

HP Service Activator

Migration Guide

Version 5.0 to 5.1

Edition: V51-1A

**for Microsoft Windows® Server 2008 R2, HP-UX 11i v3, Solaris 10,
and Red Hat Enterprise Linux 5.4 operating systems**



Manufacturing Part Number: None

July 2, 2010

© Copyright 2001-2010 Hewlett-Packard Development Company, L.P.

Legal Notices

Warranty.

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

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

Restricted Rights Legend.

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

Hewlett-Packard Company
United States of America

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

Copyright Notices.

©Copyright 2001-2010 Hewlett-Packard Development Company, L.P., all rights reserved.

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

Trademark Notices.

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Linux is a U.S. registered trademark of Linus Torvalds

Microsoft® is a U.S. registered trademark of Microsoft Corporation.

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

UNIX® is a registered trademark of the Open Group.

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

All other product names are the property of their respective trademark or service mark holders and are hereby acknowledged.

Document id: HPSA p158-pd002104

Contents

1 Migration Overview	7
2 Migration Tasks by Component	8
Database	8
Workflows	8
Inventory	9
Inventory Trees	9
Plug-ins	9
Compound Tasks	9
Custom JSPs	9
Solution	10
3 Migration Tasks	11
1. Install the New Version of Service Activator	11
2. Copy the Solution Files	12
3. Update the Solution Files on the New Server	12
Service Activator RMI Interface	12
Custom Workflow Nodes and Handlers	12
Custom Workflow Module	12
Workflows	12
Inventory	13
Inventory Trees	16
Plug-ins	16
Compound Tasks	16
Custom JSPs	16
Customized AskFor Node JSPs	16
Solution	17
4. Migrate the System and Inventory Databases	17
5. Configure the New Version of Service Activator	18
6. Deploy Your Solution	18
7. Shut Down the Old Version of Service Activator	18
8. Move the License File	19
9. Start the New Version of Service Activator	19

Install Location Descriptors

The following names are used to define install locations throughout this guide.

Descriptor	What the Descriptor Represents
<code>\$ACTIVATOR_OPT</code>	The base install location of Service Activator. The UNIX® location is <code>/opt/OV/ServiceActivator</code> The Windows® location is <code><install drive>:\HP\OpenView\ServiceActivator</code>
<code>\$ACTIVATOR_ETC</code>	The install location of specific Service Activator files. The UNIX location is <code>/etc/opt/OV/ServiceActivator</code> The Windows location is <code><install drive>:\HP\OpenView\ServiceActivator\etc</code>
<code>\$ACTIVATOR_VAR</code>	The install location of specific Service Activator files. The UNIX location is <code>/var/opt/OV/ServiceActivator</code> The Windows location is <code><install drive>:\HP\OpenView\ServiceActivator\var</code>
<code>\$ACTIVATOR_BIN</code>	The install location of specific Service Activator files. The UNIX location is <code>/opt/OV/ServiceActivator/bin</code> The Windows location is <code><install drive>:\HP\OpenView\ServiceActivator\bin</code>
<code>\$ACTIVATOR_THIRD_PARTY</code>	The location for new Java™ components such as workflow nodes and modules. The UNIX location is <code>/opt/OV/ServiceActivator/3rd-party</code> The Windows location is <code><install drive>:\HP\OpenView\Service Activator\3rd-party</code> Customized inventory files are stored in the following locations: UNIX: <code>\$ACTIVATOR_THIRD_PARTY/inventory</code> Windows: <code>\$ACTIVATOR_THIRD_PARTY\inventory</code>
<code>\$JBOSS_HOME</code>	The install location for JBoss. The UNIX location is <code>/opt/HP/jboss</code> The Windows location is <code><install drive>:\HP\jboss</code>
<code>\$JBOSS_DEPLOY</code>	The install location of the Service Activator J2EE components. The UNIX location is <code>/opt/HP/jboss/server/default/deploy</code> The Windows location is <code><install drive>:\HP\jboss\server\default\deploy</code>
<code>\$SOLUTION_HOME</code>	The directory in which your solution resides: <code>\$ACTIVATOR_OPT/SolutionName</code>

In This Guide

This guide gives instructions on migrating your HP Service Activator solution from version 5.0 to version 5.1. The guide lists the changes that have been made in version 5.1 and explains how those changes affect the migration process.

