-SC61 v1.0 Build 0016

7 Click **Next**.



ServiceCenter Upgrade Builder

Welcome to the SC Upgrade Builder

The SC Upgrade Builder will now ask you a few questions about how you want the upgrade built.

Press "Next" to continue, or "Back" to exit.

8 Type the name of the release, SC6.2. Click **Next.**

9 Type a name for the custom upgrade. Click **Next**.

10 Type the fully qualified path to the directory where the Upgrade Utility should create and export files (the empty CustomUpgrade directory). Remember this path for later use.

**Note:** This should be different than the upgrade directory used to upgrade the test system.

This path must contain a final forward slash (/) or back-slash (\), depending on your operating system. Click **Next**.

11 If the Upgrade Utility displays the screen, select the supported languages. Click **Next**.

12 Select the SC6.2 patch file from the drop-down menu. Click **Next**.

13 Select **Complete Upgrade Build**. Click **Next**.

**Note:** Choose anther option only when directed by HP OpenView Customer Support.

14 Click **Yes** to activate internal logging; click **No** to skip internal logging. Click **Next**.

15 Click **Next** to start the Upgrade Builder.

16 A warning message appears.

```
This process will destroy any existing upgrade definitions on
file. Proceed?
```

Click **Yes**. This action destroys any upgrade definitions currently on file and creates the custom upgrade for this system.

**Note:** The amount of time to create the custom upgrade varies depending on your system resources.

The ServiceCenter Upgrade Builder monitors the signature process of the build.

The signature process is complete when the Upgrade Utility displays the following message.

```
Finished creating transfer files for the upgrade.
```

17 Copy the upglang.unl and self_service.unl files from the Upgrade directory to the CustomUpgrade directory.

18 Make a checkpoint backup that includes both the data directory and the custom upgrade files. See Step 3: Back up your production system on page 30.

**Note:** The transfer.bin created by the Upgrade Utility will be up to 50% larger than the out-of-box version. The upgrade.dta file may be twice as large.

# Step 2: Test the custom upgrade

Before you can test your custom upgrade, configure a test environment where you can test all features, especially customized applications. Use the same process you used when configuring a development environment.

Add empty CustomUpgrade and Upgrade directories and complete the steps in Phase III: Preparing for an upgrade on page 27. You must complete these tasks:

- Verify that you have an archive backup of your production system DATA directory.
- Install the ServiceCenter Upgrade Utility.
- Make RDBMS preparations (optional).
- Install a ServiceCenter 6.2 client and server.
- Copy the DATA archive into the ServiceCenter 6.2 directory, replacing the out-of-box DATA directory, if you installed it.
- If upgrading from pre SC6 applications, convert the P4 database to UTF-8.

The test environment should be identical to your production environment in every way.

## Tables not upgraded by the Upgrade Utility

The Upgrade Utility does not automatically upgrade all tables and records. The patches file lists the tables and records that are moved into the custom upgrade files. Customizations made to any other table or record will not be part of the custom upgrade.

- To see the changes that were moved into the custom upgrade, refer to the patches table.

- To preserve changes not captured by the Upgrade Utility, unload the customizations and load them into the new system separately.

## Run the Upgrade Utility again

The steps to complete a custom upgrade are identical to those to complete an out-of-box upgrade with a few exceptions, such as entering the path to the custom upgrade when asked for the path to the upgrade files. Complete the steps in Phase IV: Running the Upgrade Utility on page 59. You must complete these tasks:

- Load preupg.bin and transfer.bin.
- Run the Upgrade Utility.
- Check that background schedulers are finished.

- Note the start and end time. The elapsed time indicates how long the production system will be unavailable during the production upgrade.
- Return the system to a normal operating environment.

The Upgrade Utility repeats the upgrade process, using the mirror image of the production system and custom upgrade files you created in Step 1: Build the custom upgrade on page 100. Verify that the exception reports show no conflicts. You must resolve any remaining upgrade differences.

## Test the result with ServiceCenter

Test the upgraded system with the new ServiceCenter client. If the upgrade process has any problems, you may need to repeat the steps in Phase V: Resolving conflicts on page 85 to resolve the remaining differences.

### To test the custom upgrade

1. Use the instructions in the *ServiceCenter Installation Guide*, to install and configure the ServiceCenter client for the target version.
2. Use the new ServiceCenter client to log on.
3. Review the features described in the ServiceCenter 6.2 Help.
4. Use the new ServiceCenter client to test the upgraded system thoroughly. Test all features that your users will access. Pay particular attention to areas that were modified on your system.
5. If you plan to use the Web client, you may find subtle differences with forms that need to be corrected.

When the custom upgrade works properly and passes all tests, you can use it to upgrade your production system.

# Step 3: Upgrade your production system

The final task of the upgrade process is to apply the custom upgrade that you created to your production system. This process is identical to the one you followed in upgrading your test system.

Warning: Do not apply any upgrade that has not been thoroughly tested to your production system.

When you upgrade the production system:

- The production system is not available to the users while you are applying the custom upgrade.
- Ensure the upgrade files you created are accessible to the production system (the files are located on the same server).
- If you transfer the files to your production system by FTP, set FTP to binary mode.
- You must have adequate disk space.

### To upgrade the production system

1 Ensure that you still have an archive copy of the original production system and all data.

2 Stop the production system, including clients, server, and console.

3 Log on as a system administrator.

4 Prepare the system. Complete the steps in Phase III: Preparing for an upgrade on page 27. The steps for setting up the production environment are identical to those for the development and test environments, with a few exceptions. For example you don't have to duplicate the production environment. You must complete these tasks:

   - Install the ServiceCenter Upgrade Utility.
   - Make RDBMS preparations (optional).
   - Install the client and server for the upgrade.
   - If upgrading from pre SC6 applications, convert the P4 database to UTF-8.

5 Complete the steps in Phase IV: Running the Upgrade Utility on page 59. You must complete these tasks:

   - Load preupg.bin and transfer.bin.
   - Run the Upgrade Utility.

- Check that background schedulers are finished.
- Return the system to a normal operating environment.

The Upgrade Utility repeats the upgrade process again, using the real production system and the custom upgrade files.

6 Use the new ServiceCenter client to test the upgraded system again.

When you successfully upgrade the production system, your application upgrade is complete.