First of all, read the Migration Overview sections in this guide to acquaint yourself with the migration issues. Then, when reading the Migration Tasks section, notice how those issues influence your migration tasks.

Before starting the migration process, read the *Release Notes* for HP Service Activator 5.1. Also read the *Installation Guide* to become familiar with the installing and configuring Service Activator 5.1 as well as the document *Solution Separation and the Deployment Manager* to get a full overview of Service Activator solutions. The documents are available either in the `Doc/` directory on your Service Activator DVD or in the `$ACTIVATOR_OPT/docs/` directory of the installed product.

If you wish to migrate your solution from Service Activator 4.1, you should also read the *HP Service Activator 4.1 to 5.0 Migration Guide*. This document is available in the `Doc/` directory on your Service Activator DVD.

Please note that the operating systems HP-UX 11i v2, Red Hat Enterprise Linux 5.2, and Windows Server 2003 are no longer supported. Instead Service Activator 5.1 supports HP-UX 11i v3, Red Hat Enterprise Linux 5.4, and Windows Server 2008 R2. In addition, Service Activator adds support for Oracle 11g and Oracle 11g RAC; Oracle 10g is still supported.

A number of file/directory locations are specified in this guide as environment variables, e.g. `$ACTIVATOR_OPT`, `$SOLUTION_HOME`, etc. You can find the definition for these install locations in the “Install Location Descriptors” section.

Audience

The audience for this guide is the Solutions Integrator who has developed the Service Activator 5.0 solution and wishes to migrate it to Service Activator 5.1.

1 Migration Overview

Migration is a multi-staged manual process that requires careful planning. There are several steps to follow to ensure the successful migration of a customized solution. These steps will be covered in more detail later in this document:

- Develop an understanding of whether the components of an earlier version of Service Activator must be modified in order to work correctly in a new version.
- Record and understand the customizations and configurations that have been made in an earlier version in order to properly migrate these customizations to a new installation.
- Install the new version of Service Activator on new hardware using an evaluation license.
- Migrate all components of the customer specific solution that have been applied to the earlier installation Service Activator.
- Export the content of the old Oracle DB and import it into the new Oracle DB and run the migration script in order to make the DB schema compatible with HP Service Activator 5.1. You can skip this step, if you don't wish to reuse the content of the old database or if you have your own procedure to migrate the solution data.
- Careful testing and planning of your deployment in production environment.
- Shut down the earlier version of Service Activator.
- Move the license file used by the earlier version of Service Activator to the new installation.
- Start the new version of Service Activator.

NOTE 1 This guide assumes that the solution has been deployed using the Deployment Manager; i.e. all solution components are located in the `$SOLUTION_HOME` directory.

NOTE 2 Migration to a newer version of Oracle is outside the scope of this document.

2 Migration Tasks by Component

The following sections break down the migration tasks by Service Activator component. These sections will point out the information you will need to record and consider carefully before migrating your solution.

Database

In Service Activator 5.1 the Oracle database play a major role. The schema and the contents of the Oracle database have changed since Service Activator 5.0; hence, your old database needs to be modified if you wish to reuse them in your new installation. HP Service Activator 5.1 comes with an SQL script that can be used to migrate a Service Activator 5.0 system database and the latest Hotfix to version 5.1.

NOTE

The database migration script requires that you have installed Service Pack V50-1C and Hotfix V50-1C-6 for HP Service Activator 5.0; otherwise it will not work. Also notice that there is no script to roll back a Service Activator 5.1 system database to version 5.0. Hence, you should make sure to make a backup of the system database before attempting to migrate it to version 5.1.

It is recommended that you create a new Oracle instance and/or a new Oracle user to be used by Service Activator 5.1 and then decommission your old database after you have completed all migration tasks.

Migrating service data from the old resource tables to the new database is not a major migration project, however it must be planned.

Workflows

In Service Activator 5.1 running workflows are by default persisted to the system database (as in Service Activator 5.0). Service Activator 5.0 provided new means of performance optimizing your workflows through “dirty flagging” and enabling/disabling workflow persistence at the workflow node level.