# Step 4: Purge upgrade files

After you successfully applied the custom upgrade to your production system, you can remove all the unnecessary files created by this process from your development system by running the Purge utility. You do not have to purge the files if the custom upgrade was created on a backup of your development system that will no longer be used.

Warning: Do not purge the upgrade files if the upgrade fails. Do not purge the upgrade until you have the production system running successfully.

If the upgrade fails, you can restart where it stopped. It may be necessary to rebuild a custom upgrade, but you can do so only if the upgrade files have not been purged.

### To run the Purge utility

1 Type `*aapm.upgrade.purge` on the ServiceCenter client command line. Press **Enter**.

2 Click **I'm done, and I want to remove the upgrade files completely**.

3 Click **OK**.

4 Run SCDBUTIL LFMAP, Option 4. Although the upgrade data is purged, free space is not reclaimed until you run LFMAP.

# Step 5: Train users on updated applications

Before implementing the upgraded system into production, train your users on any new features that they might need. To learn about the new features in ServiceCenter 6.2, see the ServiceCenter 6.2 Help. HP OpenView Education Services also offers a complete training schedule.

APPENDIX

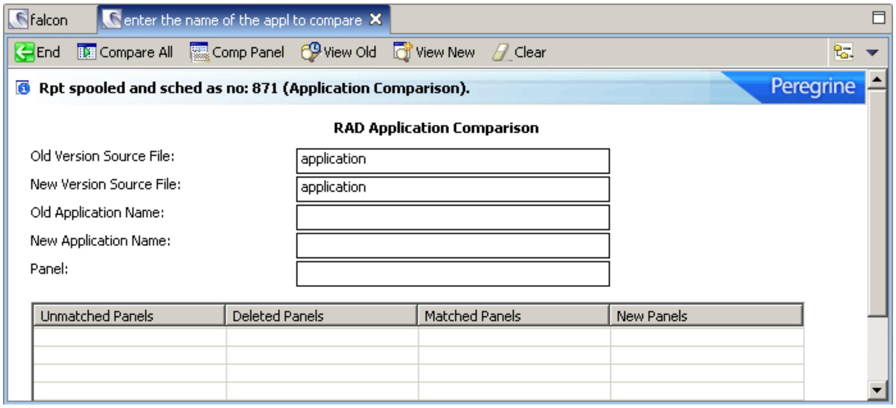
# A Using the RAD Comparison Utility

HP OpenView bundles the RAD Comparison Utility with the ServiceCenter Upgrade Utility to help you examine the differences between the new and old versions of an application. The RAD Comparison Utility displays a list of differences between elements, such as panels and lines that do not match. Use this utility to examine those applications where you must decide to keep or replace the old version or merge objects into a single version. You can view the comparison results online or print them. You can also view the old and new versions of each panel.

Topics in this section include:

# The RAD Comparison Utility

To access the RAD Application Comparison utility, type `agcompare` on the ServiceCenter client command line. Press **Enter**.



Rpt spooled and sched as no: 871 (Application Comparison).

RAD Application Comparison

| | |
|---|---|
| Old Version Source File: | application |
| New Version Source File: | application |
| Old Application Name: | |
| New Application Name: | |
| Panel: | |

| Unmatched Panels | Deleted Panels | Matched Panels | New Panels |
|---|---|---|---|

## Toolbar buttons

The following table describes the functions in the Toolbar.

| Button | Definition |
|---|---|
| End | Returns you to the previous screen. |
| Compare All | Compares all the panels  of the new version of the application named with all the panels of the old version. |
| Comp Panel | Compares the old and new versions of a single panel named in the Panel field of the Compare Application form. |
| View Old | Displays the old version of the panel named in the Panel field of the Compare Application form. |
| View New | Displays the new version of the panel named in the Panel field of the Compare Application form. |
| Clear | Clears all data from the panel comparison fields (Unmatched, Deleted, New, and Matched), and prepares the utility to perform another comparison. This option does not affect the old or new RAD applications. |
| Print All | Sends all currently displayed panel records to the printer. |

## Fields

The following table describes the fields on the RAD Application Comparison form.

| Field | Definition |
|---|---|
| Old Version Source File | Specifies the logical file containing the application records (RAD panels) referenced by **Old Application Name**. The default value of this field is `application`. |
| New Version Source File | Specifies the logical file containing the application records (RAD panels) referenced by **New Application Name**. The default value of this field is `application`. |
| Old Application Name | Specifies the name of the application that resides in **Old Version Source File**. The old version source file is the model for the application specified by **New Application Name**. |
| New Application Name | Specifies the name of the application that resides in **New Version Source File** that you want compare with the application specified by the **Old Application Name**. |
| Panel | Specifies the panel the utility should use for comparison checking. You can view the specified panel in the old or new version of the application. |
| Unmatched panels | A list of panel names where there are differences between the old and new versions of the RAD application.<br>Warning: Do not make manual changes to the content of this field. |
| Deleted panels | Contains a list of panel names in the old version of the application that are not in the new version.<br>Warning: Do not make manual changes to the content of this field. |
| Matched panels | Contains a list of panel names where there are no differences between the old and new versions of the RAD application.<br>Warning: Do not make manual changes to the content of this field. |
| New panels | Contains a list of all panel names in the new version of the application and not in the old version. This field is set to allow for modifications and easy tabbing.<br>Warning: Do not make manual changes to the content of this field. |

**Important:** By default, ServiceCenter contains one Application Library. If you find it necessary to maintain two or more Application Libraries, you must allocate and control the functionality of these files, and define the routines necessary for the exchange of data between files.

# Comparing applications

You can compare single panels or all panels in an application.

### To compare all panels

1 Access the RAD Application Comparison utility.

2 Type the appropriate data in the **Old Application Name** and **New Application Name** fields. Depending on the choice you made during the upgrade process, either the new version will have the name `NEW<vvvnnn>`, where `vvv` is the release name and nnn is the name of the object, or the old version will have the name `OLD<vvvnnn>`. For example, a new check.field.name could be `NEWSC6.2check.field.name`, or the old one could be `OLDSC4check.field.name.`

3 Click **Compare All**. ServiceCenter displays summary lists of Matched Panels, Unmatched Panels, Deleted Panels, and New Panels where they apply. Make selections from these lists and display additional panel information by choosing options in the Toolbar.

4 You can view old and new versions of application panels, review the detailed comparison results for panels in the Unmatched Panels list, and print all the panels.

5 To compare a different application, click **Clear** before proceeding. This resets all controls before processing begins.

## Printing a report

To print a report of the Comparison Results, click **Print All**. The comparison report contains the following information.

| Reported Item | Definition |
|---|---|
| Matched Panels | Summary of all panels that matched. |
| Unmatched Panels | Summary of all panels showing a change, printouts of the old and new versions of each unmatched panel, and the comparison results of each panel. |
| Deleted Panels | Summary of all deleted panels and a printout of each deleted panel. |
| New Panels | Summary of all new panels and a printout of each panel. |

# Comparing single panels

You can compare single panels before or after comparing the entire application.

### To view a detailed comparison of a single panel after comparing the entire application

- Select the desired panel in the **Unmatched Panels** column. Click **Comp Panel**.

### To view a detailed comparison of a single panel before comparing the entire application

- Type the name of the panel in the **Panel** field. Click **Comp Panel**.

The Detail Listing of Differences form displays the names of the Old and New Version Source Files and the names of the Old and New Application Name fields. The following figure shows the Detail Listing of Differences report.

## Array and scalar field differences

When the old and new versions of an array or scalar field have differences, the information appears in the following format.

```
Line xx:action prompt
```

where:

`xx` is the element number of the array that the Upgrade Utility deleted or added.

`action` is the change to an element of an array. The old version of the element is considered deleted and the new version of the element is considered added. Changes to array or scalar field elements are accompanied by the phrases ADDED TO and DELETED FROM.

`prompt` is the name of the prompt on the RAD panel that corresponds to the array or scalar field that the Upgrade Utility changed.

## Viewing versions of a panel

Click **View Old** or **View New** from the Detail Listing of Differences form or from the RAD Comparison form after a complete comparison. Type the name of the panel in the **Panel** field or select the panel from one of the lists.

## Continuation lines

If the contents of a scalar field or an element of an arrayed field are greater than 72 bytes, the comparison utility uses continuation lines. Continuation lines appear in viewable and printed comparison results.

All continuation lines start with three asterisks (***). The last two characters of a detail line appear as the first two characters of the next continuation line.

# ERD Creator definitions

You can use the ERD Creator to determine which files are associated with an application.

### To obtain the list of files associated with an application

1. Type `db` on the ServiceCenter client command line. Press **Enter**.
2. Type `erdcreate` in the File field. Click **Search**.
3. Click **Search** again to list all the records in this table. Database Manager displays a list of ServiceCenter applications.
4. Double-click any application. ServiceCenter displays a list of files related to the application you selected.

# B Application changes by release

APPENDIX

This appendix contains cumulative information by release about changes to ServiceCenter applications.