Service Activator 5.1 comes with two additional ways to performance optimize your workflows:

- Multiple workflow nodes can be executed in the same Java worker thread.
- A new activate node called `ActivateAtomicTask` that is limited to handing atomic activations has been added.

Custom Workflow Manager Modules as well as custom workflow nodes and handlers that have been developed for Service Activator 5.0 can be used directly with Service Activator 5.1, *except* for custom authentication modules.

Inventory

The inventory subsystem in Service Activator 5.1 has a lot of new and advanced features. Even though backward compatibility has always been a focal point, there are a few changes in the new inventory subsystem that may require that you make simple changes to your inventory resource definition files.

The preferred method to migrate your inventory system is to reuse only the resource definition files and then go through the following steps:

- Run the Inventory Builder with the “migrate from version 5.0” option on your Service Activator 5.0 resource definition files. This will generate an SQL script that you can execute on the inventory database to migrate your inventory data from version 5.0 to 5.1.
- Edit the resource definition files to make them compatible with the Service Activator 5.1 inventory subsystem. Note that the length of the solution name must not exceed 8 characters.
- Use the Deployment Manager to regenerate all resource files (Java beans, JSQLs, JSPs, SQL, Struts classes, and Struts configuration files) and to deploy them into their proper locations in the Service Activator 5.1 runtime system.

If you have customized the inventory resource files (e.g. JSPs or .java files), then the recommended procedure is:

- Study the *Service Activator Inventory Subsystem* document in details and determine whether you can leverage on new features instead of having to do manual customizations.
- If you still need to make customizations to the auto-generated JSPs, Struts files, etc. you need to run the Inventory Builder from command line to generate all resource files.
- You can now perform your customizations before using the Deployment Manager to deploy the resource files in the runtime system.

NOTE

If you need to make customizations, you should be acquainted with the Apache Struts framework since this is used extensively in the inventory subsystem (as in Service Activator 5.0).

Inventory Trees

Except for a solution name length limitation of 8 characters, you can use your inventory trees from Service Activator 5.0 directly in Service Activator 5.1.

Plug-ins

Plug-ins can be reused without changes. However, please note that the GenericCLI plug-in now stores activation dialog data in the DATABASE_MESSAGE table as a Java String object (in a BLOB) instead of using byte arrays.

Compound Tasks

Compound tasks can be reused without changes.

Custom JSPs

HP Service Activator 5.1 comes with support for automatically logging out users due to inactivity. Custom JSPs should be updated to support this functionality.

Solution

In order to make your solution deployable on Service Activator 5.1 you need to make a few changes to the solution's deployment descriptor and version file. In addition, you need to update the DTD files with their Service Activator 5.1 counterparts.

3 Migration Tasks

This chapter is grouped into nine major steps:

1. Install the new version of Service Activator on new hardware using an evaluation license.
2. Copy the solution files, i.e. all files from the `$SOLUTION_HOME` directory, from the old server to the `$SOLUTION_HOME` directory of the new server.
3. Update the solution files on the new server so that they are compatible with Service Activator 5.1.
4. Migrate your old system database as well as the old inventory database so that they work with Service Activator 5.1. (Optional)
5. Configure the new version of Service Activator.
6. Deploy your solution on the new version of Service Activator.
7. Shut down the earlier version of Service Activator.
8. Move the license file of the earlier version to the new installation.
9. Start the new version of Service Activator and test your solution.

NOTE 1 Step 4 can be skipped if you do not want to migrate your old data or if you have defined other processes for migrating your data.

NOTE 2 If you are running your solution in a cluster environment, you need to deploy the modified solution on all your cluster nodes. However, you should only migrate the system and inventory databases once.

1. Install the New Version of Service Activator

Install Service Activator 5.1 on new hardware following the instructions in the *Installation Guide*. This includes installing Java SE Development Kit 5.0 and creating a new Oracle database instance and/or user.

Install an evaluation license during the install phase. The evaluation license is valid for 180 days.

When installing the new version of Service Activator, you will need to supply the username, password, and database hostname, and database instance name values to `ActivatorConfig`. Make sure that you use the new database instance and make sure that the “Create database tables” checkbox is selected since `ActivatorConfig` will not accept the database schema from an earlier version of Service Activator.