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC4 | All | Database Manager now uses the Document Engine instead of specific Database Manager RAD applications. Many of the old RAD applications are modified to call the Document Engine, or they are no longer called at all. | Review your customized Database Manager RAD applications to determine if they should be called by the Document Engine. |
| SC4 | Incident Management | Problem Management renamed to Incident Management. | Review customized menus, formats, and messages that use the Problem Management name to determine if they need to be changed to match the new naming convention. The out-of-box naming convention for incidents is IM*nnn*. |
| SC4 | Request Management | If your system does not use approvals defined at the line item or model (part) level, you may now turn off this option at the quote/order phase level to improve performance when you schedule approvals in Request Management. For example, you may use only quote/order level approvals instead of line item or part level approval. | To disable this option for specific quote or order phases, set the **Use Line Item/Model Approvals** field on the Approvals tab of the appropriate phase record to false. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC4 | Request Management | References to quote, line item, and order file variables in categories and phases has been changed from $fileq, $filel, and $fileo to $L.file. | The Upgrade Utility automatically converts variables in the category and phase definitions. If you have customized code or tailoring that references these file variables, you should review it to update these references. |
| SC4 | Request Management | Request Management alerts and approvals now use the Document Engine alerts and approvals. There are new files that control this process and store the alert and approval information. The new files are:<br>Alert, AlertDef, Alertlog, Approval, ApprovalDef, ApprovalLog. | For your information; no action required.<br>The Upgrade Utility copies existing alerts, approvals, and definitions into the new files. You should review any customized code that references the old files and update your reports to point to the new files. |
| SC4 | Request Management | Request Management now uses the Document Engine instead of specific Request Management RAD applications. Many of the old RAD applications are modified to call the Document Engine, or they are no longer called at all. | Review your customized Request Management RAD applications to determine if they should be called by the Document Engine. |
| SC4 | Request Management | The following RAD applications now invoke the Document Engine when called:<br>ocm.search.objects, ocml.access, ocml.main, and ocmq.main. | If these RAD application names appear in the exception report, you must use the new HP version of the RAD application. If you customized these applications, you must re-implement them by using the Document Engine. |
| SC4 | Request Management | The prefix on Request Management displayscreens has changed from ocm.* to rm.*. For example, ocm.main.display is now rm.main.display. | Review your old customized displayscreens and displayoptions for changes that you must apply to the new displayscreens and displayoptions. |
| SC4 | Request Management | The Request Management catalog (model file) structure now uses dependencies instead of level and sequence. | For your information; no action required.<br>The Upgrade Utility automatically converts the level and sequence model to the dependencies model. Review any customized code that references the level and sequence and modify it to use the dependencies model. For more information, see the Request Management documentation. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC4 | Request Management | You can now select multiple catalog items before opening a quote. This is controlled at the Quote category level. | For your information; no action required.<br>To disable this feature, type `False` in the Multiple Selections field in the category record. |
| SC4 | Root Cause Analysis | New application added automatically by the Upgrade. | For your information; no action required. |
| SC4 | Scheduled Maintenance | New application added automatically by the Upgrade. | For your information; no action required. |
| SC4 | Utilities | The Notification engine is a new feature. | You should disable any messages generated by other sources. |
| SC5 | All | Profile groups are no longer used. The operator record now stores the names of user profiles for each application. | The upgrade automatically populates these fields in operator records based on the old profile rules. If you have customized operator forms, you should add these new fields to those forms. |
| SC5 | All | ServiceCenter creates index entries for any record with a NULL value in a key field. Then ServiceCenter indexes these NULL entries. | If this message appears in the log or active notes:<br>Unable to find entry for key #5 in file '<*filename*>' (<indexed field>=NULL).<br>Regen the file named in the error message. |
| SC5 | All | The useless.unique.key field in the status dbdict changes from character to a numeric data type. | For your information; no action required.<br>If you use this field, ensure that data in this field is numeric. |
| SC5 | All | There are changes to the company record to add passwords and password security information. | Although the Upgrade adds these fields automatically to existing tables, you should revise your info.company.g form to view these new fields on the password and message processor tabs. |
| SC5 | Change Management | Change Management alerts and approvals now use the Document Engine alerts and approvals. There are new files that control this process and store the alert and approval information. The new files are:<br>Alert, AlertDef, Alertlog, Approval, ApprovalDef, ApprovalLog. | For your information; no action required.<br>The Upgrade Utility copies existing alerts, approvals, and definitions into the new files. You should review any customized code that references the old files and update your reports to point to the new files. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC5 | Change Management | Change Management now uses the Document Engine instead of specific Change Management RAD applications. Many of the old RAD applications are modified to call the Document Engine, or they are no longer called at all. | Review your customized Change Management RAD applications to determine if they should be called by the Document Engine. |
| SC5 | Change Management | The data type for sched.outage.end field cm3r dbdict changed from character to a date/time data type. | For your information; no action required. |
| SC5 | Change Management | The data type of the number field has been changed number to character for cm3r, cm3t, cm3rpage, and cm3tpage. | Any references to these fields in customized RAD applications for tailoring may need to change to accommodate the changed data type. |
| SC5 | Change Management | The following RAD applications now invoke the Document Engine when called:<br>cm3r.main and cm3t.main. | If these RAD application names appear in the upgrade exception report, you must use the new HP version of the RAD application. You should review any customized code that references these applications and re-implement it by using the Document Engine. |
| SC5 | Incident Management | Incident Management now uses the Document Engine instead of specific Incident Management RAD applications. Many of the old RAD applications are modified to call the Document Engine, or they are no longer called at all. | Review your customized Incident Management RAD applications to determine if they should be called by the Document Engine. |
| SC5 | Incident Management | References to the header, middle, or Action structures of the problem table records in the notification, format, formatctrl, macro, and screlconfig tables are removed. | Review your customized Incident Management RAD applications to determine if they should be called by the Document Engine. |
| SC5 | Incident Management | References to the header, middle, or Action structures of the problem table records in the notification, format, formatctrl, macro, and screlconfig tables are removed. | For your information; no action required. |
| SC5 | Incident Management | The following RAD applications now invoke the Document Engine when called:<br>apm.edit.problem<br>apm.first<br>apm.list.problems<br>apm.search.problems<br>apm.show.costs | If these RAD application names appear in the upgrade exception report, you must use the new HP version of the RAD application. You should review any customized code that references these applications and re-implement it by using the Document Engine. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC5 | Incident Management | The probsummary table is the main table for Incident Management. ServiceCenter now uses the problem table only for paging. The Upgrade Utility automatically migrates existing fields (and associated data) in the problem table to the probsummary table. | Review any custom code that references the problem table and update it to use the probsummary table. |
| SC5 | System Information | Multi-Company mode is now specified in the System Information record instead of in the login.DEFAULT format control record. | If you use a customized version of the login.DEFAULT format control record, remove the line that controls Multi-Company mode. |
| SC51 | Inventory Management | Arrays of structures have support for subtables. An array of structures can be stored in another table. | If you reference a field in an array of structures by name only, you will call out the entire array. If you reference a single field, change the reference to qualify the position of the field within the array of structures. For example: *structure, fieldname, n*. If you currently reference *structure, fieldname*, you will not identify the field correctly. |
| SC51 | Inventory Management | Inventory Management now uses the Document Engine instead of specific Inventory Management RAD applications. Many of the old RAD applications are modified to call the Document Engine, or they are no longer called at all. | Review your customized Inventory Management RAD applications to determine if they should be called by the Document Engine. |
| SC51 | Inventory Management | The following RAD applications now invoke the Document Engine when called:<br>icm.access, icm.display.related, and icm.main. | If these RAD application names appear in the upgrade exception report, you must use the new HP version of the RAD application. You should review any customized code that references these applications and re-implement it by using the Document Engine. |
| SC51 | Inventory Management | The Inventory Management model changed. As a result, much of the attribute data is stored in subtables using an array of structures. Join files, such as device*xxxxxx*, no longer exist. All of the device types in earlier versions are consolidated in SC51. | If you do not use this feature, no action is required. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC51 | Inventory Management | Updated the devtype table to support join definition (joindef) files. Removed the joinfile dbdict. Moved data from the subtype file and stored in devtype record. The Upgrade checks both joindef and erddef records and creates them whenever if necessary. There are new triggers for the Cascade Update utility to use and triggers for background processing. | Future maintenance should be on the devtype record, not the subtype file. |
| SC51 | Key changes | The probsummary table has a new key.<br>The 23rd key is a new Nulls & Duplicates key.<br>**Note:** This change is required.<br>• ndthread | Make the required key change to the probsummary table. |
| SC51 | Key changes | The eventin table has a new key.<br>The eighth key is a new Nulls & Duplicates key.<br>**Note:** This change is required.<br>• evusrseq | Make the required key change to the eventin table. |
| SC51 | Key changes | The model table has a new key.<br>The eighth key is a new Nulls & Duplicates key to support new software compliance functionality.<br>**Note:** This change is required.<br>• rights | Make the required key change to the model table. |
| SC51 | Key changes | The vendor table has two new keys to support the new inventory model.<br>The eighth key is a new Nulls & Duplicates key.<br>**Note:** This change is required.<br>• is.vendor<br>• vendor<br>The ninth key is a new Nulls & Duplicates key.<br>**Note:** This change is required.<br>• is.manufacturer<br>• vendor | Make the two required key changes to the vendor table. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC51 | Key changes | The eventregister table has a new key.<br>The fifth key is a new Nulls & Duplicates key.<br>**Note:** This change is required.<br>• evftype<br>• evtype<br>• evappl<br>• evseq | Make the required key change to the eventregister table. |
| SC51 | Key changes | The eventout table has a new key.<br>The fourth key is a new Nulls & Duplicates key.<br>**Note:** This change is required.<br>• evusrseq<br>• evsysseq | Make the required key change to the eventout table. |
| SC51 | Key changes | The currency table has two new keys.<br>The fourth key is a new Nulls & Duplicates key.<br>**Note:** This change is required.<br>• active<br>The fifth key is a new Nulls & Duplicates key to support queries against new currency fields.<br>**Note:** This change is required.<br>• active<br>• currency.code | Make the two required key changes to the currency table. |
| SC51 | Key changes | The expline table has a new key.<br>The sixth key is a new Nulls & Duplicates key.<br>**Note:** This change is required.<br>• asset | Make the required key change to the expline table. |
| SC51 | Key changes | The patcotask table has a new key.<br>The sixth key is a new Nulls & Duplicates key.<br>**Note:** This change is required.<br>• active<br>• v51trigger.device | Make the required key change to the patcotask table. |
| SC51 | Key changes | The cmlabor table has a new key.<br>The third key is a new Nulls & Duplicates key.<br>**Note:** This change is required.<br>• asset | Make the required key change to the cmlabor table. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC51 | Key changes | The cmparts table has a new key.<br>The third key is a new Nulls & Duplicates key.<br>**Note:** This change is required.<br>• asset | Make the required key change to the cmparts table. |
| SC51 | Key changes | The devaudit table has a new key.<br>The third key is a new Nulls & Duplicates key.<br>**Note:** This change is required.<br>• filename<br>• last.record<br>• object1<br>• object2<br>• object3 | Make the required key change to the devaudit table. |
| SC51 | Key changes | The pcsoftware table has a unique key change.<br>**Note:** This change is required.<br>• license.number<br>• logical.name<br>• application.name<br>The fourth key is a new Nulls & Duplicates key to support new software compliance functionality.<br>**Note:** This change is required.<br>• license.number<br>The fifth key is a new Nulls & Duplicates key.<br>This change is recommended for performance reasons.<br>• license.number<br>• suite.component<br>• status<br>• authorized | Make the required key changes to the pcsoftware table.<br>The upgrade lists existing pcsoftware table records that must be updated. Evaluate whether adding the third recommended key will improve performance on your system. |
| SC51 | Link records | Introduced master link record functionality. For more information, see master link record documentation. | If you have link records with the same name as a dbdict, they are treated as a master link record. Review all links that fit this criteria to ensure that this does not create errors. |
| SC51 | Service Level Agreements | There is a new status progression field on an SLA control record to monitor the SLA life cycle. | Review SLA progression to ensure your customized status progression is accurate. |
| SC6 | IR Expert | IR Expert indexes now reside in the scirexpert table instead of in the external ir.* files. | For your information; no action required.<br>If the physical files exist, perform an IR regen. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6 | Key changes | The schedule table has a new key.<br>The ninth key is a new Nulls & Duplicates key.<br>**Note:** This change is required.<br>• application<br>• object | Make the required key change to the schedule table. |
| SC6 | Key changes | The erddef table has a new key.<br>The second key is a new Nulls & Duplicates key:<br>• distributed | Make the required key change to the erddef table. |
| SC6 | Link records | The target.file/format name split into target file name and target format name in the Link record. The result is a new column on the Link Record form. | Although the Upgrade adds this field to the existing table, you should revise your Link record form to view the new column. |
| SC6 | System Information | The default value of the case.insensitive field in the info record is false. | For your information; no action required. |
| SC6 | System Navigator | If you used a Java client and you are migrating to new SC6 client, the group field description in the menu record was expressed as *xxx:yyy* group name. The SC6 upgrade puts the portion preceding the double colon (:) into the Group field. The Navigator should look the same in the new client as it does in the Java client. | For your information; no action required. |
| SC6 | Utilities | Differential upgrade | For your information; no action required. |
| SC6 | Utilities | Erdcreate utility | For your information; no action required. |
| SC6.1 | Attachments | In earlier versions, the server stored all attachments in a record as a single blob in the SYSBLOB table. Now, the ServiceCenter server stores each attachment individually in the SYSATTACHMENTS table. | Use the attachment upgrade utility to convert attachments from the SYSBLOB table to the SYSATTACHMENT table. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6.1 | Incident Management | Format prefixes were renamed to follow the new "Incident Management" ITIL naming convention. The old convention used the prefix "problem" (for Problem Management). The new convention uses the abbreviation "IM" (for Incident Management). Because of the renames, category records may reference form names that begin with "problem". | Change the prefix of category records from "problem" to "IM". |
| SC6.1 | Key changes | The activityactions table has new Unique keys:<br>• name<br>• name,table | For your information; no action required. |
| SC6.1 | Key changes | The activitytype table has new Unique keys:<br>• activity.number<br>• activity.number,table | For your information; no action required. |
| SC6.1 | Key changes | The Alert table has new No Nulls keys:<br>• file.name,unique.key,active,alert.type<br>• file.name,unique.key,active<br>• unique.key,file.name,active,alert.time,alert.type | For your information; no action required. |
| SC6.1 | Key changes | The calholidays table has a new Unique key:<br>• holiday,start.date,end.date | For your information; no action required. |
| SC6.1 | Key changes | The company table has a new Nulls & Duplicates key:<br>• show.company | For your information; no action required. |
| SC6.1 | Key changes | The contacts table has a new No Duplicates key:<br>• user.id | For your information; no action required. |
| SC6.1 | Key changes | The contacts table has new Nulls & Duplicates keys:<br>• operator.id<br>• user.id | For your information; no action required. |
| SC6.1 | Key changes | The datadict table has a new No Duplicates key:<br>• sqlbasename. | For your information; no action required. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6.1 | Key changes | The diffupg table has new Unique keys:<br>• system.name,tag<br>• tag,system.name | For your information; no action required. |
| SC6.1 | Key changes | The displayevent table has a new Unique key:<br>• unique.id | For your information; no action required. |