NOTE If you want to migrate the system database from Service Activator 5.0, you must not run `ActivatorConfig` as part of the installation process.

2. Copy the Solution Files

Copy all solution files from the old server to the new server using the following steps:

- Create a `$(SOLUTION_HOME)` directory on the new server. The name of the directory must match that of the solution and its length must not exceed 8 characters.
- Copy all files from `$(SOLUTION_HOME)` directory on the old server to the `$(SOLUTION_HOME)` directory on the new server

3. Update the Solution Files on the New Server

This section describes at a high-level the steps that you need to follow in order to migrate your solution from Service Activator 5.0 to version 5.1. The solution components that do *not* need any modifications will not be mentioned.

Service Activator RMI Interface

The Java RMI interface in Service Activator 5.1 is backward compatible with Service Activator 5.0, except for the User Management methods. If your solution makes use of the User Management RMI interface, you need to study the Javadoc for the `com.hp.ov.activator.mwfm.UserManagementManager` interface and make the necessary changes to your solution.

Custom Workflow Nodes and Handlers

You can reuse your own custom workflow nodes and handlers with Service Activator 5.1. They will be deployed to the runtime system as part of the Deployment Manager's solution deployment process.

Custom Workflow Module

You can reuse your own custom workflow modules with Service Activator 5.1, albeit with the following exception that the interface for authentication modules has changed. If you have developed a custom authentication module you need to study the Javadocs for `com.hp.ov.activator.mwfm.engine.rmi.AdvancedAuthModule` and `com.hp.ov.activator.mwfm.UserManagementManager` as well as the Chapter 8, *Writing New Workflow Modules* in the document *Workflows and the Workflow Manager* and make the necessary changes in your authentication module.

Workflows

Your workflows from Service Activator 5.0 can be reused directly in Service Activator 5.1. If you migrate the system database using the procedure described later in this chapter, your workflows will be migrated automatically. Otherwise, your workflows will be deployed in the new system database as part for the Deployment Managers solution deployment process.

You may wish to take advantage to the two performance improvements that have been added in Service Activator 5.1:

- In Service Activator 5.0, the Workflow Manager only allowed a single workflow node to be processed before the worker thread was passed back into the thread pool. However, in Service Activator 5.1 a new element called `<Max-Nodes-Per-Thread>` has been added to the `$(ACTIVATOR_ETC)/config/mwfm.xml` file through with the number maximum number of workflow nodes per worker thread can be configured. Please note that the worker thread is always passed back to the pool when a workflow node is persisted and when a workflow jobs is placed on a request queue (for instance, during activations or when an AskFor node is used).

The default value of the `<Max-Nodes-Per-Thread>` parameter is 5; you may consider increasing it if it makes sense for your solution.

- The use of the transaction manager is only strictly necessary when calling compound tasks. Therefore, a new activate node called `ActivateAtomicTask` that is limited to handing atomic activations has been added. The advantage of the `ActivateAtomicTask` node, as opposed to the `Activate` node, is that it saves times by completely bypassing the transaction manager. To take advantage of this performance improvement, go through all your workflows and use the `ActivateAtomicTask` node instead of the `Activate` node in all places where the workflow call and atomic task.

Inventory

The Inventory Subsystem in Service Activator 5.1 has changed since version 5.0. This section describes at a high level, how to migrate your inventory subsystem from Service Activator 5.0 to version 5.1.

Database Schema Changes

A new column named `isParent__` has been added to all inventory tables. The column is needed by the new “create partial” and “delete partial” features that is new in Service Activator 5.1. The possible values of `isParent__` are 0 and 1.

In addition the column named `uniqueId__` has been deleted since it is not used anymore.

Resource Reservation Changes

The resource reservation mechanism has changed from Service Activator 5.0. In Service Activator 5.0 a column named `count__` was decreased by one each time a resource was reserved. In Service Activator 5.1 the mechanism has been reversed so that the `count__` is *increased* each time a resource is reserved.

Inventory Database Migration Script

The Service Activator 5.1 Inventory Builder can create an SQL script that can be used to automatically make the necessary schema changes as well as change the values in the `count__` column of reservable resources so that they conform to the new reservation mechanism.