| SC6.1 | Key changes | The displayevent table has new Unique keys:<br>• id<br>• screen.id,language,event,event.sig | For your information; no action required. |
| SC6.1 | Key changes | The displayoption table has a new No Nulls key:<br>• screen.id,gui.option,action | For your information; no action required. |
| SC6.1 | Key changes | The displayoption table has a new Nulls & Duplicates key:<br>• unique.id | For your information; no action required. |
| SC6.1 | Key changes | The displayoption table has new Unique keys:<br>• id<br>• screen.id,language,gui.option,gui.sig<br>• screen.id,language,txt.bank,txt.option,text.sig | For your information; no action required. |
| SC6.1 | Key changes | The help table has a new Nulls & Duplicates key:<br>• pending.review | For your information; no action required. |
| SC6.1 | Key changes | The inbox table has a new No Duplicates key:<br>• inbox.id | Make the required key change to the inbox table. |
| SC6.1 | Key changes | The inbox table has a new Nulls & Duplicates key:<br>• inbox.type,inbox.class,operator.name,groups | Make the required key change to the inbox table. |
| SC6.1 | Key changes | The inbox table has new fields in the unique key. Change the fields of the existing unique key to:<br>• container.id<br>• inbox.name<br>• inbox.type<br>• operator.name | Make the required key change to the inbox table. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6.1 | Key changes | The incidents table has new Nulls & Duplicates keys:<br>• alternate.contact,callback.contact,contact.name,open,opened.by<br>• assignment,open<br>• contract.id,open<br>• open,assignment<br>• open,contract.id | For your information; no action required. |
| SC6.1 | Key changes | The irqueue table has a new Nulls & Duplicates key:<br>• hold | For your information; no action required. |
| SC6.1 | Key changes | The ocml table has a new Nulls & Duplicates key:<br>• parent.quote,parent.line.item,addl.parents,open | For your information; no action required. |
| SC6.1 | Key changes | The operator table has a new Nulls & Duplicates key:<br>• ess.access.only,template.operator,name | For your information; no action required. |
| SC6.1 | Key changes | The outagedetail table has new keys:<br>No Nulls:<br>• logical.name,agreement.id,slo.id,year,month<br>• logical.name,agreement.id,year,month,<br>Unique:<br>• logical.name,agreement.id,slo.id,month,year,outage.id | For your information; no action required. |
| SC6.1 | Key changes | The probsummary table has a new Nulls & Duplicates key:<br>• flag,prob.mgmt.candidate | For your information; no action required. |
| SC6.1 | Key changes | The rootcause table has a new Nulls & Duplicates key:<br>• id,open | For your information; no action required. |
| SC6.1 | Key changes | The schedule table has new Nulls & Duplicates keys:<br>• application,object,type<br>• application,object | For your information; no action required. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6.1 | Key changes | The sla table has new keys:<br>No Nulls:<br>• expiration<br>Unique:<br>• title | For your information; no action required. |
| SC6.1 | Key changes | The slaactive table has new keys.<br>No Duplicates key:<br>• foreign.key,foreign.filename<br>Nulls and duplicates key:<br>• agreement.id,active | Add the new keys to the slaactive table. |
| SC6.1 | Key changes | The slaactive table has new keys:<br>No Duplicates:<br>• foreign.key,foreign.filename,<br>Nulls & Duplicates:<br>• agreement.id,active | For your information; no action required. |
| SC6.1 | Key changes | The slamonthly table has a modified unique key.<br>• agreement.id, slo.id, logical.name, year, month | Change the old unique key<br>• agreement.id, logical.name, year, month<br>to the new unique key.<br>• agreement.id, slo.id, logical.name, year, month |
| SC6.1 | Key changes | The slamonthly table has a new Nulls & Duplicates key.<br>• logical.name | Add the new key to the slamonthly table. |
| SC6.1 | Key changes | The slamonthly table has new keys:<br>Nulls & Duplicates:<br>• logical.name,<br>Unique:<br>• agreement.id,logical.name,year,month<br>• agreement.id,slo.id,logical.name,year,month | For your information; no action required. |
| SC6.1 | Key changes | The slaresponse table has a new Nulls & Duplicates key.<br>• slo.id, year, month | Add the new key to the slaresponse table. |
| SC6.1 | Key changes | The slaresponse table has a new Nulls & Duplicates key:<br>• slo.id,year,month | For your information; no action required. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6.1 | Key changes | There are changes to the sla table.<br>• Removed the Unique key: title<br>• Removed the No Nulls key: keys,expiration | For your information; no action required. |
| SC6.1 | Problem Management (formerly RootCause) | New categories, phases and tasks, similar to those in Change Management, have been added, along with alerts and notifications. | For your information; no action required. |
| SC6.1 | Problem Management (formerly RootCause) | Problem Management, formally known as RootCause, now has three different tables: rootcause, rootcausetask and knownerror. | For your information; no action required. |
| SC6.1 | Problem Management (formerly RootCause) | The profile records have been updated to contain information for Problems, Problem Tasks, and Known Errors. | After the upgrade only the PROBLEM MANAGER profile will have privileges for Problem Management Tasks or Known Errors or any new options added for Problems.<br>Update the profile records to enable users to view and modify records within Problem Management. |
| SC6.1 | RDBMS Support: DB2 | Additional fields must now be mapped as BLOBs. | Put these SQL Hints records in the sqlhints file:<br>**File Name** / **Field Name** / **BLOB**<br>ScriptLibrary / script / checked<br>triggers / javascript / checked<br>ServiceCenter will map these fields as BLOBs. |
| SC6.1 | RDBMS Support: DB2 | HP OpenView ServiceCenter has dropped support for DB2universal. | Use DB2ux. |
| SC6.1 | Service Level Management | Service Level Agreements are now stored in two files. The availability and response time objectives are now stored in a separate slo table. | For your information; no action required. |
| SC6.1 | Service Level Management | Service Level Agreements have a new field that indicates when the terms for the SLA are valid (commenced). ServiceCenter uses the start date to determine whether an SLA is valid when you set it to automatically select the appropriate SLA for a tickets being opened. The start date may also be needed for accurate reporting. | The upgrade populates this field with the date January 1, 2001, for all existing SLAs.<br>Review each SLA and enter the correct start date in the field labeled "SLA in effect from". |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6.1 | Service Level Management | Service Level Management no longer uses clocks to store response time information. Instead, this information is stored in the sloresponse table. | For your information; no action required. |
| SC6.1 | Service Level Management | Service Level Management now uses configuration records for each application to determine the fields and conditions that are used during SLA processing.<br>The upgrade populates the Response States for the Incident Management application with information from the pmstatus table. Each record in the pmstatus table has a sequence number, and this sequence determines the order of the states in the Response State Progression array field of the probsummary slamodulecontrol record (the SLA configuration record for the Incident Management application). | Review the Response State Progression field and modify or remove any states that do not apply to your implementation. Ensure that the states in this field are listed in the order that they should occur during the life cycle of an incident ticket. (For example, Closed comes after Open). |
| SC6.1 | Service Level Management | Service Level Management now uses profiles to determine user rights within the module. | After the upgrade, all users will have the DEFAULT profile which has limited permissions.<br>• Review the available profiles, edit them or add new, as necessary.<br>• Update the SLM users to have the correct profile.<br>• Update the operator records of each operator who needs access to SLA, to give appropriate permissions. |
| SC6.1 | Service Level Management | SLA now has user profiles. After the upgrade, no operator will have an assigned SLA profile. All operators will use the DEFAULT profile, which has limited permissions. | Update the operator records of each operator who needs access to SLA, to give appropriate permissions. |
| SC6.1 | Service Level Management | The “target” field no longer exists in the out-of-box system for the sla table. The data policy record had a Validation Rule for this field, and will still be present in the data policy record following the upgrade. | Remove this validation line from the datadict record for “sla”.<br>Update the operator records of each operator who needs access to SLA, to give appropriate permissions. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6.1 | Service Management | New Calls now use three different fields for contacts:<br>• Contact for this call (callback.contact)<br>• This call is for (contact.name)<br>• Reported By (alternate.contact)<br>Once you select a contact, the other two fields, if left blank, default to the first field upon saving. There is no check box for alternate.contact field; this field is now always populated. | Specify the contact for these call fields for the old tickets in the system. |
| SC6.1 | Service Management | Self Service Ticketing enables you to create self service operators by using the options menu. | To enable this feature, load self_service.unl. You may load it either before or after the application upgrade. This loads three files:<br>• SELF SERVICE profile: Grants specific privileges to the self service operator such as browse, open, or update.<br>• SELF SERVICE role: Defines a set of application profiles, capability words, and query groups for a self service operator.<br>• Template_SelfService: An operator template record that contains all required operator settings.<br>After loading these files, open the contact records, and select **Enable for Self Service** or **Mass Enable for Self Service** from the **Options** menu to create self service operators. |
| SC6.1 | Web Services API | The “WSDL configuration” entry on the HOME menu Toolkit tab (where the Database Dictionary button used to be), enables you to access to the joined extaccess record. | For your information; no action required. |
| SC6.1 | Web Services API | The extaccess table was joined with the data policy to display both extaccess data and the new API fields in the data policy (datadict) record.<br>You can view and update the data from both tables from one screen. | For your information; no action required. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6.1 | Web Services API | The extaccess table was updated with these fields:<br>• Object Name: Adds the ability to alias the table name as an object.<br>• Action Names: Adds the ability to alias the allowed actions.<br>• Service name: Adds the ability to group a number of tables as a service. | If you are upgrading from 5.1 or older, no action required.<br>If you are upgrading from 6.0 to 6.1, and you changed any of the extaccess records in 6.0, make the new changes to those records manually or update the 6.1 record with the changes you made to the 6.0 record. |
| SC6.1 | Web Services API | Three new fields were added to the data policy table for managing the new API and to the data policy record for managing the new Web Services API.<br>• API Caption: Provides a caption for a field instead of the actual field name in the dbdict. No spaces are allowed in the caption.<br>• Exclude from API: A boolean field that excludes a field from the API when set to true.<br>• API Data Type: Enforces type of data moving in and out of the system by enforcing certain XML data types. | For your information; no action required. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6.2 | ALL | ServiceCenter 6.2 requires that every operator have a contacts record. You can have a contacts record without an operator.<br>Removed redundant fields from the operator table to enable the synchronization of the contact and operator records.<br>• authcode<br>• authorid<br>• beeper<br>• beeper.group<br>• beeper.name<br>• beeper.type<br>• change.approvals<br>• change.groups<br>• change.mgr.group<br>• email.events<br>• external.flag<br>• fax<br>• gl.number<br>• graphics.printer<br>• manager<br>• nodeid<br>• page.response<br>• pager.pin<br>• pager.vendor<br>• pm.groups<br>• priority<br>• room<br>• sc.manager<br>• syslanguage<br>• time.zone<br>• voice.mailbox<br>• wdIsReadOnlyView<br>• wdMailBoxName<br>• wdResTimeStamp<br>• wdResType<br>• wdUD1<br>• wdUD2 | If both contact and operator records already exist, no action is required. If you only have operators, you may want to create the missing contact records.<br>The ServiceCenter ID in the contacts records maps to the Contact ID in the operator records.<br>• To update operator records: Contact ID (contact.name) = contact.name on the contacts record.<br>• To update contacts records: ServiceCenter ID (operator.id) = Login Name (name) on the operator record.<br>We provide four wizards that you can use to create the missing user or operator records. These wizards are available on from the Options menu of the forms contacts.g and operator.g.<br>• Create Operator - Creates an operator record from a contact record.<br>• Mass Create Operators - Creates a batch of operator records from a batch of contact records<br>• Create Contact- Creates a contact record from an operator record.<br>• Mass Create Contacts - Creates a batch of contact records from a batch of operator records.<br>These wizards call the `createUser` script in the ScriptLibrary. If you have customized data mapping or if you want to customize your data mapping, you can modify these functions in the in `createUser` before running the wizard:<br>• `MapOperatorToContact()`<br>**Note:** Be sure to turn Triggers off if you are running contact based authentication or if you are using Get-It and don't want to sync the contacts and operator records. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6.2 | ALL | The Option menus items for contacts.g have changed. In 6.1 choices were Enable Self Service & Mass Enable Self Service. In 6.2, the new choices are Create Operator and Mass Create Operator. | For your information; no action required. |
| SC6.2 | Attachments | In version 6.0 and earlier, the server stored all attachments in a record as a single blob in the SYSBLOB table. Now, the ServiceCenter server stores each attachment individually in the SYSATTACHMENTS table. | Use the attachment upgrade utility to convert attachments from the SYSBLOB table to the SYSATTACHMENT table. |
| SC6.2 | capability words | The upgrade utility makes the following capability word changes:<br>• CM3Admin -> ChMAdmin<br>• SMAdmin -> SDAdmin | For your information; no action required. |
| SC6.2 | cm3r and cm3t tables | The Upgrade utility renames the priority field to priority.code so that the three fields used for the ITIL priority calculations are consistent across modules.<br>The Upgrade utility aliases the original priority field to priority.code. Because the field has been aliased, any format control etc. applied to the old data will still work.<br>If the priority.code field already exists in the dbdict, the upgrade does not add it as an alias. | Unless you had a previously existing priority.code field in cm3r, or cm3t, no action is required.<br>If you had a previously existing priority.code field and you do not want to use the new functionality, you should do one of the following:<br>A. Remove the calculation $L.pri.calc=jscall("PriorityCalc.getPriorityCalc") from the following new Process records:<br>• cm.close.save<br>• cm.open.save<br>• cm.update.save<br>— or —<br>B. Modify the "PriorityCalc" ScriptLibrary. If you choose to update the ScriptLibrary "PriorityCalc" java script, you should be aware that other modules also use that code to calculate priority from the Initial Impact and Urgency.<br>If you do not want to use the new ITIL based Priority calculation functionality, it is easy to remove the calculation from the Process record.<br>If you do not remove the calculation, it will attempt to run using what ever information is in the severity field, which could cause unpredictable behavior. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6.2 | deviceparent table | The deviceparent table has been removed from SC6.2. It is replaced by cirelationship table.<br>The upgrade routine converts each deviceparent record into a cirelationship record.<br>After the upgrade, ServiceCenter writes new records to the "cirelationship" table.<br>The upgrade utility does not delete your pre-existing deviceparent records. It converts fields from deviceparent to fields in cirelationship as follows:<br>• Field Name in deviceparent -> Field Name in cirelationship<br>• parent ->logical.name<br>• logical.name ->1 in related.cis<br>• relationship ->relationship.type<br>• port.no ->port.no<br>• comments ->comments<br>• logical.name ->relationship.name<br>• outage.dependency ->True<br>• outage.threshold ->1 | After you convert data to cirelationship, check each record and make sure information converted correctly.<br>Do not purge the deviceparent table until all of the records have been converted successfully to the new table.<br>You can manually set fields that were not in the deviceparent, such as outage.dependency and outage.threshold.<br>The cirelationship table is one to many. You can consolidate multiple record in deviceparent into 1 record in cirelationship table. We provide a consolidation wizard. You can launch the wizard by selecting Consolidate Relationships from the options menu when viewing a CI. |
| SC6.2 | Key changes | The globallists table has a new Nulls & Duplicates key.<br>• build.startup | For your information; no action required. |
| SC6.2 | Key changes | The screlation table has a new Nulls & Duplicates key.<br>• cartItemId | For your information; no action required. |
| SC6.2 | Key changes | The operator table has a new Nulls & Duplicates key.<br>• contact.name<br>**Note:** This change is required. | Make the required key change to the operator table. |
| SC6.2 | Key changes | The cm3t table has new Nulls & Duplicates keys.<br>• header, parent.change<br>• header, number | For your information; no action required. |
| SC6.2 | Key changes | The incidents table has new Nulls & Duplicates keys.<br>• initial.impact<br>• severity<br>• priority.code | For your information; no action required. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6.2 | Key changes | The ocmgroups table has a new Nulls & Duplicates key.<br>• members | Make the required key change to the ocmgroups table. |
| SC6.2 | Key changes | The probsummary table has a new Nulls & Duplicates key.<br>• severity | For your information; no action required. |
| SC6.2 | Key changes | The probsummary table has new Nulls & Duplicates keys.<br>• priority.code<br>• severity<br>• flag<br>• initial.impact | For your information; no action required. |
| SC6.2 | Key changes | The location table has a new Nulls & Duplicates key.<br>• updated.by | For your information; no action required. |
| SC6.2 | Knowledge Management | The Upgrade utility adds these records to these files.<br>cm3rcategory:<br>• KM Document<br>cm3rcatphase:<br>• KM 1 Triage<br>• KM 2 Revise<br>• KM 3 Review Read-only<br>• KM 4 Conclude<br>**Note:** These records are required for the Change Management portion of Knowledge Management to run. | For your information; no action required.<br>The ScriptLibrary file named "KMChangeManagement" hard-codes the category "KM Document" and the cm3rcatphase "KM 1 Triage". Change this ScriptLibrary file if you change the names.<br>The RAD section of several process records named `cm.km.`<xxx> set the phase pointer to 4 and the phase to "KM Decision". Change these records if you change the number of phases or the final phase in the "KM Document" Knowledge Management category. |