To generate the inventory migration SQL script, you need to run the Inventory Builder with the *original* Service Activator 5.0 resource definition files using the `-mig50` option. See the following example:

```
InventoryBuilder -mig50 -app /tmp_inv *.xml
```

This will cause the Inventory Builder to generate a directory called `mig50` in the `/tmp_inv` directory that contains the migration SQL script.

NOTE

Be careful when you select the output (`-app`) directory. The Inventory Builder will delete the directory before generating the inventory migration SQL script.

Inventory Population scripts

If your solution contains inventory population scripts, you need to update them in the following way:

- Remove all references to the `uniqueId__` column.
- Alter all insert statement so that they set the value of the `isParent__` column to appropriate values (0 or 1).

Sequence Name Changes

The sequence name generation mechanism in the new Inventory Builder has been modified because there was a risk of sequence name conflicts in the old inventory builder (i.e. the sequence names were not always unique).

Sequence Name Migration

A new attribute on the <Field> element called `sequenceName` has been added to make it possible to explicitly specify the name of the sequence. In this way, you can ensure that sequence names will be the same as in Service Activator 5.0.

Database Table Reference Changes

In Service Activator 5.1 all SELECT statements are generated with an alias associated to every table. In the previous version some tables have not an alias associated and were referenced by using the table name instead of the table alias. See the following example:

- Example from Service Activator 5.0:

```
SELECT
  [...]
FROM
  Test_NetworkElement,
  Test_Router,
  Test_Network
WHERE
  [...]
```

- Example from Service Activator 5.1:

```
SELECT
  [...]
FROM
  Test_NetworkElement Demo#NetworkElement,
  Test_Router          Demo#Router,
  Test_Network         Demo#NetworkLnkRouter2MAN
WHERE
  [...]
```

The Service Activator 5.0 implementation can lead to problems when, for example, two different solutions define the same table names. This can be a big problem since all the references to this table (at least in one of the solutions) need to be changed.

By using aliases instead of table names, no references have to be changed since the database alias can remain the same, which makes it easier to ensure conflict-free coexistence of multiple solutions.

Database Table Reference Migration

- Own and Parents' tables:
 - In Service Activator 5.0, the references to the bean tables were simply the value of the <DBTable> element. In Service Activator tables are referenced using the value of the <DBAlias> element. If the <DBAlias> element is not present, the value will be set to `SolutionName#BeanName`. Note the <DBAlias> must be set if the `BeanName` is more than 18 characters long.
 - The value of the <DBAlias> element can be the same as the value of the <DBTable> element if you do not wish to change its references. This will work and continue doing so even if the value of the <DBTable> element is changed.

- Example:

```

SELECT
  [...]
FROM
  Test_NetworkElement TEST_NETWORKELEMENT,
  Test_Router          TEST_ROUTER,
  Test_Network         Demo#NetworkLnkRouter2MAN
WHERE
  [...]

```

- The same value of the <DBAlias> element will be used for the historic SELECT statements.
- External direct tables:
 - External direct tables are tables referenced in a key using `externalBean` attributes without <JoinBridges> elements. The generation algorithm is complex so if a reference is needed the recommended way is to use the attribute `alias` in this key to set the value.
- External indirect tables:
 - External indirect tables are tables referenced in a key using <JoinBridges> elements and each one defined in the <Jump> tags.
 - The destination table can be referenced just like external direct tables (i.e. using the `alias` attribute). This alias will also be used to generate the alias of the <Jump> tables. These aliases will be `aliasStepJump_number`. If one of the jumps inherits from another bean (and it's needed in the route to the target table) the alias of the parent table will be the same as the child with the `String Parent` as suffix.

JoinBridges Jumps Changes

The validation of the <Jump> entries in the <JoinBridge> has been modified to avoid some undesired situations. In Service Activator 5.0 it was allowed to set the value of a <Jump> element to the parent of the previous <Jump> element without a foreign key joining these beans. This was not a problem when jumping using the foreign key as was clear that the bean was another instance of the previous one and had to be in the <Jump> definition.

Without a foreign key this kind of <Jump> definition is unnecessary because the Inventory Builder already knows which bean is the parent of every bean, and it can use this to find the route to the destination.