| Version | Feature | Change | Required action |
|---|---|---|---|
| SC6.2 | probsummary table | The Upgrade utility renames the severity.code field to severity so that the three fields used for the ITIL priority calculations are consistent across modules.<br>The Upgrade utility aliases the original severity.code field to severity. Because the field has been aliased, any format control etc. applied to the old data will still work.<br>If the severity field already exists in the dbdict, the upgrade does not add it as an alias. | Unless you had a previously existing severity field in probsummary, no action is required.<br>If you had a previously existing severity field and you do not want to use the new functionality, you should do one of the following.<br>A. Remove the calculation $L.pri.calc= jscall("PriorityCalc.getPriorityCalc") from the following new Process records:<br>• im.first<br>• im.save<br>• im.close<br>— or —<br>B. Modify the "PriorityCalc" ScriptLibrary. If you choose to update the "PriorityCalc" java script, you should be aware that other modules also use that code to calculate priority from the Initial Impact and Urgency.<br>If you do not want to use the new ITIL based Priority calculation functionality, it is easy to remove the calculation from the Process record.<br>If you do not remove the calculation, it will attempt to run using what ever information is in the severity field, which could cause unpredictable behavior. |
| SC6.2 | Service Catalog | The "Service Catalog Approval" ApprovalDef record is now a system record.<br>**Note:** Do not change the name of this record. | The upgrade utility adds this record but does not populate any members.<br>You must add the appropriate members. |

# Index