JoinBridges Jump Migration

The migration task must be done manually. You need to think in entire entities and have to bear in mind that the parent bean is part of the child bean. So to jump from A to D through B and C the definition must be:

```

<JoinBridge origin="A" destination="D">
  <Jump>B</Jump>
  <Jump>C</Jump>
</JoinBridge>

```

No matter whether B has a parent and this parent has the foreign key to C, B has a relationship with C from the user's point of view.

If the middle table between the two tables is the parent of one of the two, *no* <JoinBridge> will be defined. The Inventory Builder is intelligent enough to find the path automatically.

Inventory Bean Method Changes

A number of inventory bean methods that existed in Service Activator 5.0 do not exist anymore. Most prominently, the `findByAttributes`, the `findbyarray`, and a number of advanced search methods have been removed.

Inventory Trees

If you migrate the system database from Service Activator 5.0 to version 5.1, the inventory trees will be migrated automatically. However, if the solution names in the Service Activator 5.0 inventory trees exceed 8 characters, you will need to use the Inventory Tree Deployer or the Inventory Tree Designer to undeploy your old inventory trees. After this you can deploy your new inventory trees as part of the general solution deployment process by running the Deployment Manager.

NOTE

If you undeploy your old inventory trees, you will need to recreate all role mappings to tree, operation types, and branch types.

Plug-ins

Plug-ins can be reused without changes. If you migrate the system database from Service Activator 5.0 to version 5.1, the plug-ins will be migrated automatically. Alternatively, your plug-ins can be deployed using the Deployment Manager as part of the solution deployment process.

Please note that the GenericCLI plug-in now stores activation dialog data in the `DATABASE_MESSAGE` table as a Java String object (in a BLOB) instead of using byte arrays. The system database migration script does not convert byte arrays to Java String objects. However, this will not have an impact unless you access the `DATABASE_MESSAGE` table directly.

Compound Tasks

Compound tasks can be reused without changes. If you migrate the system database from Service Activator 5.0 to version 5.1, the compound tasks will be migrated automatically. Alternatively, your compound tasks can be deployed using the Deployment Manager as part of the solution deployment process.

Custom JSPs

HP Service Activator 5.1 comes with support for automatically logging out users due to inactivity. Custom JSPs should add the following in order to be able to reset the logout timers if the user is active (i.e. on mouse or keyboard activity):

- Include the JavaScript `/activator/javascript/saUtilities.js` in your JSPs.
- Add the attributes `onmousemove="logoutTimerReset();"` and `onkeydown="logoutTimerReset();"` to the `<body>` element in your JSPs.

Customized AskFor Node JSPs

No changes are required for the customized AskFor node JSPs. Your customized AskFor node JSPs will be copied to the proper location as part of the Deployment Manager's solution deploy process.

If you are using customized AskFor JSPs you need to edit the configuration file `$_JBASS_DEPLOY/hpovact.sar/activator.war/WEB-INF/web.xml` and set the value of the `customizeAskForNodeJSP` to `true`.

Solution

In order to make your solution deployable on Service Activator 5.1 need to make the following changes:

- As solution name limitation of 8 characters is now enforced by Service Activator. Hence, if your solution name length exceeds 8 characters, you need to limit the solution name in the deployment descriptor as well as in the `version.xml` file. Please note that the presence of a `version.xml` file is now mandatory for solutions.
- You also need to go through all your workflows, inventory resource definition files, and inventory trees and ensure that the solution name does not exceed 8 characters.
- The name of the `<DTD>` element in the solution deployment descriptor has changed to `<Config>`. If you use this element in your deployment descriptors, you need to change it to `<Config>`.
- The `<Backup>` element is no longer used by the Deployment Manager and may be removed from the deployment descriptor.

Finally, since the files in the `$SOLUTION_HOME` directory on your new server are a copy of the solution, you need to do the following:

- Copy `$ACTIVATOR_ETC/config/deploy.dtd` and `$ACTIVATOR_ETC/config/version.dtd` to `$SOLUTION_HOME`.
- Copy `$ACTIVATOR_ETC/workflows/workflow.dtd` to `$SOLUTION_HOME/etc/workflows`.
- Copy `$ACTIVATOR_ETC/config/inventoryTree/inventoryTree.dtd` to `$SOLUTION_HOME/etc/config/inventoryTree`.
- Copy `$ACTIVATOR_ETC/config/bean.dtd` to `$SOLUTION_HOME/inventory`.

NOTE

Not all DTDs are strictly necessary for the solution to be deployable. However, for your own convenience you should ensure that all Service Activator 5.0 DTDs are replaced with their Service Activator 5.1 counterparts.

4. Migrate the System and Inventory Databases

This step is optional. If you do not wish to reuse your old data or if you have defined your own procedure for migrating data, you can skip this step.

The Service Activator system database schema has changed from version 5.0 to 5.1. In order to migrate your system database, you need to perform the following steps:

- Ensure that you have installed Service Pack V50-1C as well as Hotfix V50-1C-6 on your old Service Activator server. If you running an earlier version of Service Activator the database migration SQL script will not work.
- Log into the old system database server and export the database to a file using standard Oracle utilities. You need to do this when no jobs are running and no scheduled jobs will be scheduled during the migration period.
- Log into the new system database server and import the file that you just exported.
- On the new database server, run the migration SQL script `$ACTIVATOR_OPT/migration/migrate.sql`.

Now, your Service Activator system database should be ready for Service Activator 5.1.

In order to migrate your inventory database, you need to perform the following steps:

- Log into the old inventory database server and export the database to a file using standard Oracle utilities.
- Log into the new inventory database server and import the file that you just exported.
- On the new database server, run the inventory migration SQL script that you generated using the Inventory Builder.

Now, your Service Activator inventory database should be ready for Service Activator 5.1.

NOTE

If your system and inventory data are stored in the same database (which is frequently the case) you only need to run the export/import steps once.

5. Configure the New Version of Service Activator

If you have completed step 4 and migrated your databases from Service Activator 5.0 to 5.1 you will now need to run `ActivatorConfig` in order to generate new configuration files. You must remember to uncheck the “Create database tables” checkbox.

If, on the other hand, you have skipped step 4, you must now run `ActivatorConfig` and remember to *check* the “Create database tables” checkbox.

After having run `ActivatorConfig` you may want to make manual changes to Service Activator’s configuration files. This migration guide will not walk you through all Service Activator configuration files. However, the following configuration files are commonly modified in Service Activator solutions:

- `$ACTIVATOR_ETC/config/mwfm.xml`
- `$ACTIVATOR_ETC/config/resmgr.xml`
- `$ACTIVATOR_ETC/role_mappings.xml` (optional)
- `$JBOSS_DEPLOY/hpovact.sar/activator.war/WEB-INF/web.xml`

Go through all these files (as well as the other configuration files that you need to modify) and carefully migrate your configuration changes from the corresponding configuration files on the old Service Activator server.

6. Deploy Your Solution

After having modified your solution so that it is compatible with Service Activator 5.1, you can deploy it using the Deployment Manager’s `deploy solution` operation. If you are running Service Activator in a cluster environment, you need to deploy your solution on all cluster nodes. In this case you need to make sure that you only perform database operations (suchs as deployment of workflows, inventory tree, etc.) once.

7. Shut Down the Old Version of Service Activator

The recommended way to shut down Service Activator is as follows:

- Stop feeding new workflows into the Workflow Manager.
- Allow all running workflows to complete.

Now, you can stop the old version of Service Activator as follows:

- **HP-UX:** `/sbin/init.d/activator stop`
- **Linux & Solaris:** `/etc/init.d/activator stop`

- **Windows:** Stop the service called “HP Service Activator”

8. Move the License File

If you do not wish to obtain a new license you may use your existing Service Activator license. Copy the AutoPass license file *LicFile.dat* from your old Service Activator installation to the new installation. Use the AutoPass tools to import the license to make the license active. The AutoPass license file can be found in the location shown below:

- **UNIX:** `/var/opt/OV/HPOvLIC/LicFile.dat`
- **Windows:** `<drive>:\Program Files\Common Files\Hewlett-Packard\AutoPass\LicFile.dat`

9. Start the New Version of Service Activator

Start the new version of Service Activator as follows:

- **HP-UX:** `/sbin/init.d/activator start`
- **Linux & Solaris:** `/etc/init.d/activator start`
- **Windows:** Start the service called “HP Service Activator”

Finally, you should test the correct operation of your solution according to your test specification.